Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

REDUCING RISKS DURING NATURAL DISASTERS WITH OPTIMAL RESOURCE ALLOCATION BY MULTI-AGENT OPTIMIZATION

Alina Vereshchaka
Nathan Margaglio
Wen Dong

Department of Computer Science and Engineering
State University of New York at Buffalo
Buffalo, 14260, USA

ABSTRACT

Natural disasters are notable for the high costs associated with responding to and recovering from them. In this paper we address the issue of critical resources allocation during natural disaster, that incorporates the level of importance of the effected region and cost parameter. Our risk reducing model can be applied to online stochastic environments in the domain of natural disasters. The framework achieves more efficient resource allocation in response to dynamic events and is applicable to problems where disaster evolves alongside the response efforts.

1 INTRODUCTION

Natural disasters are notable for the high costs associated with responding to and recovering from them. The National Centers for Environmental Information (NCEI) that tracks and evaluates climate events reported that the U.S. has sustained 250 weather and climate disasters since 1980, with damages exceeding $1 billion. The total cost of these 250 events exceeds $1.7 trillion. The distribution of damage from disaster events is dominated by tropical cyclone losses ($934.6 billion), followed by drought ($248.4 billion), severe storms($238.8 billion) and others. These types of events are also responsible for the highest number of human-losses. The variety of damages highlights the need for management and control to eliminate the loss from these events.

International, governmental and local organizations are putting efforts into examining the types of natural hazards, their costs in human lives and economic impact, and national and international responses to them. Technological advances provide various tools and methods in predicting, managing, surviving, and mitigating effects after natural disasters. In this paper, we address the issue of critical resource allocation during natural disasters, which incorporates the level of importance of the addressed region and cost parameters. Our risk reducing model can be applied to online stochastic environments in the domain of natural disasters. This paper is a continuation of our previous work (Vereshchaka and Dong 2019), with one of the main improvements being the introduction of efficient allocation, along with the application of softmax function for more intelligent allocation.

The main contribution of this paper is to propose a hierarchical multi-agent framework for solving critical resource allocation problems during natural disasters. The framework consists of two levels. On the lower level there is a set of multi-agents that navigates within the continuous time environment by using reinforcement algorithms. On the higher level, there is a lead agent that is responsible for decision making. Our framework can be applicable to problems that incorporate nature and human life, e.g. wildfires, disease epidemics, and snowstorms, when the disaster evolves on the same timescale as the response effort, where delays in response can lead to increased disaster severity and thus greater demand for resources.

The rest of the paper is organized as follows. In Section 2, we discuss the related works and background in the area of disaster management. In Section 3, we present and analyze our model of incorporating autonomous agents on resource allocation problems. We describe the experimental framework used to evaluate the performance of the proposed algorithm in Section 4. Our conclusion and discussion of future research is presented in Section 5.

2 RELATED WORK AND BACKGROUND

In this section, we will give an overview of related works and a brief introduction to the reinforcement learning and Markov decision processes.

2.1 Modeling Social Behaviour During Natural Disasters

Recent advances in simulation platforms development aim to model disaster preparedness and response (Hwang et al. 2016; Aziz et al. 2018), human mobility during hurricanes (Wang and Taylor 2016), and evacuation of pedestrians during emergency events (Haris et al. 2018). Social behaviors have also been widely utilized for disaster management planning. Recent studies show the benefit of extracting social media posts about injuries and help requests during disasters, such as floods, epidemic outbreaks (Frias-Martinez et al. 2011; Wesolowski et al. 2012), and post-earthquake conditions (Bengtsson et al. 2011), and utilizing these posts to create a pool of reports for modeling a disaster relief response (Abbasi et al. 2012). (Petrovic et al. 2012) introduced a framework that dynamically computes optimal strategies for decision making during wildfires. Furthermore, researchers have demonstrated that disaster management platforms can capture and manage the flow of information between rescue groups and the public. (Estuar et al. 2017)

2.2 Autonomous Multi-agents for Optimal Resource Allocation

Multiple applications of autonomous multi-agent frameworks interacting with other agents for achieving a common goal are getting more attention (Albrecht and Stone 2018). There are recent advances in modeling multi-agent complex systems within the framework of the Markov decision process and reinforcement learning, including discrete-event decision process (Yang et al. 2019), multi-players games (Peng et al. 2017), multi-robot control (Matignon et al. 2012). Deep reinforcement learning has been used to solve a decentralized resource allocation mechanism for vehicle-to-vehicle (V2V) communications (Ye et al. 2018). Policy gradient methods have been widely applied for on-line environments, for example in solving micro-tolling assignment problems (Mirzaei et al. 2018). For example, Mustapha et al. (2013) incorporates multi-agent based modeling for ambulance allocation after the disaster event.

2.3 Markov Decision Process

The task of allocation optimization is framed as an infinite-horizon Markov decision process (MDP) formulation. This consists of an agent interacting with an environment through actions given states. The dynamics of our environment are defined by its state space S, action space A, state transition probability function $P : S \times A \times S \to \mathbb{R}$, initial state distribution $\rho_0 : S \to [0,1]$, and reward function $R : S \times A \to \mathbb{R}$. Since our process is infinite in horizon, we define $\gamma \in [0,1]$ as our discount factor used to terminate cumulative reward over a finite time span. The tuple $< S, A, P, r, \rho_0, \gamma >$ effectively describes our MDP.

In our work, we extend this framework to a partially observable Markov game in order to facilitate multiple agents acting within the same environment simultaneously. We accomplish this by describing the specific portion of the state visible to each agent i as observation O_i. Each agent i produces an action from an action set A_i which may be unique to them. The dynamics of each agent i, then, can be described by the policy $\pi_{i,\theta_i} : O_i \to A_i$, which is parameterized by θ_i which corresponds to the trainable weights in a neural network in a deep learning setting.

The training loop begins at time step t with the environment having produced a state $s_t \in S$. Each agent i is given their observation $o_{i,t} \in O_i$ with which it produces an action $a_{i_t} \in A_i$. This sequence of actions is then transformed to a single action $a_t \in A$ and returned to the environment. The environment, then, produces a new state $s_{t+1} \in S$ along with a reward $r_t \in \mathbb{R}$ according to the state transition probability function P and reward function R, respectively. Solving the MDP means finding a policy π^* that maximizes the future discounted reward $G = \sum_{t=0}^{T} \gamma^t r_t$, where T is the time horizon.

2.4 Policy Gradient Algorithms

Policy gradient methods provide some of the most effective techniques in reinforcement learning for accomplishing tasks involving complex, real-world control problems with continuous, high-dimensional, and partially-observable properties, such as robotic control systems (Peters and Schaal 2006). Policy gradient methods have attractive properties for reinforcement learning applications; specifically, these methods are guaranteed to converge to a local and optimal solution under mild assumptions about the step size used to update policy parameters (Moré and Thuente 1994).

Policy gradient works by optimizing the parameters θ of the policy π directly, which differs from value-based methods which indirectly derive a policy from a value function (such as with Q-Learning). The parameters are adjusted in order to maximize the objective $J(\theta) = \mathbb{E}_{s\ \rho_\pi, a\ \pi_\theta}[G]$ by updating the parameters θ with the gradient $\nabla_\theta J(\theta)$: $\hat{g} = \hat{\mathbb{E}}_t \left[\nabla_\theta \log \pi_\theta(a_t|s_t) \hat{A}_t \right]$, where $\hat{A}_t$ is an estimator of the advantage function at time step t with the expectation $\hat{\mathbb{E}}_t[...]$ indicating the empirical average over a finite batch of samples. The gradient estimation $\hat{g}$ is obtained by differentiating the objective $L^{PG}(\theta) = \hat{\mathbb{E}}_t \left[\log \pi_\theta(a_t|s_t) \hat{A}_t \right]$.

3 METHODOLOGY

In this section, we describe and formalize a generalized model for dynamic resource allocation. We first report on how we formulate the social optimization problem of critical resource allocation as a reinforcement learning framework, and then we detail our algorithm for solving the problem.

3.1 Purpose

The model is designed to make a dynamic resource allocation during natural disaster scenarios for the next time step. At each time step we utilize external factors that evolve over time, such as weather conditions, importance of region and associated cost. One of the benefits of our optimization allocation is that it helps to minimize the redundancy of resource allocation among the regions.

3.2 Assumptions

We assume a discrete action space, where a sub-agent is limited to a number of actions it can perform at each time step. On the low level, our agents have the identical goal of optimizing a given objective function, particularly a reward function that is a common objective in reinforcement learning settings.

Although we are using the "Markovian" property as a base for our algorithm, which chooses its actions based only on the most recent observation, we are incorporating a changing behaviour to the lead agent. It is able to adapt its decision making also based on the level of trust for each sub-agent. Thus, while limiting the level of complexity in the framework, tracking the rewards value increases the degree of adaptability that allows one to capture historical behaviour.

3.3 Framework Overview

A two-level framework involves a set of autonomous agents controlled by the lead agent that makes a decision on critical resource allocation as shown in Figure 1 (Vereshchaka and Dong 2019). To initialize the model, we define a region that is affected by the disaster and further this region is divided into sub-regions

to narrow down the area for more precise management. The number of agents is correlated with the total number of sub-regions.

On the lower level there is a set of multi-agents $(A_0, A_1, ..., A_n)$ that corresponds to the number of sub-regions. Each agent predicts the volume of hazards (V_t^n) and the importance of each s(U_t^n), taking as input an observation from the time-series environment. Each agent is treated as an autonomous agent in a multi-agent framework with partial observability.

On the higher level there is a lead agent, which is trained to make resource allocation decisions for assigning volume distribution of the involved resources. To make this decision, the lead agent takes the volume of hazards (V_t^n) as input from each of the lower level sub-agents, also lead agent takes cost parameters for allocating the resources and importance of each region at every time-step. The sum of distributed resources fractions $\{l_0, l_1, ..., n\}$ must be equal to 1. If the volume for ith resource is L_i, then $\sum_{i=1}^{N} L_t^i = 1$.

In designing a framework by which the decision making is controlled by the output of each sub-agent, we aim to satisfy the following list of properties:

- Autonomy. Each agent controls one sub-region. If the entire mechanism were centrally controlled, it would be more susceptible to single-point failure, requiring massive amounts of computational power. In our case, we can utilize the benefits of distributing and parralelizing the computation, that allows us to perform optimization tasks in real-time, taking into consideration the dynamic of hazards.
- Low Communication Complexity. By keeping the number of messages and the amount of information transmitted to a minimum, the system can afford to put more communication reliability measures in place. Furthermore, each resource, as an autonomous agent, may have privacy concerns which should be respected. Keeping the communication complexity low will also make the system more scalable, thus it can easily expand to control more sub-regions.
- Scalability. Sub-regions can be united into sub-groups, based on the similarity of the data-origin, which will be leaded by a sub-lead agent. Thus, the architecture can be scaled in the fashion of tree-node.

3.4 Multi-agent System: Lower Level

A bundle of agents at the lower level performs predictions about the time-series environment. The processes are done independently for each sub-agent, thus making the framework applicable for real-time scenarios. We are using a modified version of a proximal policy optimization algorithm (PPO) (Schulman et al. 2017). Observations are unique for each agent and the parameters are not shared between them.

Algorithm 1: PPO for multiple agents

Input : initial policy parameter θ_0, clipped threshold ε

Output: projected hazard volume V_t^i, projected importance level U_t^i

for *k = 0, 1, 2, ...* **do**

Collect set of partial trajectories D_k on policy $\pi_k = \pi(\theta_k)$;

Estimate advantage A_t using any advantage estimation algorithm;

Compute policy update by taking K steps of minibatch SGD,

$$\theta_{k+1} = \arg\max_{\theta} L^{CLIP}(\theta), \text{ where}$$

$$L^{CLIP}(\theta) = \mathbb{E}_t[\min(r_t(\theta)A_t, clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)A_t].$$

end

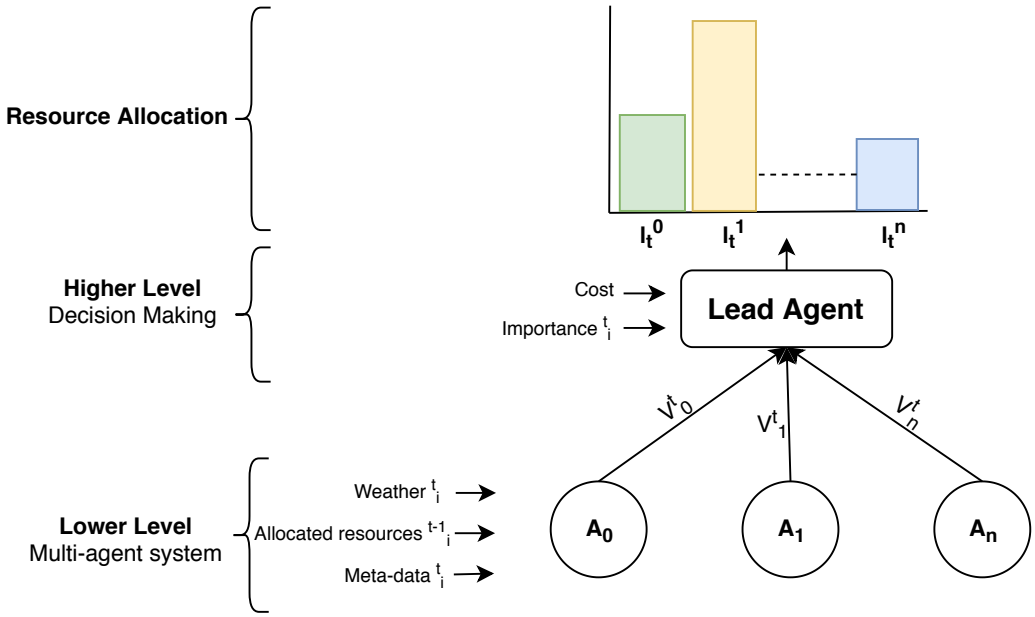


Figure 1: An overview of our framework. Lower level: view of module with a bundle of independent agents. Note that the input data and parameters are unique for each agent. Higher level: a lead agent, that makes an allocation decision, based on the data, broadcasted by each of the subagents, cost and importance of the region. As a result, the resources are disctributed among the regions.

3.5 Resource Allocation: Higher Level

On the higher level of our architecture, a lead agent takes predictions made by each of the sub-agents at the lower level, including cost and importance of each region. Based on these data, the lead agent makes a decision about the optimal resource allocation at every time-step.

Now we will define the notations. Sub-agents $A_1, A_2, A_3, ..., A_n$, where n is the number of sub-agents, equivalent to the total number of sub-regions. For each time step t, agent i broadcasts (V_t^i, U_t^i) to the lead agent, where

- V_t^i is a projected volume of work that needs to be done at sub-region^i for the next time step, e.g. the volume of snow that has to be removed
- U_t^i is a level of importance of sub-region^i, e.g. the projected traffic.

The lead agent gets as an observation W_t^i, a real volume of work that has been completed during the previous time step.

3.5.1 Importance Parameter and Error Function

The level of importance for each of the sub-regions is defined as a normalized product of the projected volume of work and the projected volume of traffic (1), and the cumulative error to be minimized by the worker agents is the discounted relative error between the actual amount of work and the predicted amount of work (2).

$$\text{Imp}^i_{t,norm} = \frac{\text{Imp}^i_t}{\sum_{i=1}^{N} \text{Imp}^i_t}, \text{ where } \text{Imp}^i_t = \max[V^i_t, \varepsilon] \cdot \max[U^i_t, \varepsilon] \tag{1}$$

$$\text{Error}^i_t = \frac{W^i_t - V^i_{t-1}}{\max[V^i_{t-1}, \varepsilon]} + \gamma \text{Error}^i_{t-1}, \text{ where } \gamma \text{ is a discounting factor.} \tag{2}$$

3.5.2 Resource Allocation Considering Importance

To make allocation decisions based on the importance parameter, we use equation (3), that returns L^i_t - resource allocation at time t per sub-region i.

$$L^i_t = \left\lfloor M \cdot X^i_t \right\rfloor, \text{ where } X^i_t = \frac{\text{Imp}^i_{t,norm}(1 + \text{Error}^i_t)}{\sum_{i=1}^{N} \text{Imp}^i_{t,norm}(1 + \text{Error}^i_t)} \tag{3}$$

3.5.3 Resource Allocation Considering Cost

To make allocation decisions based on the effectiveness of utilizing the resources, we introduce parameter C, where C is the volume of removed hazard that one resource can mitigate during one time-step. The total minimum number of resources required to remove the hazard during one time-step, including the error in prediction, is in equation (4).

$$\widetilde{L}^i_t = \left\lceil \frac{1}{C} V^i_t \cdot (1 + \text{Error}^i_{t-1}) \right\rceil \tag{4}$$

Algorithm 2: Efficient resource allocation among regions per time t

Input : volume of resource to be allocated M, projected volume of hazard V^i_t, projected level of importance U^i_t, error ε, discounting factor γ

Output: amount of allocated resources per each sub-region Y^i_t

for *i = 0, 1, 2, ...* **do**

Estimate level of importance for each sub-region using equation (1);

Normalize the importance parameter among all sub-regions;

Calculate the cumulative error for each agent using equation (2);

Calculate resource allocation using importance parameter L^i_t equation 1;

Calculate resource allocation using cost parameter $\widetilde{L}^i_t$ equation (4);

Minimizing the difference between the allocated resources at the previous time-step with the requested volume with importance and with cost $\min\{|L^i_t - \widetilde{Y}^i_{t-1}|, |\widetilde{L}^i_t - \widetilde{Y}^i_{t-1}|\}$:

$$Y^i_t = \begin{cases} \widetilde{L}^i_t & \text{if } |L^i_t - \widetilde{Y}^i_{t-1}| > |\widetilde{L}^i_t - \widetilde{Y}^i_{t-1}| \\ L^i_t & \text{otherwise} \end{cases}$$

Checking for normalizing, thus if $\sum_{j=1}^{n} Y^i_t \geq M$, then $Y^i_t = M \cdot \frac{Y^i_t}{\sum_{j=1}^{n} Y^j_t}$.

end

4 CASE STUDY

In this section, we present our experiments, which should be viewed as complementary to our theoretical work in the previous section. For our experiment we applied our framework to the heavy snowfall period in the Western New York area. We collected daily snow observations from the Global Historical Climatology Network (GHCN) stations for New York, USA, for the period of January-February 2019 (National Centers for Environmental Information 2019). During the period there was a heavy snowfall with high winds, which resulted in massive accumulations of snow across the area. Emergency managers deployed 1,602 large plow trucks with a total of 3,900 operators (Cuomo 2019).

As an importance parameter, we utilized the daily average volume of traffic, using traffic count data provided by the Greater Buffalo-Niagara Regional Transportation Council (Buffalo 2019) that contains more than 32,000 entries.

The goal of the case study is to demonstrate how our algorithm is capable of reducing risks while considering importance-dynamics during harsh weather and natural disasters through a form of optimal resource allocation. We formulate our multi-agent environment such that there are four agents working cooperatively to remove snow from their respective areas in the Buffalo-Niagara region. Our algorithm, which works as an allocation agent, performs resource allocation by distributing resources to each agent. This helps to reduce risks by providing sufficient coverage in areas where a combination of high levels of snow fall and greater risks of accidents and casualties due to higher volumes of traffic and denser populations demand more resources.

4.1 Formulation

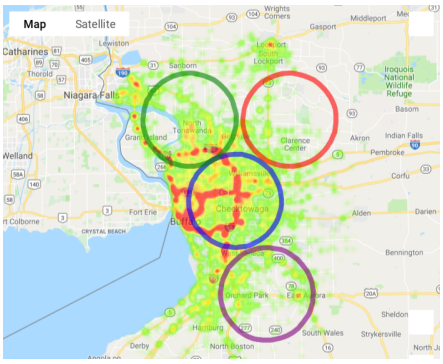
To formulate the task, we define the optimal allocation at any given time step to be one which minimizes our excess cost. In this scenario, we aim to coordinate available resources, such as plows, personnel, equipment, and others in order to effectively utilize their time. This comes in the form of required resource prediction error and moving costs.

The resource prediction error is formulated as the difference of the amount of resources an area actually requires at a given time step versus the amount of resources present at that time step. At a high level, this cost corresponds to the "wasted time" of the unnecessary resources. For example, if an area requires n plows at a time step, but we have allocated $n+1$, then we would have cleared more snow over the entire area had we moved 1 plow to an area that was under-manned.

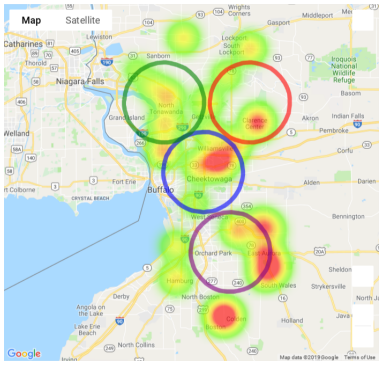
We weigh the costs of erroneous predictions by the importance of the regions, which is defined as the relative average daily traffic per region (Table 1). In our experiment, the Downtown Buffalo region (Area 0) receives the most relative traffic with 9,309 cars, then North Tonawanda (Area 1) with 4,907 cars, then Orchard Park (Area 2) with 4,178 cars, then finally Clarence (Area 3) with 2,989 cars (see Figure 2). These values are normalized to be between 1 and 2, and the resulting value are multiplied to the error costs. This represents each area's importance, and encourages our agent to provide more resources to more important areas.

The moving cost represents the time spent moving resources from one area to another. Resources are unproductive (i.e., not clearing snow) as they are transferred from one area to another. As we coordinate resources to move based on our required resource predictions, we have to consider the downtime associated with such re-allocations, and asses the costs of moving versus the cost of being under allocated. In practice, we would associate some real-world value with cost to represent a quantity which we want to minimize the loss of, such as capital or time.

For our experiment, we consider the snow fall for four areas in the Buffalo-Niagara region, NY, USA over the course of January and February in 2019 (Figure 3). We split up the areas by finding equally sized radii accounting for roughly the same amount of accumulated snow over the course of our data as well as similar traffic volume (see Figure 2). When computing the optimal allocation, which represents the ideal allocation our agent could have taken to minimize costs, we weigh the allocation for each region by its



(a) Four areas with a traffic volume heat map.



(b) Four areas with a daily snow accumulation heat map.

Figure 2: The map represents New York, USA regions. Regions are Downtown Buffalo (blue), North Tonawanda (green), Orchard Park (purple), and Clarence (red).

Table 1: Overview of the data per each area.

Area	Coordinates	Region	Average Daily Snow Fall	Average Daily Traffic
0	(42.92, -78.77)	Downtown Buffalo	2.80 inches	9,309 cars
1	(43.05, -78.87)	North Tonawanda	2.00 inches	4,907 cars
2	(42.77, -78.70)	Orchard Park	3.80 inches	4,178 cars
3	(43.05, -78.65)	Clarence	2.84 inches	2,989 cars

normalize average daily traffic value. This represents the areas importance, and will require our allocation agent to provide more resources to those areas with high importance than others.

Since the goal of our algorithm is to minimize cost, we formulate a cost function comprised of a linear combination of the error costs and the movement costs. Thus, our agent's policy is parameterized by two coefficients, so that our ability to optimize is made relatively easy. In a more complex setting, our agent's policy may become computationally infeasible to optimize, in which case we would use a more efficient function approximation, such as a neural network, which can be optimized via gradient descent.

During each time step, the allocation agent receives the predicted snow fall from each of the sub-agents found in the four regions. The allocation agent then determines, based on both the importance of the region and the expected error of the sub-agent, where to allocate the resources. Figure 4 shows the comparison between the allocation agent's determined allocation versus the optimal allocation which should have been taken to minimize error at a random time step.

We can represent the difference of our predicted allocation and the optimal allocation over each time step through line plots, such as in Figure 7. The plots provide the total cost found for each area per time step (Area Costs) as well as the sum of the error costs for each area and the sum of the movement costs for each area per time step (Aggregate Area Costs).

4.2 Baselines

A series of naive management policies are applied to the agent in order to asses our approach. First, we run through the environment using an optimal policy. Here, we provide the sub-agents with perfect snow fall prediction (Figure 5), so the allocation agent can form allocations the represent the least amount of cost possible. The resulting total cost here is 0.11.

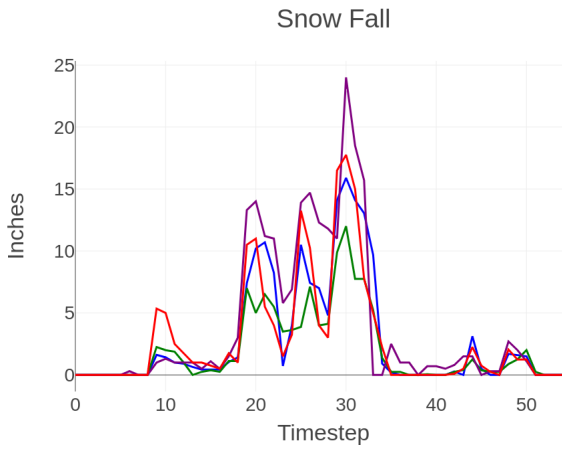


Figure 3: Snow accumulation for Jan/Feb 2019 in our four areas.

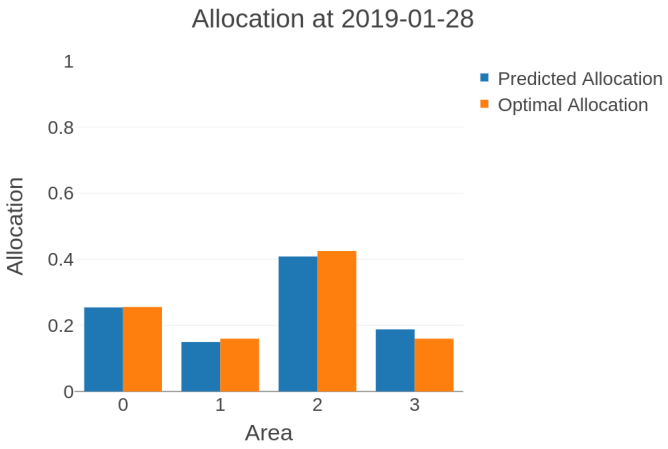


Figure 4: Sample resource allocation for January 28, 2019.

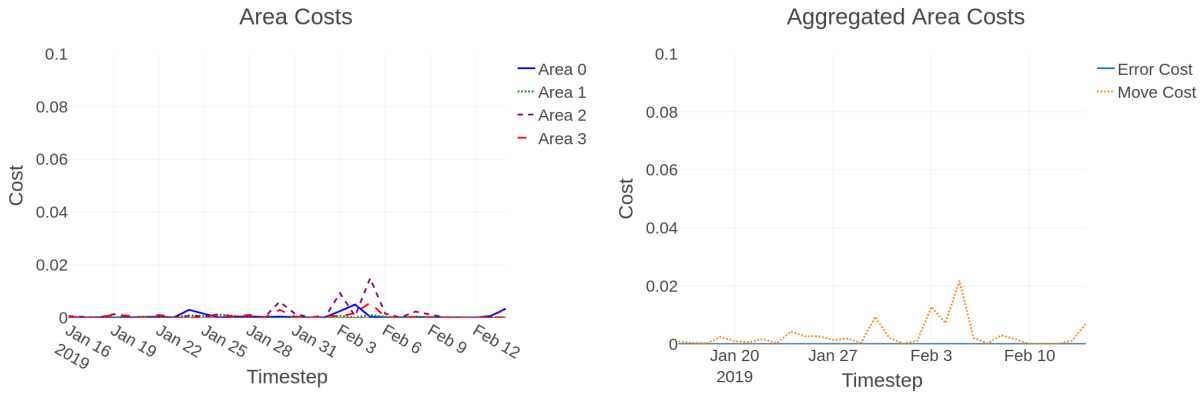


Figure 5: Costs for each area (left) and movement and error costs (right) for perfect baseline.

Next, we apply a strategy where we allocate a random amount per region to minimize prediction error (Figure 6). In this case, we sample a value from the normal distribution for each area and normalize to get our allocation. Here, the total accumulated cost is 1.41.

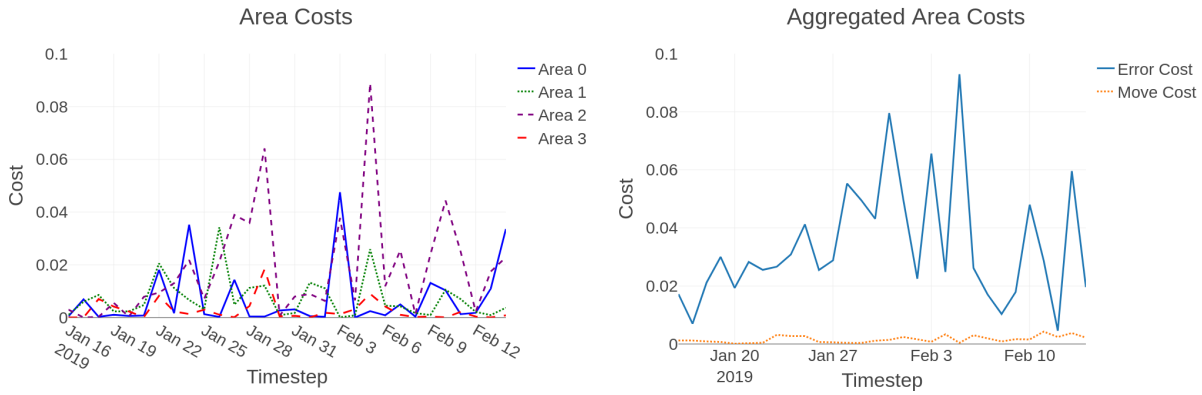


Figure 6: Costs for each area (left) and movement and error costs (right) for random baseline.

Finally, we run our environment on a trivial policy where every area is allotted the same amount of resources for every time step (Figure 7). This uniform allocation policy essentially allows us to avoid movement costs completely while offering relatively safe coverage. Here, the resulting total cost is 0.96.

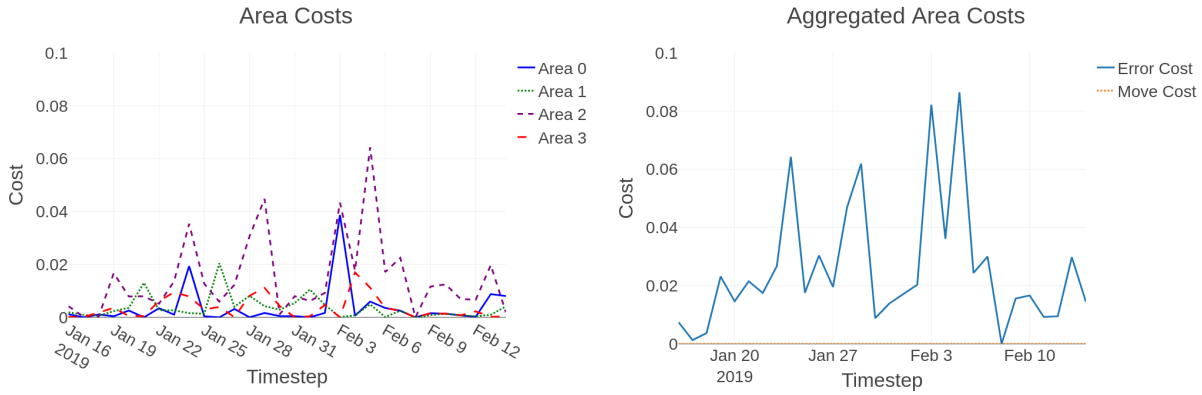


Figure 7: Costs for each area (left) and movement and error costs (right) for uniform baseline.

4.3 Evaluation

With our baselines described, we now evaluate our implementation. Our goal is to avoid error while limiting movement, and the allocation agent can facilitate this though adjusting the agents parameters. For this experiment, we will provide a grid search where we vary the move coefficient and error coefficient in a 9x9 grid of evenly spaced points between 0.1 and 0.9.

We can see that optimal points exist around (0.9, 0.2) as well as (0.9, 0.6) (Figure 8). When using the parameters of 0.9 and 0.2 for our agent, the average cost over 10 episodes is 0.75. This means that our agent can perform better than in the cases of uniform and random polices (Figure 9).

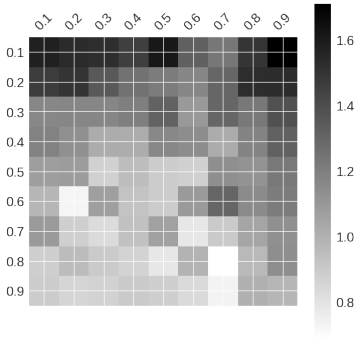


Figure 8: The cost matrix from our grid search.

5 CONCLUSIONS

In this paper we presented an algorithm that achieve optimal resource allocation during natural disasters. Our algorithm considers the level of importance of the affected region, cost of moving critical resources from one region to another and level of confidence in multi-agent prediction of the volume of hazards. Our framework can be generalized in that it can be applied to other events involving coupled dynamics of a disaster, limited resources and response at each time step. It can be implemented as a risk management tool for both planning and operational support during natural disaster response.

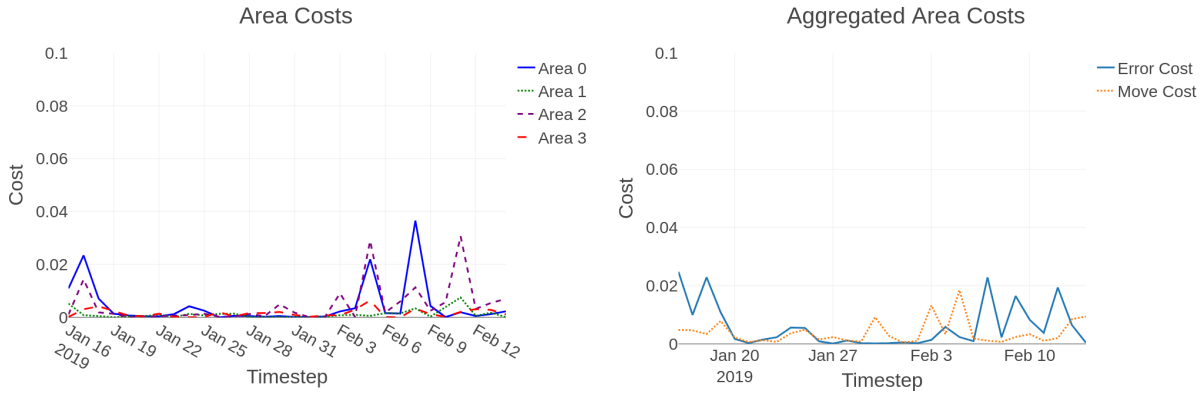


Figure 9: Costs for each area (left) and movement and error costs (right) for our algorithm.

A direction of improvement for the current framework would be to utilize a model based approach for training agents on the lower level, thus, while adding a new agent, it will allow the reuse of past experiences of the similar agents. This can improve the learning speed for new tasks. Furthermore, for the purpose of this paper we have a restriction on the deterministic action-space, this can be extended to stochastic action-space to facilitate more robust prediction. Thus, further research is required in this area.

REFERENCES

Abbasi, M.-A., S. Kumar, J. A. Andrade Filho, and H. Liu. 2012. "Lessons Learned in Using Social Media for Disaster Relief-ASU Crisis Response Game". In *Proceedings of the International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction 2012*. April 2[d]-3[d], Kentland, Maryland, 282–289.

Albrecht, S. V., and P. Stone. 2018. "Autonomous Agents Modelling Other Agents: A Comprehensive Survey and Open Problems". *Artificial Intelligence* 258:66–95.

Aziz, A., J. Mapar, and K. Atri. 2018. "Use of Modeling and Simulation in Emergency Preparedness and Response: Standard Unified Modeling, Mapping, Integration Toolkit". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, 2725–2736. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Bengtsson, L., X. Lu, A. Thorson, R. Garfield, and J. Von Schreeb. 2011. "Improved Response to Disasters and Outbreaks by Tracking Population Movements with Mobile Phone Network Data: a Post-Earthquake Geospatial Study in Haiti". *PLoS medicine* 8(8):e1001083.

Open Data Buffalo 2019. "Annual Average Daily Traffic Volume Counts". https://data.buffalony.gov/Transportation/Annual-Average-Daily-Traffic-Volume-Counts/y93c-u65y. accessed April 12[th] 2019.

Cuomo, A. M. 2019. "Rush Transcript: Governor Cuomo Holds Storm Briefing Call". https://www.governor.ny.gov/news/rush-transcript-governor-cuomo-holds-storm-briefing-call. accessed April 12[th] 2019.

Estuar, M. R. J. E., R. C. Rodrigueza, J. N. C. Victorino, M. C. V. Sevilla, M. M. De Leon, and J. C. S. Rosales. 2017. "Agent-based Modeling Approach in Understanding Behavior during Disasters: Measuring Response and Rescue in Ebayanihan Disaster Management Platform". In *Proceedings of the International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation 2017*. July 5[th]-8[th], Washington, DC, 46–52.

Frias-Martinez, E., G. Williamson, and V. Frias-Martinez. 2011. "An Agent-based Model of Epidemic Spread Using Human Mobility and Social Network Information". In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, 57–64. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Haris, M., I. Mahmood, M. Badar, and M. S. Q. Alvi. 2018. "Modeling Safest and Optimal Emergency Evacuation Plan for Large-Scale Pedestrians Environments". In *2018 Winter Simulation Conference*, edited by M. Rabe, A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, 917–928. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Hwang, S., R. Starbuck, S. Lee, M. Choi, S. Lee, and M. Park. 2016. "High Level Architecture (HLA) Compliant Distributed Simulation Platform for Disaster Preparedness and Response in Facility Management". In *Proceedings of the 2016 Winter*

Simulation Conference, edited by T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, 3365–3374. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Matignon, L., L. Jeanpierre, and A.-I. Mouaddib. 2012. "Coordinated Multi-robot Exploration under Communication Constraints Using Decentralized Markov Decision Processes". In *Proceedings of the 26th AAAI Conference on Artificial Intelligence 2012*. July 22nd-26th, Toronto, Canada, 2017–2023.

Mirzaei, H., G. Sharon, S. Boyles, T. Givargis, and P. Stone. 2018. "Enhanced Delta-tolling: Traffic Optimization via Policy Gradient Reinforcement Learning". In *2018 21st International Conference on Intelligent Transportation Systems*, 47–52. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Moré, J. J., and D. J. Thuente. 1994. "Line Search Algorithms with Guaranteed Sufficient Decrease". *ACM Transactions on Mathematical Software* 20(3):286–307.

Mustapha, K., H. Mcheick, and S. Mellouli. 2013. "Modeling and Simulation Agent-Based of Natural Disaster Complex Systems". *Procedia Computer Science* 21:148–155.

National Centers for Environmental Information 2019. "Daily U.S. Snowfall and Snow Depth". https://www.ncdc.noaa.gov/snow-and-ice/daily-snow/NY/snow-depth/20190131. accessed April 12th 2019.

Peng, P., Q. Yuan, Y. Wen, Y. Yang, Z. Tang, H. Long, and J. Wang. 2017. "Multiagent Bidirectionally-Coordinated Nets for Learning to Play Starcraft Combat Games". *arXiv preprint arXiv:1703.10069*.

Peters, J., and S. Schaal. 2006. "Policy Gradient Methods for Robotics". In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2219–2225. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Petrovic, N., D. L. Alderson, and J. M. Carlson. 2012. "Dynamic Resource Allocation in Disaster Response: Tradeoffs in Wildfire Suppression". *PloS one* 7(4):e33285.

Schulman, J., F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. 2017. "Proximal Policy Optimization Algorithms". *arXiv preprint arXiv:1707.06347*.

Vereshchaka, A., and W. Dong. 2019. "Dynamic Resource Allocation During Natural Disasters Using Multi-agent Environment". In *Proceedings of the International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation 2019*. July 9th-12th, Washington, DC, 123–132.

Wang, Q., and J. E. Taylor. 2016. "Data-driven Simulation of Urban Human Mobility Constrained by Natural Disasters". In *2016 Winter Simulation Conference*, edited by T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, 3357–3364. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Wesolowski, A., N. Eagle, A. J. Tatem, D. L. Smith, A. M. Noor, R. W. Snow, and C. O. Buckee. 2012. "Quantifying the Impact of Human Mobility on Malaria". *Science* 338(6104):267–270.

Yang, F., B. Liu, and W. Dong. 2019. "Optimal Control of Complex Systems through Variational Inference with a Discrete Event Decision Process". In *Proceedings of the 2019 International Conference on Autonomous Agents & Multiagent Systems*, 296–304. International Foundation for Autonomous Agents and Multiagent Systems.

Ye, H., G. Y. Li, and B.-H. F. Juang. 2018. "Deep Reinforcement Learning based Resource Allocation for V2V Communications". *arXiv preprint arXiv:1805.07222*.

AUTHOR BIOGRAPHIES

ALINA VERESHCHAKA is a PhD student in the Department of Computer Science and Engineering at the State University of New York at Buffalo, USA. Her current research interests include deep reinforcement learning, optimization and multi-agent modeling in stochastic environments. She has conducted studies in the application areas of optimization and transportation. She developed and leading a course for graduate students in Reinforcement Learning at the State University of New Your at Buffalo. Her is email address is avereshc@buffalo.edu. Her website is https://cse.buffalo.edu/∼avereshc/.

NATHAN MARGAGLIO is a graduate student in the Department of Computer Science and Engineering at the State University of New York at Buffalo, USA. His research interests are in the deep reinforcement learning. His email address is namargag@buffalo.edu.

WEN DONG is an Assistant Professor of Computer Science and Engineering with a joint appointment at the Institute of Sustainable Transportation and Logistics at the State University of New York at Buffalo. His research focuses on developing machine learning and signal processing tools to study the dynamics of large social systems. He has a PhD degree from the M.I.T. Media Laboratory. His email address is wendong@buffalo.edu. His website is https://cse.buffalo.edu/ wendong/.