ADAPTIVE AMBULANCE REDEPLOYMENT VIA MULTI-ARMED BANDITS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

By
Ümitcan Şahin
September 2019

Adaptive Ambulance Redeployment via Multi-armed Bandits
By Ümitcan Şahin
September 2019

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Cem Tekin(Advisor)

Aykut Koç

Elif Vural

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

ADAPTIVE AMBULANCE REDEPLOYMENT VIA MULTI-ARMED BANDITS

Ümitcan Şahin
M.S. in Electrical and Electronics Engineering
Advisor: Cem Tekin
September 2019

Emergency Medical Services (EMS) provide the necessary resources when there is a need for immediate medical attention and play a significant role in saving lives in the case of a life-threatening event. Therefore, it is necessary to design an EMS system where the arrival times to calls are as short as possible. This task includes the ambulance redeployment problem that consists of the methods of deploying ambulances to certain locations in order to minimize the arrival time and increase the coverage of the demand points. As opposed to many conventional redeployment methods where the optimization is primary concern, we propose a learning-based approach in which ambulances are redeployed without any a priori knowledge on the call distributions and the travel times, and these uncertainties are learned on the way. We cast the ambulance redeployment problem as a multi-armed bandit (MAB) problem, and propose various context-free and contextual MAB algorithms that learn to optimize redeployment locations via exploration and exploitation. We investigate the concept of risk aversion in ambulance redeployment and propose a risk-averse MAB algorithm. We construct a data-driven simulator that consists of a graph-based redeployment network and Markov traffic model and compare the performances of the algorithms on this simulator. Furthermore, we also conduct more realistic simulations by modeling the city of Ankara, Turkey and running the algorithms in this new model. Our results show that given the same conditions the presented MAB algorithms perform favorably against a method based on dynamic redeployment and similarly to a static allocation method which knows the true dynamics of the simulation setup beforehand.

Keywords: Ambulance redeployment, online learning, multi-armed bandit problem, contextual multi-armed bandit problem, risk-aversion.

ÖZET

ÇOK KOLLU HAYDUTLAR İLE UYARLANABİLİR AMBULANS KONUMLANDIRMA

Ümitcan Şahin
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Tez Danışmanı: Cem Tekin
Eylül 2019

Acil Yardım Servisleri (AYS), acil tıbbi müdahaleye ihtiyaç duyulduğunda gerekli kaynakları sağlar ve yaşamı tehdit edici bir olay durumunda hayat kurtarmada önemli bir rol oynar. Bu nedenle, çağrılara varış sürelerinin mümkün olduğu kadar kısa olduğu bir AYS sistemi tasarlamak gereklidir. Bu görev, varış zamanını en aza indirmek ve talep noktalarının kapsamını arttırmak için ambulansları belirli yerlere yerleştirme yöntemlerinden oluşan ambulans konumlandırma problemini içermektedir. Bu çalışmada, eniyilemenin birincil öneme sahip olduğu birçok geleneksel konumlandırma yönteminin aksine, çağrı dağılımları ve seyahat süreleri hakkında önceden hiçbir bilgi olmadan ambulansların konumlandırıldığı ve bu belirsizliklerin zamanla öğrenildiği, öğrenmeye dayalı bir yaklaşım önerilmiştir. Ambulans konumlandırma problemi çok kollu haydut (ÇKH) problemi olarak modellenmiş, keşif ve istifade yoluyla konumlandırma yerlerini eniyilemeyi öğrenen bağlamsız ve bağlamsal ÇKH algoritmaları önerilmiştir. Ambulans konumlandırmada riskten kaçınma kavramı incelenmiş ve riskten kaçınan bir ÇKH algoritması önerilmiştir. Grafik tabanlı bir konumlandırma ağından ve Markov trafik modelinden oluşan veri odaklı bir simülatör oluşturulmuş ve bu simülatör üzerinde yürütülen algoritmaların performansları karşılaştırılmıştır. Ayrıca, Ankara şehrini modelleyerek ve algoritmaları bu yeni model üzerinde çalıştırarak daha gerçekçi konumlandırma simülasyonları elde edilmiştir. Elde ettiğimiz sonuçlar, aynı koşullar göz önüne alındığında, sunulan ÇKH algoritmalarının, dinamik konumlandırma temelli bir yöntemden daha iyi ve önceden simülasyon kurulumunun gerçek dinamiklerini bilen statik bir tahsis yönteme benzer şekilde çalıştığını göstermektedir.

Anahtar sözcükler: Ambulans konumlandırma, online öğrenme, çok kollu haydut problemleri, bağlamsal çok kollu haydut problemleri, riskten kaçınma.

Acknowledgement

I would first like to thank my advisor Dr. Cem Tekin for his relentless support and invaluable guidance throughout my graduate studies at Bilkent University. His patience and self-discipline did not only help me to complete this work, but also helped me to acquire the necessary skill set for becoming a better researcher.

I would also like to thank the rest of my thesis committee: Asst. Prof. Aykut Koç , Asst. Prof. Elif Vural for their time and valuable feedbacks.

I am indebted to Burak Bartan, Melih Bastopçu, Berkan Kılıç, Cem Bulucu, Eralp Turğay and Kübilay Ekşioğlu for being good friends even in my hard times. I will always remember our enjoyable conversations and coffee breaks.

I would also like to thank ASELSAN Inc. and all my colleagues in ASELSAN Research Center: Dr. Aykut Koç, Dr. Veysel Yücesoy, Lütfi Kerem Şenel, Kaan Karaman, Çağatay Işıl, Safa Onur Şahin, Assoc. Dr. Mustafa Yorulmaz, Utku Girit, Oğuzhan Fatih Kar and Hatice Doyduk for their valuable help in my research.

Finally, I would like to thank my family for all their support and helping me to become who I am today.

I would like to thank TÜBİTAK for supporting this work under 2210-A Scholarship Program.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Emergency Medical Services (EMS) are an integral part of the public services and responsible for the provision of scarce resources in times of critical events. Ambulance redeployment, which is an important topic in the design of an EMS system, comprises the problem of deploying ambulances to certain locations in order to

1. minimize the average arrival time to calls, and
2. increase the coverage of the demand points.

The first objective is related to overall reduction in the arrival times. Although this objective improves the overall quality of the EMS system, it can leave some individual calls not responded within a reasonable time in favor of the overall reduction in the arrival times of the majority of the calls. However, in a typical EMS system, reducing the average arrival time at the expense of some calls being not responded on time jeopardizes the reliability of such an EMS system in real life. Therefore, redeploying ambulances closer only to the demand points from where the most of the calls originate might be an ineffective method since it decreases the coverage of all demand points. This concern is represented by the second objective. Furthermore, in an EMS system, a call has a high chance of

coming from a life-threatening event and requires an immediate response; thus, it is very important to increase the coverage of all demand points so that as many calls as possible are responded within a reasonable time. Solving the redeployment problem by taking into account both objectives requires numerous challenges to be addressed.

One of these challenges is the limit on the number of ambulances that are idle to respond to any call. Since an ambulance is busy when responding to a call, the designed model should redeploy remaining idle ambulances in a way to cover the area that is now uncovered by the dispatched ambulance. Furthermore, fixed ambulance locations, which result in static allocation of the idle ambulances, also restrict the design of an efficient method in a sense that the ambulances are dispatched from the same stations even though the locations of these fixed stations might not effectively cover the time-varying demand points. Therefore, instead of the static allocation techniques, it is shown that a dynamic redeployment approach, i.e., adjusting the positions of the ambulances with respect to the demand points as the statistics of the calls change over time, results in both the reduction of the arrival times and the increase in the coverage of the demand points [1–19].

On the other hand, the dynamic redeployment models present some new challenges due to their dynamic nature. These challenges include the curse of dimensionality due to the increase in the number of ambulance locations and the computational complexity of the dynamic models. To overcome these problems, prior works focus on the approximate dynamic methods and near-optimal solutions with strong performance guarantees. [4, 6, 14].

The region specific parameters such as the expected number of calls from the demand points, i.e., call distributions and travel times on the roads might not be completely known before the optimal redeployment locations are computed for the ambulances. Furthermore, these parameters can dynamically change in a given day or from one day to another. Therefore, it is a vital task to learn these parameters in order to efficiently redeploy ambulances to the locations that result in the maximum coverage and the minimum arrival times. For this reason,

in this thesis we present a new learning-based approach in which the ambulances are redeployed without prior knowledge on the call distributions and travel times. Since our method learns where to redeploy ambulances in an online manner, it does not require any region specific information, which makes it easily applicable to real-life EMS systems.

We cast the ambulance redeployment problem (ARP) as a multi-armed bandit (MAB) problem. MAB problems investigate the trade-off between exploration and exploitation [20, 21]. This trade-off is best explained with the following example: a gambler (i.e., the agent) facing a number of slot machines (i.e., arms) has to decide which machines to select and in what order so as to maximize his gain. Since he does not know the probability distributions that generate the rewards of the machines, he has to spend some of his money and time to learn the expected rewards of the machines. This corresponds to exploration. At the same time, he also needs to select the machines that are found to be generating good rewards to maximize his total reward. This corresponds to exploitation. These two need to be carefully balanced, because by exploring too much the gambler might never get a chance to maximize his total reward, while by exploiting too much he might get stuck at a sub-optimal machine.

In ambulance redeployment, the MAB agent sequentially learns where to redeploy ambulances by taking into account the arrival times of the previously deployed ambulances. The location in which an ambulance can be redeployed correspond to a bandit arm. The rewards of such arms are determined according to the arrival times of ambulances to calls. Initially, the MAB agent redeploys ambulances to different locations in order to explore the arrival times from these locations to calls. Then, as the number of past calls increases, it redeploys ambulances to the locations with the estimated best arrival time and coverage based on the history of the previous calls and their arrival times, while still occasionally exploring new locations.

A crucial aspect of the ambulance redeployment problem is that parameters such as travel times that depend on external factors also affect the arrival time of ambulances to calls; hence, this side information should be used by the learning

agent to make better decisions. This side-information can be utilized through a variant of the MAB problem: the contextual multi-armed bandit problem (CMABP). In the contextual MAB, the agent observes a context (side information) at the beginning of each time slot that gives a hint about the arm rewards. Then, the agent decides on which arm to select both based on its history and the current context. The contextual MAB problem finds applications in many fields including recommender systems [22], medical diagnosis [23] and cognitive radio networks [24]. In short, the MAB agent learns optimum redeployment locations (that result in minimum arrival times and maximum coverage) based on fixed travel times, while the contextual MAB agent learns the optimal redeployment locations for time-dependent travel times.

Furthermore, in order to make sure that each call is responded within a reasonable time, the variance in the arrival times should be minimized as well as the expected arrival times. In the thesis, we investigate this effect in the ARP through the concept of risk-aversion in the MABP [25, 26]. By using a risk-averse MAB algorithm, that is, by designing a MAB algorithm that takes less risks when redeploying ambulances, a reduction in the worst-case arrival times is achieved. We also show that minimizing the variance in the arrival times leads to increase the expected arrival times. Therefore, this trade-off between worst-case arrival times and the overall expected arrival times is also investigated in the thesis.

To the best of our knowledge, both the MAB, the CMAB, and the risk-averse MAB problems have not been used in the context of ambulance redeployment prior to this study.

MAB algorithms are preferred over other learning methods in ambulance redeployment for two important reasons. First, MAB algorithms provide scalability over large data sets since they do not need to store every instance of the history (e.g., past calls, travel times, traffic status etc.). Second, MAB algorithms learn where to redeploy ambulances through a feedback mechanism called *partial* or *bandit feedback*, i.e., they can learn without actually observing the arrival time of every ambulance had they been placed at every possible location. Bandit feedback is inherent in the ARP since the feedback about arrival time and coverage

can only be observed for the selected deployment.

1.1 Our Contribution

The contributions of this thesis can be summarized as follows:

1. a new learning-based method to solve the dynamic ambulance redeployment problem in which problem characterizing parameters such as call distributions and travel times do not need to be known beforehand and learned on the way,
2. design of a new discrete time data-driven redeployment simulator on which the redeployment algorithms run,
3. empirical regret analysis of the proposed algorithms in the context of ambulance redeployment,
4. a detailed comparison of the new method with an existing static allocation and dynamic redeployment models in the literature, and
5. a new risk-averse MAB algorithm that redeploys ambulances in a way not only to minimize the expected arrival times, but also to minimize the variance in the arrival times.

1.2 Organization of the Thesis

The rest of the thesis is organized as follows: Next chapter includes the related work on ambulance redeployment. We include the literature on the static allocation and dynamic redeployment models, and how our method differs from them. In Chapter 3, we define the classical MAB and contextual MAB problems and propose our own adapted algorithms that are used in ambulance redeployment. In Chapter 4, we describe the graph-based redeployment network and the

Markov traffic model that we use in our simulations. In Chapter 5, we contruct a discrete-time data-driven redeployment simulator on which the proposed algorithms are run. In Chapter 6, we conduct simulations on a 15×15 redeployment network and compare the performances of the proposed algorithms against the static (SMEXCLP) and dynamic MEXCLP (DMEXCLP) models from the literature. In Chapter 7, we investigate risk aversion in ambulance redeployment, model the city of Ankara in Turkey for more realistic simulations, and compare the performance of the algorithms on this model. In Chapter 8, we conclude the thesis and share our research direction on future work.

Chapter 2

Related Work

The existing work on ambulance redeployment can be categorized into two: static allocation and dynamic redeployment problems.

2.1 Static Allocation Problem

The static allocation problem is solved once before all ambulances are deployed and the locations of idle ambulances are not adjusted as the other ambulances that are dispatched to calls become temporarily unavailable for future calls. Hence, this type of allocation problem is called the *static* allocation problem. The static allocation problem consists of deterministic and stochastic methods.

2.1.1 Deterministic Ambulance Redeployment

In the deterministic case, uncertainties in the availability of the ambulances are ignored. That is, all ambulances are assumed to be able to respond to any call at any given time. Most of the prior research on this case depends on the *location set covering model* (LSCM) proposed by Toregas *et al.* [27] and the *maximal covering*

location problem (MCLP) proposed by Church and ReVelle [28]. The LSCM aims to minimize the number of ambulances needed to cover all demand points and provides a lower bound on the number of ambulances. On the other hand, the MCLP aims to maximize the coverage given a limited number of ambulances. Both models, however, do not consider the case of busy ambulances once they are dispatched to calls. Therefore, they do not take precautions against the areas that are presently uncovered by the dispatched ambulances. Furthermore, they do not consider the case where multiple simultaneous calls originating from the same demand points. To address these issues, several variants of the LSCM and MCLP have been proposed (see e.g., [29–32]).

2.1.2 Stochastic Ambulance Redeployment

For the stochastic case, an important example of the stochastic location problem is the *maximum expected covering location problem* (MEXCLP) [33]. In this problem, the availability of the ambulances is modeled using independent Bernoulli random variables, and it is assumed that more than one ambulance can be present at the same location. Numerous variants of the MEXCLP model are proposed, including the model where the travel time and speed of the ambulances are assumed to be stochastic [34, 35] as well as the model in which the availability of the ambulances depends on each other [33, 36].

2.2 Dynamic Redeployment Problems

With the advances in the current technologies such as the Geographical Information System (GIS) and Geographical Positioning System (GPS), the dynamic ambulance redeployment problem can be solved in real-time, and hence, the positions of the ambulances can be readjusted based on ambulance availability and the expected calls from the demand points. The previous work includes the *dynamic double standard model* (DDSM) by Gendreau *et al.* [1], and the *advanced integer programming model* [2] where the cost of redeploying ambulances and the

coverage of future calls are incorporated into the objective function. Furthermore, the ambulance redeployment problem is cast as a *Markov Decision Process* (MDP) and solved via dynamic programming in order to capture the real-life complexity of the problem (i.e., the randomness in the system due to its dynamic nature) [3,37]. Since high-dimensional state space makes it computationally very hard to compute the optimal solution, approximate dynamic programming methods are proposed in [4,5,14], which use value function approximations for MDPs. Furthermore, some studies also use heuristics in their redeployment models in order to arrive at a reasonable redeployment strategy which is not guaranteed to be optimal [8, 10, 11]. In addition, [6] uses a data-driven simulator and a greedy allocation approach with submodularity to achieve near-optimal solutions in ambulance redeployment.

Some models also consider the relocation cost so as to penalize number of relocations made among ambulance waiting stations. One such study [7] uses a time-dependent MEXCLP model and aims to maximize the time-dependent coverage of the demand points while minimizing the number of relocations made among stations and the number of ambulances waiting in the same stations. In contrast, we do not directly introduce a penalty term in our model, but we restrict the number of relocations by allowing only idle ambulances to be redeployed at a given time instance. Redeploying only idle ambulances is also used by [15] to introduce an ambulance crew-friendly approach. Similarly, [8] also uses a mixed integer programming model with variable neighborhood search heuristic and aims to maximize the coverage of the demand points under time-dependent travel times. Similar to their study, we have also considered the case of time-dependent travel times by introducing time-dependent traffic states on the roads which are determined by a Markov traffic model. Instead of a black-and-white coverage consideration (i.e., ambulances respond under a given time or not), [9] measures the performance as the survival rates of the call by introducing a penalty function which is non-decreasing and depends on the arrival time.

Apart from these works, a detailed empirical comparison of the relocation strategies in real-time ambulance redeployment is made in [10]. [19] formulates the

redeployment problem as an integer linear program and proposes a dynamic redeployment model called *maximal expected coverage relocation problem* (MECRP) that maximizes the expected covered demand points. They conduct their analysis with real-life EMS data from Montreal. [18] uses the same MECRP model but formulates a generalized assignment model that aims to minimize the total times traveled by the ambulances. Similar to [19], we conduct sensitivity analysis with varying number of ambulances in our simulations. Furthermore, similar to the combinatorial assignment model in [18], we use the Hungarian method in our model to compute the optimal assignment that results in the least travel times when assigning idle ambulances to the waiting locations.

Online redeployment methods are proposed in [11], [12], [16]. [11] uses a penalty function that puts restrictions on the number of relocations using heuristics. [12] proposes a model called the *minimum expected penalty relocation problem* (MEXPREP), which uses compliance table policies where each compliance table level indicates the desired waiting site locations for the idle ambulances. Furthermore, [16] studies the impact of the frequency of redeployments, crew workload, presence of busy ambulances, and selection of performance criterion on ambulance redeployment.

In addition, [13] proposes a two-stage stochastic program where the optimal placement of the ambulances is determined in the first stage and in the second stage the uncertainty in the location of emergency calls is represented by a finite set of scenarios, each containing a random outcome for the location of the calls, and [15] considers a dynamic version of the MEXCLP model called DMEXCLP and uses heuristics to solve the dynamic redeployment problem in real-time. Their DMEXCLP model assumes that the travel times on the roads are fixed. On the other hand, [17] considers the DMEXCLP model with uncertain driving times. Both models assume that the fraction of demands from each demand point is to be known in advance and that it is proportional to the population that comprise this demand point. Although these assumptions are necessary for the DMEXCLP model to compute a near-optimal solution, there are other factors affecting the fraction of demand points such as age and gender distributions in a population. Therefore, in our simulations, we compare the performance of the

MAB algorithms with the DMEXCLP model under the parameter settings where the algorithms do not have the knowledge of the fraction of demand points (i.e., call distributions) and estimate these fractions using average sampling technique. More details are provided in Chapter 6.

The previous works discussed so far focus on the optimization aspect of the allocation and redeployment problems and require full or partial knowledge of the system dynamics such as call distributions and travel times.

On the other hand, in this work, we propose a new reinforcement learning-based approach for real-life EMS services where call distributions and travel times are stochastic and not known a priori.

Chapter 3

Multi-armed Bandits in Ambulance Redeployment

In this chapter, we first describe the MAB problem with single and multiple arm selections. Then, we describe how numerous MAB algorithms can be adapted for ambulance redeployment. Finally, we propose our own MAB algorithms for ambulance redeployment at the end of each section. The algorithms that we consider are either deterministic (use the principle of optimism under the face of uncertainty, and select arms based on upper confidence bounds [38]) or randomized (explore with a certain probability like ϵ_n-greedy [38] or sample from the posterior distribution [20] to decide on which arms to select).

3.1 The MAB Problem

In the MAB problem, there is a set of K arms, denoted by $\mathcal{A}$, where each $a \in \mathcal{A}$ denotes a possible ambulance location. At each round t, arm a generates a random reward $r_{t,a} \in [0, 1]$ that comes from a fixed but unknown distribution,

whose unknown expected value is μ_a.[1]

Single-arm selection: In the classical setting, the MAB algorithm π selects arm π_t in each round based on the history of its selections and observations, and gets to know only the reward r_{t,π_t} of arm π_t at round t. The objective is to maximize the expected total reward over T rounds, i.e., $\max \mathrm{E}\left\{\sum_{t=1}^{T} r_{t,\pi_t}\right\}$. This is equivalent to minimizing the *regret*, which is given as

$$R(T) := \mathrm{E}\left\{\sum_{t=1}^{T} r_{t,\pi_t^*}\right\} - \mathrm{E}\left\{\sum_{t=1}^{T} r_{t,\pi_t}\right\} \tag{3.1}$$

where π^* is the benchmark single-arm selection strategy. The benchmark is usually taken to be an oracle which knows the expected rewards of all arms in advance and always selects the arm with the highest expected reward. Since the reward distributions are fixed, the optimal arm is time-independent, and its expected reward is given as $\mu^* := \max_{a \in \mathcal{A}} \mu_a$. Thus, when the benchmark is this oracle, the regret of an MAB algorithm π in T rounds can be rewritten as:

$$R(T) = \mu^* T - \mathrm{E}\left\{\sum_{t=1}^{T} r_{t,\pi_t}\right\}. \tag{3.2}$$

In the context of ambulance redeployment, (3.2) measures the loss of the MAB algorithm that deploys a single ambulance in each round with respect to the oracle which knows the expected arrival times for all locations perfectly, and deploys a single ambulance to the best location in each round.

Multiple-arm selection: In this setting, instead of selecting a single arm, the MAB algorithm selects N_t arms at round t where $N_t \leq K$. Even if the distributions of the calls are known for all locations, this is in general a combinatorial optimization problem which is NP-hard [39], [40]. Thus, it is intractable to compete against an optimal oracle that knows the best deployment of N_t ambulances. Therefore, for the multiple-arm selection setting, we consider the following optimization oracle as the benchmark: static MEXCLP (SMEXCLP) [41]. SMEXCLP is an NP-complete static allocation method which knows all the call distributions; hence, the rewards of the ambulance locations beforehand and computes the

[1]These definitions are used in the following subsections. Furthermore, the terms *bandit arm* and *ambulance location* are used interchangeably throughout the text.

optimal N (i.e. $N_t = N$ when $t = 1$) locations $\pi_1^*,\ ...,\ \pi_N^*$ via integer programming when $t = 1$. It assigns the ambulances to these locations starting from π_1^* up to π_N^*. Then, it dispatches the closest ambulance $\pi_{t,n}^* \in \{\pi_1^*,\ ...,\ \pi_N^*\},\ n \in \{1, ..., N_t\}$ to the call at round t and receives the reward r_{t,π_n^*}. When the dispatched ambulance becomes idle after delivering the patient to the hospital, it returns to its pre-assigned location if there are no active calls. The multiple-arm regret with respect to the SMEXCLP algorithm is defined as follows:

$$R_m(T) := \mathrm{E}\left\{\sum_{t=1}^{T} r_{t,\pi_{t,n}^*}\right\} - \mathrm{E}\left\{\sum_{t=1}^{T} r_{t,\pi_{t,n}}\right\} \tag{3.3}$$

where $\pi_{t,n},\ n \in \{1, ..., N_t\}$ is the reward of the closest ambulance dispatched to a call at round t by the MAB algorithm.

From this point on, we only consider the multiple-arm selection problem and the corresponding regret definition in (3.3). Therefore, the learning methods described below aim to maximize the reward (i.e., minimize the regret) for the the MAB problem with multiple-arm selection.

3.1.1 Multiple-Arm UCB1 (MaUCB1)

The original UCB1 algorithm which selects a single arm at each round can be found in [38]. For this setting, for all $K > 1$, if policy UCB1 is run on K arms having arbitrary reward distributions $P_1, ..., P_K$ with support in $[0, 1]$ then its expected regret $\mathrm{E}[R(T)]$ after any number of rounds T is at most

$$\left[8 \sum_{i:\mu_i<\mu^*} \left(\frac{\ln T}{\Delta_i}\right)\right] + \left(1 + \frac{\pi^2}{3}\right)\left(\sum_{j=1}^{K} \Delta_j\right) \tag{3.4}$$

where $\mu_1, ..., \mu_K$ are the expected values of $P_1, ..., P_K$, μ^* is as defined earlier, and $\Delta_i := \mu^* - \mu_i,\ i \in \{1, ..., K\}$.

We modify the original UCB1 algorithm to allow for multiple-arm selection and call it the MaUCB1 algorithm.

At round t, MaUCB1 computes an index for each ambulance location a based on the observations from that location as follows:

$$g_{t,a} := \bar{r}_{t,a} + \sqrt{\frac{2 \log t}{n_{t,a}}} \tag{3.5}$$

where $\bar{r}_{t,a}$ is the sample mean reward of location a and computed by averaging the reciprocals of the arrival times of the ambulances that are located at a and dispatched to a call up to round t, t is the current round, $n_{t,a}$ is the number of times an ambulance has been placed at location a and dispatched to a call up to round t. At each round, MaUCB1 redeploys N_t (i.e. the number of idle ambulances) ambulances. These redeployments are made to the locations with the N_t highest indices, denoted by $\pi_{t,1}, \pi_{t,2}, ..., \pi_{t,N_t}$:

$$\begin{aligned} \pi_{t,1} &= \arg\max_{a \in \mathcal{A}} g_{t,a}, \\ \pi_{t,2} &= \arg\max_{a \in \mathcal{A} \setminus \{\pi_{t,1}\}} g_{t,a}, \\ &\vdots \\ \pi_{t,N_t} &= \arg\max_{a \in \mathcal{A} \setminus \{\pi_{t,i}\}_{i=1}^{N_t - 1}} g_{t,a}. \end{aligned} \tag{3.6}$$

In other words, we sequentially compute the single best location that has the highest index $g_{t,a}$ among the locations in $\mathcal{A}$, which is $\pi_{t,1}$, exclude this location from $\mathcal{A}$, and then compute again the best location that has the highest index among the locations in $\mathcal{A} \setminus \{\pi_{t,1}\}$. We proceed this way until all N_t locations are selected for ambulance redeployment.

As a call arrives at a location, the closest ambulance $\pi_{t,n},\ n \in \{1, ..., N_t\}$ is dispatched to the call and its reward $r_{t,\pi_{t,n}}$ is observed at round t. $r_{t,\pi_{t,n}}$ is used in calculating the regret in (3.3). Then, the sample mean reward $\bar{r}_{t,\pi_{t,n}}$ and $n_{\pi_{t,n}}$ are updated for the next round in which (3.5) is computed again for each location. The last term on the right-hand side of (3.5) is the exploration term. The exploration term measures the uncertainty in ambulance redeployment and has greater values when t is small and shrinks when t increases. It enables MaUCB1 to occasionally select locations that are rarely selected before, discover locations with high rewards, and avoid getting stuck at sub-optimal locations.

The computational complexity of the UCB1 algorithm for single-arm selection is given in [38]. In MaUCB1, the multiple-arm selection step in (3.6) over T rounds incurs the computational complexity of $\mathcal{O}(TK\log(K))$ in big $\mathcal{O}$ notation.

3.1.2 Multiple-arm ϵ_t-greedy

The original ϵ_t-greedy algorithm proposed in [38] selects a single arm at each round. For this setting, for all $K > 1$, if policy ϵ_t-greedy is run with input parameter

$$0 < d \leq \min_{i:\mu_i<\mu^*} \Delta_i, \tag{3.7}$$

then the probability that after any number $T \geq cK/d$ of rounds, ϵ_t-greedy chooses a suboptimal arm j is at most

$$\frac{c}{d^2T} + 2\left(\frac{c}{d^2}\ln\frac{(T-1)d^2e^{1/2}}{cK}\right)\left(\frac{cK}{(T-1)d^2e^{1/2}}\right)^{c/(5d^2)} + \frac{4e}{d^2}\left(\frac{cK}{(T-1)d^2e^{1/2}}\right)^{c/2}. \tag{3.8}$$

where for c large enough (e.g. $c > 5$), the above bound is of order $c/(d^2T)+o(1/n)$ for $n \to \infty$ as the second and third terms in the bound are $\mathcal{O}(1/n^{1+\epsilon})$ for some $\epsilon > 0$ in big $\mathcal{O}$ notation.

$$\left[8\sum_{i:\mu_i<\mu^*}\left(\frac{\ln T}{\Delta_i}\right)\right] + \left(1+\frac{\pi^2}{3}\right)\left(\sum_{j=1}^{K}\Delta_j\right) \tag{3.9}$$

where $\mu_1, ..., \mu_K$ are the expected values of $P_1, ..., P_K$, μ^* is as defined earlier, and $\Delta_i := \mu^* - \mu_i$. Here ϵ_t can be considered as the probability of uncertainty in our redeployments. We modify this algorithm such that N_t ambulances are redeployed at each round, and call the modified algorithm multiple-arm ϵ_t-greedy. ϵ_t is taken to be $\dfrac{1}{t}$ so that it is a decreasing function of t. The flow of the algorithm is given as follows:

1. First perform N_t independent Bernoulli trials with success probability $1-\epsilon_t$ (a Bernoulli trial is a coin toss experiment where the probability of heads coming up is $1-\epsilon_t$ and considered as success, and the probability of tails coming up is ϵ_t and considered as failure.)

2. If there are $S_t \leq N_t$ number of successes in these trials (i.e., the number of times heads come up is S_t out of N_t coin tosses), then redeploy S_t ambulances to the locations with the highest sample mean rewards:

$$\begin{aligned}
\pi_{t,1} &= \arg\max_{a \in \mathcal{A}} \bar{r}_{t,a}, \\
\pi_{t,2} &= \arg\max_{a \in \mathcal{A} \setminus \{\pi_{t,1}\}} \bar{r}_{t,a}, \\
&\vdots \\
\pi_{t,S_t} &= \arg\max_{a \in \mathcal{A} \setminus \{\pi_{t,i}\}_{i=1}^{S_t-1}} \bar{r}_{t,a}.
\end{aligned} \tag{3.10}$$

   In other words, we sequentially select the location $\pi_{t,1}$ with the highest sample mean reward $\bar{r}_{t,a}$ which is computed by averaging the reciprocals of the arrival times of the ambulances that are dispatched to a call from a. Then we exclude a from the location set $\mathcal{A}$ and select the location with the second highest mean reward and continue in this manner until we select all N_t locations with the highest sample sample mean rewards for ambulance redeployment.

3. Choose uniformly at random $N_t - S_t$ locations from the remaining locations and redeploy the remaining ambulances to these locations.

Here, ϵ_t controls the trade-off between exploration and exploitation. It decreases as t increases so as to allow for more exploration at the beginning and more exploitation as the round number increases. This means that as ϵ_t decreases we are able to select good ambulance locations with higher probability. After the ambulances are redeployed, similar to MaUCB1, the closest ambulance π_t is dispatched to the call and its reward $r_{t,\pi_{t,n}}$ is observed and used in computing the regret in (3.3). Similar to the computational complexity analysis of MaUCB1, the multiple-arm selection step in (3.10) over T rounds incurs the computational complexity of $\mathcal{O}(TK\log(K))$.

3.1.3 Multiple-Arm Thompson Sampling (MaTS)

Thompson sampling [20] is a Bayesian approach that assumes prior probability distributions on the arms, and then, selects the arm whose probability of being the best arm (i.e., leading to the maximum reward) is the highest at each round. Then, it updates the posterior based on the observed reward of the selected arm. For the classical setting, Thompson Sampling algorithm has expected regret

$$\mathrm{E}\left[R(T)\right] \leq \mathcal{O}\left(\left(\sum_{i:\mu_i<\mu^*} \frac{1}{\Delta_i^2}\right)^2 \ln T\right) \tag{3.11}$$

in rounds T in big $\mathcal{O}$ notation. For the ambulance redeployment problem, we modify the Bernoulli Thompson sampling algorithm for the general stochastic bandit problem proposed in [42]. The flow of the modified algorithm at round t is given as follows:

1. For each ambulance location a, set success $S_{t,a}$ and failure $F_{t,a}$ rates of the beta distributions, based on the history of observations. Start with zero success and failure rates.

2. For each ambulance location a, draw a sample $\theta_a(t)$ from the posterior distribution $\mathrm{Beta}(S_{t,a}+1, F_{t,a}+1)$.

3. Redeploy ambulances to N_t locations as follows:

$$\begin{aligned} \pi_{t,1} &= \arg\max_{a\in\mathcal{A}} \theta_a(t) \\ &\vdots \\ \pi_{t,N_t} &= \arg\max_{a\in\mathcal{A}\setminus\{\pi_{t,i}\}_{i=1}^{N_t-1}} \theta_a(t). \end{aligned} \tag{3.12}$$

4. Dispatch the closest ambulance $\pi_{t,n}$ to the call and receive the reward $r_{t,\pi_{t,n}}$.

5. Perform a Bernoulli trial with success probability $r_{t,\pi_{t,n}}$: If success, then $S_{\pi_{t,n}} = S_{\pi_{t,n}} + 1$, else $F_{\pi_{t,n}} = F_{\pi_{t,n}} + 1$.

The regret is again computed according to (3.3) and the multiple-arm selection step in (3.12) over T rounds incurs the computational complexity of $\mathcal{O}(TK\log(K))$.

3.2 The Contextual MAB Problem

As mentioned in Chapter 1, a useful variant of the MAB problem is the contextual MAB problem in which the agent is provided with a side information called *context* at the beginning of each round. In the ambulance redeployment, the context represents the traffic status between each ambulance location, which determines the travel time between these locations. For this task, we modify the following MAB algorithms: LinUCB [43] and the contextual Thompson sampling [44] in order to account for multiple ambulance redeployment.

In this setting, the rewards of the arms are assumed to be the linear combinations of the K-dimensional context associated with them.[2] That is,

$$\mathrm{E}[r_{t,a}|\boldsymbol{\phi}_a(t)] = \boldsymbol{\phi}_a(t)^{\mathrm{T}}\boldsymbol{y}^* \tag{3.13}$$

where $r_{t,a}$ is the reward of the arm a at round t, $\boldsymbol{\phi}_a(t)$ is its K-dimensional context vector, and $\boldsymbol{y}^*$ is the unknown K-dimensional linear coefficient.

3.2.1 Multiple-arm LinUCB (MaLinUCB)

The LinUCB algorithm which selects a single arm is described in [43]. For this setting, if the arm set $\mathcal{A}$ is fixed and contains K arms, and the context of each arm is of K dimension and satisfies (3.13), then the expected regret of the LinUCB algorithm is at most

$$\mathrm{E}\left[R(T)\right] \leq \tilde{\mathcal{O}}\left(\sqrt{K^2T}\right) \tag{3.14}$$

in rounds T where $\tilde{\mathcal{O}}(.)$ is the same as $\mathcal{O}(.)$ but suppresses logarithmic factors.

[2]This assumption holds for ambulance redeployment. The details are given in Chapter 4.

Algorithm 1 Multiple-arm LinUCB

1: $\mathbf{A} \leftarrow \mathbf{I}_K$ ($\mathbf{I}_K$ is the $K \times K$ identity matrix)
2: $b \leftarrow 0_K$ (0_K is the $K \times 1$ zero vector)
3: **for** $t = 1, ..., T$ **do**
4: Observe the context matrix for all arms at round t: $\mathbf{\Phi}_t$ ($\mathbf{\Phi}_t$ is of dimension $K \times K$; in other words, every arm has K-dimensional context vector)
5: $\hat{\theta} = \mathbf{A}^{-1}b$
6: $h_t = \mathbf{\Phi}_t^{\mathrm{T}}\hat{\theta} + diag(\alpha\sqrt{\mathbf{\Phi}_t^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{\Phi}_t})$
7: Select N_t locations with the highest h_t terms (h_t is of dimension $K \times 1$; therefore, we first order it in a decreasing fashion and select the first N_t arms that has the highest h_t)
8: Observe a new call and the arrival time of the dispatched ambulance from $\pi_{t,n}$ out of the N_t selected locations in Step 7, construct the reward $r_{t,\pi_{t,n}}$ as the reciprocal of the arrival time
9: $\mathbf{A} = \mathbf{A} + X_{t,\pi_{t,n}}^{\mathrm{T}}\Phi_{t,\pi_{t,n}}$ ($\Phi_{t,\pi_{t,n}}$ is the context of arm $\pi_{t,n}$ in round t. In Step 4, we observe the context for all arms)
10: $b = b + \Phi_{t,\pi_{t,n}}r_{t,\pi_{t,n}}$
11: **end for**

We extend LinUCB to account for multiple arm selections and call the new algorithm MaLinUCB. At round t, MaLinUCB operates as follows:

1. Observe the K-dimensional context $\boldsymbol{\phi}_a(t)$ for each ambulance location a.
2. Based on the history $\mathcal{H}$ (i.e. previously observed contexts and the rewards of the dispatched ambulances), choose N_t ambulance locations $\{\pi_{t,i}\}_{i=1}^{N_t}$ for redeployment using the N_t of the highest indices given in (3.15).
3. Dispatch the closest ambulance $\pi_{t,n}$ out of N_t ambulances to the call and receive the reward $r_{t,\pi_{t,n}}$ whose expected value is given by (3.13).
4. Update the history $\mathcal{H}$ with the new observation $\left(\{\boldsymbol{\phi}_a(t)\}_{a\in\mathcal{A}}, \pi_{t,n}, r_{t,\pi_{t,n}}\right)$.

For step (2), similar to MaUCB1, MaLinUCB constructs an index term for each location a:

$$h_{t,a} := \boldsymbol{\phi}_a(t)^{\mathrm{T}}\hat{\boldsymbol{y}} + \alpha\sqrt{\boldsymbol{\phi}_a(t)^{\mathrm{T}}\mathbf{A}_t^{-1}\boldsymbol{\phi}_a(t)} \tag{3.15}$$

where $\mathbf{A}_t := (\mathbf{D}_t^{\mathrm{T}}\mathbf{D}_t + \mathbf{I}_K)$, $\mathbf{D}_t$ is the design matrix whose rows are the context vectors of the locations from which an ambulance is dispatched up to

round t: $\{\boldsymbol{\phi}_{\pi_\tau,n}(\tau)\}_{\tau\in\{1,\ldots,t-1\}}$, $\mathbf{I}_K$ is the $K \times K$ identity matrix. We set $\alpha = 1 + \sqrt{(log2/\delta)/2}$ for any $\delta > 0$. Furthermore, $\hat{\boldsymbol{y}} = \mathbf{A}_t^{-1}\boldsymbol{b}_t$ is an estimate of $\boldsymbol{y}^*$ where $\boldsymbol{b}_t$ is the summation of the previously observed rewards of the dispatched ambulances $\{r_{\tau,\pi_{\tau,n}}\}_{\tau=\{1,\ldots,t-1\}}$ multiplied by the context vector of the locations that the ambulances are dispatched from. Then, N_t ambulances are redeployed to the locations with the highest $h_{t,a}$ indices. The computational complexity of $\mathcal{O}(K^3T)$ is incurred in big $\mathcal{O}$ notation due to the matrix inversions in Step 5 and 6 and the for loop over T rounds (instead of using Gauss-Jordan elimination in matrix inversion, iterative matrix multiplication can be used in solving Step 5 and 6, which leads to the computational complexity of $\mathcal{O}(TK\log(K))$).

The regret definition for MaLinUCB is given by (3.3) and again computed with respect to the rewards of the closest ambulances that are dispatched to the calls. The pseudo-code of the MaLinUCB algorithm is given in Algorithm 1. In Step 6, $diag(X)$ returns the elements on the main diagonal of matrix X.

3.2.2 Multiple-arm Contextual Thompson Sampling

The multiple-arm contextual Thompson Sampling model combines the arm selection strategy of multiple-arm context-free Thompson Sampling with the linear contextual model given in (3.13). For this, we modify the contextual Thompson Sampling algorithm presented in [44] to allow for multiple-arm selection. For the single-arm selection setting where there are K arms that have K-dimensional context vectors, the total regret in rounds T for the contextual Thompson Sampling algorithm under the linear payoff function model in (3.13) is bounded by

$$\mathrm{E}\left[R(T)\right] \leq \tilde{\mathcal{O}}\left(K^2\sqrt{T}\right) \tag{3.16}$$

where $\tilde{\mathcal{O}}(.)$ is the same as $\mathcal{O}(.)$ but suppresses logarithmic factors.

The flow of the modified algorithm at round t is described as follows:

1. Let $\mathbf{A}$ and b have the same definitions as in the previous section for MaLinUCB.

Algorithm 2 Multiple-arm Contextual Thompson Sampling

$\mathbf{A} \leftarrow \mathbf{I}_K$ ($\mathbf{I}_K$ is the $K \times K$ identity matrix)
$v \leftarrow \sqrt{9K\log T}$
$b \leftarrow 0_K$ (0_K is the $K \times 1$ zero vector)
$\hat{\mu} \leftarrow 0_K$ (0_K is the $K \times 1$ zero vector)
for $t = 1, ..., T$ **do**
    Observe the context matrix for all arms at round t: $\mathbf{\Phi}_t$
    Compute the mean vector of the joint Gaussian distribution: $\hat{\mu} = \mathbf{A}^{-1}b$
    Compute the covariance matrix of the joint Gaussian distribution: $\Sigma = v^2\mathbf{A}^{-1}$
    $\Sigma = (\Sigma + \Sigma^{\mathrm{T}})/2$
    Sample an instance $\tilde{\mu}$ from the joint Gaussian distribution $\mathcal{N}(\hat{\mu}, \Sigma)$
    Compute the Thompson sampling terms for all arms: $p_t = \mathbf{\Phi}_t^{\mathrm{T}}\tilde{\mu}$
    Select the N_t locations with the highest p_t terms
    Observe a new call and the arrival time of the dispatched ambulance from $\pi_{t,n}$ out of the N_t selected locations in Step 12, construct the reward $r_{t,\pi_{t,n}}$ as the reciprocal of the arrival time
    $\mathbf{A} = \mathbf{A} + \Phi_{t,\pi_{t,n}}^{\mathrm{T}}\Phi_{t,\pi_{t,n}}$
    $b = b + \Phi_{t,\pi_{t,n}}r_{t,\pi_{t,n}}$
end for

2. Compute the mean of the prior distribution $\hat{\mu}_{t,a} = \mathbf{A}_t^{-1}b$ for each ambulance location a.

3. Form a priori distribution on the ambulance locations using multivariate Gaussian distribution $\mathcal{N}(\hat{\boldsymbol{\mu}}_t, v^2\mathbf{A}_t^{-1})$ where $\hat{\boldsymbol{\mu}}_t := \{\hat{\mu}_{t,a}\}_{a\in\mathcal{A}}$, $v = \sqrt{9K\log(T)}$, K is the dimension of the context, and T is the total number of rounds.

4. Draw a sample $\tilde{\boldsymbol{\mu}}_t$ from $\mathcal{N}(\hat{\boldsymbol{\mu}}_t, v^2\mathbf{A}_t^{-1})$.

5. Redeploy ambulances to N_t locations as follows:

$$\begin{aligned}\pi_{t,1} &= \arg\max_{a\in\mathcal{A}} \boldsymbol{\phi}_a(t)^{\mathrm{T}}\tilde{\boldsymbol{\mu}}_t \\ &\vdots \\ \pi_{t,N_t} &= \arg\max_{a\in\mathcal{A}\setminus\{\pi_{t,i}\}_{i=1}^{N_t-1}} \boldsymbol{\phi}_a(t)^{\mathrm{T}}\tilde{\boldsymbol{\mu}}_t. \end{aligned} \tag{3.17}$$

6. Dispatch the closest ambulance $\pi_{t,n}$ to the call and observe the reward $r_{t,\pi_{t,n}}$ whose expected value is given by (3.13).

The regret is again computed as given in (3.3). Similarly to the MaLinUCB, the computational complexity of $\mathcal{O}(K^3T)$ is incurred in big $\mathcal{O}$ notation due to the matrix inversions in Step 7 and 8 and the for loop over T rounds. The pseudo-code of the MaCTS algorithm is given in Algorithm 2.

Chapter 4

Ambulance Redeployment Setup and The Traffic Model

In this section, we describe the network that we use in the ambulance redeployment problem and the traffic model that generates the context for the contextual MAB algorithms.

4.1 Redeployment Network

A sample network with $K = 9$ nodes is shown in Fig. 4.1. Each node a in the graph is a demand point to which an ambulance can be redeployed and from which calls originate. The adjacent nodes including the diagonal ones are connected to each other. The edges between each node can be considered as a two-way road that is represented with traffic indices $x_{i,j}(t)$ and $x_{j,i}(t)$[1] that are real-valued numbers in $[0, 1]$ indicating the intensity of the traffic from the node i to j and j to i at round t, respectively. For example, if the intensity $x_{i,j}(t)$ is 1, then the edge is blocked and node i and node j are disconnected at round t

[1]For notational simplicity we use $x_{i,j}(t)$ instead of $x_{a_i,a_j}(t)$ when indicating the traffic index from node a_i to node a_j. Therefore, we refer to node a_i when we say node i throughout the text.

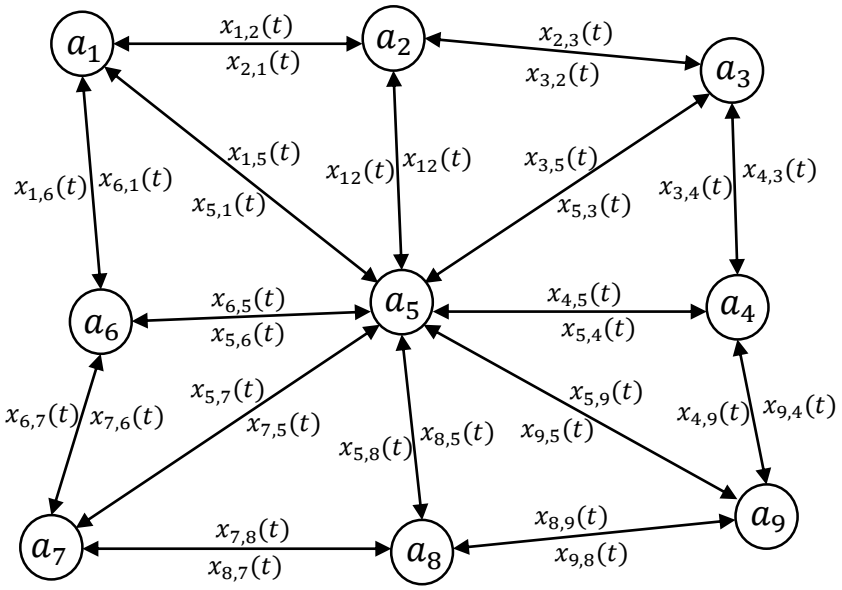


Figure 4.1: The ambulance redeployment network with $K = 9$ nodes: a directed graph that consists of ambulance location a and the traffic index $x_{i,j}(t)$ on the edge (i, j) which indicates the intensity of the traffic going from node a_i to node a_j at round t.

although the edge from node j to i might be connected at round t. The traffic indices determine the travel times at the edges. Since they are time-dependent, travel times are also time-dependent and used as context by the contextual MAB algorithms.

We assume that at round t a call can occur at one of the nodes $c(t) \in \{1, ..., K\}$ and our task is to redeploy ambulances in such a way that the average arrival time to all calls over T rounds is minimized. It is also assumed that before the call at node $c(t)$ takes place, the traffic indices $x_{i,j}(t),\ i, j \in \{1, ..., K\}$ at round t along with previously observed call nodes up to round t, i.e., $\{c(\tau)\}_{\tau \in \{1,...,t-1\}}$, are available to the learning algorithm. The length of the edge between node i and j is denoted by $d_{i,j}$. Furthermore, ambulances can move at different speeds which are determined by whether they are idle or busy and the traffic indices $x_{i,j}(t),\ i, j \in \{1, ..., K\},\ i \neq j$ at the edges. Similar to the assumption in [15], if an ambulance is returning to its waiting location after it becomes idle, its travel

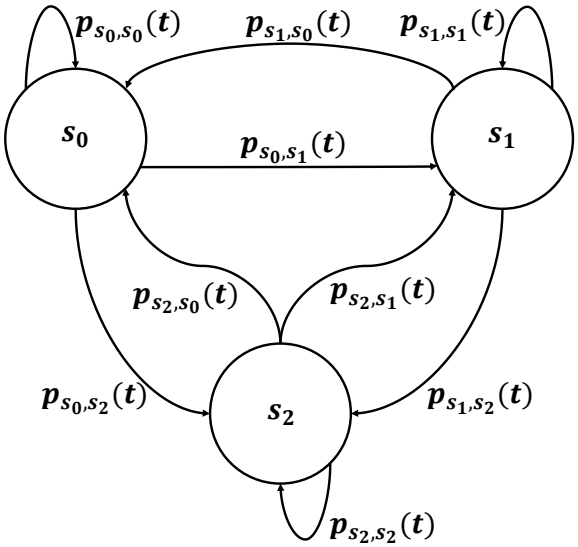


Figure 4.2: Traffic status modeled with three Markov states s_0, s_1, and s_2 that correspond to moving, light, and heavily congested traffic, respectively.

speed is 0.9 times the travel speed when it is responding to a call or going to the hospital. Therefore, the arrival time of an ambulance to $c(t)$ is computed using the traffic indices $x_{i,j}(t),\ i,j \in \{1,...,K\},\ i \neq j$ at round t, the speed of the ambulance, and the edge lengths $d_{i,j},\ i,j \in \{1,...,K\}$, which is described in detail in the following section.

4.2 Traffic Model

In addition to considering fixed travel times as in [15], we also consider time-dependent travel times similar to [8]. To generate time-dependent travel times, the traffic intensity $x_{i,j}(t)$ at edge (i,j) is modeled as a Markov chain with state space $S = \{s_0, s_1, s_2\}$ and state transition probabilities $p_{s_k,s_l}(t),\ k,l \in \{0,1,2\}$ as shown in Fig. 4.2. Traffic index $x_{i,j}(t)$ depends on the state of edge (i,j) at round t. Since $x_{i,j}(t) = 0$ corresponds to no traffic congestion on the edge (i,j) and $x_{i,j}(t) = 1$ corresponds to a disconnected edge at round t, the states s_0, s_1, and s_2 represent moving, light and heavily congested traffic states, respectively, and they are controlled by the transition probabilities given in Figure 4.2.

There are two reasons why a Markov model is used in determining the traffic states. The first one is to make the transitions between traffic states smoother. This stems from the fact that in reality we cannot expect edges to be connected and disconnected at consecutive rounds unless there is an accident which is also represented with the transition probabilities $p_{s_0,s_2}(t)$ and $p_{s_2,s_0}(t)$ in the Markov traffic model. The second reason is mainly due to introducing randomness into the system and to show how the performances of the MAB algorithms are affected by the randomness in the traffic states.

The speed of an ambulance that is traveling from node i to j at round t is calculated as $V_{i,j}(t) = (1-x_{i,j}(t))V_{max}$ where V_{max} is the maximum speed attained by the ambulances if $x_{i,j}(t) = 0$, i.e., there is no traffic congestion on edge (i,j) at round t and the ambulance is responding to a call or going to the hospital. We also note that there might be multiple paths (infinite number of paths if the loops are allowed) that goes from node i to node j. Therefore, letting $\mathcal{P}_{i,j}$ be the set of loop-free paths that goes from node i to j and $M_{i,j}$ be the number of such distinct paths, the time it takes to go from node i to j following a path $p_{ij}^m \in \mathcal{P}_{i,j}$ from these $M_{i,j}$ distinct paths is computed as follows:

$$\tau_{i,j}^m(t) = \frac{1}{V_{max}} \sum_{(k,l)\in p_{ij}^m} \frac{d_{k,l}}{1 - x_{k,l}(t)},\ m \in \{1, ..., M_{i,j}\} \tag{4.1}$$

where k and l are two consecutive nodes in the path p_{ij}^m and the superscript m denotes the mth path in $\mathcal{P}_{i,j}$ that goes from node i to j. For example, from Fig. 4.1, let $\mathcal{P}_{1,5}$ be the set of all loop-free paths going from node 1 to 5 and $p_{15}^m,\ m = 1$ denote the path that consists of the nodes 1, 6, and 5, then (4.1) can be computed as

$$\tau_{1,5}^1(t) = \frac{1}{V_{max}} \left(\frac{d_{1,6}}{1 - x_{1,6}(t)} + \frac{d_{6,5}}{1 - x_{6,5}(t)} \right).$$

Furthermore, $M_{i,j}$-dimensional vector $\boldsymbol{\tau}_{i,j}(t)$ denotes the traveling time of all the paths going from node i to j, i.e., $\tau_{i,j}^m(t) \in \boldsymbol{\tau}_{i,j}(t),\ m \in \{1, ..., M_{i,j}\}$ and we also note that $\boldsymbol{\tau}_{i,j}(t) = \mathbf{0},\ i = j$.

Using (4.1), we compute the arrival time of an ambulance at node i to a call at node j as the minimum time it takes to go from node i to j. For this, the

shortest path is defined as $m^* := \arg\min_{m\in\{1,...,M_{i,j}\}} \tau_{i,j}^m(t)$, and the arrival time of the ambulance at node i to the call at node j is calculated as

$$\gamma_{i,j}(t) := \tau_{i,j}^{m^*}(t). \tag{4.2}$$

Next, we define the context that is used in the MAB algorithms. The context between node i and node j is computed as

$$\begin{aligned} \phi_{i,j}(t) &= \frac{1}{\gamma_{i,j}(t)},\ j \in \{1,...,K\},\ j \neq i \\ \phi_{i,j}(t) &= 1,\ j = i \end{aligned} \tag{4.3}$$

where it is assumed that $\phi_{i,j}(t)$ is between 0 and 1. That is, we assume that the arrival time from a node to an adjacent node is at least 1, i.e., $\gamma_{i,j}(t) \geq 1, i \neq j$ and $\gamma_{i,j}(t) = 0, i = j$ (i.e., if an ambulance is in the same node as the call, then the arrival time is 0). The redeployment network is selected in the simulations in such a way that this assumption holds.

K-dimensional context vector of node i is denoted by $\boldsymbol{\phi}_i(t)$, whose elements are given by (4.3) for all $j \in \{1,...,K\}$. In other words, the context associated with node i is the inverse of the arrival time on the path to node j, if followed, leads to the minimum arrival time in (4.2) and the context vector is the collection of all such contexts computed from node i to all other nodes $j \in \{1,...,K\}$. In a real life EMS system, the current GIS and GPS technologies can provide the EMS responders with (4.2). To compute (4.2), we use the Dijkstra's algorithm in our simulations.

Following the redeployment network and traffic model definitions, we now show that the linearity assumption in (3.13) holds for the ambulance redeployment problem: The reward of an ambulance at node a that responds to a call at node $c(t)$ in round t is the inverse of the arrival time, i.e., $r_{t,a} = 1/(\gamma_{a,c(t)}(t))$, $\gamma_{a,c(t)}(t) \neq 0$ and $r_{t,a} = 1$ if $\gamma_{a,c(t)}(t) = 0$ due to the assumption $\gamma_{a,c(t)}(t) \geq 1,\ a \neq c(t)$ and $\gamma_{a,c(t)}(t) = 0,\ a = c(t)$ made previously.

Furthermore, the context of location a in round t is given by a K-dimensional vector $\boldsymbol{\phi}_a(t)$, whose elements are the inverse of the arrival times to calls that

originate from the nodes, which is computed in (4.3).

When the call distribution is independent and identical over rounds, denoting the probability that node i generates a call by p_i (where it holds that $0 \leq p_i \leq 1$ and $\sum_{i=1}^{K} p_i = 1$), we have by the definition of expectation

$$\mathrm{E}[r_{t,a}|\boldsymbol{\phi}_a(t)] = p_1\phi_{a,1}(t) + p_2\phi_{a,2}(t) + ... + p_K\phi_{a,K}(t)$$

where $\phi_{a,c(t)}(t) = r_{t,a},\ c(t) \in \{1, ..., K\}$ and $\phi_{a,a}(t) = 1,\ a \in \{1, ..., K\}$. Thus, we have $\boldsymbol{y}^* = [p_1, ..., p_K]^{\mathrm{T}}$ in (3.13). In other words, the unknown coefficient $\boldsymbol{y}^*$ is the nodes' likelihood of generating a call.

Chapter 5

Data-driven Ambulance Redeployment Simulator

In order to evaluate the performance of the algorithms, we designed a discrete-time data driven redeployment simulator. The simulator is given in Algorithm 3. The inputs of the simulator are defined as follows:

- K: the number of nodes in the redeployment network in Fig. 4.1.
- N: the number of idle ambulances at $t = 1$.
- T: the number of rounds.
- π: a function which takes the number of idle ambulances and the history $\mathcal{H}$, i.e., all the previous traffic information and arrival times of the ambulances up to round t, as input and outputs the new location of the ambulances. As described in Chapter 3, π is the redeployment strategy used by the MAB algorithms.
- $\mathcal{C}$: the call distribution from which the calls at different rounds are sampled.
- t_r: the number of rounds between two consecutive redeployment events.

Algorithm 3 Data-driven Redeployment Simulator

Input: K, N, T, π, R, t_r, t_c, X
$N_t = N,\ t = 1$ //number of idle ambulances
$\mathcal{H} \leftarrow \emptyset$ //history set
$\mathcal{Q} \leftarrow \emptyset$ //queue set
$\mathcal{E} \leftarrow c(t) \sim \mathcal{C},\ t = 1, 2, ..., T$ //set of call events
insert redeployment events in every t_r rounds to $\mathcal{E}$
for $t = 1, 2, ..., T$ **do** //discrete rounds
    remove arriving event list e_t from $\mathcal{E}$
    observe the traffic indices $\boldsymbol{x}_{i,j}(t) \sim X$
    for $c(i) \in \mathcal{Q}$ //starting from the top of $\mathcal{Q}$ **do**
        $N_{temp} = N_t$
        if $N_{temp} \neq 0$ **then**
            dispatch the closest ambulance π_i to $c(i)$
            observe the arrival time of π_i: $\gamma_{\pi_i, c(i)}$ in (4.2)
            $N_{t+1} = N_{temp} - 1$
            $N_{temp} = N_{temp} - 1$
            $\mathcal{H} \leftarrow \mathcal{H} \cup \{\boldsymbol{x}_{i,j}(t), \gamma_{\pi_i, c(i)}\}$
            insert call completion event at $t + t_c$ in $\mathcal{E}$
        end if
    end for
    if $c(t) \in e_t$ **then** //call arrival event
        if $N_t \neq 0$ **then** //if there are idle ambulances
            dispatch the closest ambulance π_t to $c(t)$
            observe the arrival time of π_t: $\gamma_{\pi_t, c(t)}$ in (4.2)
            $N_{t+1} = N_t - 1$
            $\mathcal{H} \leftarrow \mathcal{H} \cup \{\boldsymbol{x}_{i,j}(t), \gamma_{\pi_t, c(t)}\}$
            insert call completion event at $t + t_c$ in $\mathcal{E}$
        else // If all ambulances are busy
            put $c(t)$ at the bottom of $\mathcal{Q}$ //first-come first-serve
        end if
    end if
    if call completion event $\in e_t$ **then**
        $N_{t+1} = N_t + 1$
    end if
    if redeployment event $\in e_t$ **then**
        redeploy the idle ambulances using $\pi(N_t, \mathcal{H})$
    end if
end for

- t_c: the number of rounds needed for an ambulance to serve a call and be idle again.

- X: the traffic distribution from which the traffic indices $x_{i,j}(t),\ i,j \in \{1,...,K\},\ i \neq j$ at different rounds are sampled.

We make the following assumptions on the calls originating from the demand points: Only a single call is allowed to take place at round t, and the sampled call $c(t)$ at round t is independent from the calls in the previous rounds up to round t and only depends on the external factors (e.g., time of day, location demographics and geographies, road conditions etc.). Therefore, we take $\mathcal{C}$ to be a Poisson binomial distribution with the parameters $\{\lambda_{\mathcal{C}}(t)\}_{t=1}^{T}$.

The simulator keeps track of the idle ambulances, i.e., the ones that are not currently responding to any calls, and the events. The number of idle ambulances at round t is denoted by N_t. The event list $e_t \in \mathcal{E}$ at round t consists of call arrival, call completion, and redeployment events.

The operation of the simulator can be summarized as follows: In every round, we first observe the event list e_t and the traffic indices $\boldsymbol{x}_{i,j}(t),\ i,j \in \{1,...,K\},\ i \neq j$. Then, if e_t includes a call event $c(t)$, the following steps take place:

1. If there are call events in the queue list $\mathcal{Q}$, the idle ambulances are dispatched to the calls in this list starting from the top (i.e., the 'first-come-first-serve' strategy).

2. If a call event takes place, there is an idle ambulance (i.e., $N_t > 0$), and no call in $\mathcal{Q}$, then dispatch the closest ambulance $\pi_t = \arg\min_a \gamma_{a,c(t)}(t)$ to the call according to (4.2). If all ambulances are busy, then the event $c(t)$ is put at the bottom of $\mathcal{Q}$.

3. Observe the arrival time $\gamma_{\pi_t,c(t)}(t)$ of π_t.

4. Update the history with the traffic indices and the observed arrival time $\gamma_{\pi_t,c(t)}(t)$ at round t.

5. Add the call completion event at round $t + t_c$ to the event set $\mathcal{E}$.

If e_t also includes a call completion event of some previous call, we add the responding ambulance to the list of idle ambulances and redeploy it to its former location before the call. If e_t also includes a redeployment event, we redeploy the idle ambulances using the redeployment strategy π which takes N_t and $\mathcal{H}$ as inputs and outputs the new locations of the ambulances.

As it can be seen from Algorithm 3, we restrict the number of redeployments made per each round by allowing only idle ambulances, which could be returning to waiting locations or at the hospital, to be redeployed periodically at consecutive time intervals determined by t_r. It is known that allowing only idle ambulances to redeploy is both ambulance crew and fuel friendly [15].

Furthermore, when a redeployment event occurs the ambulances are assigned to new waiting locations using the Hungarian method, which is a combinatorial optimization algorithm. The idle ambulances are assigned to new base stations as follows: if the travel times are fixed, the shortest travel times between the current locations of the idle ambulances and new waiting locations are computed. Then, each idle ambulance is redeployed to the new location in a way that total travel time of all ambulances is minimized. Furthermore, if the travel times are time-dependent, then the traffic indices might change while the ambulances travel on the roads; therefore, the estimated travel times are used in ambulance assignment according to the current state of the system (i.e., traffic states on the roads and idle ambulance locations).

Chapter 6

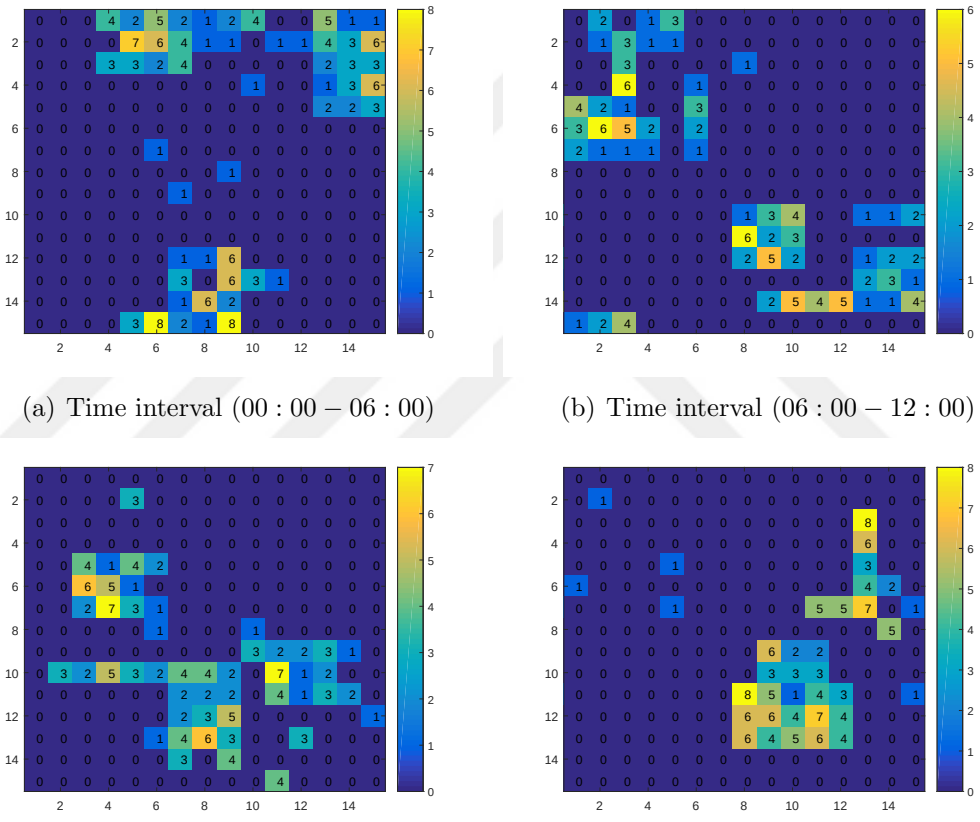
Illustrative Simulation 1: 15 x 15 Redeployment Network

In the simulations, we model a hypothetical city by using the set of parameters given in Table 6.1. As well as the fixed travel times on the roads, we also consider the case with time-dependent travel times as described in [8] where $\boldsymbol{\tau}_{i,j}(t)$ depends on the Markov traffic state in round t, which is described in Chapter 4.

In the simulations, we use a larger version of the ambulance redeployment network in Fig. 4.1 in order to model the city according to the parameter settings given in Table 6.1. We consider two cases: the first one is the context-free case where there is no traffic, i.e., fixed travel times with $x_{i,j}(t) = 0,\ \forall i, j \in \{1, ..., K\}$. In this case, the reward distribution depends only on the call distribution. The

Table 6.1: AMBULANCE REDEPLOYMENT PARAMETERS IN THE CITY

Parameter	Magnitude	Choice
λ	1/9.5 min	Call distribution parameter on a week day
K	225	Number of nodes
H	10	Number of hospitals in the region
$\gamma_{ij}(t)$		Travel time from node i to node j
d_i		Call fraction from node i
$\hat{d}_i(t)$		Estimation of d_i in round t



(a) Time interval $(00:00-06:00)$ (b) Time interval $(06:00-12:00)$

(c) Time interval $(12:00-18:00)$ (d) Time interval $(18:00-24:00)$

Figure 6.1: A redeployment scenario that consists of four different node likelihoods corresponding to the different time intervals in a day. Each node on the 15 by 15 redeployment network has a different likelihood of generating a call at a given round. The colors on the nodes indicate the number of calls generated from these nodes during the simulation.

second is the contextual case, where there are time-dependent traffic states and the reward distribution depends on both the call distribution and the travel times. We run the MAB algorithms in the context-free and the contextual cases, respectively. For comparison, we use the oracle optimization algorithm static MEXCLP (SMEXCLP) described in Chapter 3 and a dynamic extension of SMEXCLP that is introduced in [15] called the Dynamic MEXCLP (DMEXCLP) model.

In [15], the idle ambulances are dispatched to certain locations which results in the largest future marginal coverage according to the MEXCLP model. The call fraction d_i from each demand point is assumed to be known in advance and taken as the fraction of inhabitants that comprises this demand point. In practice, although it may be reasonable to approximate the call distributions with the number of inhabitants, we assume that d_i is not known for our city by the algorithms in the simulations. Therefore, we use the average of the previously observed samples up to and not including time t to estimate $\hat{d}_i(t)$. That is, $\hat{d}_i(t)$ is the sample mean estimate of d_i computed using $t-1$ samples using average sampling technique.

Similar to [15], we consider two performance metrics: the average arrival time and the level of coverage of the demand points (over T rounds). In a typical EMS system the success of a response is usually measured using a threshold for the ambulance arrival times. For instance, a particular redeployment is considered as successful only if a call is responded in no more than κ minutes. This threshold is set by the system operator, and is usually chosen based on various factors such as the road conditions, number of idle ambulances and traffic states [28]. We consider success rate under two different thresholds: 10 and 15 minutes. Therefore, the second performance metric is taken as the ratio of calls responded under 10 and 15 minutes to the total number of calls.

We use a 15 by 15 square network that consists of $K = 225$ nodes, whose structure is similar to the ambulance redeployment network given in Fig. 4.1 for $K = 9$. In this network, the nodes are equidistant to their neighbors, and the distance between two consecutive nodes (i.e., the node to the left, right, up, and down) is 2.5 kilometers, which covers a total area of 1225 km^2. The

maximum speed $V_{\max}$ of an ambulance on each edge is taken as 100 km/h when it is responding to a call or going to the hospital. The number of hospitals in the region is 10 and the nodes of the hospitals are chosen uniformly at random in each simulation run whose details are given below.

The calls are sampled according to 4 different Poisson binomial distributions $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$, and $\mathcal{C}_4$ with the parameters $\lambda_{\mathcal{C}_1}(t) = 0.10\lambda$, $\lambda_{\mathcal{C}_2}(t) = 0.20\lambda$, $\lambda_{\mathcal{C}_3}(t) = 0.35\lambda$, $\lambda_{\mathcal{C}_4}(t) = 0.35\lambda$, respectively where $\lambda = 1/9.5$ min is given in Table 6.1. This is because a day is divided into 4 equal time intervals, i.e. $(00:00-06:00)$, $(06:00-12:00)$, $(12:00-18:00)$, and $(18:00-24:00)$, and the calls in these time intervals are sampled from the distributions $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$, and $\mathcal{C}_4$, respectively.

Whether there is a call at round t in a given time interval of a day is determined according to the parameter of the call distribution of that time interval. We run the algorithms for 4 weeks of simulation time and the time intervals are assigned 300 rounds each for a given week so that there are in total 1200 rounds in a week. We perform a total of 100 simulation runs, each contains 4 weeks of simulation time. In each run, the call distributions change for each scenario and the location of hospitals are selected uniformly at random from the $K = 225$ nodes. The SMEXCLP algorithm knows $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$ and $\mathcal{C}_4$ beforehand and redeploys ambulances to the optimal N_t locations as decribed in Chapter 3. On the other hand, the MAB and DMEXCLP algorithms do not know $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$ and $\mathcal{C}_4$, and further they do not know when the distributions change from one to another, and how many rounds each time interval has. For example, if there are 40 calls sampled from $\mathcal{C}_3$ in time interval $(12:00-18:00)$, the SMEXCLP algorithm starts to compute the new optimal redeployment locations according to $\mathcal{C}_4$ after it observes these 40 calls from $\mathcal{C}_3$; whereas, the MAB and DMEXCLP algorithms do not know the call distribution changes after they observe the calls; hence, they continue their operation as if the calls are coming from the same distribution.

If a call arrives in a given round, then the node in which the call takes place is determined according to the nodes' likelihoods of generating a call. Four examples of different node likelihoods on the redeployment network are shown in Fig. 6.1 for the time intervals defined earlier. We call these four redeployment networks

with different node likelihoods *a redeployment scenario*. As mentioned earlier, we run the simulations for 4 weeks (i.e., a month) over 4 different redeployment scenarios and average the results of the algorithms over the weeks, and perform in total 100 4-week simulation runs.

The call completion time t_c is computed starting from the time the ambulance takes the patient and ending when the patient is delivered to the hospital. When travel times are time-dependent, t_c is also time-dependent. The redeployment event time t_r is taken to be 120 rounds so that the ambulances are redeployed in every 120 rounds which restricts the number of relocations made in a day.

As stated at the end of Chapter 4, the reward of an ambulance at node π_t that responds to a call $c(t)$ at round t is the inverse of the arrival time, i.e., $r_{t,\pi_t} = 1/(\gamma_{\pi_t,c(t)}(t))$, $\gamma_{\pi_t,c(t)}(t) \neq 0$, and $r_{t,\pi_t} = 1$, $\gamma_{\pi_t,c(t)}(t) = 0$. For instance, if an ambulance at node i responds to a call at node j in 5 minutes, then the reward of the ambulance is computed as $1/5$. Note that since the adjacent nodes are 2.5 kilometers apart and the maximum speed of an ambulance on an edge is 100 km/h, the minimum value of $\gamma_{\pi_t,c(t)}(t)$ when $\pi_t \neq c(t)$ is 1.5 minutes and it holds that $r_{t,\pi_t} < 1$ for all such π_t at round t, hence, the assumption made in Chapter 4 is satisfied. Furthermore, if the call originates from a node that has a deployed ambulance, i.e., $\pi_t = c(t)$ and $\gamma_{\pi_t,c(t)}(t) = 0$, then the algorithm achieves the maximum reward of 1.

6.1 Simulations for Fixed Travel Times (Context-free)

The simulation results with fixed travel times are given in Table 6.2, Fig. 6.2, Fig. 6.3, and Fig.6.4. Fig. 6.2 shows the average arrival times of the algorithms over 4 different redeployment scenarios, each consists of 4 weeks of simulation time, with respect to the ambulance number $N = 5, 10, 15, 20, 25, 30$. As seen in Fig. 6.2, the average arrival times of the MAB algorithms are lower than that of

the DMEXCLP algorithm.

Furthermore, the regrets of the algorithms with respect to the optimization oracle SMEXCLP over 4 weeks of simulation time when the ambulance number is $N = 20$ in four different scenarios are given in Fig. 6.3. This figure shows that the MAB algorithms rate of increase of the regret decreases over time; whereas the DMEXCLP algorithm has regret that linearly increases over time. This indicates that as there are more calls originating from the demand points, the MAB algorithms learn where to redeploy ambulances to minimize the arrival times; hence, start to perform similarly to the oracle SMEXCLP. In addition, the variations in the arrival times of the algorithms over 4 weeks of simulation time in four different scenarios when the ambulance number is $N = 20$ are shown in Fig. 6.4. The edges of the lines on the figure shows the maximum and minimum arrival times of the ambulances using the corresponding algorithm, and the red lines on the boxes indicate the average arrival times.

The algorithms are compared with each other according to the second performance metric (level of coverage of the demand points) in Table 6.2. The level of coverage percentage is computed for various number of ambulances N and the arrival time threshold κ.

The coverage percentage represents the percentage of the number of calls that are responded under 10 and 15 minutes, respectively. From Fig. 6.4 and the results in Table 6.2, it can be concluded that as the variance in the arrival times of the algorithms increases, their coverage percentage decreases. Therefore, multiple-arm Thompson Sampling (MaTS in the figure) and multiple-arm UCB1 (MaUCB1) perform similarly and achieve the best coverage percentages since they have the lowest variances and average arrival times, and the DMEXCLP has the lowest coverage percentages since its variance is relatively high when compared to the MAB algorithms.

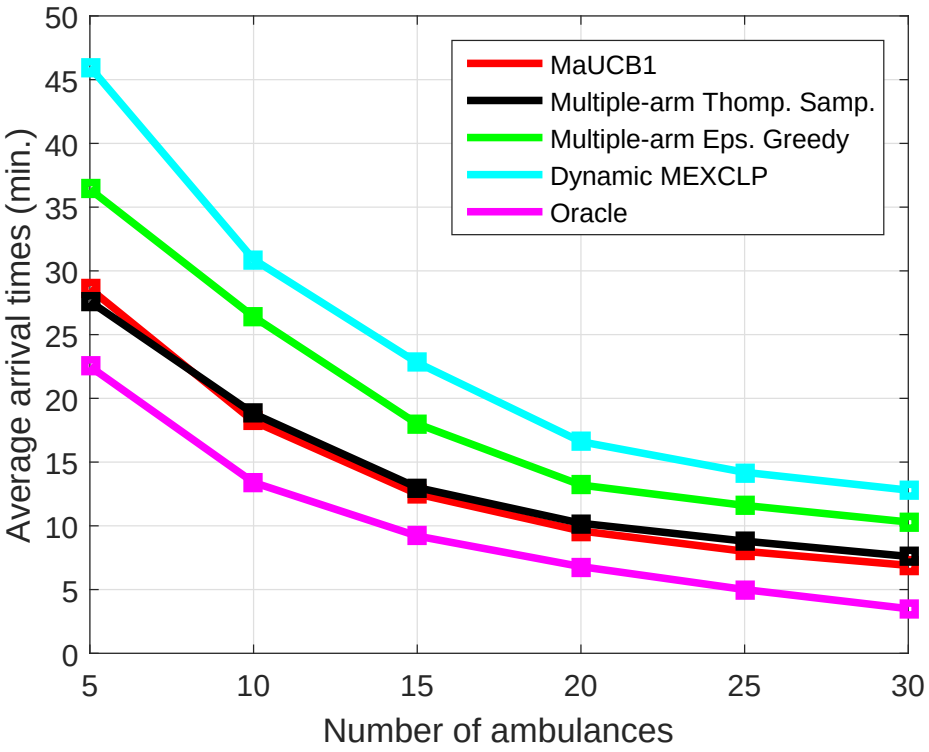


Figure 6.2: Average arrival times of the context-free MAB algorithms over 4 weeks of simulation time in 4 different redeployment scenarios with $t_r = 120$ for fixed travel times.

6.2 Simulations for Time-dependent Travel Times (Contextual)

The simulation results with time-dependent travel times are given in Table 6.3, Fig. 6.5, Fig. 6.6, and Fig. 6.7. Similar to the setup under fixed travel times, we run 4 different scenarios, each consists of 4 weeks of simulation time. For the traffic model, we use the Markov model given in Fig. 4.2 with the following

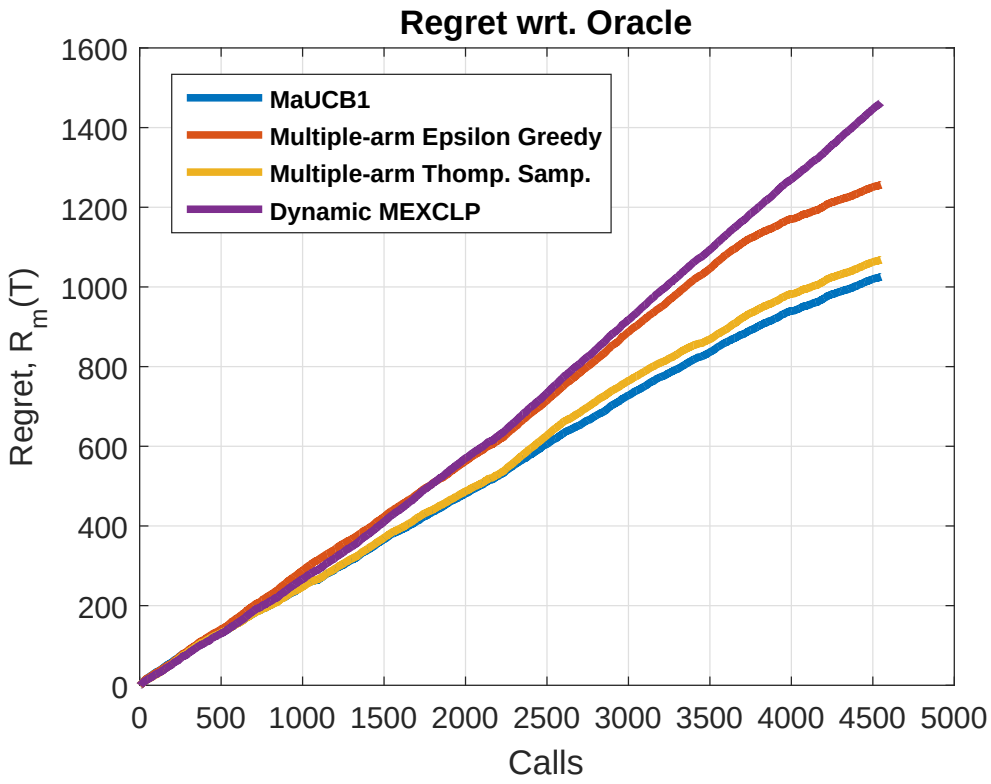


Figure 6.3: The regret of the MAB algorithms over a week of simulation time in 4 different redeployment scenarios rounds with $N = 20$ and $t_r = 120$ for fixed travel times.

transition probability matrix:

$$\mathcal{T} = \begin{array}{c} \\ s_0 \\ s_1 \\ s_2 \end{array} \begin{array}{c} \begin{array}{ccc} s_0 & s_1 & s_2 \end{array} \\ \begin{pmatrix} 0.62 & 0.30 & 0.08 \\ 0.18 & 0.64 & 0.18 \\ 0.12 & 0.18 & 0.70 \end{pmatrix} \end{array} \tag{6.1}$$

where s_i is the traffic state shown in Fig. 4.2. For instance, if the state is s_0 for edge (i, j), then $p_{s_0,s_0}(t) = 0.62$ and the next state for edge (i, j) will remain the same in the next round with probability 0.62 and go the states of light and heavily congested traffic with probabilities $p_{s_0,s_1}(t) = 0.30$ and $p_{s_0,s_2}(t) = 0.08$, respectively. Furthermore, once the traffic status is set, the speed of an ambulance on the roads is determined as follows:

- Scenario 1 (00 : 00 - 06 : 00): The traffic index $x_{i,j}$, $i, j \in \{1, ..., K\}$ is

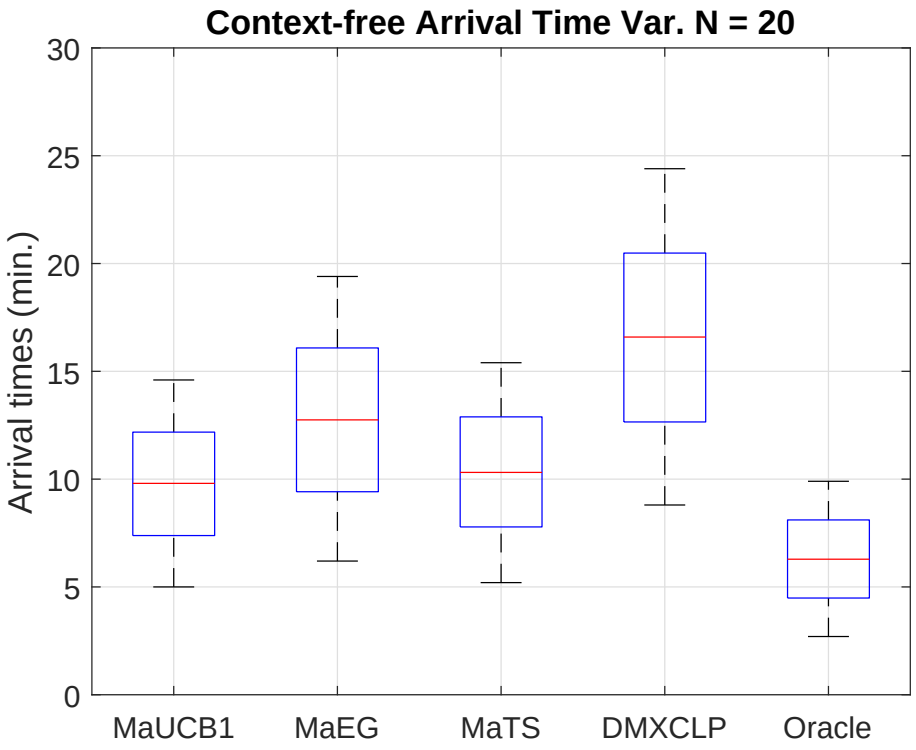


Figure 6.4: The variations in the arrival times of the MAB algorithms over 4 weeks of simulation time in 4 different redeployment scenarios with $N = 20$ and $t_r = 120$ for fixed travel times.

sampled from the uniform distribution $[0, 0.7)$ in state s_0, from $[0.7, 0.9]$ in state s_1, and from $(0.9, 1.0]$ in state s_2.

- Scenario 2 (06 : 00 - 12 : 00): The traffic index $x_{i,j},\ i, j \in \{1, ..., K\}$ is sampled from the uniform distribution $[0, 0.4)$ in state s_0, from $[0.4, 0.75]$ in state s_1, and from $(0.75, 1.0]$ in state s_2.

- Scenario 3 (12 : 00 - 18 : 00): The traffic index $x_{i,j},\ i, j \in \{1, ..., K\}$ is sampled from the uniform distribution $[0, 0.2)$ in state s_0, from $[0.2, 0.6]$ in state s_1, and from $(0.6, 1.0]$ in state s_2.

- Scenario 4 (18 : 00 - 24 : 00): The traffic index $x_{i,j},\ i, j \in \{1, ..., K\}$ is sampled from the uniform distribution $[0, 0.3)$ in state s_0, from $[0.3, 0.7]$ in state s_1, and from $(0.7, 1.0]$ in state s_2.

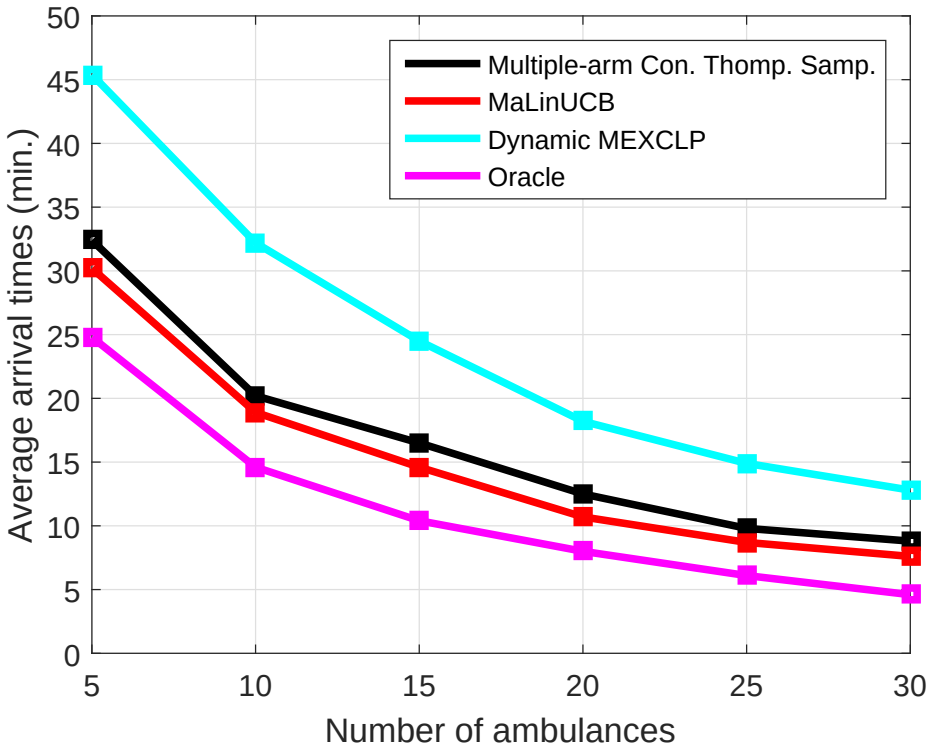


Figure 6.5: Average arrival times of the contextual MAB algorithms over 4 weeks of simulation time in 4 different redeployment scenarios with $t_r = 120$ for time-dependent travel times.

For instance, if the time of the day t is in Scenario 2, and the traffic intensity $x_{i,j}(t)$ on the road belongs to state s_0, the speed of the ambulance V_{amb} is sampled uniformly random from the interval $(0.6V_{max}, V_{max}]$. After the traffic indices are generated, the context and the arrival times on the paths are computed using (4.1) and (4.3).

Fig. 6.5 shows the average arrival times of the algorithms over 4 weeks of simulation time in 4 different redeployment scenarios. As seen from the figure, the contextual MAB algorithms have lower average arrival times than that of the DMEXCLP algorithm.

Fig. 6.6 shows the regrets of the contextual MAB and DMEXCLP algorithms with respect to the optimization oracle SMEXCLP over 4 weeks of simulation time when the ambulance number is $N = 20$ in four different scenarios. The

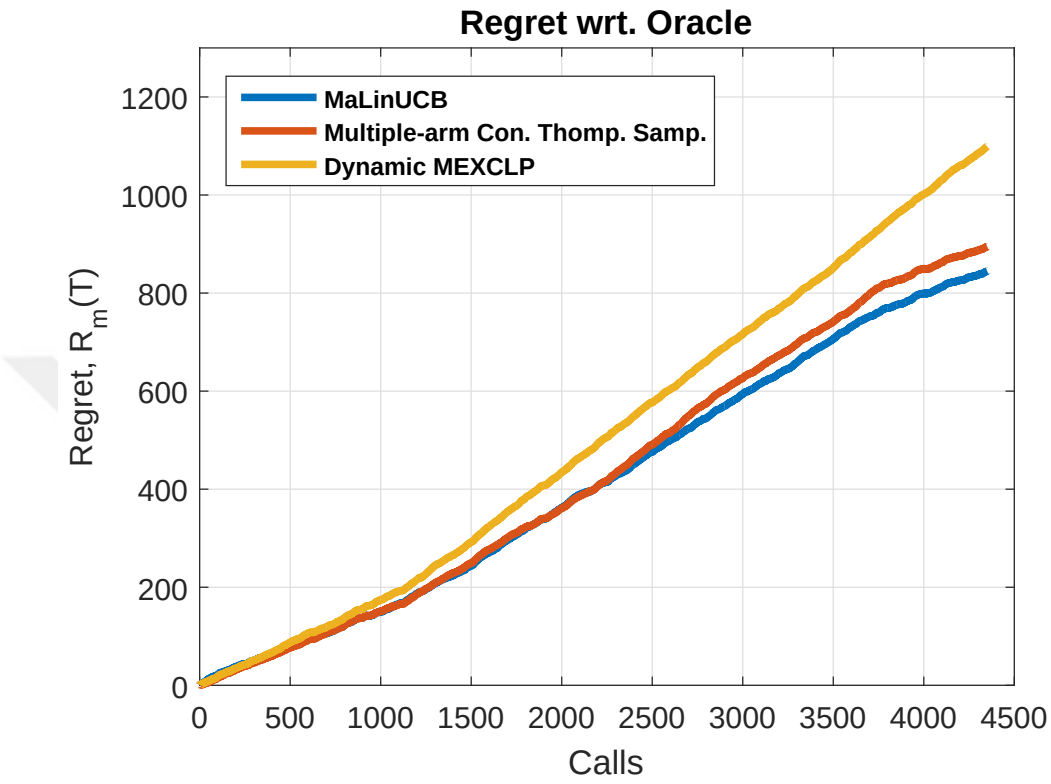


Figure 6.6: The regret of the contextual MAB algorithms over a week of simulation time in 4 different scenarios with $N = 20$ and $t_r = 120$ for time-dependent travel times.

regrets of the contextual MAB algorithms are again sublinear in the number of rounds; whereas the regret of DMEXCLP is linear. Furthermore, Fig. 6.7 shows the variations in the arrival times of the algorithms when the ambulance number is $N = 20$. The variance of the DMEXCLP algorithm is again greater than those of the contextual MAB algorithms. Furthermore, Table 6.3 shows the coverage percentage of the demand points in the cases of various ambulance numbers N and arrival time threshold κ. As seen, the contextual MAB algorithms perform better than the DMEXCLP algorithm in terms of the second performance criterion.

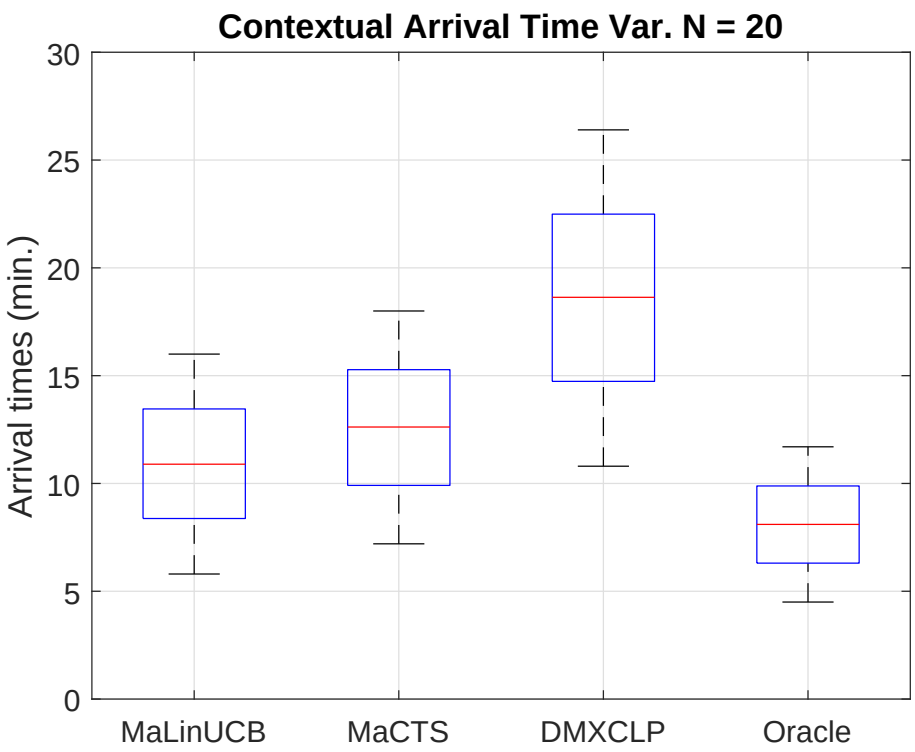


Figure 6.7: The variations in the arrival times of the contextual MAB algorithms over 4 weeks of simulation time in 4 different redeployment scenarios with $N = 20$ and $t_r = 120$ for time-dependent travel times.

Table 6.2: The coverage percentage of the demand points with respect to various ambulance numbers N and arrival time threshold κ over 4 weeks of simulation time in 4 different redeployment scenarios with fixed travel times

Ambulance number $N \rightarrow$	5		10		15	
Arrival time threshold $\kappa \rightarrow$ / Policy $\pi \downarrow$	10 (min)	15 (min)	10 (min)	15 (min)	10 (min)	15 (min)
MaUCB1	18.1%	32.5%	27.6%	48.4%	42.2%	66.3%
Multiple-arm ϵ-greedy	12.2%	24.1%	16.5%	39.2%	30.1%	52.4%
Multiple-arm Thomp. Samp.	21.5%	35.1%	26.7%	52.2%	44.2%	68.0%
Dynamic MEXCLP	16.2%	28.8%	18.4%	40.5%	30.2%	51.8%
Oracle (SMEXCLP)	24.0%	40.6%	32.5%	60.2%	52.8%	77.4%
Ambulance number $N \rightarrow$	20		25		30	
Arrival time threshold $\kappa \rightarrow$ / Policy $\pi \downarrow$	10 (min)	15 (min)	10 (min)	15 (min)	10 (min)	15 (min)
MaUCB1	60.3%	81.9%	69.3%	87.0%	74.9%	89.6%
Multiple-arm ϵ-greedy	46.8%	78.2%	57.4%	86.9%	65.9%	92.1%
Multiple-arm Thomp. Samp.	59.7%	83.7%	69.9%	89.6%	76.5%	92.6%
Dynamic MEXCLP	46.5%	74.4%	56.3%	83.9%	67.0%	90.0%
Oracle (SMEXCLP)	74.9%	84.4%	83.2%	89.7%	88.2%	93.3%

Table 6.3: The coverage percentage of the demand points with respect to various ambulance numbers N and arrival time threshold κ over 4 weeks of simulation time in 4 different redeployment scenarios under time-dependent travel times

Ambulance number $N \rightarrow$	5		10		15	
Arrival time threshold $\kappa \rightarrow$ / Policy $\pi \downarrow$	10 (min)	15 (min)	10 (min)	15 (min)	10 (min)	15 (min)
MaLinUCB	17.5%	32.1%	25.8%	47.6%	42.5%	66.5%
Multiple-arm Con. Thomp. Samp.	15.8%	30.2%	23.9%	46.2%	42.4%	65.0%
Dynamic MEXCLP	11.8%	22.6%	18.1%	39.5%	32.5%	56.2%
Oracle	19.2%	36.0%	32.4%	58.7%	52.3%	78.4%
Ambulance number $N \rightarrow$	20		25		30	
Arrival time threshold $\kappa \rightarrow$ / Policy $\pi \downarrow$	10 (min)	15 (min)	10 (min)	15 (min)	10 (min)	15 (min)
MaLinUCB	59.5%	80.9%	68.1%	85.5%	72.6%	88.6%
Multiple-arm Con. Thomp. Samp.	59.1%	78.5%	66.0%	85.0%	71.5%	87.0%
Dynamic MEXCLP	47.5%	74.2%	57.2%	81.0%	64.2%	86.7%
Oracle	72.2%	82.0%	83.0%	88.6%	86.7%	92.0%

Chapter 7

Risk-averse Multi-armed Bandits in Ambulance Redeployment

In this chapter, we investigate the concept of risk aversion in ambulance redeployment. The proposed MAB and CMAB algorithms in Chapter 3 focus only on minimizing the arrival times of the ambulances. However, this approach can leave some calls not responded within a reasonable time in favor of overall reduction in arrival times. In a typical EMS system, it is vital to respond to any call as soon as possible since every call has a high chance of being mortal and requires immediate response. Therefore, we propose a risk-averse MAB algorithm in which the worst-case scenarios (i.e., the calls that are not responded on time by the classical MAB algorithms) are also taken care of. Following the same convention in Chapter 3, we first define the new risk-averse MAB algorithm.

7.1 Multiple-arm Risk-Averse MVLCB (MaMVLCB)

The original Mean-Variance LCB (MVLCB) algorithm which selects a single arm at each round can be found in [25, 26]. Instead of constructing upper confidence

bounds on the expected rewards of the arms as UCB1 does (see Chapter 3), MVLCB constructs lower confidence bounds on the expected mean-variance of the arms. The mean-variance metric for arm a is defined as follows:

$$MV_a := \sigma_a^2 - \rho\mu_a \tag{7.1}$$

where σ_a is the standard deviation of the rewards when arm a is played, ρ is the risk coefficient, and μ_a is again the expected reward of arm a. We note that by adjusting the risk coefficient ρ we can place more importance on whether to minimize the variance in the rewards or to maximize the expected reward. In a sense, by combining the standard deviation and the expected rewards of arm a, we are able to both maximize the expected reward of arm a and minimize the variance in the rewards. In other words, we reduce the risk of playing an arm that has a high chance of generating a lower reward although the expected reward of the arm might be larger. In the context of ambulance redeployment, we minimize the risk of redeploying ambulances to the locations that are found to be close to most of the calls but far away from some calls. Instead, we redeploy ambulances to the locations that are relatively close to most of the calls but also not far away from the other calls by taking into account the variance in the arrival times.

We modify the original algorithm to allow for multiple-arm selection and call it the Multiple-arm MVLCB (MaMVLCB) algorithm. At round t, MaMVLCB computes an index for each ambulance location a based on the observations from that location as follows:

$$g_{t,a} := \hat{MV}_{t,a} + (\rho/10)\sqrt{\frac{2\log t}{n_{t,a}}} \tag{7.2}$$

where

$$\hat{MV}_{t,a} = \bar{\sigma}_{t,a}^2 - \rho\bar{\mu}_{t,a} \tag{7.3}$$

is the sample mean-variance of location a and computed by averaging the variance and mean of the reciprocals of the arrival times of the ambulances that are located at a and dispatched to a call up to round t. t is the current round, $n_{t,a}$ is the number of times an ambulance has been placed at location a and dispatched to a call up to round t. At each round, MaMVLCB redeploys N_t (i.e., the number of idle ambulances) ambulances. These redeployments are made to the

locations with the N_t *lowest* indices (the reason of choosing the lowest indices is because we want to minimize the variance and maximize the mean), denoted by $\pi_{t,1}, \pi_{t,2}, ..., \pi_{t,N_t}$:

$$\begin{aligned}
\pi_{t,1} &= \arg\min_{a \in \mathcal{A}} g_{t,a}, \\
\pi_{t,2} &= \arg\min_{a \in \mathcal{A} \setminus \{\pi_{t,1}\}} g_{t,a}, \\
&\vdots \\
\pi_{t,N_t} &= \arg\min_{a \in \mathcal{A} \setminus \{\pi_{t,i}\}_{i=1}^{N_t-1}} g_{t,a}.
\end{aligned} \tag{7.4}$$

In other words, we sequentially compute the single best location that has the lowest index $g_{t,a}$ among the locations in $\mathcal{A}$, which is $\pi_{t,1}$, exclude this location from $\mathcal{A}$, and then compute again the best location that has the lowest index among the locations in $\mathcal{A} \setminus \{\pi_{t,1}\}$. We proceed this way until all N_t locations are selected for ambulance redeployment.

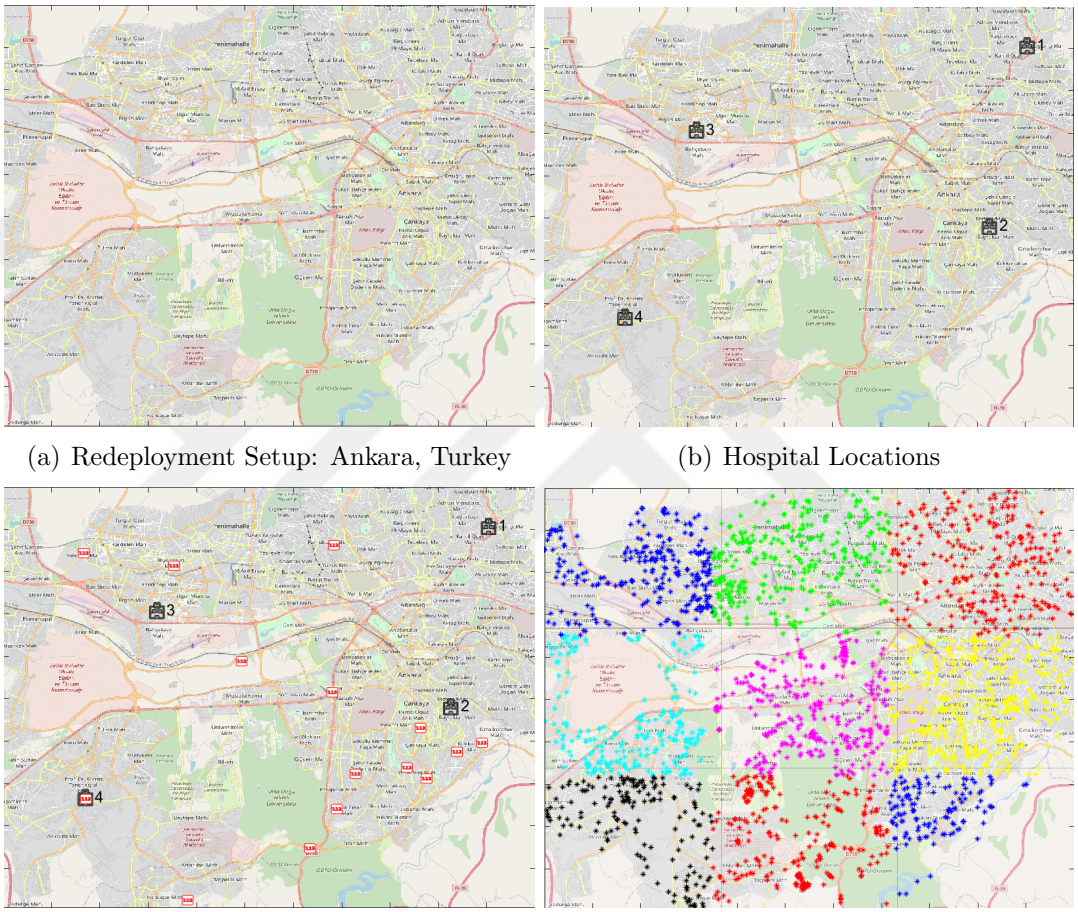
As a call originates from a location, the closest ambulance $\pi_{t,n},\ n \in \{1, ..., N_t\}$ is dispatched to the call and its reward $r_{t,\pi_{t,n}}$ is observed at round t. $r_{t,\pi_{t,n}}$ is used in calculating the regret in (3.3). Then, the sample mean reward $\bar{r}_{t,\pi_{t,n}}$, the sample variance $\bar{\sigma}^2_{t,a}$, and $n_{\pi_{t,n}}$ are updated for the next round in which (7.2) is computed again for each location. The last term on the right-hand side of (7.2) is again the exploration term. The exploration term measures the uncertainty in ambulance redeployment and has greater values when t is small and shrinks when t increases. It enables MaMVLCB to occasionally select locations that are rarely selected before, discover locations with high rewards, and avoid getting stuck at sub-optimal locations. The term $\rho/10$ is used for scaling the exploration term.

The computational complexity of the MVLCB algorithm for single-arm selection is given in [26]. In addition to the single-arm complexity, sorting and selecting N_t locations out of K locations in (7.4) also introduce in big O notation a complexity of $\mathcal{O}(N_t K \log(K))$.

7.2 Illustrative Example: A Realistic Ambulance Redeployment in Ankara

In this section, we describe the simulation setup that we use to evaluate the performance of the MaMVLCB algorithm as well as the other MAB and CMAB algorithms in terms of the expected arrival times and the coverage of the demand points (i.e., the percentage of calls responded under a certain time to the total number of calls). Instead of using a hypothetical city as we do in Chapter 6, we conduct more realistic simulations by modeling the city of Ankara in Turkey. For this purpose, we use the third-party OpenStreetMap (OSM) application. OSM allows users to download a piece of map that is in XML format. After parsing this format, we get the information on nodes, their connectivity to each other, and the number of roads that consist of these nodes. When it is compared with the redeployment network in Chapter 6, we increase the size of the network and use $K = 2400$ nodes to model the city of Ankara, Turkey. These nodes are again connected to each other as shown in Fig. 4.1. Table 7.1 shows the parameter settings for the city. As we do in Chapter 6, we consider both the fixed (i.e., context-free) and time-dependent (i.e., contextual) travel times on the roads. In the first case, there is no traffic, i.e., $x_{i,j}(t) = 0,\ \forall i, j \in 1, ..., K$. In this case, the reward distribution depends only on the call distribution. However, in the second case where the travel times are time-dependent and determined according to the Markov traffic model described in Chapter 4, the reward distribution depends on both the call distributions and travel times. The details of the Markov traffic model is given in the following section. We run the MAB and risk-averse MAB (i.e., MaMVLCB) algorithms in the context-free case, and the contextual MAB algorithms in the contextual case, respectively. Similar to Chapter 6, we use the oracle optimization algorithm SMEXCLP and the dynamic redeployment model DMEXCLP for comparison. Instead of using the true d_i values for the DMEXCLP, we use their estimates $\hat{d}_i(t)$ in our simulations (e.g., see the discussion in Chapter 6).

We again consider two performance metrics: the average arrival time and the



(a) Redeployment Setup: Ankara, Turkey

(b) Hospital Locations

(c) Example Ambulance Locations ($N = 15$)

(d) Nodes on Redeployment Network

Figure 7.1: The city of Ankara, Turkey modeled using the OpenStreetMap (OSM) application: (a) shows the redeployment setup, (b) shows the 4 hospital locations used in the simulations, (c) shows the example ambulance locations that are deployed on 15 different nodes, and (d) shows the locations of 2400 nodes that are connected to each other as shown in Fig. 4.1 where $K = 2400$. The map is divided into 9 different regions (numbered from left to right and top to bottom) such that each region i generates calls according to its own binomial Poisson distribution C_i (e.g., C_1 is the top left and C_9 is the bottom right region) and is independent from the other regions. The nodes that have different colors belong to different regions.

level of coverage of the demand points (over T rounds). In a typical EMS system the success of a response is usually measured using a threshold for the ambulance arrival times. For instance, a particular redeployment is considered as successful only if a call is responded in no more than κ minutes. This threshold is set by the system operator, and is usually chosen based on various factors such as the road conditions, number of idle ambulances and traffic states [28]. We consider success rate under two different thresholds: 10 and 15 minutes. Therefore, the second performance metric is taken as the ratio of calls responded under 10 and 15 minutes to the total number of calls.

As opposed to the 15×15 square redeployment network in Chapter 6, the nodes in the new map are now not equidistant to each other. Instead, the real locations and road connections from Ankara are used in the new simulations. Our Ankara map covers a total area of 2200 km^2. The maximum speed of $V_{\max}$ of an ambulance on each road is taken as 100 km/h when $x_{i,j}(t) = 0,\ \forall i, j = \{1, ..., 2400\}$ and when the ambulance is responding to a call or going to the hospital. The number of hospitals in the region is 4 as shown in Fig. 7.1, and unlike the random hospital locations in Chapter 6, their locations are fixed and do not change throughout the simulation runs.

To simplify the simulations, instead of dividing a day into four different time intervals as we do in Chapter 6, we now consider only a single time interval so that the call distributions are stationary from one day to another. But instead, we partition to map into 9 different regions as shown in Fig. 7.1.d numbered from left to right and top to bottom. Each region now has their own Poisson binomial distribution $\mathcal{C}_i,\ i = \{1, ..., 9\}$ with their parameter $\lambda_{\mathcal{C}_i},\ i = \{1, ..., 9\}$. It is assumed that each region generates a call independently from each other and its distribution parameter is proportional to the number of nodes that belong to this region, i.e., $\lambda_{\mathcal{C}_1} = 0.0111,\ \lambda_{\mathcal{C}_2} = 0.0136,\ \lambda_{\mathcal{C}_3} = 0.0103,\ \lambda_{\mathcal{C}_4} = 0.0103,\ \lambda_{\mathcal{C}_5} = 0.0137,\ \lambda_{\mathcal{C}_6} = 0.0182,\ \lambda_{\mathcal{C}_7} = 0.0068,\ \lambda_{\mathcal{C}_8} = 0.0109,\ \lambda_{\mathcal{C}_9} = 0.0052$. Therefore, $\lambda_{\mathcal{C}_1} + ... + \lambda_{\mathcal{C}_9} = 1/10$ since the call distribution parameter λ is $1/10$ min in a day.

Whether there is a call at round t in a given region is determined according to the parameter of the call distribution of that region. We run the algorithms for

Table 7.1: AMBULANCE REDEPLOYMENT PARAMETERS IN ANKARA, TURKEY

Parameter	Magnitude	Choice
λ	1/10 min	Call distribution parameter on a week day
K	2400	Number of nodes
H	4	Number of hospitals in the region
$\gamma_{ij}(t)$		Travel time from node i to node j
d_i		Call fraction from node i
$\hat{d_i}(t)$		Estimation of d_i in round t

12 weeks of simulation time and 144 calls are generated in each day; thus, there are a total of 720 calls in a week (we only consider week days in our simulations) and a total of 8640 calls during a single simulation run. We perform a total of 100 simulation runs, each contains 12 weeks of simulation time. In each run, N nodes are selected uniformly at random from the $K = 2400$ nodes and ambulances are deployed initially at these nodes. The SMEXCLP algorithm knows $\mathcal{C}_1, ..., \mathcal{C}_9$ beforehand and redeploys ambulances to the optimal N_t locations as decribed in Chapter 3. On the other hand, the MAB and DMEXCLP algorithms do not know $\mathcal{C}_1, ..., \mathcal{C}_9$ as before.

If a call arrives in a given region, then the node in which the call originates is selected uniformly at random from the nodes in this region. For example, the first region with Poisson binomial distribution C_1 has 266 nodes so that when a call is generated in this region, a given node out of these 266 nodes is selected with probability 1/266.

The call completion time t_c is computed starting from the time the ambulance takes the patient and ending when the patient is delivered to the hospital. When travel times are time-dependent, t_c is also time-dependent. The redeployment event time t_r is taken to be 120 rounds so that the ambulances are redeployed in every 120 rounds which restricts the number of relocations made in a day as before.

7.2.1 Simulations for Fixed Travel Times (Context-free)

The simulation results with fixed travel times are given in Table 7.2, Fig. 7.2, and Fig.7.3. Fig. 7.2 shows the average arrival times of the algorithms for 12 weeks of simulation time over 100 simulation runs, with respect to the ambulance number $N = 5, 10, 15, 20, 25, 30$. The risk coefficient ρ is selected as 0.6. As seen in Fig. 7.2, the average arrival times of the MAB algorithms are lower than that of the DMEXCLP algorithm.

In addition, the variations in the arrival times of the algorithms over 12 weeks of simulation time over 100 simulation runs when the ambulance number is $N = 20$ are shown in Fig. 7.3. The edges of the lines on the figure shows the maximum and minimum arrival times of the ambulances using the corresponding algorithm, and the red lines indicate the average arrival times.

The algorithms are compared with each other according to the second performance metric (level of coverage of the demand points) in Table 7.2. The level of coverage percentage is computed for various number of ambulances N and the arrival time threshold κ.

The coverage percentage represents the percentage of the number of calls that are responded under 10 and 15 minutes, respectively. From Fig. 7.3 and the results in Table 7.2, it can be concluded that as the variance in the arrival times of the algorithms increases, their coverage percentage decreases. Therefore, multiple-arm Thompson Sampling (MaTS in the figure) and multiple-arm UCB1 (MaUCB1) perform similarly and multiple-arm MVLCB (MaMVLCB) achieves the best coverage percentages since it has the lowest variance, and the DMEXCLP has the lowest coverage percentages since its variance is relatively high when compared to the MAB algorithms.

One important thing we note here is that from Fig. 7.2, Fig. 7.3, and Table 7.2, it is clear that although MaMVLCB performs worst than MaUCB1 and MaTS in terms of the average arrival times, it performs better than in terms of the coverage of the demand points. It is because by selecting the risk coefficient

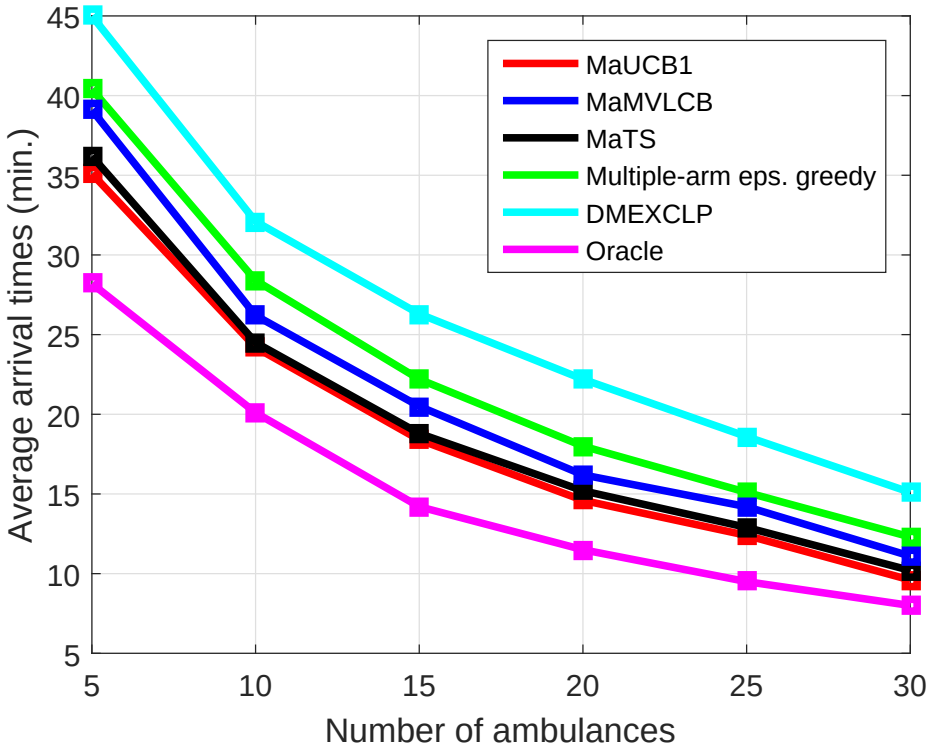


Figure 7.2: Average arrival times of the context-free and risk-averse MAB algorithms over 12 weeks of simulation time with $t_r = 120$ and $\rho = 0.6$ for fixed travel times.

$\rho = 0.6$ we put more emphasis on the variance minimization in arrival times over the expected arrival time minimization; hence, more calls are responded under 10 and 15 minutes. For this reason, by changing the risk coefficient accordingly, we can decide between whether to minimize the average arrival times or to respond to more calls on time. This trade-off between the two objectives allows us to design our algorithms according to what the EMS system requires.

7.2.2 Simulations for Time-dependent Travel Times (Contextual)

The simulation results with time-dependent travel times are given in Table 7.3, Fig. 7.4, and Fig. 7.5. Similar to the setup under fixed travel times, we run 12

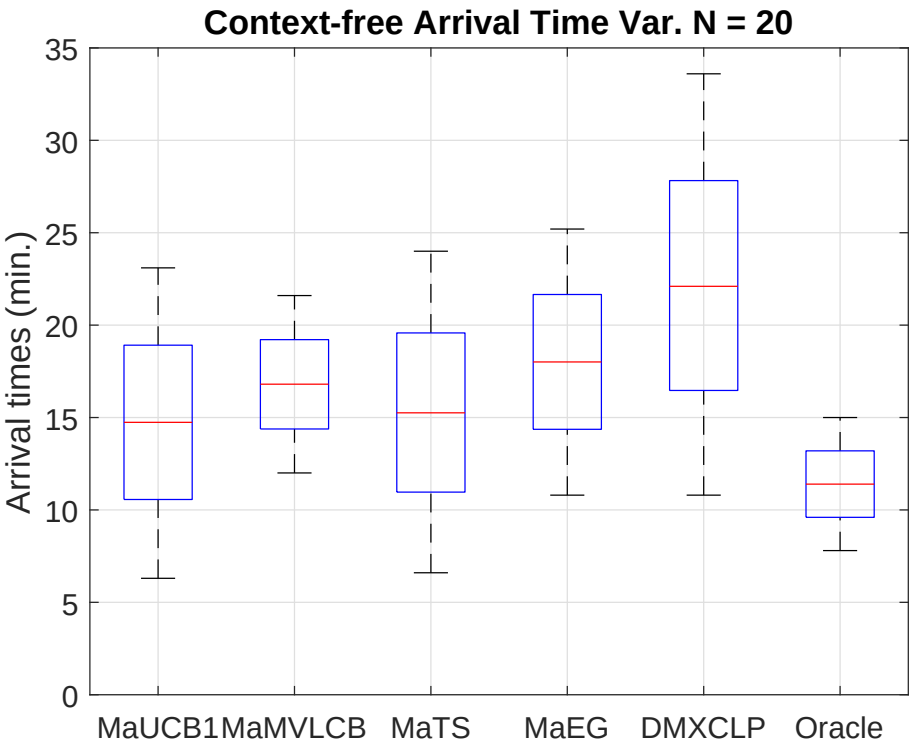


Figure 7.3: The variations in the arrival times of the context-free and risk-averse MAB algorithms over 12 weeks of simulation time with $N = 20$, $t_r = 120$, and $\rho = 0.6$ for fixed travel times.

weeks of simulation over 100 simulation runs. For the traffic model, we use the Markov model given in Fig. 4.2 with the following transition probability matrix:

$$\mathcal{T} = \begin{array}{c} \\ s_0 \\ s_1 \\ s_2 \end{array} \begin{array}{c} \begin{array}{ccc} s_0 & s_1 & s_2 \end{array} \\ \begin{pmatrix} 0.62 & 0.30 & 0.08 \\ 0.18 & 0.64 & 0.18 \\ 0.12 & 0.18 & 0.70 \end{pmatrix} \end{array} \tag{7.5}$$

where s_i is the traffic state shown in Fig. 4.2. For instance, if the state is s_0 for edge (i, j), then $p_{s_0,s_0}(t) = 0.62$ and the next state for edge (i, j) will remain the same in the next round with probability 0.62 and go the states of light and heavily congested traffic with probabilities $p_{s_0,s_1}(t) = 0.30$ and $p_{s_0,s_2}(t) = 0.08$, respectively. Furthermore, once the traffic status is set, the speed of an ambulance on the roads is determined as follows: The traffic index $x_{i,j}$, $i, j \in \{1, ..., K\}$ is

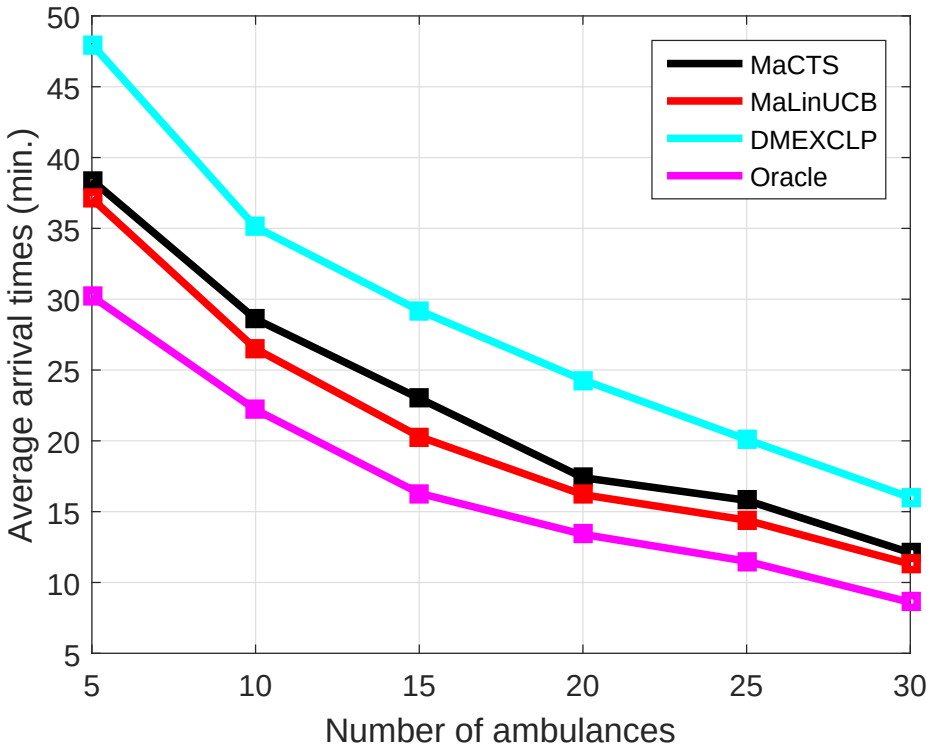


Figure 7.4: Average arrival times of the contextual MAB algorithms over 12 weeks of simulation time with $t_r = 120$ for time-dependent travel times.

sampled from the uniform distribution $[0, 0.6)$ in state s_0, from $[0.6, 0.9]$ in state s_1, and from $(0.9, 1.0]$ in state s_2. For instance, if the traffic intensity $x_{i,j}(t)$ on the road belongs to state s_2, the speed of the ambulance V_{amb} is sampled uniformly random from the interval $[0, 0.1V_{max})$. After the traffic indices are generated, the context and the arrival times on the paths are computed using (4.1) and (4.3).

Fig. 7.4 shows the average arrival times of the algorithms over 100 simulation runs, each consisting of 12 weeks of simulation time. As seen from the figure, the contextual MAB algorithms have lower average arrival times than that of the DMEXCLP algorithm.

Furthermore, Fig. 7.5 shows the variations in the arrival times of the algorithms when the ambulance number is $N = 20$. The variance of the DMEXCLP

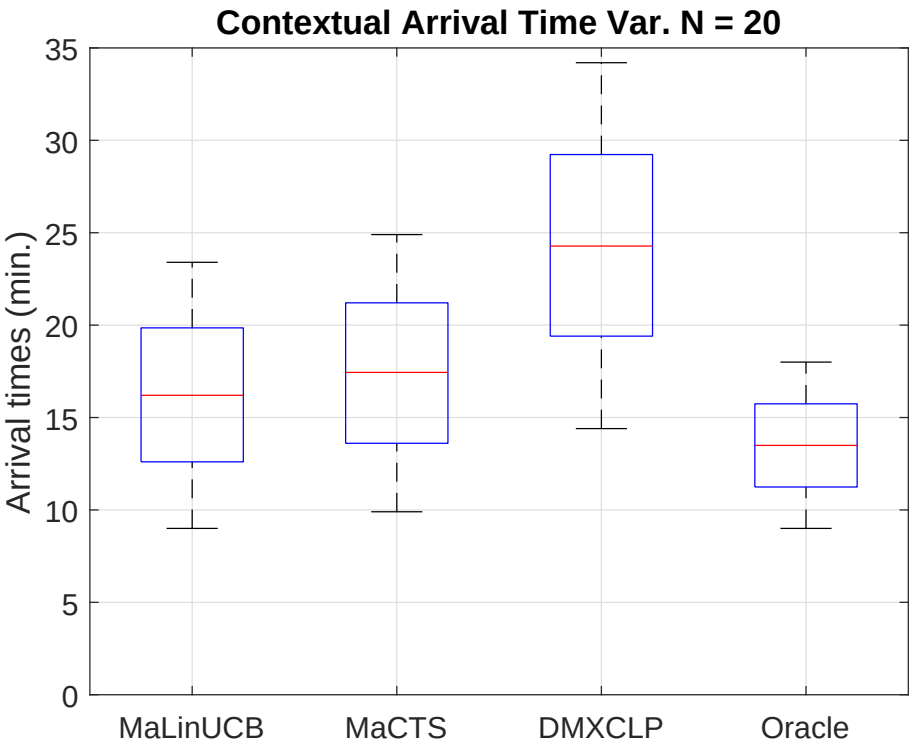


Figure 7.5: The variations in the arrival times of the contextual MAB algorithms over 12 weeks of simulation time with $N = 20$ and $t_r = 120$ for time-dependent travel times.

algorithm is again greater than those of the contextual MAB algorithms. Furthermore, Table 7.3 shows the coverage percentage of the demand points in the cases of various ambulance numbers N and arrival time threshold κ. As seen, the contextual MAB algorithms perform better than the DMEXCLP algorithm in terms of the second performance criterion.

Table 7.2: The coverage percentage of the demand points with respect to various ambulance numbers N and arrival time threshold κ over 12 weeks of simulation time for the city of Ankara with fixed travel times

Ambulance number $N \rightarrow$	5		10		15	
Arrival time threshold $\kappa \rightarrow$ / Policy $\pi \downarrow$	10 (min)	15 (min)	10 (min)	15 (min)	10 (min)	15 (min)
MaUCB1	6.5%	14.5%	12.6%	26.6%	23.1%	43.8%
MaMVLCB	7.5%	16.8%	15.3%	29.4%	25.1%	46.8%
Multiple-arm ϵ-greedy	3.8%	10.2%	8.2%	22.5%	18.8%	38.3%
Multiple-arm Thomp. Samp.	5.6%	12.2%	10.8%	24.2%	22.2%	42.2%
Dynamic MEXCLP	2.5%	6.8%	5.1%	18.1%	14.2%	34.6%
Oracle (SMEXCLP)	10.4%	24.5%	22.6%	42.8%	42.1%	53.4%
Ambulance number $N \rightarrow$	20		25		30	
Arrival time threshold $\kappa \rightarrow$ / Policy $\pi \downarrow$	10 (min)	15 (min)	10 (min)	15 (min)	10 (min)	15 (min)
MaUCB1	41.5%	52.9%	47.3%	65.4%	56.8%	72.6%
MaMVLCB	44.2%	55.3%	49.2%	69.7%	59.5%	76.2%
Multiple-arm ϵ-greedy	35.0%	46.1%	40.2%	58.6%	48.6%	65.9%
Multiple-arm Thomp. Samp.	40.2%	51.8%	46.5%	64.6%	55.2%	71.8%
Dynamic MEXCLP	22.6%	38.8%	32.3%	44.2%	40.1%	48.1%
Oracle (SMEXCLP)	48.6%	66.8%	59.0%	78.5%	67.2%	84.7%

Table 7.3: The coverage percentage of the demand points with respect to various ambulance numbers N and arrival time threshold κ over 12 weeks of simulation time under time-dependent travel times

Ambulance number $N \rightarrow$	5		10		15	
Arrival time threshold $\kappa \rightarrow$ / Policy $\pi \downarrow$	10 (min)	15 (min)	10 (min)	15 (min)	10 (min)	15 (min)
MaLinUCB	5.4%	11.8%	10.2%	23.6%	21.7%	41.5%
Multiple-arm Con. Thomp. Samp.	4.8%	10.9%	9.6%	23.0%	20.8%	40.8%
Dynamic MEXCLP	2.0%	5.4%	4.2%	15.8%	12.1%	31.8%
Oracle	9.2%	18.2%	16.4%	42.8%	36.1%	48.4%
Ambulance number $N \rightarrow$	20		25		30	
Arrival time threshold $\kappa \rightarrow$ / Policy $\pi \downarrow$	10 (min)	15 (min)	10 (min)	15 (min)	10 (min)	15 (min)
MaLinUCB	40.0%	51.5%	46.2%	64.2%	54.9%	71.5%
Multiple-arm Con. Thomp. Samp.	38.4%	50.2%	45.0%	61.9%	52.5%	69.4%
Dynamic MEXCLP	20.8%	36.4%	30.1%	41.8%	38.2%	46.8%
Oracle	45.8%	58.9%	53.5%	72.6%	63.8%	81.8%

Chapter 8

Conclusion and Future Works

In this thesis, we presented a new learning-based approach for the ambulance redeployment problem. First, we cast the problem as a MAB problem, and then, considered the cases in which the travel-times are fixed (i.e., context-free case) and time-dependent with a Markov time-dependent model (i.e., contextual case), and used a graph-based network for ambulance redeployment. Furthermore, we developed a discrete-time data-driven simulator to evaluate the performances of the algorithms.

In terms of minimizing the average arrival time (i.e., the first objective) and maximizing the coverage of the demand points (i.e., the second objective), we showed that the MAB algorithms perform better than the dynamic redeployment method DMEXCLP introduced in [15] under fixed and time-dependent travel times. We also showed that they perform close to the static allocation method SMEXCLP which is an optimization oracle that knows all call and traffic distributions and when these distributions change beforehand. The importance of using MAB based methods stems from the fact that in real-world, it is impossible to have such an oracle who knows all of the uncertainties beforehand. These unpredictable factors can easily make the solution of the redeployment problem intractable, and make the conventional optimization based methods inadequate, as justified by our simulations.

Furthermore, we investigated risk aversion in ambulance redeployment and proposed a rik-averse MAB algorithm. We modeled the city of Ankara, Turkey and conducted realistic simulations on this model. We showed that taking less risk by adjusting the risk coefficient resulted in serving more calls on time (i.e., increasing the coverage of demand points), but also resulted in increase in the average arrival times. By utilizing this trade-off between the two objectives, a more reliable EMS system can be designed in real life.

For future work, we plan to work on combinatorial multi-armed bandit (CMAB) problems in ambulance redeployment. As opposed to the one-arm selection setting in the classical MAB problem, multiple ambulances are selected for redeployment in our problem. Therefore, a need for combinatorial approach is inherent in the ambulance redeployment problem. Instead of using the greedy redeployment approach we propose in Chapter 3, combinatorial MABs can be utilized in the solution of the problem. In [45, 46], they propose a combinatorial multi-armed bandit approach where when a subset of arms is selected, other arms are probabilistically triggered. In the context of ambulance redeployment, when an ambulance is dispatched from a node (i.e., a bandit arm), it also travels to other nodes on route to the call. These nodes are selected according to the traffic states at time of the dispatch, hence in a sense they are probabilistically triggered. Therefore, in addition to observing the arrival time of the ambulance from the node from which it is dispatched, it is also possible to observe the arrival times from the nodes on route to the call. Therefore, although the ambulance is dispatched from a single node, other nodes are also explored in the process. This property of the CMAB problem with probabilistically triggered arms can make the learning in ambulance redeployment faster and more accurate.

Bibliography

[1] M. Gendreau, G. Laporte, and F. Semet, "A dynamic model and parallel tabu search heuristic for real-time ambulance relocation," *Parallel Comput.*, vol. 27, no. 12, pp. 1641–1653, 2001.

[2] R. Nair and E. Miller-Hooks, "Fleet management for vehicle sharing operations," *Transportation Sci.*, vol. 45, no. 4, pp. 524–540, 2011.

[3] O. Zhang, A. Mason, and A. Philpott, "Simulation and optimisation for ambulance logistics and relocation," *Presentation INFORMS Conf.*, 2008.

[4] M. S. Maxwell, S. G. Henderson, and H. Topaloglu, "Ambulance redeployment: An approximate dynamic programming approach," *Winter Sim. Conf.*, pp. 1850–1860, 2009.

[5] S. G. Henderson, "Operations research tools for addressing current challenges in emergency medical services," *Wiley Encyclopedia Op. Res. and Mgmt. Sci.*, 2011.

[6] Y. Yue, L. Marla, and R. Krishnan, "An efficient simulation-based approach to ambulance fleet allocation and dynamic redeployment," *AAAI*, 2012.

[7] P. L. van den Berg and K. Aardal, "Time-dependent mexclp with start-up and relocation cost," *EU J. Op. Res.*, vol. 242, pp. 383–389, 2015.

[8] V. Schmid and K. F. Doerner, "Ambulance location and relocation problems with time-dependent travel times," *EU J. Op. Res.*, vol. 207, pp. 1293–1303, 2010.

[9] D. Degel, L. Wiesche, S. Rachuba, and B. Werners, "Time-dependent ambulance allocation considering data-driven empirically required coverage," *Health Care Mgmt. Sci.*, pp. 444–458, 2015.

[10] V. Bélanger, Y. Kergosien, A. Ruiz, and P. Soriano, "An empirical comparison of relocation strategies in real-time ambulance fleet management," *Computers Industrial Eng.*, pp. 216–229, 2016.

[11] T. van Barneveld, S. Bhulai, and R. D. van der Mei, "The effect of ambulance relocations on the performance of ambulance service providers," *EU J. Op. Res.*, vol. 252, pp. 257–269, 2016.

[12] T. van Barneveld, "The minimum expected penalty relocation problem for the computation of compliance tables for ambulance vehicles," *INFORMS J. Comput.*, vol. 28, pp. 370–384, 2016.

[13] J. Naoum-Sawaya and S. Elhedhli, "A stochastic optimization model for real-time ambulance redeployment," *Comput. Op. Res.*, vol. 40, pp. 1972–1978, 2013.

[14] V. Schmid, "Solving the dynamic ambulance relocation and dispatching problem using approximate dynamic programming," *EU J. Op. Res.*, vol. 219, pp. 611–621, 2012.

[15] C. J. Jagtenberg, S. Bhulai, and R. D. van der Mei, "An efficient heuristic for real-time ambulance redeployment," *Op. Res. Health Care*, vol. 4, pp. 27–35, 2015.

[16] T. van Barneveld, C. Jagtenberg, S. Bhulai, and R. D. van der Mei, "Real-time ambulance relocation: Assessing real-time redeployment strategies for ambulance relocation," *Socio-Econ. Plan. Sci.*, vol. 62, pp. 129–42, 2016.

[17] M. S. A. A. Hofmeijer, "Dynamic ambulance redeployment with uncertain driving times," *Bachelor thesis, Erasmus Uni. Rotterdam*, 2016.

[18] M. Maleki, N. Majlesinasab, and M. M. Sepehri, "Two new models for redeployment of ambulances," *Computers Industrial Eng.*, vol. 78, pp. 271–84, 2014.

[19] M. Gendreau, G. Laporte, and F. Semet, "The maximal expected coverage relocation problem for emergency vechicles," *J. Op. Res. Society*, vol. 57, pp. 22–28, 2006.

[20] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3/4, pp. 285–294, 1933.

[21] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Adv. Appl. Math.*, vol. 6, no. 1, pp. 4–22, 1985.

[22] C. Tekin, S. Zhang, and M. van der Schaar, "Distributed online learning in social recommender systems," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 4, pp. 638–652, 2014.

[23] C. Tekin, O. Atan, and M. van der Schaar, "Discover the expert: Context-adaptive expert selection for medical diagnosis," *IEEE Trans. Emerging Topics Comput.*, vol. 3, no. 2, pp. 220–234, 2015.

[24] L. Song, C. Tekin, and M. van der Schaar, "Online learning in large-scale contextual recommender systems," *IEEE Trans. Services Comput.*, vol. 9, no. 3, pp. 433–445, 2016.

[25] A. Sani, A. Lazaric, and R. Munos, "Risk-aversion in multi-armed bandits," *Adv. Neural Inf. Process. Sys.*, pp. 3275–3283, 2012.

[26] S. Vakili and Q. Zhao, "Risk-averse multi-armed bandit problems under mean-variance measure," *IEEE J. Sel. Topics Signal Process.*, pp. 1093–1111, 2016.

[27] C. Toregas, R. Swain, C. ReVelle, and L. Bergman, "The location of emergency service facilities," *Op. Res.*, vol. 19, no. 6, pp. 1363–1373, 1971.

[28] R. Church and C. R. Velle, "The maximal covering location problem," *Papers Reg. Sci.*, vol. 32, no. 1, pp. 101–118, 1974.

[29] D. J. Eaton, M. S. Daskin, D. Simmons, B. Bulloch, and G. Jansma, "Determining emergency medical service vehicle deployment in Austin, Texas," *Interfaces*, vol. 15, no. 1, pp. 96–108, 1985.

[30] M. S. Daskin and E. H. Stern, "A hierarchical objective set covering model for emergency medical service vehicle deployment," *Transportation Sci.*, vol. 15, no. 2, pp. 137–152, 1981.

[31] K. Hogan and C. ReVelle, "Concepts and applications of backup coverage," *Mgmt. Sci.*, vol. 32, no. 11, pp. 1434–1444, 1986.

[32] M. Gendreau, G. Laporte, and F. Semet, "Solving an ambulance location model by tabu search," *Loc. Sci.*, vol. 5, no. 2, pp. 75–88, 1997.

[33] R. Batta, J. M. Dolan, and N. N. Krishnamurthy, "The maximal expected covering location problem: Revisited," *Transportation Sci.*, vol. 23, no. 4, pp. 277–287, 1989.

[34] J. F. Repede and J. J. Bernardo, "Developing and validating a decision support system for locating emergency medical vehicles in Louisville, Kentucky," *EU J. Op. Res.*, vol. 75, no. 3, pp. 567–581, 1994.

[35] A. Ingolfsson, S. Budge, and E. Erkut, "Optimal ambulance location with random delays and travel times," *Health Care Mgmt. Sci.*, pp. 262–274, 2008.

[36] V. Marianov and C. ReVelle, "The capacitated standard response fire protection siting problem: deterministic and probabilistic models," *Ann. Op. Res.*, vol. 40, no. 1, pp. 303–322, 1992.

[37] O. Berman, "Dynamic repositioning of indistinguishable service units on transportation networks," *Transportation Sci.*, vol. 15, no. 2, pp. 115–136, 1981.

[38] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, no. 2-3, pp. 235–256, 2002.

[39] G. L. Nemhauser and L. A. Wolsey, "Integer programming and combinatorial optimization," *Constraint Class. Mixed Int. Prog. Formulations, COAL Bulletin*, vol. 20, pp. 8–12, 1988.

[40] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: Algorithms and complexity*. Prentice Hall, Inc., 1982.

[41] M. S. Daskin, "A maximum expected covering location model: Formulation," 1983.

[42] S. Agrawal and N. Goyal, "Analysis of thompson sampling for the multi-armed bandit problem," *Conf. Learn. Theory*, pp. 39–1, 2012.

[43] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," *Proc. 19th Int. Conf. Wold Wide Web*, pp. 661–670, 2010.

[44] S. Agrawal and N. Goyal, "Thompson sampling for contextual bandits with linear payoffs," *Int. Conf. Mach. Learn.*, pp. 127–135, 2013.

[45] W. Chen, Y. Wang, Y. Yuan, and Q. Wang, "Combinatorial multi-armed bandit and its extension to probabilistically triggered arms," *J. Mach. Learn. Res.*, vol. 16, pp. 1–33, 2016.

[46] A. O. Saritaç and C. Tekin, "Combinatorial multi-armed bandit problem with probabilistically triggered arms: A case with bounded regret," *IEEE Glob. Conf. Signal Info. Process. (GlobalSIP)*, pp. 111–115, 2017.