Editor Responsável: Prof. Dr. Hermes Moretti Ribeiro da Silva

Recebido: 03/11/2021
Aprovado: 17/11/2021

Towards Unifying Scheduling and Location Problems: A Non-Stationary Hypercube Model

Regiane Máximo Siqueira[1], State University of São Paulo Julio de Mesquita Filho, Bauru, São Paulo, Brazil
Caio Vitor Beojone[2], École Polytechnique Fédérale de Lausanne, Switzerland

RESUMO

Propósito – Este trabalho tem como objetivo desenvolver um modelo hipercubo não-estacionário capaz de unir as propriedades que os modelos para ambos os problemas de localização e programação de turnos.
Framework Teórico – Apresentamos o modelo proposto usando uma cadeia de Markov de tempo discreto-contínuo mista e comparamos com uma simulação de evento discreto por meio de um exemplo ilustrativo.
Design/metodologia/abordagem, – O método usado neste artigo é quantitativo com uma comparação entre uma abordagem de simulação e um modelo exato.
Resultados – Os resultados mostram uma grande similaridade entre os dois modelos. No entanto, o modelo proposto não apresenta ruído em medidas de desempenho como tempos de espera e tempo de deslocamento. No entanto, o estudo de seus resíduos revelou que o modelo proposto apresenta menor sensibilidade a eventos, como finais de turnos e imperfeições nas preferências de despacho. Novos estudos podem reduzir essa variação por meio de melhorias nos cálculos das medições de desempenho.
Pesquisa, Prática & Implicações Sociais – Os resultados citados sugerem que o modelo proposto pode se tornar uma opção para aplicações unindo problemas de localização e programação de turnos.
Originalidade/valor – Ao desenvolver problemas de localização, buscamos modelos que sejam capazes de representar as características geográficas pertinentes ao problema. Por outro lado, ao desenvolver problemas de programação de turnos, buscamos modelos capazes de captar flutuações transitórias nos componentes (como demanda, tempos de atendimento, mão de obra disponível, entre outros) de tal sistema. Portanto, na busca de melhorar o desempenho diário de sistemas, tais como sistemas de serviço de emergência (ambulâncias, polícia, bombeiros), usando qualquer um dos dois problemas individualmente, pode levar a conclusões errôneas.
Palavras-Chave - Sistemas de Serviço de Emergência; Teoria de filas; Hipercubo não-estacionário; Simulação de eventos discretos; Medidas de desempenho.

ABSTRACT

Purpose *– this paper aims to develop a non-stationary hypercube model capable of uniting the properties that models for both problems seek (location and shift-scheduling problems).*
Theoretical framework – *We present the proposed model using a mixed discrete-continuous time Markov chain and compares it to a discrete-event simulation through an illustrative example.*
Design/methodology/approach *– The method used in this paper is quantitative with a comparison between an approach of simulation and an exact model.*
Findings *– The results show a high similarity between both models. However, the proposed model does not present noise in performance measures such as waiting times and travel times. Nevertheless, the study of their residuals revealed that the proposed model has a lower sensitivity to events, such as shift endings and imperfections in dispatch preferences. Further studies may reduce such a variation by improvements in the calculations of performance measurements.*
Research, Practical & Social implications – *The mentioned results suggest that the proposed model may become an option for applications uniting location and shift-scheduling problems.*
Originality/value *– When developing location problems, we seek models that are capable of representing the pertinent geographic characteristics to the problem. On the other hand, when developing shift-scheduling problems, we seek models capable of capturing transient fluctuations in the components (such as demand, service times, available workforce, among others) of such a system. Therefore, in the search to improve the daily operations of systems, such as emergency service systems (ambulances, police, firefighters) using either of the two problems individually, it may lead to flawed conclusions.*
Keywords *- Emergency Service Systems; Queueing Theory; Hypercube non-stationary; Discrete Event Simulation; Performance Measurement.*

1. Av. Eng. Luís Edmundo Carrijo Coube, 14-01 - Vargem Limpa, Bauru - SP, 17033-360; regiane@feb.unesp.br, https://orcid.org/0000-0002-4695-2678; 2. caio.beojone@epfl.ch; https://orcid.org/0000-0002-6491-7104.
SIQUEIRA, R.M.; BEOJONE, C.V. Towards Unifying Scheduling and Location Problems: A Non-Stationary Hypercube Model. **GEPROS. Gestão da Produção, Operações e Sistemas**, v.16, nº 4, p. 137 – 161, 2021.

DOI: http://dx.doi.org/10.15675/gepros.v16i4.2865

1. INTRODUCTION

Operating services, such as Emergency Service Systems (ESS), require that the available staffing capacity matches the demand for service throughout the day. The manager's challenge is to schedule service hours to match with demand at different times of the day, while keeping costs under control and respecting all applicable laws (INGOLFSSON *et al.*, 2002). The fundamental requirement is that there is enough staff working to achieve planned service levels (GREEN *et al.*, 2001).

Scheduling servers is a challenge for any service, from banks, restaurants, stores, and airports, to call centers (INGOLFSSON *et al.*, 2010). Call centers may be the most studied operation for such problems, since they may be part of the customer service, help desk, and ESS operation (GANS *et al.*, 2003). Its operation is subject to uncertainties regarding demand and staff availability (MANDELBAUM *et al.*, 2009; PATRICK *et al.*, 2008).

Scheduling problems have a non-linear nature due to time-varying and stochastic characteristics. For example, the arrival processes may follow a non-homogeneous Poisson process; the number of working servers may vary in time. As such, these problems have dynamic and stochastic aspects, being called non-stationary, and Queueing Theory is one of the capable tools to handle these features. Those interested in solving scheduling problems may look for Defraeye e Van Nieuwenhuyse (2016) and the references therein.

Chapman-Kolmogorov differential equations represent the exact behavior of a non-stationary queueing system (INGOLFSSON *et al.*, 2002). These equations only have analytical solutions in special cases – when the system has infinite servers and the arrival and service rate functions have no discontinuities (SCHWARZ *et al.*, 2016). Therefore, in most cases they are numerically solved by methods such as Euler or Runge-Kutta. Although computationally expensive, these numerical solutions serve as benchmarks for approximations (GILLARD; KNIGHT, 2014; SCHWARZ *et al.*, 2016).

Among the challenges to model such systems, one must cope with time-varying staffing behavior. Ingolfsson et al. (2007) presents, continuing from Ingolfsson (2005), a way to model a specific behavior of systems during end-of-shifts. Such a behavior is called end-of-shift discipline. The problem shown in Ingolfsson (2005) was to represent end-of-shift in situations that the servers must finish the ongoing services before leaving the system. The authors refer to this end-of-shift discipline as exhaustive (we call it as non-preemptive

throughout the text). The counterpoint to an exhaustive discipline is a pre-emptive discipline, where servers interrupt their ongoing services so that users go back into the head of the line.

Many ESSs are characterized as spatially distributed queue systems. Examples of these services include police, firefighters and ambulances (GALVÃO; MORABITO, 2008). Usually, the hypercube queuing model (LARSON, 1974) is used to model such services. To the best of our knowledge, the work with the hypercube model use stationary approximations, analyzing systems at their peak period, as in Takeda *et al*. (2007), Atkinson *et al.* (2008), Burwell *et al.* (1993), Iannoni *et al.* (2015), Geroliminis *et al.* (2011), among others.

For being able to cope with a city's geographic complexity and complex dispatch policies, hypercube models are a popular tool for probabilistic location problems. As a descriptive model, the hypercube model alone does not find solutions to location problems but is able to provide decision makers with performance measures for any previous server location scheme (MARIANOV; REVELLE, 1996). Note that the scope of this study focuses only on the descriptive model (hypercube model). Therefore, we do not present or discuss location problems in details. Readers interested in further literature on the hypercube model and location problems are encouraged to read Boyaci and Geroliminis (2015), Rodrigues *et al*. (2017) and Owen and Daskin (1998) and the references therein.

Although many studies have developed extensions to Larson's (1974) classic hypercube model on the the operation of the most diverse ESSs, few use the hypercube model beyond the peak period. One of the examples is Souza *et al.* (2015), who considered three different periods of the day (morning, afternoon and night), when studying a Brazilian Emergency Medical System (EMS). Another example is Rajagopalan (2008), which also used an approximate hypercube model in three periods of the day. Ansari *et al.* (2017) studies an EMS in two periods of the day.

However, an analysis of ESSs only at limited periods of the day may fail for several reasons. Firstly, only observe arrival rate stability. This can be a problem in case there is any end-of-shift or meal breaks during the period of steady demand. Secondly, experience shows that ESSs have low event frequencies and therefore take longer to reach equilibrium. This problem has already been pointed out in Green *et al. (*1995).

As such, the strict application of a location problem can be compromised, turning the

combination with scheduling problems attractive. Therefore, it is necessary to develop a model capable of dealing with the dynamic, non-linear, and spatial aspects of the daily operation of an ESS.

In this study, the hypercube model considers time-varying parameters such as arrival rates, number of servers, service times, etc. We present the proposed model using a mixed discrete-continuous time Markov chain. The model considers non-preemptive end-of-shift discipline (INGOLFSSON *et al.*, 2007). Then, the results of the proposed model are compared to those obtained by a discrete-event simulation (independent of the proposed model) in an illustrative example. Simulation is usually used when developing hypercube models to check for possible errors during the modeling process, since simulations are capable of providing the same performance measures with similar assumptions (IANNONI *et al*., 2015). The comparison evaluates the model's ability to represent the same system as a noise-free simulation, as well as its computational performance.

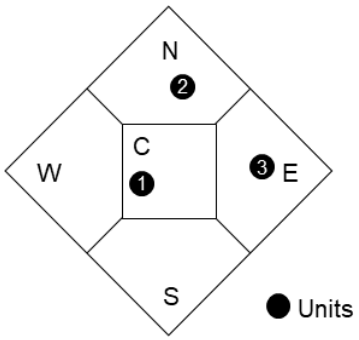
The study is structured as follows. Section 2 presents the non-stationary hypercube model using an illustrative example. Section 3 describes the construction of the discrete-event simulation, as well as its characteristics and data collection for comparison. Section 4 presents the computational results obtained by the proposed model and the simulation and the comparisons of the performance measures. Finally, Section 5 provides final considerations and further research propositions.

2. THEORETICAL FOUNDATION

2.1 Non-Stationary Hypercube Model

An illustrative example helps to present the proposed model. shows a five-atom system that operates over 24 hours. Atoms suffer no changes in size or identification over time. The system has three servers located at North, Central and East atoms, respectively, operating in three shifts. The first shift is the only one in which all three servers operate, in the other two shifts the server of the east atom is removed. All operating servers are scheduled to leave at the end of a shift. They follow a non-preemptive end-of-shift discipline. Moreover, they do not have to wait for the return of a unit to start operating. For modeling purposes, the queue has been limited to 5 users.

Figure 1 – Illustrative example map containing atoms and location of units, accompanying the dispatch preference list and inter-atom travel times.



Dispatch preference list

Atom	Preference		
C	1	2	3
N	2	3	1
S	1	3	2
E	3	1	2
W	3	2	1

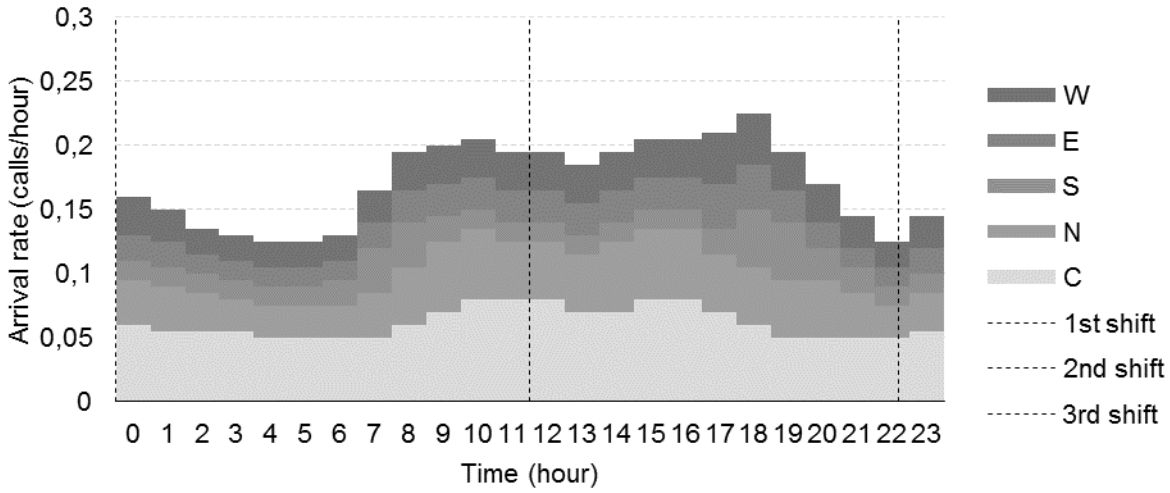
Inter-atom average travel time

Atoms	C	N	S	E	W
C	10	15	15	15	10
N	15	10	20	15	15
S	15	20	10	15	15
E	15	15	15	10	20
W	10	15	15	20	10

Source: The authors.

The call arrival process follows a non-homogeneous Poisson process (KIM; WHITT, 2014), which is a result of time-varying probabilities for a user to enter the system. A common approach to model this process is to break time into intervals with constant rates (BROWN *et al.*, 2005). Figure 2 shows total arrival rates throughout the 24 hours of operation and server work shifts. Service times are discussed in Section 3, along with each scenario.

Figure 2 – Arrival rates and server shifts.



Source: The authors.

Since the hypercube model is an expansion of the states of a *M*/*M*/*s*/*K* model (for cases with a limit on the maximum queue size) to represent the servers individually, the system's dynamic is straightforwardly obtained from Chapman-Kolmogorov differential equations. (TAHA, 2008). The system of differential equations (represents system behavior throughout the first shift (between 6 and 15 hours) when all servers are operating.

$$
\begin{aligned}
\pi'_{000}(t) &= -\lambda(t)\pi_{000}(t) + \mu_1(t)\pi_{100}(t) + \mu_2(t)\pi_{010}(t) + \mu_3(t)\pi_{001}(t) \\
\pi'_{100}(t) &= -\big(\lambda(t) + \mu_1(t)\big)\pi_{100}(t) + \mu_2(t)\pi_{110}(t) + \mu_3(t)\pi_{101}(t) \\
&\quad + \big(\lambda_C(t) + \lambda_S(t)\big)\pi_{000}(t) \\
\pi'_{010}(t) &= -\big(\lambda(t) + \mu_2(t)\big)\pi_{010}(t) + \mu_1(t)\pi_{110}(t) + \mu_3(t)\pi_{011}(t) + \lambda_N(t)\pi_{000}(t) \\
\pi'_{001}(t) &= -\big(\lambda(t) + \mu_3(t)\big)\pi_{001}(t) + \mu_1(t)\pi_{101}(t) + \mu_2(t)\pi_{011}(t) \\
&\quad + \big(\lambda_E(t) + \lambda_W(t)\big)\pi_{000}(t) \\
&\vdots \\
\pi'_{111}(t) &= -\big(\lambda(t) + \mu(t)\big)\pi_{111}(t) + \mu(t)\pi_{111\vee1}(t) + \lambda(t)\big(\pi_{011}(t) + \pi_{101}(t) + \pi_{110}(t)\big) \\
\pi'_{111\vee1}(t) &= -\big(\lambda(t) + \mu(t)\big)\pi_{111\vee1}(t) + \mu(t)\pi_{111\vee2}(t) + \lambda(t)\pi_{111}(t) \\
&\vdots \\
\pi'_{111\vee Q}(t) &= -\big(\lambda(t) + \mu(t)\big)\pi_{111\vee Q}(t) + \mu(t)\pi_{111\vee Q+1}(t) + \lambda(t)\pi_{111\vee Q-1}(t)
\end{aligned}
\tag{1}
$$

If the system does not face changes in the number of servers and in the arrival rate, solving this set of differential equations (a continuous-time Markov chain) would suffice. Otherwise, a discrete-time Markov chain must represent the system behavior the moment a server leaves the system.

Table 1 presents a list of the variables used throughout this study. Note that system states are represented generically by the letters r and q and can be described as follows: $r = \{N_r \vee Q\}$. Performance measures have been left out of this table and are presented in more detail in Section 2.3.

Table 1 – Description and definitions for the variables and parameters used.

Variable	Variable Type	Meaning	Representation
$\pi(t)$	Real vector	System state probabilities vector for instant t.	$\{\pi_r(t), \pi_q(t), \dots\}$
$\mu(t)$	Calls/hour	System total service rate for instant t.	$\sum_i \mu_i$
N_A	Natural number	Number of atoms in the system.	-
$\lambda(t)$	Calls/hour	System total arrival rate for instant t.	$\sum_j \lambda_j$
$A(t)$	Real matrix	Continuous-time Markov chain for instant t. Coefficient matrix $a_{r,q}$.	$\begin{bmatrix} a_{r,q} & \cdots \\ \vdots & \ddots \end{bmatrix}$
$B(t)$	Real matrix	Discrete-time Markov chain for instant t. Transition matrix with elements $b_{r,q}$.	$\begin{bmatrix} b_{r,q} & \cdots \\ \vdots & \ddots \end{bmatrix}$
N_r	Binary vector	Indicates which servers are occupied (1) or not (0) in state r.	$\{n_1, n_2, n_3\}$
$s(t)$	Binary vector	Indicates which servers are currently operating (1) or not (0) for instant t.	$\{s_1(t), s_2(t), s_3(t)\}$
δn	Binary vector	Indicates which users in attendance will be "ejected" (1) or not (0).	$\{\delta n_1, \delta n_2, \delta n_3\}$
δs	Binary vector	Indicates servers that are leaving (1) or not (0). It exists for all end-of-shifts (even if no server is leaving).	$\{\delta s_1, \delta s_2, \delta s_3\}$
δQ	Binary vector	Number of queued users assigned to newly available servers.	$\{\delta Q_1, \delta Q_2, \delta Q_3\}$
Q	Natural number	Number of users in the queue.	-
$P(\delta Q, t)$	Real number	Occurrence probability of vector δQ at instant t.	-
$\{Pn(N_A, Q)\}$	Set	Set of permutations that queued users can assume.	-
t^- and t^+	Time (hours)	Time instant just before t and right after instant t, respectively.	

Source: The authors.

2.2 Exhaustive end-of-shift discipline (non-preemptive)

If we consider that the illustrative example is an EMS, servers are now called ambulances. The model follows a non-preemptive end-of-shift discipline, that is ambulances always finish their calls before leaving the system at the end of the shift. Therefore, as in the model *M*(*t*)/*M*/*s*(*t*) by Ingolfsson *et al.* (2007), users in attendance by ambulances to leave must be “ejected” from the system, since their servers do not answer to any other users afterward. Keep in mind that the model does not eject users from the system, in reality. They are only disregarded in the analysis because they do not affect further performance measurements.

As in Ingolfsson *et al.* (2007), the model can be defined as a mixed discrete-continuous time Markov chain. Equation (1) illustrates how the model works.

$$Mode = \begin{cases} \text{Continuous time Markov Chain } \pi'(t) = \pi(t)A(t), & \text{if } \delta s_i(t) = 0 \ \forall i \\ \text{Discrete time Markov Chain } \pi(t^+) = \pi(t^-)B(t), & \text{otherwise} \end{cases} \quad (1)$$

The transition matrix $B(t)$ is built from the following events: users are "ejected", and queued users are distributed to the new available servers. The following equations (3-5) work for the three-server illustrative example. However, one can easily be extended them to systems with more servers. The first set of elements $b_{r,q}$ from $B(t)$ is computed using Equation (3). They represent the situation where users are "ejected" (δn_i) and there are no calls in the queue to be distributed to the new servers. The second set of elements is computed using Equation (4). In this case, apart from "ejecting" users, the queued ones are assigned to the new available servers and at least one server will remain available. Finally, Equation (5) represents the case that even after "ejecting" users and distributing queued ones, the system will remain saturated (no available servers).

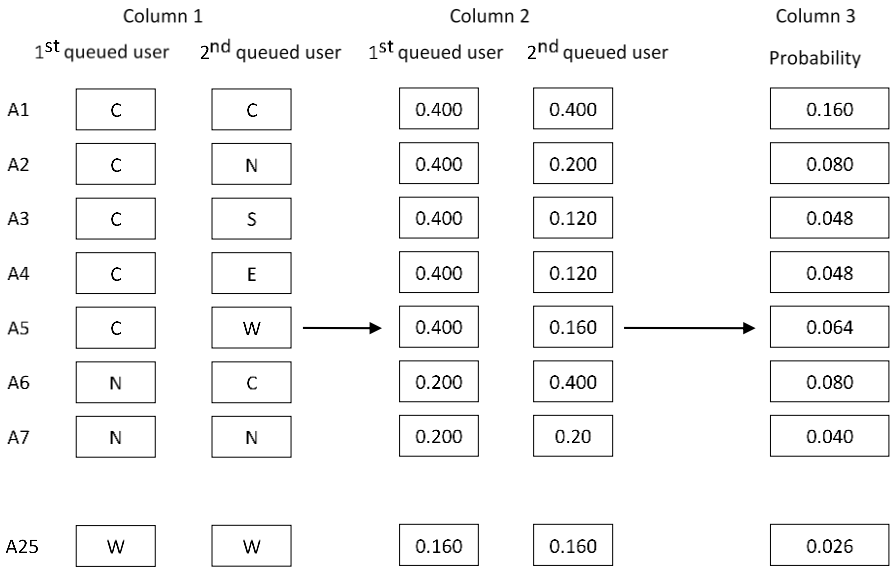
$$b_{\{n_1,n_2,n_3 \vee 0\},\{n_1-\delta n_1,n_2-\delta n_2,n_3-\delta n_3 \vee 0\}} = 1, for n_i = 0,1, \delta n_i = min(\delta s_i, n_i), \forall i \quad (2)$$

$$b_{\{n_1,n_2,n_3 \vee Q\},\{n_1-\delta n_1+\delta Q_1,n_2-\delta n_2+\delta Q_2,n_3-\delta n_3+\delta Q_3 \vee 0\}} = P(\delta Q,t), for Q > 0, Q = \sum_i \delta Q_i, n_i = 0,1, n_1 - \delta n_1 + \delta Q_1 \leq 1, \delta n_i = min(\delta s_i, n_i), \forall i \quad (3)$$

$$b_{\{n_1,n_2,n_3 \vee Q\},\{1,1,1 \vee K\}} = 1, for n_i = 0,1, Q + N^{--\delta s \geq N^+, K = Q - \delta s, \forall i} \quad (4)$$

To calculate the likelihood of assigning queued users to servers $P(\delta Q, t)$, it is necessary to estimate the possible permutations that the queued users form and the probabilities of the existence of each permutation. It happens because the model does not keep track of queued users' locations. The process for estimating the probability of each permutation is illustrated in Figure 3 for a situation with two queued users during the shift change at 06:00 hrs.

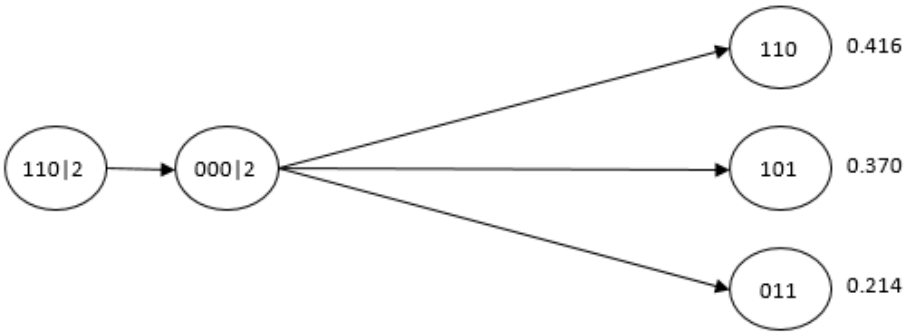
Figure 3 – Computing permutations of queued users and their probabilities (example with two queued users).



Source: The authors.

The model uses the dispatch preference list to assign queued users from each permutation to new servers. Figure 4 presents the result of the transition to the mentioned instant. Note that in instant t^- server 3 is marked as idle {0}. However, it is not operating during the third shift, thus the system gets saturated when servers 1 and 2 are busy. The transition is shown in two steps. In the first step, the users with busy servers are "ejected" because they are leaving the system for new servers to enter in their positions. Recall that users are not ejected from the system, in reality. In the second step, the model computes to which state each permutation of queued users would take the system. In other words, it assigns the queued users (following a FIFO discipline) to the newly available servers. For example, permutation 3 is assigned to state {101} because the first user in the queue originates from atom C and has server 1 as its preferred server; while the second call in the queue originates from atom S and would also have server 1 as preferred, however server 3, the first backup to atom S, is sent (server 1 got busy with the first call in the queue).

Figure 4 – Example of an instantaneous state transition occurring at the end of a shift, illustrating the assignment of queued users' permutations to the new servers.



Source: The authors.

}

2.3 Non-stationary hypercube model assumptions

The construction of a non-stationary hypercube model, as shown up to this point, is subject to the adherence to simplifying assumptions (such as the stationary hypercube model). The assumptions are listed below:

i) Existence of geographic atoms: the region where services are provided should be divided into N_A geographic atoms, where each atom corresponds to an independent source of users. Atoms cannot change over time;

ii) Arrival process: follows a Poisson process (usually non-homogeneous). The users of each atom request service through a Poisson process, where the calls are independent to each other. In addition, the arrival rate functions, $\lambda_j(t)$, for each atom must be known throughout the period of analysis;

iii) Travel Times to Atoms: The function of travel times $\zeta_{kj}(t)$ of each pair of atoms k and j must be known or estimated for the entire period of analysis;

iv) System servers: there is a vector $s(t)$ which represents the number of servers spatially distributed throughout the system over the period of analysis. All servers can travel and serve at any of the atoms. There is a known vector, $\delta s(t)$, which represents the number of servers ending their shift at instant t. The end of a shift should follow a well-defined discipline according to system operation, primarily a preemptive (not discussed in this paper) or non-preemptive discipline;

v) Server location: Server location should be known at least probabilistically throughout the period of analysis;

vi) Server dispatches: To answer any call, only one server is sent to the location. If no servers are available, calls are queued, with a *FIFO* discipline, or are considered system losses (case queue size is limited);

vii) Server dispatch policy: for every instant, there is a dispatch preference list for each atom. The list may change over time;

viii) Service Time: The service time of a server encompasses the setup time and the time taken to return to the base (or area) of origin. Service times should be exponentially distributed and do not vary over the period of analysis; and

ix) Travel times and service times: Service times are the sum of on-scene time and travel times. Considering that average travel times of servers vary slowly over time, service times need to be calibrated for each instant of time. t in the following manner: $\mu_i^{-1}(t) = \overline{T}_i^{on-scene} + \overline{TU}_i$.

x) Initial solution of the system: The initial situation of the system must be known or estimated probabilistically in the form of a probability vector $\pi(0)$.

2.4 Performance measures

The solution of the mixed discrete-continuous time Markov chain is used to calculate various performance measures for the system. This section shows selected performance measures. The notations used to calculate performance measures are shown in Table 2.

Table 2 – Notations for performance measures.

Measure	Meaning
$SL(t,\tau)$	Service level at instant t, for a length of time τ.
$P_S(t)$	Instantaneous probability of saturation
a	Expected number of completed services.
$E[L_Q(t)]$	Expected number of queued users at instant t.
$P(W_Q(t) > \tau)$	Likelihood that an arriving user will have to wait more than τ time units at instant t.
$E[X]$	Expected value for any variable X.
$F_X(X)$	Cumulative probability function ($P(X \leq x)$) for a variable X.
$E[W_Q(t)]$	Expected waiting time for an arriving user at instant t.
$\rho_i(t)$	Instantaneous workload of server i.

$f_{ij}(t)$	Instantaneous dispatch frequency of a server i to an atom j.
$f_{ij}^{(nq)}(t)$	Instantaneous dispatch frequency of a server i to an atom j and incur no queue delay.
$f_{ij}^{(q)}(t)$	Instantaneous dispatch frequency of a server i to an atom j and incur a positive delay.
$\zeta_{kj}(t)$	Average travel time between atoms k and j at instant t.
$l_{ik}(t)$	Probability that a server i has its base on atom k at instant t.
$\overline{T}_Q(t)$	Average travel time to a random service request that is delayed in queue at instant t.
$\overline{T}(t)$	System-wide average travel time at instant t.
$\overline{TA}_j(t)$	Average travel time to atom j at instant t.
$\overline{TU}_i(t)$	Average travel time of server i at instant t.

Source: The authors.

Equation (6) shows the service level calculation as the probability of a user being attended to within τ units of time (adapted from Ingolfsson *et al*., 2007). Consider that $\{N_r = s(t) \vee i\} \equiv \{s(t) \vee i\}$ represents the saturated states with i users in the queue. Note that a is the simplified notation of the expected number of services finished in the interval $(t, t+\tau]$. Case $\tau = 0$, then $\sum_{j=0}^{i} \frac{a^{-j}e^{-a}}{j!} = 1$. Therefore, $SL(t,0)$ is the probability that an arriving user will receive service without waiting. Thus, $P_S(t) \equiv 1 - SL(t,0)$.

$$SL(t,\tau) = \begin{cases} 1 - \sum_{q=0}^{K} \pi_{\{s(t)\vee q\}} \sum_{l=0}^{q} \frac{a^{-l}e^{-a}}{l!}, & if \nexists \delta s for (t, t+\tau] \\ 1 - \sum_{q:q+N(t)-\delta s \geq N(t+\tau)} \pi_{\{s(t)\vee q\}} \sum_{l=0}^{q} \frac{a^{-l}e^{-a}}{l!}, & otherwise \end{cases} \quad (5)$$

The average number of exits from the system, a, is calculated according to Equation (7). Note that service rate, $\mu_i(t)$, is considered as a function of time. It is considered that the service rate varies slowly over time and its variations are small compared to its total time. Because of this, it is possible to approximate the average number of exits considering a constant service time. The calculation was extended only for the case of a single end-of-shift throughout the interval. $(t, t+\tau]$.

$$a \equiv \sum_i \int_t^{t+\tau} \mu_i(u)du \cong \begin{cases} \sum_i \mu_i(t)\tau & if \nexists \delta s for (t, t+\tau] \\ \sum_i \mu_i(t)\epsilon + \sum_i \mu_i(t)(\tau - \epsilon) & otherwise \end{cases} \quad (6)$$

The instantaneous expected number of queued users, $E[L_Q(t)]$, is calculated by Equation (8). The calculation is the same as for $M(t)/M/s(t)/K$ systems. Where K is the maximum limit of users allowed in the queue.

$$E[L_Q(t)] = \sum_{l=1}^{K} l\pi_{\{s(t)\vee l\}}(t) \tag{7}$$

Ingolfsson (2005) shows the calculation for a user that needs to wait longer than τ units of time in a $M(t)/M/s(t)/K$. Here, this calculation is extended to calculate the instantaneous average waiting time. The first part of the calculation is the probability that an arriving user at time t will wait more than τ units of time, as shown in Equation (9). Note that new servers coming into operation not only increase the total service rate but also take queued users from the waiting line. Therefore, $v(t,\tau) = \sum_i n_i(t+\tau) + \delta s_i - n_i(t)$ represents the number of users who will leave the queue due to an end-of-shift in the interval $(t, t+\tau]$.

$$P(W_Q(t) > \tau) = \sum_{q=v(t,\tau)}^{K} \pi_{\{s(t)\vee q\}}(t) \sum_{i=1}^{q-v(t,\tau)+1} e^{-a} \frac{a^{i-1}}{(i-1)!} \tag{8}$$

It is important to remember that the expected value of a random variable can be calculated following Equation (10). Where the cumulative probability function is represented by $F_X(X) = P(X \leq x)$.

$$E[X] = \int_0^{\infty} (1 - F_X(X))dx \tag{9}$$

Therefore, it is derived that the expected value of the waiting time for an arriving user at instant t is given by Equation (11).

$$E[W_Q(t)] = \int_0^{\infty} P(W_Q(t) > \tau)d\tau \tag{10}$$

The workload is calculated directly by the sum of the instantaneous probabilities of the server being busy, as shown in Equation (12).

$$\rho_i(t) = \sum_{B\in M:n_i=1} \pi_B(t) \tag{11}$$

The calculation of dispatch frequency (Equation 13) has been separated for dispatch frequencies of requests that did not incur in delays (*nq*) (Equation 14), and the requests that incurred a positive delay (q) (Equation 15).

$$f_{ij}(t) = f_{ij}^{(nq)}(t) + f_{ij}^{(q)}(t) \tag{12}$$

$$f_{ij}^{(nq)}(t) = \frac{\lambda_j(t)}{\lambda(t)} \sum_{D \in E_{i,j}(t)} P_D(t) \tag{13}$$

$$f_{ij}^{(q)}(t) = \frac{\lambda_j(t)}{\lambda(t)} P_S(t) \frac{\mu_i(t)}{\mu(t)} \tag{14}$$

The average travel times of the system can be estimated from the function of the average travel times between atoms, $\zeta_{jk}(t)$. Using the server location matrix ($l_{ik}(t)$), the average travel time of a server to an atom can be calculated by Equation (16).

$$t_{ij}(t) = \sum_{k=1}^{N_A} l_{ik}(t)\zeta_{kj}(t) \tag{15}$$

The average travel time for calls subject to delays is calculated by Equation (17). Note that this calculation is different from the approximation seen in Larson (1974) and Larson and Odoni (1981) (and widely used in the hypercube model literature) as it explicitly supports heterogeneous servers and considers that some atoms may not have a server in there. As such, it is calculated based on the $f_{ij}^{(q)}$ and no longer on the proportion of arrivals of each atom.

$$\overline{T}_Q(t) = \frac{1}{P_S(t)} \sum_{i=1}^{N} \sum_{j=1}^{N_A} f_{ij}^{(q)}(t) t_{ij}(t) \tag{16}$$

The average travel time for the system can be calculated from Equation (18).

$$\overline{T}(t) = \sum_{i=1}^{N} \sum_{j=1}^{N_A} f_{ij}(t) t_{ij}(t) \tag{17}$$

Average travel times to atoms can be calculated by Equation (19).

$$\overline{TA}_j(t) = \frac{\sum_{i=1}^{N} f_{ij}(t) t_{ij}(t)}{\sum_{i=1}^{N} f_{ij}(t)} \tag{18}$$

Average server travel times can be calculated by Equation (20). Recall that this measure is used to calculate the average service time for server i, as shown in assumption (ix).

$$\overline{TU}_i(t) = \frac{\sum_{j=1}^{N_A} f_{ij}(t) t_{ij}(t)}{\sum_{j=1}^{N_A} f_{ij}(t)} \quad (19)$$

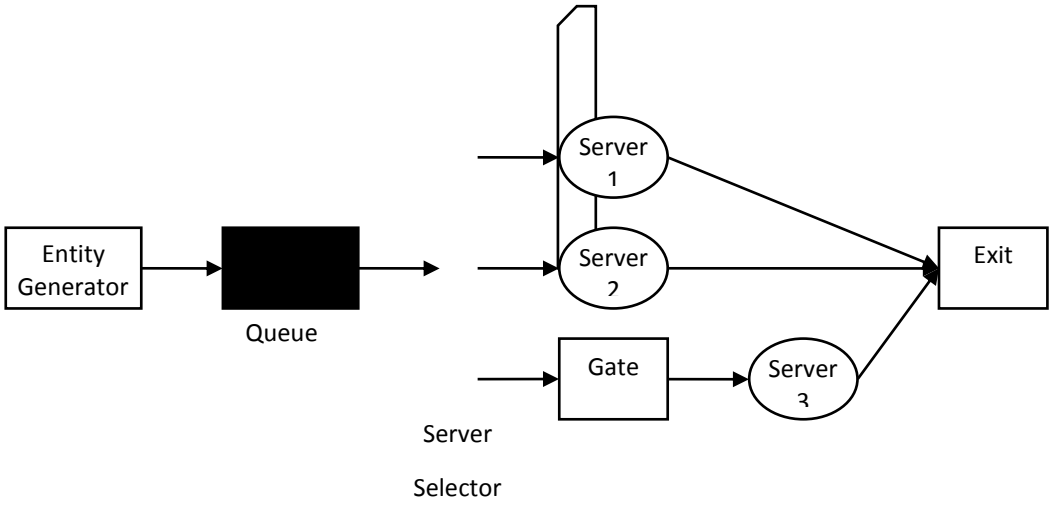
3. METHODOLOGICAL PROCEDURES

3.1 Simulation of discrete events

An independent discrete-event simulation model verifies the non-stationary hypercube model. The simulation uses the data from the illustrative example. The simulation aims to point out errors or limitations of modeling the proposed hypercube model.

Figure 5 provides a simplified schematic representation of the discrete-event simulation. We used the Simulink module found in the MATLAB software. Each block was programmed according to the assumptions presented for the non-stationary hypercube model.

Figure 5 – Schematic representation of discrete-event simulation.



Before describing the blocks, it is important to note that the arrows in the schematic figure represent the flow of entities (users) throughout the system. Another important point

is to define the characteristics that users have: time it was generated, location of the request, and time in the queue.

The 'Entity Generator' block has two functions. Firstly, it uses a random number generator that follows a non-homogeneous Poisson process to generate entities according to the total rate of arrival as shown in Figure 2. Before leaving the block, the location of the generated user is defined according to the fractions of the arrival rates of the atoms in relation to the total rate of the instant that the user was generated. No user can be stored within this block.

The next block is the 'Queue'. It follows a FIFO discipline and has capacity of up to 5 users. The time that the user queues is saved for the purpose of calculating performance measures. We also collect the number of queued users for the same purpose.

The 'Server Selector' represents the dispatch policy of the system (dispatch preference list. No user is allowed to remain in the server selector, serving only as a gateway and not affecting the system queue capacity.

The Gate used in front of Server 3 operates to delimit the shift in which Server 3 operates. Therefore, the gate is only open for the first shift (between 06:00 hrs. and 15:00 hrs.). For the rest of the time, this gate remains closed not allowing users to pass.

The next set of blocks is the server set. Firstly, the servers have predetermined specific on-scene times. Before calculating the total service time, a random on-scene time is added to the average travel time from the server to the user's location. This sum results in the service time. The on-scene time is exponentially distributed. If a service will end after the end of the server's shift, the service in completed at the moment of the shift change, simulating an "ejected" user. "Ejected" users had their service times ignored. The waiting time of a user is calculated once it enters any of the server blocks. We compute whether the server was busy or not at each time step to calculate their workloads.

Finally, after the service ends, users exit the system through the 'Exit' block. This block only collects the total number of users that have arrived.

4. RESULTS

The continuous time Markov Chain part of the non-stationary hypercube model was solved using the Runge-Kutta method through the "ode45" function found in the MATLAB. The simulation considered a total period of 360,000 hours. The simulation took 59.9 seconds per round, without calculating performance measures, and 152.5 seconds per round, with the calculated performance measures.

For the hypercube model, it was considered to start empty (in t=0) and runs for 48 hours with 24-hour cycles. Only the last 24-hour cycle was considered to build the figures and other results. Note that in the illustrative example, the servers have no mealtime breaks. One round of the proposed model took around 16.7 seconds, without calculating performance measures, and 595 seconds with calculated performance measures. Given that it only took 578 seconds to calculate the average waiting times, a costly calculation due to the use of numerical integration. Without this measurement, the total time would be 17.6 seconds, while the simulation would be 148.3 seconds.

Figure 6 presents the average waiting time for the hypercube model and the simulation. We established a threshold of τ=4 hours to calculate waiting time, given P(W_Q (t)>4)<10^(-4) at any time in the illustrative example and considering that the equation converges asymptotically. Figure 6a shows the results for both the hypercube model and an average obtained at 3-minute intervals for the simulation. Note that the decreases to near zero at the moment of the three shift handovers (06:00 hrs., 15:00 hrs. and 23:00 hrs.), as expected, as all servers leave the system and new ones start operating. Figure 6b, on the other hand, shows that models are better correlated when waiting times are shorter. As the average waiting times increase, the dispersion also increases with a tendency for the hypercube to calculate longer waiting times.

Figure 6 – Average waiting times (a) and correlation (b) between the simulation and the non-stationary hypercube model for the illustrative example.

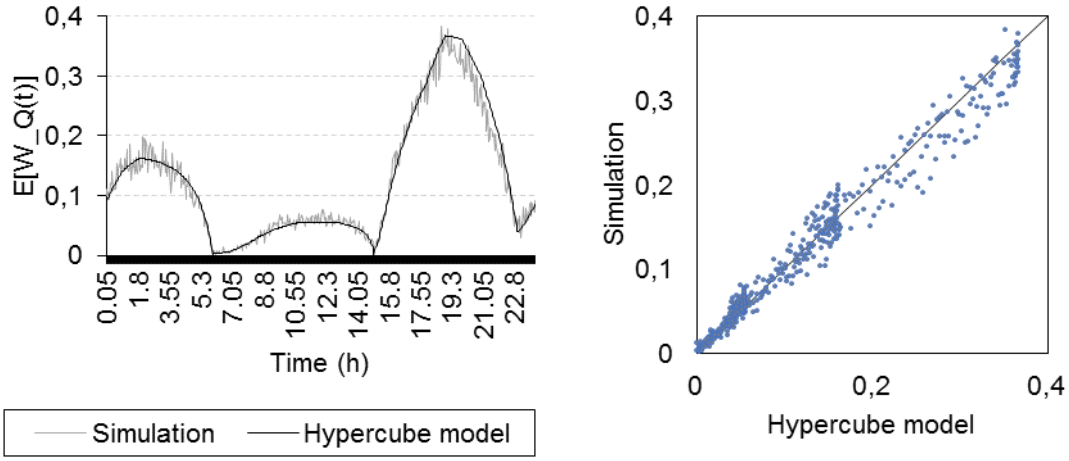
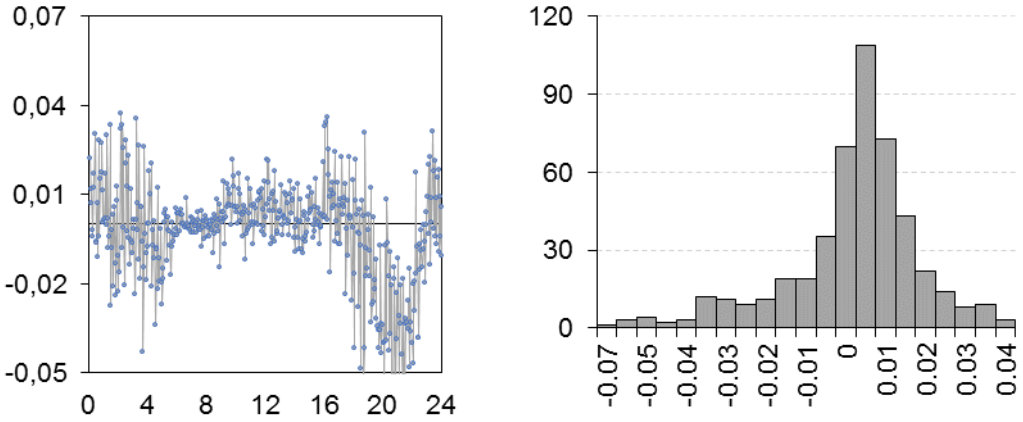


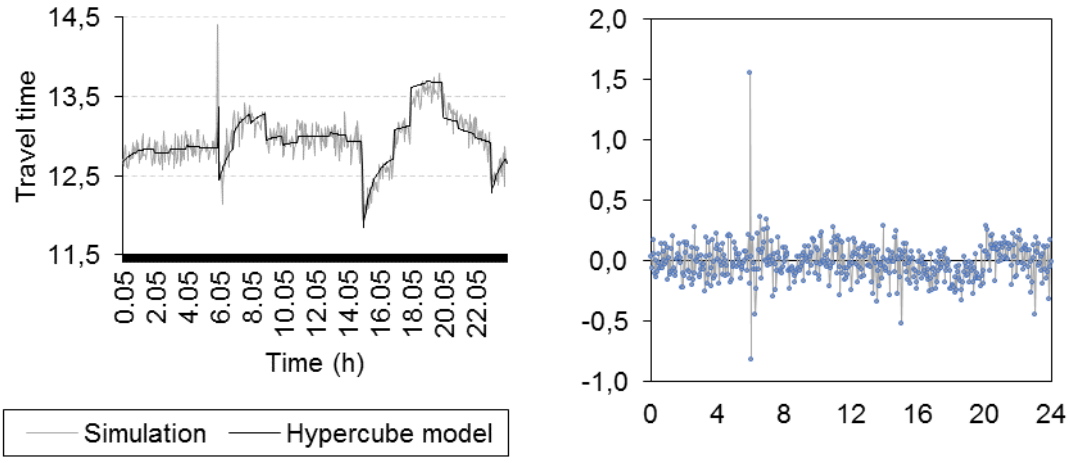
Figure 7 explores the residuals of the average waiting times. Residuals were calculated as the differences between the pairs of the simulation and the non-stationary hypercube model. Residuals are smaller during the first shift, with shorter average waiting times, whereas in the other shifts the residuals have increased dispersion, moving away from the zero. Figure 7b shows a histogram of the residuals that, although centered close to zero, with an average residual of around 10-3, has an elongated tail to the left.

Figure 7 - Residuals (a) and histogram (b) of the average waiting time residuals between the simulation and the non-stationary hypercube model.



System-Wide travel times change over time when dispatching fixed-base servers to serve requests with time-varying arrival rates (Figure 8). In Figure 8a, it is possible to see that, although both models present similar behaviors, around 06:00 hrs., there is a disturbance affecting both models. However, as shown in Figure 8b, the simulation was more sensitive, with a deviation of more than 1.5 minutes in relation to the hypercube model. The operation of server 3 is the cause of such disturbance. It has dispatch priority for atom W, even with an average travel time of 20 minutes. Observe that this disturbance spread to the next few instants, since immediately thereafter there is a negative deviation of almost 1 minute. Finally, during the second shift (15:00-23:00 hrs.) the residuals show a decreasing trend until 20:00 hrs. (same period in which the average waiting times are strongly increasing). This relationship helps to explain the cause of the deviations, since when calculating dispatch frequencies for delayed users, only the instantaneous arrival rate is considered. This was a necessary approximation since the non-stationary hypercube model is memoryless. Thus, it is not possible to predict, with any certainty, the number of users from each atom in the queue states.

Figure 8 - Average travel times (a) and residuals (b) of average travel times between simulation and non-stationary hypercube model.



Service times were calibrated according to assumption (ix) for the hypercube model. The simulation also has time-varying service times, depending on the travel times to the requests. In Figure 9A, the noise found in the simulation becomes clear, because of a maximum amplitude (difference between the largest and smallest element) of 0.6 calls/hour

for server 2, whereas the hypercube model had a maximum amplitude of 0.1 calls/hour for the same server. As such, there was little difference in the service times in the hypercube model. Figure 9B shows the histograms of the residuals. The average of the absolute residuals was in the order of 0.05 and the standard deviations in the order of 0.04. This gives a variance coefficient of close to 1, which is expected for processes with exponentially distributed times. However, the histograms of the residuals do not have regular and symmetrical formats, although they are concentrated around 0. Finally, in particular, the service rate of server 2 at 06:00 hrs. had a disturbance (as average travel times had), and the simulation was more sensitive, with a difference of 0.4 calls/hour. This result can be seen in both Figure 9a and the outlier found in the respective histogram in Figure 9B.

Figure 9 - Service rates (a) and residual histogram (b) for all illustrative example servers.

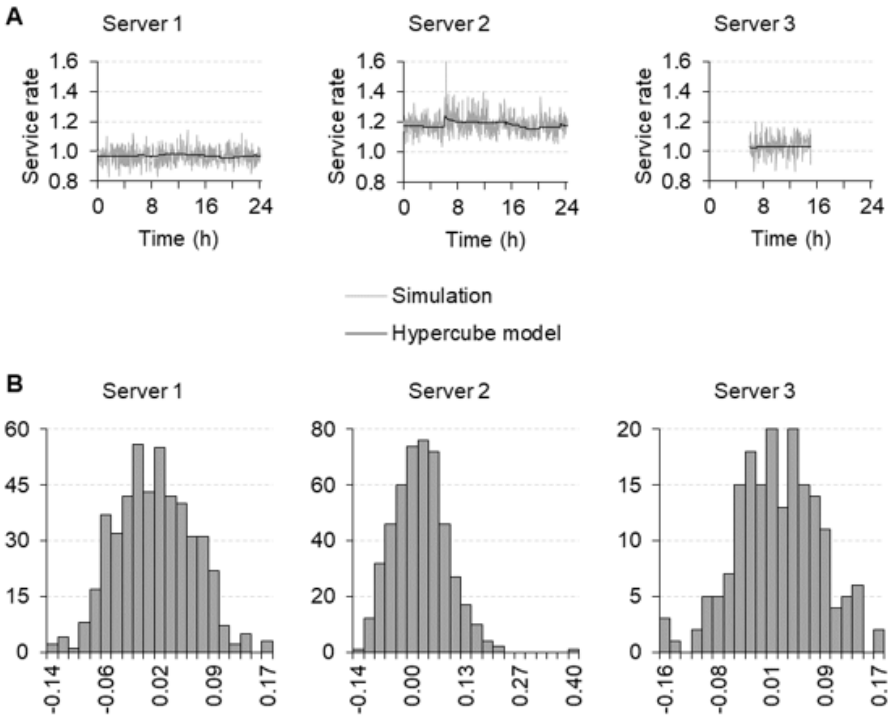
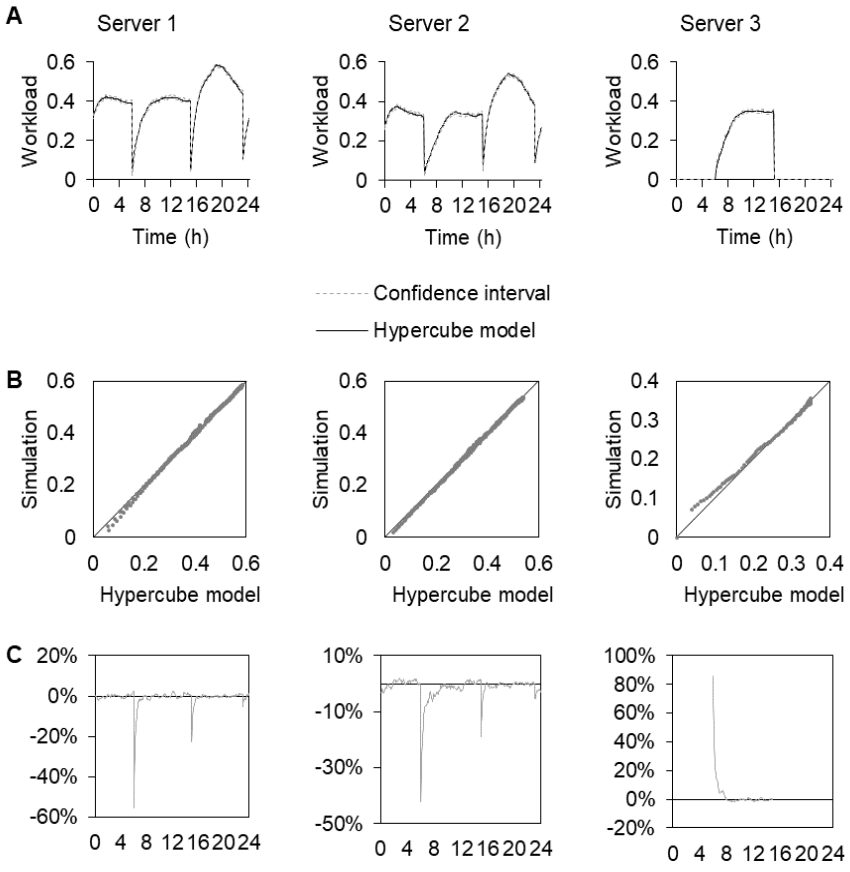


Figure 10A shows the workloads for the illustrative example and the confidence interval for the simulation workload. Firstly, one can observe decreases in workload at end-of-shifts, an effect of replacing all server and the non-preemptive end-of-shift discipline. It is

also important to note that the workload is a continuous, instantaneous measurement (differently from the others so far). For this reason, noises previously present in the other measurements do not appear in Figure 10A. In Figure 10B, it is possible to observe that although both models present quite similar measures for workloads, they present a deviation for low workloads, usually less than 0.1 call/hour. Through Figure 10C, it is possible to observe the deviations on shift changes, at which time there are peaks in the relative residuals for all servers. However, it is important to mention that after these peaks, the values tend back towards zero. This effect can have two meanings. Firstly, the continuous-time Markov chain serves as a good way to represent system behavior over time. Secondly, the discrete-time Markov chain for shift handovers needs to be improved since, while not compromising the measurement comparison itself, it generates significant residuals in relation to the simulation.

Figure 10 - Workloads (a), correlations (b), and relative residuals (c) for all servers in the illustrative example.



6. CONCLUSION

Throughout this paper, we presented a tool capable of unifying location and scheduling problems. The presented non-stationary hypercube model combines the characteristics sought by models for both mentioned problems: the time variations of a non-stationary model and the geographic characteristics of a location problem. The model was presented using an illustrative example, which was also implemented in a discrete-event simulation model. We used the simulation to assess the ability of non-stationary hypercube model to represent the same system with less noise and faster computation. However, in shift handovers, when using a discrete-time Markov chain, the deviations were more pronounced. While such deviations have not mischaracterized any performance measures, it is appropriate to foster further studies to improve the way shift handovers are represented. In addition, it is important to remember that such deviations have been reduced over time with the use of continuous-time Markov chains.

This has shown that location and scheduling problems can be addressed jointly, and no longer individually for ESSs. It is not claimed that stationary hypercube models are completely invalid. However, applications of the hypercube model should not only focus on arrival processes, but also shift handovers, even if the number of operating servers does not change.

It is also understood that future applications can be realized from approximations to the exact model shown here. Such approximations can be assessed, as seen in Ingolfsson *et al*. (2007), from the point of view of non-stationary models. They can also be developed from simplifications of the hypercube model, such as the approximate models presented by Larson (1975) and Jarvis (1985). Recalling that the proposed model will serve as benchmarking for the proposed approaches.

Finally, further research may address the deployment of the non-stationary hypercube model in some heuristics that formally unifies scheduling and location problems. In case computational times are still prohibitive, one can assume constant service times, for example. Another option would be to use forms of server aggregation, as proposed in Boyaci and Geroliminis (2015) to reduce the state space of the system. It is also understood that some performance measures can be improved, such as calculating the average wait time, which is quite costly due to numerical integration. One suggestion is to work in the same

manner as presented in Green and Soares (2007) for this calculation. Another measure that can be improved is the dispatch frequency, which, for delayed users, suffers with the memoryless property of hypercube models. A possible solution to this problem is to use a representation for the state space that is similar to that shown in Rodrigues *et al.* (2017). It is also understood that applications do not need to be limited to the classic hypercube model, so relaxing your simplifying assumptions according to the reality of the systems studied is critical. Finally, improve modeling for mealtime breaks is suggested, even for M(t)/M/s(t) models that consider non-preemptive end-of-shift disciplines. Perhaps the solution to this last suggestion lies in non-Markovian models.

**** This article was invited to be published in Gepros.***

References

ANSARI, S., MCLAY, L. A.; MAYORGA, M. E. A Maximum Expected Covering Problem for District Design. **Transportation Science**, v. 51, n. 1p. 376-390, 2017.

ATKINSON, J. B., KOVALENKO, I. N., KUZNETSOV, N.; MYKHALEVYCH, K. V. A hypercube queueing loss model with customer-dependent service rates. **European Journal of Operational Research**, v.191, p. 223-239, 2008.

BOYACI, B.; GEROLIMINIS, N. Approximation methods for large-scale spatial queueing systems. **Transportation Research** Part B, v.74, p. 151-181, 2015.

BROWN, L., GANS, N., MANDELBAUM, A., SAKOV, A., SHEN, H., ZELTYN, S.; ZHAO, L. Statistical Analysis of a Telephone Call Center: A Queueing-Science Perspective. **Journal of the American Statistical Association**, v.100, n. 469, p. 36-50, 2005.

BURWELL, T. H., JARVIS, J. P.; MCKNEW, M. A. Modeling co-located servers and dispatch ties in the hypercube model. **Computers & Operations Research**, v. 20, n. 2, p. 113-119, 1993.

DEFRAEYE, M.; VAN NIEUWENHUYSE, I. Staffing and Scheduling under nonstationary demand for service: A literature review. **Omega**, v. 58, p. 4-25, 2016.

GALVÃO, R. D.; REINALDO, M. Emergency service systems: The use of hypercube queueing model in the solution of probabilistic location problems. **International Transactions in Operational Research**, v.15, p. 522-549, 2008.

GANS, N., KOOLE, G.; MANDELBAUM, A. Telephone Call Centers: Tutorial, Review, and Research Propects. **Manufacturing & Service Operations Management**, v.5, n.2, p. 79-141, 2003.

GEROLIMINIS, N., KEPAPTSOGLOU, K..; KARLAFTIS, M. G. A hybrid hypercube - Genetic algorithm approach for deploying many emergency response mobile units in an urban network. **European Journal of Operational Research**, v. 210, p. 287-300, 2011.

GILLARD, J.; KNIGHT, V. Using Singular Spectrum Analysis to obtain staffing level requirements in emergency units. **Journal of the Operational Research Society**, v. 65, p. 735-746, 2014.

GREEN, L. V.; KOLESAR, P. J. On the Accuracy of the Simple Peak Hour Approximation for Markovian Queues. **Management Science**, v.41, n. 8, p. 1353-1370, 1995.

GREEN, L. V.; SOARES, J. Note-Computing Time-Dependent Waiting Time Probabilities in M(t)/M/s(t) Queuing Systems. **Manufacturing & Service Operations Management**, v. 9, n. 1, p. 54-61, 2007.

GREEN, L. V., KOLESAR, P. J.; SOARES, J. Improving the Sipp Approach for Staffing Service Systems That Have Cyclic Demands. **Operations Research**, v. 49, n. 4, p. 549-564, 2001.

IANNONI, A. P., CHIYOSHI, F.; MORABITO, R. A spatially distributed queuing model considering dispatching policies with server reservation. **Transportation Research Part E**, v.75, p. 49-66, 2015.

INGOLFSSON, A. (2005). **Modeling the M(t)/M/s(t) Queue with Exhaustive Discipline.** Disponível em: http://www.bus.ualberta.ca/aingolfsson/working_papers.htm. Acesso: 01 nov. 2021.

INGOLFSSON, A., AKHMETSHINA, E., BUDGE, S., LI, Y.; WU, X. A Suvey and Experimental Comparison of Service-Level-Approximation Methods for Nonstationary M(t)/M/s(t) Queueing Systems with Exhaustive Discipline. INFORMS **Journal on Computing**, v.19, n. 2, p. 201-214, 2007.

INGOLFSSON, A., CAMPELLO, F., WU, X.; CABRAL, E. Combining integer programming and the randomization method to schedule employees. **European Journal of Operational Research**, v. 202, p. 153-163, 2010.

INGOLFSSON, A., HAQUE, A.; UMNIKOV, A. Accounting for time-varying queueing effects in workforce scheduling. **European Journal of Operational Research**, v. 139, p. 585-597, 2002.

JARVIS, J. P. Approximating the Equilibrium Behavior of Multi-Server Loss Systems. **Management Science**, v.31, n. 2, p. 235-239, 1985.

KIM, S.-H.; WHITT, W. Are Call Centers and Hospital Arrivals Well Modeled by Nonhomogeneous Poisson Processes. **Manufacturing & Service Operations Management**, v.16, n. 3, p. 464-480, 2014.

LARSON, R. A Hypercube Queueing Model for Facility Location and Redistricting in Urban Emergency Services. **Computers & Operations Research**, v.1, p. 67-95, 1974.

LARSON, R. Approximating the Performance of Urban Emergency Service Systems. **Operations Research**, v. 23, n. 5, p. 845-868. 1975.

MANDELBAUM, A.; ZELTYN, S. Staffing Many-Server Queues with Impatient Customers: Constraint Satisfaction in Call Centers. **Operations Research**, v. 57, n. 5, p. 1189-1205, 2009.

MARIANOV, V.; REVELLE, C. The Queueing Maximal Availability Location Problem: A model for the siting of emergency vehicles. **European Journal of Operational Research**, v. 93, p. 110-120, 1996.

OWEN, S. H.; DASKIN, M. S. Strategic facility location: A review. **European Journal of Operational Research**, v. 111, p. 423-447, 1998.

PATRICK, J., PUTERMAN, M. L.; QUEYRANNE, M. Dynamic Multipriority Patient Scheduling for a Diagnostic Resource. **Operations Research**, v.56, n.6, p. 1507-1525, 2008.

RAJAGOPALAN, H. K., SAYDAM, C.; XIAO, J. A multiperiod set covering location model for dynamic redeployment of ambulances. **Computers & Operations Research**, v. 35, p. 814-826, 2008.

RODRIGUES, L. F., MORABITO, R., CHIYOSHI, F., IANNONI, A. P.; SAYDAM, C. Towards hypercube queuing models for dispatch policies with priority in queue and partial backup. **Computers & Operations Research**, v. 84, p. 92-105, 2017.

SCHWARZ, J. A.; SELINKA, G.; STOLLETZ, R. Performance analysis of time-dependent queueing systems: Survey and classification. **Omega,** v. 63, p. 170-189, 2016.

SOUZA, R., MORABITO, R., CHIYOSHI, F.; IANNONI, A. Incorporating priorities for waiting customers in the hypercube queuing model with application to an emergency medical service system in Brazil. **European Journal of Operational Research**, v. 242, p. 274-285, 2015.

TAHA, H. A. (2008). **Operations Research**: An Introduction (8th ed.). Prentice Hall.

TAKEDA, R. A., WIDMER, J. A.; MORABITO, R. Analysis of ambulance descentralization in an urban emergency medical service using the hypercube queueing model. **Computers & Operations Research**, v. 34, p. 727-741, 2007.