12-2020

# $\ell$-CTP: Utilizing Multiple Agents to Find Efficient Routes in Disrupted Networks

Andrew Alseth
*University of Arkansas, Fayetteville*

# $\ell$-CTP: Utilizing Multiple Agents to Find Efficient Routes in Disrupted Networks

A thesis submitted in partial fulfillment of
the requirements for the degree of
Master of Science in Industrial Engineering

by

Andrew Alseth
Missouri University of Science and Technology
Bachelor of Science in Chemical Engineering

December 2020
University of Arkansas

This thesis is approved for recommendation to the Graduate Council

Ashlea Bennett Milburn, Ph.D.
Thesis Chair

Burak Eksioglu, Ph.D.
Committee Member

Kelly Sullivan, Ph.D
Committee Member

**Abstract**

Recent hurricane seasons have demonstrated the need for more effective methods of coping with flooding of roadways. A key complaint of logistics managers is the lack of knowledge when developing routes for vehicles attempting to navigate through areas which may be flooded. In particular, it can be difficult to re-route large vehicles upon encountering a flooded roadway. We utilize the Canadian Traveller's Problem (CTP) to construct an online framework for utilizing multiple vehicles to discover low-cost paths through networks with failed edges unknown to one or more agents *a priori*. This thesis demonstrates the following results: first, we develop the $\ell$-CTP framework to extend a theoretically validated set of path planning policies for a single agent in combination with the iterative penalty method, which incentivizes a group of $\ell > 1$ agents to explore dissimilar paths on a graph between a common origin and destination. Second, we carry out simulations on random graphs to determine the impact of the addition of agents on the path cost found. Through statistical analysis of graphs of multiple sizes, we validate our technique against prior work and demonstrate that path cost can be modeled as an exponential decay function on the number of agents. Finally, we demonstrate that our approach can scale to large graphs, and the results found on random graphs hold for a simulation of the Houston metro area during hurricane Harvey.

**Acknowledgements**

First and foremost I acknowledge my advisor, Dr. Ashlea Bennett Milburn. She has molded me into the researcher that I have become to date, and provided an incredible environment for me to both explore my interests and refine my skills in her research group. It's been my honor to work with you.

I am thankful for the guidance and feedback from my committee - this thesis is much stronger for their input, and I appreciate their generosity of time in these most uncertain times.

Finally, I want to thank the Industrial Engineering department for helping me find my intellectual passions.

**Dedications**

I could not have begun this journey without the push from my wife Samantha - I am eternally grateful for the belief and patience you have in me. I am a much better human for building a life with you.

Many of the ideas in this thesis were formulated and refined in morning runs with Trek and Tahoe. Life is infinitely better with those two Australian Shepherds.

**Contents**


**1 Introduction** **1**

**2 Problem Formulation** **4**

**3 Literature Review** **6**
3.1 Single Agent CTP . . . . . 7
3.2 Multi-Agent CTP . . . . . 8

**4 Methodology** **10**
4.1 $\ell$-CTP instance . . . . . 11
4.1.1 Policy comparisons . . . . . 11
4.2 Agents . . . . . 12
4.3 Policies . . . . . 14
4.3.1 Optimist (OMT) . . . . . 15
4.3.2 Hindsight Optimist (HOP) . . . . . 15
4.4 Path Generation . . . . . 16
4.4.1 Generating Paths Using Policies and A* . . . . . 16
4.4.2 Using Iterative Penalty Method to Generate Disjoint Paths . . . . . 19

**5 Computational Study** **21**
5.1 Trials on Random Delaunay Graphs . . . . . 21
5.1.1 Design of Experiments . . . . . 21
5.1.2 Random Delaunay Graph Generation . . . . . 22
5.2 Trials on Random Euclidean Graphs . . . . . 22
5.3 Case Study for Hurricane Harvey . . . . . 23
5.4 Parameter Tuning . . . . . 25

**6 Results** **26**
6.1 Trials on Random Delaunay Graphs . . . . . 28
6.1.1 Equivalence between weathers . . . . . 30
6.1.2 Impact of $A^*$ planning on single agent using HOP . . . . . 34
6.1.3 Impact of Replanning Order . . . . . 36
6.1.4 Impact of number of Agents . . . . . 38
6.2 Trials on Random Euclidean Graphs . . . . . 41
6.3 Case Study for Hurricane Harvey . . . . . 45

**7 Conclusions** **48**

**8 Bibliography** **52**

**9 Appendix** **56**

**List of Figures**


1 Example Graph, HOP policy . . . . . 18

2 Delaunany triangulation using 100 randomly placed vertices [13] . . . . . 22

3 Harris County road network from Open Streetmap . . . . . 25

4 Tuning Penalties for $\ell$-CTP . . . . . 27

5 Decay parameter of OMT-EC . . . . . 39

6 Decay parameter of HOP-EC . . . . . 40

7 Impact of agents on shortest path costs, HOP-EC policy . . . . . 41

8 Impact of agents on shortest path costs, OMT-EC policy . . . . . 42

9 Comparison of performance of trials on Euclidean graphs . . . . . 43

10 Results of $\ell$-CTP instances on Houston road network . . . . . 46

11 Performance of OMT-EC on Houston road network . . . . . 47

**List of Tables**


1 OMT-EF average path lengths on random Delaunay graphs . . . . . 29

2 OMT-EF analysis of performance decrease across number of agents . . . . . 31

3 Two-sample Z-test between Eyerich *et al.* [15] and single agent OMT results. . . . 33

4 Two-sample Z-test between Eyerich *et al.* [15] and $\ell$-CTP single agent HOP results 35

5 Paired Student's t-test on OMT replanning variants across all graphs . . . . . 36

6 Paired Student's t-test on HOP replanning variants across all graphs . . . . . 38

7 Equations of fit lines, OMT-EC & HOP-EC . . . . . 40

8 Results of 100 weathers on HOU graph . . . . . 45

9 Equations of best fit lines on HOU . . . . . 45

10 Comparison of Competitive Ratio between instances of $\ell$-CTP and Shiri-Salman . . 56

11 Comparison of Competitive Ratio between instances of $\ell$-CTP and Shiri-Salman, cont'd . . . . . 57

12 OMT-NP average path lengths on random Delaunay graphs . . . . . 58

13 OMT-NP analysis of performance decrease across number of agents . . . . . 59

14 OMT-OO average path lengths on random Delaunay graphs . . . . . 60

15 OMT-OO analysis of performance decrease across number of agents . . . . . 61

16 OMT-EC average path lengths on random Delaunay graphs . . . . . 62

17 OMT-EC analysis of performance decrease across number of agents . . . . . 63

18 HOP-OO average path lengths on random Delaunay graphs . . . . . 64

19 HOP-OO analysis of performance decrease across number of agents . . . . . 65

20 HOP-EC average path lengths on random Delaunay graphs . . . . . 66

21 HOP-EC analysis of performance decrease across number of agents . . . . . 67

22 HOP-EF average path lengths on random Delaunay graphs . . . . . 68

23 HOP-EF analysis of performance decrease across number of agents . . . . . 69

## 1 Introduction

Since the 1980's, the United States has seen a drastic increase in billion-dollar (inflation-adjusted) disasters, from an overall average of 6.5 per year to 13.8 per year in the most recent decade (2010-2019) [32]. While flooding and tropical cyclones contribute to only 29.5% of the frequency of billion dollar disasters, they contribute to 62.2% of the total monetary costs and 53.3% of the deaths [32]. The trends in the warming of the Earth indicate that the potential exists for further increases in the severity of cyclones and similar events [20]. Smaller, nuisance flood events have increased in frequency for coastal cities from once every five years to once every three months in the last 50 years of coastal flood tracking [33]. In order to adapt to the changing climate and increase our capabilities to respond to disasters in the future, we must utilize our finite capability to respond in the most efficient and efficacious manner possible.

In disasters, infrastructure of varying types can be disrupted. These disruptions lead to deprivation of critical resources such as electricity or water [25]. Certain vulnerable populations are more sensitive to these types of disruptions, including patients in hospitals and displaced populations in shelters. Without access to electricity, patients that are critically ill and sustained by equipment rely upon backup generators [35]. Water infrastructure is particularly vulnerable to disasters due to effects such as loss of power or contamination of water sources. As such, sheltered populations in areas that experience disruption in water utilities require water to be brought in from outside the disaster impact area. In both the described cases, ensuring affected populations have access to fuel and water via truckload is critically important to preventing long-term injury or death due to deprivation.

Flooding and water-related disasters are major disruptors of road networks and directly inhibit the ability for ground-based operations in impacted areas [12, 17]. Unlike earthquakes and damage due to wind which require extensive debris removal operations to return roads to a passable state, the depth of water on a road is typically based upon an accumulation of water which will recede given a net outflow of water from an area. There exists an extensive body of

work into the modeling of flood events. Computational tools are available which can predict the scope and depth of flood events based on predicted rainfall amounts and topology, among other things [19, 34]. Typically, direct flood level measurement devices are located on rivers to measure fluvial flooding; this neglects the ability to measure surface (pluvial) flooding on roadways due to heavy rain events in urban areas. Flood models can provide estimates which span entire metro areas, which provides more information than could possibly be directly observed in a cost-effective manner.

The motivating question of this thesis is as follows: given a set of predictions (and no initial direct observations) about the level of flooding for all edges in a road network, how can we incorporate this data in combination with real-time observations of disrupted links, discovered while traversing the network, to utilize limited logistics resources more effectively? The answer to this question may enable government agencies and commercial entities to more efficiently and effectively carry out logistics operations in the aftermath of disasters. Route planning and graph search have been key areas of focus for many research communities, with results in fields as varied as transportation, robotics, and artificial intelligence. Algorithms which find shortest paths with full knowledge of graph status are abundant and computationally efficient, such as Dijkstra's, Floyd-Warshall and A$^*$[1]. Developing solutions based upon partial information requires building upon this foundation. However, these plans assume perfect information about the network at hand; an unrealistic assumption at best and deadly at worst during disasters.

Moving away from deterministic frameworks requires that solutions incorporate a state space of scenarios, increasing the complexity of computing solutions. One of the first investigations of shortest paths in networks with edge failures is in Andretta *et al.* [4], which investigates the *Stochastic Shortest Path with Recourse* (SSPR) on a directed network. SSPR takes into account that initial plans may include arcs which fail, and generates secondary (recourse) plans upon updating their knowledge of the true state of the network. A problem formulation which builds upon SSPR is the *Canadian Traveler's Problem* (CTP). In this formulation as initially presented

by Papadimitriou and Yannakakis [26], an initial road map is known, but certain roads (unknown to the agent before embarking on their journey) may be inaccessible to an agent. Due to the robust theoretical and experimental body of literature developed in the last 30 years, we adapt CTP to answer our motivating question.

When attempting to find an optimal route through a disrupted network using predictions of edge status, the best possible plan is that with the lowest expected cost across all potential configurations of failed edges [7]. Because the state space of CTP is exponential in the size of a graph, this is a computationally difficult task. There exist a number of heuristic policies which perform well when estimating the expected costs, but regardless of the expectation decision makers must deal with the actual trafficability of roads. A strategy widely utilized is managing exploration and exploitation. A single agent utilizing an optimal policy could be considered close to a depth-first search; they explore only as much as the adjacent arcs they see, but travel directly along what we anticipate to be the path with the lowest cost. Utilizing multiple agents increases the rate of learning the true configuration of the network, at the cost of consuming more resources. When considering the multi-agent CTP, a number of papers have been written on its theoretic properties which support the thought that exploration is beneficial to the overall solution, but the results are represented in worst-case competitive ratios that would lead to unacceptable travel times in most real-world applications. The author knows of only one experimental result which exists in the multi-agent CTP literature from Shiri and Salman [29], but the evidence provided points to more realistic performance.

This thesis seeks to rigorously experimentally validate the theoretical work to date on the CTP with multiple agents, which we denote as $\ell$-CTP to align with the usage of the letter 'L' as a variable for number of agents in theoretical results [38, 6]. First, we expand upon prior experiments in the the literature and develop a formulation of $\ell$-CTP. We implement this formulation and adapt existing heuristic policies to enable agents to utilize the information learned by other agents and collaborate to plan dissimilar routes. We show that our adapted

heuristic policies $\ell$-CTP generate shorter paths than multi-agent CTP algorithms currently in the literature through statistical analysis of simulations. Second, we quantify the benefits of adding agents by carrying out instances of $\ell$-CTP across a variety of randomly-generated Delaunay graphs. As a side effect of testing multiple routing strategies, we provide evidence and intuition for heuristic policy $\ell$-CTP variants which lead to low costs. Finally, we implement an instance of $\ell$-CTP on a case study generated from flooding prediction of Hurricane Harvey's impact on the city of Houston, Texas in 2017 to demonstrate the potential real-world benefits of this framework. As part of this analysis, we find that the model which describes the results on randomly generated graphs holds for the graph of the city of Houston.

The remainder of this thesis is organized as follows. Section 2 provides a problem framework for $\ell$-CTP . Section 3 is a literature review of similar works in single agent CTP and multi-agent CTP. Section 4 describes in detail the algorithms and heuristics utilized in experiments. Section 5 provides an overview of the experiments using random graphs, the experiments which provide policy-specific penalties and a real-world case study on using $\ell$-CTP to model response logistics on disrupted Houston road networks during Hurricane Harvey. Section 6 provides results and Section 7 summarizes critical findings from these results, limitations of this study, and areas for future investigation.

## 2 Problem Formulation

The *Canadian Traveler's Problem* (CTP) comes from the following general idea: a driver has a map of a city and is attempting to make their way to some destination, but the weather conditions are such that the only way to know if a road is clear is to observe it in person [26]. Our work builds upon results from Eyerich *et al.* [15], as it provides one of the most compelling sets of experimental results for single agent CTP with a framework which can be relatively easily expanded to multiple agents.

An instance of $\ell$-CTP is a 7-tuple: $(V, E, p, c, v_o, v_d, L)$, defined as follows.

- $(V, E)$ are the set of vertices and edges, respectively, that define a graph. $E$ are undirected edges.
- $0 < p_e \leq 1, \forall e \in E$ is the probability that edge $e \in E$ is available for travel. As such, our value of $p$ defines the Bernoulli distribution of that specific edge, where a success (1) indicates the edge is available, and a failure (0) indicates the edge is not available.
- $c_e \in R_*^+, \forall e \in E$ are the costs for an agent to travel along edge $e \in E$.
- $v_o \in V$, $v_d \in V \setminus v_o$ are our origin and destination vertices, respectively.
- $L$ is the set of agents which are instantiated at the beginning of a problem. While $|L| = 1$ is valid in our problem definition, we focus on $|L| \geq 2$.

For each instance of $\ell$-CTP, a set of failed arcs initially unknown to agents are generated. We define a *weather*, $W \subseteq E$, as a set of traversable edges in a CTP instance. $W$ is generated by carrying out a Bernoulli trial on edge $e$, denoted as $B(e)$. Our weather can be defined as $W = \{e \in E \,|\, B(e) = 1\}$. If the weather W contains a path from $v_o$ to $v_d$, we define the weather as *good*, otherwise if no path exists the weather is defined as *bad*. We make two assumptions with respect to $W$:

- $W$ remains constant throughout the CTP instance - edges do not change their availability before or after they are observed by an agent. This assumption holds in real-world applications when the rate of change in the road network is slower than the travel times of our agents.
- Every $W$ admitted to a CTP instance is good - in our experiments, we do not generate useful data by testing over instances with no feasible path from our origin to our destination.

Given a weather, the goal is for one or more of the agents at $v_o$ to travel to $v_d$ incurring the lowest cost possible. The cost incurred by an agent is denoted $z_\ell$. We judge the performance of a set of

agents as the minimum cost path identified across all agents:

$$\min_{\ell \in L} z_\ell. \tag{1}$$

For the comparison of our results with those found by Shiri and Salman, we define competitive ratio. We divide the lowest cost path found by an agent, as defined in eq. (1), to the offline shortest path cost $z^*$ with full knowledge of $W$:

$$\min_{\ell \in L} \frac{z_\ell}{z^*}. \tag{2}$$

We focus on optimizing the shortest agent path, with the goal of finding an available path with as little cost incurred as possible. The thought is that we are scouting a guaranteed path using smaller, lower cost vehicles (e.g., pickup trucks) before sending a larger vehicle filled with supplies (e.g., tractor trailer) along the guaranteed path. An instance of $\ell$-CTP starts with all agents initially located at $v_o$. As agents explore vertices chosen for their expected low cost, edges adjacent to the vertices are observed and compared to the initial graph provided, $(V, E)$. At each vertex, the agent is able to reveal if adjacent edges are present in $W$. Edges which are found to be available are placed in a set $E_A$, whereas edges which are found to be not available (failed) are placed in a set $E_F$. With full communication, these two sets are fully known by all agents and the communication between all agents is assumed to be instantaneous. This information is included in the choices made by the agents through updates to the expected costs of vertices. Once any agent reaches $v_d$, the remaining agents utilize the edges in $E_A$ to plan a route to $v_d$.

## 3 Literature Review

The literature on CTP has received attention in both theoretical and experimental realms since being introduced by Papadimitrou and Yannakakis in 1991 [26]. This section is split into two parts to focus on the contributions made to the single-agent CTP and the multi-agent CTP. Single-agent CTP has a robust set of literature exploring both theoretical and experimental

aspects of CTP. Multi-agent CTP has a number of theoretical results since being introduced by Zhang *et al.* in 2013 [38], however the experimental literature is lacking compared to single-agent CTP.

### 3.1 Single Agent CTP

Papadimitrou and Yannakakis are the authors who coined the terminology *Canadian Traveler's Problem*. There are a number of related works which cover similar topics in the same timeframe; Blei and Kaelbling introduce a similar problem called the *Bridge Problem*, with an additional variant called the *Dynamic Bridge Problem* [9]. Bertsekas and Tsitsiklis carried out an extensive, general analysis of *Stochastic Shortest Path Problems* [7]. As noted in the introduction, Andreatta and Romeo developed the SSPR which focuses on directed networks. A key commonality between these initial results is recognition that these problems can be converted into Markov Decision Processes (MDP), specifically Partially Observable Markov Decision Processes (POMDP). Similarly, both Papadimitrou and Yannakakis and Blei and Kaelbling recognized the exponential growth of the number of states [26, 9]. The exponential growth of the state space in MDP is driven by the potential edge status: unknown, known failed, known available, leading to $3^{|E|}$. Exact solutions can be computed in polynomial time for specific cases. On directed acyclic graphs, Nikolova and Karger proved that the optimal policy (minimizing the expected cost) can be solved in $O(|E|)$ [24].

In the absence of deterministic algorithms to generate optimal paths in polynomial time, a number of results have utilized the idea of competitive ratios as introduced by Sleator and Tarjan [31] to generate strategies guaranteed to meet bounds as determined by competitive ratio analysis. The `Backtrack` algorithm introduced by Westphal [36] provides the best guaranteed competitive ratio for a single agent as $2k+1$, where $k$ is the number of failed edges. The key idea behind `Backtrack` is that an agent travels until a failed edge is encountered, whereupon the agent then travels back to $v_o$ and attempts a new path. Bnaya *et al.* combine this result with disjoint shortest paths to guarantee that the optimal travel policy is to utilize `Backtrack` with shortest

paths in ascending cost [10]. While `Backtrack` strategy has the best competitive ratio, having a vehicle backtrack to a start point and restart a shortest path may be an unrealistic strategy in practice. A greedy strategy proposed by Xu *et al.* [37] has a competitive ratio of $2^{k+1} - 1$, however in grid networks of $m$ rows and $n$ columns of vertices, if the number of failures ($k$) is roughly equivalent to both the number of rows and columns (*i.e.*, $k \approx m \approx n$), the competitive ratio nears 3 [37]. As such the greedy policy can outperform the `Backtrack` strategy in nearly every such grid instance; in the case of a single failure, $k = 1$, the `Backtrack` strategy will generate a competitive ratio of $2k + 1 = 2 * 1 + 1 = 3$. The experiments carried out in this thesis draw upon these theoretical results to craft strategies which enable agents to travel paths with low average-case, versus worst-case, competitive ratios. Additionally, random strategies have been proposed for single agent CTP by Bender and Westphal, and Demaine *et al.* [5, 14].

One notable contribution from Blei and Kaelbling is their work into utilizing reinforcement learning as a substitute for solving an MDP to determine the expected costs of paths [9]. Eyerich *et al.* [15] provide compelling evidence of the capabilities of utilizing probabilistic heuristics in CTP. The agents utilize a heuristic, which the authors call a policy, that provides an expectation of the cost for an agent to travel from any given vertex to $v_d$ given the current set of edges which have been discovered [15]. They tested 5 policies, proved the theoretical potential of the policies, and carried out 3000 experiments on randomly generated graphs of 3 sizes to determine the best performing policies. We will further investigate some of the policies, as they are utilized in the experimental results of this thesis. Aksakalli et al. introduce the CAO$^*$ algorithm, which utilizes AO$^*$ search trees with caching to prune the potential state space and find the expected best path [3].

### 3.2 Multi-Agent CTP

The first paper which extensively models the multi-agent CTP is Zhang *et al.* [38]. In this paper, the overall theme is to investigate the same goals as the single agent CTP - to incur as little cost as possible by an agent in the attempt to reach a destination node, $v_d$, where all agents begin at

the same origin, $v_o$ [38]. Zhang *et al.* focuses on the individual cost incurred by any agent in particular, not the total cost incurred by all agents. The key benefit of adding multiple agents is their ability to simultaneously explore the network, determine available edges and share that information with the rest of the agents. In order to share information, there is an expected capability for agents that explore edge availability to communicate this information with the rest of the group. In a real-world application, this is analogous to technologies such as two-way radios and cellular phones [38]. Zhang *et al.* investigates two types of communication - complete and limited [38]. Complete communication assumes all agents have the ability to both send network information they discover and receive information, whereas limited communication has R-type agents which can only receive information and R&S type agents which are able to send their information to all agents and receive information as well [38]. For a network with $k$ failed edges and $L$ agents with full communication utilizing the `Backtrack` strategy, the best competitive ratio of any agent is proven to be $2\lfloor\frac{k}{L}\rfloor+1$ [38]. In this thesis, we will focus on the experimental benefits of full communication agents.

Shiri and Salman continue this analysis of the impact of multiple agents with limited communication [30]. The two communication levels from Zhang *et al.* are utilized, however an additional level of communication is presented, called IL3, such that some agents of type R&S can plan the actions of type R agents [30]. Shiri and Salman prove that with agents of type IL3, the optimal bound for the competitive ratio of $2\lfloor\frac{k}{L}\rfloor+1$ found by Zhang *et al.* for full communication agents holds [30]. Bergé *et al.* follow the results of Shiri & Salman [29] investigating a number of limited communication scenarios. A communication type called $\mathrm{P}_{\mathrm{initial}}$ is introduced, where agents are unable to communicate after leaving $v_o$, however can collaborate and initially plan disjoint paths [6]. The competitive ratio of this policy depends upon the number of agents in comparison to the number of failed edges; given $k+1 \leq L$, a competitive ratio of $k+1$ is proven for the first agent to reach $v_d$, and given $k+1 > L$ the competitive ratio is proven to be $2(k+1)-L$ [6]. This indicates that even with a sufficiently large number of agents with no ability to communicate after the initial plan development, you are able to quickly find $v_d$.

To date, the author knows of only one usage of multi-agent CTP as defined similarly to the literature described above in the experimental realm. This study was carried out by Shiri and Salman [29] in the context of search and rescue operations. Their experiments covered both abstract networks and real road networks. As demonstrated by the theoretical results, agents with full communication have the best potential competitive ratios and were utilized. A second aspect of utilizing multiple agents is forcing agents to take separate routes. This is accomplished by utilizing the iterative penalty method introduced by Akgün *et al.* [2]. The iterative penalty method for multiple agents generates paths by adding a penalty to a node or edge after being included in a prior agent's path. Shiri and Salman incorporate the iterative penalty method into their routing strategy by doubling an edge's visible weight after an agent utilizes that edge in their route. Agents plan their routes by assuming all edges are available and generating the shortest path after the iterative penalty method is applied for prior agents. If an agent finds a failed edge, all agents with that edge on their path generate a new plan and continue their traversal of the network. With this framework on a randomly generated network with 500 nodes, 1500 total edges, 600 edges randomly blocked, and only 7 agents, in 100 trials the average case ratio between actual path length and optimal path length was found to be slightly less than 1.5 [29]; the best competitive ratio guaranteed by Zhang et al. utilzing `Backtrack` would be $2 * \lfloor \frac{600}{7} \rfloor + 1 = 171$. This study by Shiri & Salman [29] provides excellent evidence that well-designed strategies even with basic path planning can efficiently navigate disrupted networks.

## 4 Methodology

Given the results in Shiri and Salman [29], we seek to advance the experimental multi-agent CTP literature by incorporating probabilistic policies to enable a small number of agents to provide similar levels of performance (or greater) as more sophisticated path generating techniques. We utilize much of the framework from Eyerich *et al.* [15] when developing the expected costs of a node, and combine this with the iterative penalty method as proposed by Shiri & Salman [29]. In

order to merge these two approaches, this section outlines a novel methodology which utilizes the A$^*$ graph search algorithm to incorporate the iterative penalty method with path cost estimation at vertices to solve an instance of $\ell$-CTP.

### 4.1 $\ell$-CTP instance

The state space of an instance $I$ of single agent CTP is of size $V3^E$. This increases to $V^L3^E$ in $\ell$-CTP with full communication, as we must keep track of agent locations. Due to full communication the knowledge of edge status is shared by all agents, and thus edge status does not change its contribution to the state space size. We define the set of all states in the potential combination of edge statuses and agent locations as $Q$.

Given an $\ell$-CTP formulation, we define agents as making their own movement choices. We choose to not incorporate a central planner which prescribes the exact path to be taken by agents. As such, there must be a mechanism to encourage diversity of paths; this is discussed in detail in §4.4.

The pseudocode which describes the actions carried out to solve an instance of $\ell$-CTP is described in Run_$\ell$-CTP. This pseudocode was generated directly from the implementation of $\ell$-CTP in the Python programming language. We utilize a base Agent class as a template for generating a set of initially identical agents as defined in the following section. These procedures are a simplification of the state transitions in a Markov Decision Process.

#### 4.1.1 Policy comparisons

In order to analyze the performance of our approach across multiple instances of $\ell$-CTP, we define nomenclature to aggregate sets of simulations. While weathers are randomly generated from graphs, in our simulations we are able to re-use those weathers and compare performance of the different policies directly. Similarly, policies will perform identically across a number of agents given specific ordering algorithms as defined in §4.4.2. Whenever performance of a policy

**Pseudocode** Run_$\ell$-CTP($V$, $E$, $p$, $c$, $v_o$, $v_d$, $L$)

```
Generate W ;
Create empty set of all failed edges E_F ;
Create empty set of all known available edges E_A ;
Create empty list of agents Agent_List ;
for i = 1 to L do
    Append instance of Agent class to Agent_List ;
Generate initial paths for all agents in Agent_List;
while | Agent_List | != 0 do
    Select agent in Agent_List with lowest cost, ties broken by smallest agent.number ;
    Update E_A, E_F with edges visible to current agent ;
    if next node in agent path not available then
        agent.generate_path(E_F, E_A);
    Set agent.current_node = next node in agent.path ;
    Add cost of arc to agent.cost ;
    Pop prior node from agent.path ;
    if agent.current_node = v_d then
        remove agent from Agent_List;
```

is discussed, we must include the ordering utilized. As such, when discussing performance of a policy we denote the the vertex cost estimation (XXX) and the agent ordering (YY) in the format XXX-YY. For example, for the optimist policy (OMT) which generates agent ordering in order of lowest expected future (EF) cost, we denote it OMT-EF. We compare performance across sets of graphs, weathers, and numbers of agents.

### 4.2 Agents

We define agents as rational entities which utilize a policy, $\pi$, to decide upon their course of action. The agent has a set of variables which are unique to that specific agent:

- Agent number assigned based upon order in which agent instantiation occurred
- Vertex where the agent is currently located
- Remaining path to arrive at $v_d$
- List of vertices visited, in order visited

- Cost incurred to reach current vertex

While we keep track of all the above variables, three independent variables are the most important: current location of the agent, the list of vertices visited and the remaining path as planned. All other variables can be derived from the above.

For a Markov Decision Process being solved online, at every state an agent seeks to maximize expected reward (or minimize expected cost, as in our case). There are two actions which change the current state in CTP - sensing the edges adjacent to the current vertex, and moving from one vertex to another. The sensing action in $\ell$-CTP is free, however both Eyerich *et al.* [15] and Bnaya *et al.* [10, 15] investigate remotely sensing edges not adjacent to the current agent's vertex for a cost. In $\ell$-CTP , we are in essence paying the premium of having additional agents to both sense more edges and exploit expected low cost paths which are discovered. An agent moving from one vertex to an adjacent vertex incurs a cost equal to that of the edge which connects the two vertices. Moving forward, we no longer discuss the sensing action in the context of moving between potential states in $Q$.

We present the greedy methodology which describes the method utilized by Eyerich *et al.* [15], however in solving an instance of $\ell$-CTP we utilize A$^*$ to develop routes, as described in §4.4.1. In order for an agent to minimize its expected cost to reach $v_d$, it must make a series of choices at individual states $q \in Q$ to move to an adjacent state with minimal expected cost increase, $q'$, which brings it closer to a state where the agent is located at $v_d$. Each decision takes into account the expected cost to reach $v_d$ from the state in question. Since an agent does not have visibility to all edges, it must utilize an estimation of the cost to reach $v_d$ from any vertex $v \in V$ given the information it knows in its current state, $q$. This cost expectation for all $q \in Q$ is denoted $C_\pi(q)$, where $\pi$ is defined as a policy, of which we investigate in further depth in §4.3. For the decision in question, we only utilize states reachable from our current state. We define a set of reachable states from $q$, $Q_{FS}$, as states which can be accessed form $q$ by known available edges. The choice of a state is by an agent is greedy; the agent selects its movement to a state with the smallest sum

of expected cost of that state and the known path cost to reach it. This is shown in eq. (3); in this formulation we utilize $c_{q,q_{FS}}$ to denote the shortest path cost between the current state and reachable state in question using edges known to be available:

$$q' = \arg\min_{q_{FS} \in Q_{FS}} c_{q,q_{FS}} + C_\pi(q_{FS}). \tag{3}$$

### 4.3 Policies

The policies mentioned in this section are the method by which agents estimate the cost which it takes to travel from any node $v \in V$ to $v_d$. A cost generated by a policy, $\pi$, is denoted $C_\pi$. This estimate can be accomplished in two main ways - determinism or probabilism. We define deterministic policies as the process of assigning expected cost directly from the known attributes of the graph, in a repeatable manner. We define probabilistic policies as the process of assigning expected cost based off of a sampling of weathers generated. While both deterministic and probabilistic policies are repeatable, the outcome for a probabilistic policy is based upon the sampled weather; if a probabilistic policy is given two identical sets of weathers, it will generate the same expectation for both sets.

For probabilistic policies, the term *rollout* is used to describe the weathers generated for the purpose of creating cost estimates of nodes. As with weathers admitted to an $\ell$-CTP instance, rollouts are also guaranteed to have a path from $v_o$ to $v_d$. The key difference of a rollout is that it utilizes the information learned by agents. For a single rollout, $r$, we generate a weather ($W$), ensure available edges ($E_A$) are included, and exclude any known failed edges ($E_F$). The set of available edges can be represented as follows: $r = (W \setminus E_F) \cup E_A$. If no path exists from $v_o$ to $v_d$ given this updated information, we throw out the rollout and generate a new one using the same process. The set operations carried out on $r$ guarantee that arcs we know have failed are excluded, and arcs that have been guaranteed available are included; this reflects the conditional probability of a weather being updated by our prior learned information. A set of rollouts is

defined as $R$.

### 4.3.1 Optimist (OMT)

The optimist policy, denoted as OMT, is a deterministic policy such that the agent estimates the expected cost from every node to $v_d$ by assuming every non-visible edge is available. OMT provides the lowest possible bound for an expected cost, and is also an admissible heuristic for A$^*$; that is, it never overestimates the cost to reach $v_d$ from any node. To generate a cost estimate for every node, we simply carry out Dijkstra's on $(V, E \setminus E_F)$ where our start node is $v_d$, and $E_F$ is the set of arcs known to be failed by an agent. $C_{OMT}$ is determined from the vertex labels generated by Dijkstra's algorithm.

OMT is investigated for a number of reasons. First, it is simple and computationally efficient. Dijkstra's algorithm is well studied with efficient implementations that are polynomial in their computational complexity [1]. Secondly, in robot literature the 'freespace assumption' - any action possible is assumed to be valid until observed otherwise - is commonly utilized [22].

### 4.3.2 Hindsight Optimist (HOP)

Hindsight optimist policy, denoted as HOP, is a probabilistic policy that averages the best possible costs from any given node to the destination node across a set of rollouts. As the term hindsight implies, the policy develops these costs with full knowledge of the edges removed in each rollout. For a set of rollouts, $R$, first apply Dijkstra's algorithm on $(V, r)\ \forall r \in R$ utilizing $v_d$ as the start node. We calculate $C_{HOP}$ by averaging the vertex labels for all vertices across all rollouts.

We include HOP in our investigation as it remains polynomial in its computational complexity; it is equivalent to Dijkstra's with an additional factor $R$ to denote the number of rollouts. This policy is also called 'averaging over clairvoyance' [28]. Based upon the results of Eyerich *et al.* [15], for a single agent the HOP policy performs 7.8% better on average than OMT with minimal

additional computational overhead.

### 4.4 Path Generation

A key benefit of multiple agents is the potential to explore multiple paths simultaneously and communicate the information gained between all agents [38, 6, 30]. The theoretical literature compares multiple different options for routing agents, but as noted these strategies are typically analyzed in competitive ratios that would be impractical in the real world. This sections describes how the paths are generated by both single agents, and how multiple agents generate disjoint paths to increase their exploration of edges.

#### 4.4.1 Generating Paths Using Policies and A*

We take a differing approach from Eyerich *et al.* [15] and align closer to Shiri & Salman [29] with our routing strategies for multiple agents. Instead of each agent selecting only the next state (vertex) to visit based upon expected low cost, agents generate a path from their current location to $v_d$. In our path generation framework, agents first generate cost estimates for all nodes in the network utilizing their policy. Second, agents generate a path utilizing A$^*$. The A$^*$ algorithm labels vertices starting with the origin, and ending only when the destination vertex is labeled. It begins at $v_o$ and checks the nodes which are reachable from $v_o$. The reachable vertex $v$ from $v_o$ which minimizes the function $f(v) = g(v) + h(v)$ is labeled, where $h(v)$ is the heuristic function (some measure of closeness to the destination - for example, straight-line distance) and $g(v)$ is the total path cost incurred to reach the vertex [18]. The labeling process continues until the destination vertex is labeled, and a path is found from $v_o$ to $v_d$ [18]. In determining shortest paths, the heuristic typically serves to reduce the number of nodes expanded by selecting vertices which are both low cost and minimize the heuristic function. Heuristic functions themselves have properties, and we consider the property of admissibility. Admissibility is defined as an heuristic which never overestimates the cost to travel from the current vertex to $v_d$ [18]. Admissibility leads to a guarantee of finding the shortest path [18]. However, a key premise of $\ell$-CTP is that the

shortest path in normal conditions is not necessarily the shortest path in the current weather. In fact, we seek to find paths which are higher cost than the shorter path, if they provide us with a low expected cost given the information known regarding edge availability and edge cost. The only policy which is guaranteed to be admissible is OMT, which aligns with its generation of expected costs based upon shortest path distances.

When utilizing an admissible heuristic, A$^*$ exactly replicates the paths generated by eq. (3) using that same admissible heuristic. We can think of A$^*$ as having our single agent exploring the entire graph using our current belief state, and then generating the resulting best path as a plan for that agent to follow. The A$^*$ algorithm using $C_\pi$ as heuristic functions will expand vertices which minimize the sum of the actual cost incurred to reach a vertex and the expected cost to reach the destination from that vertex. The main case for using A$^*$ is to incorporate the use of iterative penalties to generate disjoint paths between agents, as described in §4.4.2. Another benefit of using A$^*$ to generate paths is that agents can look further ahead, take into account high expected cost vertices, and exploit lower cost vertices. We utilize an example directly from Eyerich *et al.* [15] to demonstrate intuition as why we see better performance from A$^*$. We examine the case of utilizing HOP where $C_{HOP}$ have been calculated for all vertices, as demonstrated in fig. 1. Using the greedy policy, Eyerich *et al.* [15] showed that an agent using HOP will travel sub-optimally along path 0-1-0-5-7. With A$^*$, we generate the optimal path 0-5-7. Initially, the A$^*$ expands 1, as $C_{HOP}(1) + c_{0,1} \simeq 85$, compared to the other two adjacent vertices 5 and 7 such that $C_{HOP}(5) + c_{0,5} \simeq 90$ and $C_{HOP}(7) + c_{0,7} = 100$. We expand again, this time calculating the values for vertices 2, 3 and 4 - $C_{HOP}(2) + c_{0,1} + c_{1,2} \simeq 135$, and both vertices 3 and 4 take the same value. Given these labels, A$^*$ expands vertex 5. A$^*$ finally expands to vertex 7, with an expected path cost of 90. We can see that using A$^*$ with the HOP policy tests potential paths similar to the HOP policy using the greedy pathfinding methodology, but it does so before actually moving to prevent obvious sub-optimal choices. This results in a final path which minimizes the expected cost.

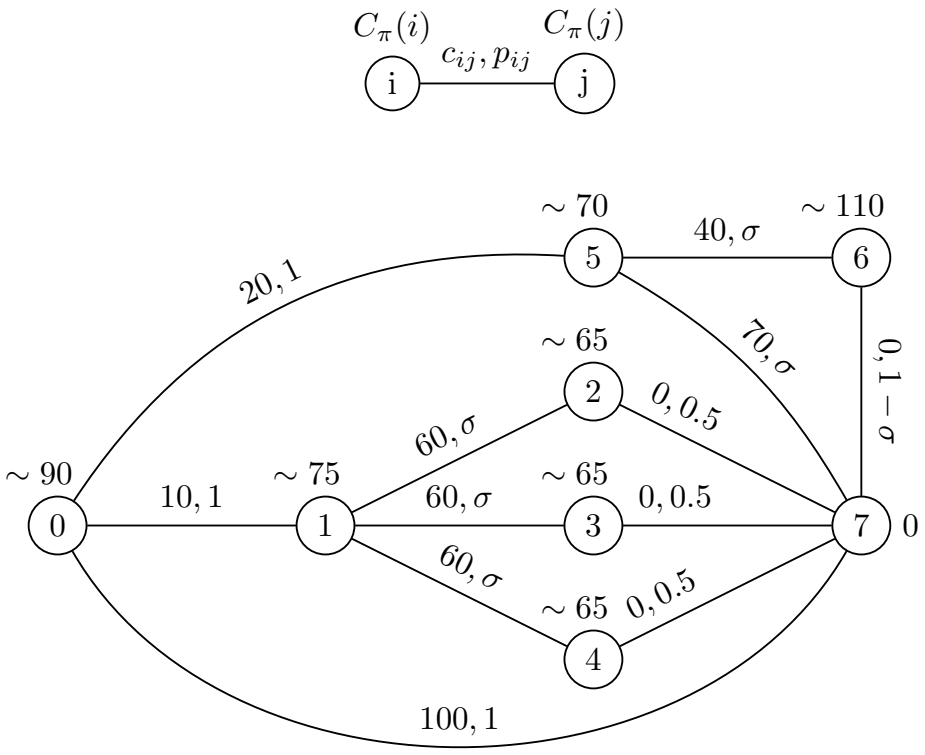



Figure 1: Example Graph, HOP policy

A notable downside is the computational impact of estimating expected costs for all nodes; for OMT this is trivial, but for the probabilistic policies this may require carrying out rollouts for all nodes when planning. For HOP this requires a small additional overhead, as Dijkstra's algorithm automatically calculates all node labels in the course of its computations. A second downside is that we plan an entire path using only the information at hand. Since agents develop a path and travel along this path until they reach $v_d$ or encounter a failed edge which they expect to be available, this could potentially lead to choices which cause higher costs by excluding some learned information in the course of traversing the graph. We address these concerns in experiments carried out in §5.4. A final note is that when any agent reaches $v_d$, we can guarantee that a path exists using only known edges for all agents from their current location to $v_d$. Once an agent reaches the final node, all other agents immediately generate a shortest path from their current node to $v_d$ using only available edges. In the worst case, this will be similar to the `backtrack` algorithm proposed by Westphal [36].

### 4.4.2 Using Iterative Penalty Method to Generate Disjoint Paths

As shown in both theoretical and experimental results, the act of information sharing and path planning is key to the performance of multi-agent CTP [29, 6]; as such, central to the $\ell$-CTP formulation is the development of low cost, disjoint paths. We utilize the iterative penalty method from Akgün *et al.* on nodes [2]. This is different than Shiri & Salman [29], who utilize iterative penalty method on the edges. We choose to weight the nodes based on the way that we generate expectations; the motivating principle is that we artificially increase the expectation of every node visited. The process for generating a path is as defined in generate_path, and this function is a method of an agent; thus it has access to its own information by the keyword 'self'. We define a weighting variable, $w_\pi$, which is a constant tuned for each specific policy based off of experiments described in §5.4. Note that in Run_$\ell$-CTP plans are generated as the agents are instantiated - this allows for the first agent to choose the expected lowest cost path. As agents are added, the agent list is increased and further agents must incorporate the plans generated of prior

agents.

Given our agents initially at $v_0$, using the iterative penalty method is straightforward. As agents progress through the network they will likely encounter failed edges along their path and must then generate new plans. Naïvely, we could simply replan only the current agent's path. Using this heuristic we would then have to incorporate the plans generated by the other agents, and the iterative penalty method would penalize vertices currently planned to be visited by other agents. However, if our current agent has traveled a low cost path and must generate a new plan, that new plan may be restricted by the existing plans made by other agents. Given this thought, we test the following methodologies for replanning all agents online:

- NP - Replan only the agent which encountered a failed edge
- OO - Replan all agents, by original order of instantiation (i.e., 1, 2, . . . , L)
- EF - Replan all agents, by lowest expected future cost (i.e., by order of $C_\pi(i)$, where $i$ is the vertex an agent is currently at)
- EC - Replan all agents, by lowest total expected path cost (current path cost incurred + $C_\pi(i)$, where $i$ is the vertex the an agent is currently at)

**Pseudocode** generate_path($V$, $E$, $E_A$, $E_F$, $p$, $c$, $v_o$, $v_d$, $\pi$, $w_\pi$, Agent_List)

```
Generate list of zeros visits of length |V| ;
for v ∈ V do
    Generate C_π(v) ;
Generate initial paths for all agents in Agent_List;
for a ∈ Agent_List \ self do
    for i ∈ a.path do
        visits[i]+=1 ;
for v ∈ V do
    C_π(v) = C_π(v) + visits[v] × w_π;
Generate path using A* with C_π as heuristic ;
return path
```

# 5 Computational Study

We utilize the methodology defined in the prior section to generate a set of experiments to test our motivating question: can we combine multiple agents who communicate with pathfinding strategies that take into account information on the probability of edge availability? We address this in two sets of experiments - on randomly generated graphs, and utilizing real-world road networks impacted by catastrophic levels of flooding.

## 5.1 Trials on Random Delaunay Graphs

The first set of experimentation is developed to test the performance of our approach on instances of $\ell$-CTP. We vary the numbers of agents which utilize policies defined in §4.3 and §4.4.2 on random graphs. We use random graphs to ensure the robustness of the policies in a network where the best paths are not obvious by inspection.

### 5.1.1 Design of Experiments

For our experiments, we align our study closely with Eyerich *et al.* [15]. We test our instances using the same graph sizes; 20, 50 and 100 nodes. Within each graph size we use 10 graphs. The author is grateful for the generosity of the authors of Eyerich *et al.* [15], as they provided the exact graphs on which they generated their data. These graphs are generated in the method laid out in §5.1.2. In each graph, $v_o = 1$, $v_d = |V|$ as utilized by Eyerich *et al.* [15]. For each graph, we will test 1-5 agents and the policies OMT and HOP on 1000 weathers. The penalties applied for generating disjoint paths are determined from §5.4. Additionally, we test the impact of different replanning policies - 4 versions for OMT, and 3 versions for HOP. This leads to $10 \times 1000 \times 5 \times 3 \times (3+4) = 1{,}050{,}000$ trials.

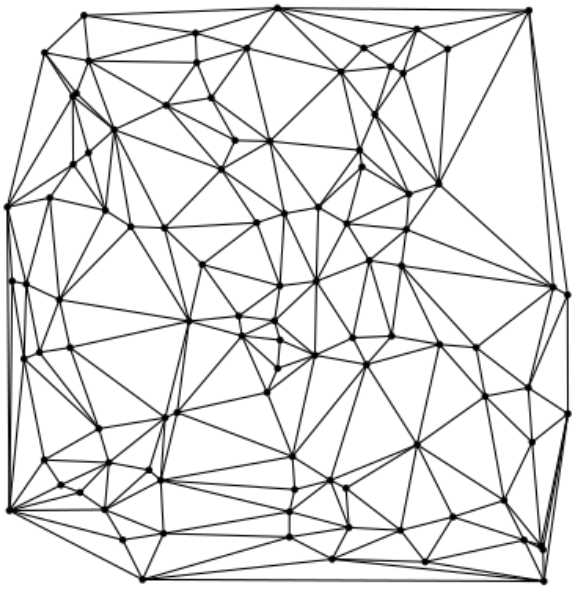


Figure 2: Delaunany triangulation using 100 randomly placed vertices [13]

### 5.1.2 Random Delaunay Graph Generation

To experimentally validate the performance of multiple agents on $\ell$-CTP instances, we carry out trials on Delaunay graphs which serve as close approximations to real world networks [10]. These graphs are generated by assigning a desired number of vertices a tuple of (x,y) coordinates on 2-axis grid, where x and y are individual identically distributed (iid) uniform random variables on the range [0,1]. The vertices are numbered in the order they are created in the set 1, 2, ..., $|V|$. These vertices are connected with edges using Delaunay triangulation methods - a typical resulting graph from this method using 100 vertices can be seen in figure 2. Each edge cost is an iid uniform random variable, $c_e \in [1, 50]$, and an iid uniform random variable describes probability of availability $p_e \in (0, 1]$.

## 5.2 Trials on Random Euclidean Graphs

The trials on random Euclidean graphs are a direct comparison between our algorithms using OMT policy with A$^*$ planning and the results from Shiri & Salman [29] to compare the merits of using IPM on nodes to utilizing IPM with edges. The authors of [29] made their experimental

code available, and using that we recreated their results and generated $\ell$-CTP instances given the same weathers and number of agents for each graph. The set of graphs and tests are as follows:

- Graphs of 100, 200, 300, 400, 500 nodes are generated by randomly connecting 3 vertices to another, and assigning a cost to those edges (thus containing 300, 600, 900, 1200, and 1500 edges)
- Edge failures (i.e., weathers) of 10%, 20%, 30%, and 40% are generated by randomly selecting a subset of edges equal to the percentage; for example, an edge failure of 20% on a graph of order 200 and size 600 will remove 120 edges at random
- 100 weathers are tested for each percentage listed above

Three different set of agents are tested on each weather, leading to $5 \times 4 \times 3 \times 100 = 6000$ trials.

### 5.3 Case Study for Hurricane Harvey

Hurricane Harvey was unique in its level of impact to the Houston metro area; the highest rainfall in the United States ever recorded for a tropical cyclone rainfall event was reported at 60.58 inches at a weather gauge near Nederland, Texas [8]. Large portions of the Houston metro area were inundated [8].

Given the immense flooding that occurred, Hurricane Harvey led to populations which required emergency aid to be delivered from federal agencies [27]. We seek to address the question of how to improve routing in disrupted networks. Having established the problem formulation of $\ell$-CTP, we can reduce the problem of finding the shortest path between two points in the Houston metro area during Hurricane Harvey as $\ell$-CTP given a set of edges which have associated costs ($c_e$) and probabilities of availability ($p_e$).

For this case study, we utilize the road network of Harris County, downloaded from Open StreetMap utilizing the python package OSMnx [11]. We include only major arterial roads, and exclude residential roads; the resulting network can be seen in fig. 3. This is done to model the

availability of larger roads which can be traversed by emergency vehicles and large commercial trucks. Costs are relatively straightforward - in our analysis, we will utilize edge lengths as costs. To develop probabilities of availability, we must know in some manner which roads may be flooded, and how often they may be flooded.

The availability of the road network is determined from flooding data generated by Pacific Northwest National Lab's Rapid Infrastructure Flood Tool (RIFT) [16]. RIFT utilizes sophisticated hydrodynamic models and Quantitative Predictive Forecasts (QPF) to estimate the velocity and depth of floodwaters at hourly intervals. Given velocity and inundation information, we assign a criterion to determine if an edge is traversable. Kramer *et al.* [23] experimentally found that that using a derivation of the Bernoulli equation in eq. (4), emergency vehicles are able to traverse a road up to $h_e = 0.6m$ safely, where $h$ is the water depth, $v$ is the velocity of the water, and $g$ is the gravitational acceleration constant:

$$h_E = h + \frac{v^2}{2g}. \tag{4}$$

Given this criteria and the RIFT dataset, we are able to assign a `True/False` value to each road segment, where `True` indicates a road is available and `False` indicates a road is not available. Since meteorological forecasts have inherent uncertainty, we account for this uncertainty as the timing of flooding in severe events such as Hurricane Harvey. We look at a five hour time window, starting at the expected start time with a snapshot of $h_e$. We then look on the hour, at the two hours prior and the two hours following our expected travel time. Using these five datapoints, we sum the times that an edge is found to be available and divide by 5; thus each edge will be assigned an availability in the set $\{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. Due to variability of the forecast, we do not want to assign a probability of availability of 0.0; thus we allow at least a probability of availability for all edges of 0.05. As such, $p_e \in \{0.05, 0.2, 0.4, 0.6, 0.8, 1.0\}\, \forall e \in E$. We test an origin/destination pair using 100 weathers, over 1, 2 and 3 agents, and we utilize HOP-EC and OMT-EF policies. This results in $100 \times 3 \times 2 = 600$ trials.

Figure 3: Harris County road network from Open Streetmap

### 5.4 Parameter Tuning

In order to generate routes of similar length with different paths, the iterative penalty method as described by Akgün *et al.* [2] weighted edges across multiple paths, varying from 1% penalty to 100%. The penalty is applied when a route uses that edge. As discovered in [2], dispersion increased dramatically from a 1% penalty to about 10%, leveling off in penalties from 10% up to 100%. Given these results, we carry out similar experiments to determine the average cost ratio of our solution to $\ell$-CTP as a function of magnitude of the penalty factor applied.

We set up an experiment to determine the value of the weighting parameter applied to vertices in generate_path, $w_\pi$, for each policy, $\pi$, which minimizes the lowest cost ratio defined in equation 2. For this experiment we utilize a randomly generated Delaunay graph with 100 vertices as our network as described in §5.1.2. In these experiments, we carry out 100 instances of $\ell$-CTP using 3 agents for each level of penalty and policy. The weathers in the 100 instances are copied from the initial weathers and repeated for each of the penalty and policy combinations. We additionally compare pathfinding using A$^*$ as described in §4.4.1 and fully online. The fully

online pathfinding is a direct implementation of equation 2. While iterating over each agent until they reach the destination node in Run_$\ell$-CTP, instead of using a plan they recalculate the expected cost for unvisited but reachable vertices using known available edges to select the next vertex to visit. In essence, they only generate a plan for the next vetex to utilize. This is a direct comparison to how Eyerich *et al.* [15] carry out experiments.

We derive four key observations from the results as seen in figure 4. First, multiple agents alone do not inherently add value. With no penalty, multiple agents will take the same route as one another and not provide benefit. Secondly, the generation of disjoint routes with 3 agents via iterative penalty method provides more efficient routing than a single agent. Using A$^*$ with OMT and iterative penalties decreases average run cost from being 80% greater than the optimal to only 28% greater than optimal. HOP similarly drops from 37% to 19%. Third, different policies benefit from different penalties. The performance of HOP peaks at a penalty of approximately 20%, whereas the performance of OMT peaks at approximately 50%. At a penalty of 50%, HOP would see worse performance, similar to OMT at 20% penalty. Finally, utilizing A$^*$ provides no worse performance than the online pathfinding. In our test instances we see better performance from A$^*$ in both test instances; A$^*$ with OMT is nearly 10 percentage points better than online at 50% penalty, and A$^*$ with HOP is 4 percentage points better than online at its best performance. From the results, we set $w_{OMT} = 0.5$, $w_{HOP} = 0.2$.

## 6 Results

In this section, the results of the computational studies are organized as follows. §6.1 contains the results from the trials on Delaunay graphs, §6.2 covers results from the trials on Euclidean graphs, and §6.3 describes the solutions to $\ell$-CTP instances generated from the real-world Houston road network.

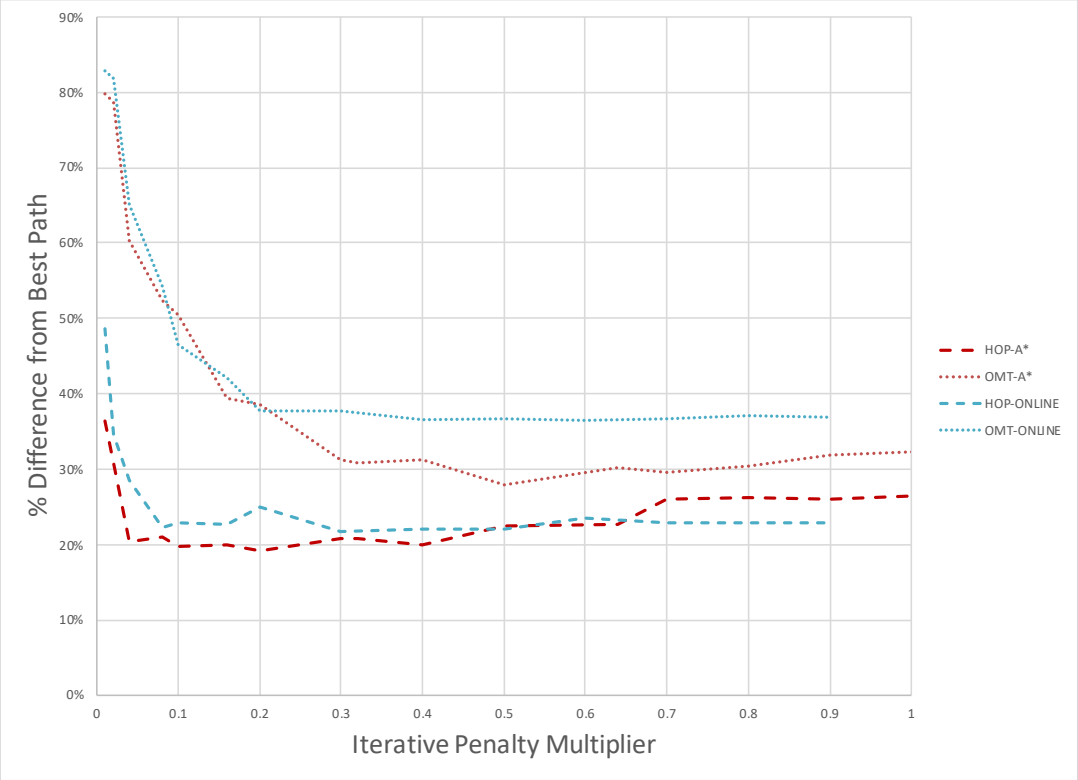



Figure 4: Tuning Penalties for $\ell$-CTP

### 6.1 Trials on Random Delaunay Graphs

The results of our simulation trials are developed from the experimental framework developed in §5.1.2. As described, we compare the average cost of $\ell$-CTP instances over 1000 weathers on graphs with 20, 50, and 100 vertices. The result tables are formatted as seen in table 1 for a specific policy and replanning variant (XXX-YY; OMT-EF in this case). The bolded cells in the columns under '$\ell$-CTP, number of agents' indicate the number of agents at which our solution method performs better (i.e., has a lower cost) than the UCTO policy from with a single agent [15] on that same graph. We make this comparison to UCTO for two purposes. UCTO stands for *upper confidence bounds applied to trees*, and is a sophisticated tree search algorithm which finds near-optimal solutions to Markov decision processes [21] - in CTP, this applies to cost estimates of vertices. First, we seek to determine for what number of agents a naïve policy, like OMT, reaches the level of performance of the most sophisticated and best performing single agent algorithm in the Eyerich *et al.* [15] results, UCTO. This would indicate that performance levels of advanced policies can be replicated by simple strategies with more agents. This is an important observation, as the simpler a strategy is, the more likely it can be successfully implemented in practice. Secondly, we want to test if a slightly more advanced policy, HOP, can further advance the performance improvement made possible with the use of multiple agents following policy OMT. If a set of agents can find lower cost paths than UCTO, we want to compare the magnitude of that difference, and also the relative benefit to adding an agent for HOP as compared to OMT.

As an example of this data, table 1 indicates that for graph 20-7, two agents using OMT-EF incur an average cost of 137.1, which is better than the single-agent UCTO average cost of 148.2 in Eyerich *et al.* [15]. Across all graphs of size 20, two agents using OMT-EF incur a lower cost on average than single-agent UCTO, with an average cost of 154.1 compared with 154.2. This is seen by comparing the bolded cell in the $\ell = 2$ column of the Average row with the value of the cell in the UCTO column in the same row. The remainder of these tables are found in the appendix in tables 12, 14, 16, 18, 20 and 22

Table 1: OMT-EF average path lengths on random Delaunay graphs

| | Eyerich *et al.* [15] | | $\ell$-CTP, number of agents | | | | |
|---|---|---|---|---|---|---|---|
| Graph | OMT | UCTO | 1 | 2 | 3 | 4 | 5 |
| 20-1 | 205.9 | 169.0 | 200.7 | **170.7** | 156.4 | 150.5 | 146.6 |
| 20-2 | 187.0 | 148.9 | 187.3 | 164.5 | 152.7 | **148.1** | 145.8 |
| 20-3 | 139.5 | 132.5 | 142.9 | **125.9** | 122.0 | 119.9 | 118.9 |
| 20-4 | 266.2 | 235.2 | 244.7 | **204.2** | 193.0 | 186.5 | 182.1 |
| 20-5 | 163.1 | 111.3 | 166.5 | 123.1 | 114.5 | **106.2** | 103.2 |
| 20-6 | 180.2 | 133.1 | 182.0 | 135.7 | **124.4** | 120.1 | 116.5 |
| 20-7 | 172.2 | 148.2 | 170.5 | **137.1** | 131.8 | 128.1 | 126.5 |
| 20-8 | 150.1 | 134.5 | 152.4 | 140.8 | **134.0** | 128.3 | 125.6 |
| 20-9 | 222.0 | 173.9 | 222.2 | 182.4 | **172.4** | 167.4 | 162.3 |
| 20-10 | 178.2 | 167.0 | 175.7 | **157.0** | 150.3 | 144.5 | 140.9 |
| Average | 186.5 | 154.2 | 184.5 | **154.1** | 145.2 | 140.0 | 136.9 |
| 50-1 | 255.5 | 186.1 | 254.5 | **170.5** | 157.6 | 144.4 | 142.5 |
| 50-2 | 467.1 | 365.5 | 463.6 | 365.7 | **346.1** | 334.1 | 324.0 |
| 50-3 | 281.5 | 255.6 | 289.2 | **235.8** | 217.8 | 210.3 | 207.5 |
| 50-4 | 289.8 | 230.5 | 287.6 | 233.8 | **211.2** | 202.9 | 197.4 |
| 50-5 | 285.5 | 225.4 | 285.7 | **214.0** | 198.7 | 189.1 | 186.5 |
| 50-6 | 251.3 | 236.3 | 243.0 | **224.9** | 214.0 | 208.5 | 204.4 |
| 50-7 | 242.2 | 206.3 | 235.9 | 205.3 | **187.5** | 178.3 | 174.4 |
| 50-8 | 355.1 | 277.6 | 363.6 | 285.0 | **259.4** | 248.9 | 243.0 |
| 50-9 | 327.4 | 222.5 | 331.0 | 242.0 | **215.2** | 203.7 | 198.8 |
| 50-10 | 281.6 | 240.8 | 292.5 | **220.2** | 202.9 | 196.5 | 190.8 |
| Average | 303.7 | 244.7 | 304.7 | **239.7** | 221.0 | 211.7 | 206.9 |
| 100-1 | 370.9 | 286.8 | 354.7 | **283.7** | 256.6 | 241.1 | 233.3 |
| 100-2 | 160.6 | 151.5 | 169.0 | 155.9 | **153.1** | 151.0 | 149.9 |
| 100-3 | 550.2 | 412.2 | 550.1 | **408.1** | 371.0 | 345.6 | 334.3 |
| 100-4 | 420.1 | 314.3 | 431.2 | 328.3 | **312.9** | 298.8 | 296.0 |
| 100-5 | 397.0 | 348.3 | 411.4 | **340.3** | 319.6 | 305.0 | 295.6 |
| 100-6 | 455.0 | 396.2 | 478.7 | 398.2 | **372.2** | 356.9 | 351.4 |
| 100-7 | 431.4 | 358.2 | 435.8 | **332.2** | 306.1 | 293.3 | 285.4 |
| 100-8 | 335.6 | 293.3 | 328.0 | **259.6** | 234.0 | 222.4 | 215.2 |
| 100-9 | 327.5 | 262.0 | 329.9 | 268.2 | **240.2** | 231.9 | 223.3 |
| 100-10 | 381.5 | 342.3 | 395.8 | **311.1** | 279.6 | 267.3 | 259.0 |
| Average | 383.0 | 316.5 | 388.4 | **308.6** | 284.5 | 271.3 | 264.3 |

We carried out a second level of analysis on the performance of multiple agents on these weathers, due to the results of OMT-NP on graph size 100-4 in table 12, found in the appendix. In this case, average path cost increased from 303.1 to 306.0 when adding a fifth agent. This result is particularly surprising, as we predicted that in the worst case adding agents would cause no performance gains. This data point provides evidence to the contrary - adding an agent has the possibility of increasing the cost of paths found. To investigate this peculiar finding, the data were analyzed for instances when adding agents in specific weathers caused worse performance, i.e. higher cost. This is accomplished by subtracting the best cost path using $\ell$ agents from the performance of the best cost path using $\ell - 1$ agents. The results generated from carrying out this analysis on the graphs and weathers summarized in table 1 are tabulated in table 2. The number of weathers where agents generated worse paths with the addition of an agent are listed in the column labeled #, and the average percent increase in costs of those paths are in the column labeled %. Continuing our analysis of graph 20-7, 10 out of the 1,000 weathers had worse plans created when adding a second agent, where those plans, on average, were 8.0% more expensive. Across graphs of size 20 for 2 agents, 81 out of the 10,000 total weathers had worse performance, and on average those instances led to cost increases of 15.2%. While this indicates that the choice of adding agents is not without risk of incurring higher cost, in 99.19% of weathers, adding a second agent with OMT-EF leads to a lower cost path. The remainder of these data tables are located in the appendix in tables 13, 15, 17, 19, 21 and 23.

### 6.1.1 Equivalence between weathers

While we did not have access to identical weathers to Eyerich *et al.* [15], we did have identical graphs, and we can show equivalence of weathers with the experiments carried out using our solution methodology to $\ell$-CTP by examining the results of 1 agent using the OMT policy. As noted, the OMT policy provides an admissible heuristic to A$^*$ and will generate the shortest path, which is the same as OMT using the greedy policy as in Eyerich *et al.* [15]. Given that we are utilizing identical graphs and the same origin/destination node pairs, we expect the mean of the

Table 2: OMT-EF analysis of performance decrease across number of agents

| | Number of agent | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | | 2 | | 3 | | 4 | | 5 | |
| Graph | # | % | # | % | # | % | # | % | # | % |
| 20-1 | — | — | 6 | 7.7 | 19 | 29.4 | 37 | 17.3 | 34 | 11.0 |
| 20-2 | — | — | 16 | 4.6 | 13 | 6.0 | 21 | 6.5 | 16 | 7.5 |
| 20-3 | — | — | 6 | 9.6 | 1 | 36.8 | 6 | 10.1 | 3 | 7.7 |
| 20-4 | — | — | 5 | 14.7 | 2 | 10.6 | 13 | 13.4 | 9 | 22.6 |
| 20-5 | — | — | 6 | 49.9 | 8 | 8.5 | 4 | 4.3 | 6 | 11.3 |
| 20-6 | — | — | 12 | 15.7 | 16 | 11.5 | 11 | 6.0 | 18 | 8.6 |
| 20-7 | — | — | 10 | 8.0 | 12 | 9.5 | 8 | 6.9 | 6 | 9.5 |
| 20-8 | — | — | 0 | — | 7 | 7.5 | 8 | 7.2 | 18 | 27.3 |
| 20-9 | — | — | 11 | 9.2 | 39 | 16.3 | 33 | 16.3 | 50 | 9.5 |
| 20-10 | — | — | 9 | 17.0 | 10 | 7.9 | 23 | 12.7 | 18 | 9.7 |
| Average | — | — | — | 15.2 | — | 14.4 | — | 10.1 | — | 12.5 |
| Total | — | — | 81 | — | 127 | — | 164 | — | 178 | — |
| 50-1 | — | — | 12 | 7.8 | 14 | 7.8 | 43 | 12.6 | 23 | 12.0 |
| 50-2 | — | — | 25 | 8.2 | 83 | 10.3 | 99 | 8.7 | 109 | 7.6 |
| 50-3 | — | — | 71 | 7.1 | 95 | 7.6 | 93 | 8.1 | 144 | 9.8 |
| 50-4 | — | — | 38 | 5.4 | 84 | 11.9 | 97 | 9.7 | 101 | 9.1 |
| 50-5 | — | — | 14 | 28.6 | 36 | 15.7 | 57 | 12.5 | 64 | 12.3 |
| 50-6 | — | — | 19 | 12.3 | 27 | 9.7 | 33 | 4.9 | 39 | 6.6 |
| 50-7 | — | — | 20 | 8.9 | 25 | 12.8 | 27 | 6.4 | 55 | 8.7 |
| 50-8 | — | — | 58 | 11.1 | 67 | 9.2 | 88 | 7.1 | 122 | 8.1 |
| 50-9 | — | — | 26 | 20.5 | 46 | 12.9 | 38 | 11.3 | 74 | 10.9 |
| 50-10 | — | — | 62 | 17.5 | 56 | 13.2 | 88 | 12.7 | 164 | 11.4 |
| Average | — | — | — | 12.7 | — | 11.1 | — | 9.4 | — | 9.7 |
| Total | — | — | 345 | — | 533 | — | 663 | — | 895 | — |
| 100-1 | — | — | 34 | 13.0 | 61 | 12.8 | 60 | 11.3 | 105 | 11.1 |
| 100-2 | — | — | 7 | 10.4 | 36 | 10.4 | 21 | 4.2 | 34 | 7.9 |
| 100-3 | — | — | 36 | 15.1 | 71 | 10.7 | 118 | 9.5 | 154 | 12.3 |
| 100-4 | — | — | 59 | 13.7 | 136 | 10.3 | 179 | 10.2 | 328 | 10.3 |
| 100-5 | — | — | 43 | 12.9 | 96 | 8.3 | 119 | 6.6 | 113 | 7.2 |
| 100-6 | — | — | 68 | 10.5 | 79 | 8.7 | 114 | 6.0 | 178 | 8.7 |
| 100-7 | — | — | 37 | 10.0 | 80 | 8.5 | 107 | 8.9 | 115 | 8.1 |
| 100-8 | — | — | 23 | 9.2 | 45 | 7.2 | 74 | 5.2 | 68 | 8.2 |
| 100-9 | — | — | 18 | 14.0 | 22 | 10.5 | 61 | 5.6 | 53 | 5.4 |
| 100-10 | — | — | 75 | 11.8 | 69 | 9.0 | 93 | 9.2 | 113 | 9.5 |
| Average | — | — | — | 12.1 | — | 9.6 | — | 7.7 | — | 8.9 |
| Total | — | — | 400 | — | 695 | — | 946 | — | 1261 | — |

costs to converge in probability. The mean costs with 1 agent are within 1.1%, 0.33% and 1.4% of the Eyerich *et al.* [15] results for graphs of size 20, 50, and 100 vertices, respectively. We carry out two-sample Z-tests comparing the sample means generated by both Eyerich *et al.* [15] and our results, shown in table 3. Each row lists the statistics which define the agent costs found by Eyerich *et al.* [15] and our results on a single graph across 1000 weathers, where $\mu$ is the sample mean, and $\sigma$ is the standard deviation. For results generated as solutions to $\ell$-CTP instances, we directly calculated $\mu$ and $\sigma$ from the raw data. The values of $\mu$ and 95% CI for Eyerich *et al.* [15] are directly taken from their results in their table 1. While the sample standard deviations are not provided in [15], the 95 percent confidence intervals are included. We calculate $\sigma$ for Eyerich *et al.* [15] from the confidence intervals in order to calculate the test statistic $Z_0$. Our null and alternative hypotheses are as follows:

$H_0 : \mu_{Eyerich,OMT} = \mu_{1-CTP,OMT}$, $H_1 : \mu_{Eyerich,OMT} \neq \mu_{1-CTP,OMT}$. We reject the null hypothesis in favor of the alternative when $Z_0 > Z_{\alpha/2}$ or $Z_0 < -Z_{\alpha/2}$, where $Z_{\alpha/2} = 1.96$. Across the 30 graphs, we fail to reject the null hypothesis in 27/30 graphs, concluding there is not enough evidence in the samples to indicate the two solution approaches are not equivalent. The remaining three graphs, 20-4, 100-1 and 100-6, fall in the critical region, indicating a statistically significant difference between the single-agent OMT results in our paper and in Eyerich *et al.* [15]. A potential factor which leads to rejecting $H_0$ is that the confidence intervals provided have only one significant figure; this is a source of error in our calculations of $Z_0$ as rounding could cause a variation in the confidence interval of up to 1 unit. Additionally, 2/3 graphs with a test statistic in the critical region are of size 100; as the number of edges in the graph grows, the state space of weathers grows exponentially. However, this result indicates that in 27/30 graphs there is no statistically significant difference in the OMT path costs with a confidence level of 95%, providing evidence that our weathers, on average, approximate those found by Eyerich *et al.* [15].

Table 3: Two-sample Z-test between Eyerich *et al.* [15] and single agent OMT results.

| | Eyerich *et al.* [15], OMT | | | $\ell$-CTP, OMT | | |
|---|---|---|---|---|---|---|
| Graph | $\mu$ | 95% CI | $\sigma$ | $\mu$ | $\sigma$ | $Z_0$ |
| 20-1 | 205.9 | 7 | 112.9 | 200.7 | 107.7 | 1.05 |
| 20-2 | 187.0 | 5 | 80.7 | 187.3 | 82.8 | -0.08 |
| 20-3 | 139.5 | 6 | 96.8 | 142.9 | 103.2 | -0.76 |
| 20-4 | 266.2 | 8 | 129.0 | 244.7 | 118.7 | 3.88* |
| 20-5 | 163.1 | 7 | 112.9 | 166.5 | 117.0 | -0.66 |
| 20-6 | 180.2 | 6 | 96.8 | 182.0 | 89.7 | -0.43 |
| 20-7 | 172.2 | 5 | 80.7 | 170.5 | 78.0 | 0.48 |
| 20-8 | 150.1 | 6 | 96.8 | 152.4 | 97.2 | -0.53 |
| 20-9 | 222.0 | 5 | 80.7 | 222.2 | 74.9 | -0.05 |
| 20-10 | 178.2 | 6 | 96.8 | 175.7 | 96.3 | 0.57 |
| 50-1 | 255.5 | 10 | 161.3 | 254.5 | 165.8 | 0.13 |
| 50-2 | 467.1 | 11 | 177.5 | 463.6 | 173.2 | 0.44 |
| 50-3 | 281.5 | 9 | 145.2 | 289.2 | 160.5 | -1.12 |
| 50-4 | 289.8 | 9 | 145.2 | 287.6 | 142.1 | 0.34 |
| 50-5 | 285.5 | 10 | 161.3 | 285.7 | 152.3 | -0.03 |
| 50-6 | 251.3 | 10 | 161.3 | 243.0 | 147.1 | 1.20 |
| 50-7 | 242.2 | 9 | 145.2 | 235.9 | 133.7 | 1.00 |
| 50-8 | 355.1 | 11 | 177.5 | 363.6 | 168.3 | -1.10 |
| 50-9 | 327.4 | 13 | 209.7 | 331.0 | 200.0 | -0.39 |
| 50-10 | 281.6 | 8 | 129.0 | 292.5 | 140.2 | -1.81 |
| 100-1 | 370.9 | 11 | 177.5 | 354.7 | 173.0 | 2.07* |
| 100-2 | 160.6 | 8 | 129.0 | 169.0 | 132.0 | -1.43 |
| 100-3 | 550.2 | 18 | 290.4 | 550.1 | 290.2 | 0.00 |
| 100-4 | 420.1 | 10 | 161.3 | 431.2 | 173.8 | -1.48 |
| 100-5 | 397.0 | 16 | 258.1 | 411.4 | 260.0 | -1.24 |
| 100-6 | 455.0 | 12 | 193.6 | 478.7 | 218.5 | -2.57* |
| 100-7 | 431.4 | 15 | 242.0 | 435.8 | 245.6 | -0.40 |
| 100-8 | 335.6 | 12 | 193.6 | 328.0 | 188.1 | 0.89 |
| 100-9 | 327.5 | 14 | 225.9 | 329.9 | 240.7 | -0.23 |
| 100-10 | 381.5 | 11 | 177.5 | 396.3 | 181.8 | -1.84 |

* indicates results with statistically significant differences between means

### 6.1.2 Impact of A* planning on single agent using HOP

As described in the prior section, OMT performance with 1 agent is not affected by using A* instead of a greedy approach as OMT utilizes a consistent heuristic for A*. However, the path cost of a single agent using HOP decreased by 3.9% from Eyerich *et al.* [15]. This aligns with our intuition developed in §4.4.1. In table 20, we can see on graphs 20-6, 20-7, 50-4, and 100-1 that a single agent utilizing HOP in our approach performs better than a single agent using the UCTO policy in Eyerich *et al.* [15]. We analyze the difference between the average shortest paths costs between the Eyerich *et al.* [15] implementation of HOP and our own, both for a single agent, utilizing a two-sample Z-test, seen in table 4. Our hypothesis follows:
$H_0 : \mu_{Eyerich,HOP} = \mu_{1-CTP,HOP}$, $H_1 : \mu_{Eyerich,HOP} > \mu_{1-CTP,HOP}$. We reject the null hypothesis in favor of the alternative when $Z_0 > Z_\alpha$, where $Z_\alpha = 1.64$. Across the 30 graphs, we reject the null hypothesis in 19 of those graphs. This indicates that A* usually, but not always, improves the performance of a single agent using HOP. Additionally, we note that A* performs worse in graphs 20-5, 20-8, 50-8, 100-2, and 100-4. While it would not be unexpected that in some instances worse paths could be generated, we look at the performance of the OMT policy on these weathers. The paths generated by agents utilizing OMT in these $\ell$-CTP instances have higher costs than the Eyerich *et al.* [15] results, indicating that the weathers sampled for those graphs, on average, are worse than those sampled by Eyerich *et al.* [15]. Given this information, we attribute some portion of the difference in performance to the weathers. However, in 8/10 graphs of size 100 and 8/10 graphs of size 50, agents utilizing HOP perform better using A*, versus 3/10 in the smallest graph. We intuit that in graphs of 20 vertices there exist fewer choices which leads to minimal benefit to generating a full path versus using a greedy heuristic. This is supported by the fact that the average number of vertices which separate the origin and destination in graphs of size 20, 50, and 100 are 2.3, 3.5, and 4.6, respectively.

Table 4: Two-sample Z-test between Eyerich *et al.* [15] and $\ell$-CTP single agent HOP results

| | Eyerich *et al.* [15], HOP | | | $\ell$-CTP, HOP | | |
|---|---|---|---|---|---|---|
| Graph | $\mu$ | 95% CI | $\sigma$ | $\mu$ | $\sigma$ | $Z_0$ |
| 20-1 | 171.6 | 6 | 96.8 | 169.4 | 89.6 | 0.52 |
| 20-2 | 155.8 | 3 | 48.4 | 154.7 | 58.9 | 0.45 |
| 20-3 | 138.7 | 6 | 96.8 | 138.3 | 96.5 | 0.09 |
| 20-4 | 286.8 | 8 | 129.0 | 241.4 | 114.4 | 8.32 |
| 20-5 | 113.3 | 5 | 80.6 | 124.3 | 106.2 | -2.60 |
| 20-6 | 142.0 | 4 | 64.5 | 132.5 | 57.5 | 3.47 |
| 20-7 | 150.2 | 4 | 64.5 | 147.9 | 68.4 | 0.77 |
| 20-8 | 133.6 | 5 | 80.6 | 137.5 | 81.9 | -1.07 |
| 20-9 | 177.1 | 4 | 64.5 | 174.4 | 66.7 | 0.91 |
| 20-10 | 188.1 | 6 | 96.8 | 171.7 | 91.4 | 3.89 |
| 50-1 | 250.6 | 9 | 145.2 | 230.4 | 138.6 | 3.17 |
| 50-2 | 375.4 | 7 | 112.9 | 370.9 | 118.3 | 0.86 |
| 50-3 | 294.5 | 7 | 112.9 | 274.5 | 135.7 | 3.57 |
| 50-4 | 263.9 | 7 | 112.9 | 228.5 | 105.5 | 7.24 |
| 50-5 | 239.5 | 8 | 129.0 | 226.7 | 106.5 | 2.41 |
| 50-6 | 253.2 | 9 | 145.2 | 238.5 | 141.3 | 2.27 |
| 50-7 | 221.9 | 7 | 112.9 | 207.2 | 106.5 | 2.99 |
| 50-8 | 302.2 | 9 | 145.2 | 305.3 | 144.3 | -0.48 |
| 50-9 | 281.8 | 11 | 177.4 | 243.9 | 159.5 | 5.01 |
| 50-10 | 271.2 | 7 | 112.9 | 261.0 | 118.2 | 1.96 |
| 100-1 | 319.3 | 9 | 145.2 | 281.3 | 120.4 | 6.36 |
| 100-2 | 154.5 | 7 | 112.9 | 158.2 | 109.5 | -0.74 |
| 100-3 | 488.1 | 15 | 242.0 | 446.5 | 247.2 | 3.80 |
| 100-4 | 329.8 | 7 | 112.9 | 333.8 | 131.3 | -0.73 |
| 100-5 | 452.4 | 18 | 290.4 | 385.0 | 233.5 | 5.71 |
| 100-6 | 487.9 | 11 | 177.4 | 421.5 | 179.3 | 8.32 |
| 100-7 | 403.9 | 14 | 225.8 | 380.0 | 221.1 | 2.39 |
| 100-8 | 322.0 | 12 | 193.6 | 295.0 | 171.3 | 3.30 |
| 100-9 | 366.1 | 15 | 242.0 | 273.4 | 192.0 | 9.48 |
| 100-10 | 388.4 | 11 | 177.4 | 354.9 | 152.7 | 4.52 |

Table 5: Paired Student's t-test on OMT replanning variants across all graphs

| | | *j* | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | OMT-EC | | OMT-EF | | OMT-NP | | OMT-OO | |
| *i* | # agents | *t* | *p* | *t* | *p* | *t* | *p* | *t* | *p* |
| OMT-EC | 2 | — | — | -12.49 | 0.0 | 3.23 | 0.001 | 3.01 | 0.003 |
| | 3 | — | — | -13.09 | 0.0 | 3.36 | 0.001 | -1.17 | 0.241 |
| | 4 | — | — | -13.20 | 0.0 | 4.39 | 0.000 | 1.59 | 0.111 |
| | 5 | — | — | -9.26 | 0.0 | 6.83 | 0.000 | 0.64 | 0.521 |
| OMT-EF | 2 | — | — | — | — | 10.01 | 0.000 | 13.18 | 0.000 |
| | 3 | — | — | — | — | 11.63 | 0.000 | 9.94 | 0.000 |
| | 4 | — | — | — | — | 12.99 | 0.000 | 13.23 | 0.000 |
| | 5 | — | — | — | — | 13.25 | 0.000 | 8.70 | 0.000 |
| OMT-NP | 2 | — | — | — | — | — | — | -1.41 | 0.159 |
| | 3 | — | — | — | — | — | — | -4.26 | 0.000 |
| | 4 | — | — | — | — | — | — | -3.23 | 0.001 |
| | 5 | — | — | — | — | — | — | -6.42 | 0.000 |

### 6.1.3 Impact of Replanning Order

As discussed in §4.4.2, we posited that the order in which agents generate paths after encountering failed edges may impact the cost of the paths which they generate. This is rooted in tension between exploration and exploitation of a graph where the failed edges are not known *a priori*. We first investigate the results from table 12 (found in the appendix) which cover the case of only the agent which encounters a failed edge replans (NP). Given the use of IPM, we supposed in §4.4.2 that NP may lead to the generation of artificially higher cost paths as vertices are already utilized by other agents when the lone agent creates a new plan. This prediction is correct, and as mentioned in §6.1 the addition of a fifth agent in graph 100-4 leads to worse outcomes across 1000 weathers on average, going from 303.1 to 306.0 as seen in table 12. Results in table 13 provide additional detail, as 36.2% of all weathers for graph 100-4 saw a performance decrease of, on average, 17.3%.

To determine the best-performing replanning policy for OMT, we carry out paired Student's t-tests across all graphs, comparing minimum agent costs on identical weathers based upon the

replanning policy utilized by the agents. The hypothesis that we use for these tests are as follows for two replanning policies $i$ and $j$. We compare the change from policy $i$ to $j$. We first define $\mu_D = \mu_{OMT_i} - \mu_{OMT_j}$. We use the null hypothesis $H_0 : \mu_D = 0$. We vary the alternative hypothesis based upon the value of the test statistic. For $t < 0$, $H_1 : \mu_D < 0$. For $t > 0$, $H_1 : \mu_D > 0$.We reject the null hypothesis in favor of the alternative when $p < \alpha = 0.05$. When the statistic $t > 0$, this indicates we see a decrease in path cost from $i$ to $j$; if $t < 0$, this indicates we an increase in the best path cost from $i$ to $j$. The results of these tests are reported in table 5.

The results in table 5 support our prior intuition developed from table 12. We see a decrease in agent cost between $\ell$-CTP trials for OMT on similar weathers from only replanning with one agent (OMT-NP) to replanning with all agents (OMT-EC, OMT-EF, OMT-OO) - additionally, as the number of agents increases in the trials, we see the costs incurred decrease further between OMT-NP and all other policies. This indicates that relative to the other policies, higher cost paths are generated as agents are added to $\ell$-CTP instances utilizing NP. We also seek to answer which replanning policy performs the best. When carrying out the paired Student's t-tests, we see increases in path cost from all policies to the best policy. In table 12, this is OMT-EF. The statistic for all t-tests is negative, and all have $p < 0.05$. This indicates that switching from all other policies to this policy leads to a cost decrease. Given this data, an ordering of the policies from lowest agent costs to highest agent costs can be made as following: EF $<$ OO $\leq$ EC $<$ NP.

Since replanning using NP was shown to incur significantly greater costs across all replanning policies, when carrying out simulations using HOP we include only HOP-EC, HOP-EF, and HOP-OO in table 6. Similarly to OMT, a single policy dominates. Agents utilizing HOP-EF incur significantly lower costs than HOP-OO, with $p < 0.05$. HOP-OO is the worst, as agent costs increase from both HOP-EF and HOP-EC. HOP-EC is in between, as it is significantly less than HOP-OO, but greater than HOP-EF, but with significance levels much greater than 0.05. Given this data, an ordering of the policies from lowest agent costs to highest agent costs can be made as following: EF $\leq$ EC $<$ OO.

Table 6: Paired Student's t-test on HOP replanning variants across all graphs

| | | *j* | | | | | |
|---|---|---|---|---|---|---|---|
| | | HOP-EC | | HOP-EF | | HOP-OO | |
| *i* | # agents | *t* | *p* | *t* | *p* | *t* | *p* |
| HOP-EC | 2 | — | — | -0.73 | 0.23 | 0.95 | 0.17 |
| | 3 | — | — | -0.40 | 0.34 | 1.96 | 0.03 |
| | 4 | — | — | -0.20 | 0.42 | 2.49 | 0.01 |
| | 5 | — | — | -0.19 | 0.42 | 2.81 | 0.00 |
| HOP-EF | 2 | — | — | — | — | 1.65 | 0.05 |
| | 3 | — | — | — | — | 2.32 | 0.01 |
| | 4 | — | — | — | — | 2.64 | 0.00 |
| | 5 | — | — | — | — | 2.95 | 0.00 |

Overall, we propose that EF is the best order to re-plan agents for HOP and OMT based upon our findings. This result is surprising as it indicates that both OMT and HOP generate low-cost paths with multiple agents when only considering their current vertex's expected cost, not any prior information in the form of the original order of planning or the cost incurred to reach the current vertex.

### 6.1.4 Impact of number of Agents

One of the key items this thesis seeks to address is to quantify the relative performance gains achieved by adding agents. To accomplish this, we plot the cost of the agents versus the number of agents. We first investigate OMT using the total expected cost replanning policy across the set of Delaunay graphs with 20, 50, 100, 250, 500, 750, 1000, 1500, and 2000 vertices. These results were generated before the analysis of best replanning policy was complete, hence why OMT-EF was not used. These results are plotted in fig. 8. Given that there exists a known lower bound of possible costs defined by the weathers, we expect a relationship to have a horizontal asymptote at or above that theoretical lower limit (i.e., when the competitive ratio equals 1). Across all graph sizes, the line of best fit follows an exponential decay function of the form:

$$y = A \cdot \exp(-B(\ell - 1)) + C, \tag{5}$$

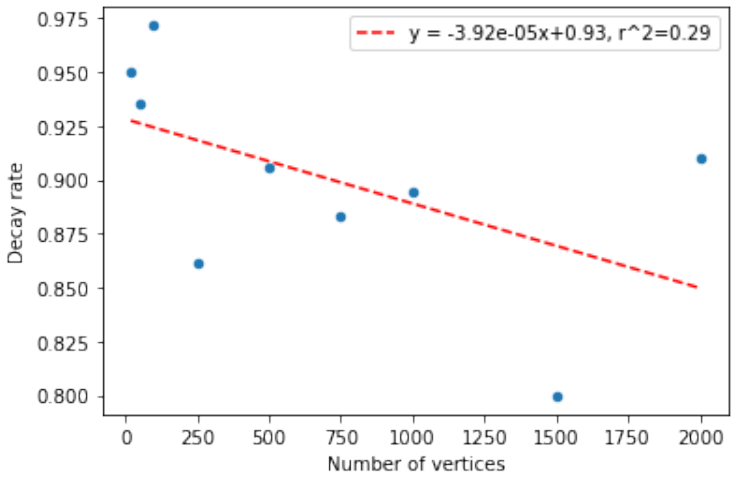



Figure 5: Decay parameter of OMT-EC

where $\ell$ is the number of agents on a graph, C is the y-value of the asymptote, A is the magnitude of the cost of a single agent above the asymptote, and B is the rate at which the the cost decreases when adding an agent. The corresponding parameters for figs. 7 and 8 are located in table 7. A number of key findings result from these data. First, we initially expected that the value of C would be equivalent to the best possible cost. However, that is not the case. We see an offset from the best possible costs, indicating that even with an infinite number of agents, neither OMT nor HOP would not reach the best possible cost. While this is not directly predicted, we can draw comparisons with some of the theoretical work completed in [15], in which contains a proof that both OMT and HOP will underestimate the optimal policy. Secondly, if the exponential decay rate, B, is known, we can estimate the benefit provided by each additional agent. Given our optimist policy (OMT) on random Delaunay graphs, the relationship between the size of the graph and best fit for the decay rate of OMT is near constant as shown in fig. 5. The linear best fit for the decay rates for HOP has a positive slope as seen in fig. 6, however is also nearly constant. Both lines of best fit for HOP and OMT have $R^2$ values of 0.29, indicating that in both cases the residual sum of squares of the best fit line is similar in magnitude to that of the mean.

Across all graphs tested, on average we see that it takes only 1-2 additional agents to outperform the UCTO policy in the trials by Eyerich *et al.* [15] for OMT, and only 1 additional agent for

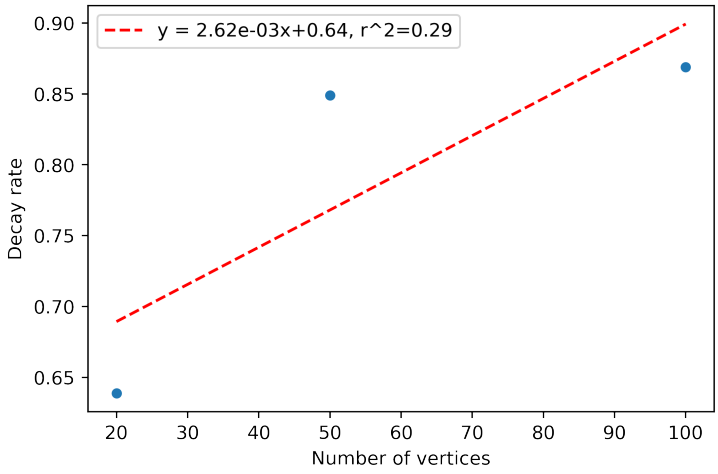



Figure 6: Decay parameter of HOP-EC

Table 7: Equations of fit lines, OMT-EC & HOP-EC

| Planning Policy | Vertices | A | B | C | Best cost | $R^2$ |
|---|---|---|---|---|---|---|
| OMT-EC | 20 | 47.4 | 0.950 | 136.8 | 124.0 | 0.997 |
| | 50 | 94.0 | 0.936 | 210.3 | 176.3 | 0.998 |
| | 100 | 124.3 | 0.972 | 263.8 | 216.7 | 0.998 |
| | 250 | 112.9 | 0.862 | 262.8 | 214.8 | 0.998 |
| | 500 | 186.3 | 0.905 | 390.4 | 301.5 | 0.999 |
| | 750 | 181.6 | 0.883 | 378.2 | 285.4 | 0.999 |
| | 1000 | 202.1 | 0.895 | 472.1 | 348.2 | 0.998 |
| | 1500 | 186.5 | 0.800 | 517.1 | 380.4 | 0.998 |
| | 2000 | 246.0 | 0.910 | 675.6 | 473.2 | 0.997 |
| HOP-EC | 20 | 28.3 | 0.639 | 130.8 | 124.0 | 0.999 |
| | 50 | 63.5 | 0.849 | 195.0 | 176.3 | 0.999 |
| | 100 | 83.2 | 0.869 | 249.5 | 216.7 | 0.999 |

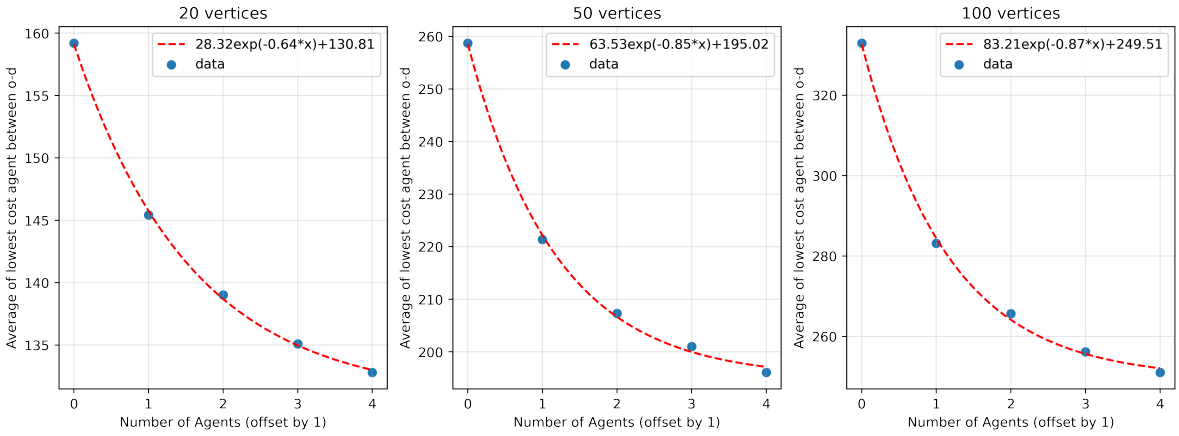



Figure 7: Impact of agents on shortest path costs, HOP-EC policy

HOP (and in some cases, even no additional agent using A* with HOP outperforms UCTO). Given that UCTO will very closely estimate the actual expectation of the cost to reach any given destination, how can a small number of additional agents using the methodologies described in this text achieve this level of performance? While these experiments are unable to directly identify the mechanism of this gain, we estimate that benefits are derived from exposing failed edges. Even highly improbable events can occur in any given weather, and quickly learning the true structure of the graph by using multiple agents allows agents to exploit the graph structure for the specific weather.

### 6.2 Trials on Random Euclidean Graphs

We compare the results of the online algorithm utilized by Shiri & Salman [29] to results generated by our solution method of $\ell$-CTP. This comparison is made to validate that the methodology developed to solve instances of $\ell$-CTP, including the usage of vertex-disjoint iterative penalty method and A* path generation, improve upon the best-known results in the literature. These data are found in tables 10 and 11 in the appendix; we index the results by the number of vertices, then number of agents, and finally percentage of edges blocked. The columns labeled '$\ell$-CTP' and 'Shiri-Salman' are the average of the competitive ratios found by the best agent in the 100 weathers for the respective algorithm. The final column, $p$, is generated from

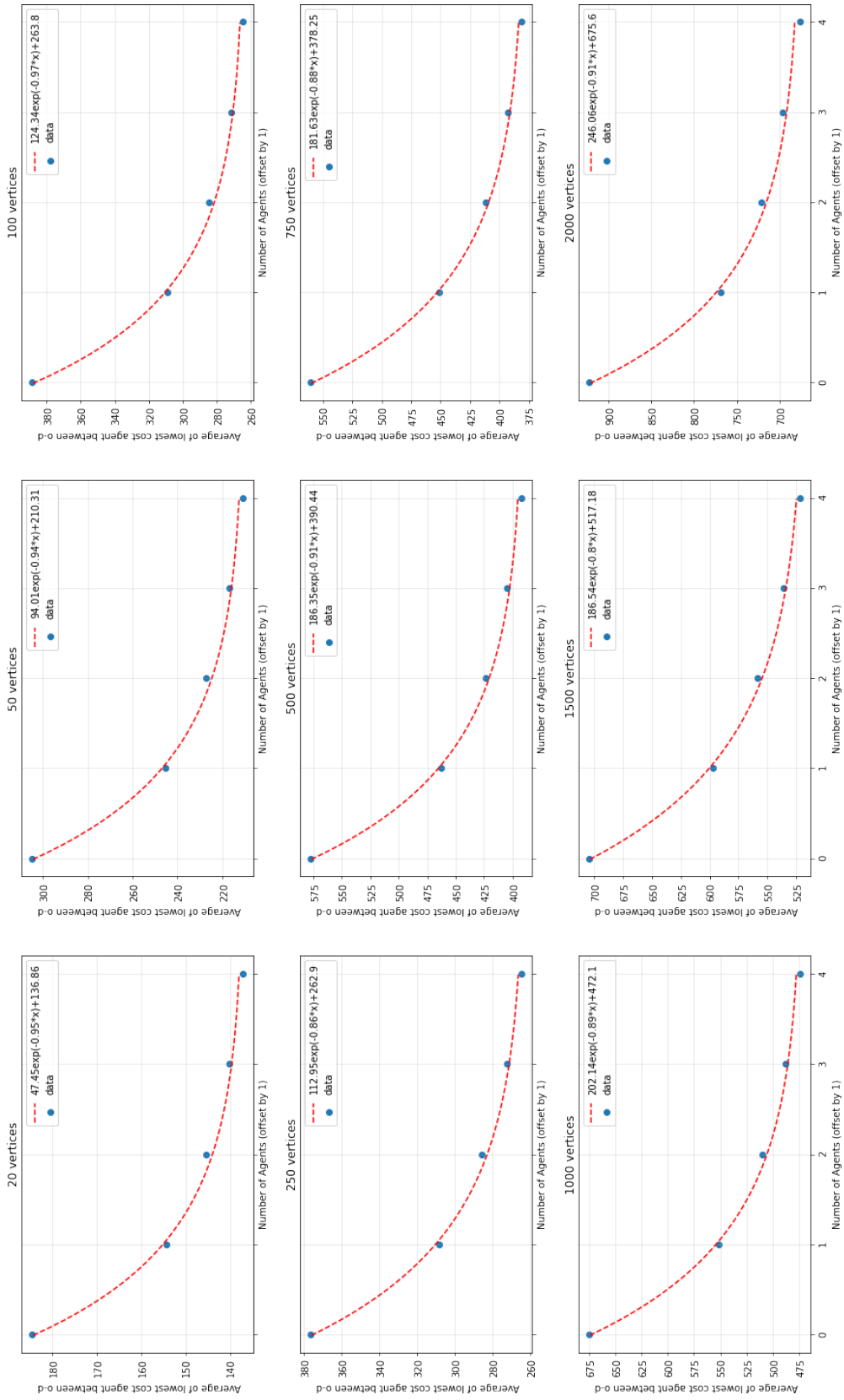



Figure 8: Impact of agents on shortest path costs, OMT-EC policy

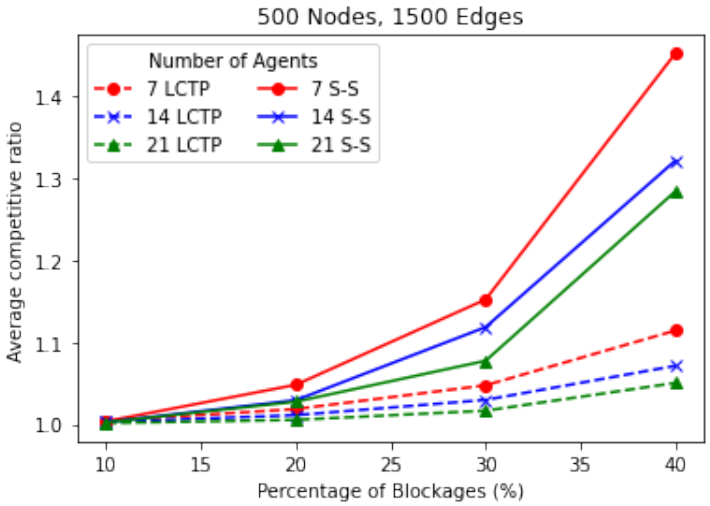



Figure 9: Comparison of performance of trials on Euclidean graphs

carrying out paired Student's t-tests between the competitive ratios of '$\ell$-CTP' and 'Shiri-Salman' in individual trials - these values are discussed at the end of this section. Overall, we find that the average competitive ratio of paths found by our implementation of $\ell$-CTP was 1.039, compared to 1.115 for the online algorithm. This represents a 6.8% improvement from the Shiri & Salman [29] results. However, the instances with 10% and 20% of failed edges will lead to relatively similar performance due to the small portion of failed edges. When filtering on the trials with edge failures of 10% and 20%, the competitive ratios for instances of $\ell$-CTP and Shiri-Salman were 1.012 and 1.016, respectively. Comparing the tests which look at 30% and 40% failed edges is more illustrative of the differences between the policies - the performance gap jumps to 1.066 and 1.214, a 12.2% improvement. We recreated a figure visualizing data from the Euclidean random graph with 500 vertices and 1500 edges from [29], shown in fig. 9. The solid lines are the results from the original trials, and the dashed lines indicate the data generated when solving the $\ell$-CTP instances reduced from Euclidean graphs with the same weathers.

These results are attributable to two differences between our methodology and the online algorithm from [29]. First, the penalty factors used in our approach are tuned for OMT to a value of 0.5, whereas the online algorithm has a penalty factor of 1.0 and was not mentioned to have been tuned. As shown in fig. 4, path costs increased for OMT after the penalty factor was raised

past 0.6, indicating routes were too divergent. Secondly, the online algorithm only utilizes IPM for the initial path generation process. Upon encountering a failed edge, the edge is removed from the visible set and any agent utilizing the edge generates a new shortest path from their current node to $v_d$ without incorporating the path of other agents. With no penalizing factor, there is no incentive for agents to explore the graph further, potentially reducing the exploration by the agents. The algorithm we developed maintains exploration of the graph using IPM with replanning methodologies described in §4.4.2 until a guaranteed path from $v_o$ to $v_d$ exists.

We additionally carry out statistical analysis of these results to determine their significance using Student's t-test. We first define $\mu_D = \mu_{Shiri-Salman} - \mu_{OMT-EF}$. The hypotheses we test are as follows: $H_0 : \mu_D = 0$, $H_1 : \mu_D > 0$. We reject the null hypothesis in favor of the alternative when $p < \alpha = 0.05$. We first carry out Student's t-test over all graphs, number of agents, and percentage of edge failures. Across all tests, we find that $p = 1.01 \times 10^{-7}$. As such, with all examples grouped together we reject the null hypothesis. We carry out the tests at a more granular level, testing at each of the unique combinations of graphs, agents, and edge failures. The remaining results are reported in tables 10 and 11. We reject $H_0$ in favor of $H_1$ in all cases with 30% and 40% of edges failed. We fail to reject $H_0$ in all instances with 10% of edges failed, and in 20% of edges failed we reject $H_0$ on graphs of size 100, 200, 300, and 400. With low percentages of edges failed, we cannot determine conclusively the benefit of our solution methodology. This does not necessarily reflect any failures of our approach, rather the competitive ratios found by both algorithms indicate routes are very nearly optimal; the average of the competitive ratios found in graphs with 10% of edges failed is less that 1% away from optimal in all cases. As such, there is little room for improvement between our methodology and the online algorithm. On the other hand, with larger percentages of edges failed, we conclude that our solution approach produces a significant improvement in performance, compared with the prior results in the literature.

Table 8: Results of 100 weathers on HOU graph

| Policy | Best path length (m) | Path length by # of agents (m) 1 | 2 | 3 |
|---|---|---|---|---|
| HOP-EC | 35910 | 77177 | 64207 | 56712 |
| OMT-EF | 35910 | 106102 | 89237 | 78590 |

Table 9: Equations of best fit lines on HOU

| Policy | A | B | C | $R^2$ |
|---|---|---|---|---|
| HOP-EC | 30726 | 0.548 | 46451 | 1.000 |
| OMT-EF | 45740 | 0.460 | 60362 | 1.000 |

### 6.3 Case Study for Hurricane Harvey

In the case study for Hurricane Harvey, we test the performance of OMT-EF and HOP-EC in 100 weathers on a representation of the Houston road network, travelling from an origin south of downtown Houston to a hospital to the northwest. These results were generated before the analysis of best replanning policy was complete, hence why HOP-EF was not used. The results generated are found in table 8. We list the best possible path length, along with the path lengths found (in units of meters) for the given number of agents. The results are plotted in fig. 10, with the exponential decay best fit line included. We first note that the exponential fit function found in §6.1.4 carries over to the results on this complex graph, which is initially shown in fig. 3. Given the 3 parameters, we are able to fit the curve using only 3 datapoints - 1, 2, and 3 agents. The parameters of the fit lines are tabulated in table 9. The parameters predict the best possible path length for agents using OMT-EF on average is 60,362 m. Given that the average shortest possible path length is 35,910 m, using OMT-EF we are restricted to a best path more than 68% longer than the best possible path. We see that HOP-EC generates lower costs paths than OMT-EF, as its best possible length is predicted as 45,740 m. Additionally, values of the B parameter indicate that HOP-EC path lengths decrease more per agent added than OMT-EF, with $B_{HOP} = 0.548$ and $B_{OMT} = 0.460$.

We look at a representative simulation to develop intuition as to why the best possible path length

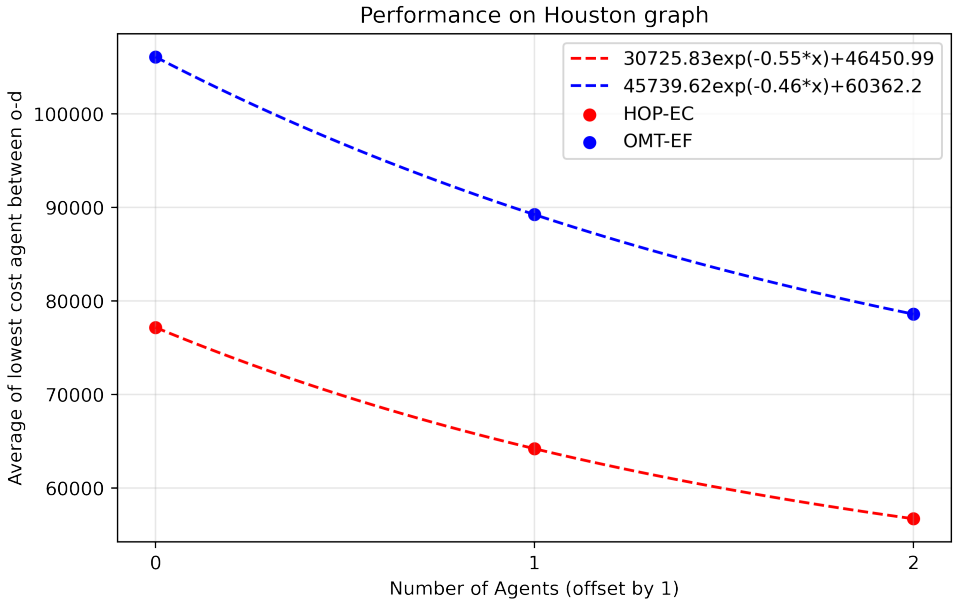
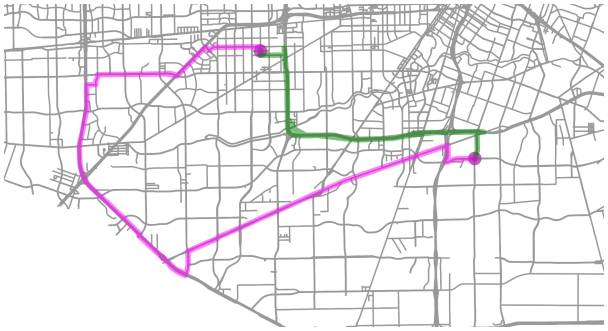


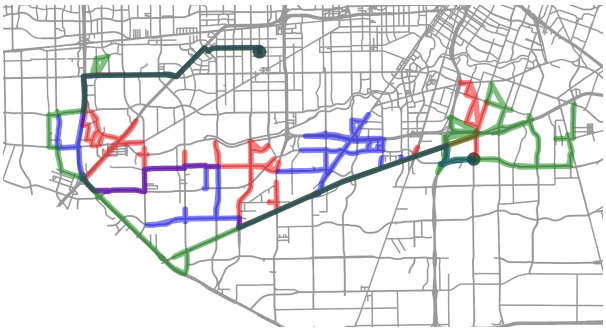

Figure 10: Results of $\ell$-CTP instances on Houston road network

for OMT is inflated compared to HOP. For this, we have plotted the path of a representative $\ell$-CTP run with 3 agents, as seen in fig. 11b. The red, blue and green paths are those generated by agent number 1, 2 and 3, respectively. To get a better sense of the impact the weather has on the graph, we plot the shortest paths with no blockages (green path) and shortest path with blockages from the current weather (magenta) in fig. 11a. The results in fig. 11a indicate the increase in shortest path length is in part due to major north/south paths restricted by flooding, such that the shortest path in our sampled weather requires detouring around the west side of the graph. Contrasting the paths generated by HOP-EC (fig. 11c) and OMT-EF (fig. 11b), we first notice that OMT explores more of the possible north/south main roads, whereas HOP-EC is more 'cautious' and all three agents travel along many of the same edges, making excursions to explore non-penalized routes of similar length. This is at least in part due to the weighting factor of IPM. Given that $w_{HOP} = 0.2$, $w_{OMT} = 0.5$, HOP is penalized less than OMT for traveling the same path.

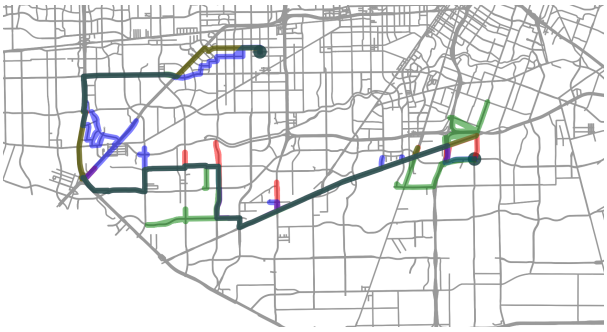
While OMT was shown outperform UCTO on random graphs with the addition of only 1-2



(a) Comparison of best path in normal conditions (green) to best path in example weather (magenta)



(b) Performance of OMT-EF on Houston road network in weather



(c) Performance of HOP-EC on Houston road network in weather

Figure 11: Performance of OMT-EC on Houston road network

agents, these results on the Houston road network indicate that OMT with a large number of agents can only achieve performance roughly equivalent to that of 2 agents with HOP. We have provided a scenario which is difficult for agents using OMT to deal with, as the optimistic assumption of all roads being open is invalid in the weathers generated. The cost estimates by HOP are better able to determine that many of the north-south roads are unavailable through sampling of possible weathers. This indicates that while the performance differential between HOP and OMT on random graphs is less pronounced, knowledge of the likelihood of edge availability can enable HOP to find low-cost routes in graphs where OMT cannot.

## 7 Conclusions

This thesis has demonstrated a robust framework for $\ell$-CTP, illustrating the various benefits of adding agents to aid in the online discovery of the shortest paths in a network with failed edges. Our methodology for solving instances of $\ell$-CTP utilizes multiple path generation methods found in the literature for single agent CTP, and combines these with heuristics to incentivize exploration among various agents. We first developed a formulation of $\ell$-CTP, beginning by developing intuition around the performance of agents when using A$^*$ to generate paths across networks for both policies HOP and OMT. We carried out parameter tuning for the iterative penalty method with 3 agents using both OMT and HOP policies, which led to agents taking routes with lengths within 20% of the optimal path length.

We carried out a wide set of simulations on graphs utilized by Eyerich *et al.* [15], and analyzed these results with statistical methods to demonstrate the equivalence between the simulated weathers on which our approach was tested and the results from Eyerich *et al.* [15]. These results demonstrated statistically significant benefits of utilizing A$^*$ to generate routes versus utilizing a greedy, vertex-by-vertex approach for the HOP policy. We also investigated a variety of techniques for generating new agent plans when agents discover failed edges which impede their movement. We show that the order in which a group of agents develop new plans can affect the length of the best path found. The best method by which agents develop new plans does not depend upon the policy those agents are utilizing; for both HOP and OMT, having agents generate plans in their order of expected future cost is best. Finally, we analyzed the impact of multiple agents. The length of the best path found ($y$) as a function of the number of agents ($\ell$) is modeled by an exponential decay curve of the form $y = A \cdot \exp(-B(\ell - 1)) + C$. This holds true for all the random graphs tested, and for agents which utilize HOP and OMT. The parameters of this equation provide insight into the performance, and limits, of a large number of agents. Generally, exponential decay indicates that most of the benefit of multiple agents (using appropriate planning methodologies) is derived from the first 2-5 agents. After the first few

agents, each additional agent provides a decreasing benefit defined by $B$. The asymptote, $C$, defines the best possible cost achievable by agents using a specific planning policy.

We compare our solution approach to $\ell$-CTP to the results of Shiri & Salman [29], who carried out the only simulation of multi-agent CTP in the literature prior to the experiments outlined in this text as known to the author. Through statistical analysis, we demonstrate that our solution approach to $\ell$-CTP using the OMT-EF policy outperforms the online algorithm developed by Shiri & Salman [29] on random Euclidean graphs with variable edge failures. The exact reason for this performance is not certain, however two key factors are attributed: the tuning of the penalty factor, and maintaining diverse paths between all agents throughout the implementation of $\ell$-CTP.

The final experiment is the application of $\ell$-CTP to the Houston road network, with edge probabilities derived from flooding simulations of hurricane Harvey in 2017. Our solution approach still exhibited exponential decay of path cost as a function of the number of agents, however for OMT-EF the asymptote, $C$, was 68% greater than best possible path lengths, compared to HOP-EC which was only 28% greater. This indicates that for large and complex graphs, OMT-EF is limited in its best possible performance on graphs where the shortest path takes a significantly different route as compared to the shortest path on a graph with no failed edges. This indicates that while OMT may be able to generate low-cost paths on random graphs, the ability to estimate likely path costs allow HOP to significantly outperform OMT when edge costs and probabilities are not randomly distributed.

The results in this thesis are promising for application to real-world scenarios, however a number of questions arise. First, while OMT allows for a simple implementation on any graph, serious limitations in the generation of paths can occur when its freespace assumption is not valid. There may be ways to incorporate information into our cost estimates without using HOP, which, while relatively straightforward, does require a rigorous implementation. Second, while we have modeled the best path cost as an exponential decay function, the mechanisms for this decrease are

not well understood. Intuitively we understand the trade-off between exploration and exploitation, where adding more agents incurs the cost of their usage to learn more information about the graph. We propose that this benefit is derived from the ability to sample edges and exploit differences between the expectation of path cost with the weather agents are operating in. A surprising result in the data is that the agent which incurs the lowest path cost in the vast majority of simulations (95%+) is the first agent - the agent which began traversing the graph with the best expected path for that policy. This implies that in most cases, the other agents are revealing information about the graph which the first agent exploits to generate a better path. Determining which information leads to better path generation could lead to improved online pathfinding algorithms.

There exist a number of paths to extend this work. The application of multiple agents all traveling from a single source to a single destination is a limited use case. We can very easily expand this $\ell$-CTP framework to answer more interesting questions. In a scenario where multiple agents have separate origin and destination nodes, can we see similar performance benefits from sharing information? This scenario arises during normal operations of a fleet of vehicles, such as delivery vehicles or emergency responders with separate tasks operating in the same geographical area.

We make a number of simplifying assumptions with regards to the weathers and the probabilities that any particular edge may be available. First, we assume that weathers remain constant. When we know that the estimated time to traverse a graph is lower than the expected rate of change in a weather, this is a valid assumption. When edge availability may shift midway through an agent traversing a network, how does an agent (or set of agents) develop paths? Secondly, we assume independence of edge availability. Let us take the example of flooding, where an agent discovers a road is flooded. That agent could rightly update the likelihood that nearby roads of lower elevation have a lower probability of being available. Incorporating Bayesian models with prior distributions into route planning requires even more complex algorithms than the methodologies proposed in this thesis as solutions to instances of $\ell$-CTP, but could allow for better usage of

information gathered when generating routes.

# 8 Bibliography


[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network flows*. Publisher: Cambridge, Mass.: Alfred P. Sloan School of Management, Massachusetts ... 1988.

[2] Vedat Akgün, Erhan Erkut, and Rajan Batta. "On Finding dissimilar paths". en. In: *European Journal of Operational Research* 121.2 (2000), pp. 232–246.

[3] Vural Aksakalli, O. Furkan Sahin, and Ibrahim Ari. "An AO* Based Exact Algorithm for the Canadian Traveler Problem". en. In: *INFORMS Journal on Computing* 28.1 (Feb. 2016), pp. 96–111. ISSN: 1091-9856, 1526-5528. DOI: 10.1287/ijoc.2015.0668. URL: http://pubsonline.informs.org/doi/10.1287/ijoc.2015.0668 (visited on 01/27/2020).

[4] Giovanni Andreatta and Luciano Romeo. "Stochastic shortest paths with recourse". In: *Networks* 18.3 (1988). Publisher: Wiley Online Library, pp. 193–204.

[5] Marco Bender and Stephan Westphal. "An optimal randomized online algorithm for the k-Canadian Traveller Problem on node-disjoint paths". In: *Journal of Combinatorial Optimization* 30.1 (2015). Publisher: Springer, pp. 87–96.

[6] Pierre Bergé et al. "Multiple Canadians on the road: minimizing the distance competitive ratio". en. In: *Journal of Combinatorial Optimization* 38.4 (Nov. 2019), pp. 1086–1100. ISSN: 1382-6905, 1573-2886. DOI: 10.1007/s10878-019-00438-6. URL: http://link.springer.com/10.1007/s10878-019-00438-6 (visited on 01/27/2020).

[7] Dimitri P. Bertsekas and John N. Tsitsiklis. "An Analysis of Stochastic Shortest Path Problems". In: *Mathematics of Operations Research* 16.3 (Aug. 1991). Publisher: INFORMS, pp. 580–595. ISSN: 0364-765X. DOI: 10.1287/moor.16.3.580. URL: https://pubsonline.informs.org/doi/abs/10.1287/moor.16.3.580 (visited on 08/31/2020).

[8] ES Blake and DA Zelinsky. *Hurricane Harvey*. en. Tropical Cyclone Report AL092017. National Hurricane Center: National Oceanic and Atmospheric Administration, May 2018, p. 77. (Visited on 08/12/2020).

[9] David Meir Blei and Leslie Pack Kaelbling. "Shortest paths in a dynamic uncertain domain". In: *IJCAI Workshop on Adaptive Spatial Representations of Dynamic Environments*. Vol. 4. 1999, p. 2.

[10] Zahy Bnaya, Ariel Felner, and Solomon Eyal Shimony. "Canadian traveler problem with remote sensing". In: *Twenty-First International Joint Conference on Artificial Intelligence*. 2009.

[11] Geoff Boeing. “OSMnx: New Methods for Acquiring, Constructing, Analyzing, and Visualizing Complex Street Networks”. In: *Computers Environment and Urban Systems* 65 (July 2017), pp. 126–139. DOI: 10.1016/j.compenvurbsys.2017.05.004.

[12] Daniel Coles et al. “Beyond ‘flood hotspots’: Modelling emergency service accessibility during flooding in York, UK”. en. In: *Journal of Hydrology* 546 (Mar. 2017), pp. 419–436. ISSN: 0022-1694. DOI: 10.1016/j.jhydrol.2016.12.013. URL: http://www.sciencedirect.com/science/article/pii/S0022169416308022 (visited on 08/12/2020).

[13] *Delaunay triangulation*. en. Page Version ID: 974660983. Aug. 2020. URL: https://en.wikipedia.org/w/index.php?title=Delaunay_triangulation&oldid=974660983 (visited on 09/09/2020).

[14] Erik D. Demaine et al. “Canadians Should Travel Randomly”. en. In: *Automata, Languages, and Programming*. Ed. by David Hutchison et al. Vol. 8572. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 380–391. ISBN: 978-3-662-43947-0 978-3-662-43948-7. DOI: 10.1007/978-3-662-43948-7_32. URL: http://link.springer.com/10.1007/978-3-662-43948-7_32 (visited on 01/27/2020).

[15] Patrick Eyerich, Thomas Keller, and Malte Helmert. “High-Quality Policies for the Canadian Traveler’s Problem.” In: *AAAI*. 2010.

[16] Youcan Feng, David R. Judi, and Cynthia L. Rakowski. “An Assessment of the Influence of Uncertainty in Temporally Evolving Streamflow Forecasts on Riverine Inundation Modeling”. en. In: *Water* 12.3 (Mar. 2020). Number: 3 Publisher: Multidisciplinary Digital Publishing Institute, p. 911. DOI: 10.3390/w12030911. URL: https://www.mdpi.com/2073-4441/12/3/911 (visited on 08/18/2020).

[17] J. Gil and P. Steinbach. “From flood risk to indirect flood impact: evaluation of street network performance for effective management, response and repair”. en. In: *Flood Recovery, Innovation and Response I*. Vol. I. ISSN: 1743-3541, 1746-448X. London, England: WIT Press, June 2008, pp. 335–344. ISBN: 978-1-84564-132-0. DOI: 10.2495/FRIAR080321. URL: http://library.witpress.com/viewpaper.asp?pcode=FRIAR08-032-1 (visited on 08/18/2020).

[18] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. “A formal basis for the heuristic determination of minimum cost paths”. In: *IEEE transactions on Systems Science and Cybernetics* 4.2 (1968). Publisher: IEEE, pp. 100–107.

[19] David R. Judi et al. “Integrated Modeling Approach for the Development of Climate-Informed, Actionable Information”. en. In: *Water* 10.6 (June 2018). Number: 6 Publisher: Multidisciplinary Digital Publishing Institute, p. 775. DOI: 10.3390/w10060775. URL: https://www.mdpi.com/2073-4441/10/6/775 (visited on 08/11/2020).

[20] Thomas R. Knutson et al. “Global Projections of Intense Tropical Cyclone Activity for the Late Twenty-First Century from Dynamical Downscaling of CMIP5/RCP4.5 Scenarios”.

en. In: *Journal of Climate* 28.18 (Sept. 2015). Publisher: American Meteorological Society, pp. 7203–7224. ISSN: 0894-8755. DOI: 10.1175/JCLI-D-15-0129.1. URL: https://journals.ametsoc.org/jcli/article/28/18/7203/34019/Global-Projections-of-Intense-Tropical-Cyclone (visited on 08/13/2020).

[21] Levente Kocsis and Csaba Szepesvári. "Bandit based monte-carlo planning". In: *European conference on machine learning*. Springer, 2006, pp. 282–293.

[22] Sven Koenig and Yury Smirnov. "Sensor-based planning with the freespace assumption". In: *Proceedings of International Conference on Robotics and Automation*. Vol. 4. Albuquerque, NM, USA: IEEE, 1997, pp. 3540–3545. ISBN: 0-7803-3612-7. DOI: 10.1109/ROBOT.1997.606883.

[23] M. Kramer, K. Terheiden, and S. Wieprecht. "Safety criteria for the trafficability of inundated roads in urban floodings". en. In: *International Journal of Disaster Risk Reduction* 17 (Aug. 2016), pp. 77–84. ISSN: 2212-4209. DOI: 10.1016/j.ijdrr.2016.04.003. URL: http://www.sciencedirect.com/science/article/pii/S2212420915301783 (visited on 06/25/2020).

[24] Evdokia Nikolova and David R Karger. "Route Planning under Uncertainty: The Canadian Traveller Problem". In: *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*. 2008, pp. 969–974.

[25] R. Pant et al. "Critical infrastructure impact assessment due to flood exposure". en. In: *Journal of Flood Risk Management* 11.1 (2018), pp. 22–33. ISSN: 1753-318X. DOI: 10.1111/jfr3.12288. (Visited on 08/11/2020).

[26] Christos H. Papadimitriou and Mihalis Yannakakis. "Shortest paths without a map". en. In: *Theoretical Computer Science* 84.1 (July 1991), pp. 127–150. ISSN: 03043975. DOI: 10.1016/0304-3975(91)90263-2. URL: https://linkinghub.elsevier.com/retrieve/pii/0304397591902632 (visited on 01/27/2020).

[27] Kimberly Parker. *JBSA-Seguin ISB supports Hurricane Harvey disaster relief efforts*. en-US. Sept. 2017. URL: https://www.dla.mil/AboutDLA/News/NewsArticleView/Article/1299032/jbsa-seguin-isb-supports-hurricane-harvey-disaster-relief-efforts/ (visited on 08/14/2020).

[28] Stuart Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Prentice Hall Upper Saddle River, NJ, USA: 2003.

[29] Davood Shiri, Vahid Akbari, and F. Sibel Salman. "Online routing and scheduling of search-and-rescue teams". en. In: *OR Spectrum* (July 2020). ISSN: 0171-6468, 1436-6304. DOI: 10.1007/s00291-020-00594-w. URL: http://link.springer.com/10.1007/s00291-020-00594-w (visited on 07/14/2020).

[30] Davood Shiri and F. Sibel Salman. “On the online multi-agent O–D k-Canadian Traveler Problem”. In: *Journal of Combinatorial Optimization* 34.2 (2017). Publisher: Springer, pp. 453–461.

[31] Daniel D. Sleator and Robert E. Tarjan. “Amortized efficiency of list update and paging rules”. In: *Communications of the ACM* 28.2 (1985). Publisher: ACM New York, NY, USA, pp. 202–208.

[32] Adam Smith. *2010-2019: A landmark decade of U.S. billion-dollar weather and climate disasters — NOAA Climate.gov*. Jan. 2020. URL: https://www.climate.gov/news-features/blogs/beyond-data/2010-2019-landmark-decade-us-billion-dollar-weather-and-climate (visited on 08/13/2020).

[33] William VanderVeer Sweet et al. *Sea level rise and nuisance flood frequency changes around the United States*. Technical Report NOS CO-OPS 073. Silver Spring, MD: US Department of Commerce, National Oceanic and Atmospheric Administration . . ., June 2014, p. 58.

[34] Byron Tasseff, David Judi, and USDOE. *Nuflood, Version 1.x*. en. Tech. rep. Nuflood. Los Alamos National Lab. (LANL), Los Alamos, NM (United States), Aug. 2016. DOI: 10.11578/dc.20171025.1807. URL: https://www.osti.gov/biblio/1492706 (visited on 08/19/2020).

[35] Amit Uppal et al. “In Search of the Silver Lining. The Impact of Superstorm Sandy on Bellevue Hospital”. In: *Annals of the American Thoracic Society* 10.2 (Apr. 2013). Publisher: American Thoracic Society - AJRCCM, pp. 135–142. ISSN: 2329-6933. DOI: 10.1513/AnnalsATS.201212-116OT. URL: https://www.atsjournals.org/doi/full/10.1513/annalsats.201212-116ot (visited on 08/31/2020).

[36] Stephan Westphal. “A note on the k-Canadian Traveller Problem”. en. In: *Information Processing Letters* 106.3 (Apr. 2008), pp. 87–89. ISSN: 00200190. DOI: 10.1016/j.ipl.2007.10.004. URL: https://linkinghub.elsevier.com/retrieve/pii/S0020019007002876 (visited on 01/27/2020).

[37] Yinfeng Xu et al. “The Canadian traveller problem and its competitive analysis”. In: *Journal of combinatorial optimization* 18.2 (2009). Publisher: Springer, pp. 195–205.

[38] Huili Zhang, Yinfeng Xu, and Lan Qin. “The k-Canadian Travelers Problem with communication”. en. In: *Journal of Combinatorial Optimization* 26.2 (Aug. 2013), pp. 251–265. ISSN: 1382-6905, 1573-2886. DOI: 10.1007/s10878-012-9503-x. URL: http://link.springer.com/10.1007/s10878-012-9503-x (visited on 01/27/2020).

# 9 Appendix

Table 10: Comparison of Competitive Ratio between instances of $\ell$-CTP and Shiri-Salman

| Vertices | L | % Blocked | $\ell$-CTP | Shiri-Salman | $p$ |
|---|---|---|---|---|---|
| 100 | 3 | 10.0 | 1.011220 | 1.011721 | 0.476 |
| | | 20.0 | 1.038938 | 1.045371 | 0.109 |
| | | 30.0 | 1.048932 | 1.230993 | 0.000 |
| | | 40.0 | 1.107447 | 1.448394 | 0.000 |
| | 6 | 10.0 | 1.001256 | 1.004940 | 0.222 |
| | | 20.0 | 1.012129 | 1.013292 | 0.415 |
| | | 30.0 | 1.029870 | 1.093416 | 0.008 |
| | | 40.0 | 1.061844 | 1.244553 | 0.001 |
| | 9 | 10.0 | 1.000717 | 1.001200 | 0.347 |
| | | 20.0 | 1.008962 | 1.012479 | 0.249 |
| | | 30.0 | 1.023036 | 1.046323 | 0.043 |
| | | 40.0 | 1.042821 | 1.191101 | 0.002 |
| 200 | 4 | 10.0 | 1.003795 | 1.001753 | 0.246 |
| | | 20.0 | 1.032042 | 1.044586 | 0.190 |
| | | 30.0 | 1.104478 | 1.171653 | 0.033 |
| | | 40.0 | 1.186860 | 1.362152 | 0.002 |
| | 8 | 10.0 | 1.000125 | 1.000125 | 0.160 |
| | | 20.0 | 1.019616 | 1.017891 | 0.417 |
| | | 30.0 | 1.047884 | 1.091345 | 0.033 |
| | | 40.0 | 1.061113 | 1.178562 | 0.000 |
| | 12 | 10.0 | 1.000000 | 1.000125 | 0.160 |
| | | 20.0 | 1.007458 | 1.014544 | 0.092 |
| | | 30.0 | 1.021046 | 1.073407 | 0.006 |
| | | 40.0 | 1.029788 | 1.131317 | 0.000 |

Table 11: Comparison of Competitive Ratio between instances of $\ell$-CTP and Shiri-Salman, cont'd

| Vertices | L | % Blocked | $\ell$-CTP | Shiri-Salman | p |
|---|---|---|---|---|---|
| 300 | 5 | 10.0 | 1.001687 | 1.002747 | 0.069 |
| | | 20.0 | 1.058652 | 1.072134 | 0.235 |
| | | 30.0 | 1.117621 | 1.217402 | 0.001 |
| | | 40.0 | 1.172621 | 1.497565 | 0.000 |
| | 10 | 10.0 | 1.000313 | 1.001089 | 0.120 |
| | | 20.0 | 1.026305 | 1.029282 | 0.366 |
| | | 30.0 | 1.040526 | 1.095952 | 0.005 |
| | | 40.0 | 1.083595 | 1.249595 | 0.001 |
| | 15 | 10.0 | 1.000313 | 1.000958 | 0.160 |
| | | 20.0 | 1.014909 | 1.019424 | 0.249 |
| | | 30.0 | 1.033369 | 1.081265 | 0.014 |
| | | 40.0 | 1.037882 | 1.246371 | 0.000 |
| 400 | 6 | 10.0 | 1.013903 | 1.006048 | 0.138 |
| | | 20.0 | 1.029013 | 1.031149 | 0.445 |
| | | 30.0 | 1.074763 | 1.209254 | 0.001 |
| | | 40.0 | 1.144261 | 1.512231 | 0.000 |
| | 12 | 10.0 | 1.006566 | 1.005671 | 0.439 |
| | | 20.0 | 1.009146 | 1.018592 | 0.141 |
| | | 30.0 | 1.033589 | 1.102355 | 0.000 |
| | | 40.0 | 1.078010 | 1.238647 | 0.001 |
| | 18 | 10.0 | 1.005845 | 1.005671 | 0.488 |
| | | 20.0 | 1.005931 | 1.018160 | 0.104 |
| | | 30.0 | 1.026931 | 1.062473 | 0.009 |
| | | 40.0 | 1.042935 | 1.225611 | 0.000 |
| 500 | 7 | 10.0 | 1.004057 | 1.003641 | 0.379 |
| | | 20.0 | 1.018585 | 1.048129 | 0.011 |
| | | 30.0 | 1.047475 | 1.152036 | 0.001 |
| | | 40.0 | 1.114214 | 1.451561 | 0.000 |
| | 14 | 10.0 | 1.002596 | 1.002639 | 0.483 |
| | | 20.0 | 1.011188 | 1.029395 | 0.039 |
| | | 30.0 | 1.029724 | 1.118340 | 0.002 |
| | | 40.0 | 1.071229 | 1.321016 | 0.000 |
| | 21 | 10.0 | 1.001537 | 1.002431 | 0.129 |
| | | 20.0 | 1.005452 | 1.027566 | 0.017 |
| | | 30.0 | 1.016618 | 1.077370 | 0.001 |
| | | 40.0 | 1.050390 | 1.283681 | 0.000 |

Table 12: OMT-NP average path lengths on random Delaunay graphs

| | Eyerich *et al.* [15] | | $\ell$-CTP , # of agents | | | | |
|---|---|---|---|---|---|---|---|
| Graph | OMT | UCTO | 1 | 2 | 3 | 4 | 5 |
| 20-1 | 205.9 | 169.0 | 200.7 | 172.1 | **161.9** | 156.0 | 152.3 |
| 20-2 | 187.0 | 148.9 | 187.3 | 165.1 | 159.8 | 152.1 | 150.5 |
| 20-3 | 139.5 | 132.5 | 142.9 | **127.9** | 122.2 | 119.8 | 117.9 |
| 20-4 | 266.2 | 235.2 | 244.7 | **205.8** | 199.7 | 192.0 | 185.7 |
| 20-5 | 163.1 | 111.3 | 166.5 | 126.6 | 112.5 | **106.3** | 104.9 |
| 20-6 | 180.2 | 133.1 | 182.0 | 142.3 | **122.2** | 118.3 | 114.2 |
| 20-7 | 172.2 | 148.2 | 170.5 | **138.3** | 129.8 | 128.9 | 126.7 |
| 20-8 | 150.1 | 134.5 | 152.4 | 140.0 | **134.1** | 130.6 | 125.4 |
| 20-9 | 222.0 | 173.9 | 222.2 | 175.9 | **166.3** | 160.9 | 158.5 |
| 20-10 | 178.2 | 167.0 | 175.7 | **155.3** | 143.4 | 139.7 | 138.0 |
| Average | 186.5 | 154.2 | 184.5 | 154.9 | **145.2** | 140.4 | 137.4 |
| 50-1 | 255.5 | 186.1 | 254.5 | **173.8** | 163.9 | 146.6 | 144.6 |
| 50-2 | 467.1 | 365.5 | 463.6 | 367.8 | **351.1** | 334.2 | 326.8 |
| 50-3 | 281.5 | 255.6 | 289.2 | **242.2** | 223.0 | 216.1 | 209.9 |
| 50-4 | 289.8 | 230.5 | 287.6 | 235.3 | **214.3** | 203.5 | 196.4 |
| 50-5 | 285.5 | 225.4 | 285.7 | **214.0** | 202.8 | 192.7 | 189.8 |
| 50-6 | 251.3 | 236.3 | 243.0 | **229.2** | 221.1 | 217.8 | 211.4 |
| 50-7 | 242.2 | 206.3 | 235.9 | 208.7 | **189.4** | 178.3 | 174.6 |
| 50-8 | 355.1 | 277.6 | 363.6 | 284.8 | **263.2** | 254.5 | 246.1 |
| 50-9 | 327.4 | 222.5 | 331.0 | 249.2 | 223.4 | **213.9** | 212.0 |
| 50-10 | 281.6 | 240.8 | 292.5 | **238.6** | 212.3 | 194.5 | 188.6 |
| Average | 303.7 | 244.7 | 304.7 | **244.4** | 226.4 | 215.2 | 210.0 |
| 100-1 | 370.9 | 286.8 | 354.7 | 291.9 | **263.9** | 254.1 | 242.0 |
| 100-2 | 160.6 | 151.5 | 169.0 | 158.3 | 155.4 | 152.4 | 152.0 |
| 100-3 | 550.2 | 412.2 | 550.1 | **411.0** | 376.0 | 353.4 | 338.1 |
| 100-4 | 420.1 | 314.3 | 431.2 | 333.0 | 319.1 | **303.1** | 306.0 |
| 100-5 | 397.0 | 348.3 | 411.4 | **346.0** | 323.0 | 308.4 | 303.6 |
| 100-6 | 455.0 | 396.2 | 478.7 | **393.4** | 365.8 | 357.6 | 348.7 |
| 100-7 | 431.4 | 358.2 | 435.8 | **330.6** | 310.5 | 300.9 | 293.3 |
| 100-8 | 335.6 | 293.3 | 328.0 | **276.2** | 240.5 | 222.8 | 216.7 |
| 100-9 | 327.5 | 262.0 | 329.9 | 273.1 | **240.5** | 233.7 | 223.5 |
| 100-10 | 381.5 | 342.3 | 396.3 | **309.6** | 283.3 | 271.9 | 267.2 |
| Average | 383.0 | 316.5 | 388.5 | **312.3** | 287.8 | 271.8 | 269.1 |

Red highlighted box indicates instance where average performance decreases.

Table 13: OMT-NP analysis of performance decrease across number of agents

| | Number of agents | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | | 2 | | 3 | | 4 | | 5 | |
| Graph | # | % | # | % | # | % | # | % | # | % |
| 20-1 | — | — | 113 | 23.9 | 70 | 32.5 | 39 | 23.1 | 38 | 17.1 |
| 20-2 | — | — | 68 | 9.7 | 57 | 14.8 | 60 | 14.0 | 47 | 10.0 |
| 20-3 | — | — | 16 | 17.4 | 21 | 30.6 | 13 | 22.4 | 11 | 13.2 |
| 20-4 | — | — | 80 | 41.9 | 31 | 44.3 | 41 | 49.1 | 33 | 15.7 |
| 20-5 | — | — | 25 | 30.0 | 51 | 57.3 | 17 | 15.1 | 13 | 21.8 |
| 20-6 | — | — | 74 | 23.9 | 37 | 7.5 | 76 | 22.8 | 52 | 10.1 |
| 20-7 | — | — | 35 | 14.8 | 19 | 10.6 | 42 | 17.7 | 13 | 13.5 |
| 20-8 | — | — | 28 | 9.2 | 23 | 11.8 | 173 | 21.9 | 28 | 16.5 |
| 20-9 | — | — | 86 | 13.8 | 113 | 19.3 | 97 | 21.0 | 104 | 17.5 |
| 20-10 | — | — | 17 | 19.1 | 32 | 15.1 | 22 | 9.8 | 29 | 11.6 |
| Average | — | — | — | 20.4 | — | 24.4 | — | 21.7 | — | 14.7 |
| Total | — | — | 542 | — | 454 | — | 580 | — | 368 | — |
| 50-1 | — | — | 38 | 30.8 | 65 | 43.6 | 43 | 18.0 | 39 | 13.7 |
| 50-2 | — | — | 85 | 12.9 | 209 | 13.8 | 209 | 13.4 | 246 | 12.9 |
| 50-3 | — | — | 215 | 16.8 | 195 | 13.6 | 213 | 14.4 | 150 | 12.9 |
| 50-4 | — | — | 137 | 10.9 | 119 | 12.8 | 152 | 13.5 | 132 | 11.4 |
| 50-5 | — | — | 55 | 31.8 | 137 | 23.3 | 101 | 13.4 | 158 | 13.0 |
| 50-6 | — | — | 129 | 8.7 | 79 | 9.4 | 133 | 11.4 | 80 | 9.8 |
| 50-7 | — | — | 190 | 49.0 | 102 | 22.0 | 79 | 10.2 | 79 | 12.6 |
| 50-8 | — | — | 113 | 18.0 | 215 | 9.7 | 196 | 11.5 | 177 | 11.4 |
| 50-9 | — | — | 66 | 21.1 | 121 | 14.8 | 114 | 11.8 | 121 | 18.8 |
| 50-10 | — | — | 244 | 21.3 | 125 | 19.4 | 137 | 13.3 | 123 | 14.1 |
| Average | — | — | — | 22.1 | — | 18.2 | — | 13.1 | — | 13.1 |
| Total | — | — | 1272 | — | 1367 | — | 1377 | — | 1305 | — |
| 100-1 | — | — | 89 | 17.7 | 89 | 14.7 | 114 | 16.7 | 96 | 20.7 |
| 100-2 | — | — | 86 | 12.4 | 120 | 11.0 | 65 | 11.8 | 69 | 14.1 |
| 100-3 | — | — | 90 | 17.5 | 127 | 14.6 | 170 | 14.1 | 231 | 13.3 |
| 100-4 | — | — | 128 | 16.4 | 243 | 18.2 | 249 | 13.3 | 362 | 17.3 |
| 100-5 | — | — | 92 | 12.8 | 162 | 12.4 | 158 | 12.3 | 198 | 11.0 |
| 100-6 | — | — | 138 | 13.1 | 173 | 11.6 | 263 | 10.7 | 231 | 11.0 |
| 100-7 | — | — | 117 | 17.1 | 194 | 15.3 | 223 | 14.2 | 227 | 10.9 |
| 100-8 | — | — | 143 | 14.8 | 101 | 10.2 | 95 | 10.9 | 98 | 12.6 |
| 100-9 | — | — | 86 | 20.0 | 51 | 11.3 | 89 | 8.0 | 69 | 11.2 |
| 100-10 | — | — | 146 | 15.6 | 178 | 14.8 | 234 | 14.0 | 270 | 13.1 |
| Average | — | — | — | 15.7 | — | 13.4 | — | 12.6 | — | 13.5 |
| Total | — | — | 1115 | — | 1438 | — | 1660 | — | 1851 | — |

Table 14: OMT-OO average path lengths on random Delaunay graphs

| | Eyerich *et al.* [15] | | $\ell$-CTP , # of agents | | | | |
|---|---|---|---|---|---|---|---|
| Graph | OMT | UCTO | 1 | 2 | 3 | 4 | 5 |
| 20-1 | 205.9 | 169.0 | 200.7 | 169.9 | **154.0** | 148.6 | 147.4 |
| 20-2 | 187.0 | 148.9 | 187.3 | 166.4 | 153.4 | 149.2 | **146.5** |
| 20-3 | 139.5 | 132.5 | 142.9 | **125.6** | 121.8 | 119.9 | 119.1 |
| 20-4 | 266.2 | 235.2 | 244.7 | **205.1** | 193.6 | 186.0 | 181.7 |
| 20-5 | 163.1 | 111.3 | 166.5 | 122.4 | 114.2 | **106.5** | 103.2 |
| 20-6 | 180.2 | 133.1 | 182.0 | 136.5 | **125.5** | 121.2 | 116.7 |
| 20-7 | 172.2 | 148.2 | 170.5 | **137.2** | 132.1 | 128.8 | 127.2 |
| 20-8 | 150.1 | 134.5 | 152.4 | 141.1 | **133.6** | 127.6 | 125.3 |
| 20-9 | 222.0 | 173.9 | 222.2 | 180.5 | **170.2** | 164.7 | 162.7 |
| 20-10 | 178.2 | 167.0 | 175.7 | **156.7** | 150.6 | 143.7 | 141.6 |
| Average | 186.5 | 154.2 | 184.5 | **154.1** | 144.9 | 139.6 | 137.2 |
| 50-1 | 255.5 | 186.1 | 254.5 | **170.3** | 158.9 | 145.2 | 143.1 |
| 50-2 | 467.1 | 365.5 | 463.6 | **392.5** | 352.4 | 341.9 | 329.1 |
| 50-3 | 281.5 | 255.6 | 289.2 | **240.0** | 221.9 | 212.8 | 209.1 |
| 50-4 | 289.8 | 230.5 | 287.6 | 238.7 | **214.6** | 203.6 | 197.6 |
| 50-5 | 285.5 | 225.4 | 285.7 | **215.3** | 202.4 | 194.1 | 190.2 |
| 50-6 | 251.3 | 236.3 | 243.0 | **225.9** | 215.6 | 210.8 | 206.3 |
| 50-7 | 242.2 | 206.3 | 235.9 | **205.0** | 187.7 | 178.9 | 173.3 |
| 50-8 | 355.1 | 277.6 | 363.6 | 290.5 | **261.0** | 249.5 | 243.8 |
| 50-9 | 327.4 | 222.5 | 331.0 | 246.5 | **220.4** | 207.2 | 200.8 |
| 50-10 | 281.6 | 240.8 | 292.5 | **223.9** | 203.7 | 197.6 | 191.1 |
| Average | 303.7 | 244.7 | 304.7 | 244.9 | **223.9** | 214.2 | 208.4 |
| 100-1 | 370.9 | 286.8 | 354.7 | 289.2 | **259.8** | 245.5 | 237.0 |
| 100-2 | 160.6 | 151.5 | 169.0 | 156.6 | 154.8 | 153.0 | 151.8 |
| 100-3 | 550.2 | 412.2 | 550.1 | **410.8** | 372.5 | 353.2 | 336.5 |
| 100-4 | 420.1 | 314.3 | 431.2 | 333.4 | 315.4 | **303.9** | 298.0 |
| 100-5 | 397.0 | 348.3 | 411.4 | **344.5** | 323.7 | 307.3 | 300.5 |
| 100-6 | 455.0 | 396.2 | 478.7 | 398.7 | **372.1** | 358.7 | 351.2 |
| 100-7 | 431.4 | 358.2 | 435.8 | **333.4** | 309.0 | 297.4 | 288.8 |
| 100-8 | 335.6 | 293.3 | 328.0 | **260.9** | 236.8 | 223.6 | 216.1 |
| 100-9 | 327.5 | 262.0 | 329.9 | 271.3 | **240.1** | 234.1 | 223.3 |
| 100-10 | 381.5 | 342.3 | 396.2 | **312.9** | 287.9 | 277.3 | 265.9 |
| Average | 383.0 | 316.5 | 388.5 | **311.2** | 287.2 | 275.4 | 266.9 |

Table 15: OMT-OO analysis of performance decrease across number of agents

| | Number of agents | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | | 2 | | 3 | | 4 | | 5 | |
| Graph | # | % | # | % | # | % | # | % | # | % |
| 20-1 | — | — | 0 | — | 7 | 7.6 | 5 | 10.5 | 3 | 12.3 |
| 20-2 | — | — | 0 | — | 2 | 47.7 | 4 | 41.3 | 2 | 5.1 |
| 20-3 | — | — | 6 | 9.6 | 1 | 36.8 | 2 | 10.0 | 0 | — |
| 20-4 | — | — | 0 | — | 2 | 27.7 | 3 | 7.3 | 4 | 14.5 |
| 20-5 | — | — | 0 | — | 1 | 10.9 | 2 | 6.7 | 1 | 12.1 |
| 20-6 | — | — | 4 | 14.4 | 5 | 11.6 | 2 | 6.1 | 2 | 7.9 |
| 20-7 | — | — | 1 | 1.4 | 0 | — | 1 | 4.2 | 1 | 40.4 |
| 20-8 | — | — | 1 | 16.9 | 1 | 15.7 | 1 | 12.6 | 1 | 12.4 |
| 20-9 | — | — | 2 | 15.7 | 4 | 17.9 | 6 | 32.2 | 12 | 11.1 |
| 20-10 | — | — | 0 | — | 1 | 30.5 | 2 | 8.7 | 1 | 12.7 |
| Average | — | — | — | 11.6 | — | 22.9 | — | 14.0 | — | 14.3 |
| Total | — | — | 14 | — | 24 | — | 28 | — | 27 | — |
| 50-1 | — | — | 2 | 9.4 | 11 | 9.2 | 9 | 9.5 | 6 | 8.5 |
| 50-2 | — | — | 5 | 3.7 | 42 | 8.5 | 41 | 8.3 | 66 | 9.7 |
| 50-3 | — | — | 3 | 2.7 | 10 | 12.7 | 15 | 12.5 | 18 | 13.8 |
| 50-4 | — | — | 9 | 6.5 | 19 | 11.3 | 31 | 11.3 | 11 | 10.3 |
| 50-5 | — | — | 4 | 27.7 | 17 | 15.9 | 36 | 20.9 | 12 | 5.0 |
| 50-6 | — | — | 2 | 20.2 | 9 | 14.7 | 13 | 7.0 | 5 | 17.1 |
| 50-7 | — | — | 5 | 20.5 | 20 | 12.9 | 16 | 7.5 | 12 | 10.4 |
| 50-8 | — | — | 7 | 10.5 | 37 | 10.7 | 22 | 9.1 | 21 | 10.7 |
| 50-9 | — | — | 8 | 22.7 | 9 | 13.2 | 12 | 13.1 | 19 | 17.8 |
| 50-10 | — | — | 7 | 11.8 | 13 | 20.1 | 21 | 7.9 | 19 | 13.7 |
| Average | — | — | — | 13.6 | — | 12.9 | — | 10.7 | — | 11.7 |
| Total | — | — | 52 | — | 187 | — | 216 | — | 189 | — |
| 100-1 | — | — | 5 | 4.6 | 10 | 7.1 | 26 | 10.6 | 27 | 10.3 |
| 100-2 | — | — | 2 | 15.3 | 6 | 9.0 | 5 | 11.5 | 3 | 1.7 |
| 100-3 | — | — | 18 | 15.4 | 29 | 9.7 | 42 | 6.4 | 56 | 9.6 |
| 100-4 | — | — | 23 | 11.2 | 22 | 11.9 | 44 | 14.6 | 31 | 10.5 |
| 100-5 | — | — | 3 | 8.4 | 38 | 10.4 | 41 | 9.3 | 52 | 9.7 |
| 100-6 | — | — | 13 | 6.9 | 36 | 10.6 | 50 | 8.1 | 63 | 9.3 |
| 100-7 | — | — | 10 | 8.1 | 23 | 6.6 | 31 | 10.9 | 38 | 9.7 |
| 100-8 | — | — | 8 | 10.3 | 23 | 7.8 | 19 | 6.5 | 26 | 9.4 |
| 100-9 | — | — | 2 | 10.2 | 13 | 16.1 | 12 | 19.2 | 44 | 9.2 |
| 100-10 | — | — | 7 | 8.6 | 40 | 10.4 | 59 | 12.4 | 56 | 8.2 |
| Average | — | — | — | 9.9 | — | 10.0 | — | 11.0 | — | 8.8 |
| Total | — | — | 91 | — | 240 | — | 329 | — | 39.6 | — |

Table 16: OMT-EC average path lengths on random Delaunay graphs

| | Eyerich *et al.* [15] | | $\ell$-CTP , # of agents | | | | |
|---|---|---|---|---|---|---|---|
| Graph | OMT | UCTO | 1 | 2 | 3 | 4 | 5 |
| 20-1 | 205.9 | 169.0 | 200.7 | 170.7 | **156.7** | 150.9 | 146.5 |
| 20-2 | 187.0 | 148.9 | 187.3 | 164.9 | 152.7 | **148.6** | 146.1 |
| 20-3 | 139.5 | 132.5 | 142.9 | **125.7** | 122.0 | 120.0 | 119.2 |
| 20-4 | 266.2 | 235.2 | 244.7 | **204.4** | 192.9 | 186.1 | 181.9 |
| 20-5 | 163.1 | 111.3 | 166.5 | 123.0 | 114.4 | **106.4** | 103.6 |
| 20-6 | 180.2 | 133.1 | 182.0 | 135.4 | **124.4** | 120.3 | 116.6 |
| 20-7 | 172.2 | 148.2 | 170.5 | **137.2** | 131.7 | 128.2 | 126.5 |
| 20-8 | 150.1 | 134.5 | 152.4 | 140.7 | **134.1** | 128.6 | 125.4 |
| 20-9 | 222.0 | 173.9 | 222.2 | 182.8 | **173.8** | 167.4 | 163.4 |
| 20-10 | 178.2 | 167.0 | 175.7 | **157.2** | 150.1 | 143.8 | 140.8 |
| Average | 186.5 | 154.2 | 184.5 | **154.2** | 145.3 | 140.0 | 137.0 |
| 50-1 | 255.5 | 186.1 | 254.5 | **174.5** | 159.5 | 145.8 | 143.7 |
| 50-2 | 467.1 | 365.5 | 463.6 | 387.8 | **362.2** | 350.5 | 333.2 |
| 50-3 | 281.5 | 255.6 | 289.2 | **241.4** | 224.3 | 217.0 | 212.5 |
| 50-4 | 289.8 | 230.5 | 287.6 | 240.4 | **224.4** | 208.7 | 201.7 |
| 50-5 | 285.5 | 225.4 | 285.7 | **214.7** | 202.4 | 192.9 | 190.1 |
| 50-6 | 251.3 | 236.3 | 243.0 | **225.7** | 216.4 | 211.9 | 208.6 |
| 50-7 | 242.2 | 206.3 | 235.9 | **205.6** | 190.2 | 180.4 | 174.5 |
| 50-8 | 355.1 | 277.6 | 363.6 | **292.0** | 264.2 | 252.3 | 246.0 |
| 50-9 | 327.4 | 222.5 | 331.0 | 248.0 | **221.9** | 209.8 | 202.6 |
| 50-10 | 281.6 | 240.8 | 292.5 | **223.2** | 205.8 | 200.2 | 194.5 |
| Average | 303.7 | 244.7 | 304.7 | 245.3 | **227.1** | 217.0 | 210.7 |
| 100-1 | 370.9 | 286.8 | 354.7 | **283.7** | 256.0 | 241.2 | 233.5 |
| 100-2 | 160.6 | 151.5 | 169.0 | 155.7 | 153.0 | **151.1** | 149.7 |
| 100-3 | 550.2 | 412.2 | 550.1 | **409.0** | 370.1 | 346.2 | 336.3 |
| 100-4 | 420.1 | 314.3 | 431.2 | 328.7 | **310.5** | 296.0 | 293.1 |
| 100-5 | 397.0 | 348.3 | 411.4 | **340.5** | 319.0 | 304.0 | 296.1 |
| 100-6 | 455.0 | 396.2 | 478.7 | 398.5 | **372.7** | 357.5 | 350.5 |
| 100-7 | 431.4 | 358.2 | 435.8 | **333.7** | 305.6 | 292.9 | 285.4 |
| 100-8 | 335.6 | 293.3 | 328.0 | **260.3** | 234.2 | 223.1 | 215.6 |
| 100-9 | 327.5 | 262.0 | 329.9 | 267.7 | **240.6** | 232.3 | 224.1 |
| 100-10 | 381.5 | 342.3 | 396.2 | **311.0** | 281.3 | 269.4 | 260.4 |
| Average | 383.0 | 316.5 | 388.5 | **311.2** | 287.2 | 275.4 | 266.9 |

Table 17: OMT-EC analysis of performance decrease across number of agents

| | Number of agents | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | | 2 | | 3 | | 4 | | 5 | |
| Graph | # | % | # | % | # | % | # | % | # | % |
| 20-1 | — | — | 1 | 12.5 | 21 | 28.6 | 15 | 11.0 | 4 | 4.8 |
| 20-2 | — | — | 3 | 10.3 | 8 | 15.8 | 15 | 8.4 | 6 | 9.0 |
| 20-3 | — | — | 6 | 9.6 | 2 | 15.5 | 4 | 14.5 | 2 | 7.5 |
| 20-4 | — | — | 0 | — | 0 | — | 5 | 25.2 | 9 | 25.2 |
| 20-5 | — | — | 6 | 49.1 | 4 | 12.0 | 1 | 7.7 | 3 | 15.7 |
| 20-6 | — | — | 5 | 13.2 | 14 | 23.6 | 4 | 11.4 | 15 | 9.5 |
| 20-7 | — | — | 2 | 5.2 | 4 | 13.5 | 0 | — | 0 | — |
| 20-8 | — | — | 0 | — | 5 | 9.5 | 4 | 5.2 | 3 | 13.0 |
| 20-9 | — | — | 8 | 12.2 | 23 | 16.3 | 21 | 14.6 | 17 | 10.4 |
| 20-10 | — | — | 3 | 35.3 | 3 | 13.7 | 5 | 22.7 | 3 | 10.1 |
| Average | — | — | — | 18.4 | — | 16.5 | — | 13.4 | — | 11.7 |
| Total | — | — | 34 | — | 84 | — | 74 | — | 62 | — |
| 50-1 | — | — | 8 | 36.2 | 82 | 5.5 | 37 | 11.9 | 45 | 12.9 |
| 50-2 | — | — | 55 | 10.9 | 121 | 11.4 | 178 | 8.1 | 153 | 8.2 |
| 50-3 | — | — | 68 | 13.7 | 124 | 13.0 | 158 | 10.7 | 135 | 11.9 |
| 50-4 | — | — | 59 | 3.8 | 86 | 14.5 | 117 | 11.8 | 170 | 12.0 |
| 50-5 | — | — | 5 | 4.8 | 30 | 8.4 | 69 | 11.7 | 92 | 14.1 |
| 50-6 | — | — | 22 | 8.0 | 42 | 7.8 | 48 | 11.1 | 63 | 10.0 |
| 50-7 | — | — | 26 | 10.7 | 41 | 7.5 | 55 | 8.2 | 87 | 10.8 |
| 50-8 | — | — | 26 | 10.6 | 121 | 6.3 | 120 | 8.1 | 171 | 7.7 |
| 50-9 | — | — | 12 | 21.2 | 51 | 11.5 | 46 | 10.9 | 76 | 10.9 |
| 50-10 | — | — | 40 | 12.3 | 55 | 11.5 | 88 | 10.6 | 68 | 10.4 |
| Average | — | — | — | 13.2 | — | 9.7 | — | 10.3 | — | 10.9 |
| Total | — | — | 321 | — | 753 | — | 916 | — | 1060 | — |
| 100-1 | — | — | 5 | 4.6 | 10 | 7.1 | 26 | 10.6 | 27 | 10.3 |
| 100-2 | — | — | 2 | 15.3 | 6 | 9.0 | 5 | 11.5 | 3 | 1.7 |
| 100-3 | — | — | 18 | 15.4 | 29 | 9.7 | 42 | 6.4 | 56 | 9.6 |
| 100-4 | — | — | 23 | 11.2 | 22 | 11.9 | 44 | 14.6 | 31 | 10.5 |
| 100-5 | — | — | 3 | 8.4 | 38 | 10.4 | 41 | 9.3 | 52 | 9.7 |
| 100-6 | — | — | 13 | 6.9 | 36 | 10.6 | 50 | 8.1 | 63 | 9.3 |
| 100-7 | — | — | 10 | 8.1 | 23 | 6.6 | 31 | 10.9 | 38 | 9.7 |
| 100-8 | — | — | 8 | 10.3 | 23 | 7.8 | 19 | 6.5 | 26 | 9.4 |
| 100-9 | — | — | 2 | 10.2 | 13 | 16.1 | 12 | 19.2 | 44 | 9.2 |
| 100-10 | — | — | 7 | 8.6 | 40 | 10.4 | 59 | 12.4 | 56 | 8.2 |
| Average | — | — | — | 9.9 | — | 10.0 | — | 10.9 | — | 8.8 |
| Total | — | — | 91 | — | 240 | — | 329 | — | 396 | — |

Table 18: HOP-OO average path lengths on random Delaunay graphs

| | Eyerich *et al.* [15] | | $\ell$-CTP , # of agents | | | | |
|---|---|---|---|---|---|---|---|
| Graph | HOP | UCTO | 1 | 2 | 3 | 4 | 5 |
| 20-1 | 171.6 | 169.0 | 169.4 | **151.7** | 144.9 | 141.8 | 139.4 |
| 20-2 | 155.8 | 148.9 | 154.7 | **143.8** | 134.9 | 133.5 | 131.4 |
| 20-3 | 138.7 | 132.5 | 138.3 | **129.7** | 120.7 | 118.5 | 117.0 |
| 20-4 | 286.8 | 235.2 | 241.4 | **214.4** | 204.4 | 195.9 | 189.4 |
| 20-5 | 113.3 | 111.3 | 124.3 | **107.5** | 104.7 | 100.2 | 99.2 |
| 20-6 | 142.0 | 133.1 | **132.5** | 124.1 | 118.5 | 114.9 | 113.2 |
| 20-7 | 150.2 | 148.2 | **147.9** | 139.1 | 131.6 | 128.0 | 126.5 |
| 20-8 | 133.6 | 134.5 | 137.5 | **128.7** | 125.4 | 122.1 | 121.0 |
| 20-9 | 177.1 | 173.9 | 174.4 | **160.9** | 155.8 | 153.1 | 151.5 |
| 20-10 | 188.1 | 167.0 | 171.7 | **155.6** | 150.8 | 147.8 | 146.2 |
| Average | 165.7 | 154.2 | 159.2 | **145.5** | 139.2 | 135.6 | 133.5 |
| 50-1 | 250.6 | 186.1 | 230.4 | **171.9** | 154.3 | 148.8 | 139.6 |
| 50-2 | 375.4 | 365.5 | 370.9 | **335.0** | 324.0 | 316.9 | 312.1 |
| 50-3 | 294.5 | 255.6 | 274.5 | **236.0** | 216.7 | 209.4 | 204.9 |
| 50-4 | 263.9 | 230.5 | **228.5** | 204.7 | 191.4 | 187.2 | 184.0 |
| 50-5 | 239.5 | 225.4 | 226.7 | **200.7** | 188.3 | 184.4 | 181.6 |
| 50-6 | 253.2 | 236.3 | 238.5 | **211.4** | 204.1 | 198.2 | 194.1 |
| 50-7 | 221.9 | 206.3 | 207.2 | **176.7** | 165.7 | 160.4 | 157.8 |
| 50-8 | 302.2 | 277.6 | 305.3 | **262.6** | 244.8 | 238.5 | 233.9 |
| 50-9 | 281.8 | 222.5 | 243.9 | **210.0** | 197.0 | 190.4 | 186.3 |
| 50-10 | 271.2 | 240.8 | 261.0 | **212.6** | 196.8 | 190.3 | 185.7 |
| Average | 275.4 | 244.7 | 258.7 | **222.2** | 208.3 | 202.4 | 198.0 |
| 100-1 | 319.3 | 286.8 | **281.3** | 244.7 | 231.9 | 225.7 | 220.2 |
| 100-2 | 154.5 | 151.5 | 158.2 | **151.1** | 148.9 | 148.3 | 147.8 |
| 100-3 | 488.1 | 412.2 | 446.5 | **362.3** | 333.4 | 320.2 | 310.8 |
| 100-4 | 329.8 | 314.3 | 333.8 | **297.5** | 286.2 | 280.0 | 276.2 |
| 100-5 | 452.4 | 348.3 | 385.0 | **319.3** | 301.7 | 290.6 | 284.0 |
| 100-6 | 487.9 | 396.2 | 421.5 | **377.8** | 361.9 | 348.3 | 341.5 |
| 100-7 | 403.9 | 358.2 | 380.0 | **316.1** | 299.2 | 286.6 | 281.4 |
| 100-8 | 322.0 | 293.3 | 295.0 | **246.0** | 228.6 | 215.8 | 208.6 |
| 100-9 | 366.1 | 262.0 | 273.4 | **238.8** | 227.5 | 219.0 | 214.9 |
| 100-10 | 388.4 | 342.3 | 354.9 | **295.6** | 274.5 | 264.7 | 259.5 |
| Average | 371.3 | 316.5 | 333.0 | **284.9** | 269.4 | 259.9 | 254.5 |

Table 19: HOP-OO analysis of performance decrease across number of agents

| | Number of agents | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | | 2 | | 3 | | 4 | | 5 | |
| Graph | # | % | # | % | # | % | # | % | # | % |
| 20-1 | — | — | 1 | 27.6 | 1 | 38.4 | 12 | 11.9 | 17 | 10.1 |
| 20-2 | — | — | 20 | 15.0 | 15 | 11.8 | 20 | 14.6 | 4 | 25.5 |
| 20-3 | — | — | 0 | — | 9 | 23.2 | 2 | 17.2 | 4 | 18.4 |
| 20-4 | — | — | 3 | 35.5 | 3 | 11.2 | 2 | 1.5 | 16 | 11.8 |
| 20-5 | — | — | 4 | 41.3 | 3 | 20.3 | 2 | 37.6 | 2 | 13.9 |
| 20-6 | — | — | 1 | 10.4 | 7 | 15.6 | 8 | 12.1 | 3 | 2.5 |
| 20-7 | — | — | 2 | 13.6 | 4 | 8.4 | 1 | 2.2 | 1 | 1.6 |
| 20-8 | — | — | 3 | 3.2 | 3 | 10.4 | 2 | 6.9 | 6 | 5.4 |
| 20-9 | — | — | 0 | — | 6 | 25.5 | 0 | — | 3 | 5.7 |
| 20-10 | — | — | 9 | 13.8 | 0 | — | 2 | 23.3 | 3 | 13.2 |
| Average | — | — | — | 20.1 | — | 18.3 | — | 14.1 | — | 10.8 |
| Total | — | — | 43 | — | 51 | — | 51 | — | 59 | — |
| 50-1 | — | — | 5 | 6.9 | 12 | 16.2 | 11 | 12.9 | 23 | 10.2 |
| 50-2 | — | — | 34 | 8.2 | 45 | 9.0 | 44 | 9.5 | 34 | 6.4 |
| 50-3 | — | — | 9 | 9.0 | 11 | 10.9 | 11 | 7.8 | 17 | 11.7 |
| 50-4 | — | — | 17 | 17.8 | 56 | 15.8 | 18 | 19.2 | 34 | 6.9 |
| 50-5 | — | — | 18 | 24.9 | 12 | 22.7 | 16 | 7.0 | 12 | 6.9 |
| 50-6 | — | — | 17 | 6.1 | 6 | 8.6 | 11 | 6.2 | 19 | 14.8 |
| 50-7 | — | — | 9 | 11.6 | 25 | 11.1 | 14 | 8.6 | 13 | 11.4 |
| 50-8 | — | — | 19 | 10.7 | 33 | 9.6 | 28 | 8.5 | 25 | 8.2 |
| 50-9 | — | — | 10 | 7.7 | 14 | 8.4 | 19 | 11.9 | 31 | 11.4 |
| 50-10 | — | — | 15 | 14.4 | 32 | 16.5 | 20 | 11.1 | 37 | 13.0 |
| Average | — | — | — | 11.7 | — | 12.9 | — | 10.3 | — | 10.1 |
| Total | — | — | 153 | — | 246 | — | 192 | — | 245 | — |
| 100-1 | — | — | 9 | 10.1 | 25 | 17.0 | 23 | 8.5 | 28 | 7.0 |
| 100-2 | — | — | 4 | 5.3 | 3 | 3.6 | 6 | 6.7 | 4 | 3.7 |
| 100-3 | — | — | 42 | 10.7 | 69 | 14.3 | 72 | 9.3 | 70 | 9.4 |
| 100-4 | — | — | 7 | 26.8 | 46 | 5.9 | 46 | 7.3 | 40 | 6.2 |
| 100-5 | — | — | 31 | 12.7 | 42 | 8.3 | 60 | 9.5 | 58 | 10.7 |
| 100-6 | — | — | 30 | 5.0 | 59 | 9.8 | 65 | 8.5 | 47 | 7.2 |
| 100-7 | — | — | 20 | 10.0 | 40 | 9.2 | 54 | 7.4 | 47 | 9.1 |
| 100-8 | — | — | 9 | 20.8 | 31 | 12.7 | 26 | 14.4 | 35 | 10.7 |
| 100-9 | — | — | 9 | 3.5 | 21 | 11.4 | 31 | 8.8 | 25 | 9.1 |
| 100-10 | — | — | 19 | 8.9 | 53 | 12.2 | 55 | 9.1 | 85 | 10.1 |
| Average | — | — | — | 11.4 | — | 10.4 | — | 9.0 | — | 8.3 |
| Total | — | — | 180 | — | 389 | — | 438 | — | 439 | — |

Table 20: HOP-EC average path lengths on random Delaunay graphs

| | Eyerich *et al.* [15] | | $\ell$-CTP, # of agents | | | | |
|---|---|---|---|---|---|---|---|
| Graph | HOP | UCTO | 1 | 2 | 3 | 4 | 5 |
| 20-1 | 171.6 | 169.0 | 169.4 | **151.2** | 144.0 | 140.7 | 138.1 |
| 20-2 | 155.8 | 148.9 | 154.7 | **144.0** | 136.9 | 133.5 | 130.4 |
| 20-3 | 138.7 | 132.5 | 138.3 | **129.7** | 119.8 | 118.0 | 116.6 |
| 20-4 | 286.8 | 235.2 | 241.4 | **214.7** | 204.6 | 196.4 | 189.1 |
| 20-5 | 113.3 | 111.3 | 124.3 | **107.2** | 104.2 | 99.9 | 98.6 |
| 20-6 | 142.0 | 133.1 | **132.5** | 123.6 | 118.2 | 115.0 | 113.7 |
| 20-7 | 150.2 | 148.2 | **147.9** | 139.0 | 131.2 | 127.5 | 127.0 |
| 20-8 | 133.6 | 134.5 | 137.5 | **128.7** | 125.3 | 120.2 | 118.7 |
| 20-9 | 177.1 | 173.9 | 174.4 | **160.6** | 155.1 | 152.2 | 150.2 |
| 20-10 | 188.1 | 167.0 | 171.7 | **155.6** | 151.1 | 147.6 | 145.6 |
| Average | 165.7 | 154.2 | 159.2 | **145.4** | 139.0 | 135.1 | 132.8 |
| 50-1 | 250.6 | 186.1 | 230.5 | **171.9** | 154.6 | 148.9 | 138.9 |
| 50-2 | 375.4 | 365.5 | 370.9 | **333.1** | 320.9 | 313.1 | 308.9 |
| 50-3 | 294.5 | 255.6 | 274.5 | **234.6** | 216.8 | 209.8 | 203.8 |
| 50-4 | 263.9 | 230.5 | **228.5** | 202.2 | 187.9 | 182.1 | 178.2 |
| 50-5 | 239.5 | 225.4 | 226.7 | **200.9** | 187.8 | 183.3 | 180.4 |
| 50-6 | 253.2 | 236.3 | 238.6 | **212.3** | 202.8 | 197.3 | 193.4 |
| 50-7 | 221.9 | 206.3 | 207.2 | **176.6** | 164.3 | 159.2 | 156.7 |
| 50-8 | 302.2 | 277.6 | 305.4 | **259.4** | 243.2 | 236.7 | 231.2 |
| 50-9 | 281.8 | 222.5 | 244.0 | **210.2** | 196.7 | 190.7 | 186.2 |
| 50-10 | 271.2 | 240.8 | 261.0 | **212.9** | 198.3 | 189.2 | 183.0 |
| Average | 275.4 | 244.7 | 258.7 | 221.4 | 207.3 | 201.0 | 196.1 |
| 100-1 | 319.3 | 286.8 | **281.3** | 244.1 | 230.9 | 225.2 | 219.3 |
| 100-2 | 154.5 | 151.5 | 158.2 | **150.8** | 148.8 | 147.4 | 146.7 |
| 100-3 | 488.1 | 412.2 | 446.5 | **359.0** | 332.0 | 316.1 | 308.2 |
| 100-4 | 329.8 | 314.3 | 333.8 | **295.3** | 281.7 | 273.7 | 269.5 |
| 100-5 | 452.4 | 348.3 | 385.0 | **317.7** | 297.4 | 286.1 | 279.2 |
| 100-6 | 487.9 | 396.2 | 421.5 | **374.7** | 353.2 | 340.5 | 334.5 |
| 100-7 | 403.9 | 358.2 | 380.0 | **313.8** | 295.7 | 283.3 | 277.9 |
| 100-8 | 322.0 | 293.3 | 295.0 | **243.5** | 219.4 | 211.7 | 207.5 |
| 100-9 | 366.1 | 262.0 | 273.4 | **237.3** | 224.1 | 216.5 | 212.9 |
| 100-10 | 388.4 | 342.3 | 354.9 | **311.1** | 279.6 | 267.3 | 259.0 |
| Average | 371.3 | 316.5 | 333.0 | 241.4 | 228.8 | 222.1 | 218.4 |

Table 21: HOP-EC analysis of performance decrease across number of agents

| | Number of agents | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | | 2 | | 3 | | 4 | | 5 | |
| Graph | # | % | # | % | # | % | # | % | # | % |
| 20-1 | — | — | 1 | 27.6 | 14 | 10.1 | 9 | 13.2 | 8 | 6.4 |
| 20-2 | — | — | 19 | 11.9 | 22 | 11.8 | 23 | 18.4 | 16 | 12.9 |
| 20-3 | — | — | 0 | — | 4 | 19.7 | 2 | 11.7 | 4 | 8.8 |
| 20-4 | — | — | 2 | 1.7 | 13 | 13.6 | 8 | 10.5 | 14 | 11.3 |
| 20-5 | — | — | 4 | 41.3 | 3 | 18.7 | 4 | 22.5 | 0 | — |
| 20-6 | — | — | 25 | 9.0 | 14 | 6.5 | 13 | 13.4 | 14 | 9.5 |
| 20-7 | — | — | 3 | 10.6 | 13 | 3.3 | 4 | 16.0 | 22 | 12.3 |
| 20-8 | — | — | 2 | 1.2 | 6 | 5.8 | 4 | 4.2 | 9 | 6.6 |
| 20-9 | — | — | 3 | 11.3 | 15 | 13.9 | 18 | 12.5 | 5 | 3.1 |
| 20-10 | — | — | 16 | 10.3 | 3 | 14.5 | 1 | 28.9 | 3 | 16.1 |
| Average | — | — | — | 13.9 | — | 11.8 | — | 15.1 | — | 9.7 |
| Total | — | — | 75 | — | 107 | — | 86 | — | 95 | — |
| 50-1 | — | — | 9 | 8.8 | 24 | 12.8 | 20 | 7.1 | 10 | 4.8 |
| 50-2 | — | — | 29 | 7.2 | 48 | 7.8 | 73 | 4.8 | 45 | 6.0 |
| 50-3 | — | — | 11 | 7.3 | 46 | 8.5 | 53 | 8.9 | 61 | 6.4 |
| 50-4 | — | — | 10 | 12.7 | 56 | 8.8 | 28 | 6.4 | 40 | 7.5 |
| 50-5 | — | — | 21 | 16.6 | 38 | 10.9 | 25 | 8.0 | 30 | 9.0 |
| 50-6 | — | — | 10 | 7.4 | 18 | 6.6 | 50 | 7.7 | 27 | 9.7 |
| 50-7 | — | — | 13 | 11.1 | 18 | 8.8 | 28 | 8.6 | 20 | 8.2 |
| 50-8 | — | — | 38 | 5.8 | 69 | 6.7 | 56 | 7.0 | 51 | 6.1 |
| 50-9 | — | — | 10 | 15.0 | 23 | 8.0 | 60 | 10.3 | 34 | 8.7 |
| 50-10 | — | — | 8 | 16.3 | 76 | 14.4 | 45 | 8.8 | 81 | 8.6 |
| Average | — | — | — | 10.8 | — | 9.3 | — | 7.8 | — | 7.5 |
| Total | — | — | 159 | — | 416 | — | 438 | — | 399 | — |
| 100-1 | — | — | 7 | 11.3 | 44 | 10.4 | 46 | 9.2 | 46 | 7.6 |
| 100-2 | — | — | 5 | 6.8 | 7 | 4.4 | 11 | 4.2 | 8 | 3.3 |
| 100-3 | — | — | 0 | — | 0 | — | 0 | — | 0 | — |
| 100-4 | — | — | 21 | 5.1 | 80 | 5.6 | 61 | 6.4 | 93 | 5.7 |
| 100-5 | — | — | 22 | 9.8 | 107 | 8.9 | 93 | 8.2 | 121 | 5.8 |
| 100-6 | — | — | 26 | 9.7 | 56 | 9.7 | 77 | 5.5 | 81 | 6.0 |
| 100-7 | — | — | 22 | 5.0 | 65 | 13.7 | 75 | 6.7 | 92 | 5.7 |
| 100-8 | — | — | 3 | 12.0 | 35 | 11.9 | 34 | 12.1 | 77 | 8.5 |
| 100-9 | — | — | 18 | 3.2 | 31 | 8.9 | 30 | 5.5 | 37 | 7.5 |
| 100-10 | — | — | 1 | 2.4 | 17 | 4.1 | 0 | — | 7 | 2.7 |
| Average | — | — | — | 7.3 | — | 8.6 | — | 7.2 | — | 5.9 |
| Total | — | — | 125 | — | 442 | — | 427 | — | 562 | — |

Table 22: HOP-EF average path lengths on random Delaunay graphs

| | Eyerich *et al.* [15] | | $\ell$-CTP, # of agents | | | | |
|---|---|---|---|---|---|---|---|
| Graph | HOP | UCTO | 1 | 2 | 3 | 4 | 5 |
| 20-1 | 171.6 | 169.0 | 169.4 | **150.8** | 144.0 | 141.2 | 138.7 |
| 20-2 | 155.8 | 148.9 | 154.7 | **143.3** | 134.7 | 132.4 | 130.2 |
| 20-3 | 138.7 | 132.5 | 138.3 | **129.6** | 119.7 | 117.7 | 116.3 |
| 20-4 | 286.8 | 235.2 | 241.4 | **214.3** | 204.1 | 195.5 | 188.5 |
| 20-5 | 113.3 | 111.3 | 124.3 | **107.2** | 104.2 | 100.0 | 98.8 |
| 20-6 | 142.0 | 133.1 | **132.5** | 124.1 | 118.4 | 114.8 | 113.4 |
| 20-7 | 150.2 | 148.2 | **147.9** | 138.4 | 131.3 | 127.2 | 126.5 |
| 20-8 | 133.6 | 134.5 | 137.5 | **128.7** | 125.1 | 120.9 | 119.6 |
| 20-9 | 177.1 | 173.9 | 174.4 | **161.2** | 154.6 | 152.1 | 150.1 |
| 20-10 | 188.1 | 167.0 | 171.7 | **155.5** | 150.7 | 147.5 | 145.6 |
| Average | 165.7 | 154.2 | 159.2 | 145.3 | 138.7 | 134.9 | 132.8 |
| 50-1 | 250.6 | 186.1 | 230.5 | **171.9** | 154.7 | 149.8 | 138.6 |
| 50-2 | 375.4 | 365.5 | 370.9 | **334.7** | 322.3 | 314.6 | 310.0 |
| 50-3 | 294.5 | 255.6 | 274.5 | **233.4** | 215.6 | 208.3 | 204.0 |
| 50-4 | 263.9 | 230.5 | **228.5** | 202.6 | 187.8 | 182.7 | 179.0 |
| 50-5 | 239.5 | 225.4 | 226.7 | **200.5** | 188.3 | 183.3 | 180.3 |
| 50-6 | 253.2 | 236.3 | 238.6 | **211.8** | 202.7 | 197.6 | 193.8 |
| 50-7 | 221.9 | 206.3 | 207.2 | **174.5** | 163.4 | 158.8 | 156.2 |
| 50-8 | 302.2 | 277.6 | 305.4 | **259.0** | 243.8 | 236.2 | 231.7 |
| 50-9 | 281.8 | 222.5 | 244.0 | **207.5** | 195.1 | 187.7 | 184.3 |
| 50-10 | 271.2 | 240.8 | 261.0 | **212.0** | 198.3 | 190.1 | 183.3 |
| Average | 275.4 | 244.7 | 258.7 | 220.8 | 207.2 | 200.9 | 196.1 |
| 100-1 | 319.3 | 286.8 | **281.3** | 242.8 | 229.3 | 223.8 | 217.2 |
| 100-2 | 154.5 | 151.5 | 158.2 | **150.8** | 148.7 | 147.8 | 147.2 |
| 100-3 | 488.1 | 412.2 | 446.5 | **359.0** | 330.7 | 317.6 | 308.3 |
| 100-4 | 329.8 | 314.3 | 333.8 | **295.8** | 282.1 | 274.2 | 270.8 |
| 100-5 | 452.4 | 348.3 | 385.0 | **317.7** | 297.2 | 285.7 | 279.1 |
| 100-6 | 487.9 | 396.2 | 421.5 | **374.0** | 353.9 | 340.5 | 334.7 |
| 100-7 | 403.9 | 358.2 | 380.0 | **315.0** | 296.6 | 283.8 | 278.4 |
| 100-8 | 322.0 | 293.3 | 295.0 | **232.5** | 217.3 | 209.6 | 205.3 |
| 100-9 | 366.1 | 262.0 | 273.4 | **238.1** | 224.2 | 216.4 | 211.0 |
| 100-10 | 388.4 | 342.3 | 354.9 | **293.2** | 271.7 | 260.9 | 254.0 |
| Average | 371.3 | 316.5 | 333.0 | 281.9* | 265.2 | 256.0* | 250.6 |

Table 23: HOP-EF analysis of performance decrease across number of agents

| | Number of agents | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | | 2 | | 3 | | 4 | | 5 | |
| Graph | # | % | # | % | # | % | # | % | # | % |
| 20-1 | — | — | 27 | 7.3 | 3 | 8.0 | 27 | 15.7 | 14 | 9.8 |
| 20-2 | — | — | 20 | 11.3 | 31 | 7.0 | 25 | 12.1 | 21 | 24.4 |
| 20-3 | — | — | 0 | — | 5 | 16.2 | 3 | 12.1 | 1 | 10.6 |
| 20-4 | — | — | 5 | 8.9 | 17 | 11.3 | 10 | 5.3 | 19 | 10.8 |
| 20-5 | — | — | 6 | 28.0 | 4 | 18.4 | 3 | 26.2 | 0 | — |
| 20-6 | — | — | 25 | 10.1 | 21 | 9.0 | 16 | 7.2 | 8 | 7.7 |
| 20-7 | — | — | 3 | 10.6 | 7 | 5.7 | 10 | 12.8 | 4 | 8.3 |
| 20-8 | — | — | 7 | 2.9 | 4 | 8.4 | 5 | 4.5 | 10 | 6.0 |
| 20-9 | — | — | 12 | 36.5 | 11 | 15.0 | 15 | 11.8 | 18 | 9.3 |
| 20-10 | — | — | 14 | 11.0 | 3 | 24.9 | 6 | 12.8 | 6 | 13.2 |
| Average | — | — | — | 14.1 | — | 12.4 | — | 12.1 | — | 11.1 |
| Total | — | — | 119 | — | 106 | — | 120 | — | 101 | — |
| 50-1 | — | — | 9 | 8.8 | 27 | 12.6 | 14 | 7.2 | 20 | 10.0 |
| 50-2 | — | — | 52 | 6.6 | 69 | 6.3 | 76 | 5.8 | 77 | 5.1 |
| 50-3 | — | — | 40 | 5.9 | 55 | 8.9 | 60 | 6.9 | 102 | 7.9 |
| 50-4 | — | — | 26 | 11.2 | 45 | 6.1 | 48 | 6.5 | 73 | 6.1 |
| 50-5 | — | — | 65 | 10.8 | 49 | 9.5 | 40 | 7.8 | 29 | 7.5 |
| 50-6 | — | — | 17 | 8.2 | 24 | 5.3 | 51 | 6.5 | 47 | 6.8 |
| 50-7 | — | — | 21 | 7.7 | 44 | 5.0 | 97 | 6.0 | 41 | 6.0 |
| 50-8 | — | — | 27 | 9.0 | 55 | 9.4 | 72 | 8.4 | 110 | 6.2 |
| 50-9 | — | — | 16 | 10.1 | 24 | 5.8 | 31 | 8.8 | 47 | 7.5 |
| 50-10 | — | — | 17 | 9.2 | 52 | 12.5 | 85 | 11.8 | 90 | 10.4 |
| Average | — | — | — | 8.8 | — | 8.1 | — | 7.6 | — | 7.4 |
| Total | — | — | 290 | — | 444 | — | 574 | — | 636 | — |
| 100-1 | — | — | 20 | 4.9 | 56 | 9.9 | 60 | 6.9 | 73 | 5.3 |
| 100-2 | — | — | 7 | 5.1 | 10 | 3.6 | 26 | 4.4 | 21 | 4.6 |
| 100-3 | — | — | 45 | 9.8 | 86 | 7.8 | 123 | 11.3 | 144 | 8.2 |
| 100-4 | — | — | 36 | 6.0 | 168 | 5.2 | 121 | 5.8 | 158 | 5.1 |
| 100-5 | — | — | 49 | 8.9 | 100 | 7.0 | 131 | 6.8 | 160 | 6.4 |
| 100-6 | — | — | 35 | 5.5 | 78 | 7.4 | 126 | 5.5 | 102 | 7.0 |
| 100-7 | — | — | 43 | 10.2 | 72 | 8.9 | 96 | 7.5 | 127 | 7.0 |
| 100-8 | — | — | 15 | 5.9 | 34 | 9.3 | 53 | 7.4 | 59 | 8.6 |
| 100-9 | — | — | 32 | 3.8 | 34 | 7.1 | 46 | 6.3 | 59 | 8.2 |
| 100-10 | — | — | 72 | 7.4 | 115 | 7.4 | 188 | 7.5 | 185 | 7.1 |
| Average | — | — | — | 6.8 | — | 7.4 | — | 6.9 | — | 6.7 |
| Total | — | — | 354 | — | 753 | — | 970 | — | 1088 | — |