Ambulance Dispatch via Deep Reinforcement Learning

Kunpeng Liu
University of Central Florida
Orlando, Florida, United States
kunpengliu@knights.ucf.edu

Xiaolin Li
Nanjing University
Nanjing, Jiangsu, China
lixl@nju.edu.cn

Cliff C. Zou
University of Central Florida
Orlando, Florida, United States
czou@cs.ucf.edu

Haibo Huang
University of Central Florida
Orlando, Florida, United States
haibo.huang@knights.ucf.edu

Yanjie Fu*
University of Central Florida
Orlando, Florida, United States
yanjie.fu@ucf.edu

ABSTRACT

In this paper, we solve the ambulance dispatch problem with a reinforcement learning oriented strategy. The ambulance dispatch problem is defined as deciding which ambulance to pick up which patient. Traditional studies on ambulance dispatch mainly focus on predefined protocols and are verified on simple simulation data, which are not flexible enough when facing the dynamically changing real-world cases. In this paper, we propose an efficient ambulance dispatch method based on the reinforcement learning framework, i.e., Multi-Agent Q-Network with Experience Replay(MAQR). Specifically, we firstly reformulate the ambulance dispatch problem with a multi-agent reinforcement learning framework, and then design the state, action, and reward function correspondingly for the framework. Thirdly, we design a simulator that controls ambulance status, generates patient requests and interacts with ambulances. Finally, we design extensive experiments to demonstrate the superiority of the proposed method.

CCS CONCEPTS

• **Computing methodologies** → **Multi-agent reinforcement learning**; • **Information systems** → **Spatial-temporal systems**.

KEYWORDS

reinforcement learning, ambulance dispatch, experience replay

ACM Reference Format:
Kunpeng Liu, Xiaolin Li, Cliff C. Zou, Haibo Huang, and Yanjie Fu. 2020. Ambulance Dispatch via Deep Reinforcement Learning. In *28th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '20), November 3–6, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3397536.3422204

1 INTRODUCTION

The dispatch of ambulance, aiming at reducing patients' waiting time, is always a hot topic and attracts much attention in operation research, computer science and urban computing. Existing work on improving ambulance dispatch efficiency can be categorized into three methods, i.e., nearest dispatching [18, 23], center re-building [2, 4, 7] and smooth routing planning [6, 8, 15]. However, these methods either require large budget or are too fixed to meet the dynamic changing of the real-time environment.

In this paper, we adapt the multi-agent reinforcement learning framework, where each ambulance is considered as one agent and they cooperate and to reduce the overall waiting time of patients. We propose to partition the whole map by rectangles into grids, and agent action is defined as the grid it goes to. We regard ambulances belonging to the same ambulance center as homogeneous. These homogeneous ambulances share the same policy, meaning we only need to design one policy for each ambulance center instead of each ambulance.

To better represent the environment, we incorporate the distribution of available ambulance, ambulance request and waiting time to the state representation. To guide and inspire the training of reinforcement learning, we incorporate the request waiting time and the time cost of the delivery into reward function. We design a simulator to provide the interactive environment. In this simulator, we can control ambulance status, generate requests and interact with reinforcement learning agents. The simulator is controlled by probabilistic models, and we calibrate its parameters with the real-world data.

To summarize, in this paper, we propose to solve the ambulance dispatch problem with a multi-agent reinforcement learning framework. Specifically, our contributions are as follows: (1) We formulate the ambulance dispatch problem into a reinforcement learning framework. (2) We propose a Multi-Agent Q-Network with Experience Replay (MAQR) method to solve the reformulated problem. (3) We design a probabilistic model based simulator to interact and evaluate the proposed reinforcement learning methods. (4) We conduct extensive experiments to demonstrate the enhanced performances of our proposed method.

2 PROPOSED METHOD

2.1 Definitions

In this paper, we study the problem of ambulance allocation, with the goal of improving the ambulance dispatch efficiency through optimized allocation strategy so as to reduce patients' waiting time. To better formulate the problem, we partition the whole city into N_g grids by rectangles and the whole day into time windows by 5 minutes [12]. In each time window, ambulance requests arise from

*Contact Author.

SIGSPATIAL '20, November 3–6, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-8019-5/20/11.
https://doi.org/10.1145/3397536.3422204

grids and answered by each ambulance center. The goal is to design a strategy to decide how many ambulances should the centers send to each grid in each time window, so that the overall waiting time of requests would be minimized. We propose to use reinforcement learning framework to tackle this problem. We define the agent, state, action and reward as follows.

Agent. We consider the multi-agent case, where an available ambulance is regarded as one agent and ambulances belonging to the same center are considered as homogeneous agents that share the same allocation strategy.

State. The global state shared by all the agents $\mathbf{s}_t$ at time t, is defined based on the spatio-temporal distribution of the ambulances and requests. The state consists of three components, i.e., available ambulance distribution, ambulance request distribution, and waiting time distribution.

Action. For each agent, the action is defined by which grid it can go to. Since there are N_g grids, the action space contains N_g actions. We define action $a_t^i = \{k|k = 1, 2, ..., N_g\}$ and $a_t^i = k$ means the agent i would go to the k-th grid at time t.

Reward. we define the reward function of agent i which goes to the k-th grid at time t by three factors, i.e., the request number in the k-th grid $m_{k,t}$, the sum of waiting time over all requests in the k-th grid $w_{k,t}$, and the time cost to make a round trip, T_k^i. The reward function is defined as:

$$r_t^i = h_1 * m_{k,t} + h_2 * w_{k,t} + h_3 * T_k^i \tag{1}$$

where h_1, h_2, h_3 are weights for the factors.

2.2 DQN for Ambulance Dispatch

In the multi-agent reinforcement learning scenario, the state transitions and reward of an individual agent are affected by the joint actions with all other agents instead of merely by its own actions. Accordingly, the action value function of one agent should consider interactions with other agents. Since the computation burden of multi-agent reinforcement learning is heavy, in this paper, we adopt the shared environment (represented by the global state $\mathbf{s}_t$) as the only factor involving interaction between agents [20]. To be more explicit, every agent tries to maximize its own discounted accumulated reward $E[\sum_{k=0}^{\infty} \gamma^k r_{t+k}^i]$, where γ is the discount factor and r_{t+k}^i is the reward for agent i at time $t+k$. r_{t+k}^i is highly related with the environment, and the action of each agent a_t^i changes the environment. By this way, interactions between agents can be reflected on the action reward function by sharing the same global state $\mathbf{s}_t$.

We adopt deep Q-Network (DQN) to learn the action reward function. For agent i, the Q-Network parameters are learned by optimizing the Bellman Equation:

$$Q(\mathbf{s}_t, a_t^i, \theta^i) = r_{t+1}^i + \gamma \, max_{a_{t+1}^i} Q(\mathbf{s}_{t+1}, a_{t+1}^i, \theta^{i-}) \tag{2}$$

where γ denotes the discount factor, θ^i denotes parameters of Q-Network, and θ^{i-} denotes parameters of target network. θ^{i-} synchronizes with θ^i every C steps and keeps fixed in other update intervals [13].

Intuitively, we need to maintain N_a Q-Network for N_a agents. Since the agents belonging to the same center are consider as homogeneous, these agents can share the same Q-Network:

$$Q(\mathbf{s}_t, a_t^i, \theta^i) = Q(\mathbf{s}_t, a_t^j, \theta^j) \tag{3}$$

where agent i is affiliated to center j. Thus we reduce the number of independent Q-Networks from N_a to N_c.

However, in some cases, there are relatively many ambulance centers, making N_c big and the computation burden still heavy. Therefore, we furthermore integrate the N_c Q-networks into one Q-Network, and agents are distinguished by their IDs [25]. Then We have the new Bellman Equation,

$$Q(\mathbf{s}_t, a_t^j, \theta) = r_{t+1}^i + \gamma \, max_{a_{t+1}^j} Q(\mathbf{s}_{t+1}, a_{t+1}^j, \theta^-) \tag{4}$$

2.3 Simulator Design

It is internally required by reinforcement learning to have an interactive environment in the process of training and testing. The most straightforward way is to deploy the algorithm in the real world and do training and testing thereafter. However, in most cases it is not realistic to implement the algorithm directly on real-world platform due to the cost on budget and risk on safety. One alternative way is to build a simulator which models the real-world environment [10, 22]. Basically, we design a simulator to realize the following three functions:

1) Ambulance status control. Considering on the current state of motion, the status of one ambulance would be updated as 'online', 'on-service', 'off-service' at each time window. And also, when the status is 'online' and the ambulance is available, it could switch to 'off-line' with a probability learned from the real data. Similarly, 'off-line' ambulance can also switch to 'on-line' with probability.
2) Request generation. We have two ways to generate 'real' requests. One way is to compress all the real requests (from 361 days) into one day and then get sample 'real ' sample by binomial distribution with probability $p = 1/361$, and it is the default way in the experiment unless specified. The other way is to suppose the request generation as a Poisson process, derive its parameter by maximum likelihood function (MLE), and generate requests using this model.
3) Interaction with agents. With allocation policies, the agents would take optimized actions. After each action, the state of the environment (i.e., ambulance distribution, request distribution and waiting time distribution) would change. The simulator keep the state updated at each time window and calculate the reward for each action.

We use the real-world data to calibrate the simulator. We evaluate the effectiveness of the simulator by comparing the difference of request numbers over time.

3 EXPERIMENTAL RESULTS

We provide an empirical evaluation on the performance of the proposed method with the simulator calibrated by real-world data. Table 1 shows the statistics of the dataset.

Table 1: Overview of the Dataset

Time Range	Records	Ambulances		
		No.	Center No.	Status No.
06/05/2016-05/31/2017	18,355,733	139	19	20

3.1 Baseline algorithms and Evaluation Metrics

We compare the proposed method with the following baseline algorithms: random allocation (**RA**) which randomly chooses one available ambulance to answer the request, location-based allocation (**LBA**) which assigns the ambulances to their nearest request, time-based allocation (**TBA**) which assigns the ambulances to requests with longer waiting time, request-based allocation (**RBA**) which assigns ambulances to the grids with more requests, and multi-agent deep Q-Network (**MAQ**), which is the raw DQN algorithms without the experience replay technology.

We evaluate the algorithms' performance by two metrics, i.e., Normalized overall waiting time (**NOW Time**) and normalized request answer rate (**NRAR**). In NOW Time, we summarize the waiting time of all of the unanswered requests and then do normalization. In NRAR, we divide the number of answered requests by the number of overall requests, and then do normalization.

3.2 Overall performance comparison

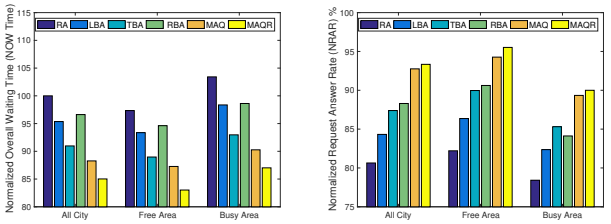
We firstly compare MAQR with baselines on the entire city over a time period of 1 day. As Table 2 shows, comparing with baseline algorithms, MAQR can improve the allocation efficiency to a certain extent. Among baselines, the Time-Based Allocation (TBA) has the best performance on NOW Time. This is because TBA is time-oriented thus it is the most direct allocation strategy towards waiting time. Similarly, RBA has the best performance on Normalized Request Answer Rate (NRAR) due to its focus on the grid's unanswered request number. The Multi-Agent Deep Q-Network (MAQ) has significant advantage over aforementioned baseline algorithms because it takes more factors into consideration thus can better optimize the allocation. With the assistance of experience replay, the performance of MAQR is boosted comparing with the raw MAQ.

Table 2: Overall performance comparison

	NOW Time	NRAR
RA	100.00 ± 3.41	80.64% ± 1.29%
LBA	95.36 ± 2.79	84.32% ± 1.37%
TBA	90.97 ± 2.38	87.39% ± 1.65%
RBA	96.62 ± 2.71	88.30% ± 2.17%
MAQ	88.27 ± 1.59	92.76% ± 1.54%
MAQR	85.02 ± 1.45	93.34% ± 1.43%

3.3 Robustness Check

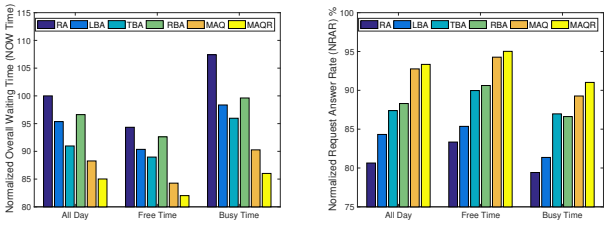
We apply the learned MAQR policy to different grids and time windows, to test the robustness of our method with variances in spatial and temporal views.



(a) NOW Time (b) NRAR

Figure 1: Spatial Robustness Check

As Figure 1 shows, besides the entire city, we choose two representatives of free area and busy area. The performance in free area is better than in the entire city, while performance in busy area is relatively worse. In free area, the demand of ambulance is low, and thus patients' waiting time is short and the request answer rate is high. Additionally, the traffic in free area is usually uncrowded which also improves the delivery efficiency. For busy area, the conditions are opposite thus the performance is worse. However, we emphasize that in all cases, our proposed algorithm MAQR still outperforms other baseline algorithms, and it has relatively good performance even in busy area.



(a) NOW Time (b) NRAR

Figure 2: Temporal Robustness Check

As Figure 2 shows, besides the entire day, we choose two representatives of free time and busy time. The performance at free time is better than the entire day, while performance at busy time is relatively worse. At free time, the demand of ambulance is low, and thus patients' waiting time is short and the request answer rate is high. Additionally, the traffic at busy time is usually uncrowded which also improves the delivery efficiency. For busy time, the conditions are opposite thus the performance is worse. However, we emphasize that in all cases, our proposed algorithm MAQR still outperforms other baseline algorithms, and it has relatively good performance even at busy time.

4 RELATED WORK

Related work can be grouped into two categories: work studies ambulance dispatch problem and work studies application of reinforcement learning.

4.1 Ambulance Dispatch Management

In the literature, a lot of methods have been proposed to deal with ambulance allocation issues. Most of existing work focused on the

routing optimization [3, 14] and design predefined strategies to improve the management efficiency [6, 15, 17]. For example, *Hayes et al.* studied the complexity of ambulance command and control by observing and interviewing fourteen ambulance dispatchers and then formulated decision strategies utilized by the dispatchers when working in the communication centers [8]. *Alessandrini et al.* evaluated the exposure–response relationship of ambulance dispatch data in association with biometeorological conditions using time series techniques [1]. *Barneveld et al.* studied models for relocation actions for idle ambulances that incorporate different performance measures related to the response times [21].

4.2 Reinforcement Learning

In multi-agent problems, the major challenge is to coordinate the action choices of different agents. *Tampuu et al.* extended the Deep Q-Learning framework to multi-agent environments to investigate the interaction between two learning agents [20]. *Stankovic et al.* proposed new algorithms for multi-agent distributed iterative value function approximation where the agents are allowed to have different behavior policies while evaluating the response to a single target policy. [19] *Liao et al.* proposed Multi-objective Optimization by Reinforcement Learning (MORL) to solve the optimal power system dispatch and voltage stability problem, which is undertaken on individual dimension in a high-dimensional space via a path selected by an estimated path value which represents the potential of finding a better solution [9] . *Liu et al.* reformulated the feature selection problem into a multi-agent reinforcement learning framework where the selection of each feature is controlled by its corresponding feature agent [5, 11]. *Yang et al.* developed deep reinforcement learning algorithms which could handle large scale agents with effective communication protocol [16, 24]. *Lin et al.* proposed to tackle the large-scale fleet management problem using reinforcement learning, and proposed a contextual multi-agent reinforcement learning framework which successfully tackled the taxi fleet management problem [10].

5 CONCLUSION REMARKS

In this paper, we study the problem of ambulance allocation. We present a multi-agent reinforcement learning framework to obtain an optimized allocation policy with the goal of minimizing the patients' waiting time. We consider each ambulance as an independent agent and their actions are stimulated by the well-designed reward function. We design a simulator to model the request generation, ambulance status and delivery process. With this simulator, we deploy and evaluate our multi-agent deep Q-Network. The experimental results show the proposed framework can effectively improve the ambulance delivery efficiency.

ACKNOWLEDGEMENTS

This research was partially supported by the National Science Foundation (NSF) via the grant numbers: 1755946, I2040950, 2006889.

REFERENCES

[1] Ester Alessandrini, Stefano Zauli Sajani, Fabiana Scotto, Rossella Miglio, Stefano Marchesi, and Paolo Lauriola. 2011. Emergency ambulance dispatches and apparent temperature: a time series analysis in Emilia–Romagna, Italy. *Environmental research* 111, 8 (2011), 1192–1200.

[2] Luce Brotcorne, Gilbert Laporte, and Frederic Semet. 2003. Ambulance location and relocation models. *European journal of operational research* 147, 3 (2003), 451–463.

[3] Timothy A Carnes, Shane G Henderson, David B Shmoys, Mahvareh Ahghari, and Russell D MacDonald. 2013. Mathematical programming guides air-ambulance routing at ornge. *Interfaces* 43, 3 (2013), 232–239.

[4] Richard Church and Charles ReVelle. 1974. The maximal covering location problem. In *Papers of the Regional Science Association*, Vol. 32. Springer-Verlag, 101–118.

[5] Wei Fan, Kunpeng Liu, Hao Liu, Pengyang Wang, Yong Ge, and Yanjie Fu. 2020. AutoFS: Automated Feature Selection via Diversity-aware Interactive Reinforcement Learning. *arXiv preprint arXiv:2008.12001* (2020).

[6] Michel Gendreau, Gilbert Laporte, and Frédéric Semet. 2001. A dynamic model and parallel tabu search heuristic for real-time ambulance relocation. *Parallel computing* 27, 12 (2001), 1641–1653.

[7] Jeffrey B Goldberg. 2004. Operations research models for the deployment of emergency services vehicles. *EMS management Journal* 1, 1 (2004), 20–39.

[8] Jared Hayes, Antoni Moore, George Benwell, and BL William Wong. 2004. Ambulance dispatch complexity and dispatcher decision strategies: Implications for interface design. In *Asia-Pacific Conference on Computer Human Interaction*. Springer, 589–593.

[9] HL Liao, QH Wu, and L Jiang. 2010. Multi-objective optimization by reinforcement learning for power system dispatch and voltage stability. In *Innovative Smart Grid Technologies Conference Europe (ISGT Europe), 2010 IEEE PES*. IEEE, 1–8.

[10] Kaixiang Lin, Renyu Zhao, Zhe Xu, and Jiayu Zhou. 2018. Efficient Large-Scale Fleet Management via Multi-Agent Deep Reinforcement Learning. *arXiv preprint arXiv:1802.06444* (2018).

[11] Kunpeng Liu, Yanjie Fu, Pengfei Wang, Le Wu, Rui Bo, and Xiaolin Li. 2019. Automating Feature Subspace Exploration via Multi-Agent Reinforcement Learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 207–215.

[12] Kunpeng Liu, Pengyang Wang, Jiawei Zhang, Yanjie Fu, and Sajal K Das. 2018. Modeling the Interaction Coupling of Multi-View Spatiotemporal Contexts for Destination Prediction. In *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 171–179.

[13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.

[14] Noraimi Azlin Mohd Nordin, Zati Aqmar Zaharudin, Mohd Azdi Maasar, and Nor Amalina Nordin. 2012. Finding shortest path of the ambulance routing: Interface of A algorithm using C# programming. In *Humanities, Science and Engineering Research (SHUSER), 2012 IEEE Symposium on*. IEEE, 1569–1573.

[15] Imtiyaz Pasha. 2006. *Ambulance management system using GIS*. Universitetsbibliotek.

[16] Peng Peng, Ying Wen, Yaodong Yang, Quan Yuan, Zhenkun Tang, Haitao Long, and Jun Wang. 2017. Multiagent Bidirectionally-Coordinated Nets: Emergence of Human-level Coordination in Learning to Play StarCraft Combat Games. *arXiv preprint arXiv:1703.10069* (2017).

[17] Martina Petralli, Marco Morabito, Lorenzo Cecchi, Alfonso Crisci, and Simone Orlandini. 2012. Urban morbidity in summer: ambulance dispatch data, periodicity and weather. *Central European Journal of Medicine* 7, 6 (2012), 775–782.

[18] John F Repede and John J Bernardo. 1994. Developing and validating a decision support system for locating emergency medical vehicles in Louisville, Kentucky. *European journal of operational research* 75, 3 (1994), 567–581.

[19] Milos Stankovic. 2016. Multi-agent reinforcement learning. In *Neural Networks and Applications (NEUREL), 2016 13th Symposium on*. IEEE, 1–1.

[20] Ardi Tampuu, Tambet Matiisen, Dorian Kodelja, Ilya Kuzovkin, Kristjan Korjus, Juhan Aru, Jaan Aru, and Raul Vicente. 2017. Multiagent cooperation and competition with deep reinforcement learning. *PloS one* 12, 4 (2017), e0172395.

[21] TC Van Barneveld, S Bhulai, and RD Van der Mei. 2017. A dynamic ambulance management model for rural areas. *Health care management science* 20, 2 (2017), 165–186.

[22] Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. 2018. IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2496–2505.

[23] Andres Weintraub, Julio Aboud, C Fernandez, Gilbert Laporte, and E Ramirez. 1999. An emergency vehicle dispatching system for an electric utility in Chile. *Journal of the Operational Research Society* 50, 7 (1999), 690–696.

[24] Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. 2018. Mean Field Multi-Agent Reinforcement Learning. *arXiv preprint arXiv:1802.05438* (2018).

[25] Lianmin Zheng, Jiacheng Yang, Han Cai, Weinan Zhang, Jun Wang, and Yong Yu. 2017. MAgent: A Many-Agent Reinforcement Learning Platform for Artificial Collective Intelligence. *arXiv preprint arXiv:1712.00600* (2017).