Pesquisa Operacional (2011) 31(2): 271-299

Printed version ISSN 0101-7438 / Online version ISSN 1678-5142
www.scielo.br/pope

# A TUTORIAL ON HYPERCUBE QUEUEING MODELS AND SOME PRACTICAL APPLICATIONS IN EMERGENCY SERVICE SYSTEMS

Fernando Chiyoshi[1], Ana Paula Iannoni[2] and Reinaldo Morabito[3*]

Received February 19, 2010 / Accepted December 27, 2010

**ABSTRACT.** This paper presents some extensions and applications of hypercube queueing models to describe server-to-customer type Emergency Service Systems. The classical hypercube is a well-known spatially distributed queueing model effective in analyzing these systems, based on Markovian analysis approximations. Experience has shown that each real life Emergency Service System may have its own unique characteristics so that each system may require a particular hypercube queueing model incorporating those characteristics. Some of these distinctive characteristics are considered in the extensions presented in this tutorial such as dispatch policy based on random selection of the server to take an incoming call; partial cooperation among servers whereby depending on where the call is coming from, some servers cannot take the call; servers with additional workload coming from walk-in nonemergency customers such as in customer-to-server systems; existence of calls requiring the dispatch of more than one server; existence of more than one type of servers in the system, for instance paramedical and medical units in emergency medical systems. In this study, we present a set of these models based on the smallest non-trivial service systems. For each model, the construction of the system of equations for equilibrium hypercube state probabilities and the evaluation of particular operational characteristics are described.

**Keywords**: hypercube queueing model, emergency service systems, toy-models, server-to-customer systems.

## 1 INTRODUCTION

Emergency Service Systems (ESS) provide emergency assistance to incidents, protecting and ensuring public health and safety, and they can be classified as either customer-to-server systems or server-to-customer systems. In the first case, service units are considered immobile in the sense that customers must travel to where servers are located, whereas in the latter the servers

*Corresponding author
[1]Department of Production Engineering, COPPE/Universidade Federal do Rio de Janeiro, RJ, Brazil.
E-mail: chiyoshi@pobox.com
[2]Ecole Centrale Paris, Laboratoire Genie Industriel, France. E-mail: iannoni93@hotmail.com
[3]Department of Production Engineering, Universidade Federal de São Carlos, SP, Brazil. E-mail: morabito@ufscar.br

are mobile, that is, they travel to the scene of the emergency. In this case, the system is also called an emergency response system; some examples are requests for urgent medical assistance, police patrol as well as fire combat units and emergency repairs. In these systems, mobile servers are more accurately modeled as distinguishable from one another in the sense that the servers are spatially distributed in the region and have different characteristics (*e.g.*, different preferential regions and mean service times), they share the system workload due to cooperation, and their workloads may be different and may vary according to the location of the servers.

Surveys on probabilistic location models that use mobile servers in ESS can be found, for example, in Larson & Odoni (1981), Kolesar & Swersey (1986), ReVelle (1989), Swersey (1994), Owen & Daskin (1998), Brotcorne *et al.* (2003), Marianov & Serra (2003), Goldberg (2004), ReVelle & Eiselt (2005) and Galvao & Morabito (2008). The hypercube queueing model developed by Larson (1974) and extended by other authors (Swersey, 1994) is an effective descriptive model for designing and planning server-to-customer ESS. The basic idea of the model is to expand the state space description of a queueing system with multiple servers in order to represent each server individually and incorporate more complex dispatch policies. The model takes into account geographical and temporal complexities of the region under consideration and it is appropriate to analyze coordinated (centralized) systems, where the user requiring service calls the central station of the system and the system manager dispatches a server to service the call. If there is no server available, the call either enters a waiting line until a server becomes available, or it is transferred to another ESS.

The name hypercube derives from the state space describing the status of the servers. Each server can be free (0) or busy (1) in a given time instant. A particular state of the system is given by the entire listing of servers that are free and busy. For example, the state {110} corresponds to a 3-server system, with server 1 free and servers 2 and 3 busy (note that {110} describes the state of the servers from right to left). Taking this into account, the state space is given by the vertices of a cube. If the system has more than three servers, we have a hypercube (Larson, 1974). Given the system configuration, the hypercube model is able to evaluate a variety of performance measures relevant for decision-making, either region-wide or for each server or region. These include server workloads, mean user response times, fraction of dispatches of each server to each region, among others. In solving location problems in which the best locations of the servers are searched, the hypercube model is used as a descriptive tool to evaluate the quality of a given set of locations.

Some examples of applications of the hypercube queueing model include the location of ambulances in urban areas (Brandeau & Larson, 1986; Burwell *et al.*, 1993; Takeda *et al.*, 2007; Rajagopalan *et al.*, 2008) and ambulances and service patrol vehicles on highways (Mendonça and Morabito, 2001; Iannoni *et al.*, 2008, 2009; Geroliminis *et al.*, 2009, 2011); the deployment of police patrol units (Chelst and Barlach, 1981; Larson and Macknew, 1982; Sacks and Grief, 1994); the design of repair services related to interruptions in the distribution of electrical energy (Albino, 1994), and the programs for visits by the social service (Larson & Odoni, 1981). The hypercube has also been considered as a deployment model for response to terrorism attacks and

other major emergencies (Larson, 2004). Other references related to applications and extensions of the hypercube model can be found in Halpern (1977), Jarvis (1985), Batta *et al.* (1989), Swersey (1994), Chiyoshi *et al.* (2000, 2003), Galvão *et al.* (2003, 2005), Saydam & Aytug (2003), Costa (2004), Iannoni & Morabito (2007), Luque (2007) and Atkinson *et al.* (2006, 2008).

Each system involved in the practical applications of the hypercube queueing model has its own distinctive characteristics. The objective of this paper is to present a set of models to show how these characteristics are handled and how the associated output measures are evaluated. To that end, we use the smallest, non-trivial structures that incorporate the characteristics of systems under analysis. These structures, referred to as "toy-models", are used to provide useful insights into problems of interest, but are mostly unable to directly address the complexity of real world ESS.

We start discussing single dispatch hypercube models in which it is assumed that the arriving calls at the ESS require the dispatching of only one server (Section 2). In Section 2.1, we review the basic single dispatch model considering both infinite and finite (possibly zero) capacity waiting lines with either homogeneous or non-homogeneous servers. Then the centralized model with random dispatches is described (Section 2.2), where all servers share the same home location and there are no preferences as to the server to be dispatched to service a call, which is inspired in the case of an urban SAMU (*Service d'Aide Médicale Urgente*) studied in Takeda *et al.* (2007). In Section 2.3, we present the partial backup model in which it is assumed that each region of the ESS is the home location of a server, and that a server can only service calls from its preferential region and from the second closest region. The model considers a zero capacity waiting line system with homogeneous or non-homogeneous servers. This is the case of some emergency medical systems on motorways, as for example in the SAU's (*Sistema de Atendimento ao Usuário* – highway's user response system) studied in Mendonça & Morabito (2001), Iannoni & Morabito (2007) and Iannoni *et al.* (2009, 2011).

Then in Section 3 we analyze multiple dispatch hypercube models which consider that some arriving calls at the ESS require the simultaneous dispatching of two or more servers. Some examples appear in emergency medical systems, police patrol systems and fire emergency systems. For instance, in emergency medical systems, the multiple dispatching of ambulances is needed especially when incidents involving several victims require two or more ambulances for assistance and transport. In policy patrol systems, frequently two police patrols with one or two police officers are sent to the same event to work together, for example, one offering cover to the other in case of danger. In fire control systems, more than one unit is usually dispatched to control the fire according to the proportion of the incident and the number of required pieces of equipment and personnel (Swersey, 1994).

In Section 3.1, we review the basic multi-dispatch model as firstly studied in Chelst & Barlach (1981) for zero capacity waiting line police patrol systems with either total or partial backup. Then in Section 3.2, we describe the multi-dispatch model with a third status of each server, in which servers can be busy in other activities than only emergency calls and events assigned by the call operations center, or they can be busy giving assistance to patients at their base (*i.e.*, walk-in

callers). This case appears, for example, in police deployment systems where a police patrol could be busy in an event as a result of something illegal that the police officer sees from the patrol car (*i.e.*, an event that was not received by the call operations center, namely PIA – patrol initiated activity), as studied in Larson & Mcknew (1982). Another example appears in highway emergency medical systems where a third status of the ambulances refers to eventual assistance to patients requesting service at the ambulance bases, as studied in Iannoni & Morabito (2007) and Atkinson *et al.* (2008).

In Section 3.3, we consider the multi-dispatch model with differentiated servers. In several ESS, a dispatch policy that considers different types of servers according to the type of vehicle, personnel or equipment required by the call type are common. For example, in emergency medical systems in urban areas, the ambulances can be either advanced support vehicles (ASV) or basic support vehicles (BSV), and the calls are divided into two classes: advanced calls (preferentially serviced by an ASV) and basic calls (preferentially serviced by a BSV) (Brandeau & Larson, 1986; Takeda *et al.*, 2007). Another example appears in emergency medical systems on highways with medical vehicles and ambulances. The medical vehicles transport specialized medical personnel (*e.g.*, doctors, nurses and rescuers), some basic medications and tools for pre-clinical care. Conversely, the ambulances transport the patients, medical personnel, clinical and pre-hospital care equipment and other heavier equipment (*e.g.*, hardware breakers, fire control equipment, etc.), as studied in Iannoni & Morabito (2007). Finally, in Section 4 we present some concluding remarks and discuss some opportunities for future research.

## 2 THE SINGLE DISPATCH HYPERCUBE MODEL

### 2.1 The basic model

For the purpose of building the toy-models, we take a simple network consisting of only three atoms (*i.e.*, pre-defined sub regions of the ESS region) connected by a one-way ring road. It is assumed that the centroids of the atoms are located at the vertices of an equilateral triangle with sides of unit length, as shown in Figure 1. The inter-atoms travel distance matrix for this network is shown in Table 1.

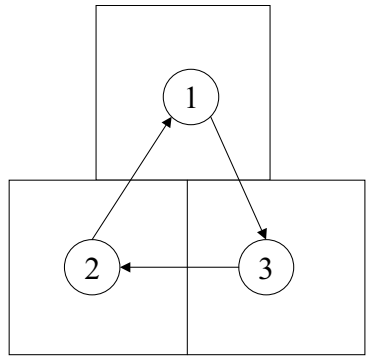


**Figure 1** – 3-atom network for the toy-models.

**Table 1** – Inter-atom travel distances.

| From Atom | To Atom | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| 1 | 0 | 2 | 1 |
| 2 | 1 | 0 | 2 |
| 3 | 2 | 1 | 0 |

We start by considering the basic hypercube queueing model as known in the literature (Larson, 1974; Larson & Odoni, 1981). For this model, we assume that each atom is the home location of a server, and a fixed-preference dispatch policy, with preferences set according to shortest distances is in use. The preference matrix is shown in Table 2.

**Table 2** – Server dispatch preferences.

| Atom | Server Preferences | | |
|---|---|---|---|
| | $1^{st}$ | $2^{nd}$ | $3^{rd}$ |
| 1 | 1 | 2 | 3 |
| 2 | 2 | 3 | 1 |
| 3 | 3 | 1 | 2 |

The hypercube state probabilities are defined by a set of flow-balancing equations built around the hypercube states. To that end, we represent these states through a triad of $0/1$ variables (0 – free, 1 – busy) of the form $\{ijk\}$, each one associated with a server, with the correspondence of variable to the server made from right to left. In addition, we define:

- $\lambda_j$ as the arrival rate of calls from atom $j$;
- $\mu_i$ as the service rate of server (or unit) $i$;
- $\lambda = \lambda_1 + \lambda_2 + \lambda_3$ as the total arrival rate;
- $\mu = \mu_1 + \mu_2 + \mu_3$ as the total service rate.

In order to build the balance equation for a given state, say $\{011\}$, we argue as follows: the system leaves state $\{011\}$ if a call arrives, or a service is completed by server 1 or 2. The rate associated with this transition is $(\lambda + \mu_1 + \mu_2)P\{011\}$. Conversely, the system enters state $\{011\}$: (i) from state $\{001\}$ when a call arrives from atoms 1 or 2 (in accordance with Table 2); (ii) from state $\{010\}$ when a call arrives from atom 1; or (iii) from state $\{111\}$ when a service is completed by server 3. The rate associated with this transition is $(\lambda_1 + \lambda_2)P\{001\} + \lambda_1 P\{010\} + \mu_3 P\{111\}$. By arguing that in steady state the transition rates of the system out of and into a given state must be equal, we write the balance equation for state $\{011\}$ as:

$$(\lambda + \mu_1 + \mu_2)P\{011\} = (\lambda_1 + \lambda_2)P\{001\} + \lambda_1 P\{010\} + \mu_3 P\{111\}$$

or

$$(\lambda_1 + \lambda_2)P\{001\} + \lambda_1 P\{010\} - (\lambda + \mu_1 + \mu_2)P\{011\} + \mu_3 P\{111\} = 0\,.$$

The transition rate equilibrium equations around other states of the system can be built in a similar fashion (Chiyoshi *et al.*, 2000, 2003). The full coefficient matrix of the system of equations for the zero capacity waiting line hypercube model (*i.e.*, a loss system with no waiting room) is shown in Table 3. Figure 2 illustrates the possible states for this example on the vertices of a cube, states {000}, {001}, . . . , {111}. As aforementioned, if this example had more than three servers, the corresponding figure would be represented by a hypercube.

**Table 3** – Coefficients matrix of the system of equations for the basic model with zero waiting line.

| State | {000} | {001} | {010} | {100} | {011} | {101} | {110} | {111} |
|---|---|---|---|---|---|---|---|---|
| {000} | $-\lambda$ | $\mu_1$ | $\mu_2$ | $\mu_3$ | | | | |
| {001} | $\lambda_1$ | $-(\lambda+\mu_1)$ | | | $\mu_2$ | $\mu_3$ | | |
| {010} | $\lambda_2$ | | $-(\lambda+\mu_2)$ | | $\mu_1$ | | $\mu_3$ | |
| {100} | $\lambda_3$ | | | $-(\lambda+\mu_3)$ | | $\mu_1$ | $\mu_2$ | |
| {011} | | $\lambda_1+\lambda_2$ | $\lambda_1$ | | $-(\lambda+\mu_1+\mu_2)$ | | | $\mu_3$ |
| {101} | | $\lambda_3$ | | $\lambda_1+\lambda_3$ | | $-(\lambda+\mu_1+\mu_3)$ | | $\mu_2$ |
| {110} | | | $\lambda_2+\lambda_3$ | $\lambda_2$ | | | $-(\lambda+\mu_2+\mu_3)$ | $\mu_1$ |
| {111} | | | | | $\lambda$ | $\lambda$ | $\lambda$ | $-\mu$ |

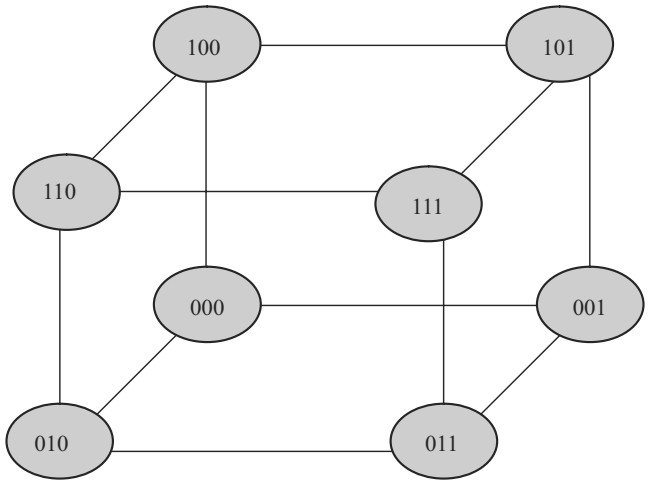


**Figure 2** – Cube representing the possible states of the system.

It is worth noticing that the entries in the coefficient matrix of Table 3, except for the diagonal entries, represent the transition rates from column states to row states. The diagonal entries are the symmetrics of the transition rates of the system out of the corresponding states. Each row of this matrix displays the coefficients of the flow rate balance equation built around the state shown in the leftmost column.

It turns out that the system of equations based on flow balance around hypercube states is singular; an additional equation is thus required to determine the hypercube state probabilities. The

standard procedure is to replace one of the equations by an equation of probabilities normalization condition. For the zero capacity waiting line hypercube, this equation is simply:

$$P\{000\} + P\{001\} + \cdots + P\{111\} = 1\,.$$

The state probabilities for the zero capacity waiting line can be adjusted to handle non-loss systems with an arbitrary number of waiting spots. Suppose room is added to accommodate one additional call. Then the system must be augmented with an additional state, say $\{S_4\}$, with four calls in the system, three being serviced and one waiting in queue. Since the state $\{111\}$ is already in balance with the adjacent states of the hypercube, the transitions between states $\{111\}$ and $\{S_4\}$ must also be in balance, so that:

$$\lambda P\{111\} = \mu P\{S_4\} \quad \text{or} \quad P\{S_4\} = \rho P\{111\}\,,$$

where $\rho = \lambda/\mu$ is the average workload of the system.

When a new state is added to the hypercube states, the set of numbers $P\{000\}, \ldots, P\{111\}$, $P\{S_4\}$ is no longer a proper probability distribution since the sum of these numbers can be greater than 1. This condition can be restored via sum-one normalization. Similar procedure can be used to handle a waiting line with any capacity, augmenting the system with additional states $\{S_4\}, \{S_5\}, \{S_6\}, \ldots$ For an infinity capacity waiting line, the equation of probabilities normalization condition becomes (Chiyoshi *et al.*, 2000):

$$P\{000\} + P\{001\} + \cdots + P\{111\}/(1-\rho) = 1\,.$$

To show the ways in which the output measures of a hypercube models are derived from its state probabilities, we consider two such measures, namely server workload and dispatch frequencies. The workload of a server $i$ is defined as the ratio between its output (production) and its capacity. It can be evaluated as the fraction of time the server is busy by summing the probabilities of the states in which server $i$ is busy. The workload of server 1 for the non-zero capacity waiting line would be:

$$\rho_1 = P\{001\} + P\{011\} + P\{101\} + P\{111\} + P\{Q\}\,,$$

where $Q$ is an additional state in which there is at least one call waiting for service.

Another major output measure is the server dispatch frequencies. The dispatch frequency of unit $i$ to atom $j$ (denoted by $f_{ij}$) is defined as the fraction of a call associated with the dispatch of unit $i$ to atom $j$. These frequencies can also be derived from the state probabilities. For a non-zero capacity waiting line, they have two components, the first associated to unqueued calls (denoted by $f_{ij}^{[u]}$) and the second associated to queued calls (denoted by $f_{ij}^{[q]}$). If we consider the dispatch of unit 1 to atom 2 to service unqueued calls, for instance, its frequency can be evaluated as the joint probability that: (i) server 1 is free and server 2 (preferential server of atom 2) is busy, and (ii) the incoming call is originated from atom 2:

$$f_{12}^{[u]} = (P\{010\} + P\{110\})\ (\lambda_2/\lambda)\,.$$

The dispatch frequency of server 1 to atom 2 to service a queued call is given by the joint probability that: (i) all servers are busy, given by $P\{111\} + P\{Q\}$, (ii) server 1 is the first of all busy servers to become free, given by $\mu_1/\mu$ (see the Appendix), and (iii) atom 2 is the origin of the first waiting call for service. Therefore we have:

$$f_{12}^{[q]} = (P\{111\} + P\{Q\})\ (\lambda_2/\lambda)(\mu_1/\mu)$$

for the dispatch frequency of unit 1 to atom 2 to service a queued call.

The toy-models can be coded into small *ad hoc* codes which can serve (at least) two purposes. The first one is to layout the basic logic larger codes to deal with more complex problems. The other is to "play" with the models in a "what if" sort of approach in order to assess the way in which the systems definition parameters affect its output measures.

We start by looking at the basic characteristics of a double homogeneous system (homogeneous in both demand and service rates) by fixing $\lambda_1 = \lambda_2 = \lambda_3 = 0.5$ and $\mu_1 = \mu_2 = \mu_3 = 1.0$. The state probabilities for this system are shown in Table 4. The row labeled *Loss* of this table shows the state probabilities associated with the zero capacity waiting line. The state probabilities adjusted for the infinite capacity waiting line are shown in the following row (labeled $Q(\infty)$).

**Table 4** – Hypercube states probabilities.

| State | {000} | {001} | {010} | {100} | {011} | {101} | {110} | {111} | {$Q$} |
|---|---|---|---|---|---|---|---|---|---|
| Loss | 0.239 | 0.119 | 0.119 | 0.119 | 0.090 | 0.090 | 0.090 | 0.134 | – |
| $Q(\infty)$ | 0.211 | 0.105 | 0.105 | 0.105 | 0.079 | 0.079 | 0.079 | 0.118 | 0.118 |

The dispatch frequencies for the infinite waiting line are shown in Table 5. The symmetric nature of the model is reflected in these frequencies: (i) all servers have the same total dispatch frequency, (ii) all atoms have the same total frequency of units dispatched to them, and (iii) the dispatch vectors have the same pattern for all units, in which dispatch frequencies decrease according to whether the unit is dispatched as preferential, first or second backup unit.

**Table 5** – Dispatch frequencies for infinite waiting line.

| Server/Atom | 1 | 2 | 3 | Total |
|---|---|---|---|---|
| 1 | 0.1930 | 0.0526 | 0.0877 | 0.3333 |
| 2 | 0.0877 | 0.1930 | 0.0526 | 0.3333 |
| 3 | 0.0526 | 0.0877 | 0.1930 | 0.3333 |
| Total | 0.3333 | 0.3333 | 0.3333 | 1.0000 |

The toy-model can be used to analyze the effects of varying the arrival and service rates in the output measures (for more details, the reader can consult, *e.g.*, Chiyoshi *et al.*, 2001; Morabito *et al.*, 2008). We could, for instance, evaluate the effects of service rates imbalance on workload and dispatch frequencies of the units. To that end, we can compare two service rate profiles: (a) homogeneous ($\mu_1 = \mu_2 = \mu_3 = 1.0$) and (b) non-homogeneous in which the service rate of the

first unit is twice the rate of units 2 and 3, keeping the same total rate ($\mu_1 = 1.50$, $\mu_2 = 0.75$, $\mu_3 = 0.75$). The relevant data for the infinite waiting line are shown in Tables 6 and 7. It can be observed that the faster unit has lower workload and it is dispatched more frequently than the slower units. Although the slower units have the same service rate, unit 2 is seen to have a lower occupation rate than unit 3. This difference is due to the fact that unit 2 is the first backup of the faster and less busy server (unit 1).

**Table 6** – Workloads: (a) homogenous servers, (b) server 1 faster than servers 2 and 3.

| Server | Profile | |
|---|---|---|
| | (a) | (b) |
| 1 | 0.5000 | 0.4380 |
| 2 | 0.5000 | 0.5540 |
| 3 | 0.5000 | 0.5700 |
| Average | 0.5000 | 0.5000 |

**Table 7** – Total dispatch frequencies: (a) homogenous servers, (b) server 1 faster than servers 2 and 3.

| Server | Profile | |
|---|---|---|
| | (a) | (b) |
| 1 | 0.3333 | 0.4380 |
| 2 | 0.3333 | 0.2770 |
| 3 | 0.3333 | 0.2850 |
| Total | 1.0000 | 1.0000 |

### 2.2 The centralized model with random dispatches

In this toy-model, it is assumed that all service units share the same home location and there are no preferences as to the unit being dispatched to service a call. In building the system of equations for equilibrium state probabilities, we must argue differently from the previous model. The transition rate, from say, state {000} to state {001}, must be 1/3 of the total arrival rate since the random dispatch will select server 1, on average, 1/3 of the time out of three free units. In a similar way, the transition rate from state {001} to state {011} must be 1/2 of the total arrival rate because, in this case, we have two free servers to choose randomly from, and, in the long run, server 2 will be chosen half the time out of two free units.

In order to build the balance equation for a given state, say {011}, we argue the following: the system leaves state {011} if a call arrives, or a service is completed by server 1 or 2. The rate associated with this transition is $(\lambda + \mu_1 + \mu_2)P\{011\}$. Conversely, the system enters state {011}: (i) from state {001} half the time a call arrives in the system; (ii) from state {010} half the time a call arrives or (iii) from state {111} when a service is completed by server 3. The rate associated with these transitions are $(\lambda/2)P\{001\} + (\lambda/2)P\{010\} + \mu_3 P\{111\}$. The steady state flow balance reasoning leads to the following equation for state {011}:

$$(\lambda + \mu_1 + \mu_2)P\{011\} = (\lambda/2)P\{001\} + (\lambda/2)P\{010\} + \mu_3 P\{111\}$$

or

$$(\lambda/2)P\{001\} + (\lambda/2)P\{010\} - (\lambda + \mu_1 + \mu_2)P\{011\} + \mu_3 P\{111\} = 0\,.$$

The full coefficient matrix of the system of equations for this model is shown in Table 8. Note that even if $\lambda_1 = \lambda_2 = \lambda_3 = \lambda/3$, this matrix does not coincide with the one in Table 3 of the basic model.

**Table 8** – Coefficient matrix of the system of equations for the centralized model with random dispatches.

| State | {000} | {001} | {010} | {100} | {011} | {101} | {110} | {111} |
|---|---|---|---|---|---|---|---|---|
| {000} | $-\lambda$ | $\mu_1$ | $\mu_2$ | $\mu_3$ | | | | |
| {001} | $\lambda/3$ | $-(\lambda+\mu_1)$ | | | $\mu_2$ | $\mu_3$ | | |
| {010} | $\lambda/3$ | | $-(\lambda+\mu_2)$ | | $\mu_1$ | | $\mu_3$ | |
| {100} | $\lambda/3$ | | | $-(\lambda+\mu_3)$ | | $\mu_1$ | $\mu_2$ | |
| {011} | | $\lambda/2$ | $\lambda/2$ | | $-(\lambda+\mu_1+\mu_2)$ | | | $\mu_3$ |
| {101} | | $\lambda/2$ | | $\lambda/2$ | | $-(\lambda+\mu_1+\mu_3)$ | | $\mu_2$ |
| {110} | | | $\lambda/2$ | $\lambda/2$ | | | $-(\lambda+\mu_2+\mu_3)$ | $\mu_1$ |
| {111} | | | | | $\lambda$ | $\lambda$ | $\lambda$ | $-\mu$ |

In evaluating the dispatch frequencies of unqueued calls ($f_{ij}^{[u]}$), the random nature of the dispatch policy must be taken into account, and appropriate formulas constructed. For example, the dispatch frequency of server 1 to atom 2 is obtained by combining the probability that the incoming call comes from atom 2, which is $\lambda_2/\lambda$ with: (i) 1/3 of the probability that all servers are free; (ii) half the probability that server 1 is one of two free servers; and (iii) the probability that server 1 is the only free server. The resulting expression is thus:

$$f_{12}^{[u]} = (\lambda_2/\lambda)\Big((1/3)P\{000\} + (1/2)\big(P\{010\} + P\{100\}\big) + P\{110\}\Big)$$

The dispatch frequencies for servicing queued calls are not affected by server preferences, depending only on arrival pattern and the probability that a given server is the first one to complete service when all servers are busy. They can be evaluated following the same procedure described for the basic model.

A centralized system can be viewed as a stage in the natural growth of a small emergency medical service built based on a local hospital. The growing demand of such a system both in volume of the calls and covered area can be faced by increasing the number of ambulances located in the hospital. The time may come in which it is asked as to the effect of decentralizing the system to improve its performance in terms of, say, response time. For an example of a related situation in practice, the reader is referred to the case of the *SAMU-Campinas* studied in Takeda *et al.* (2007). This problem can be addressed by looking at the average travelled distances in both centralized and decentralized models. The average travelled distance per call can be evaluated as the weighted sum of the (home location of the) units to atom distances, the weighting factors being the dispatch frequencies.

If all servers are located in atom 1, the unit to atom distances are as shown in Table 9. If we fix $\lambda_1 = \lambda_2 = \lambda_3 = 0.5$ and $\mu_1 = \mu_2 = \mu_3 = 1.0$ in the random dispatch model, the dispatch frequency of every pair server-atom is seen to be 1/9 and the average travelled distance per call turns out to be 1.00. If homogenous servers and full backup are assumed, the aggregate states of the hypercube model can be described by Markovian M/M/3 queueing models. For the decentralized model in which each atom is the home location of a server, the inter atom distances matrix shown in Table 1 turns out to be also the unit to atom distance matrix. By taking the

average of the distances in Table 1 weighed by the dispatch frequencies of Table 5, the average travelled distance per call for the decentralized model turns out to be 0.58.

**Table 9** – Unit to atom distances matrix for the centralized model.

| Server/Atom | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 2 | 1 |
| 2 | 0 | 2 | 1 |
| 3 | 0 | 2 | 1 |

### 2.3 The partial backup model

The next toy-model we look at is a partial backup model. It assumes that each atom is the home location of a server, and that a server can only service calls from its preferential atom and from the second closest atom. Server 1, for example, can take calls from atoms 1 and 2, but not from atom 3. The corresponding dispatch matrix is shown in Table 10.

**Table 10** – Server dispatch preferences for the partial backup model.

| Atom $i$ | Server preferences | |
|---|---|---|
| | 1st | 2nd |
| 1 | 1 | 2 |
| 2 | 2 | 3 |
| 3 | 3 | 1 |

It may be assumed that the decision to not dispatch server 1 to atom 2 is based on the existence of a secondary ESS to handle such a call and a zero capacity waiting line model seems appropriate. Some practical examples appear in SAUs on highways, as the ones of *Anjos do Asfalto* studied in Mendonça & Morabito (2001) and *Centrovias* studied in Iannoni & Morabito (2007). The flow equations of the system can be built in a way similar to that used for the basic model. Care must be taken, however, in determining some of the upward transition rates. Consider state {011}, for instance, where both servers 1 and 2 are busy and server 3 is free. A call from atom 1 will not change the state of the system because it cannot be serviced by server 3, the only server available (Table 10); the call will be lost. The upward transition rate from state {011} to state {111} will therefore be: $(\lambda_2 + \lambda_3)$. The flow equation around state {011} will be:

$$(\lambda_2 + \lambda_3 + \mu_1 + \mu_2)P\{011\} = (\lambda_1 + \lambda_2)P\{001\} + \lambda_1 P\{010\} + \mu_3 P\{111\}\,.$$

The full coefficient matrix for this model is shown in Table 11.

The dispatch frequencies can be evaluated in two steps, keeping in mind that, in this model, all dispatches are for unqueued calls. The dispatch frequency of server 1 to atom 2, for example,

will be evaluated as the joint probability that the incoming call has originated in atom 2, server 2 is busy and server 1 is free:

$$f_{12}^{[u]} = (\lambda_2/\lambda)(P\{010\} + P\{110\})$$

The dispatch frequency of server $i$ to atom $j$ is defined simply as the fraction of all dispatches that send server $i$ to atom $j$; so we would expect $\sum_{i,j} f_{ij}^{[u]} = 1$. Due to lost calls, however, the dispatch frequencies based on the formula above will add up to less than one; in fact, they will add up to the fraction of calls that are serviced by the system, leaving out the lost calls. For this reason, a second step is required in which results from the first step are normalized to sum one.

**Table 11** – Coefficient matrix of the system of equations for the partial backup model.

| State | {000} | {001} | {010} | {100} | {011} | {101} | {110} | {111} |
|---|---|---|---|---|---|---|---|---|
| {000} | $-\lambda$ | $\mu_1$ | $\mu_2$ | $\mu_3$ | | | | |
| {001} | $\lambda_1$ | $-(\lambda+\mu_1)$ | | | $\mu_2$ | $\mu_3$ | | |
| {010} | $\lambda_2$ | | $-(\lambda+\mu_2)$ | | $\mu_1$ | | $\mu_3$ | |
| {100} | $\lambda_3$ | | | $-(\lambda+\mu_3)$ | | $\mu_1$ | $\mu_2$ | |
| {011} | | $\lambda_1+\lambda_2$ | $\lambda_1$ | | $-(\lambda_2+\lambda_3$ $+\mu_1+\mu_2)$ | | | $\mu_3$ |
| {101} | | $\lambda_3$ | | $\lambda_1+\lambda_3$ | | $-(\lambda_1+\lambda_2$ $+\mu_1+\mu_3)$ | | $\mu_2$ |
| {110} | | | $\lambda_2+\lambda_3$ | $\lambda_2$ | | | $-(\lambda_1+\lambda_3$ $\mu_2+\mu_3)$ | $\mu_1$ |
| {111} | | | | | $\lambda_2+\lambda_3$ | $\lambda_1+\lambda_2$ | $\lambda_1+\lambda_3$ | $-\mu$ |

The most distinctive characteristic of this model is the fraction of lost calls, both per atom and system wide. To show how the fraction of lost calls can be evaluated, we work out an instance in which $\lambda_1 = 0.4$, $\lambda_2 = 0.6$, $\lambda_3 = 0.8$ and $\mu_1 = \mu_2 = \mu_3 = 1.0$. By solving the corresponding system of linear equations, we obtain the hypercube state probabilities as shown in Table 12. From these equilibrium probabilities, the dispatch frequencies are evaluated (Table 13).

**Table 12** – Hypercube state probabilities for the partial backup model.

| State | {000} | {001} | {010} | {100} | {011} | {101} | {110} | {111} |
|---|---|---|---|---|---|---|---|---|
| Prob | 0.1979 | 0.1018 | 0.1120 | 0.1425 | 0.0799 | 0.1259 | 0.1148 | 0.1252 |

**Table 13** – Dispatch frequencies.

| Server/Atom | 1 | 2 | 3 | Total |
|---|---|---|---|---|
| 1 | 0.1261 | | 0.1144 | 0.2404 |
| 2 | 0.0506 | 0.1894 | | 0.2400 |
| 3 | | 0.0640 | 0.2185 | 0.2824 |
| Total | 0.1766 | 0.2533 | 0.3329 | 0.7628 |

The first thing to observe in Table 13 is the total dispatch frequency of 0.7628, which corresponds to the fraction of calls that are serviced by the system, resulting in a fraction of lost calls of 0.2372. The dispatch frequencies as displayed in Table 13 refer to frequencies per call demanding service. To obtain the dispatch frequencies per serviced calls, those frequencies must be normalized to add up to one. The fraction of lost calls of a given atom corresponds to the sum of the probabilities associated to the states in which both preferential and the backup units are busy.

For atom 1, for instance, the states to consider are those in which units 1 and 2 are busy, namely {011} and {111}. The loss probabilities per atom are shown in Table 14. The last column of this table shows the number of calls lost per unit of time, obtained by multiplying the demand rate of the atom and the corresponding probability. The ratio of the total lost call rate to the total demand rate gives the system wide fraction of lost calls. It can be observed that, in the partial backup loss model, the probability that all servers are busy does not equal the loss probability (see Table 12).

**Table 14** – Probabilities of loss and rate of lost calls.

| Atom | Loss | |
|---|---|---|
| | Prob | Rate |
| 1 | 0.2051 | 0.0820 |
| 2 | 0.2400 | 0.1440 |
| 3 | 0.2511 | 0.2009 |
| Total | 0.2372 | 0.4269 |

## 3 THE MULTIPLE DISPATCH HYPERCUBE MODEL

### 3.1 The basic model

In order to describe the basic multiple dispatch hypercube model, we consider once again our system with three atoms connected by a one-way ring road (Fig. 1) with identical servers and preference matrix of Table 2. However, now we assume that in each atom $j$, calls can be of two types: type 1 calls (with arrival rate $\lambda_j^{[1]}$) that require the dispatch of only one unit, and type 2 calls (with arrival rate $\lambda_j^{[2]}$) that require the simultaneous response of two units. The total arrival rate of the system is:

$$\lambda = \lambda_1^{[1]} + \lambda_1^{[2]} + \lambda_2^{[1]} + \lambda_2^{[2]} + \lambda_3^{[1]} + \lambda_3^{[2]};$$

the total arrival rate of type 1 calls is: $\lambda^{[1]} = \sum_j \lambda_j^{[1]}$, and the total arrival rate of type 2 calls is: $\lambda^{[2]} = \sum_j \lambda_j^{[2]}$.

Firstly, we consider a zero capacity waiting line system with total backup (*i.e.*, every server can be sent to an atom, as the basic single dispatch toy-model in Section 2). The first preferential unit is dispatched when a type 1 call arrives; if this unit is busy, the backup units can be sent according to the preference order and availability. In the case of type 2 calls, the first two preferential units

are sent simultaneously; if one of them is busy, the third preferential is dispatched. When only one of the three preferential servers is available, it is assigned as a single dispatch. For both type 1 and type 2 calls, if all servers are busy, the call is lost to the system. Examples of this multi-dispatch ESS in practice are the police patrol systems studied in Chelst & Barlach (1981).

Regarding the service process, type 1 calls are serviced by a single unit $i$ with mean service rate $\mu_i$, whereas type 2 calls are serviced by two units $i$ and $k$, operating independently with mean service rates $\mu_i$ and $\mu_k$, respectively. Consequently, the model considers the two units servicing the same type 2 call as the same as two units servicing two separated type 1 calls.

The new detail observed in building the set of flow-balancing equations with respect to the toy-models of Section 2 is that: two servers become busy in the same upward transition when a type 2 call arrives and they correspond to the two highest ranked units available in the atoms server preference list. Moreover, a single server becomes busy in a transition if a type 1 call arrives at the system and this server is the first or next backup available, as well as if a type 2 call arrives at the system and the server is the only unit available in the preference list. Consequently, note that some upward transitions can occur on the diagonals of the cube representing the toy-model in Figure 2, rather than just transitions on the cube edges, such as: $\{000\} \rightarrow \{011\}$, $\{000\} \rightarrow \{101\}$, $\{000\} \rightarrow \{110\}$, $\{001\} \rightarrow \{111\}$, $\{010\} \rightarrow \{111\}$ and $\{100\} \rightarrow \{111\}$.

For instance, consider once again state $\{011\}$, where both servers 1 and 2 are busy and server 3 is free. At this state, unit 3 can be dispatched to any atom to service a call as the preferential or backup server (see server preference list on Table 2), and the downward transition $\{000\} \rightarrow \{011\}$ also needs to be included, since servers 1 and 2 become simultaneously busy with transition rate $(\lambda_1^{[2]})$. Therefore, the flow equation of state $\{011\}$ is:

$$\left(\lambda + \mu_1 + \mu_2\right) P\{011\} = \lambda_1^{[2]} P\{000\} + \left(\lambda_1^{[1]} + \lambda_2^{[1]}\right) P\{001\} + \lambda_1^{[1]} P\{010\} + \mu_3 P\{111\}\,.$$

Note that the system leaves state $\{011\}$ if a call arrives or a service is completed by server 1 or 2 (left hand side of equation). Conversely, in the right hand side of the equation the system enters state $\{011\}$: (i) from state $\{000\}$ when a type 2 call arrives from atom 1; (ii) from state $\{001\}$ when a type 1 call arrives from atoms 1 or 2; (iii) from state $\{010\}$ when a type 1 call arrives from atom 1; or (iv) when a service is completed by server 3. The full coefficient matrix of the system of equations for this toy-model (with zero line capacity and total backup) is shown in Table 15. By comparing to Table 3 of Section 2.1, we can observe that some blank cells in Table 3 are filled in Table 15 (*e.g.*, in the first column and last line) representing the transition rates on diagonals or double dispatch (when two servers become busy simultaneously).

Now consider a partial backup dispatch policy with zero capacity waiting line (*i.e.*, a server can only service calls from its preferential atom and from the second closest atom) for this system, as discussed in Section 2.3, and the dispatch preference matrix in Table 10. The preferential unit is sent to a type 1 call and, if it is busy, the second (backup) is dispatched. In the case of a type 2 call, the first two units on the list are dispatched and, if only one of them is available, it is dispatched as a single dispatch (perhaps with the help of other ESS). If both preferential units are busy, then regardless of the type of the call, it is lost, since in the partial backup policy a third

**Table 15** – Coefficient matrix of the system of equations for the multiple dispatch total backup model.

| State | {000} | {001} | {010} | {100} | {011} | {101} | {110} | {111} |
|---|---|---|---|---|---|---|---|---|
| {000} | $-\lambda$ | $\mu_1$ | $\mu_2$ | $\mu_3$ | | | | |
| {001} | $\lambda_1^{[1]}$ | $-(\lambda+\mu_1)$ | | | $\mu_2$ | $\mu_3$ | | |
| {010} | $\lambda_2^{[1]}$ | | $-(\lambda+\mu_2)$ | | $\mu_1$ | | $\mu_3$ | |
| {100} | $\lambda_3^{[1]}$ | | | $-(\lambda+\mu_3)$ | | $\mu_1$ | $\mu_2$ | |
| {011} | $\lambda_1^{[2]}$ | $\lambda_1^{[1]}+\lambda_2^{[1]}$ | $\lambda_1^{[1]}$ | | $-(\lambda+\mu_1+\mu_2)$ | | | $\mu_3$ |
| {101} | $\lambda_3^{[2]}$ | $\lambda_3^{[1]}$ | | $\lambda_1^{[1]}+\lambda_3^{[1]}$ | | $-(\lambda+\mu_1+\mu_3)$ | | $\mu_2$ |
| {110} | $\lambda_2^{[2]}$ | | $\lambda_2^{[1]}+\lambda_3^{[1]}$ | $\lambda_1^{[2]}$ | | | $-(\lambda+\mu_2+\mu_3)$ | $\mu_1$ |
| {111} | | $\lambda^{[2]}$ | $\lambda^{[2]}$ | $\lambda^{[2]}$ | $\lambda$ | $\lambda$ | $\lambda$ | $-\mu$ |

unit is never assigned. A practical example of such a system is the *SAU-Centrovias* case studied in Iannoni *et al.* (2008, 2009). The flow equations of the system can be built in a way similar to the single partial backup model, and by taking into account multiple dispatch transitions.

For instance, for state {011}, server 3 is idle and can service only calls arriving from atoms 2 and 3, and any call arriving from atom 1 will be lost. In case of type 2 calls arriving from atoms 2 and 3, server 3 is sent as a single dispatch since it is the only server available. The flow equation of state {011} for this zero waiting line partial backup multiple dispatch model is:

$$\left(\lambda_2^{[1]}+\lambda_2^{[2]}+\lambda_3^{[1]}+\lambda_3^{[2]}+\mu_1+\mu_2\right)P\{011\}$$

$$=\lambda_1^{[2]}P\{000\}+\left(\lambda_1^{[1]}+\lambda_1^{[2]}+\lambda_2^{[1]}\right)P\{001\}+\left(\lambda_1^{[1]}+\lambda_1^{[2]}\right)P\{010\}+\mu_3 P\{111\}\,.$$

Observe on the left hand side of this flow equation that the system leaves state {011} if either a call arrives at atoms 2 and 3, or a service is completed by server 1 or 2. Moreover, observe on the right hand side that the system enters state {011}: (i) from state {000} when a type 2 call arrives from atom 1 with rate $\lambda_1^{[2]}$; (ii) from state {001} when a type 1 or type 2 call arrives from atom 1 (server 2 is sent as a single dispatch to service type 2 calls, since one of the two first preferential servers is busy), or a type 1 call arrives from atom 2 (see Table 10); (iii) from state {010} when a type 1 or type 2 call arrives from atom 1; or (iv) when a service is completed by server 3. The full coefficient matrix for this toy-model (with zero line capacity and partial backup) is shown in Table 16 (compare to Table 11 of Section 2.3).

In the case of the multi-dispatch model with total backup, the loss probability is equal to the state probability of all busy servers, $P\{111\}$, whereas in the partial backup multi-dispatch, a call can be lost to the system even when there are servers available, as shown in Section 2.3. Therefore, in a multiple dispatch toy-model, we have: the loss probability of type 1 calls, $P_{loss}^{[1]}$, the loss probability of type 2 calls, $P_{loss}^{[2]}$, and the loss probability of any call, $P_{loss}$. For example, the loss probability $P_{loss}$, considering partial backup, for the system in Figure 1 can be formulated as:

$$P_{loss}=\frac{\lambda_1^{[1]}+\lambda_1^{[2]}}{\lambda}P\{011\}+\frac{\lambda_2^{[1]}+\lambda_2^{[2]}}{\lambda}P\{110\}+\frac{\lambda_3^{[1]}+\lambda_3^{[2]}}{\lambda}P\{101\}+P\{111\}\,.$$

**Table 16** – Coefficient matrix of the system of equations for the multiple dispatch partial backup model.

| State | {000} | {001} | {010} | {100} | {011} | {101} | {110} | {111} |
|---|---|---|---|---|---|---|---|---|
| {000} | $-\lambda$ | $\mu_1$ | $\mu_2$ | $\mu_3$ | | | | |
| {001} | $\lambda_1^{[1]}$ | $-(\lambda+\mu_1)$ | | | $\mu_2$ | $\mu_3$ | | |
| {010} | $\lambda_2^{[1]}$ | | $-(\lambda+\mu_2)$ | | $\mu_1$ | | $\mu_3$ | |
| {100} | $\lambda_3^{[1]}$ | | | $-(\lambda+\mu_3)$ | | $\mu_1$ | $\mu_2$ | |
| {011} | $\lambda_1^{[2]}$ | $\lambda_1^{[1]}+\lambda_2^{[1]}$ $+\lambda_2^{[2]}$ | $\lambda_1^{[1]}+\lambda_2^{[1]}$ | | $-(\lambda_2^{[1]}+\lambda_3^{[1]}+\lambda_2^{[2]}$ $+\lambda_3^{[2]}+\mu_1+\mu_2)$ | | | $\mu_3$ |
| {101} | $\lambda_3^{[2]}$ | $\lambda_3^{[1]}+\lambda_3^{[2]}$ | | $\lambda_3^{[1]}+\lambda_3^{[2]}$ $+\lambda_3^{[1]}$ | | $-(\lambda_1^{[1]}+\lambda_2^{[1]}+\lambda_1^{[2]}$ $+\lambda_2^{[2]}+\mu_1+\mu_3)$ | | $\mu_2$ |
| {110} | $\lambda_2^{[2]}$ | | $\lambda_2^{[1]}+\lambda_3^{[1]}$ $+\lambda_2^{[2]}$ | $\lambda_2^{[1]}+\lambda_2^{[2]}$ | | | $-(\lambda_1^{[1]}+\lambda_3^{[1]}+\lambda_1^{[2]}$ $+(\lambda_3^{[2]}+\mu_2+\mu_3$ | $\mu_1$ |
| {111} | | $\lambda^{[2]}$ | $\lambda^{[2]}$ | $\lambda^{[2]}$ | $\lambda_2^{[1]}+\lambda_3^{[1]}$ $+\lambda_2^{[2]}+\lambda_3^{[2]}$ | $\lambda_1^{[1]}+\lambda_2^{[1]}$ $+\lambda_1^{[2]}+\lambda_2^{[2]}$ | $\lambda_1^{[1]}+\lambda_3^{[1]}$ $+\lambda_1^{[2]}+\lambda_3^{[2]}$ | $-\mu$ |

Note in this equation that each term is obtained as the joint probability that the arriving call originates at atom $j$ (type 1 or type 2) and the two preferential servers are busy (see Table 10). For example, for atom 1, a call arriving (with probability $\left(\lambda_1^{[1]} + \lambda_1^{[2]}\right)/\lambda$) when servers 1 and 2 are busy (with probabilities $P\{011\}$ and $P\{111\}$) is lost to the system. As mentioned before, in case of type 2 calls, they can be serviced by a single server (possibly with the help of another system) if just one server from its dispatch list (with two servers) is available, and they are lost if both servers are busy.

Regarding the dispatch frequencies, we can calculate other statistics in addition to the fraction of dispatches that send server $i$ to atom $j$ ($f_{ij}$), such as: the fraction of dispatches that send server $i$ to atom $j$ to service a type 1 call ($f_{ij}^{[1]}$) (considering only type 1 calls), the fraction of dispatches that send server $i$ to atom $j$ to service a type 2 call ($f_{ij}^{[2]}$) (considering only type 2 calls), and the fraction of dispatches that send servers $i$ and $k$ to atom $j$ to service a type 2 call ($f_{(i,k)j}^{[2]}$). For example, considering the zero line total backup multiple dispatch model, the dispatch frequency of server 3 to atom 2 to service a type 2 call (taking into account only type 2 calls) as a single dispatch ($f_{32}^{[2]}$) is evaluated as the joint probability that the incoming type 2 call has arrived from atom 2 and server 3 is the only available one (as a result, it is sent as single dispatch):

$$f_{32}^{[2]} = \frac{(\lambda_2^{[2]}/\lambda^{[2]})P\{011\}}{(1 - P_{loss}^{[2]})}, \text{ where } \lambda^{[2]} = \lambda_1^{[2]} + \lambda_2^{[2]} + \lambda_3^{[2]} \text{ and } P_{loss}^{[2]} = P_{loss} = P\{111\}.$$

Alternatively, the dispatch frequency of sending servers 2 and 3 to atom 2 to answer a type 2 call is evaluated as the joint probability that the incoming type 2 call arrives from atom 2 and server 2 and 3 are idle:

$$f_{(2,3)2}^{[2]} = \frac{(\lambda_2^{[2]}/\lambda^{[2]})(P\{000\} + P\{001\})}{(1 - P_{loss}^{[2]})}.$$

Similarly to the single dispatch frequencies of Section 2, the sum of these measures to all type 2 dispatches yields:

$$\sum_{j=1}^{3}\left[\sum_{i=1}^{3}\left(f_{ij}^{[2]}\right) + \sum_{i=1}^{2}\sum_{k=i+1}^{3} f_{(i,k)j}^{[2]}\right] = 1.$$

Using the fractions of dispatches and the inter-atoms travel distance matrix in Table 1, we obtain interesting travel distance measures, such as the mean travel distance to service type 1 calls, $\bar{T}^{[1]}$ and the mean travel distance to service type 2 calls, $\bar{T}^{[2]}$, defined as:

$$\bar{T}^{[1]} = \sum_{j=1}^{3}\sum_{i=1}^{3} f_{ij}^{[1]}\bar{t}_{ij}$$

and

$$\bar{T}^{[2]} = \sum_{j=1}^{3}\left[\sum_{i=1}^{2}\sum_{k=1+1}^{3} f_{(i,k)j}^{[2]}\min(\bar{t}_{ij}, \bar{t}_{kj}) + \sum_{i=1}^{3} f_{ij}^{[2]}\bar{t}_{ij}\right],$$

where $\bar{t}_{ij}$ is the mean travel distance from the base of server $i$ to atom $j$ given in Table 1. Moreover, the mean travel time to the first and to the second ambulance arriving at a given type 2 call location can be determined, respectively, as follows:

$$\bar{T}_f = \frac{\sum_{j=1}^{3}\sum_{i=1}^{2}\sum_{k=i+1}^{3} f_{(i,k)j}^{[2]} \min(\bar{t}_{ij}, \bar{t}_{kj})}{\sum_{j=1}^{3}\sum_{i=1}^{2}\sum_{k=i+1}^{3} f_{(i,k)j}^{[2]}}$$

$$\bar{T}_s = \frac{\sum_{j=1}^{3}\sum_{i=1}^{2}\sum_{k=i+1}^{3} f_{(i,k)j}^{[2]} \max(\bar{t}_{ij}, \bar{t}_{kj})}{\sum_{j=1}^{3}\sum_{i=1}^{2}\sum_{k=i+1}^{3} f_{(i,k)j}^{[2]}}$$

In order to show how some of these measures can be evaluated in this zero line toy-model (considering total and partial backup), we solve the example of Section 2 with $\lambda_1^{[1]} = 0.30$, $\lambda_2^{[1]} = 0.45$, $\lambda_3^{[1]} = 0.60$, $\lambda_1^{[2]} = 0.10$, $\lambda_2^{[2]} = 0.15$, $\lambda_3^{[2]} = 0.20$ and $\lambda = 1.80$ (note now that type 2 calls correspond to 25% of the total calls in each atom), and $\mu_1 = \mu_2 = \mu_3 = 1.0$. By solving the corresponding system of linear equations of the zero line capacity and multiple dispatch model with total and partial backup, we obtain the hypercube state probabilities of Table 17. Table 18 presents the workload statistics obtained by these models.

**Table 17** – Hypercube state probabilities for the multiple dispatch models with finite capacity queue.

| | State | {000} | {001} | {010} | {100} | {011} | {101} | {110} | {111} |
|---|---|---|---|---|---|---|---|---|---|
| Total backup | Prob | 0.1660 | 0.0890 | 0.0969 | 0.1129 | 0.0898 | 0.1097 | 0.1069 | 0.2287 |
| Partial backup | | 0.1834 | 0.0986 | 0.1020 | 0.1296 | 0.0856 | 0.1354 | 0.1174 | 0.1481 |

**Table 18** – Server workloads for the multiple dispatch models.

| Server $i$ | Workloads | |
|---|---|---|
| | Total backup | Partial backup |
| 1 | 0.5172 | 0.4676 |
| 2 | 0.5222 | 0.4531 |
| 3 | 0.5582 | 0.5305 |

The system loss probability ($P_{loss}$) for the total backup model is 22.86%, whereas for the partial backup model, it is 26.64%. For the fraction of dispatch statistics as defined above, Table 19 presents the type 1 dispatch frequencies ($f_{ij}^{[1]}$) obtained by these toy-models. Calculating the travel distance measures described above using the distance matrix of Table 1, we obtain: $\bar{T}^{[1]} = 0.535$, $\bar{T}^{[2]} = 0.535$, $\bar{T}_f = 0.220$ and $\bar{T}_s = 1.432$ for the total backup model, and $\bar{T}^{[1]} = 0.306$, $\bar{T}^{[2]} = 0.306$, $\bar{T}_f = 0.0$ and $\bar{T}_s = 1.0$ for the partial backup model.

### 3.2 The model with a third state of each server

In this toy-model, we observe some important modifications to the basic multiple dispatch model with a zero capacity line in Section 3.1. First, calls arriving from atom $j$ can be of three types:

**Table 19** – Fraction of all type 1 dispatches that sends unit $i$ to atom $j$.

| Total backup | | | | Partial backup | | | |
|---|---|---|---|---|---|---|---|
| Server/Atom | 1 | 2 | 3 | Server/Atom | 1 | 2 | 3 |
| 1 | 0.1391 | 0.0462 | 0.1267 | 1 | 0.1613 | 0.0 | 0.1496 |
| 2 | 0.0573 | 0.2065 | 0.0632 | 2 | 0.0709 | 0.2485 | 0.0 |
| 3 | 0.0259 | 0.0807 | 0.2546 | 3 | 0.0 | 0.0852 | 0.2845 |
| Total | 0.2222 | 0.3334 | 0.4445 | Total | 0.2322 | 0.3337 | 0.4341 |

type 1 calls (single dispatch), type 2 calls (double dispatch) and type 0 "calls" (type 0 refers to locally occurring demand for service not received by the operations center, as the PIA aforementioned, so that it does not involve actual calls or dispatches) with arrival rate $\lambda_j^{[0]}$. Then the total arrival rate at an atom $j$ becomes: $\lambda_j = \lambda_j^{[0]} + \lambda_j^{[1]} + \lambda_j^{[2]}$. Therefore, at a given time instant, the status of each server can be: (0) idle, (1) busy servicing a type 1 or a type 2 call, (2) busy servicing a type 0 "call". Consequently, there are $3^3 = 27$ possible states for the system.

Furthermore, the mean service rate of non-assigned calls (type 0 calls) distinguishes from the mean service rate of assigned calls (type 1 and type 2 calls), and usually there is no backup for these calls, since if the preferential server is not available when a type 0 event occurs, this call is lost to the system. Accordingly, type 0 "calls" have null travel time and no backup server, and each server $i$ has mean service rates $\mu_i^{[I]}$ for types 1 and 2 calls, and $\mu_i^{[II]}$ for type 0 calls. Examples of related situations in practice are the police deployment system studied in Larson & Mcknew (1982) and the highway emergency medical system studied in Iannoni & Morabito (2007).

For instance, in building the flow balance equation to state {011} and considering total backup and the preference matrix in Table 2, we take into account that server 3 can also service type 0 "calls" at atom 3 (with rate $\lambda_3^{[0]}$), and that type 0 "calls" occurring in atoms 1 and 2 are lost since servers 1 and 2 are busy. Moreover, the system can enter state {011} from state {211} when a type 0 "call" service is completed by server 3 with rate $\mu_3^{[II]}$. As a result, the flow equation for the state {011} for this zero line capacity total backup model becomes:

$$\left(\lambda - \left(\lambda_1^{[0]} + \lambda_2^{[0]}\right) + \mu_1^{[I]} + \mu_2^{[I]}\right) P\{011\}$$

$$= \left(\lambda_1^{[1]} + \lambda_2^{[1]}\right) P\{001\} + \lambda_1^{[1]} P\{010\} + \mu_3^{[I]} P\{111\} + \mu_3^{[II]} P\{211\}$$

Analyzing other states in this toy-model, for instance, state {121} where both units 1 and 3 are busy servicing a type 1 or a type 2 call, and unit 2 is busy servicing a type 0 call, we obtain the following flow equation:

$$\left(\mu_1^{[I]} + \mu_2^{[II]} + \mu_3^{[I]}\right) P\{121\} = \lambda_3^{[2]} P\{020\}$$

$$+ \left(\lambda - \left(\lambda_1^{[0]} + \lambda_2^{[0]} + \lambda_3^{[0]}\right)\right) P\{021\} + \lambda_2^{[0]} P\{101\} + \left(\lambda - \left(\lambda_1^{[0]} + \lambda_2^{[0]} + \lambda_3^{[0]}\right)\right) P\{120\}.$$

Observe in this equation that the system leaves state {121} if a service type 1 or type 2 is completed by server 1 or server 3, or if a service type 0 is completed by server 2, whereas the system enters state {121}: (i) from state {020} when a call type 2 arrives from atom 3 with arrival rate $\lambda_3^{[2]}$; (ii) from state {021} when a call arrives in the system, except type 0 "calls" from any atom; (iii) from state {101} when a type 0 "call" arrives at atom 2, and (iv) from state {120} when a call arrives in the system, except to type 0 "calls" from any atom. Due to its size and complexity (even for a toy-model), the full coefficient matrix for this model is omitted.

Observe that the system loss probability is given by the joint probability that: (i) a type 0 call is originated at atom $j$, and the system is at a state in which the preferential server of atom $j$ is busy, and (ii) a call of any type arrive at the system and all servers are busy, given by:

$$\begin{aligned} P_{loss} &= \lambda_1^{[0]}/\lambda \sum_{i,j,k\neq 0} P\{ijk\} + \lambda_2^{[0]}/\lambda \sum_{i,j\neq 0,k} P\{ijk\} \\ &\quad +\lambda_3^{[0]}/\lambda \sum_{i\neq 0,j,k} P\{ijk\} + \sum_{i\neq 0,j\neq 0,k\neq 0} P\{ijk\}\,. \end{aligned}$$

We slightly modified the 3-server toy-model of Section 3.1, by including the type 0 "call" arrival rates and mean service times of these events. We examine an instance in which: $\lambda_1^{[1]} = 0.30$, $\lambda_2^{[1]} = 0.45$, $\lambda_3^{[1]} = 0.60$, $\lambda_1^{[2]} = 0.10$, $\lambda_2^{[2]} = 0.15$, $\lambda_3^{[2]} = 0.20$, $\lambda_1^{[0]} = 0.04$, $\lambda_2^{[0]} = 0.06$, $\lambda_3^{[0]} = 0.08$, and the respective service rates for each unit $i$ are: $\mu_1^{[I]} = \mu_2^{[I]} = \mu_3^{[I]} = 1.0$ and $\mu_1^{[II]} = \mu_2^{[II]} = \mu_3^{[II]} = 0.75$. By solving the total backup zero line capacity multi dispatch toy-model with the third status discussed in this section, we computed the 27 state probabilities values in order to evaluate the mean performance measures. Table 20 presents the server workloads obtained considering the two busy status for each server $i$: $\rho_i^{[I]}$ in type 1 and 2 calls and $\rho_i^{[II]}$ in type 0 calls. For instance, the expressions to evaluate these workloads to server 3 are:

$$\begin{aligned} \rho_3^{[I]} &= P\{100\} + P\{101\} + P\{102\} + P\{110\} + P\{120\} \\ &\quad +P\{111\} + P\{112\} + P\{121\} + P\{122\} \\ \rho_3^{[II]} &= P\{200\} + P\{201\} + P\{202\} + P\{210\} + P\{220\} \\ &\quad +P\{211\} + P\{212\} + P\{221\} + P\{222\}\,. \end{aligned}$$

Table 21 presents the type 1 dispatch frequencies by solving the total backup zero line capacity model (calculated in a similar way as the toy-models from previous sections). The system loss probability for all calls is $P_{loss} = 27.30\%$. In particular, for type 0 "calls" this probability is $P_{loss}^{[0]} = 55.93\%$.

Other performance measures can be computed for this system as, for example, type 2 dispatch frequencies ($f_{(i,k)j}^{[2]}$), *i.e.*, the fraction of dispatches sending two servers $i$ and $k$ simultaneously (where unit $i$ is the first ranked available server preference of atom $j$ between $i$ and $k$): $f_{(1,2)1}^{[2]} = 0.0753$, $f_{(1,3)1}^{[2]} = 0.0280$, $f_{(2,3)1}^{[2]} = 0.0253$, $f_{(2,3)2}^{[2]} = 0.1017$, $f_{(2,1)2}^{[2]} = 0.0492$, $f_{(3,1)2}^{[2]} = 0.0420$, $f_{(3,1)3}^{[2]} = 0.1410$, $f_{(3,2)3}^{[2]} = 0.0505$, $f_{(1,2)3}^{[2]} = 0.0656$. In terms of the aggregated travel time measures, we found: $\bar{T}^{[1]} = 0.560$; $\bar{T}^{[2]} = 0.560$; and $\bar{T}_f = 0.230$ and $\bar{T}_s = 1.450$.

**Table 20** – Server workloads for the zero line capacity multi dispatch total backup model.

| Server $i$ | Workloads | |
|---|---|---|
| | $\rho_i^{[\mathrm{I}]}$ | $\rho_i^{[\mathrm{II}]}$ |
| 1 | 0.5114 | 0.0247 |
| 2 | 0.5084 | 0.0364 |
| 3 | 0.5371 | 0.0446 |

**Table 21** – Fraction of all type 1 dispatches that sends unit $i$ to atom $j$.

| Server/Atom | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0.1364 | 0.0497 | 0.1318 |
| 2 | 0.0585 | 0.2008 | 0.0666 |
| 3 | 0.0272 | 0.0828 | 0.2461 |
| Total | 0.2221 | 0.3334 | 0.4445 |

### 3.3 The model with differentiated servers

In this last toy-model, we investigate a zero line capacity multiple dispatch partial backup model, where the servers can be of different types. As mentioned before, there are several ESS operating with differentiated servers. For example, in some urban emergency medical systems, the ambulances can be either advanced support vehicles or basic support vehicles (case study of *SAMU-Campinas* in Takeda *et al.*, 2007). Furthermore, in some systems on urban areas and on highways, the dispatch policies involve ambulances and medical vehicles – the medical vehicle is smaller and faster, but it cannot transport patients (case study of *SAU-Centrovias* in Iannoni & Morabito (2007). There are other ESS in which single or multi dispatches of differentiated servers occur in accordance with the type of event in common, such as vehicles and equipment for fire control services (Swersey, 1994). Consider that in our 3-servers system in Figure 1, server 1 is distinct from servers 2 and 3, since it involves other type of personnel and equipment. Assume that each atom is the home location of a server, and that we have partial backup dispatches according to the type of call and type of servers required. Thus, server 2 and 3 cannot be the backup of server 1. Consequently, in this modified version of the multiple dispatch model, calls in each atom of the system can be of 5 types, instead of only 2 types as in the basic model in Section 3.1:

- Type 1a calls (with arrival rate $\lambda_j^{[1a]}$): calls that require server 1 only. If server 1 is busy, the call is lost since server 2 or 3 is not dispatched as backup.
- Type 1b calls (with arrival rate $\lambda_j^{[1b]}$): calls that require the single dispatch of servers 2 or 3, and there are two candidate servers in its dispatch preference list as preferential and backup (servers 2 and 3, as shown in Table 22).

– Type 2a calls (with arrival rate $\lambda_j^{[2a]}$): calls that require the double dispatch of two distinct servers, server 1 and server 2 or server 3 (the first of the two available in the atom preference list). In the dispatch preference list, for these calls there are three candidate servers (server 1 as first preference and servers 2 and 3 as second or third preferences). In this case, if server 1 is busy and the second preference is idle (servers 2 or 3), the former is sent as single dispatch even if the third preference is also idle, since servers 2 or 3 cannot replace server 1 (they are never assigned together to a type 2a call). However, the third preference can be sent with server 1 in case the second preference is busy.

– Type 2b calls (with arrival rate $\lambda_j^{[2b]}$): calls requiring the double dispatch of identical servers (servers 2 or 3). In this case, server 1 is never dispatched. Moreover, similar to type 2 calls of the toy-model of Section 3.1, there are two candidate servers. If only one is idle, it is sent as a single dispatch, whenever the backup unit is also busy, the call is lost.

– Type 3a calls (with arrival rate $\lambda_j^{[3a]}$): calls requiring the triple dispatch, involving servers 1, 2 and 3. However, when two or only one of them is available, a double or even a single dispatch may be assigned to such calls. If the three servers are busy, the call is lost.

The total arrival rate from atom $j$ is defined as: $\lambda_j = \lambda_j^{[1]} + \lambda_j^{[2]} + \lambda_j^{[3]}$, where $\lambda_j^{[1]} = \lambda_j^{[1a]} + \lambda_j^{[1b]}$, $\lambda_j^{[2]} = \lambda_j^{[2a]} + \lambda_j^{[2b]}$, and $\lambda_j^{[3]} = \lambda_j^{[3a]}$ are, respectively, the arrival rates of calls requiring single (type 1 calls), double (type 2 calls) and triple (type 3 calls) dispatches to atom $j$. Similarly to the basic model in Section 3.1, type 1 calls are serviced by a single server $i$ with a mean service rate $\mu_i$. Type 2 calls are serviced by two servers $i$ and $k$, which operate independently with mean service rates $\mu_i$ and $\mu_k$, respectively, whereas type 3 calls are serviced by three servers $i$, $k$, and $l$, which operate independently, with mean service rates $\mu_i$, $\mu_k$ and $\mu_l$, respectively.

In order to model the dispatch policy, we use separate dispatch preference lists for each type of call, splitting each atom into sub-atoms A and B. This procedure is referred to as *layering* (Larson & Odoni, 1981; Takeda *et al.*, 2007, Iannoni & Morabito, 2007). Therefore, each atom $j$ is divided into two layers: layer A (sub-atom $jA$) for type 1a, 2a and 3a calls serviced by distinct servers (1, 2, 3); and layer B (sub-atom $jB$) for type 1b, 2b calls serviced by identical servers (2 and 3). The original three atoms system is analyzed as a six sub-atoms system. The corresponding dispatch matrix is shown in Table 22.

**Table 22** – Server preference list (multi dispatch with differentiated servers).

| Atoms | Sub-atoms | Types of call | 1st server | 2nd server | 3rd server |
|---|---|---|---|---|---|
| 1 | 1A | 1a, 2a, 3a | 1 | 2 | 3 |
| | 1B | 1b, 2b | 2 | 3 | |
| 2 | 2A | 1a, 2a, 3a | 1 | 2 | 3 |
| | 2B | 1b, 2b | 2 | 3 | |
| 3 | 3A | 1a, 2a, 3a | 1 | 3 | 2 |
| | 3B | 1b, 2b | 3 | 2 | |

Consider once again the state {011} and the dispatch preferences in Table 22. The system leaves this state when a call arrives, except calls type 1a from atoms 1 (1A), 2 (2A) and 3 (3A) that may be lost to the system, given that server 1 is busy and it is the only server that can service this type of call. Conversely, the system enters state {011}: (i) from state {000} when a type 2a call arrives from atoms 1 or 2 requiring the double dispatch of distinct servers 1 and 2; (ii) from state {001} when a type 1b call arrives from atoms 1 or 2, requiring a single dispatch and server 2 is sent; or when a type 2a call arrives from atoms 1 or 2, requiring a double dispatch of two distinct servers (1 and 2), and, as server 1 is busy (and it cannot be replaced by server 3), server 2 goes to the call location as single; (iii) from {010} when a type 1a call arrives from atoms 1, 2 or 3 (single dispatch of server 1); and (iv) from state {111} when server 3 completes service. Hence, the flow equation around state {011} is:

$$\begin{aligned}&\left(\lambda - \left(\lambda_1^{1a} + \lambda_2^{1a} + \lambda_3^{1a}\right) + \mu_1 + \mu_2\right) P\{011\} \\ &= \left(\lambda_1^{[2a]} + \lambda_2^{[2a]}\right) P\{000\} + \left(\lambda_1^{[1b]} + \lambda_1^{[2a]} + \lambda_2^{[1b]} + \lambda_2^{[2a]}\right) P\{001\} \\ &+ \left(\lambda_1^{1a} + \lambda_2^{1a} + \lambda_3^{1a}\right) P\{010\} + \mu_3 P\{111\}\,.\end{aligned}$$

The full coefficient matrix for this model is shown in Table 23.

As one would expect, the multi dispatch model with distinct servers and different types of calls provides additional performance statistics than the multi dispatch models presented in Sections 3.1 and 3.2. Most of these performance measures are described in details in Iannoni & Morabito (2007). For example, we may have: the loss probability of any call ($P_{loss}$) and the loss probability of each type of call (*e.g.*, the loss probability of type $m$ call ($P_{loss}^{[m]}$), $m = 0$, 1 (1a and 1b), 2 (2a and 2b), 3 (3a)). We can also obtain several frequency statistics, according to the type of call in the system, such as: single dispatch frequencies to type 1 calls (types 1a and 1b); double dispatch frequencies to type 2 calls (types 2a and 2b); double dispatch frequencies to type 3 calls (when one of the three candidate servers is busy); single dispatch frequencies to type 2b calls (when one of the two candidates servers is busy) and to type 2a and 3 calls (when two of the three candidates servers are busy), triple dispatch frequencies to type 3 calls; and the fraction of all types of dispatches from server $i$ to atom $j$ ($f_{ij}$).

In order to show some results for this model, let us consider once again the toy-model with 3-servers of Figure 1, where now the arrival rates to each type of call (preserving the total arrival rate $\lambda = 1.80$) is given by Table 24, and the mean service rates are: $\mu_1 = \mu_2 = \mu_3 = 1.0$.

The state equilibrium probabilities of this system are shown in Table 25. By using these probabilities we can calculate all the other measures, for example the workload for each server $i$, is: $\rho_1 = 0.2994$, $\rho_2 = 0.5494$, $\rho_3 = 0.5356$. The system loss probability for any call ($P_{loss}$) is 31.75%; for type 1 calls ($P_{loss}^{[1]}$) the loss probability is 33.53%; for type 2 calls ($P_{loss}^{[2]}$) is 27.20%; and for type 3 calls ($P_{loss}^{[3]}$) is 11.55%.

Calculating dispatch frequencies statistics in a similar way to Section 3.1, we find the type 1 dispatch frequencies ($f_{ij}^{[1]}$) presented in Table 26 (where $\sum_{i=1}^{3}\sum_{j=1}^{3} f_{ij}^{[1]} = 1$). Other performance measures can be computed for this system as, for example, type 2 dispatch frequencies

**Table 23** – Coefficient matrix of the system of equations for the multiple dispatch partial backup and zero line capacity model (with distinct servers).

| State | {000} | {001} | {010} | {100} | {011} | {101} | {110} | {111} |
|---|---|---|---|---|---|---|---|---|
| {000} | $-\lambda$ | $\mu_1$ | $\mu_2$ | $\mu_3$ | | | | |
| {001} | $\lambda_1^{[1a]}+\lambda_2^{[1a]}$ $+\lambda_3^{[1a]}$ | $-\big(\lambda-\big(\lambda_1^{[1a]}$ $+\lambda_2^{[1a]}+\lambda_3^{[1a]}\big)$ $+\mu_1\big)$ | | | $\mu_2$ | $\mu_3$ | | |
| {010} | $\lambda_1^{[1b]}+\lambda_2^{[1b]}$ | | $-(\mu_2+\lambda)$ | | $\mu_1$ | | $\mu_3$ | |
| {100} | $\lambda_3^{[1b]}$ | | | $-(\mu_3+\lambda)$ | | $\mu_1$ | $\mu_2$ | |
| {011} | $\lambda_1^{[2a]}+\lambda_2^{[2a]}$ | $\lambda_1^{[1b]}+\lambda_2^{[1b]}$ $+\lambda_1^{[2a]}+\lambda_2^{[2a]}$ | $\lambda_1^{[1a]}+\lambda_2^{[1a]}$ $+\lambda_3^{[1a]}$ | | $-\big(\lambda-\big(\lambda_1^{[1a]}$ $+\lambda_2^{[1a]}+\lambda_3^{[1a]}\big)$ $+\mu_1+\mu_2\big)$ | | | $\mu_3$ |
| {101} | $\lambda_3^{[2a]}$ | $\lambda_3^{[1b]}+\lambda_3^{[2a]}$ | | $\lambda_1^{[1a]}+\lambda_2^{[1a]}$ $+\lambda_3^{[1a]}$ | | $-\big(\lambda-\big(\lambda_1^{[1a]}$ $+\lambda_2^{[1a]}+\lambda_3^{[1a]}\big)$ $+\mu_1+\mu_3\big)$ | | $\mu_2$ |
| {110} | $\lambda_1^{[2b]}+\lambda_2^{[2b]}$ $+\lambda_3^{[2b]}$ | | $\lambda_1^{[2b]}+\lambda_2^{[2b]}$ $+\lambda_3^{[2b]}+\lambda_1^{[1b]}$ $+\lambda_2^{[1b]}+\lambda_3^{[1b]}$ | $\lambda_1^{[1b]}+\lambda_2^{[1b]}$ $+\lambda_3^{[1b]}+\lambda_1^{[2b]}$ $+\lambda_2^{[2b]}+\lambda_3^{[2b]}$ | | | $-\big(\lambda-\big(\lambda_1^{[1b]}$ $+\lambda_2^{[1b]}+\lambda_3^{[1b]}$ $+\lambda_1^{[2b]}+\lambda_2^{[2b]}$ $+\lambda_3^{[2b]}\big)+\mu_2+\mu_3\big)$ | $\mu_1$ |
| {111} | $\lambda_1^{[3a]}+\lambda_2^{[3a]}$ $+\lambda_3^{[3a]}$ | $\lambda_1^{[2b]}+\lambda_2^{[2b]}$ $+\lambda_3^{[2b]}+\lambda_1^{[3a]}$ $+\lambda_2^{[3a]}+\lambda_3^{[3a]}$ | $\lambda_1^{[2a]}+\lambda_2^{[2a]}$ $+\lambda_3^{[2a]}+\lambda_1^{[3a]}$ $+\lambda_2^{[3a]}+\lambda_3^{[3a]}$ | $\lambda_1^{[2a]}+\lambda_2^{[2a]}$ $+\lambda_3^{[2a]}+\lambda_1^{[3a]}$ $+\lambda_2^{[3a]}+\lambda_3^{[3a]}$ | $-\big(\lambda-\big(\lambda_1^{[1a]}$ $+\lambda_2^{[1a]}+\lambda_3^{[1a]}\big)$ $+\mu_1+\mu_2\big)$ | $-\big(\lambda-\big(\lambda_1^{[1a]}$ $+\lambda_2^{[1a]}+\lambda_3^{[1a]}\big)$ $+\mu_1+\mu_2\big)$ | $\lambda^{[1a]}+\lambda^{[2a]}$ $+\lambda^{[3a]}$ | $-\mu$ |

**Table 24** – Arrival rates each type of call for atom $j$.

| Atoms | Arrival rate $\lambda_j$ | Type | Type 1 $\lambda_j^{1a}, \lambda_j^{1b}$ | Type 2 $\lambda_j^{2a}, \lambda_j^{2b}$ | Type 3 $\lambda_j^{3a}$ |
|---|---|---|---|---|---|
| 1 | 0.4 | a | 0.06 | 0.03 | 0.005 |
| | | b | 0.24 | 0.065 | |
| 2 | 0.6 | a | 0.09 | 0.045 | 0.0075 |
| | | b | 0.36 | 0.0975 | |
| 3 | 0.8 | a | 0.12 | 0.06 | 0.01 |
| | | b | 0.48 | 0.13 | |

**Table 25** – Hypercube state probabilities for the multi dispatch zero line capacity model with distinct servers and partial backup.

| State | {000} | {001} | {010} | {100} | {011} | {101} | {110} | {111} |
|---|---|---|---|---|---|---|---|---|
| Prob | 0.1924 | 0.0668 | 0.1446 | 0.1348 | 0.0606 | 0.0565 | 0.2287 | 0.1155 |

(just considering type 2 calls – types 2a and 2b), $f^{[2]}_{(i,k)j}$. For instance, the fraction of dispatches sending server $i = 1$ (distinct server) to atoms 1, 2 and 3 to service a type 2a call (as double and single dispatch) are: $f^{[2]}_{(1,2)1} = 0.0315$, $f^{[2]}_{(1,3)1} = 0.0139$, $f^{[2]}_{(1,2)2} = 0.0473$, $f^{[2]}_{(1,3)2} = 0.0209$, $f^{[2]}_{(1,2)3} = 0.0260$, $f^{[2]}_{(1,3)3} = 0.0649$, $f^{[2]}_{11} = 0.0220$, $f^{[2]}_{12} = 0.0331$ and $f^{[2]}_{13} = 0.0441$ (where $\sum_{j=1}^{3}[\sum_{i=1}^{3} f^{[2]}_{ij} + \sum_{i=1}^{2}\sum_{k=i+1}^{3} f^{[2]}_{(i,k)j}] = 1$) ). Regarding travel distance measures, we obtain: $\bar{T}^{[1]} = 0.7515$, $\bar{T}^{[2]} = 0.7957$, $\bar{T}_f = 0.6280$ and $\bar{T}_s = 1.2009$, and the region wide travel distance aggregating all call types is: $\bar{T} = 0.7671$.

**Table 26** – Fraction of all type 1 dispatches that sends unit $i$ to atom $j$ for the multi dispatch model with distinct servers and partial backup.

| Server/Atom | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0.0468 | 0.0703 | 0.0937 |
| 2 | 0.1205 | 0.1808 | 0.1024 |
| 3 | 0.0549 | 0.0823 | 0.2484 |
| Total | 0.2222 | 0.3334 | 0.4445 |

## 4 CONCLUDING REMARKS

In this paper, we present a tutorial of hypercube family models, which includes the basic single dispatch model proposed by Larson (1974) and some of its extensions motivated by practical ESS applications. In order to describe the distinctive main particularities of each model as compared to the original hypercube model, we use the smallest, non-trivial structure that incorporates the characteristics of each system under analysis. These structures, referred to as "toy-models", are

used to provide useful insights into problems of interest, but are mostly unable to directly address the complexity of real world ESS.

The application of hypercube family models to analyze real world ESS are promising since they are often able to capture the dispatch particularities of these systems. Some interesting ongoing and future research includes the extension of these models to: (i) capture demand scenarios during different periods of the day or week of the ESS; (ii) represent priority disciplines to service queued calls; (iii) represent the re-dispatching of servers when the server can be dispatched during the trip back to its base; (iv) include travel time distributions in analyzing ESS where the travel times are a significant portion of the service times (*e.g.*, rural areas, highways); (v) consider multiple dispatch where the multiple servers are not sent simultaneously (*e.g.*, the second server is sent later, if required); (vi) perform transient analyzes in situations where the steady-state assumption is not reasonable. Furthermore, other motivating future research would be the development of more effective approximate hypercube methods for embedding optimization procedures in analyzing large-scale ESS, in which the use of the exact models is computationally prohibitive.

## A APPENDIX

The probability that server 1 is the first server to become free when all servers are busy is given by $P(X_1 = \min(X_1, X_2, X_3)$. By using an auxiliary random variable $Z = \min(X_2, X_3)$, we can write the desired probability as $P(X_1 < Z)$. To obtain the distribution of $Z$, we observe that $X_2$ and $X_3$ are independent exponential random variables with parameters $\mu_2$ and $\mu_3$, respectively, so that:

$$P(Z > t) = P(X_2 > t)P(X_3 > t) = (e^{-\mu_2 t})(e^{-\mu_3 t}) = e^{-(\mu_2+\mu_3)t}$$

or

$$P(Z \leq t) = 1 - e^{-(\mu_2+\mu_3)t}$$

It turns out that $Z = \min(X_2, X_3)$ has an exponential distribution with parameter $(\mu_2 + \mu_3)$. Then the desired probability can be obtained as:

$$P(X_1 < Z) = \int_{x_1=0}^{\infty} \int_{z=x_1}^{\infty} (\mu_1 e^{-\mu_1 x_1}) \left[(\mu_2 + \mu_3) e^{-(\mu_2+\mu_3)z}\right] dx_1 dz$$

$$P(X_1 < Z) = \int_{x_1=0}^{\infty} (\mu_1 e^{-\mu_1 x_1}) \left[e^{-(\mu_2+\mu_3)x_1}\right] dx_1$$

By multiplying and dividing the integrand by $\mu = (\mu_1 + \mu_2 + \mu_3)$, we obtain:

$$P(X_1 < Z) = \left(\frac{\mu_1}{\mu}\right) \int_{x_1=0}^{\infty} \mu e^{-\mu x_1} dx_1 \, .$$

Since the integrand in this expression is the density function of an exponential random variable, the integral is 1 and the probability that unit 1 is the first one to complete service amounts to $\mu_1/\mu$.

## ACKNOWLEDGEMENTS

We would like to thank the three anonymous reviewers for their useful comments and suggestions. This research was partially supported by CNPq, Brazil, and Fonds AXA pour la Recherche, France.

## REFERENCES

[1] ALBINO JCC. 1994. Quantificação e locação de unidades móveis de atendimento de emergência e interrupções em redes de distribuição de energia elétrica: aplicação do Modelo Hipercubo. Dissertation, Universidade Federal de Santa Catarina, Florianópolis, Brazil.

[2] ATKINSON JB, KOVALENKO IN, KUZNETSOV N & MYKHALEVYCH KV. 2006. Heuristic solution methods for a hypercube queueing model of the deployment of emergency systems. *Cybernetics and Systems Analysis*, **42**(3): 379–391.

[3] ATKINSON JB, KOVALENKO IN, KUZNETSOV N & MYKHALEVYCH KV. 2008. A hypercube queueing loss model with customer-dependent service rates. *European Journal of Operational Research*, **191**(1): 223–239.

[4] BATTA R, DOLAN JM & KRISHNAMURTHY NN. 1989. The maximal expected covering location problem: Revisited. *Transportation Science*, **23**: 277–287.

[5] BRANDEAU M & LARSON RC. 1986. Extending and applying the hypercube queuing model to deploy ambulances in Boston. In: SWERSEY AJ & IGNALL EJ. (Eds). Delivery of Urban Services. *TIMS Studies in the Management Science*, **22**: Elsevier, 121–153.

[6] BROTCORNE L, LAPORTE G & SEMET F. 2003. Ambulance location and relocation models. *European Journal of Operational Research*, **147**: 451–63.

[7] BURWELL TH, JARVIS JP & MCKNEW MA. 1993. Modeling co-located servers and dispatch ties in the hypercube model. *Computers & Operations Research*, **20**(2): 113–119.

[8] CHELST K & BARLACH Z. 1981. Multiple unit dispatches in emergency services: models to estimate system performance. *Management Science*, **27**(12): 1390–1409.

[9] CHIYOSHI F, GALVÃO RD & MORABITO R. 2000. O uso do modelo hipercubo na solução de problemas de localização probabilísticos. *Gestão & Produção*, **7**(2): 146–174.

[10] CHIYOSHI FY, GALVÃO RD & MORABITO R. 2001. Modelo hipercubo: análise e resultados para o caso de servidores não-homogêneos. *Pesquisa Operacional*, **21**: 199–218.

[11] CHIYOSHI FY, GALVÃO RD & MORABITO R. 2003. A note on solutions to the maximal expected covering location problem. *Computers & Operations Research*, **30**: 87–96.

[12] COSTA DMB. 2004. Uma metodologia iterativa para determinação de zonas de atendimento de serviços emergenciais. Dissertation, Universidade Federal de Santa Catarina, Florianópolis, Brazil.

[13] GALVÃO RD, CHIYOSHI FY, ESPEJO LGA & RIVAS MPA. 2003. Solução do problema de localização de máxima disponibilidade utilizando o modelo hipercubo. *Pesquisa Operacional*, **23**: 61–78.

[14] GALVÃO RD, CHIYOSHI FY & MORABITO R. 2005. Towards unified formulations and extensions of two classical probabilistic location problems. *Computers & Operations Research*, **32**: 15–33.

[15] GALVÃO RD & MORABITO R. 2008. Emergency service systems: The use of the hypercube queueing model in the solution of probabilistic location problems. *International Transactions in Operational Research*, **15**: 525–549.

[16] GEROLIMINIS N, KARLAFTIS M & SKABARDONIS A. 2009. A spatial queuing model for the emergency vehicle districting and location problem. *Transportation Research B*, **43**: 798–811.

[17] GEROLIMINIS N, KEPAPTSOGLOU K & KARLAFTIS M. 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.

[18] GOLDBERG JB. 2004. Operations research models for the deployment of emergency services vehicles. *EMS Management Journal*, **1**(1): 20–39.

[19] HALPERN J. 1977. The accuracy of estimates for the performance criteria in certain emergency service queueing systems. *Transportation Science*, **11**: 223–242.

[20] IANNONI AP & MORABITO R. 2007. A multiple dispatch and partial backup hypercube queuing model to analyze emergency medical systems on highways. *Transportation Research E*, **43**(6): 755–771.

[21] IANNONI AP, MORABITO R & SAYDAM C. 2008. A hypercube queueing model embedded into a genetic algorithm for ambulance deployment on highways. *Annals of Operations Research*, **157**(1): 207–224.

[22] IANNONI AP, MORABITO R & SAYDAM C. 2009. An optimization approach for ambulance location and the districting of the response segments on highways. *European Journal of Operational Research*, **195**: 528–542.

[23] IANNONI AP, MORABITO R & SAYDAM C. 2011. Optimization large-scale emergency medical system operations on highways using the hypercube queuing model. *Socio-Economic Planning Sciences*, **45**: 105–117. doi:10.10.16/j.seps.2010.11.001.

[24] JARVIS JP. 1985. Approximating the equilibrium behavior of multi-server loss systems. *Management Science*, **31**: 235–239.

[25] KOLESAR P & SWERSEY AJ. 1986. The deployment of urban emergency units: a survey. In: Management science and the delivery of urban services, edited by A. SWERSEY & E. INGALL. *TIMS Studies in the Management Science*, North-Holland, Elsevier **22**: 87–119.

[26] LARSON RC. 1974. A hypercube queuing model for facility location and redistricting in urban emergency services. *Computers & Operations Research*, **1**: 67–95.

[27] LARSON R & MCKNEW MA. 1982. Police patrol-initiated activities within a systems queuing model. *Management Science*, **28**(7): 759–774.

[28] LARSON RC. 2004. OR models for homeland security. *OR/MS Today*, **31**: 22–29.

[29] LARSON RC & ODONI AR. 1981. Urban operations research. Prentice Hall. New Jersey.

[30] LUQUE L. 2007. Um estudo dos métodos de solução do modelo hipercubo de filas para sistemas emergenciais de grande porte. Dissertation, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, Brazil.

[31] MARIANOV V & SERRA D. 2003. Location-allocation of multiple-server service centers with constrained queues or waiting times. *Annals of Operations Research*, **111**: 35–50.

[32] MARIANOV V, BOFFEY B & GALVÃO RD. 2009. Optimal location of multi-server congestible facilities operating as M/Er/m/N queues. *Journal of the Operational Research Society*, **60**: 674–684.

[33] MENDONÇA FC & MORABITO R. 2001. Analyzing emergency service ambulance deployment on a Brazilian highway using the hypercube model. *Journal of the Operation Research Society*, **52**: 261–268.

[34] MORABITO R, CHIYOSHI F & GALVÃO R. 2008. Non-homogeneous servers in emergency medical systems: practical applications using the hypercube queueing model. *Socio-Economic Planning Sciences*, **42**: 255–270.

[35] OWEN SH & DASKIN MS. 1998. Strategic facility location: A review. *European Journal of Operational Research*, **111**: 423–447.

[36] RAJAGOPALAN HK, SAYDAM C & XIAO J. 2008. A multiperiod set covering location model for a dynamic redeployment of ambulances. *Computers & Operations Research*, **35**(3): 814–826.

[37] REVELLE CS. 1989. Review, extension and prediction in emergency service siting models. *European Journal of Operational Research*, **40**: 58–69.

[38] REVELLE CS & EISELT HA. 2005. Location analysis: A synthesis and survey. *European Journal of Operational Research*, **165**: 1–19.

[39] SACKS SR & GRIEF S. 1994. Orlando Police Department uses OR/MS methodology, new software to design patrol districts. *OR/MS Today*, Baltimore, 30–32.

[40] SAYDAM C & AYTUG H. 2003. Accurate estimation of expected coverage: revisited. *Socio-Economic Planning Sciences*, **37**: 69–80.

[41] SWERSEY AJ. 1994. *Handbooks in OR/MS*. Amsterdam: Elsevier Science BV, **6**: 151–200.

[42] TAKEDA RA, WIDMER JA & MORABITO R. 2007. Analysis of ambulance decentralization in urban emergency medical service using the hypercube queueing model. *Computers & Operations Research*, **34**(3): 727–741.