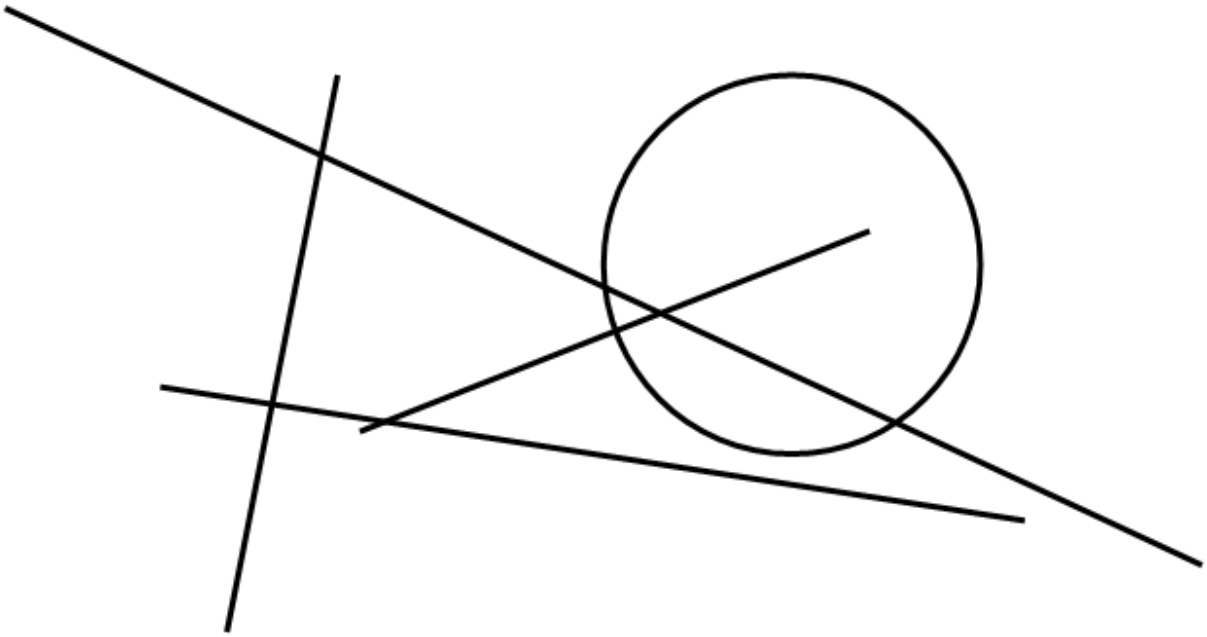

Hypercube Queueing Models for Emergency Response Systems

Burak Boyacı

Nikolas Geroliminis

Urban Transport Systems Laboratory (LUTS)

May 2014

STRC

14th Swiss Transport Research Conference

Monte Verità / Ascona, 14 – 16 May, 2014

Urban Transport Systems Laboratory (LUTS)

Hypercube Queueing Models for Emergency Response Systems

Burak Boyacı
Management Science
Lancaster University Management School
B58A, LA1 4YX, Lancaster, United Kingdom
phone: +44-1524-592732
fax: +44-1524-844885
b.boyaci@lancaster.ac.uk

Nikolas Geroliminis
Urban Transport Systems Laboratory
Ecole Polytechnique Fédérale de Lausanne
GC C2 389, Station 18, 1015 Lausanne,
Switzerland
phone: +41-21-693 24 81
fax: +41-21-693 50 60
nikolas.geroliminis@epfl.ch

May 2014

Abstract

Spatial queueing systems (SQS) can be defined as a type of queue that mobile servers are assigned to travel to the customer and provide on-scene service or the customers travel to service facilities to have service. It has a lot of application areas in literature from emergency response to vehicle repair services, dial-a-ride to paratransit.

In this research, our aim is to find a rapid approach to calculate performance measures of SQS. Our ultimate aim is to utilize this rapid approach as an instance solver inside some optimization algorithms such as simulated annealing (SA) and variable neighborhood search (VNS) to find better location for systems such as ambulances, fire brigades. For this purpose, we have developed two methods to calculate performance measures of an instance of SQS. To check accuracy and efficiency, the approach is compared with simulation results on some instances. Then the two methods are used with SA and VNS to improve server locations. Results show that the approach is promising and can be applied as a tool inside some optimization algorithms.

Keywords

emergency response, spatial queues, hypercube queueing models, approximation algorithms

1 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.

Location-allocation problems are the set problems in the operations research literature that are analyzed extensively. Different than the general location-allocation problems, in emergency response system location problems, both the demand and service time are stochastic which results in congestion and losses in every system. This stochasticity is addressed in many researches but the server based state of the system has been taken into consideration in a few approaches applicable to small-sized systems.

The model proposed by Larson (1974) models the problem as a spatial queueing system (SQS) which is also known as 2^n hypercube queueing model (HQM). In this approach, each emergency response unit is modeled as a server with two states available and busy. If n is the number of servers in the system, there are 2^n states in these models. In Larson (1974)'s model, time spent on the way to service is assumed to be negligible. He assumed, service rate is a function of the dispatched server but not the region receiving the service. This may be acceptable for fire brigades but not for ambulances.

In this research, we are proposing a new 3^n HQM which enables to apply different service rates for different server-region pairs. In 3^n HQM, each server has 3 states: available, busy inside its primary service area and outside its primary service area. However this new model can be intractable even more small sized problems (with more than 8 servers). For this purpose, we also propose and aggregate model namely, 3^n aggregate HQM (AHQM). In this last model, instead of keeping each servers' condition separately at each state, it keeps number of servers in different states at each *bin* (i.e. set of servers). In our research for small cases we use 3^n HQM model whereas for larger instances we implemented an algorithm that combines set of iterations containing partitioning, 3^n HQM and AHQM, mix aggregate hypercube queueing algorithm (MHQA). Both 3^n HQM and MHQA are used to calculate the performance of the locations of emergency vehicles in Euclidean networks. We also implement two optimization methods, variable neighborhood search (VNS) and simulated annealing (SA) to improve the locations of emergency vehicles on two networks.

We continue the paper with literature survey. Afterwards, the existing 2^n HQM and newly proposed two models 3^n HQM and AHQM are defined. In the next section, MHQA is described. The last two sections contain computational results and, conclusions with future research directions.

2 Previous Related Research

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 (1964) 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 (Galvão and Raggi, 1989, Avella and Sassano, 2001) and heuristic methods (Daskin and Haghani, 1984, Schilling *et al.,* 1993) as well. Mladenović *et al.* (2007) wrote 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.* (1971) and Church and ReVelle (1974) approach coverage models from two different directions. 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 (Marianov and ReVelle, 1996). Daskin and Stern (1981) and Gendreau *et al.* (1997) altered the MCLP and proposed two models that maximize the number of regions that are covered more than once.

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 (1968) in which the main aim is to relocate a warehouse in such a way that maximizes the profit in a finite horizon. Scott (1971) works with the extension of this problem with more than one facilities. Schilling (1980) 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, 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 (1961), Daskin (1983), ReVelle and Hogan (1989) and, Marianov and ReVelle (1996) are some of the important articles written in this literature.

Larson (1974) 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 (Larson and Odoni, 1981). 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" (Galvão and Morabito, 2008). HQM models the current state as a continuous-time Markov process but does not determine the optimal configuration. Police patrolling (Sacks and Grief, 1994) and ambulance location (Brandeau and Larson, 1986) are two applications modeled by HQM. Marianov and ReVelle (1996) extended the MALP and developed *queueing maximum availability location problem*.

The first model proposed by Larson (1974) 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.* (2008) proposed a partial 3^n HQM that assumes different service rates for each server in the system with equal interdistrict or intradistrict responses. Iannoni and Morabito (2007) and Iannoni *et al.* (2008) 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.* (2009) integrate the location and distracting decisions in the same optimization and solve the problem by using steepest descent for up to 10 servers. Geroliminis *et al.* (2011) extended the previous work to deal with larger instances with spatially homogeneous demand. They use a genetic algorithm that is using an entity called *superdistrict* which is similar to bin. However, they have not taken interactions between superdistricts into consideration which seems important as it is shown later in this paper.

3 Hypercube Queueing Models

In this section, we start by describing the Larson (1974)'s model. Then we describe the two models we propose: 3^n HQM and 3^n AHQM.

3.1 2^n Hypercube Queueing Model

2^n HQM proposed by Larson (1974). Each state name contains n binary variables where n stands for the number of servers in the system. i^{th} digit of the state name contains condition of server i: available (0) or busy (1). For each region, which is named as *atom* in HQM literature, there exists a priority list of servers. Atoms are served by the available server that has the highest priority in their list. If there are no available server that can serve the atom, either the request is lost or joins to a queue to be served later depending on the system structure. Both interarrival times of incidents at each atom (λ_j) and service times of servers (μ_i) are exponentially distributed. A transition diagram with three servers for a 2^n HQM can be seen in Figure 1

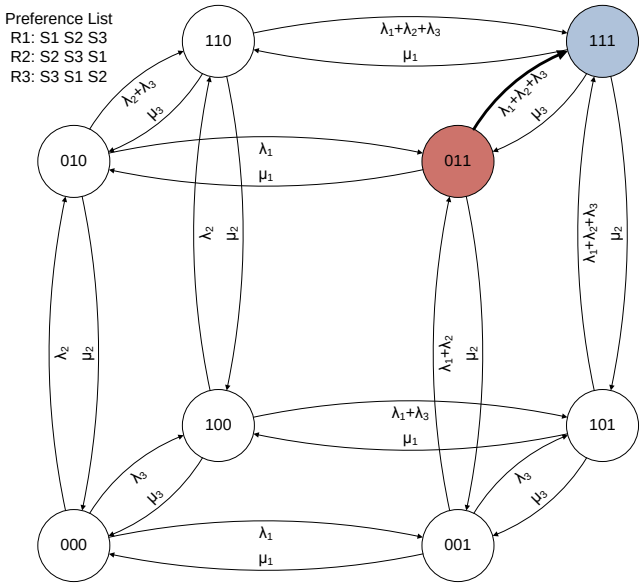

Figure 1: Larson (1974)'s 2^n HQM for three servers with equal intra and interdistrict service rates (μ_i). State "011", "111" and the transition connecting them is shown with different colors.

3.2 3^n Hypercube Queueing Model

In our first proposed model, we define three states for each server. More precisely, dispatching servers to primary service area (intradistrict) and secondary service area (interdistrict) are

differentiated from each other. We use three states: available (0), busy with intradistrict (1) and busy with interdistrict (2). A transition diagram of a 3^n HQM with 3 servers can be seen in Figure 2. In this figure, μ_i and μ_i' stands for the intradistrict and interdistrict service rates of server i respectively and λ_j is the interarrival rate in atom j. Not that, in this system the server has always priority for its own intradistrict area. However, this does not prevent to have states such as "222".

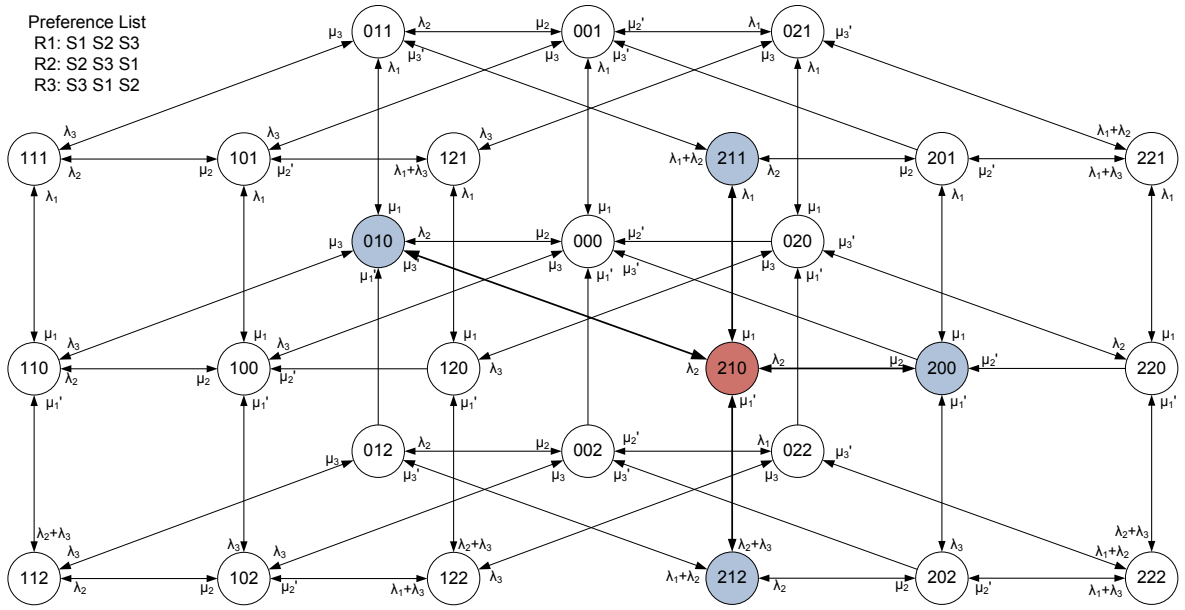

Figure 2: 3^n HQM model for two servers with different intra (μ_i) and interdistrict (μ_i') service rates. State "210", states directly connected to it and transitions are colored differently.

To have an illustrative example, the transition equation for the state "210" is given below. In state "210", the rightmost digit ("0") represents the first server and shows it is available; the middle digit ("1") shows the condition of the second server is busy with intradistrict response and; the leftmost digit ("2") indicates the third server is busy with interdistrict response. Please note $\mathbb{P}_r$ shows the steady state probability of state r.

$$\mathbb{P}_{210}\left(\lambda_1+\lambda_2+\lambda_3+\mu_3'+\mu_2\right)=\mu_1\mathbb{P}_{211}+\mu_1'\mathbb{P}_{212}+\lambda_2\mathbb{P}_{200}+\lambda_2\mathbb{P}_{010} \tag{1}$$

3.3 3^n Aggregate Hypercube Queueing Model

As stated above, the size of 3^n HQM makes it intractable even for medium sized problems. For his reason, we propose 3^n AHQM which is less accurate but more efficient than 3^n HQM. In this new model, we propose a new concept called *bin* instead of individual servers. Bins can be

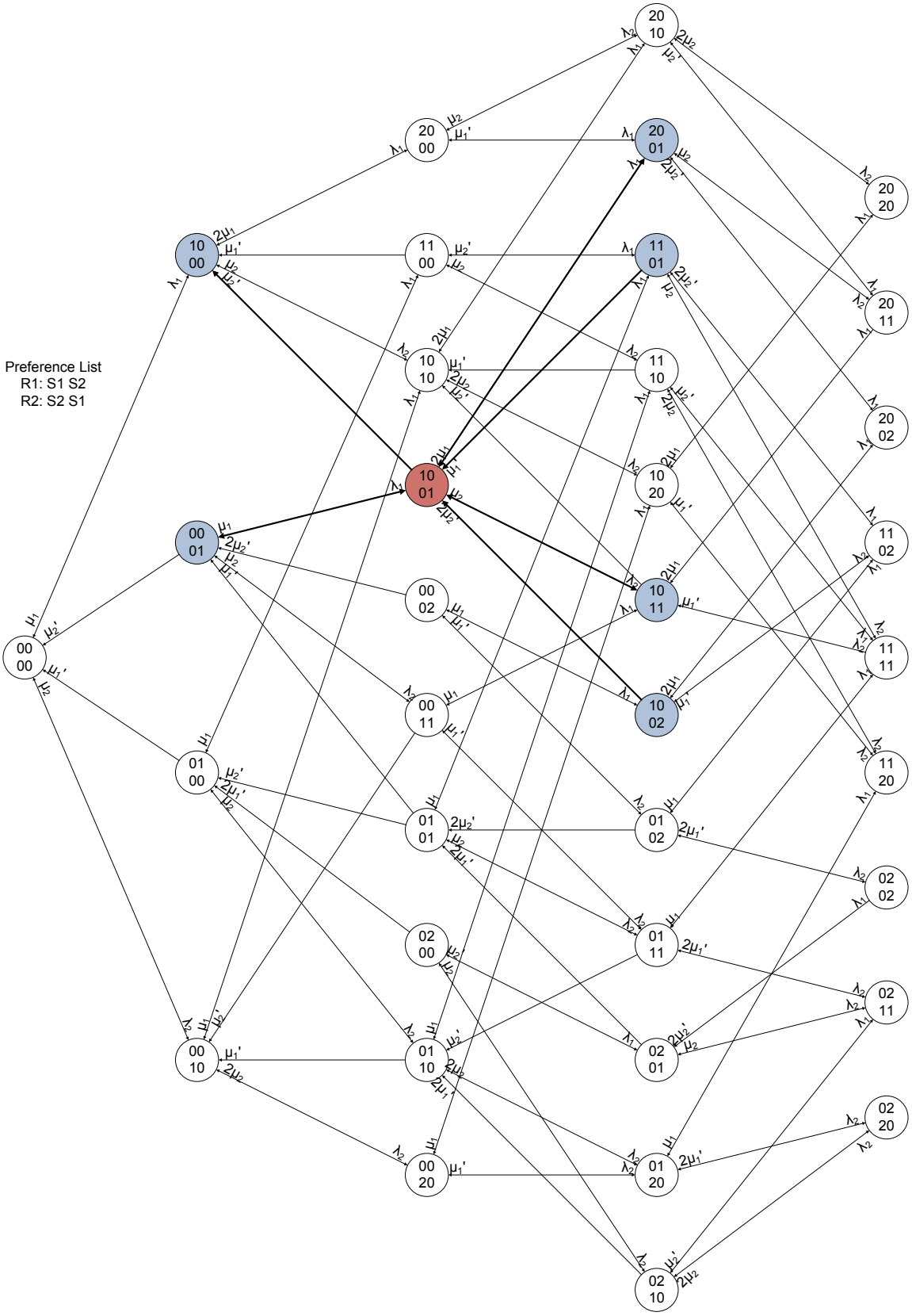

Figure 3: 3^n AHQM for two bins containing two servers in each bin with different intra (μ_b) and interdistrict (μ'_b) service rates, and primary demand areas (λ_j). State "10/01" and states connected to it are filled with different colors to show an example of transition equations.

regarded as group of servers. Instead of keeping each servers' conditions separately in the states, the number of servers in different states at each bin is kept. Since the total number of servers is known for each bin, we represent the condition of each bin with two values. The number of servers in intradistrict and interdistrict response. For μ_b and μ'_b stands for the intradistrict and interdistrict service rates of each server in bin b respectively and λ_j is the interarrival rate in atom j, the transition diagram for a system with two bins with two servers each can be illustrated as in Figure 3.

In Figure 3 the transition equation for the state "10/01" (shown with red) can be written as in Equation 2. Please note, the first line of the state name "10" shows the condition of the first bin. The value on the left ("1") shows the number of busy servers in intradistrict response and the value on the right ("0") shows the number of busy servers in interdistrict response. Similarly, "0" and "1" shows the number of busy servers in intra and interdistrict responses in bin 2. To sum up, in state "10/01" there are one busy server in intradistrict response in bin 1 and one busy server in interdistrict response in bin 2.

$$\begin{aligned} &\mathbb{P}_r\left[\mathbb{1}\left(\exists n : \tilde{T}(r,b,\text{free}) \neq 0\right)\sum_j \lambda_j + \sum_b \tilde{T}(r,b,\text{intra})\mu_b + \sum_b \tilde{T}(r,b,\text{inter})\mu'_b\right] \\ = &\sum_{\substack{q,b:\\ \tilde{D}(q,r,b,\text{intra})=1}} \mathbb{P}_q\mu_b + \sum_{\substack{q,b:\\ \tilde{D}(q,r,b,\text{inter})=1}} \mathbb{P}_q\mu'_b + \sum_{\substack{q,b:\\ \tilde{D}(r,q,b,\text{intra})=1}} \mathbb{P}_b\sum_{j\in R_b}\lambda_j + \sum_{\substack{q,b:\\ \tilde{D}(r,q,b,\text{inter})=1}} \mathbb{P}_b\sum_{j\in S(r,b)}\lambda_j \end{aligned} \tag{2}$$

For a 3^n AHQM, if C_b equals to the number of servers in bin b, total number of states equals $\prod_b \frac{(C_b+2)(C_b+1)}{2}$ which is far less than 3^n. For most of the cases with two bins, this value is even less than 2^n (i.e. for the cases with 8 or more servers). For instance, the system with 20 servers has over one million3 states in 2^n and 3.5 billion states in 3^n HQM whereas a 3^n AHQM of two bins with 10 servers each has only 4356 states. In the next section, we describe the MHQA that utilizes the two new 3^n models, i.e. 3^n HQM and 3^n AHQM.

4 Mix Aggregate Hypercube Queueing Algorithm

The increase in the number of states makes 3^n HQM not applicable to real life instances. Because of that, we propose 3^n AHQM. However, 3^n should be applied in a way that will keep the results in some accuracy level in an efficient way. In this section we will briefly describe this method. Further information will be given in a journal paper which is under review right now.

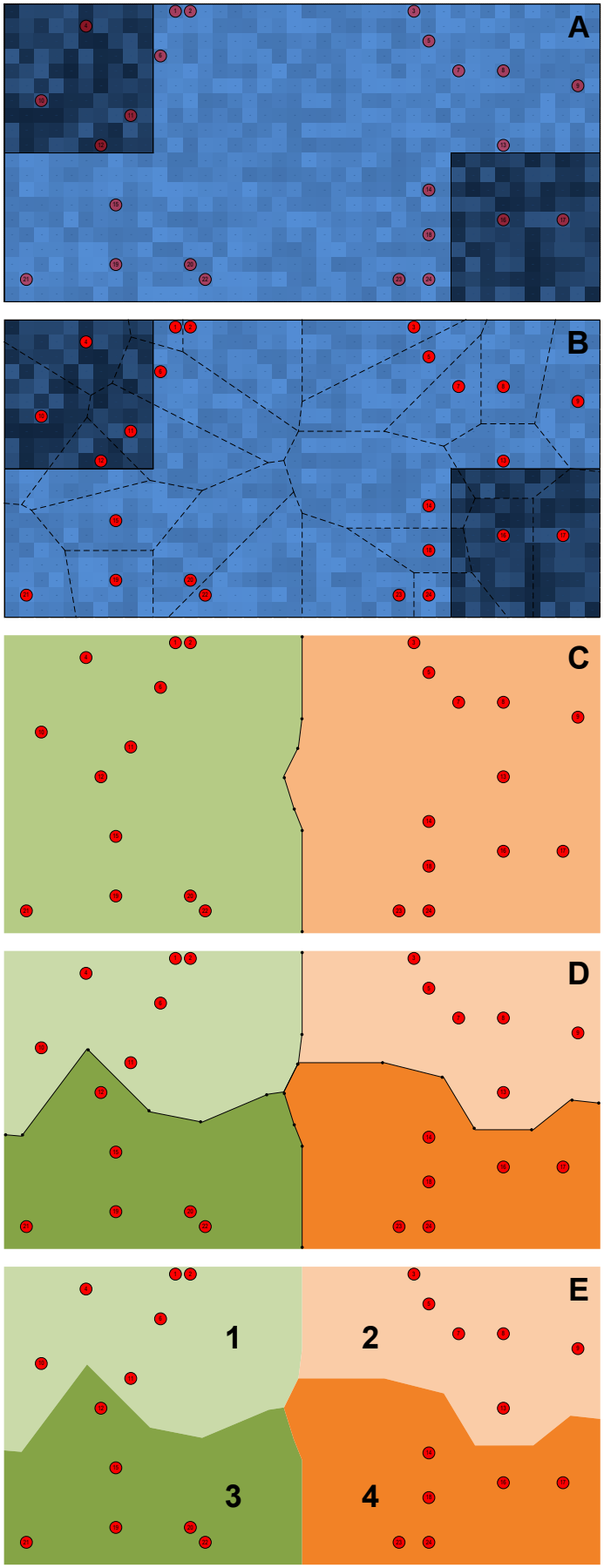

Figure 4: An illustration of the partitioning approach.

An illustration of the procedure can be seen in Figure 4. Figure 4a shows the whole problem area with all servers (red dots). Dark color represents atoms of high demand and lighter color the ones of lower demand. Figure 4b shows the primary areas of responsibilities of each server estimated with a Voronoi approach based on Euclidean distance. Figures 4b-d shows the results of the sequential partitioning method which is briefly described in this section.

In this method, the core subregions are merged to larger *compound* subregions, which are modeled as a 3^n AHQM. When compound regions are merged to form larger subregions, again 3^n AHQM is used. This process is repeated until the whole problem area is covered. Note that, at each merging step, only two subregions are used.

As stated above we developed MHQA to have accurate results in reasonable time. In order to have that, we also need a partitioning algorithm that partitions the whole problem area as we want. We try to develop a partitioning algorithm that satisfies the following properties:

1. The number of servers in each partition should be the parameter of the partitioning algorithm. We need to set the number of servers in each subregion. There is a maximum size that is efficiently solvable with both hypercube models and over partitioning (i.e. using more partitions than needed) decreases the accuracy of the final result.
2. Servers in the same partition should be adjacent to each other. This prevents disconnected atoms and helps to have connected subregions which improves accuracy of the method.
3. Partitioning should be sequential in order to apply the approximation algorithm.
4. Partitioning should be efficient. Our aim in developing an approximation algorithm is to evaluate instances in an optimization framework. To do that, we need efficient algorithms in all steps of the evaluation.

For this purpose an algorithm is developed that generates “cuts” composed of paths on the problem area and creates subregions. This algorithm is applied on the Voronoi diagram of server locations. Afterwards, we solve one or more flow problems that divides the whole problem area into two or more partitions. Readers who are interested in the mathematical model of this approach can refer to our journal paper which is under review right now.

5 Computational Results

After describing our methods briefly, in this section we first evaluate the accuracy of the model described in the previous two sections, by comparing them with a discrete event simulation. Then these methods are combined with two optimization methods variable neighborhood search

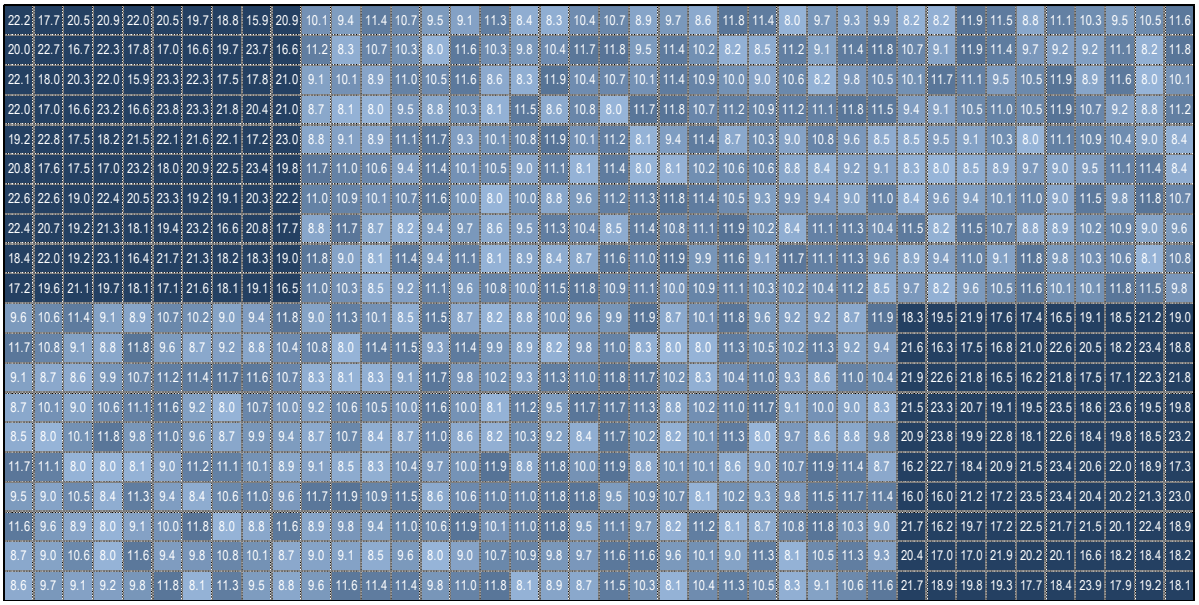

Figure 5: The demand distribution of the network used in experiments.

(VNS) (Mladenović and Hansen, 1997) and simulated annealing (SA) (Kirkpatrick *et al.,* 1983) to find better facility locations. All of the algorithms in this work are developed under C# .NET environment. For partitioning algorithm IBM ILOG Cplex 12.4 is used through Concert user interface. MATLAB 7.9 through MATLAB Automation Server interface is used for matrix operations. All experiments are conducted on a PC with Intel Core2 Quad 3.00 Ghz processor.

We use an experimental network given in 5. Each square in this figure shows $q\text{km}^2$ area. The value at each square shows 10^4 times the ratio of the arrival rate to total arrival rate. Euclidean norm is used in distance calculation but other norms can also be used with minor changes in the algorithm. It is assumed that total service time equals to the sum of total travel time (going to scene and coming back from scene) and on scene service time.

5.1 Accuracy of 3^n HQM

In this part, MHQA is compared with discrete event simulation for a case with 12 servers. We tested following instances with three demand (5, 15, 45 instance/hour), average on-scene service time (5, 10, 20min) and accessibility distance (10, 15, 20kms) for each demand distribution which makes 27 scenarios in total and compare lost rates. We set the speed of servers to 1km/min. On-scene service time and inter-arrival times are distributed exponentially. 500 random instances with 12 facilities were generated for each setting. We used partitions of 6 servers in approximation algorithm. Both simulation and our method are run over over experimental network. The percentage of error in loss rates are reported in Figure 6. The ratio of the difference of lost rate between each simulation result and the MHQA result to the lost rate

found by simulation are reported as error.

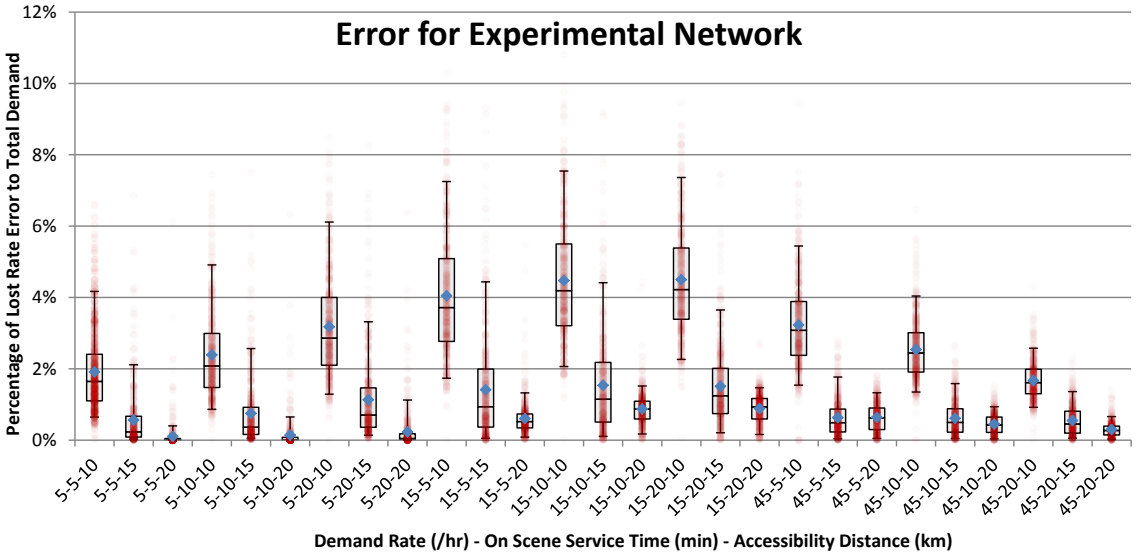

Figure 6: The ratio of difference between the loss rates calculated by simulation and the MHQA.

25 parallel simulation instances were created with 11 batches simulating length of 50 days each. The mean of the last 10 batches of all 25 parallel simulations are reported. Length of the simulations are selected in a way that calculated values have tight enough confidence intervals to guarantee steady state.

For each instance, simulation took around 25s whereas our method took around 7s on average. The comparison showed that compared to simulation, our method gives results with acceptable error (less than 5% error on average and 10% in the worst case) in less run time (28% of simulation run time). This error gets even less for increased server range, for the scenarios with the range of 20km, average error is less than 1% and in 95% of the cases error is less than 2%. Furthermore, simulation needs to run longer to have accurate results if more detailed performance measures (e.g. loss rate per number of busy servers) are needed.

5.2 Heuristics for Better Location of Servers

In this section, our exact 3^n HQM (for cases with less than or equal to 8 servers) and mix aggregate hypercube (for cases with more than 8 servers) algorithms are tested inside two heuristic approaches to identify close-to-optimum locations of servers: variable neighborhood search (Mladenović and Hansen, 1997) and simulated annealing (Kirkpatrick *et al.,* 1983). Both methods are initialized with the maximum expected coverage location model (MEXCLP) (Daskin, 1983). MEXCLP is selected because it is efficient and gives good results. In VNS algorithm, it is assumed that if two instances' all but one servers are in different locations, they

# servers	on scene serv. time	demand	MEXCLP	VNS		SA	
				value	% impr.	value	% impr.
6	5	5	0.076	0.055	26.72	0.055	26.67
		8	0.725	0.594	18.09	0.594	18.09
		10	1.691	1.452	14.1	1.452	14.1
	20	5	0.432	0.377	12.63	0.377	12.67
		8	2.074	1.926	7.11	1.926	7.11
		10	3.626	3.439	5.16	3.439	5.16
7	5	5	0.022	0.013	39.27	0.013	39.27
		8	0.343	0.252	26.29	0.252	26.29
		10	0.994	0.775	22	0.775	22
	20	5	0.207	0.167	19.34	0.168	19.15
		8	1.436	1.267	11.81	1.27	11.62
		10	2.813	2.566	8.78	2.569	8.69
8	5	5	0.005	0.003	47.51	0.003	47.51
		8	0.129	0.098	23.94	0.098	23.94
		10	0.483	0.387	19.82	0.387	19.82
	20	5	0.084	0.065	22.77	0.068	19.71
		8	0.894	0.776	13.21	0.798	10.73
		10	2.035	1.826	10.28	1.848	9.21
12	5	10	0.003	<1E-3	84.54	<1E-3	84.54
		16	0.263	0.054	79.36	0.054	79.36
		20	1.272	0.472	62.90	0.472	62.9
	20	10	0.200	0.095	52.49	0.099	50.18
		16	2.680	2.062	23.03	2.064	22.99
		20	5.714	5.004	12.42	4.989	12.68
16	5	16	0.004	0.001	87.32	0.001	87.32
		20	0.068	0.008	87.92	0.008	87.92
		30	2.423	0.786	67.58	0.786	67.58
	20	16	0.551	0.282	48.85	0.290	47.37
		20	2.203	1.458	33.84	1.448	34.29
		30	9.947	9.058	8.94	9.172	7.8

Table 1: The best lost rate found by MEXCLP, VNS and SA algorithms for the experimental network given in Figure 5.

are neighbors. In every iteration, a randomly selected server's location is changed. We use the same neighborhood structure in SA algorithm. We set the starting "temperature" coefficient to 1 and increase it by 10% in every 20 iterations. Temperature is assumed to be the division of temperature coefficient with the average lost rate in every iteration. We have applied 3^n HQM for cases with 6,7 and 8 servers for total arrival rates of 5, 8, 10, 15 and 20 requests/hour. For cases with 12 and 16 servers, arrival rates are doubled (i.e. 10, 16, 20, 30, 40 requests/hour), problems are solved by MHQA with two partitions of equal size. Run times for the former three (i.e. 6, 7 and 8-server) cases are set to one hour, whereas the latter two (i.e. 12 and 16-server) cases are run for four hours. We have applied two different on scene service times: 5 and 20 minutes. For all cases, maximum accessibility distance is set to 30km. Found minimum lost rate and percent lost rate improvements after MEXCLP by both heuristics (i.e. VNS and SA) for cases with realistic lost ratios (ratio of lost rate to the total arrival rate) can be seen in table 1.

From the experiments, we observed that there is an improvement of more than 20% on average for the lost rates over MEXCLP. We have not observed any significant difference between VNS and SA. Since our primary goal in this research is to test the applicability of hypercube models inside search algorithms, we have not searched for parameters that may give better final results for both VNS and SA. One can also observe that the lost rates dramatically increase with the increase of on scene service time. Small increase in demand has also considerable influence on the lost rate. Queueing systems are unpredictably complex and need custom-built algorithms to be tested. We also observe that the performance of SA and VNS against MEXCLP get better with the increase in the number of servers. Last but not least, careful readers might realize that for the same value we have calculated different percent improvements. This is the consequence of showing results with limited precision. We also noticed (not shown here) that even after 30 minutes (instead of 4 hours) the VNS and SA methods provide similar improvement with a 4 hour run.

6 Conclusions

In this short paper, we have briefly describe two new 3^n hypercube models and two algorithms that utilize these two methods that we propose. For evaluating our methods, we first compare our approach's results with the results of a discrete event simulation and show the accuracy of our approach for different parameters on an experimental network. To see the applicability of our two approximation algorithms (3^n HQM and MHQA) inside an optimization framework, the two methods are implemented with variable neighborhood search and simulated annealing. Experiments have shown that, although hypercube queueing models are not optimization models, they can be utilized inside optimization frameworks.

7 References

Atkinson, J., I. Kovalenko, N. Kuznetsov and K. Mykhalevych (2008) A hypercube queueing loss model with customer-dependent service rates, *European Journal of Operational Research*, **191** (1) 223 – 239, ISSN 0377-2217.

Avella, P. and A. Sassano (2001) On the p-median polytope, *Mathematical Programming*, **89**, 395–411, ISSN 0025-5610. 10.1007/PL00011405.

Ballou, R. H. (1968) Dynamic warehouse location analysis, *Journal of Marketing Research*, **5** (3) pp. 271–276, ISSN 00222437.

Brandeau, M. and R. Larson (1986) Extending and applying the hypercube queueing model to deploy ambulances in boston, *TIMS Studies in Management Science*, **22**, 121–153.

Church, R. and C. ReVelle (1974) The maximal covering location problem, *Papers in Regional Science*, **32** (1) 101–118.

Daskin, M. (1983) A maximum expected covering location model: Formulation, properties and heuristic solution, *Transportation Science*, **17** (1) 48–70.

Daskin, M. and E. Stern (1981) A hierarchical objective set covering model for emergency medical service vehicle deployment, *Transportation Science*, **15** (2) 137–152.

Daskin, M. S. and A. Haghani (1984) Multiple vehicle routing and dispatching to an emergency scene, *Environment and Planning A*, **16** (10) 1349–1359.

Galvão, R. and R. Morabito (2008) Emergency service systems: The use of the hypercube queueing model in the solution of probabilistic location problems, *International Transactions in Operational Research*, **15** (5) 525–549, ISSN 1475-3995.

Galvão, R. and L. Raggi (1989) A method for solving to optimality uncapacitated location problems, *Annals of Operations Research*, **18**, 225–244, ISSN 0254-5330. 10.1007/BF02097805.

Gendreau, M., G. Laporte and F. Semet (1997) Solving an ambulance location model by tabu search, *Location Science*, **5** (2) 75–88.

Geroliminis, N., M. Karlaftis and A. Skabardonis (2009) A spatial queuing model for the emergency vehicle districting and location problem, *Transportation Research Part B: Methodological*, **43** (7) 798 – 811, ISSN 0191-2615.

Geroliminis, N., K. Kepaptsoglou and M. Karlaftis (2011) A hybrid hypercube - genetic algorithm approach for deploying many emergency response mobile units in an urban network, *European Journal of Operational Research*, **210** (2) 287–300, ISSN 0377-2217.

Hakimi, S. (1964) Optimum locations of switching centers and the absolute centers and medians of a graph, *Operations Research*, **12** (3) 450–459, ISSN 0030364X.

Iannoni, A., R. Morabito and C. Saydam (2008) A hypercube queueing model embedded into a genetic algorithm for ambulance deployment on highways, *Annals of Operations Research*, **157**, 207–224, ISSN 0254-5330. 10.1007/s10479-007-0195-z.

Iannoni, A. P. and R. Morabito (2007) A multiple dispatch and partial backup hypercube queuing model to analyze emergency medical systems on highways, *Transportation Research Part E: Logistics and Transportation Review*, **43** (6) 755 – 771, ISSN 1366-5545. Challenges of Emergency Logistics Management.

Kirkpatrick, S., C. Gelatt and M. Vecchi (1983) Optimization by simulated annealing, *Science*, **220** (4598) 671–680.

Larson, R. (1974) A hypercube queuing model for facility location and redistricting in urban emergency services, *Computers & Operations Research*, **1** (1) 67 – 95, ISSN 0305-0548.

Larson, R. and A. Odoni (1981) *Urban Operations Research*, Prentice-Hall, Englewood Cliffs, N.J.

Manne, A. (1961) Capacity expansion and probabilistic growth, *Econometrica*, **29** (4) 632–649, ISSN 00129682.

Marianov, V. and C. ReVelle (1996) The queueing maximal availability location problem: A model for the siting of emergency vehicles, *European Journal of Operational Research*, **93** (1) 110–120, ISSN 0377-2217.

Mladenović, N., J. Brimberg, P. Hansen and J. Moreno-Pérez (2007) The p-median problem: A survey of metaheuristic approaches, *European Journal of Operational Research*, **179** (3) 927 – 939, ISSN 0377-2217.

Mladenović, N. and P. Hansen (1997) Variable neighborhood search, *Computers & Operations Research*, **24** (11) 1097–1100.

ReVelle, C. and K. Hogan (1989) The maximum availability location problem, *Transportation Science*, **23** (3) 192–200, August 1989.

Sacks, S. and S. Grief (1994) Orlando police department uses or/ms methodology new software to design patrol district, *OR/MS Today*, 30–42.

Schilling, D., V. Jayaraman and R. Barkhi (1993) A review of covering problems in facility location, *Location Science*, **1** (1) 25–55.

Schilling, D. A. (1980) Dynamic location modeling for public-sector facilities: A multicriteria approach, *Decision Sciences*, **11** (4) 714–724, ISSN 1540-5915.

Scott, A. (1971) Dynamic location - allocation systems: some basic planning strategies, *Environment and Plann*, **3** (1) 73–82.

Toregas, C., R. Swain, C. ReVelle and L. Bergman (1971) The location of emergency service facilities, *Operations Research*, **19** (6) 1363–1373.