Extended Hypercube Models for Large Scale Spatial Queueing Systems

by

Burak Boyacı

School of Architecture, Civil and Environmental Engineering
Urban Transport Systems Laboratory
Ecole Polytechnique Fédérale de Lausanne (EPFL)
GC C2 406, Station 18, 1015 Lausanne, Switzerland
Phone: +41-21-69-32486
e-mail: burak.boyaci@epfl.ch

Nikolas Geroliminis*

School of Architecture, Civil and Environmental Engineering
Urban Transport Systems Laboratory
Ecole Polytechnique Fédérale de Lausanne (EPFL)
GC C2 389, Station 18, 1015 Lausanne, Switzerland
Phone: +41-21-69-32481, Fax: +41-21-69-35060
e-mail: nikolas.geroliminis@epfl.ch

*Corresponding author

5400 words + 8 figures + 0 tables

For Presentation only
91th Annual Meeting
Transportation Research Board
Washington, D.C.
January 2012

ABSTRACT

Different than the conventional queueing systems, in spatial queues, servers travel to the customers and provide service on the scene. This property makes them applicable to emergency response (e.g. ambulances, police, fire brigades) and on-demand transportation systems (e.g. shuttle bus services, paratransit, taxis). The difference between the spatial queues and conventional queueing systems is various types of customers and servers and different service rates for different customer-server pairs. For the Markovian arrival and service characteristics, one of the methods to find system performance measures is to model and calculate steady state probability of the Markov chain for the hypercube queueing model (*1*).

One of the obstacles on the way to apply hypercube queueing models to real life problems is the size of the problem; it grows exponentially with the number of servers and a linear system with exponential number of variables should be solved for each instance. In this research, in order to increase scalability of the problem, we propose two new models. In addition to that, we modeled the problem by using Monte Carlo simulation and tested the convergence and stability properties of the simulation results and compare them with stationary distributions. In the final part, a mixed integer linear programming formulation is given for optimal server configuration with different objectives improving different performance measures. As a future work, we are planning to use the optimal solutions of this formulation to evaluate different dispatching policies.

INTRODUCTION

Emergency response systems are important for modern societies. They protect public health, provide assistance and ensure safety. Response areas of ambulances, design of police-beats or locations of fire brigades are important decisions for these systems. Although the demand rate is low on average for emergency response systems, the service availability is important when they are needed. In other words, in addition to adequate coverage, rapid and reliable response times are also important for emergency response systems.

One of the main issues that emergency response systems should cope with is the level of congestion in the service requests. Although on average the system utilization is not close to one, the probabilistic nature of the demand and service times can build congestion. The amount of congestion is directly related to the number of servers and the budget that is dedicated to these systems. But the smart strategic decisions have great effect on them as well. This is the reason we need scientific approaches, which are consistent but also applicable. Clever allocation of the resources can improve the level of service without increasing the dedicated budgets.

On-demand transportation (also known as demand responsive transport, dial-a-ride transit) is an advanced, user-oriented form of public transport with flexible routing and scheduling of vehicles operating in shared-ride mode between pick-up and drop-off locations according to passengers' needs. These systems provide service in areas with low passenger demand where regular bus service is not applicable. Shuttle bus services, paratransit, shared taxis and taxicabs are some types of on-demand transportation systems.

Although intelligent transportation systems technologies (e.g. signal priority, exclusive lanes, route guidance information) help on-demand transportation systems to work better, there is still a need for efficient scheduling and dispatching strategies for these highly variant and congested systems. For instance, deciding the borders of sub-regions and number of paratransit vehicles needed in each region, to maximize service rate with limited number of vehicles is an interesting and important question for these systems.

In this research, we are aiming to find methods that will improve the performance of systems of servers dealing with stochastic demand (e.g. emergency response systems, on-demand transportation systems). Specifically in the problems that we are interested in the stochasticity existing in the time and location of the demand. Although there are several different approaches to these problems, we are more interested in the spatial queueing models. Because they take the association between servers into consideration and with this property more close to reality.

Structure of the paper is constructed as follows. The following section gives a brief literature about the emergency response systems. Next section continues with the definition of one of the first solution procedures of spatial queues, hypercube queueing model. This is followed by "Extended Hypercube Queueing Models" section in which we deal with the two new hypercube queueing models that are altered versions of the conventional ones. Next section, namely "Monte Carlo Sampling", describes the solution procedure of extended hypercube queueing models through simulation. In this part both the convergence and stability properties of the simulation are checked by comparing it with the results estimated from (extended) hypercube queueing models. In the next part, mixed integer linear programming formulations for different performance measures are given. The results of these models can be seen as ideal dispatching policies because of two reasons. The model knows all the future from the beginning and the policy that improves given performance measure is calculated optimally. Our aim with these models is to find locations of the servers' that will improve the system and to evaluate different dispatching policies in the

future. In the final part of this section, the results for different measures are compared with the results of the most basic dispatching policy: "assign closest available". Finally, we present outcomes of current models and potential future dimensions of the research.

LITERATURE SURVEY

The early models dealing with the location of emergency response systems assume deterministic demand. They ignored stochastic nature of the problem and dealt on coverage and median models.

Median problems locate the facilities on discrete candidate locations that minimize average response time or distance. Hakimi (*2*) proposed p*-median* problem in which the main aim is to locate p facilities on a finite set of candidate locations in such a way that minimizes total transportation cost of n customers. Although it is a combinatorial optimization problem, there are some exact algorithms (*3, 4, 5*) and heuristic methods (*6, 7*) as well. Recently Mladenović et al. (*8*) write a survey which covers most of the literature on meta-heuristics about this subject.

Coverage models are used to locate limited number of facilities (i.e. emergency response systems) which maximize total coverage. Toregas et al. (*9*) proposed the *location set covering problem* in which the objective is to cover the entire area within a desired distance by minimum number of facilities . The *maximal covering location problem* (MCLP) which is proposed by Church and ReVelle (*10*), maximizes coverage within a desired distance S by locating a fixed number of facilities. In the probabilistic version of this problem, namely *maximum availability location problem* (MALP), the maximized value is the regions which are covered with α-reliability (*11*). Daskin and Stern (*12*) altered the MCLP and proposed a model named *backup coverage model* that maximizes the number of regions that are covered more than once. Gendreau et al. (*13*) modified the backup coverage model with two time limits.

Although the literature mainly covers static and deterministic location models, in recent models uncertainty is also taken into account. This uncertainty can be either related to planning future periods (dynamic models) or input model parameters (probabilistic models). *Dynamic models* are suitable for models which, are considering the relocation of vehicles. The first article on this subject is written by Ballou (*14*) in which the main aim is to relocate a warehouse in such a way that maximizes the profit in a finite horizon. Scott (*15*) works with the extension of this problem with more than one facilities. Schilling (*16*) extends MCLP with additional time constraint.

For urban problems, it is obvious that *probabilistic models* are the most suitable ones. For location and allocation of the emergency response systems and other service on-demand vehicles (e.g. taxis), it is more convenient to model both the demands and the duration of the time the facility serving these demands with probabilistic models. In these models, with some probability, it is always possible to have demand which cannot be intervened by any facility, because of stochasticity in both demand and service times. Manne (*17*), Daskin (*18*), ReVelle and Hogan (*19*) and, Marianov and ReVelle (*11*) are some of the important articles written in this literature.

Larson (*1*) proposed a *hypercube queueing model* (HQM) which is the first model that embeds the *queueing theory* in facility location allocation problems. This model analyzes systems such as emergency services (e.g. police, fire, ambulance, emergency repair), door-to-door pickup and delivery services (e.g. mail delivery, solid waste collection), neighborhood service centers (e.g. outpatient clinics, libraries, social work agencies) and transportation services (e.g. bus and subway services, taxicab services, dial-a-ride systems) which has response district design and service-to-customer mode (20). The solution of this model provides state probabilities and associated system performance measures (e.g. workload, average service rate, loss rate) for given server locations.

"The HQM is not an optimization model; it is only a descriptive model that permits the analysis of scenarios" (21). HQM models the current state as a continuous-time Markov process but does not determine the optimal configuration. Police patrolling (22) and ambulance location (23) are two applications modeled by HQM. Marianov and ReVelle (*11*) extended the MALP and developed *queueing maximum availability location problem*.

The first model proposed by Larson (*1*) assumes that the service time is independent of the locations of the calls for service and the dispatched unit. This argument was supported by the idea that time spend on the road is negligible compared to service time. This can be a fact for fire brigades but not for the ambulances and on-demand vehicles. However even with this simplification, as number of servers (n) increases, number of states (2^n) grows exponentially. As an extension, Atkinson et al. (*24*) assume different service rates for each server in the system with equal interdistrict or intradistrict responses which increases number of states (3^n) significantly. Recently, Iannoni and Morabito (*25*) and Iannoni et al. (*26*) embedded hypercube in a genetic algorithm framework to locate emergency vehicles along a highway. They extend the problem to enable multiple dispatch (e.g. more than one server can intervene for the same incident). Geroliminis et al. integrate the location and distracting decisions in the same optimization and solve the problem by using steepest descent (27) and genetic algorithms (28).

HYPERCUBE QUEUEING MODELS

The conventional HQM models (*1*) include *hypercube* in the name since the transition graph of the continuous time Markov chain used to model this structure has a hypercube structure when the number of servers is more than three. State variables contain n binary variables which shows if server i is available (0) or busy (1). In other words, each state is a number in base 2 and each digit shows the state of the corresponding server. For each region which is called *atom* (j) there exists a priority list of servers. Incidents in each region are served by the available server with the highest priority for this atom. If there does not exist any available server, either the call is lost (i.e. call for ambulance is dispatched by a backup) or joins a queue to be served (i.e. taxi customers are asked to wait until there is one available), depending on the problem assumptions. Service requests arrive from each atom according to an independent Poisson process with parameter λ_j. Larson (*1*) assumes each server has exponentially distributed equal service rates μ_i for any region. The transition graph of HQM with three states for this model can be seen in Figure 1a. Note that, as the system gets full, in other words more servers get busy, the burden on the free servers increases. For instance in state "110" all the servers but the first are busy. That is why the next incident in any region will be served by the first server. This is also the reason of having high transition rate ($\lambda_1 + \lambda_2 + \lambda_3$) from state "110" to "111".

For different rates of inter and intradistrict responses, the size of the model will increase. In this model we have three different possibilities for each server: available (0), busy with intradistrict response (1) and busy with interdistrict response (2) (*24*). Figure 1b is a transition graph of an example with two servers. As an example "20", represents the state where the first server is available and the second server intervenes an incident outside its own region.

It is good to note here that, the intradistrict server has always priority for the incidents inside its own region. We can see this in the transition diagram. When the system is empty, if there is an incident in a region, we cannot assign server from another region. This is also the case for Larson's model. However, we should also note here that, this does not prevent having states such as "22". Although, practically it is rare for lightly congested systems, it has a positive probability

in theory.

EXTENDED HYPERCUBE QUEUEING MODELS

In the models proposed by Larson (*1*) and its counterpart (*24*) with different inter and intradistrict service rates, there are 2^n and 3^n states respectively which makes each limiting probability difficult to calculate, for even very small cases. For instance, if there are 20 servers, number of states for 2^n model is around one million whereas for the 3^n case this is more than three billions. In other words, we need to solve a system of equations with over million unknowns. One of the ways to get rid of this problem can be looking at the problem in an aggregate level.

In extended HQMs, we alter the conventional model in such a way that more than one servers can be assigned to each region. We model the problem as a discrete location problem with queueing characteristics. In this new model, there are I types of servers which we call them *bins*. Each server in these bins serves his own customers and the rest with different service rates (μ_i and μ_i'). There is only one queue and this queue works in first in first out (FIFO) manner. Each customer who enters the system or leaves the queue to have service chooses the server which serves him with the maximum rate (which is given as priority list). We have n servers and our aim is to decide how many servers should we assign for each bin (n_i for $\forall i$) to optimize the average performance of the system (e.g. minimize interdistrict response, loss rate and/or maximize average service rate). If the interarrival time of each customer and service time of each server had deterministic distributions, this model would be modeled as a simple discrete location allocation problem and could be solved by a linear programming formulation. However, we are interested in stochastic systems and this model is more appropriate for the problems with probabilistic demand and service times.

At first glance the proposed model can be seen suboptimal for emergency response systems because, making responsible regions smaller and assigning one emergency response system for each region would give better results. However in the conventional HQM, number of states increases extremely fast and with this extension, the model can be used to solve real life instances with good accuracy. Furthermore, for deciding the location and allocation of on-demand transportation systems, this approach is more convenient. Cities can be partitioned into regions and number of on-demand vehicles which will optimize the overall system performance can be assigned to these regions accordingly. Although this model has exponential number of states $((n_1+1)(n_2+1)...(n_I+1))$ it is far less than the conventional hypercube models. As an example, a system of 3 bins with 9, 6 and 5 servers in each for different inter and intradistrict service rate case ($\mu_i \neq \mu_i'$ for $\forall i$) number of states is 32340 whereas this number is 420 if we assume equal service rates for inter and intradistrict responses ($\mu_i = \mu_i'$ for $\forall i$). Please note that for the same total number of servers, the conventional two models need over million states.

The first *extended hypercube queueing model* (EHQM) that we are proposing assumes equal intra and interdistrict service rates. Each number in the state name presents number of busy servers in this *bin*. For instance "132" stands for 2, 3 and 1 busy servers in the first, second and third bins respectively. An EHQM model contains $((n_1+1)(n_2+1)...(n_I+1))$ states in which n_i is assigned number of servers in bin i and I is the total number of bins. The transition graph of an example with three bins which has 3,2 and 1 servers respectively in each bin is shown in Figure 2. Note that this model has 24 states, which is 64 for Larson's traditional hypercube model. By using the following transition graph we can write the transition equations for each state and can calculate steady state probabilities. For instance for the state "012" in which there are 2, 1 and 0

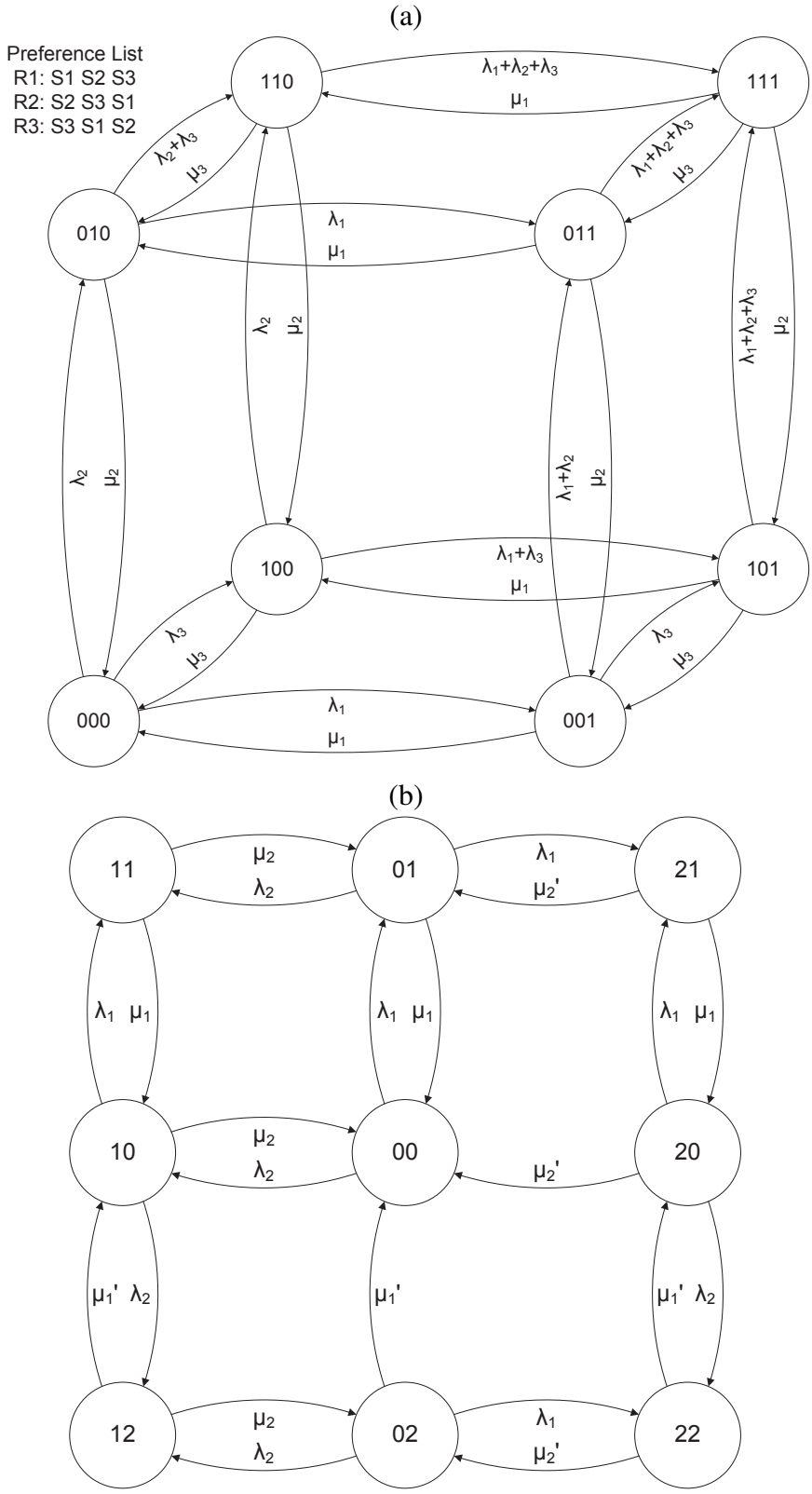


Figure 1: (a) Larson (*1*)'s HQM for three servers with equal inter (μ_i) and intradistrict (μ_i') service rates and, (b) HQM for two servers with different inter (μ_i) and intradistrict (μ_i') service rates

busy servers in the first, second and third bins we can write the following transition equation:

$$P_{012}\left(\lambda_1+\lambda_2+\lambda_3+2\mu_1+\mu_2\right)=\lambda_1 P_{011}+\lambda_2 P_{002}+3\mu_1 P_{013}+2\mu_2 P_{022}+\mu_3 P_{112} \qquad (1)$$

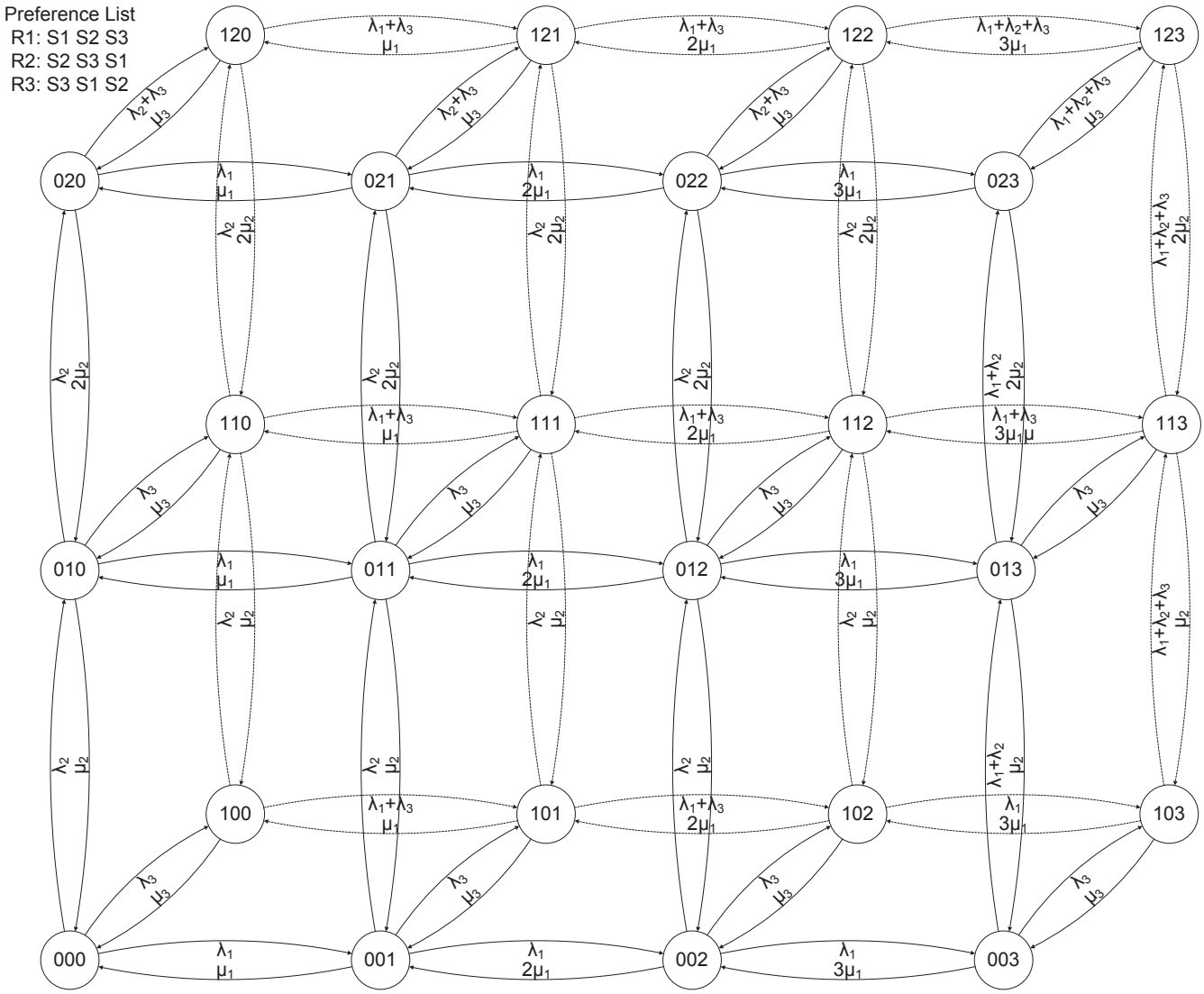


Figure 2: EHQM for three bins with equal inter and intradistrict service rates (μ_i)

For different intra and inter-arrival service rates EHQM is not descriptive enough. For this reason, we are proposing another model (EHQM′) which has more states but differentiates the intra and interdistrict responses from each other. Note that it is also possible to construct a model which is more detailed with different service rates for each customer-server pair but this will create excessive amount of states. That's why we have only two different service rates for each server, μ_i (intradistrict) or μ_i' (interdistrict). In addition to that, in the states, there are two variables for each server. A variable pair for each server shows the number of intra and interdistrict responses dispatched by corresponding server separately. Thus, the new model has $\prod_i \binom{n_i+2}{2}$ states where n_i is number of servers assigned to bin i.

In Figure 3 one can see a transition matrix of an (EHQM′) model which has 2 bins with 2 servers in each. Each row of the state name gives information about one of the bins. For instance in the state "$\substack{11\\10}$", both servers in bin one are busy where the left side shows the number of servers in intradistrict response and the right side shows the number of servers in an interdistrict response. In bin two (second row), there is one server in intradistrict response and no server in interdistrict response. This state can also be seen separately with the states connected to it in Figure 4. The transition equation can be written as:

$$P_{\substack{11\\10}}\left(\lambda_1+\lambda_2+\mu_1+\mu_1'+\mu_2\right)=\lambda_1 P_{\substack{01\\10}}+\lambda_2 P_{\substack{11\\00}}+\mu_2' P_{\substack{11\\11}}+2\mu_2 P_{\substack{11\\20}} \tag{2}$$

MONTE CARLO SAMPLING

Monte Carlo sampling is a class of computational algorithm that utilizes repeated random sampling to compute the results. In our problem, we plan to apply the algorithm to generate random arrival samples. Our main aim is to model spatial queueing systems as a *mixed integer linear programming* (MILP) formulation. We plan to use the generated arrival samples with deterministic service time estimation to find where to locate the servers.

Before modeling the spatial queueing systems by using a MILP formulation, we need to be sure that the properties of the HQM can be represented by Monte Carlo sampling. Note also that the analytical solution of HQMs (with Markovian equations) represent steady-state conditions, which might not be the case when demand is time-varying (as in reality). For this purpose, a discrete time event simulation is created and the results acquired from the simulator are compared with the results of the HQMs. We compare both convergence and stability properties of the simulation. For the convergence check, we compared the convergence rate of the simulation to the probabilities calculated by solving Markov model of the HQM. Similar to that, for the stability performance of the method, we investigated the probabilities of the simulation after sharp and tense changes. The experiments give promising results and show that this method is applicable for any HQM. Before going into details of these experiments, let us start with describing the simulation environment.

In these experiments we tested the 3^n and 3^n aggregate HQM without queue. The main difficulty of 3^n aggregate models with queue is the computational complexity as (different than 2^n aggregate model) this model with queue has more than one tails. In other words, if we want to implement a 3^n aggregate HQM with queue, our problem size is hardly limited. We plan to deal with this problem in our future work. In a system without queue, each customer is either served right after arrival occurs if there is an available server eligible to serve that customer or lost and left the system without service. The interarrival distribution of each customer type and the service time distribution of each server for given customer type are predetermined.

We have two events in the system: arrival and departure of a customer. At the beginning of the simulation we create arrivals for each type of customer. When simulation clock hits an arrival a new arrival event is created by using the given arrival distribution for the customer type and added to the event list. If it is served by a server, service time of the customer is also calculated and a departure event is added to the event list. The pseudocode of the simulation can be seen in Figure 5. $\tilde{*}$ is a value generated from the distribution $*$.

The convergence rate of the system is investigated by comparing the simulation results with the calculated steady state probabilities from the Markov chain of the same HQM model. We have done the comparisons with plenty of different scenarios and random number seeds. Here we

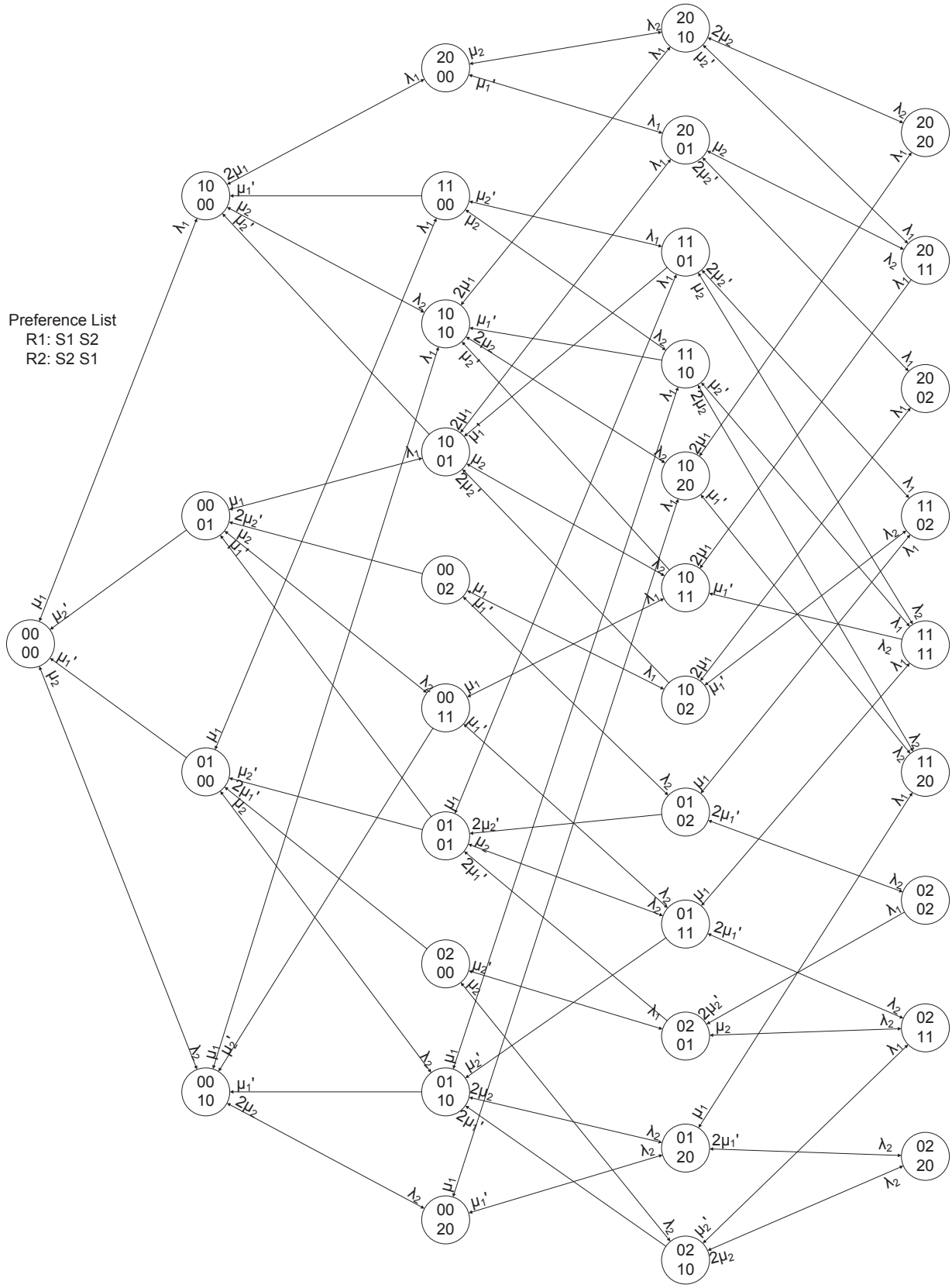


Figure 3: EHQM′ for two bins containing two servers in each bin with different intra (μ_i) and interdistrict(μ_i') service rates

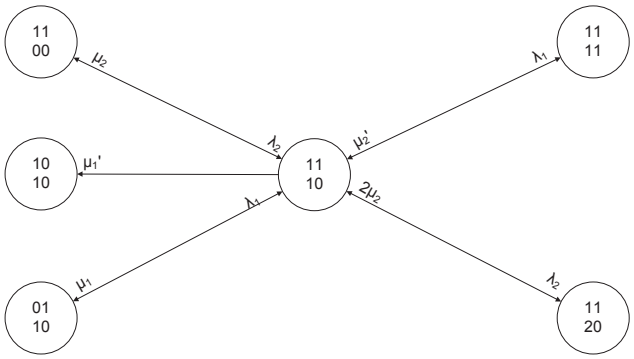


Figure 4: Single state with its connected states from EHQM′

---

input: T (arrival end time), A_j (random variable for interarrival of customer type j),
S_{ij} (random variable for service time of server type i to customer type j),
P_j (priority list of server types for customer j), n_j (number of servers of type j)

1. Initialize the event list $E = \{\}$, available server list $C_j = 0$ for $\forall j$
2. For each j
   (a) Generate arrival event e of type j with occurrence time $\tilde{A}_j$
   (b) Add event e to the event list
3. Repeat while there is an event in the list
   (a) Take the earliest event e from the event list with time stamp t and type j
   (b) If e is an arrival event
      i. Generate arrival event e^{new} of type j with occurrence time $t + \tilde{A}_j$
      ii. Add e^{new} to the event list if its occurrence time is less than T
      iii. Let i be the server who has the highest priority for customer type j
      iv. If i is a number
         A. Generate a departure event e^{new} of server type i serving customer type j with occurrence time $t + \tilde{S_{ij}}$
         B. Decrease n_j by 1
      v. Else
         A. Increase loss customers count by 1
   (c) Else (if e is a departure event of server type i serving customer type j)
      i. Increase n_j by 1

Figure 5: Pseudocode for the discrete event simulation for HQM model without queue

present the results from three scenarios with different demand intensity. In the first scenario, we checked the convergence rate on conventional 3^n HQM with 8 servers. In the second and third examples, 3^n aggregate models with 4 bins are taken into consideration. The demand is almost equally distributed in the second scenario. Opposite to that, in the third scenario we simulated an instance with different demand rates. In order to create heavily (lightly) congested systems, the demand rates are multiplied (divided) by two.

The convergence graphs for the generated scenarios can be seen in Figure 6a. In these graphs, the x-axes show number of incidents (arrival or departure) whereas y-axes are the average absolute difference between steady state probabilities and the probabilities calculated from the beginning of the simulation. In each scenario we used two different service time distributions: Markovian and deterministic.

From the convergence graph given in Figure 6a, it is seen that, the convergence rate of the scenario to the exact solution is quite fast. After 1000 incidents (arrivals and departures are separate incidents) the difference between exact solution and the simulation solution becomes less than 2%. One of the other observations from these graphs is the effect of service time distribution to the system. It can be seen clearly that, the simulations with Markovian and deterministic service times have almost the same convergence curves to the exact solution. These two findings tell us that these spatial queueing problems can be modeled as a MILP problems with an incident list of 2000 arrivals and deterministic arrivals (for errors less than 1%). Note that, this behavior is not specific to the scenarios represented; we observed the same results in almost all of the scenarios that we have tested.

Besides convergence, stability in rapid changes is also investigated. In all three scenarios, demand rate of each region is multiplied by 5 between time interval 400 and 410. Different than the convergence graphs, in stability graphs, x-axes represent the time and the y-axes are the probability differences between the exact value and the value calculated from the scenario. The stability graphs for the three cases can be seen in Figure 6b.

Note that the duration of the three peaks in Figure 6b is different. The explanation for this phenomenon is the following: In order to show all three scenarios in the same graph, we use simulation clock in x-axes. However convergence rate depends on number of incidents but not the simulation clock. If we check the peak durations in each graph, we will realize that, half demand intensity peak is twice as long as the normal demand intensity. This relationship is similar for normal and double demand intensity comparisons as well. In other words, if we convert the x-axes into number of iterations all three peaks have equal lengths. An interesting finding is that a large fluctuation (demand 5 times higher) of the system for a duration of t, is observed approximately after $3t$, which means that these systems return to their steady-state fast.

We can also see from Figure 6b that the convergence of the system to the steady state probabilities after rapid changes is fast. In other words, simulation reacts the changes in the demand as soon as they occurs but recovers with the same pace; fluctuation in the demand effects the simulation but compensates this dramatic increase in the demand quite fast as well. This is a beneficiary property of the spatial queueing systems. Because of this property, HQM can be applied to demands with fluctuations. When a time-dependent smooth demand is applied, an HQM can be solved at each time step to identify the steady state performance measures, which will be close to reality as fast convergence shows. Last but not least, in these graphs, it can also be seen that the deterministic service time assumption gives very close results to the Markovian one, which will make the solution procedure of the next section much simpler.

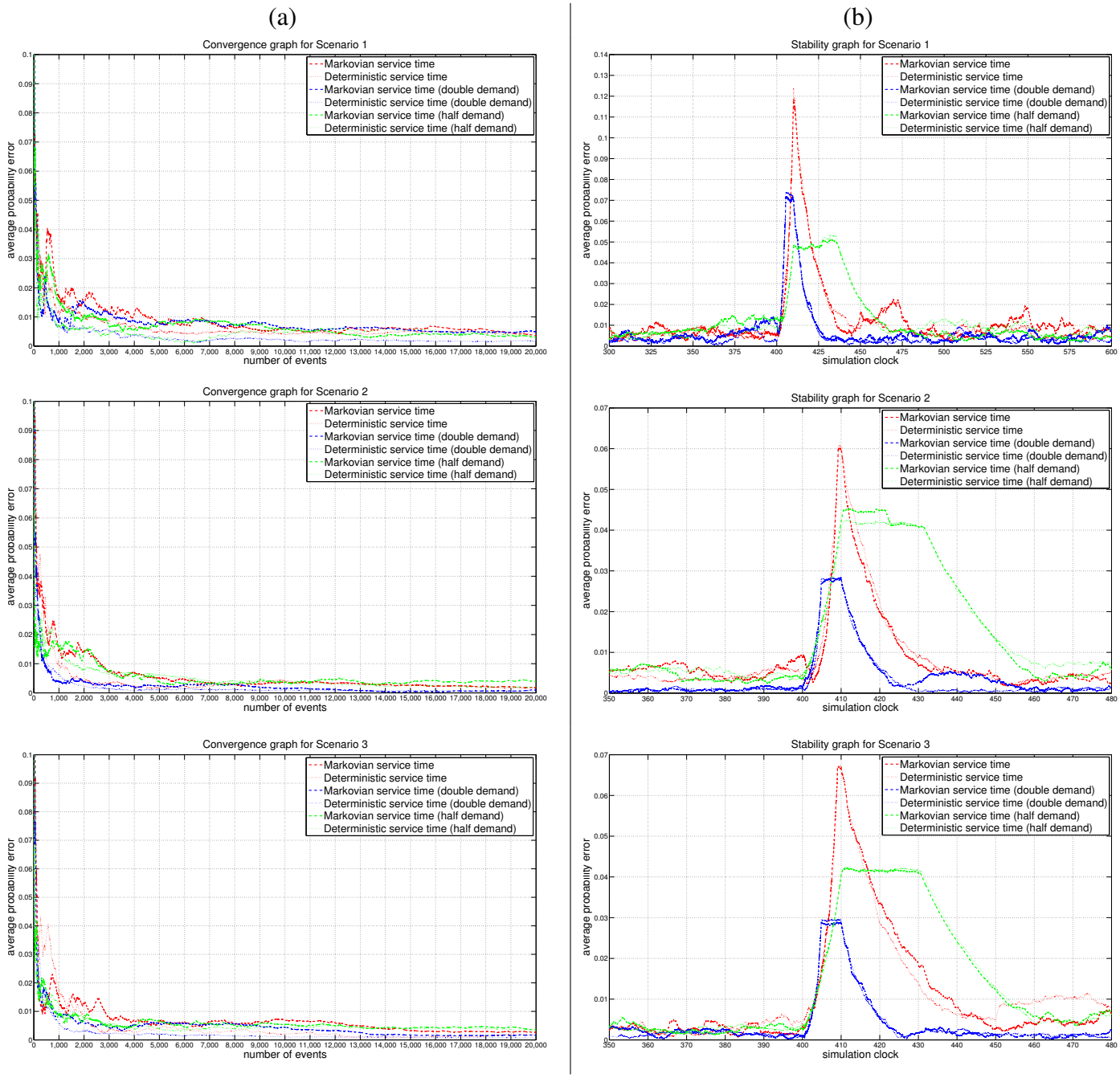


Figure 6: (a) Stability and (b) convergence graphs of the three scenarios (one event is an arrival or departure)

MIXED INTEGER LINEAR PROGRAMMING FORMULATION

Although, the EHQMs significantly decrease the sizes of the solvable instances once compared with existing HQMs and make them applicable to larger real life problems, in order to model fluctuating demand with different distributions than exponential, we propose a MILP formulation. Advantages of the proposed MILP formulation are (i) it can give faster results for even larger instances and (ii) dispatching policies can be an endogenous variable of the problem (not the case in HQM or EHQM). Besides, with the model we propose, we can differentiate the locations of the incidents inside the regions as well. In other words, we can divide the city into regions with similar demand profiles and then generate the locations of the incidents inside the regions for any given distributions. This will help us to model more realistic and applicable results. In other words, we can use this model to find better server locations that will improve overall system performance.

In addition, the proposed MILP formulation gives an ideal dispatching policy for given demand profile. It optimizes the dispatching policy for given sequence of demand as it can see the future knows every incidents in advance. We can either set facility locations or number of facilities and can compare the results of the model with any dispatching policy results. In other words, we give all the incidents with their locations and times and the model finds the optimal dispatching and number of servers in each candidate location. Of course in reality, knowing the exact time and location of a stochastic incident is impossible but this model can be a good tool to evaluate different dispatching policies.

We can now define the mathematical model. For given indices, sets and parameters:

i, j : incident indices,
k : candidate location index ($k = 0$ stands for the dummy server),
s : scenario index,
A : total number of servers,
I : number of incidents in each scenario,
K : number of candidate locations,
S : number of scenarios,
$\overline{n}_k$: capacity of candidate location k,

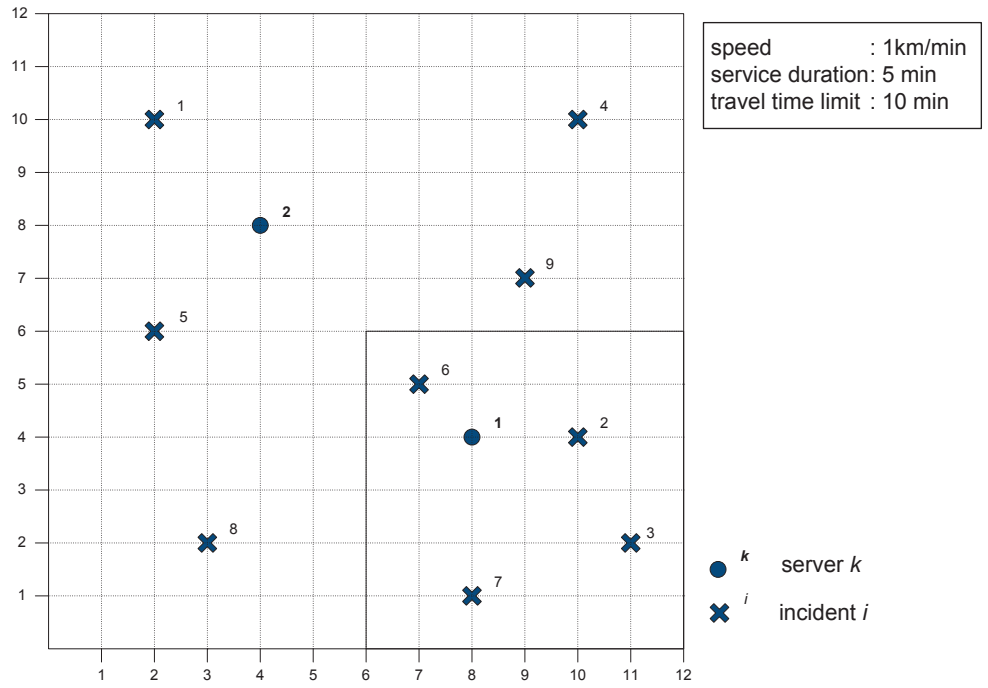
S independent demand profiles (scenarios) with K incidents in each are created with their coordinates, $x_j^s = \left(x_j^{s(1)}, x_j^{s(2)}\right)$, and time, t_j^s. Please note that in order to have satisfactory results, both the number of scenarios and incidents should be adequate. Previous section has created some intuition towards this direction.

As a second step, the binary parameters, e_{ik}^s and r_{ijk}^s are calculated for all $i, j \in I, k \in K$ and $s \in S$. We can define e_{ik}^s and r_{ijk}^s as follows:

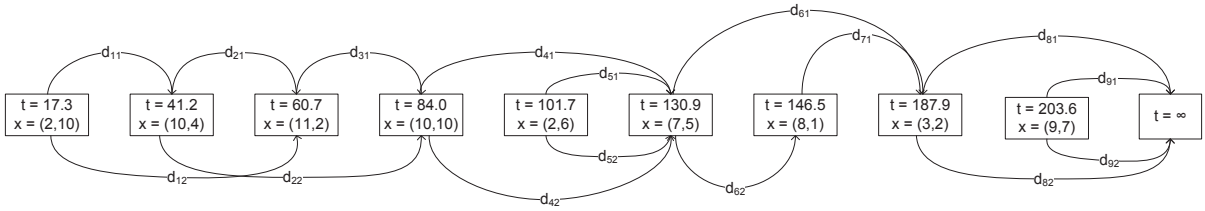
$$e_{ik}^s = \begin{cases} 1, & \text{if any server located in candidate location } k \text{ is eligible for incident } i \\ & \text{in scenario } s \\ 0, & \text{o/w} \end{cases}$$

$$r_{ijk}^s = \begin{cases} 1, & \text{if any server located in candidate location } k \text{ becomes available in time} \\ & \text{interval } \left(t_{j-1}^s, t_j^s\right) \text{ after serving incident } i \text{ in scenario } s \\ 0, & \text{o/w} \end{cases}$$

Note that, r_{ijk}^s may depend on several different parameters such as speed of the servers, duration of the service etc. Besides r_{ijk}^s can be set to 0 if server k is not eligible for incident i ($e_{ik}^s = 0$) for example if distance between incident and the candidate location is large. In Figure 7, you can see how r_{ijk}^s are calculated on a small example.



Incident	Time (minute)	Distance		Service Duration		Service End Time	
		server 1	server 2	server 1	server 2	server 1	server 2
1	17.3	10	4	25	13	42.3	30.3
2	41.2	2	10	9	25	50.2	66.2
3	60.7	5	13	15	31	75.7	91.7
4	84.0	8	8	21	21	105.0	105.0
5	101.7	8	4	21	13	122.7	114.7
6	130.9	2	6	9	17	139.9	147.9
7	146.5	3	11	11	27	157.5	173.5
8	187.9	7	7	19	19	206.9	206.9
9	203.6	4	6	13	17	216.6	220.6



The r^s_{ijk} that are 1 are (s is omitted since there is only one scenario):

$r_{121}, r_{132}, r_{231}, r_{242}, r_{341}, r_{461}, r_{462}, r_{561}, r_{562}, r_{681}, r_{672}, r_{781}, r_{8\infty1}, r_{8\infty2}, r_{9\infty1}, r_{9\infty2}$.

Figure 7: Example showing the calculation of r^s_{ijk}

After the calculation of the parameters r^s_{ijk}, for decision variables:

n_k : (initial) number of assigned servers to candidate location k

a^s_{ik} : number of available servers in candidate location k at time t^s_i in scenario s

d^s_{ik} = $\begin{cases} 1, \text{ if incident } i \text{ is served by a server in candidate location } k \text{ in scenario } s \\ 0, \text{ o/w} \end{cases}$

d^s_{i0} = $\begin{cases} 1, \text{ if incident } i \text{ in scenario } s \text{ is not served} \\ 0, \text{ o/w} \end{cases}$

the mathematical model that minimizes number of unserved incidents can be formulated as:

$$\min \sum_{s=1}^{S}\sum_{i=1}^{I} d^s_{i0} \tag{3}$$

$$\text{s.t.} \sum_{k=1}^{K} d^s_{ik} + d^s_{i0} = 1 \qquad \forall\,(i,s) \tag{4}$$

$$A\sum_{k=1}^{K} d^s_{ik} \geq \sum_{k=1}^{K} a^s_{ik} e^s_{ik} \qquad \forall\,(i,s) \tag{5}$$

$$a^s_{ik} \geq d^s_{ik} \qquad \forall\,(i,k,s) \tag{6}$$

$$a^s_{ik} = a^s_{(i-1)k} - d^s_{(i-1)k} + \sum_{j=1}^{i} \left(r^s_{ijk} d^s_{jk}\right) \qquad \forall\,(i,k,s) \tag{7}$$

$$\sum_{k=1}^{K} n_k = A \tag{8}$$

$$n_k \leq \overline{n}_k \qquad \forall k \tag{9}$$

$$n_k = a^s_{1k} \qquad \forall s \tag{10}$$

$$n_k \in \mathcal{N} \qquad \forall k \tag{11}$$

$$a^s_{ik} \in \mathcal{N} \qquad \forall\,(i,k,s) \tag{12}$$

$$d^s_{ik} \in \{0,1\} \qquad \forall\,(i,k,s) \tag{13}$$

$$d^s_{i0} \in \{0,1\} \qquad \forall\,(i,s)\,. \tag{14}$$

In the model given above, Objective 3 minimizes the summation of the unserved incidents. Constraints 4 forces each incident to be served by a real ($k = 1, ..., K$) or dummy ($k = 0$) server. This will help us to penalize lost incidents. With Constraints 5 the model is obliged to assign a server if there exists at least one eligible server in the system. Constraints 6 check if there is an available server to be assigned for every incident i in scenario s. Constraints 7 are the general balance equations for the number of available servers for given incident times. Constraints 8 and 9 limit the number of servers in the whole system and each candidate locations. Constraints 10 are used to assign the same number of servers at the beginning in different scenarios. The rest of the constraints (11 - 14) are nonnegativity, integer and/or binary constraints for the decision variables.

For,

α^s_{ik} : service duration if server k serves incident i in scenario s

α^s_{i0} : : penalty if incident i is not served by any server in scenario s

a mathematical model that minimizes total service time can be formulated as:

$$\min \sum_{s=1}^{S}\sum_{i=1}^{I}\sum_{k=0}^{K} \alpha_{ik}^{s} d_{ik}^{s} + \sum_{s=1}^{S}\sum_{i=1}^{I} \alpha_{0k}^{s} d_{0k}^{s} \tag{15}$$

$$\text{s.t. Constraints } 4-14. \tag{16}$$

We can also model a problem that minimizes number of servers that will serve all the incidents (zero loss rate). This problem can be formulated as follows:

$$\min \sum_{k=1}^{K} n_k \tag{17}$$

$$\text{s.t. } \sum_{k=1}^{K} d_{ik}^{s} = 1 \qquad \forall\,(i,s) \tag{18}$$

$$\text{Constraints } 6-7, 9-13. \tag{19}$$

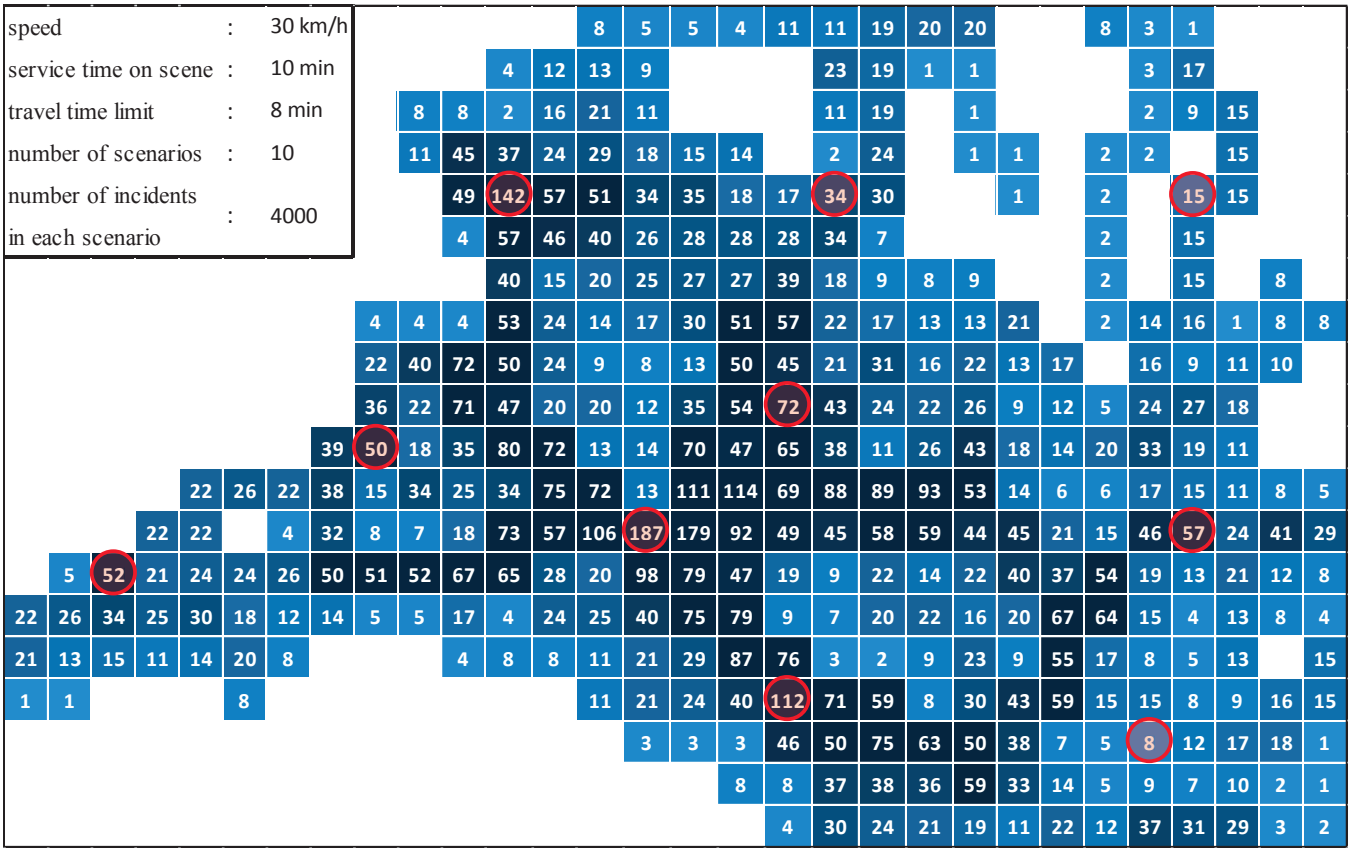
EXPERIMENTAL RESULTS

In this section we evaluate the performance of simulation with the "assign nearest available" policy by using the MILP formulation. We demonstrate the model for locating repair and tow-away vehicles for public transport in Athens (Greece) surface transportation network. This network contains around 3000 buses of different size. This system is used by 1.7 million passengers. Although the whole area is about 650 km^2 we deal with the 150 km^2 area of the highly populated part which contains more than 85% of the demand.

In Athens, the buses are handled by city's bus company (ETHEL) whereas the Athens Public Transportation Organization (OASA) is responsible for planning and managing the bus system. In Figure 8a you can see incident percentages that are derived from 10-year historical data and normalized to 10000 per cells that are squares with 0.5 km in each side. In this example, 10 candidate locations for transit mobile repair units (TMRUs) are selected (pointed out with red circles) and number of TMRUs needed in each candidate location is calculated for given demand intensity. The reader can refer to Karlaftis et al. (*29*) for more information about the data.

In the experiments we used four different arrival rates: 0.2, 0.4, 0.8 and 1.6 arrivals per minute for the whole region. In other words, an arrival is generated with given rates and assigned one of the regions with the probability proportional to the demands given in Figure 8a. Travel pace is selected as 0.5 km/minute, maximum travel time limit is 8 minutes and service time on-scene is set to 10 minutes. In each run, 10 different scenarios are used with 4000 incidents in each as given in Figure 8a.

Firstly, for each demand rate we find minimum number of servers needed to serve all incidents (17-19). These values can be seen in the first row of Figure 8b. Then, we set total number of servers to these values and run two models that minimize (i) penalized average service time (15-16) or (ii) number of unserved incidents (3-14) and find number of servers in each candidate locations. It is worth to note here that, in both models, we put the other objective as secondary objective to have inferior results if there is a multiple optima. When average service time is minimized, the



b

		$A = 24$		$A = 31$		$A = 38$		$A = 54$	
	λ	minimize lost incidents	minimize service time	minimize lost incidents	minimize service time	minimize lost incidents	minimize service time	minimize lost incidents	minimize service time
MILP	0.2	0%	15.752	0%	15.708	0%	15.702	0%	15.701
	0.4	0.05%	15.976	0%	15.750	0%	15.713	0%	15.706
	0.8	5.62%	16.840	0.23%	16.196	0%	15.801	0%	15.737
	1.2	21.82%	16.582	5.76%	16.821	0.62%	16.098	0%	15.872
SIM	0.2	0.22%	15.790	0.02%	15.716	0.03%	15.703	0%	15.702
	0.4	1.58%	16.105	0.17%	15.807	0.15%	15.723	0.02%	15.706
	0.8	12.57%	17.180	3.18%	16.363	1.39%	15.910	0.21%	15.787
	1.2	27.93%	18.111	12.84%	17.281	5.92%	16.457	1.25%	16.097

c

$A = 24$		minimize lost incidents			minimize service time		
	λ	lost customer percentage	average service time	penalized average service time	lost customer percentage	average service time	penalized average service time
MILP	0.8	5.62%	17.189	22.066	2.60%	16.840	19.101
	1.2	21.82%	16.665	35.724	16.90%	16.582	31.351
SIM	0.8	12.57%	17.731	28.571	8.38%	17.180	24.452
	1.2	27.93%	18.439	42.334	23.15%	18.111	37.996

Figure 8: Demand and potential locations (a) and experimental results (b,c) for central Athens network

unserved incidents are penalized with four times maximum service time (which is a value very close to 26 minutes because of travel time limit and service time on-scene). In the second model total service time is added to the objective function by multiplying it with very small value ϵ.

After finding the number of servers in each candidate location, we fix the number of servers in each location and run the models with different incident lists to check the performance of optimal dispatching and "assign nearest available" policies for different demand rates. Here is an example. For arrival rate $\lambda = 0.2$, number of servers needed to serve all customers is 24. We set this as the total number of servers to the model and find the locations that will minimize (i) penalized average service time and (ii) number of unserved incidents. Then we fix the number of servers in each candidate locations (by adding equality constraints to the related model) and find optimal and "assign nearest available" strategies performance with all arrival rates. In the table given in Figure 8b the average service time and lost customer percent can be seen for both optimal dispatching (MILP) and "assign nearest available" (simulation) strategies. In Figure 8c the detailed information for four runs can be seen as well.

As we expect, lost incident percentage increases as demand increases. However, this is not the case at all in the service time. In the case where number of servers is minimum ($A = 24$) and arrival rate is maximum ($\lambda = 1.2$), average service time decreases. In fact these two changes are the reason of the lost incidents. Because, in these cases which can be seen more detailed in Figure 8c, lost customer percentages are more than 16%. As a result, the optimal dispatching algorithm selects the incidents which are closer. This fact is also supported by the simulation results. In simulation we do not see such a decrease when arrival rate is increased from 0.8 to 1.2.

One of the other important findings from these experiments is the robustness of two strategies. Obviously, the server configuration that minimizes service time is more robust to demand changes than the configuration that minimizes number of lost incidents. In other words, as demand increases, the server locations that minimizes service time for normal demand performs better than the server locations which minimize lost incidents. This is an important finding because, if we look for efficient configurations performing well for different demand rates, the latter formulation should be applied.

CONCLUSIONS

In this paper, we have investigated the location-allocation models with stochastic demand which are quite applicable to emergency response and on-demand transportation systems. We have started with two extensions to conventional hypercube queueing models. Extended models are appropriate to larger real life problems because of the limitations of conventional hypercube queueing models to deal with larger number of servers. In the section after, we have checked the convergence and stability properties of Monte Carlo simulation approach. It is seen that discrete time simulation converges sufficiently fast to the values calculated by hypercube models. Furthermore, we have also realized that, distribution of the service time has minimal effect on the system. Using deterministic service time does not change much the results. In the mixed integer linear programming part, we proposed three different models to find number of servers to locate on predefined candidate locations. In the experimental results section, we compared the dispatching policy which is optimal and "assign nearest available". Experiments showed that, although it is one of the best strategies, there is a significant gap between optimal dispatching and "assign nearest available". But, optimal dispatching strategy is a limit for the other dispatching policies and considers the events exactly happening in the future, which can not be predicted in real life. However, investi-

gating this dispatching policy in detailed can help us to find better dispatching policy. This is a future direction for our work. Besides, it can be used to evaluate performance of other dispatching policies objectively.

As a future research, we plan to continue in several different dimensions. First of all, one of the limitations of working with larger problems is the number of variables in the linear programming formulation. In the experiments each instance has 800000 variables in total. However, the structure of the problem makes it most of the time solvable in reasonable durations. In order to work with larger problems we plan to implement a column generation procedure. The difficulty emerging from this implementation is that column generation solves linear relaxation of the problem. We should find some methods to apply branch and price to our problem. One of the other dimensions is having a model which has queue capacity different than zero. In this model, we plan to give some waiting time options to the incidents and penalize the waiting time in the problem. Although this is not very common in emergency response systems, it is convenient in on-demand transportation. Using other methods such as robust optimization or stochastic programming is one of the other plans to realize. Although they will make problems more difficult to solve, these methods are appropriate for problems with stochastic nature. We can tackle the solution procedure difficulties by implementing a heuristic method instead of solving the implemented models optimally as well.

ACKNOWLEDGMENT

The authors thank Athens Public Transport Organization (OASA) and Prof. M.G. Karlaftis from NTUA for providing the data for the application of the model.

REFERENCES

[1] Larson, R., A hypercube queuing model for facility location and redistricting in urban emergency services. *Computers & Operations Research*, Vol. 1, No. 1, 1974, pp. 67 – 95.

[2] Hakimi, S., Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph. *Operations Research*, Vol. 12, No. 3, 1964, pp. 450–459.

[3] Galvão, R. and L. Raggi, A method for solving to optimality uncapacitated location problems. *Annals of Operations Research*, Vol. 18, 1989, pp. 225–244, 10.1007/BF02097805.

[4] Körkel, M., On the exact solution of large-scale simple plant location problems. *European Journal of Operational Research*, Vol. 39, No. 2, 1989, pp. 157 – 173.

[5] Avella, P. and A. Sassano, On The p-Median Polytope. *Mathematical Programming*, Vol. 89, 2001, pp. 395–411, 10.1007/PL00011405.

[6] Daskin, M. S. and A. Haghani, Multiple vehicle routing and dispatching to an emergency scene. *Environment and Planning A*, Vol. 16, No. 10, 1984, pp. 1349–1359.

[7] Schilling, D., V. Jayaraman, and R. Barkhi, A Review of Covering Problems in Facility Location. *Location Science*, Vol. 1, No. 1, 1993, pp. 25–55.

[8] Mladenović, N., J. Brimberg, P. Hansen, and J. Moreno-Pérez, The p-median problem: A survey of metaheuristic approaches. *European Journal of Operational Research*, Vol. 179, No. 3, 2007, pp. 927 – 939.

[9] Toregas, C., R. Swain, C. ReVelle, and L. Bergman, The Location of Emergency Service Facilities. *Operations Research*, Vol. 19, No. 6, 1971, pp. 1363–1373.

[10] Church, R. and C. ReVelle, The Maximal Covering Location Problem. *Papers in Regional Science*, Vol. 32, 1974, pp. 101–118.

[11] Marianov, V. and C. ReVelle, The queueing maximal availability location problem: A model for the siting of emergency vehicles. *European Journal of Operational Research*, Vol. 93, No. 1, 1996, pp. 110–120.

[12] Daskin, M. and E. Stern, A Hierarchical Objective Set Covering Model for Emergency Medical Service Vehicle Deployment. *Transportation Science*, Vol. 15, No. 2, 1981, pp. 137–152.

[13] Gendreau, M., G. Laporte, and F. Semet, Solving an Ambulance Location Model by Tabu Search. *Location Science*, Vol. 5, No. 2, 1997, pp. 75–88.

[14] Ballou, R. H., Dynamic Warehouse Location Analysis. *Journal of Marketing Research*, Vol. 5, No. 3, 1968, pp. pp. 271–276.

[15] Scott, A., Dynamic location - allocation systems: some basic planning strategies. *Environment and Plann*, Vol. 3, No. 1, 1971, pp. 73–82.

[16] Schilling, D. A., Dynamic Location Modeling for Public-Sector Facilities: A Multicriteria Approach. *Decision Sciences*, Vol. 11, No. 4, 1980, pp. 714–724.

[17] Manne, A., Capacity Expansion and Probabilistic Growth. *Econometrica*, Vol. 29, No. 4, 1961, pp. 632–649.

[18] Daskin, M., A Maximum Expected Covering Location Model: Formulation, Properties and Heuristic Solution. *Transportation Science*, Vol. 17, No. 1, 1983, pp. 48–70.

[19] ReVelle, C. and K. Hogan, The Maximum Availability Location Problem. *Transportation Science*, Vol. 23, No. 3, 1989, pp. 192–200.

[20] Larson, R. and A. Odoni, *Urban Operations Research*. Prentice-Hall, Englewood Cliffs, N.J., 1981.

[21] Galvão, R. and R. Morabito, Emergency service systems: The use of the hypercube queueing model in the solution of probabilistic location problems. *International Transactions in Operational Research*, Vol. 15, No. 5, 2008, pp. 525–549.

[22] Sacks, S. and S. Grief, Orlando Police Department uses OR/MS methodology new software to design patrol district. *OR/MS Today*, 1994, pp. 30–42.

[23] Brandeau, M. and R. Larson, Extending and applying the hypercube queueing model to deploy ambulances in Boston. *TIMS Studies in Management Science*, Vol. 22, 1986, pp. 121–153.

[24] Atkinson, J., I. Kovalenko, N. Kuznetsov, and K. Mykhalevych, A hypercube queueing loss model with customer-dependent service rates. *European Journal of Operational Research*, Vol. 191, No. 1, 2008, pp. 223 – 239.

[25] Iannoni, A. P. and R. Morabito, A multiple dispatch and partial backup hypercube queuing model to analyze emergency medical systems on highways. *Transportation Research Part E: Logistics and Transportation Review*, Vol. 43, No. 6, 2007, pp. 755 – 771, challenges of Emergency Logistics Management.

[26] Iannoni, A., R. Morabito, and C. Saydam, A hypercube queueing model embedded into a genetic algorithm for ambulance deployment on highways. *Annals of Operations Research*, Vol. 157, 2008, pp. 207–224, 10.1007/s10479-007-0195-z.

[27] Geroliminis, N., M. Karlaftis, and A. Skabardonis, A spatial queuing model for the emergency vehicle districting and location problem. *Transportation Research Part B: Methodological*, Vol. 43, No. 7, 2009, pp. 798 – 811.

[28] Geroliminis, N., K. Kepaptsoglou, and M. Karlaftis, A hybrid hypercube - Genetic algorithm approach for deploying many emergency response mobile units in an urban network. *European Journal of Operational Research*, Vol. 210, No. 2, 2011, pp. 287–300.

[29] Karlaftis, M., K. Kepaptsoglou, and A. Stathopoulos, Genetic Algorithm-Based Approach for Optimal Location of Transit Repair Vehicles on a Large Urban Network. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 1879, 2004, pp. 41–50.