A game theoretic model of the behavioural gaming that takes place at the EMS - ED interface

Michalis Panayides, Paul Harper, Vince Knight

March 11, 2021

Abstract

The main focus of this research is the construction of a 3-player game theoretic model between two queueing systems and a service that distributes individuals to them. The resultant model will then be used to explore dynamics between all players.

The first aspect of this work is the development of a queueing system with two consecutive waiting spaces. The strategic managerial behaviour corresponds to how individuals use these waiting spaces. Two modelling techniques were used: discrete event simulation and Markov chains. The state probabilities of the Markov chain system have been used to extract the performance measures of the queueing model (e.g. mean time in each waiting room, mean number of individuals in each room, etc.).

A 3-player game theoretic model is proposed between the two queueing systems and the service that distributes individuals to them. In particular this can be seen as a 2-player normal-form game where the utilities are determined by a third player with its own strategies and objectives. A backwards induction technique is used to get the utilities of the normal-form game between the two queueing systems.

This particular system can be applied in a healthcare scenario where it captures the emergent behaviour between, for example, the Emergency Medical Service (EMS) and the Emergency Department (ED). This will be used to investigate the impact of target measures on patient welfare.

Contents

1 Introduction - Motivation

2 Overview of game theoretic model

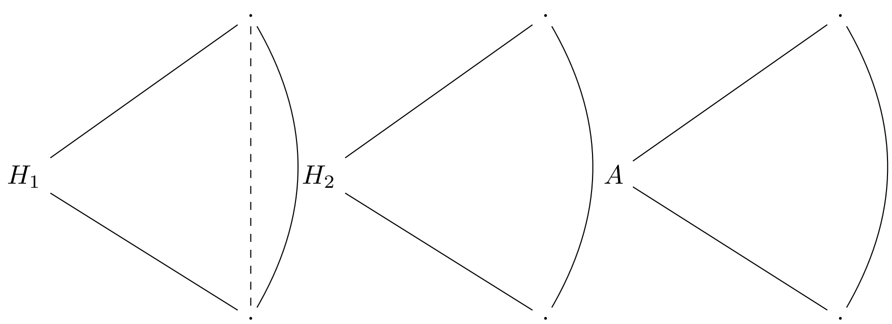

Figure 1: Ambulance Decision Problem

States:

1. A = Ambulance
2. H_i = Hospital i

Notation:

- Λ = total number of patients that need to be hospitalised
- p_i = proportion of patients going to Hospital i ($p_i\Lambda$ = number of patients going to hospital i)
- $\hat{c}_i$ = capacity of hospital i
- $W(c, \lambda\, \mu)$ = waiting time in the system function
- μ_i = service rate of hospital i
- λ_i^o = arrival rate of other patients to the hospital (not by ambulance)
- $C_i(p_i) = d_i + W(c = \hat{c}_i,\ \lambda = p_i\Lambda + \lambda_i^o,\ \mu = \mu_i)$

Players:

- Ambulance
- Hospital A
- Hospital B

Strategies of players:

- Hospital i:
  1. Close doors at $\hat{c}_i = 1$
  2. Close doors at $\hat{c}_i = 2$

3. ...
4. Close doors at $\hat{c}_i = C_i$

- Ambulance:

1. Choose $p_1 \in [0, 1]$

Cost Function: Probability of waiting time less than target.

3 A queueing model with 2 consecutive buffer centres

3.1 System

The following Markov chain represents the transition between states of a service centre while capturing the interactions between it and a buffer centre. The service centre accepts two types of individuals; Class 1 and Class 2. Class 2 individuals are accepted until a pre-determined threshold T of individuals is reached. When reached, all Class 2 individuals that arrive will remain *"blocked"* in the buffer centre until the number of people in the system is reduced below T. Additionally, if the people in the service centre keep rising, they may exceed the number of servers C available, which will in turn mean that every new person will have to wait for a server to become free. The states of the Markov chain are denoted by (u, v) where:

- u = number of Class 2 individuals blocked
- v = number of Class 1 individuals in the service centre

3.1.1 Markov-chain state mapping function

The transition matrix of the Markov-chain representation described above can be denoted by a state mapping function. The state space of this function is defined as:

$$\begin{aligned} S(T) =& S_1(T) \cup S_2(T) \text{ where:} \\ S_1(T) =& \left\{(0, v) \in \mathbb{N}_0^2 \mid v < T\right\} \\ S_2(T) =& \{(u, v) \in \mathbb{N}_0^2 \mid v \geq T\} \end{aligned} \tag{1}$$

Therefore, the entries of the transition matrix Q, can be given by $q_{i,j} = q_{(u_i,v_i),(u_j,v_j)}$ which is the transition rate from state $i = (u_i, v_i)$ to state $j = (u_j, v_j)$ for all $(u_i, v_i), (u_j, v_j) \in S$.

$$q_{i,j} = \begin{cases} \Lambda, & \textbf{if } (u_i, v_i) - (u_j, v_j) = (0, -1) \textbf{ and } v_i < \text{t} \\ \lambda_1, & \textbf{if } (u_i, v_i) - (u_j, v_j) = (0, -1) \textbf{ and } v_i \geq \text{t} \\ \lambda_2, & \textbf{if } (u_i, v_i) - (u_j, v_j) = (-1, 0) \\ v_i\mu, & \textbf{if } (u_i, v_i) - (u_j, v_j) = (0, 1) \textbf{ and } v_i \leq C \textbf{ or} \\ & \quad (u_i, v_i) - (u_j, v_j) = (1, 0) \textbf{ and } v_i = T \leq C \\ C\mu, & \textbf{if } (u_i, v_i) - (u_j, v_j) = (0, 1) \textbf{ and } v_i > C \textbf{ or} \\ & \quad (u_i, v_i) - (u_j, v_j) = (1, 0) \textbf{ and } v_i = T > C \\ -\sum_{j=1}^{|Q|} q_{i,j} & \textbf{if } i = j \\ 0, & \textbf{otherwise} \end{cases} \tag{2}$$

In order to acquire an exact solution of the problem a slight adjustment needs to be considered. The problem defined above assumes no upper boundary to the number of individuals that can wait for service or the ones that are blocked in the buffer centre. Therefore, a different state space $\tilde{S}$ needs to be constructed where $\tilde{S} \subseteq S$ and there is a maximum allowed number of people N that can be in the system and a maximum allowed number of people M that can be blocked in the buffer centre:

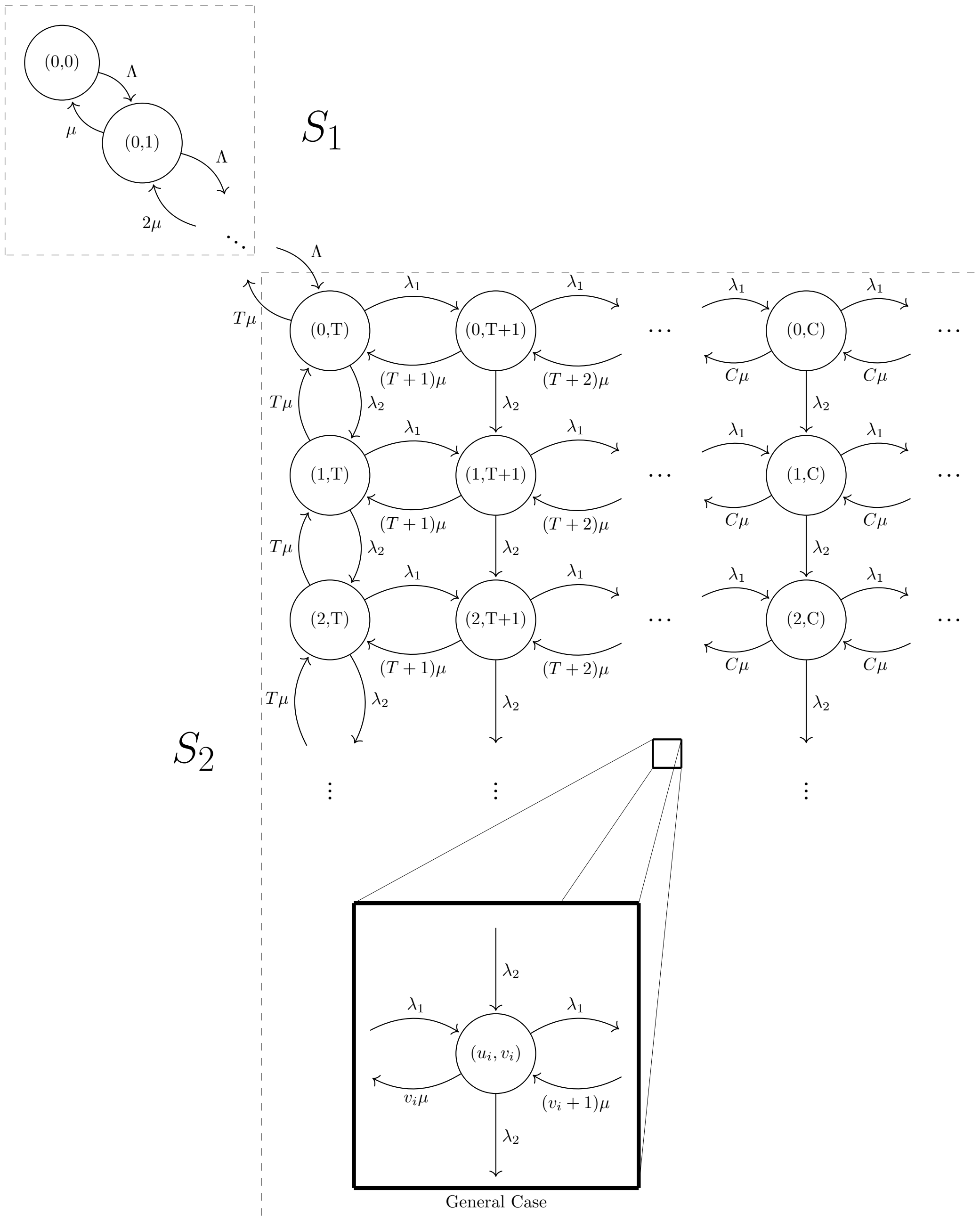

Figure 2: Markov chain

$$\tilde{S} = \{(u, v) \in S \,|u \leq M, v \leq N\} \tag{3}$$

3.1.2 Steady State

Having calculated the transition matrix Q for a given set of parameters the probability vector π needs to be considered. The vector π is commonly used to study such stochastic systems and it's main purpose is to keep track of the probability of being at any given state of the system. The term *steady state* refers to the instance of the vector π where the probabilities of being at any state become stable over time. Thus, by considering the steady state vector π the relationship between it and Q is given by:

$$\frac{d\pi}{dt} = \pi Q = 0$$

There are numerous methods that can be used to solve problems of such kind. In this paper only numeric and algebraic approaches will be considered.

The first approach to be considered is to solve the differential equation numerically by observing the behaviour of the model over time. The solution is obtained via python's SciPy library. The functions odeint and solve_ivp have been used in order to find a solution to the problem. Both of these functions can be used to solve any system of first order ODEs.

3.2 Performance Measures

One may easily derive the average number of individuals that are at any given state using *pi*. The average number of individuals in state i can be calculated by multiplying the number of individuals that are present in state i with the probability of being at that particular state (i.e $\pi_i(u_i + v_i)$). Using this logic it is possible to calculate any performance measures that are related to the mean number of individuals in the system.

Average number of people in the system:

$$L = \sum_{i=1}^{|\pi|} \pi_i (u_i + v_i) \tag{4}$$

Average number of people in the service centre:

$$L_H = \sum_{i=1}^{|\pi|} \pi_i v_i \tag{5}$$

Average number of people in the buffer centre:

$$L_A = \sum_{i=1}^{|\pi|} \pi_i u_i \tag{6}$$

Consequently getting the performance measures that are related to the duration of time is not as straightforward. Such performance measures are the mean waiting time in the system and the mean time blocked in the system. Under the scope of this study three approaches have been considered to calculate these performance measures; a direct approach, a recursive algorithm and consequently a closed-form formula.

The research question that needs to be answered here is: "When a class 1/2 individuals enters the system, what is the expected time that they will have to wait?". In order to formulate the answer to that question one needs to consider all possible scenarios of what state the system can be in when an individual arrives. Furthermore, different formulas arises for class 1 individuals and a different one for class 2 individuals.

3.2.1 Mean waiting time

Upon closer inspection of the recursive formula a more compact formula can arise. The equivalent closed-form formula eliminates the need for recursion and thus makes the computation of waiting times much more efficient. Just like in the recursive part there are two formulas; one for *class 1* and one for class 2 individuals. The formulas are given by:

$$W^{(1)} = \frac{\sum_{\substack{(u,v) \in S_A^{(1)} \\ v \geq C}} \frac{1}{C\mu} \times (v - C + 1) \times \pi(u,v)}{\sum_{(u,v) \in S_A^{(1)}} \pi(u,v)} \tag{7}$$

$$W^{(2)} = \frac{\sum_{\substack{(u,v) \in S_A^{(2)} \\ min(v,T) \geq C}} \frac{1}{C\mu} \times (\min(v+1,T) - C) \times \pi(u,v)}{\sum_{(u,v) \in S_A^{(2)}} \pi(u,v)} \tag{8}$$

Note here that the summation, in both equations 7 and 8, goes through all states in the set of accepting states of either class 1 or class 2 individuals respectively, where a wait incurs. In equation 7 that includes all states (u, v) in the set of accepting states of class 1 individuals such that $v \geq C$; i.e. whenever an arrival occurs and the system is at a state where the number of individuals in the system is more than or equal to C. Consequently, for the states that are included in the summation the expression $v - C + 1$ indicates the amount of people in service one would have to wait for upon arrival at the hospital.

Additionally, the minimisation function in equation 8 ensures that when a class 2 individual arrives at any state that is greater than the predetermined threshold, the wait that the individual will have to endure remains the same. In essence, the expression $\min(v + 1, T) - C$ returns the number of people in line in front of a particular individual upon arrival.

3.2.2 Overall Waiting Time

Consequently, the overall waiting time should can be estimated by a linear combination of the waiting times of class 1 and class 2 individuals. The overall waiting time can be then given by the following equation where c_1 and c_2 are the coefficients of each individual's type waiting time:

$$W = c_1 W^{(1)} + c_2 W^{(2)} \tag{9}$$

The two coefficients represent the proportion of individuals of each type that traversed through the model. Theoretically, getting these percentages should be as simple as looking at the arrival rates of each type but in practise if the service centre or the buffer centre is full, some individuals may be lost to the system. Thus, one should account for the probability that an individual is lost to the system. This probability can be easily calculated by using the two sets of accepting states

$S_A^{(2)}$ and $S_A^{(1)}$ defined earlier in equations. Let us define here the probability, for either class type, that an individual is not lost in the system by:

$$P(L_1') = \sum_{(u,v) \in S_A^{(1)}} \pi(u,v) \qquad\qquad P(L_2') = \sum_{(u,v) \in S_A^{(2)}} \pi(u,v)$$

Having defined these probabilities one may combine them with the arrival rates of each class type in such a way to get the expected proportions of class 1 and class 2 individuals in the model. Thus, by using these values as the coefficient of equation 9 the resultant equation can be used to get the overall waiting time. Note here that the equation below gets the overall waiting time for both the recursive and the closed-form formula.

$$W = \frac{\lambda_1 P(L_1')}{\lambda_2 P(L_2') + \lambda_1 P(L_1')} W^{(1)} + \frac{\lambda_2 P(L_2')}{\lambda_2 P(L_2') + \lambda_1 P(L_1')} W^{(2)} \tag{10}$$

3.2.3 Mean blocking time

Unlike the waiting time, the blocking time is only calculated for class 2 individuals. That is because class 1 individuals cannot be blocked. Thus, one only needs to consider the pathway of class 2 individuals to get the mean blocking time of the system. Blocking occurs at states (u, v) where $u > 0$. Thus, the set of blocking states can be defined as:

$$S_b = \{(u,v) \in S \mid u > 0\}$$

In order to not consider individuals that will be lost to the system, the set of accepting states needs to be taken into account. The set of accepting states is given by:

$$S_A^{(2)} = \begin{cases} \{(u,v) \in S \mid u < M\} & \textbf{if } T \leq N \\ \{(u,v) \in S \mid v < N\} & \textbf{otherwise} \end{cases}$$

For the waiting time formula, the mean sojourn time for each state was considered, ignoring any arrivals. Here, the same approach is used but ignoring only class 2 arrivals. That is because for the waiting time formula, once an individual enters the service centre (i.e. starts waiting) any individual arriving after them will not affect their pathway. That is not the case for blocking time. When a class 2 individual is blocked, any class 1 individual that arrives will cause the blocked individual to remain blocked for more time. Therefore, class 1 arrivals are considered here:

$$c(u,v) = \begin{cases} \frac{1}{\min(v,C)\mu}, & \text{if } v = C \\ \frac{1}{\min(v,C)\mu + \lambda_1}, & \text{otherwise} \end{cases} \tag{11}$$

In equation 11, both service completions and class 1 arrivals are considered. Thus, from a blocked individual's perspective whenever the system moves from one state (u, v) to another state it can either:

- be because of a service being completed: we will denote the probability of this happening by $p_s(u, v)$.
- be because of an arrival of an individual of class 1: denoting such probability by $p_o(u, v)$.

The probabilities are given by:

$$p_s(u,v) = \frac{\min(v,C)\mu}{\lambda_1 + \min(v,C)\mu}, \qquad p_o(u,v) = \frac{\lambda_1}{\lambda_1 + \min(v,C)\mu}$$

Having defined $c(u,v)$ and S_b a formula for the blocking time that is expected to occur at each state can be given by:

$$b(u,v) = \begin{cases} 0, & \textbf{if } (u,v) \notin S_b \\ c(u,v) + b(u-1,v), & \textbf{if } v = N = T \\ c(u,v) + b(u,v-1), & \textbf{if } v = N \neq T \\ c(u,v) + p_s(u,v)b(u-1,v) + p_o(u,v)b(u,v+1), & \textbf{if } u > 0 \textbf{ and } v = T \\ c(u,v) + p_s(u,v)b(u,v-1) + p_o(u,v)b(u,v+1), & \textbf{otherwise} \end{cases} \tag{12}$$

Equation (12) will not be solved recursively. A direct approach will be used to solve this equation here. By enumerating all equations of (12) for all states (u,v) that belong in S_b a system of linear equations arises where the unknown variables are all the $b(u,v)$ terms. For instance, let us consider a Markov model where $C = 2, T = 3, N = 6, M = 2$. The Markov model is shown in Figure 3 and the equivalent equations are (13)-(18). The equations considered here are only the ones that correspond to the blocking states.

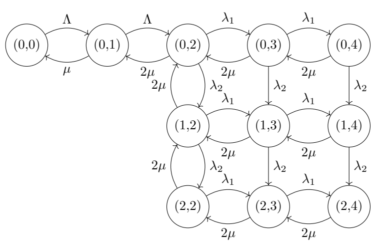

Figure 3: Example of Markov chain

$$b(1,2) = c(1,2) + p_o b(1,3) \tag{13}$$

$$b(1,3) = c(1,3) + p_s b(1,2) + p_o b(1,4) \tag{14}$$

$$b(1,4) = c(1,4) + b(1,3) \tag{15}$$

$$b(2,2) = c(2,2) + p_s b(1,2) + p_o b(2,3) \tag{16}$$

$$b(2,3) = c(2,3) + p_s b(2,2) + p_o b(1,4) \tag{17}$$

$$b(2,4) = c(2,4) + b(2,3) \tag{18}$$

Additionally, the above equations can be transformed into a linear system of the form $Zx = y$ where:

$$Z = \begin{pmatrix} -1 & p_o & 0 & 0 & 0 & 0 \\ p_s & -1 & p_o & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ p_s & 0 & 0 & -1 & p_o & 0 \\ 0 & 0 & 0 & p_s & -1 & p_o \\ 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}, x = \begin{pmatrix} b(1,2) \\ b(1,3) \\ b(1,4) \\ b(2,2) \\ b(2,3) \\ b(2,4) \end{pmatrix}, y = \begin{pmatrix} -c(1,2) \\ -c(1,3) \\ -c(1,4) \\ -c(2,2) \\ -c(2,3) \\ -c(2,4) \end{pmatrix} \tag{19}$$

A more generalised form of the equations in (19) can thus be given for any value of C, T, N, M by:

$$\begin{aligned} b(1,T) =& c(1,T) + p_o b(1,T+1) && (20) \\ b(1,T+1) =& c(1,T+1) + p_s(1,T) + p_o b(1,T+1) && (21) \\ b(1,T+2) =& c(1,T+2) + p_s(1,T+1) + p_o b(1,T+3) && (22) \\ &\vdots \\ b(1,N) =& c(1,N) + b(1,N-1) && (23) \\ b(2,T) =& c(2,T) + p_s b(1,T) + p_o b(2,T+1) && (24) \\ b(2,T+1) =& c(2,T+1) + p_s b(2,T) + p_o b(2,T+2) && (25) \\ &\vdots \\ b(M,T) =& c(M,T) + b(M,T-1) && (26) \end{aligned}$$

The equivalent matrix form of the linear system of equations (20) - (26) is given by $Zx = y$, where:

$$Z = \begin{pmatrix} -1 & p_o & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ p_s & -1 & p_o & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & p_s & -1 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & -1 & 0 & 0 & 0 & \dots & 0 & 0 \\ p_s & 0 & 0 & \dots & 0 & 0 & -1 & p_o & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & p_s & -1 & p_o & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 1 & -1 \end{pmatrix}, x = \begin{pmatrix} b(1,T) \\ b(1,T+1) \\ b(1,T+2) \\ \vdots \\ b(1,N) \\ b(2,T) \\ b(2,T+1) \\ \vdots \\ b(M,T) \end{pmatrix}, y = \begin{pmatrix} -c(1,T) \\ -c(1,T+1) \\ -c(1,T+2) \\ \vdots \\ -c(1,N) \\ -c(2,T) \\ -c(2,T+1) \\ \vdots \\ -c(M,T) \end{pmatrix} \tag{27}$$

Thus, having calculated the mean blocking time for all blocking states $b(u,v)$, it only remains to put them together in a formula. The resultant blocking time formula is given by:

$$B = \frac{\sum_{(u,v)\in S_A} \pi_{(u,v)}\ b(u,v)}{\sum_{(u,v)\in S_A} \pi_{(u,v)}} \tag{28}$$

3.3 Example

4 Behavioural Methodology

4.1 Backwards Induction

4.2 Nash Equilibrium

4.3 Learning Algorithms

5 EMS-ED application

5.1 Application

5.2 Data analysis of generated problem

6 Conclusion