UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

MASTER'S THESIS

Exploiting limited signalling capabilities in autonomous agents: Robust and efficient signalling policies in search-and-rescue over uncertain environments

---

Author:
Vijeth Hebbar

Supervisor:
Prof. Cedric Langbort

Submitted in partial fulfillment of the requirements for the degree of Master of Science

in the

Department of Aerospace Engineering
Graduate College

June 18, 2020

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

Abstract

Faculty Name
Department of Aerospace Engineering

Master of Science

Exploiting limited signalling capabilities in autonomous agents: Robust and efficient signalling policies in search-and-rescue over uncertain environments

by Vijeth Hebbar

Search and rescue (SAR) operations are challenging in the absence of a medium of communication between the rescuers and the rescuee. Natural signalling, 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 signalling his intent to the rescuee.

Contents

Abstract ii

1 Introduction 2
1.0.1 Motivating Illustration . . . 3

2 Optimal Signalling Policy 4
2.1 Rescuee Policy . . . 4
2.2 Rescuer Optimal Policy . . . 5

3 Sensitivity 7
3.1 Sensitivity to Velocity . . . 7

4 Robust Optimal Signalling Policy 11
4.1 Robust Optimal Candidate Rendezvous Set . . . 12
4.2 Robust optimal signalling policy . . . 13
4.3 Case I: At-least one robust candidate rendezvous point . . . 15
4.4 Case II: No robust candidate rendezvous point . . . 16
4.5 Robustness to Velocity Variation . . . 16

A Proofs for results 17
A.1 Justifying the inequality 3.4 . . . 17

B Proof of Correctness: Algorithm 1 18

Bibliography 22

List of Figures

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. 3

3.1 Illustrating the sensitivity of the optimal signal to parameter k_v 7
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. 8
3.3 Shortest paths for the rescuee to both target goals. 8

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.- 20
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 $\{w_{ij}^{k'}\}$ 20

List of Tables

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. 7

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 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. There has been a rising interest in studying intent-expression and legibility of robotic motion in recent years. Dragan, Lee, and Srinivasa, 2013 and Dragan, Bauman, et al., 2015 studied legible motion for robotic arms in settings involving human-robot collaboration. Szafir, Mutlu, and Fong, 2014 studied the communicative ability of a UAV using modified trajectories, while in their later work (Szafir, Mutlu, and Fong, 2015) they looked at a more explicit medium of communication, using lights to convey directionality. Both these works illustrate the limited signalling 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 signalling mechanism. Here we wish to see how such signalling capabilities can be exploited to influence the rescuee into taking certain actions which benefit both the players.

Game theory has been used extensively to model human-robot interactions in recent years (Li et al., 2016 and Yua, Tsengb, and Langaria, 2018). The signalling 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. This paper discusses an approach to arrive at the optimal signal policy to be employed by the rescuer in such a game.

1.0.1 Motivating Illustration

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 ((blue 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 rescuer (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 (UAV) believes that the signal is observed and interpreted correctly by the rescuee; a reasonable assumption at that as the rescuee might expect the drone 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.

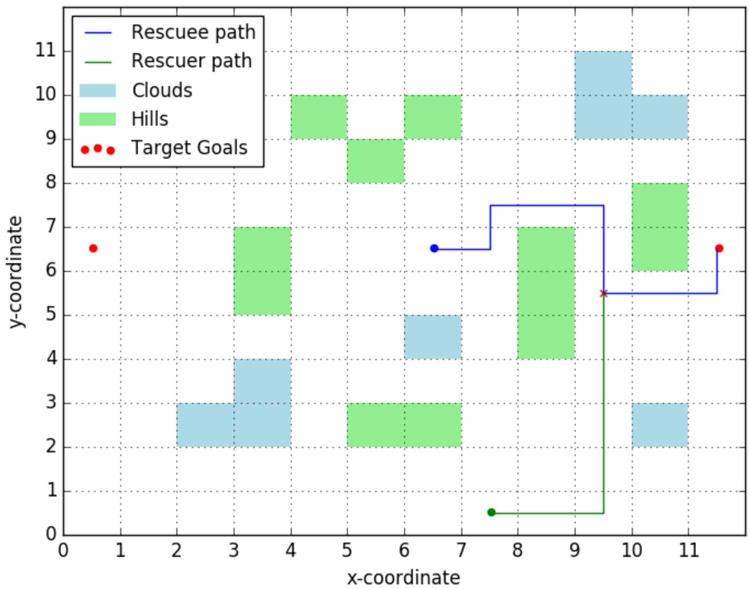


FIGURE 1.1: Rescue area topology. The two 'plain' regions (red dots) are used as target goals. Rescuee takes optimal path (blue) to perceived target of the rescuer. The rescuer picks an optimal rendezvous point (red cross) to meet the rescuee. Clouds (blue shading) act as obstacles to the UAV and hills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

Chapter 2

Optimal Signalling 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^r_{ij} and w^R_{ij} 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

$$U_r(m, P) = \phi_r(P) \quad P \in \mathcal{P}_{v_r \to v_m} \tag{2.1}$$

over paths P. 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 program,

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

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

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. In general, the solution to the shortest path problem is a directed sub-graph with every path in the sub-graph being a shortest path 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 $\mathcal{G}_m$.

Assumption 1 *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/she 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 $\mathcal{V}$. We 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 (E.g. ensuring flight path in a relatively safe environment). Accordingly, we define the cost function for the rescuer as,

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

where $v_x \in \mathcal{X}_m$ is the rendezvous point, $P^x_R \in \mathcal{P}_{v_R \to v_x}$ and $P^x_r \in \mathcal{P}_{v_r \to v_x}$. The chosen v_x and the corresponding paths P^x_r and P^x_R must satisfy the constraint

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

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 to reach the chosen rendezvous node v_x respectively. 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 1.

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),

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

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

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

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

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

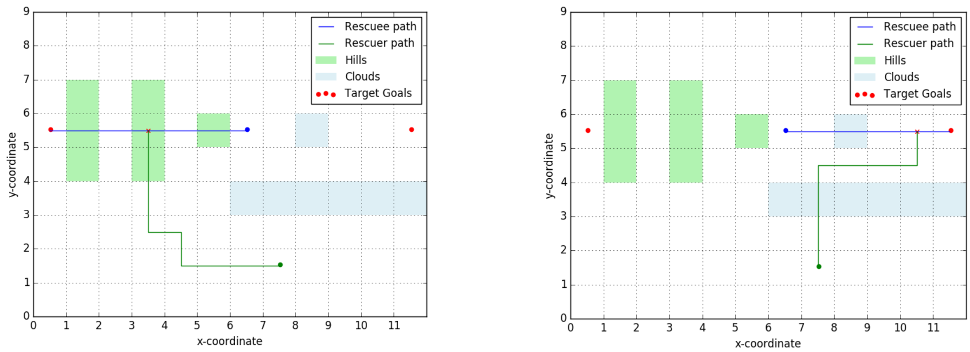
As indicated in Section 2.1 we can solve the linear programs described in (2.10) and (2.12) using either simplex methods or by implementing the Dijkstra's Algorithm (DA). Having solved the optimisation in (2.10) and (2.12) for each node in $\mathcal{X}_m$, the constrained optimisation in (2.6) can be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

Chapter 3

Sensitivity

3.1 Sensitivity to Velocity

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

Table 3.1 presents the the optimal signal m_{opt} and the cost U_R in sending that signal for various values of the velocity ratio k_v. 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/her there. But once the rescuee has traversed the hill and is passing through a region of low cost he/she 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 the general layout of the target goals and the locations of the rescuee and the rescuer as described in 3.3, to attain M switches in the optimal signal to be sent, we need a grid of dimension N, where N scales linearly with M.* $\square$

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 layout that will give us the sensitive behaviour we want.

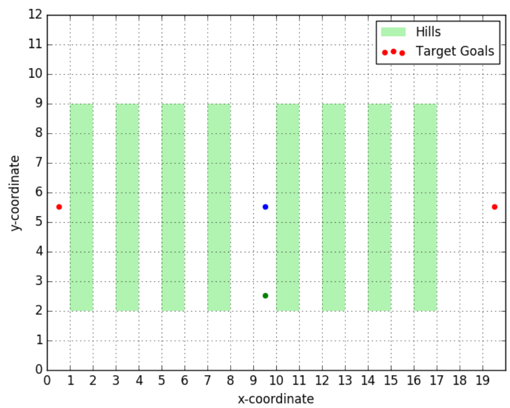


FIGURE 3.2: A possible terrain to obtain multiple switches. The 'hill' regions consist of edges having thrice the cost to traverse compared to other regions. The initial location of the rescuer and the rescuee is given by the green and blue circles respectively.

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).

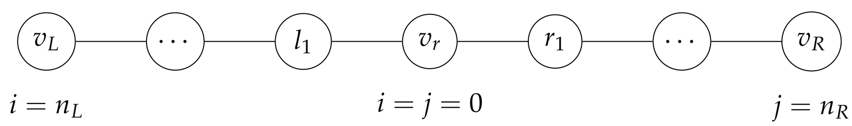


FIGURE 3.3: Shortest paths for the rescuee to both target goals.

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 let n_R denote the index and the number of nodes on the path going to R. $\{l_i\}$ and $\{j\}$ 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,

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

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

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 constant being 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.1) leads us to the following inequality for any even,

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

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

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 increase 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 signalling 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 signalling policy that is robust to uncertainty of velocity of players.

Chapter 4

Robust Optimal Signalling Policy

For this section of the work, we will assume that the edge weights w^r_{ij} and w^R_{ij} are bounded random variables such that $\underline{w}^r_{ij} \leq w^r_{ij} \leq \overline{w}^r_{ij}$ and $\underline{w}^R_{ij} \leq w^R_{ij} \leq \overline{w}^R_{ij}$. All edge weights are assumed to mutually independent. The set of edge weights $\{w^r_{ij}\}$ and $\{w^R_{ij}\}$ are supported over bounded sets Ω^r and Ω^R respectively.

We make an observation that in the analysis presented in Section 2 the edge weights only show up when we seek to find the shortest paths over the graph. There can be multiple approaches in arriving at the signalling policy in this new framework with stochastic shortest routes. 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. Sigal, Pritsker, and Solberg, 1980 and Frank, 1969 proposed methods to maximise the probability that a certain path realizes the least weight. Such probabilistic approaches are reasonable when we are running the stochastic scenario over multiple iterations and seek only to minimize the expected cost over the runs. However, in success critical problems such as our rescue scenario, we wish to be completely risk-averse.

We presented the linear program formulation of the shortest path problem in Section 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, Ghaoui, and Nemirovski, 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, Ghaoui, and Nemirovski, 2009 we have the following notions.

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

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

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 *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*

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

Definition 5 *The Robust Counterpart of the uncertain optimisation problem $O_{\mathcal{U}}$ is the optimization problem,*

$$min_x \{ \sup_{(w,v)\in\mathcal{U}} f(x,w) : g(x,v) < 0 \quad \forall (w,v) \in \mathcal{U} \} \tag{4.6}$$

where we are minimising over all the robust feasible solutions.

Formally we can pose our question as, "What signalling 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 stochastic graph. We define the set of *robust candidate rendezvous points*.

Definition 6 *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,*

$$\hat{\mathcal{X}}_m = \bigcap_{\{w^r_{ij}\}\in\Omega^r} \mathcal{X}^w_m \tag{4.7}$$

In Section 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 (4.7) 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". Here, we present a novel algorithm for obtaining this set

$\hat{\mathcal{X}}_m$.

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

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 above 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 signalling policy.

4.2 Robust optimal signalling policy

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

Assumption 2 *Let*

$$w_{\max}^{R} \triangleq \max_{ij} \overline{w}_{ij}^{R}$$

$$w_{\min}^{r} \triangleq \min_{ij} \underline{w}_{ij}^{r}$$

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

$$k_v \triangleq \frac{V_R}{V_r} \geq \frac{w_{\max}^{D}}{w_{\min}^{H}} \tag{4.8}$$

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 alone 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.

Definition 7 *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 2 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,*

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

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

Proof: 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 $\phi_R(\xi_R^*(v_R, v_{x,n})) \leq \phi_R(\xi_R^*(v_R, v_{x,m})) + \phi_R(\xi_R^*(v_{x,m}, v_{x,n}))$.

$$\begin{aligned} \implies \frac{\phi_R^*(v_{x,n})}{\phi_r^*(v_{x,n})} &\leq \frac{\phi_R(\xi_R^*(v_R, v_{x,m})) + \phi_R(\xi_r^*(v_{x,m}, v_{x,n}))}{\phi_r(\xi_r^*(v_r, v_{x,m})) + \phi_r(\xi_r^*(v_{x,m}, v_{x,n}))} (\because \xi_R^* \quad \text{is the shortest path}) \\ &\leq \frac{\phi_R(\xi_R^*(v_R, v_{x,m})) + w_{\max}^R(m-n)}{\phi_r(\xi_r^*(v_r, v_{x,m})) + w_{\min}^r(m-n)} \quad (\because w_{ij}^R \leq w_{\max}^R, w_{ij}^r \geq w_{\min}^r) \\ &\leq \frac{k_v \phi_r(\xi^*(v_r, v_{x,m})) + k_v w_{\min}^r(n-m)}{\phi_r(\xi^*(v_r, v_{x,m})) + w_{\min}^r(n-m)} \quad (\text{Assumption 2}) \\ &= k_v \end{aligned}$$

□

We basically showed that if a node is a robust candidate rendezvous point then all nodes in $\hat{\mathcal{X}}_m$ succeeding (ordered by Definition 7) 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 v_m is also a robust candidate rendezvous point.*

We are now equipped to analyse the problem of finding the optimal signalling 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

Then, for any $\{w^r_{ij}\} \in \Omega_r$ and $\{w^R_{ij}\} \in \Omega_R$ the constraint in (2.7) simplifies to

$$\phi^*_R(v_x) - k_v\phi^*_r(v_x) \leq 0 \tag{4.9}$$

Then we make a proposition,

Proposition 1 *Any robust feasible rendezvous point v_x for (4.9) satisfies,*

$$\phi^*_{R,\max}(v_x) - k_v\phi^*_{r,\min}(v_x) \leq 0 \tag{4.10}$$

Where,

$$\phi^*_{R,\max}(v_x) = \min_{x_{ij}\geq 0} \sum_{ij\in\mathcal{E}} \overline{w}^R_{ij} x_{ij} \tag{4.11}$$

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

$$\phi^*_{r,\min}(v_x) = \min_{x_{ij}\geq 0} \sum_{ij\in\mathcal{E}} \underline{w}^r_{ij} x_{ij} \tag{4.12}$$

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

□

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

$$\min_{m\in\mathcal{M}} \min_{v_x\in\mathcal{X}_m} k_1\phi^*_{R,\max}(v_x) + k_2\phi^*_{r,\max}(v_x) \tag{4.13}$$

subject to (4.10), where, $\phi^*_{r,\max}(v_x)$ can be obtained by replacing $\underline{w}^r_{ij}$ in (4.12) with $\overline{w}^r_{ij}$.

Motivated by Definition 5, the robust counterpart to the uncertain optimisation problem presented in (2.6) is given by,

$$\min_{m\in\mathcal{M}} \min_{v_x\in\mathcal{X}_m} \max_{w^R_{ij}, w^R_{ij}} k_1\phi^*_R(v_x) + k_2\phi^*_r(v_x) \tag{4.14a}$$

$$\text{S.T.} \quad \max_{w^R_{ij}, w^R_{ij}} \phi^*_R(v_x) - k_v\phi^*_r(v_x) \leq 0 \tag{4.14b}$$

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 4.10 then it satisfies (4.14b) for all values of w^r_{ij} and w^r_{ij}. In other words, such a solution is robust feasible by Definition 4.

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 (4.14) is given by (4.13) subject to (4.10).

4.4 Case II: No robust candidate rendezvous point

By Assumption 1, 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 the target goal node v_m. Thus the robust counterpart to (2.6) in this case is simply

$$\min_{m \in \mathcal{M}} \min_{v_x \in \mathcal{X}_m} k_1 \phi^*_{R,\max}(v_m) + k_2 \phi^*_{r,\max}(v_m) \tag{4.15}$$

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

To complete our discussion on the robust optimal signalling policy for the rescuer we will provide an algorithm to arrive at it.

4.5 Robustness to Velocity Variation

We will encompass the variation in velocity of the rescuer and the rescuee in variations of the parameter k_v. We assume that k_v is a random variable taking values over a range $[\underline{k}_v, \overline{k}_v]$. The objective function in (2.6) is unaffected by the 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,

$$\min_{m \in \mathcal{M}} \min_{v_x \in \mathcal{X}_m} k_1 \phi^*_R(v_x) + k_2 \phi^*_r(v_x) \tag{4.16}$$

$$\text{S.T.} \quad (\phi^*_R(v_x) - \underline{k}_v \phi^*_r(v_x)) \mathbb{1}_{v_x \neq v_m} \leq 0 \tag{4.17}$$

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 2 with respect to the edge weights and the parameter k_v can be done independently. For the subsequent robustification of the constraint in (4.17) with respect to edge-weight uncertainty, we only need the $\underline{k}_v$ to satisfy the constraint in Assumption 2.

Appendix A

Proofs for results

A.1 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 $V_R(l_{i+1}) > V_R(l_{i+2}) \quad \forall i \geq 4$.

Appendix B

Proof of Correctness: Algorithm 1

Definition 8 *(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 9 *(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 4 *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 4, we first present two propositions.

Proposition 2 *Consider the graph $\mathcal{G}^k$ and let $w_{ij}^{k'}$ be any set of edge-weights on $\mathcal{G}^k$ satisfying the property*

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

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 3 *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 4: 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 3 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 2 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 2 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 1 *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 $\mathcal{G}_2^k$. □

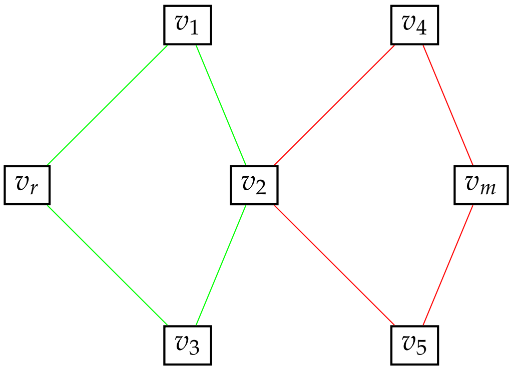


FIGURE B.1: A possible representation of $\mathcal{G}^1$. All edge weights are minimum. $\hat{\mathcal{X}}^1$ would contain $\{v_r, v_2, v_m\}$. v_2 here connects the two sub-graphs red and green.-

As a direct result from Lemma 1 we have,

Corollary 2

Proof for Proposition 2: 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.

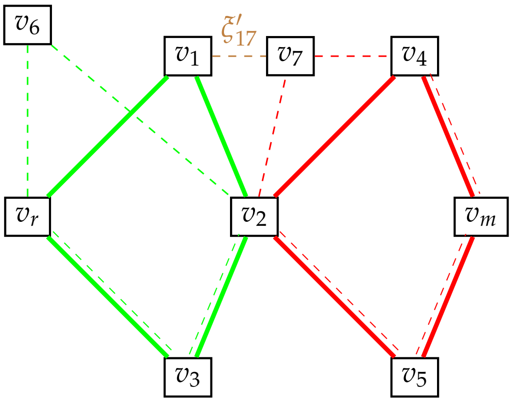


FIGURE B.2: A possible representation of $\mathcal{G}^k$. All thick edge weights are maximum and all thin edges are minimum weight. $\hat{\mathcal{X}}^k$ would contain $\{v_r, v_2, v_m\}$. v_2 here connects the two sub-graphs red and green. The dashed lines indicate edges in $\mathcal{G}^{*k}$ sub-graph. 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 $\{\xi'_{s_i\to t_i} : i \in [K]\}$ where K denotes the number of splices of ξ outside $\mathcal{G}^{k-1}$. We can consider a corresponding collection $\xi^*_{s\to t} = \{\xi^*_{s_i\to t_i} : i \in [K]\}$ of splices within the graph $\mathcal{G}^{k-1}$ and the same proof holds with minor changes in vocabulary used. □

Bibliography

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

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

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

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

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.

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.

— (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.* New York, NY, USA: Cambridge University Press.

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.