Capacity Planning of Prisons in the Netherlands

Ronald Korporaal
Rommert Dekker[1]
Econometrisch Instituut
Erasmus Universiteit
Rotterdam

Ad Ridder[2]
Dept. of Econometrics
Vrije Universiteit
Amsterdam

Peter Kloprogge[3]
Point Logic Systems
Rotterdam

Econometric Institute Report EI-9909/A

Abstract

In this paper we describe a decision support system developed to help in assessing the need for various type of prison cells. In particular we predict the probability that a criminal has to be sent home because of a shortage of cells. The problem is modelled through a queueing network with blocking after service. We focus in particular on the new analytical method to solve this network.

[1]Corresponding author: Econometrisch Instituut, Erasmus Universiteit Rotterdam, PO Box 1738, 3000 DR Rotterdam, the Netherlands. Email: rdekker@few.eur.nl
[2]Dept. of Econometrics, Vrije Universiteit Amsterdam, Email: aridder@econ.vu.nl
[3]Point Logic Systems, Stationsplein 45, 3013 AK Rotterdam

1 Introduction

Crime puts society on high costs. However, confining criminals in prisons, costs the society a lot as well! Calculations executed by the Dutch Ministry of Justice[3] indicate that the building of a standard prison cell requires about Hfl 275.000 (125.000 Euro) while keeping someone enprisoned about Hfl. 80.000 (36.000 Euro) per year. Accordingly, it is quite important to make a proper estimate for the need of prison cells.

According to Dutch law, no two severely punished criminals are put in one cell. If no cell is available, then offenders of small crimes are sent home where they wait for their sentence. They have to report at a later time to the authorities at so-called self-reporting institutions. Due to this shortage of cells the total number of home-sendings in 1994 amounted to more than 5000. At that time there were about 8700 detainees. The huge number of home-sendings caused quite some negative publicity. However, the problem is not that easy since the demand for cells varies from day to day and there are several types of prisons.

In this paper we describe a decision support system which has been build for the Dutch Penitentiary Agency to support them in their assessments of the need for prison cells. In this system the problem is modelled as a queueing network with blocking after service. We especially focus on the new analytical method developed to solve this queueing network, because this type of blocking is new. There does exist some literature on operational research applied to crime and justice (see the overview given in Maltz[4]) but only very few papers describe the assessment of the need for prison cells. Cuvelier[5] presents a simulation model for use in the United States but does not model the possibility of sending criminals home because of cell shortage, which is the essential aspect in the Dutch case. Although the Dutch situation may be very specific, we think the problem is interesting both from a methodological and a problem oriented point of view.

The structure of the paper is as follows. In the next section we describe the penitentiary system in the Netherlands. Then we describe the decision support system. In sections 4 and 5 we give the details of the queueing model and the solution procedure. Finally in section 6, we present the results of the case study.

2 The Penitentiary System

The penitentiary system in the Netherlands consists of several prisons or institutions of different types. These prisons may be viewed as connected to each other and to society through flows of prisoners.

A person who is arrested on the suspicion of having committed a crime is first put into a police cell. There he (we will use the male form as most, but not all, criminals have that sex) is allowed to stay for at most 72 hours. In some cases this may be extended once with again 72 hours. Thereafter he has to be transferred to a remand centre where he waits for his trial. If no cell in such an institution is available, or if his crime is not serious enough, he is sent home. This is called a *home-sending*. A suspect who is sent home, is adjudged later, and when he gets sentenced he reports at a self-reporting institution. He will spend the first term of his sentence in such a prison. If he does not report, he is

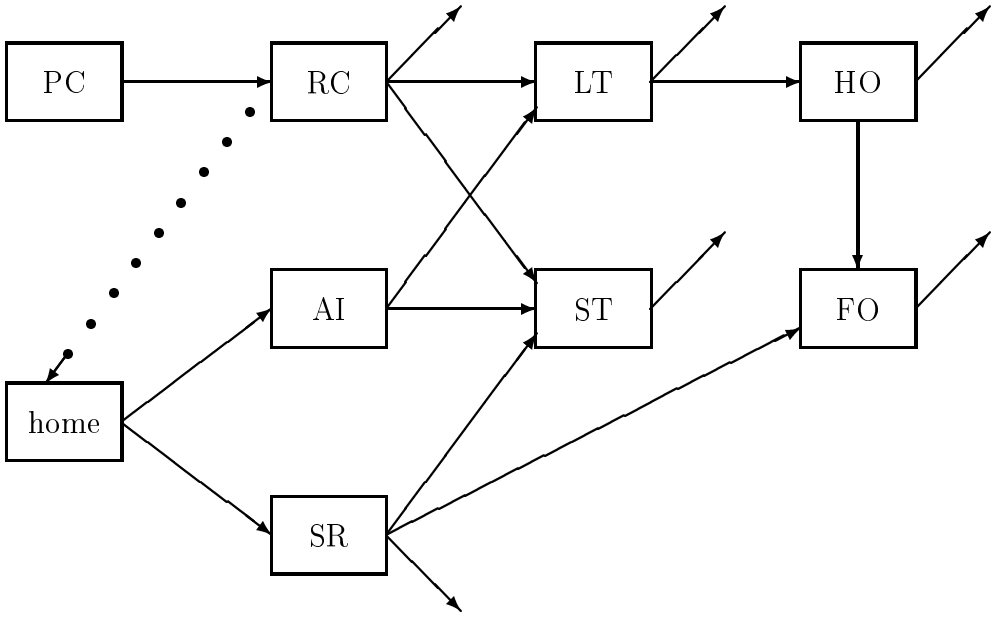

Figure 1: The prison types and the (most common) prisoner flows in the Dutch penitentiary system. PC: police cell; RC: remand centres; AI: institution of arrested persons; SR: self-reporting institution; ST: short-term institution; LT: long-term institution; HO: half-open institution; FO: fully-open institution.

arrested again and brought to an institution for arrested persons where he will spend the first term of his sentence.

The other suspects wait in the remand centres for their trial. After trial and sentence the criminal is sent to a short-term or a long-term detention prison, depending on the length of his sentence. If he behaves well, he may go after some time to either a half or a fully open detention. For very short sentences the criminals go to the self-reporting institutions mentioned before. Finally, there are specially protected prisons and prisons for women and youngsters. As the latter groups are rather small we leave them in this paper out of consideration. Figure 1 illustrates the possible flows and transfers of detainees in this described penitentiary system. The dotted line is the flow of suspects who are sent home because of cell shortage. The arrows pointing outside represent the return to society because of completion of detention.

The moments of transfer between institutions are laid down in official rules. Yet if no place in the next institution is available the prisoner stays in his present cell. We say that he waits for transfer, and we call him a *waiting person*, and we call the time lag his *waiting time*. It may even happen that he skips an institution, e.g. go over from a prison for long sentences to directly an open institution, instead of staying for some time in a half-open institution.

3 The Decision Support System

To make optimal use of all cell capacity the Dutch Penitentiary Agency developed a nation wide information system indicating which cells in which institutions were available

at which times. The information system only collects data of all events in the prisons. The next step was to build a more advanced system which could determine the need for each type of prison cell given estimated inflow of prisoners and their verdicts, and given a required performance. The *performance* of the penitentiary system is expressed most importantly in

- the number of home-sendings,
- the waiting times of prisoners,
- and the times of occupation of the cells (as fraction).

The purpose of this system was to support the political decision to which amount to invest in building new cells. In an ideal prison system the home-sendings and waiting times are low, the times of occupation are high.

This decision support system has been developed by the consulting company Point Logic Systems jointly with people of the Econometrisch Instituut of the Erasmus University in Rotterdam at the end of 1995 and the beginning of 1996. The system consists of two major components, CellSim and CellNet. Each component is a PC-based, stand-alone program. CellSim is a simulation program of the penitentiary system, CellNet is a numerical program of an approximate model of the penitentiary system.

In order to run the simulation program CellSim, one has to specify its input: (i) the number of different prison cells, (ii) the rate of incoming detainees, (iii) the distribution of sentences, (iv) the rules which control the processes of the prisoners. The output of the simulation gives estimates of the performance of the system. The major disadvantage of the program is that it requires long runs, basically because in a realistic system the prisons are almost constantly full. Hence the so-called warming-up period (before the simulation reaches equilibrium) takes a long time. Furthermore, in the program each individual must be monitored throughout his presence in the system. In a realistic simulation study this requires monitoring some eight to nine thousand people. This causes a heavy use of memory capacity and processor capacity, because of the frequent sorting of events. Therefore, we developed CellNet, which calculates the performance of an approximate queueing model of the penitentiary system. In this model individuals are grouped into classes or types. The calculations are based on equilibrium properties of queueing models resulting in an iterative algorithm. In the following sections we describe the queueing model and the iterative algorithm in more detail. In section 6 we evaluate the two programs.

An indication of different running times might be that a typical run of the CellSim program takes two hours and that it takes about half a minute to run CellNet (on a Pentium 75 Mhz PC). The programs are used in practice in serial order. Through CellNet many scenarios of capacity extensions are considered and their performances are calculated (approximately). Those scenarios which seem to result in required performance, are simulated in CellSim in order to get more accurate answers. As an example, one of the striking results was that the scenario of extending the long-term prisons resulted in a much smaller number of waiting persons than the scenario of extending the remand centres. In section 6.3 we come back to this example.

Finally we remark that it took five months to develop the CellSim program. This was much longer than initially planned. The reason was that the program should simulate the

penitentiary system with all its complex rules as realistic as possible. (We shall report on the validation in section 6.) As a consequence, the program is somewhat rigid since any change in rules need reprogramming. Contrary, we think that CellNet is much more robust because it is based on an approximation of the system. And it is interesting to remark that the development took only three months.

4 The Mathematical Model in CellNet

We model the penitentiary system as a network of J finite capacity service facilities (*queues*). Each queue represents one of the prisons which we described earlier. The detainees form the *customers* of the network. The *servers* at the queues represent the cells of detention.

In reality each detainee follows a specific route through the penitentiary system and knows in advance his minimal duration in each of the prisons and his maximal sojourn time in the system. This may be modeled in the queueing network by introducing different customers classes each with his own route and service times[6]. However, we are interested in average long-run performances of the system, thus we observe the detainees only as a singular flow through the system[2]. In other words, in our queueing model we aggregate all customers into one class and we average over all possible routes and service times.

In this way we apply the classical modeling concepts of a queueing network[7] which we will summarize in section 4.1. In section 4.2 we will give the performance measures of interest in this network, and in section 6.1 we will test our modelling assumptions statistically against data collected in the actual system.

4.1 Queueing Network

Customers from outside the network arrive at queue j according to a Poisson process with rate γ_j, $j = 1, 2, \ldots, J$. The service times at the j-th queue form a sequence of i.i.d. random variables which are exponentially distributed with mean μ_j^{-1}. We call these service times the *scheduled* service times. All arrival and service processes are independent. After service completion at queue i, the customer routes to queue j with probability r_{ij} or leaves the network with probability r_{i0}, where

$$\sum_{j=0}^{J} r_{ij} = 1, \quad i = 1, 2, \ldots, J.$$

Each queue is a bufferless queue with finitely many servers. That means that each customer currently present at the queue has his own server to whom he has been assigned immediately upon entering the queue. Let c_j be the number of servers at the j-th queue. A blocking protocol is required which regulates the evolution of the network when all c_j servers are occupied. An external arrival at queue j finding all c_j servers occupied, is blocked and lost. Now suppose that a customer completes his service at queue i and chooses queue j to be the next queue on his route. When, at that moment, all c_j servers

are occupied, the customer is blocked and remains in queue i until a server becomes available in queue j. During his blocking time, the customer keeps holding his server of queue i. This server is blocked and not available for other customers who wish to enter queue i. As soon as a space becomes free at queue j, the blocked customer moves to it and frees his server at queue i. The time during which a customer blocks a server at queue i due to waiting for a free server at queue j is denoted as the *blocking* time at queue i and the *transfer* time to queue j. The implemented blocking protocol is known as *blocking after service*[8]. When more customers wait for entering queue j, the protocol assigns them according to the *first-blocked-first-enter* rule. In an arbitrary network configuration it is possible that a deadlock occurs. In that case the first-blocked-first-enter rule is violated because the deadlock is resolved by exchanging blocked customers[9].

There is one more specific property of the penitentiary system that we have to model. It deals with the fact that the scheduled term of imprisonment in a prison is diminished with the time duration of waiting for a transfer when the prison turns out to be full at the planned time of the transfer. As soon as a customer in our queueing model enters a free server at queue j, he subtracts his transfer time from his scheduled service time. Hence, the *effective* service time of a customer at a queue is the resultant of the scheduled service time (at queue j) minus his transfer time (to queue j) plus his blocking time (at queue j). Of course, a detainee is released as soon as his sentence has been reached. Translated in the queueing model: as soon as the total effective service times (summed over all visited queues) exceeds the total scheduled service times (summed over all queues) the customer leaves the network.

4.2 Model Performance

The queueing network given above is an example of a stochastic discrete event system. Events occur every time a new customer arrives or a service is completed. A generalized semi-Markov process (GSMP) is suited to model such a system[10]. The states of a GSMP reflect both physical states and remaining durations in these physical states (so-called *clock times*). Let $\mathbf{X} = \{X(t) : t \geq 0\}$ be such a GSMP that describes the evolution of the queueing network. A state of the process gives information on

(a) the current number of occupied and blocked servers at all queues,

(b) the remaining service time of the occupied servers (scheduled minus transfer),

(c) the destination of all blocked customers, i.e., the queue for which they wait for a server to become free,

(d) the waiting time of blocked customers,

(e) the order of waiting customers who wish to enter a full queue.

Clearly, this is a nontrivial state description of the network. However, modeled as a GSMP, one can formulate easily performance measures of the system, and one has a framework for executing a simulation of the system[8]. In our queueing network the performance

measures of interest are: (i) the mean numbers of occupied and blocked servers, (ii) the mean sojourn and blocking times, and (iii) the loss probabilities. These are denoted by (for $j = 1, 2, \ldots, J$)

- $L_j :=$ mean number of occupied (including blocked) servers at queue j,
- $B_j :=$ mean number of blocked servers at queue j,
- $S_j :=$ mean sojourn time at queue j,
- $W_j :=$ mean blocking time at queue j,
- $p_j^{\text{loss}} :=$ loss probability at queue j,
- $U_j :=$ utilisation at queue j (fraction of time that a server is busy).

The throughput of the j-th queue is the mean number of departures per unit time, denoted by Λ_j. Then by Little, the mean sojourn time and the mean number at queue j are related via

$$L_j = \Lambda_j S_j, \quad j = 1, 2, \ldots, J.$$

Since only external arriving customers are lost, the throughputs $\Lambda_j,\ j = 1, 2, \ldots, J$ satisfy the following set of linear (*traffic*) equations,

$$\Lambda_j = \gamma_j \left(1 - p_j^{\text{loss}}\right) + \sum_{i=1}^{J} \Lambda_i r_{ij}, \quad j = 1, 2, \ldots, J. \tag{1}$$

4.3 Translation to Penitentiary System

The performance measures of the mathematical queueing network are translated to the performances of interest in the penitentiary system through the following obvious relations. B_j, the mean number of blocked servers, corresponds to the number of detainees waiting for transfer in prison j, W_j, the mean blocking time, corresponds to the waiting time of a detainee in prison j; p_j^{loss}, the loss probability, corresponds to the percentage of home-sendings at prison j; U_j, the utilisation, corresponds to the time of occupation of prison j.

5 Model Solution

When we would know the stationary distribution of the states of the GSMP $\mathbf{X}$, we could calculate the the performance measures of interest. Moreover, open networks without blocking exhibit in many cases a so-called product form solution of the stationary distribution[4,5], making the calculation a lot easier. However, due to the blocking phenomena, there is no analytical expression of the stationary distribution in the model we study. Furthermore, the state space of the model is too complicated to consider a numerical approach.

In this section we propose an approximation algorithm. The idea is to approximate the model by a product form queueing network. In the following section we shall evaluate our algorithm by comparing it with simulation studies of the model.

5.1 Historical Background

Many approximate algorithms for queueing networks with blocking have been reported in the literature[11]. The general idea is to decompose the network in smaller subsystems or in individual queues which are analyzed in isolation[12]. The main effort of each method lies in finding the revised arrival rates and the revised service time distributions so that the stationary distributions of the individual queues approximate the marginal stationary distributions of the network.

For tandem configurations of queues with a single exponential server, finite capacity and blocking after service, Hillier and Boling[13] implement exponential service times of the isolated queues. The capacity of an isolated queue is augmented by one in order to accommodate the blocked customer of the upstream queue. The algorithm of Altiok[14] approximates the effective service time distributions by Coxian-2 distribution by looking only one queue ahead which may be blocked. The effective service time distributions are approximated by general Coxian distributions in the algorithm of Perros and Altiok[15] by taking into account that blocking is backlogged over a number of successive queues. In Altiok[16] the original service times are allowed to be of a general phase-type.

An alternative approach decomposes the tandem network in successive pairs of adjacent queues. This so-called two-node decomposition algorithm has been developed by Brandwajn and Jow[17] in case of exponential interarrival and service time distributions with state dependent rates.

Single-node decomposition algorithms for arbitrary network configurations of queues with a single exponential server, finite capacity, blocking after service, and first-blocked-first-enter rule, have been developed by Takahashi et. al.[18] (without deadlock detection). Again, the approximate effective service times follow an exponential distribution. Altiok and Perros[19] use phase-type distributions to approximate the effective service time distributions in feed-forward networks with either external or internal arrivals at any queue. This is generalized in Jun and Perros[7] for arbitrary network models with deadlock detection and resolution. The phase-type mechanism becomes quite complex because many different blocking configurations have to be incorporated. Lee and Pollock[20] propose an algorithm for feed-forward networks which approximates the marginal occupancy probabilities. The queues in isolation are analyzed as finite birth-death processes.

Also other approximation methods have been developed to include more general service times. The isolation method[21] is based on diffusion approximations of queues with repetitive-service blocking. The maximum entropy method[22] decomposes the network in isolated queues and maximizes constrained entropy functionals. Again, the repetitive-service blocking is assumed. The expansion method[23] is based on (i) a reconfiguration of the network by adding holding queues with appropriate routings, and (ii) parameters adjustment. The method applies to feed-forward models.

Much less work has been done on approximate algorithms of queueing networks with multiple servers and blocking. The maximum entropy method[24] applies to models with

repetitive-service blocking. The method in Han and Smith[25] applies to two-layered feed-forward networks. The effective service time distributions in the first layer are approximated by Coxian distributions. They provide approximate blocking probabilities and system throughput. Closed networks with multiple exponential servers and blocking are analysed in Akyildiz[26]. The approximate algorithm modifies the network into a nonblocking network by a state space transformation.

Networks of bufferless queues are models applied to specific production lines in manufacturing systems[27]. However, the analysis is restricted to single server models and concerned with the throughput of the system[28,29]. To our best knowledge, no algorithms have been reported on networks of bufferless multiple server queues with the blocking-after-service rule. Furthermore, a complicating factor in our model of the penitentiary system is the subtraction of the transfer time from the regular service time. Consequently, the effective service time in our model has a different interpretation and implementation than in the literature we have found so far.

5.2 An Algorithm For Tandem Configurations

In this section we describe our algorithm for the model of a tandem series of J queues. Recall that the j-th queue has c_j servers, no buffers, and exponential scheduled service time distributions with rate μ_j. Upon service completion the customer moves downstream to queue $j+1$ while it blocks his server as long as there is no server available at this queue. Customers arrive at the network at queue 1 according to a Poisson process with rate γ_1 and leave the network after the J-th queue.

Our approximate algorithm decomposes the network in J isolated (independent) exponential queues of type $M(\lambda)/M(\mu)/c/N$. This is a queueing model with Poisson (λ) arrivals, exponential (μ) service times, c servers, and N buffer or waiting places. In other words, we approximate the process of arrival epochs at the network queues by Poisson processes in the isolated queues. The effective service time distributions are approximated by exponential distributions in the isolated queues. The mean number of customers waiting in the j-th isolated $M(\lambda)/M(\mu)/c/N$ queue approximates the number of blocked servers in the $j-1$-st network queue. These servers are not usable and, therefore, we let the number of servers in the $j-1$-st isolated queue to be equal to the original c_{j-1} minus the mean length of the waiting line of the j-th isolated queue. Finally, since all c_{j-1} servers in the original network may produce a blocked customer waiting to enter queue j, we let the buffer size of the j-th isolated queue to be c_{j-1}. The main effort lies in determining the arrival and service rates of the isolated queues.

Equilibrium performance measures of an $M(\lambda)/M(\mu)/c/N$ queue are easily obtained (e.g. section 2.4[30]). Of interest are

- $L(\lambda, \mu, c, N)$ for the mean number of customers in the system,
- $Q(\lambda, \mu, c, N)$ for the mean number of waiting customers,
- $S(\lambda, \mu, c, N)$ for the mean sojourn time,

- $W(\lambda, \mu, c, N)$ for the mean waiting time,
- $B(\lambda, \mu, c, N)$ for the blocking (or loss) probability (full system),
- $U(\lambda, \mu, c, N)$ for the utilisation.

Because we deal with a tandem series, the throughput of all queues in the network is the same, $\Lambda_j = \Lambda$, where

$$\Lambda := \gamma_1 \left(1 - p_1^{\text{loss}}\right).$$

Suppose that we know this number, then the j-th isolated queue becomes an

$$M(\lambda_j)/M(\epsilon_j)/s_j/N_j$$

queue. According to our heuristic reasoning about the service and buffer size we let

$$s_j = c_j - Q(\lambda_{j+1}, \epsilon_{j+1}, s_{j+1}, N_{j+1}), \quad N_j = c_{j-1}, \tag{2}$$

with $s_J = c_J$ and $N_1 = 0$. The first isolated queue should reflect the (only) entrance queue of the network, with the possibility of loss. Once in the network no more customers are lost, i.e., the arrival rates of the other queues are equal to the network throughput. Hence,

$$\lambda_1 = \gamma_1, \quad \lambda_j = \Lambda \ \ (j = 2, 3, \ldots, J). \tag{3}$$

The service rates ϵ_j in the isolated queues approximate the effective service rates in the network queues. Recall that the effective service time is the scheduled service time minus the transfer time plus the blocking time. By decreasing the number of servers in (2) the blocking time is already taken care of. Hence,

$$\epsilon_1 = \mu_1, \quad \frac{1}{\epsilon_j} = \frac{1}{\mu_j} - W(\lambda_j, \epsilon_j, s_j, N_j) \ \ (j = 2, 3, \ldots, J). \tag{4}$$

Note that the mean sojourn time in the j-th isolated queue equals the mean waiting time plus the mean service time. Thus, by (4), $S(\lambda_j, \epsilon_j, s_j, N_j) = 1/\mu_j$.

The algorithm is an iterative method. It is initialized by estimating the loss probability p_1^{loss} by $p^{(0)} = 0$. Let $p^{(n)}$ be the estimation of the loss probability in the n-th iteration. The network throughput is estimated by $\gamma_1 \left(1 - p^{(n)}\right)$. The isolated queues are analyzed recursively $j = J, J-1, \ldots, 1$ by implementing (2), (3), and by solving the equations (4). The blocking probability of the first isolated queue becomes

$$p_1 = B(\gamma_1, \mu_1, s_1, 0),$$

which should be compared with the estimated loss probability $p^{(n)}$ at the beginning of the iteration. If the difference is relatively large, a new iteration starts with estimated loss probability $p^{(n+1)} := (p^{(n)} + p_1)/2$, otherwise the iteration is stopped, and we approximate the performance measures of section 4 .2 by

$$\begin{aligned} L_j &\approx L(\lambda_j, \epsilon_j, s_j, N_j) - Q(\lambda_j, \epsilon_j, s_j, N_j) + Q(\lambda_{j+1}, \epsilon_{j+1}, s_{j+1}, N_{j+1}), \\ B_j &\approx Q(\lambda_{j+1}, \epsilon_{j+1}, s_{j+1}, N_{j+1}), \end{aligned}$$

$$
\begin{aligned}
S_j &\approx \frac{1}{\mu_j} - W(\lambda_j, \epsilon_j, s_j, N_j) + W(\lambda_{j+1}, \epsilon_{j+1}, s_{j+1}, N_{j+1}) \\
&\qquad = \frac{1}{\epsilon_j} + W(\lambda_{j+1}, \epsilon_{j+1}, s_{j+1}, N_{j+1}), \\
W_j &\approx W(\lambda_{j+1}, \epsilon_{j+1}, s_{j+1}, N_{j+1}), \\
U_j &\approx U(\lambda_j, \epsilon_j, s_j, N_j), \\
p_1^{\text{loss}} &\approx B(\gamma_1, \mu_1, s_1, 0).
\end{aligned}
$$

Remark. We could not prove the convergence of our algorithm, a fact that happens more often in such approximations[12,13,15,17,18]. However, in all our experiments we attained rather quickly (less than 25 iterations) the stopping criterion. In these experiments we applied linear interpolation (of performance measures) to deal with non-integer service capacities s_j in (2).

Remark. We have considered a minor adaptation in reasoning for the arrival rates λ_j of the isolated queues. Because the throughput of the j-th isolated queue should be equal to the network throughput Λ and because the j-th isolated queue looses blocked customers, we should better impose

$$\Lambda = \lambda_j(1 - B(\lambda_j, \epsilon_j, s_j, N_j)), \qquad j = 2, 3, \ldots, J.$$

We have tried to solve the system of two equations originated by this equation to equation (4) for pairs (λ_j, ϵ_j). Although solutions could be determined for separate queues, in many cases we found that the iteration algorithm was not robust. So we decided to use (3) and (4).

5.3 An Algorithm For Arbitrary Configurations

The principles of our algorithm for arbitrary network configurations are similar to those described above. The approximate network consists of J independent (isolated) queues of type $M(\gamma) + M(\lambda)/M(\mu)/c/N$. This queue differs from the previous $M(\lambda)/M(\mu)/c/N$ model in the additional Poisson (γ) arrival process. Customers arriving via the γ stream are called X-type, and customers arriving via the λ stream are called I-type. When all c servers are occupied, arriving I-type customers join the waiting line, arriving X-type customers are lost. An arriving I-type customer is lost when he finds all servers and all N waiting places occupied. The performance measures are now labeled with 5-tuples, $\text{PF}(\gamma, \lambda, \mu, c, N)$, and are determined easily by noticing that the $M(\gamma)+M(\lambda)/M(\mu)/c/N$ queue is a finite birth-death process. Moreover, we distinguish two loss probabilities, $B_X(\gamma, \lambda, \mu, c, N)$ of the X-type customers, and $B_I(\gamma, \lambda, \mu, c, N)$ of the I-type customers.

Suppose that the network loss probabilities p_j^{loss} are known. Then the throughputs in the network are given as the (unique) solution of (1). In our algorithm we approximate the j-th network queue by the queue

$$M(\gamma_j) + M(\lambda_j)/M(\epsilon_j)/s_j/N_j.$$

The X-type customers of this queue represent the external arrivals to network queue j, whereas the I-type customers represent arrivals from other queues in the network. The parameters $\lambda_j, \epsilon_j, s_j, N_j$ are determined by the same line of heuristic reasoning as in the tandem configuration.

$$\begin{aligned} s_j &= c_j - \sum_{i=1}^{J} \frac{\Lambda_j r_{ji}}{\Lambda_i} Q(\gamma_i, \lambda_i, \epsilon_i, s_i, N_i), \\ N_j &= \sum_{i=1}^{J} c_i r_{ij}, \\ \lambda_j &= \sum_{i=1}^{J} \Lambda_i r_{ij}, \\ \frac{1}{\epsilon_j} &= \frac{1}{\mu_j} - W(\gamma_j, \lambda_j, \epsilon_j, s_j, N_j). \end{aligned} \tag{5}$$

The algorithm is an iterative method. The n-th iteration starts with estimates $p_j^{(n)}$ of the loss probabilities p_j^{loss} $(j = 1, 2, \ldots, J)$. After solving the traffic equations (1) for getting approximate throughputs, the set of equations (5) is solved for the parameters of the isolated queues. For the mean numbers of waiting customers $Q(\gamma_i, \lambda_i, \epsilon_i, s_i, N_i)$ in the equation of the server size s_j we used the most recently updated. Thus, some are determined in the n-th iteration, others previously in the $n-1$-st. This depends on the routing probabilities and on the order of considering the queues. Notice that the performance measures $Q()$ and $W()$ in these equations refer to the waiting queue, hence to the internal customers. The n-th iteration ends with calculating the loss probabilities of X-type customers of the isolated queues:

$$p_j = B_X(\gamma_j, \lambda_j, \epsilon_j, s_j, N_j), \quad j = 1, 2, \ldots, J.$$

We stop the iteration when

$$\max_{j=1,2,\ldots,J} \left(|p_j - p_j^{(n)}| \right) < \epsilon,$$

for some small given ϵ, otherwise we continue the iteration with estimated loss probabilities $p_j^{(n+1)} = (p_j + p_j^{(n)})/2$.

6 Application

In this section we return to the original problem of capacity planning of the penitentiary system described in section 2. However, before we can do that, we shall validate our simulation program CellSim and evaluate our numerical program CellNet.

6.1 Data Collection

CellNet calculates the performance of a Poisson-exponential queueing model which should approximate the penitentiary system. The assumption of Poisson arrivals of the detainees is based on the Poisson-exponential model of criminality proposed by Avi-Itzhak and

Sinnar[31]. That is, (i) a criminal commits crimes throughout his career (while not in prison) under a Poisson process with constant rate; (ii) after committing a crime he is imprisoned with a fixed probability; and (iii) the length of his career is exponentially distributed.

We have examined the available data to test the Poisson assumption on the detainees inflow. These data are collected in 1995 at a daily basis by the department of police information of the Dutch Ministry of Justice. As mentioned in the introduction, we distinguish three types of inflow,

- into the remand centres (*preventive inflow*),
- into the self-reporting institutions (*self-reporting inflow*),
- into the institutions for arrested persons (*inflow of arrested persons*).

We discuss here the inflow into the five remand centres only. In 1995 this inflow amounted to about 16500 detainees, in Table 1 specified per remand centre at a daily average. From the figures one immediately conjectures that the coefficient of variation equals one, i.e., a Poisson distribution. In fact, the Kolmogorov-Smirnov goodness-of-fit test does not reject this hypothesis. On the other hand, when we specify the inflows per specific weekday per remand centre and test for stationarity by applying the Dickey-Fuller tests for unit roots, we could not conclude so. For instance, the inflows at Saturdays and Sundays were far smaller than at Mondays. However, these differences have hardly influence on the long-run performances of the system (except perhaps for the percentage of home-sendings).

The other two inflow streams, the self-reporting inflow and the inflow of arrested persons, show the same properties after testing the data with the appropriate tests. That is, Poisson distributions may well be assumed when we consider overall daily streams but not stationarity per day.

	Amsterdam	Arnhem	Den Bosch	Den Haag	Leeuwarden
average daily inflow	15.1	5.8	8.2	14.2	2.3
standard deviation	13.8	4.9	8.8	13.2	2.1

Table 1. Daily inflow in the five remand centres in 1995.

The other issue is the assumption of exponential distributions for the terms of imprisonment given to the prisoners. Unfortunately, data of these terms are not available, because these are not kept in the information system. However, we have data of the realised sojourn times of the detainees at the various prisons which we called the effective service times in our queueing model. In 1995 these data compromised about 24500 completed sojourn times distributed over the prisons. Table 2 summarizes the averages and standard deviations at the different prisons. We expected here no theoretical distributions. Indeed, Kolmogorov-Smirnov tests for exponential distributions gave p-values of 0.0000. Also no other fits could be found with the available sojourn time data. This may be a consequence of the large amount of these data.

	RC	AI	SR	LT	ST	HO	FO
average sojourn time	116.5	84.9	48.8	235.1	83.0	162.2	100.5
standard deviation	131.4	68.4	65.9	222.6	55.8	116.5	317.0

Table 2. Average times spend in prisons (in days).

The conclusion is that our assumptions on Poisson-exponential distributions in the queueing model are first order approximations of the actual stochastic behaviour. These assumptions are made mainly to develop the approximate algorithm which we described above in sections 5.2 and 5.3.

6.2 Program Evaluation

The programs CellNet and CellSim are evaluated through a number of quantitative experiments:

(a) simulation of the penitentiary system (CellSim) with actual inflow and sojourn times data (section 6.1),

(b) simulation of the penitentiary system (CellSim) with Poisson-exponential arrivals and service times,

(c) the approximate algorithm (sections 5.2 and 5.3) of the queueing model (CellNet).

The means of the assumed Poisson-exponential distributions in the experiments (b) and (c) are estimated by the data averages (cf. section 6.1). Furthermore, the routing probabilities (r_{ij}) in the queueing network model are estimated by the average transfers between the penal institutions. These data were supplied also by the Ministry of Justice.

We executed experiments (a) for validating CellSim. The simulation results were compared with the actual performance measures. The outcomes were satisfactory except for one aspect. Using 95% confidence intervals we obtain satisfying agreements between the simulation outcomes and the given data for the sojourn times (cf. Tables 2 and 3). But the percentage of home-sendings at a full remand centre came out too low in the simulation: [8.8% : 9.9%] 95%-confidence interval against 16.4% realised. Maybe, here nonstationarity of inflow plays a role which we ignore in the simulation. Therefore, one should use the outcomes more as relative measures to allow comparisons, than as absolute measures. We refer to Korporaal[32] for more details.

RC	[112.7 : 113.6]
AI	[82.0 : 83.8]
SR	[49.2 : 50.0]
LT	[231.1 : 236.0]

ST	[81.4 : 84.1]
HO	[157.7 : 162.4]
FO	[100.4 : 104.5]

Table 3. After simulation: 95% confidence intervals of estimates of average times spend in prisons (in days).

The experiments (b) are executed for testing our approximate algorithm: both the experiments (b) and (c) assume the same stochastic modeling. All these tests do assure that the algorithm is indeed a good approximation, in the sense that the majority of the calculated performance measures agree with the simulation outcomes. To illustrate our findings, we shall give the comparison in case the input parameters of arrivals (γ_i), terms of imprisonment (μ_i), routing (r_{ij}), and capacities (c_i) are given by the actual figures of 1995, summarized in Table 4. Recall that the scheduled service rates μ_i were actually not known from the data. We first executed some simulation tests to estimate these from the given realised service times[30].

The results of both the algorithm and the simulations are listed in Table 5. The simulations were executed with the batch means method and a fixed sample size: 40 batches, each batch realises 20000 events due to routing (transfers from one prison to another or back to society). The table gives the performances after translation to the terms of the penitentiary system: the mean realised sojourn times S_i, the mean waiting times W_i, the mean number of waiting persons B_i, the percentage of home-sendings p_i^{loss}, and the times of occupation U_i. As we can see, the latter ones are underestimated by the approximation algorithm. A reason may be that the algorithm gives full capacities (100% utilisation) only if the inflow is exceptionally large.

The results of the simulations are reported only through the 95% confidence intervals of the estimates.

i	c_i	γ_i (daily)	$1/\mu_i$ (days)	r_{ij} RC	AI	SR	LT	ST	HO	FO	free
RC	5044	45.6	102				0.12	0.06	0.05	0.02	0.75
AI	592	9.0	70	0.09			0.08	0.18	0.02		0.63
SR	394	12.0	48	0.05				0.01		0.03	0.91
LT	1061		228	0.14				0.02	0.09	0.06	0.69
ST	182		79	0.01							0.99
HO	348		150	0.07						0.34	0.59
FO	171		102	0.09							0.91

Table 4. Input data. Empty entries are 0 (are not possible).

i	S_i (days)	W_i (days)	B_i	U_i (%)	p_i^{loss} (%)
RC	113.5	11.4	506.1	99.7	7.8
	[112.7 : 113.5]	[11.2 : 11.8]	[501.3 : 523.7]	[99.7 : 99.8]	[7.0 : 8.0]
AI	81.6	11.6	83.4	99.3	
	[81.6 : 82.5]	[11.3 : 12.0]	[81.1 : 85.9]	[99.3 : 99.4]	
SR	49.4	1.4	10.7	99.5	
	[48.7 : 49.6]	[1.2 : 1.4]	[10.0 : 11.4]	[99.5 : 99.5]	
LT	179.9	6.4	37.9	98.8	
	[176.4 : 182.1]	[4.8 : 5.3]	[28.5 : 31.5]	[100.0 : 100.0]	
ST	43.6			99.5	
	[43.4 : 44.9]			[100.0 : 100.0]	
HO	119.8	11.3	32.6	99.5	
	[123.0 : 127.3]	[8.0 : 9.4]	[22.2 : 26.4]	[100.0 : 100.0]	
FO	68.9			99.2	
	[77.5 : 81.3]			[100.0 : 100.0]	

Table 5. Results of CellNet and CellSim. Empty entries are 0 (do not occur).

6.3 CellNet Use

Using CellNet, performance of different scenarios of capacity extensions can be calculated quickly. As an example, consider the situation given above in Table 4. Suppose that the total capacity of all prisons can be extended by 1000 cells and suppose that the cost of a cell is the same in all the different kinds of prisons. The question is which prison to extend to what amount. The conditions are that certain performance criteria should be met. Suppose that the allocation scenarios of Table 6 are considered.

	RC	AI	SR	LT	ST	HO	FO
scenario 1	1000	0	0	0	0	0	0
scenario 2	0	0	0	500	200	100	100
scenario 3	300	0	0	500	100	100	0

Table 6. Allocation scenarios of 1000 extra cells.

We do not present the detailed performances of each of the scenarios as in Table 5. We summarize these by showing in Table 7 the cumulative total of the waiting times, i.e., $\sum_i W_i$, the cumulative total of the numbers of waiting persons, $\sum_i B_i$, and the percentage of home-sendings at the remand centres. (The cumulative total of the sojourn times, $\sum_i S_i$, remains the same in all scenarios.)

	$\sum_i W_i$	$\sum_i B_i$	p^{loss}
original	42	670	7.8
scenario 1	48	983	1.7
scenario 2	1	14	5.8
scenario 3	22	289	3.4

Table 7. Performances of 3 extension scenarios compared to the original situation (in Table 5).

It is up to the decision maker whether it is more important to have a small percentage of home-sendings, meaning that most arrestants can be imprisoned immediately, but resulting in longer waiting times (scenario 1). Or that the waiting times are low, meaning that most scheduled transfers can be realised immediately, but resulting in more home-sendings (scenario 2). Scenario 3 seems to take both criteria into account and its performances are averages of the other two scenarios. (The average occupation times in scenarios 1 and 3 is more than 99%, in scenario 2 almost 98%.) Also we see that the number of waiting persons decreases dramatically in scenario 2.

7 Conclusion

In this paper we considered the problem of assessing the required number of prison cells in various detention centers as to meet several performance criteria, like fraction of criminals sent home, waiting time for transfers, etc. The problem is highly relevant because of the high costs associated to prison cells and the public sensitivity of sending home criminals. Proper decision making should rely on proper information and tools to assess the effects

of various possible actions. Generating proper information is costly and a difficult task because of the many different interpretations of the concepts used. Yet, even while not enough information is available, decisions have to be taken to deal with the actual problem. The CellNet and CellSim programs are useful instruments to support decision making by the Ministry of Justice. Although simulation is a very flexible tool, its usefulness in this case was hindered by a lack of proper information about the actual process and by long development and computation times. Although the approximate method underlying the CellNet program uses rough approximations of the actual process, it produced reasonable outcomes within relatively short computation and development times. Furthermore, we think that CellNet can be used for other purposes as well. For instance, it can evaluate quickly the performance when the numbers of inflow of detainees change, or when the terms of sentences change. Finally we remark that since the time of development of CellNet and our involvement with testing, the government of the Netherlands indeed decided to extend the number of cells considerably. At the time of writing there are about 11600 persons imprisoned, while there were only 400 home-sendings during 1998. We must add here that the occupation was only 96%. The government expects, though, that after year 2000 shortage of cells will happen again.

Acknowledgement

The authors would like to thank Arnold van Gemmert of the department of policy information of the Ministry of Justice for his comments and suggestions.

References

[1] Ministry of Justice. Department of police information. *Calculation of nominal prices of prisons 1997* (in Dutch), 1996.

[2] M.D. Maltz. Operations research in studying crime and justice: its history and accomplishments. In: *Handbooks in Operations Research and Management Science*, Vol. 6 *Operations Research in the Public Sector*, eds. Pollock, Rothkopf and Barnett. North-Holland, Amsterdam 1994, p. 201 – 262.

[3] S.J.C. Cuvelier. Transforming projections into policy, the evolution of computer simulation in correctional research. *Int. Seminar on Prison Population Projections*, Manchester, England, July 1991

[4] F.P. Kelly. Networks of queues. *Advances of Applied Probability.* Vol. 8, p. 416 – 432, 1976.

[5] J. Walrand. *An introduction to queueing networks.* Prentice Hall. Englewood Cliffs 1988.

[6] S. Balsamo & V. de Nitto Personè. A survey of product form queueing networks with blocking and their equivalences. *Annals of Operations Research* Vol. 48, p. 31 – 61, 1994.

[7] K.-P. Jun & H.G. Perros. Approximate analysis of arbitrary configurations of queueing networks with blocking and deadlock. In: *Proceedings First International Conference on Queueing Networks with Blocking*, eds. Perros and Altiok. North-Holland, Amsterdam 1989, p. 259 – 279.

[8] P. Glasserman. *Gradient estimation via perturbation analysis.* Kluwer. Norwell 1991.

[9] H.G. Perros. *Queueing networks with blocking.* Oxford University Press, 1994.

[10] Y. Dallery & Y. Frein. On decomposition methods for tandem queueing networks with blocking. *Operations Research* Vol. 41, p. 386 – 399, 1993.

[11] F.S. Hillier & R.W. Boling. Finite queues in series with exponential or Erlang service times – a numerical approach. *Operations Research* Vol. 15, p. 286 – 303, 1967.

[12] T. Altiok. Approximate analysis of exponential tandem queues with blocking. *European Journal of Operational Research* Vol. 11, p. 390 – 398, 1982.

[13] H.G. Perros & T. Altiok. Approximate analysis of open networks of queues with blocking: tandem configurations. *IEEE Transactions on Software Engineering* Vol. 12, p. 450 – 461, 1986.

[14] T. Altiok. Approximate analysis of queues in series with phase-type service times and blocking. *Operations Research* Vol. 37, p. 601 – 610, 1989.

[15] A. Brandwajn & Y.-L.L. Jow. An approximation method for tandem queues with blocking. *Operations Research* Vol. 36, p. 73 – 83, 1988.

[16] Y. Takahashi, H. Miyahara & T. Hasegawa. An approximation method for open restricted queueing networks. *Operations Research* Vol. 28, p. 594 – 602, 1980.

[17] T. Altiok & H.G. Perros. Approximate analysis of arbitrary configurations of open queueing networks with blocking. *Annals of Operations Research* Vol. 9, p. 481 – 509, 1987.

[18] H.-S. Lee & S.M. Pollock. Approximation analysis of open acyclic exponential queueing networks with blocking. *Operations Research* Vol. 38, p. 1123 – 1134, 1990.

[19] J. Labetoulle & G. Pujolle. Isolation method in a network of queues. *IEEE Transactions on Software Engineering* Vol. 6, p. 373 – 381, 1980.

[20] D.D. Kouvatsos & N.P. Xenios. Maximum entropy analysis of general queueing networks with blocking. In: *Proceedings First International Conference on Queueing Networks with Blocking*, eds. Perros and Altiok. North-Holland, Amsterdam 1989, p. 281 – 309.

[21] L. Kerbache & J.M. Smith. The generalized expansion method for open finite queueing networks. *European Journal of Operational Research* Vol. 32, p. 448 – 461, 1987.

[22] D.D. Kouvatsos & N.P. Xenios. MEM for arbitrary queueing networks with multiple general servers and repetitive-service blocking. *Performance Evaluation* Vol. 10, p. 169 – 195, 1989.

[23] Y. Han & J.M. Smith. Approximate analysis of open M/M/C/K queueing networks. In: *Proceedings Second International Conference on Queueing Networks with Finite Capacity*, eds. Onvural and Akyildiz. North-Holland, Amsterdam 1993, p. 113 – 126.

[24] I.F. Akyildiz. Product form approximations for queueing networks with multiple servers and blocking. *IEEE Transactions on Computers* Vol. 38, p. 99 – 115, 1989.

[25] H.T. Papadopoulos & C. Heavey. Queueing theory in manufacturing systems analysis and design: A classification of models for production and transfer lines. *European Journal of Operational Research* Vol. 92, p. 1 – 27, 1996.

[26] E.J. Muth. Stochastic processes and their network representations associated with a production line model. *European Journal of Operational Research* Vol. 15, p. 63 – 83, 1984.

[27] H.T. Papadopoulos. The throughput of multistation production lines with no intermediate buffers. *Operations Research* Vol. 43, p. 712 – 717, 1995.

[28] D. Gross & C.M. Harris. *Fundamentals of queueing theory.* 2-nd ed. Wiley. New York 1985.

[29] B. Avi-Itzhak & R. Shinnar. Quantitative models in crime control. *Journal of Criminal Justice* Vol. 1, p. 185 – 217, 1973.

[30] R. Korporaal. Decision support for capacity planning of prisons. Master Thesis. Econometrisch Instituut, Erasmus Universiteit Rotterdam, 1996 (in Dutch).