



INTERNATIONAL ASSOCIATION FOR RESEARCH AND TEACHING
Economics, Finance, Operations Research, Econometrics and Statistics

++ research ++ teaching ++

ECORE DISCUSSION PAPER

2008/33

Approximating Multiple Class Queueing Models With Loss Models.

Philippe CHEVALIER
Jean-Christophe VAN DEN SCHRIECK

Approximating Multiple Class Queueing Models with Loss Models

Philippe Chevalier

philippe.chevalier@uclouvain.be

Jean-Christophe Van den Schrieck

jc.vandenschrieck@uclouvain.be

Louvain School of Management – CORE
Université catholique de Louvain, Belgium

Feb, 2008

Abstract

Multiple class queueing models arise in situations where some flexibility is sought through pooling of demands for different services. Earlier research has shown that most of the benefits of flexibility can be obtained with only a small proportion of cross-trained operators. Predicting the performance of a system with different types of demands and operator pools with different skills is very difficult. We present an approximation method that is based on equivalent loss systems. We successively develop approximations for the waiting probability, the average waiting time and the service level. Our approximations are validated using a series of simulations. Along the way we present some interesting insights into some similarities between queueing systems and equivalent loss systems that have to our knowledge never been reported in the literature.

1 Introduction

Flexibility is of increasing importance in service industries. In order to meet this competitive challenge, many companies use cross-trained staff, i.e. employees that have some expertise in more than one field. Although cross-training brings more flexibility, it is also more expensive than hiring single-skilled employees. Needless to say, such cross-trained employees are scarce too, making it impractical to have all employees capable of handling each task. Earlier research showed that “a little flexibility goes a long way”. In other words, it is possible to reap most of the benefits of a fully cross-trained workforce with much less cross-training. On this topic, the reader might be interested by [Wallace and Whitt, 2005] or [Chevalier et al., 2005]. The former illustrates that hiring double skilled agents permits to capture most of the variability. In the second paper, the authors find out that a good practice would be to dedicate twenty percent of a staffing budget on flexible agents.

This is especially true for call center companies. Nowadays they often handle many different types of calls each requiring specific skills. Flexibility is critical as demand is stochastic and requires quick response from the service provider. This has resulted into the development of multi-class queueing models in the literature. For a review of the main models used, we refer to [Gans et al., 2003] and [Aksin et al., 2007]. On the general problem of routing and staffing in multi-skill call centers we highly advise [Koole and Pot, 2006].

Evaluating the performance of a multi-skill queueing system is a challenging issue per se. [Shumsky, 2004] proposes an interesting approximation to evaluate the performance of a queueing system with two types of demands and two types of operators, one being single-skilled and the other type being polyvalent. The author divides the state-space into different areas. This procedure permits to diminish the computation burden significantly, even for large number of operators, while keeping accurate results. In [Avramidis et al., 2008] a multi-skill queueing model is proposed that is then solved in

order to find good solutions. The final purpose is similar to this article, but we present a very different approach.

Among the queueing systems, the systems with zero queue length are worth mentioning. These systems are often referred to as loss systems. Of course the no-queue assumption is very restrictive and unrealistic in most cases but these systems have the advantage of being much simpler and easier to analyse than other queueing systems. We propose to make use of this valuable advantage to approximate the performance of queueing systems.

Our main objective is to build approximations for multi-class queueing systems that would be easy to use and be quick to compute. In an earlier paper, [Chevalier and Van den Schrieck, 2006], we noticed that the relative performance of loss systems is very often a very good proxy for the relative performance of similar queueing systems. This gave us the idea of a more thorough investigation of the potential to use loss systems to build approximation techniques for the performance measurement of queueing systems. In the current paper we show that very good results can be obtained with such techniques.

We propose to work with a model that assumes infinite queue length and that does not consider impatience. The main reason is that queues of infinite length are in a sense at the other extreme compared to loss systems, that have queues of zero length. We actually believe that if we can find good approximations for infinite queues, finding approximations for systems with queues of limited length – which are closer to loss systems – should be possible.

The use of loss systems as a benchmark for the performance of queueing systems has been widely discussed. It is mentioned in [Franx et al., 2006]. [Koole et al., 2003] argue that the relative performance of a multi-skill loss system can be used to approximate the same system with queues. They also note that for a single class exponential server there is a closed form formula to compute the waiting probability based on the loss probability of an equivalent blocking system (see f.i. [Cooper, 1972]). Here we will present extensions to this formula for multiple class systems and for different

performance measures.

Many measures are used to evaluate the performance of a queueing system. The most usual are the probability of having to wait, the average waiting time and the service level, i.e. the probability of being answered within a certain time interval. We try in this paper to develop a method for each of these measures. The outline is therefore as follows. In the next section we present the two models we intend to compare. In section 3 we briefly describe the set of simulations we made to illustrate and evaluate the approximations. This is followed by a section that presents a method to compute the probability of waiting. In section 5 a method for obtaining the average waiting time is presented. Section 6 focuses on the service level. We end with a concluding section that also lists some of the possible extensions.

2 Multiclass queues

We study queueing systems that handle multiple classes of demands. Each class of demand requires some specific competence from the server that will handle it. In order to respond to these demands, the queueing system comprises pools of operators that can handle some subset of the demand classes. Pools are homogeneous groups of agents, that have exactly the same competences.

The area where such queueing systems are most widely used in practice is certainly call centers, where each competence might for example correspond to a language. There are other areas where such systems are used such as maintenance services, but to make this article more concrete, from now on we will focus on call centers as the underlying application.

By grouping different types of calls, call centers actually try to benefit from the economies of scale made possible by pooling. There exists an abundant literature on pooling. [Mandelbaum and Reiman, 1996] reviews the different types of pooling in call centers. They analyse when pooling is adequate and the cases where pooling is counterproductive using an efficiency index.

We suppose that the arrival streams are Poisson processes and that the processing times are exponentially distributed. The service time is the same for all call types and in any operator pool. The routing of calls is supposed to follow a static priority index: that is, for each particular call type, the different pools that are liable to handle it are ordered. When a call arrives it is sent successively to each pool in the list according to this order, until an operator is found that can handle it. On the other hand when an operator finishes a call, the call that has arrived earliest among the different types this operator can handle is sent to him/her. If there are no calls waiting that the operator can handle the operator will remain idle.

To compute the performance measures we draw on the analogy with a loss system where arrivals, operator pools and the routing of arriving calls are identical but, if no available operator can be found to handle a call immediately, the call is rejected rather than put on hold in a queue. Consequently when an operator finishes a call, he/she will remain idle until a new call arrives that is sent to him/her. Figure 1 shows a simple example of the type of systems we are studying.

A crucial aspect of the analysis of multi-skill loss models is the analysis of overflows. These are flows of calls that are not answered at a given pool. They have specific characteristics that make them difficult to analyse. The major difficulty is to determine the performance of a pool when its input is an overflow or a combination of various overflows. To face this problem many approximation methods have been developed. Among others we can cite the hyperexponential method that is presented in [Franx et al., 2006] and, to a lesser extent, in [Koole et al., 2003] or the Equivalent Random Method first presented in [Wilkinson, 1956]. It is also described in [Jagerman et al., 1997].

We propose to use another method: the Hayward approximation. This approximation was first presented in [Wilkinson, 1956]. It was further developed in [Fredericks, 1980] and was extended to call centers in [Chevalier and Tabordon, 2003]. The idea is to take the volatility of the overflow into account by working with a new parameter: the peakedness. The peakedness

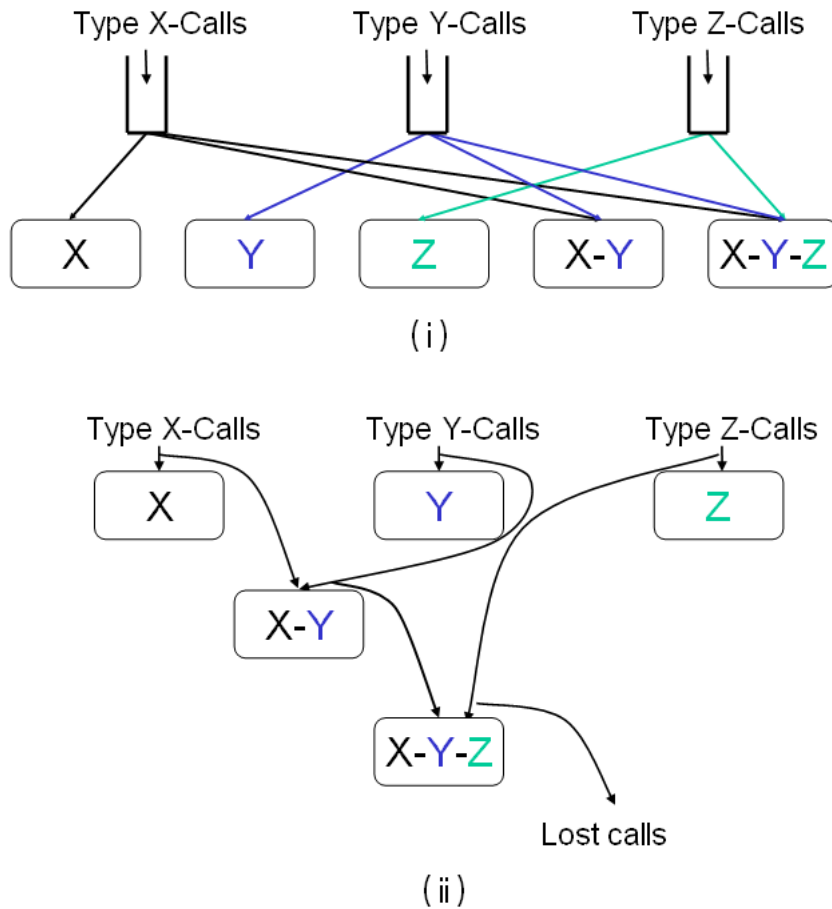


Figure 1: : a simple example of a multi-class, multi-pool system: (i) represents the system with queues and (ii) is the equivalent system without queues

is the ratio of the variance over the mean of the number of operators that are busy if the analysed flow would be treated by a pool with an infinite number of operators. It is relatively easy to see that the peakedness of a Poisson process is equal to one. This follows directly from the properties of an $M/G/\infty$ queueing system. For an overflow the peakedness is larger than one, reflecting the “bursty” nature of this type of flow. The Hayward ap-

proximation consists of introducing the peakedness in the Erlang-B formula – or its continuous version – by dividing both parameters of this formula, namely the offered load and the number of operators, by the value of the peakedness. To summarize, we have the blocking probability :

$$B(a, z, s) \approx B_E(a/z, s/z), \quad (1)$$

where $B_E(.,.)$ is the Erlang-B formula. a is the offered load to the system, i.e. the arrival rate λ divided by the service rate μ . s is the number of operators and z is the peakedness of the incoming flow.

The peakedness for an overflow can be computed exactly when the incoming flow is Poisson using the following formula:

$$z = 1 - aB(a, s) + \frac{a}{s + 1 + aB(a, s) - a} \quad (2)$$

In case the input is not Poisson, the peakedness can be approximated by the formula proposed in [Fredericks, 1980]:

$$z^{out} \simeq z^{in} \left(1 - \frac{a}{z^{in}} B\left(\frac{s}{z^{in}}, \frac{a}{z^{in}}\right) + \frac{a}{s + z^{in} + aB\left(\frac{s}{z^{in}}, \frac{a}{z^{in}}\right) - a} \right) \quad (3)$$

For an evaluation of the method the interested reader is referred to [Tabordon, 2002]. She shows that this approximation is both simple and accurate, making it very tractable in practice.

3 Notations and Description of the Simulation Data Set

To describe our method we will use a small example with two types of calls and three pools of operators. Figure 2 depicts the structure of this system. The generalization to more complex situations is straightforward, but it would entice a lot of cumbersome notations. The calls are referred to as type- X calls and type- Y calls and are assumed to arrive according to two independent Poisson processes respectively of rate λ_X and λ_Y . The system consists in two dedicated pools P_X and P_Y and one cross-trained pool P_{XY} ,

this latter pool being able to handle both demand streams. We assume that each call has the same (exponential) service time distribution. The number of operators in one pool is noted N_j ($j \in \{X, Y, XY\}$).

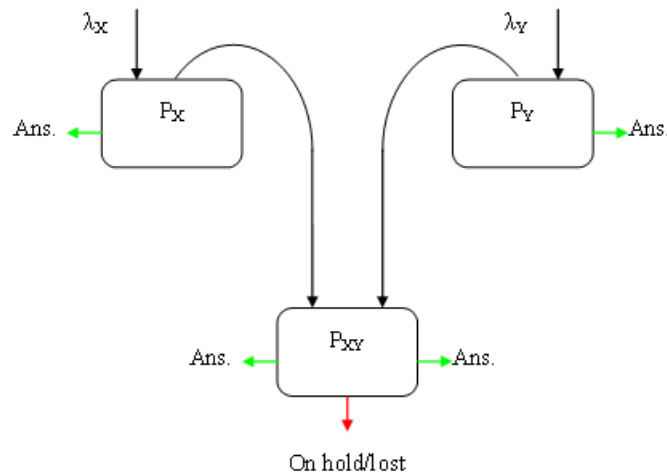


Figure 2: : structure of a 2 call types call center

The priority rules are $\{X, XY\}$ and $\{Y, XY\}$ for type- X and type- Y calls respectively. This means that in both cases the calls are first sent to the pool with specialized operators and then to the pool with polyvalent operators. The objective of this routing policy is to keep the more polyvalent operators available for future uncertain demand.

In this paper, we only present an illustrative part of the simulations we made. All the methods presented in the next three sections will be illustrated with this data set. This set consists in systems with two call types. There are six combinations of demands, as shown in Table 1. The different cases were built such as to vary the total load of the system as well as the imbalance between both arrival rates.

Each of the combinations of arrival streams presented in Table 1 is combined with three series of 5 different pool size combinations to obtain a series

| Example | λ_X | λ_Y |
|---------|-------------|-------------|
| 1 | 2 | 2 |
| 2 | 2 | 3 |
| 3 | 2 | 5 |
| 4 | 3 | 10 |
| 5 | 4 | 5 |
| 6 | 5 | 5 |

Table 1: The six different combinations of arrivals.

of 90 experiments. The set of experiments was created in order to try to more or less exhaustively test all combinations with utilizations varying from 0.7 to 0.95. Table 2 provides all the information about the experiments. The expected service time is 1 in all experiments.

| Arrival rates | | | series 1 | | | series 2 | | | series 3 | | |
|---------------|-------------|-------------|----------|-------|--------------------------|----------------------------|-------|----------|----------|---------------------------|----------|
| Ex. | λ_X | λ_Y | N_X | N_Y | N_{XY} | N_X | N_Y | N_{XY} | N_X | N_Y | N_{XY} |
| 1 | 2 | 2 | 2 | 2 | $x \in \{1, \dots, 5\}$ | $x \in \{1, \dots, 5\}$ | 3 | 1 | 2 | $x \in \{1, 3, 4, 5, 6\}$ | 2 |
| 2 | 2 | 3 | 2 | 2 | $x \in \{2, \dots, 6\}$ | $x \in \{1, \dots, 5\}$ | 3 | 2 | 1 | $x \in \{1, \dots, 5\}$ | 5 |
| 3 | 2 | 5 | 3 | 3 | $x \in \{3, \dots, 7\}$ | $x \in \{1, \dots, 5\}$ | 4 | 3 | 3 | $x \in \{1, \dots, 5\}$ | 2 |
| 4 | 3 | 10 | 5 | 5 | $x \in \{6, \dots, 10\}$ | $x \in \{1, \dots, 5\}$ | 8 | 5 | 2 | $x \in \{10, \dots, 14\}$ | 2 |
| 5 | 4 | 5 | 3 | 3 | $x \in \{4, \dots, 8\}$ | $x \in \{2, 4, 6, 8, 10\}$ | 4 | 5 | 5 | $x \in \{4, \dots, 8\}$ | 3 |
| 6 | 5 | 5 | 4 | 4 | $x \in \{3, \dots, 7\}$ | $x \in \{2, 4, 6, 8, 10\}$ | 4 | 5 | 5 | $x \in \{4, \dots, 8\}$ | 3 |

Table 2: The different settings used. In the first series, the number of operators in the polyvalent pool, N_{XY} , varies. In the second and third series, this is respectively N_Y and N_X which change.

For each of the 90 cases we conducted 15 different simulations of 16000 time units with a warm-up period of 1000 time units, making a total of 15000 time units available for analysis. For each set of simulation, we computed confidence intervals for the loss probabilities, waiting probabilities and average waiting time. The relative error was less than 5%, lying in general around 1 or 2%.

For each setting, the simulations were made in such a way that the two systems receive exactly the same input. Practically our simulation tool first generates a set of arrivals for the entire simulation length. For each arrival, a service time is also generated. These values are then recorded so that they can be used in both systems.

4 Approximating the Waiting Probability

The waiting probability is the performance measure for the queueing system that is closest to the loss probability. Basic algebra reveals a link between the Erlang-B formula, that gives the loss probability, and the Erlang-C formula. The latter computes the waiting probability in the $M/M/s$ context. This link is mentioned in [Cooper, 1972] and in [Koole et al., 2003]. Formally we can write that:

$$C(s, a) = \frac{sB(s, a)}{s - a(1 - B(s, a))}, \quad (4)$$

where s is the number of operators, $a = \frac{\lambda}{\mu}$ is the incoming load and $B(s, a)$ and $C(s, a)$ are respectively the Erlang-B and -C probabilities.

In a multi-skill call center, there are separate queues for each type of arrivals. The waiting probability can therefore be very different from one type of arrival to another although the cross-trained operators that can handle different types of calls will create some dependence between the different call types. We need to compute the waiting probability for each type of calls.

By analogy with equation (4), we will estimate the waiting probability for type i calls as:

$$\hat{WP}_i = \frac{s_i L_i}{s_i - a_i(1 - L_i)}, \quad (5)$$

where

L_i is the loss probability for type i calls if they were treated by the equivalent loss system as explained in section 2,

a_i is the load for type i arrivals (λ_i/μ_i),

s_i is the equivalent number of operators that handle type i calls in the global system.

Although L_i is difficult to compute exactly, we can use the method outlined in section 2, the Hayward approximation, to obtain a very good estimation of the value of this parameter. The load a_i is given and poses no problem. The major difficulty is to determine an adequate value for s_i . This is the goal of the next paragraph.

To determine the value s_i , we tested the hypothesis that the fraction of the busy time for the cross-trained servers devoted to each call type is almost identical for the loss and the queueing system. To our knowledge no study about this property has been published so far. Figure 3 shows the comparison between the simulated loss system and the simulated queueing system. This seems to be a key finding for our approximation.

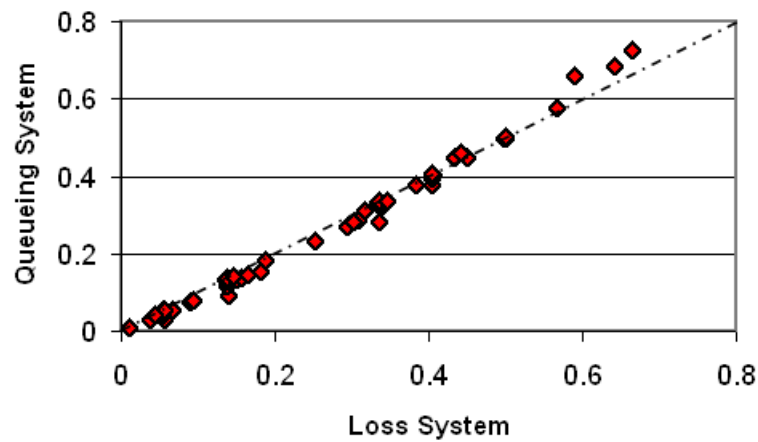


Figure 3: : Comparison of the proportion of time dedicated to the type- X calls at the cross trained pool in the loss and queueing systems.

The simulation study strongly supports our hypothesis. For most of the

examples the proportion is nearly identical for both systems. The cases where (slight) divergences are observed correspond in general to the heavily loaded systems. Note also that this result is dependent on the fact that no priority is given between two classes of waiting customers. As explained earlier, when a cross-trained operator becomes available (s)he gets to handle the call that arrived earliest among all calls (s)he is able to handle.

Based on this observation we can estimate values s_i for equation (5) in the following way. We use the equivalent loss system to estimate the proportion of time the different pools of operators spend on each type of call. We then split the number of operators in each pool according to these proportions in subgroups for each type of call. Finally, we sum the operators for each type of call from the subgroups of each pool.

Figure 4 presents a comparison between the approximation we obtained and the simulation results for the waiting probability.

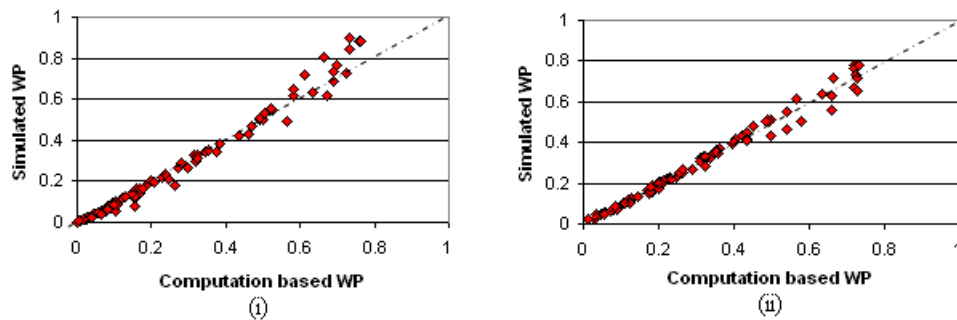


Figure 4: : Comparison of the waiting probability observed in simulation with the approximation based on the equivalent loss systems. (i) gives WP_X and (ii) gives WP_Y .

These results are quite good. They are very accurate when the waiting probability is lower. We notice again that when the system is heavily loaded our approximation is not as accurate.

5 Average waiting time

Another measure of importance is the average waiting time before being served. For an $M/M/s$ system, it is possible to compute it using the following formula:

$$WT = \frac{1}{\mu} \frac{a^s}{(s-1)!(s-a)^2} p_0 \quad (6)$$

$$p_0 = \left(\sum_{i=0}^{s-1} \frac{a^i}{i!} + \frac{a^s}{s!} \frac{s}{(s-a)} \right)^{-1} \quad (7)$$

We can express it from the Erlang-C :

$$WT = \frac{1}{\mu} \frac{C(s, a)}{(s-a)} \quad (8)$$

This last formula can be interpreted as the expected service time multiplied by the waiting probability and divided by the average idleness rate of all servers. We will use this observation to build our approximation.

5.1 Bounding the average waiting time

Intuitively the waiting calls all benefit to some extent from the idle capacity of all pools thanks to the first come first served rule. Indeed, for a call of a particular type that is waiting, the fact that calls of other types are handled quickly increases the probability of this call being the one that has waited longest when an adequate operator becomes available.

From this we can derive bounds on the estimations of the waiting time. Indeed, a lower bound on the waiting time is obtained if we suppose all call types fully benefit from the total idleness rate of all operators of all pools. This would give the following estimate for the average waiting time:

$$\hat{WT}_{i,low} = \frac{1}{\mu} \frac{\hat{W}P_i}{(\sum_{j \in C_i} n_j - \sum_k a_k)} \quad (9)$$

Where C_i is the set of all pools that can answer type- i calls.

On the other hand we can derive an upper bound on the waiting time if we suppose that the system behaves as if there was no interaction between

the different call types. To compute this we use the equivalent number of operators dedicated to type- i calls as derived in section 4.

$$\hat{W}T_{i,up} = \frac{1}{\mu} \frac{\hat{W}P_i}{(s_i - a_i)} \quad (10)$$

We use the same simulation data as in the preceding section to illustrate these bounds. In Figure 5 we compare these bounds (vertical axis) with the observed average waiting times (horizontal axis).

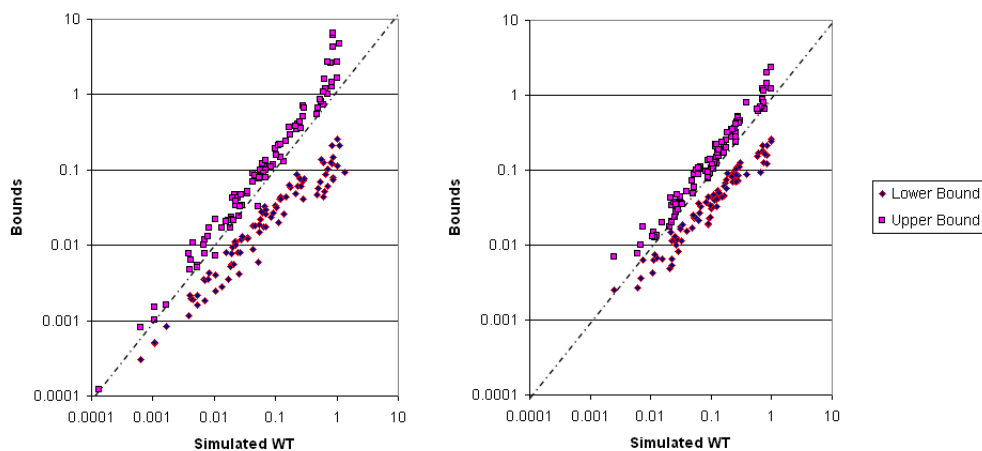


Figure 5: : the bounds on the average waiting time, as functions of the simulated waiting time. (i) gives the bounds on WT_X and (ii) on WT_Y (the axes have a logarithmic scale)

We observe that the values obtained by computation are very good bounds on the waiting time: the upper bounds lie above the 45 degrees line while the lower bounds are below.

5.2 An approximation for the waiting time

The previous results confirm that our interpretation of equation (8) seems to give good results. In order to improve our approximation of the average

waiting time we try to estimate the idleness rates, I_j , of each pool of operators, i.e., the exceeding capacity when taking into account all calls that are answered on average at the pool. For a given call type we then sum the idleness rates of all the pools that are liable to handle that call type.

Consequently we need to find the proportion of calls that is answered by each pool of operators. Again we have to find a way to approximate that proportion and once again we propose to compare the situation in the queueing system with the situation in the equivalent loss system. This proposal is justified by the results of Figure 6 which clearly shows that this proportion is roughly the same in a loss system and in the corresponding queueing system.

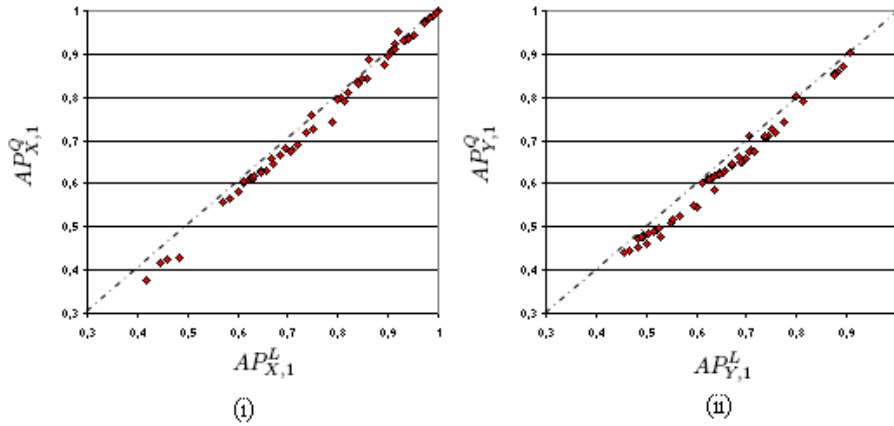


Figure 6: : Comparison of the proportion of calls treated by the cross-trained operators in the Loss and in the Queueing Systems. (respectively $AP_{i,1}^L$ and $AP_{i,1}^Q$ for call types $i = \{X, Y\}$)

We observe that in general the proportion is a bit higher in the loss system than in the equivalent queueing one. The difference is however sufficiently for it to be overlooked.

Our approximation will thus be computed as follows:

1. we compute the overflows at each level of the loss system, from this

we deduce for each call type the proportion that is handled by each pool.

2. We extrapolate these proportions for the queueing system (where all calls are treated, contrary to the loss system).
3. We compute the rates of calls that will be handled at each pool.
4. We deduce the idleness rate I_j for each pool j .
5. We use equation (8) where we replace the Erlang C with the waiting probability found in the previous section and the denominator with the sum of the idleness rates for the pools that handle the corresponding call type.

This gives us the following formula:

$$\hat{W}T_i = \frac{1}{\mu} \frac{\hat{W}P_i}{\left(\sum_{j \in G_i} I_j\right)} \quad (11)$$

Where G_i stands for the set of pools able to handle type i calls.

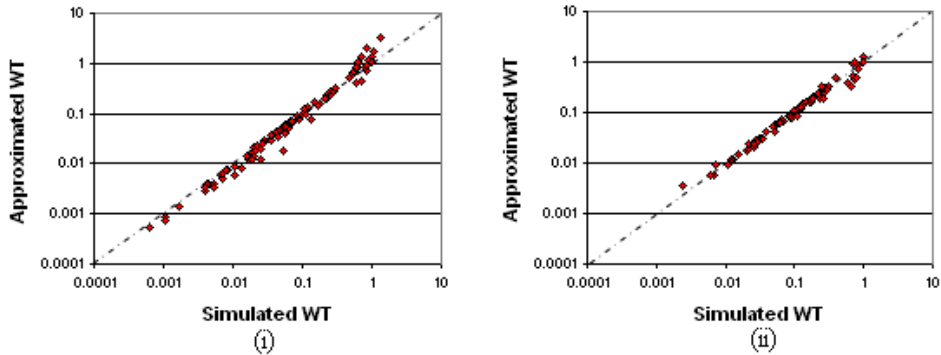


Figure 7: : The approximation of the average waiting time for the X-calls (i) and the Y-calls (ii) based on computations. The axes have a logarithmic scale.

The results of Figure 7 show again that the quality of the approximation is quite good, with some deterioration for the heavily loaded systems. Notice

that we switched to a logarithmic scale in order to have more evenly spread values. The relative accuracy of our approximation is not as good as for the waiting probability though.

6 Service level

A third measure of performance is the service level. It gives the proportion of calls that are being answered within a given time. In other words this is the proportion of calls that do not wait more than a given limit. This measure is important as there exist regulations in some countries that impose minimum performances in terms of service level and as many contracts in call center industry use service level as the performance measure. See [Avramidis et al., 2008] or [Hasija et al., 2008] for some applications involving service level measurements.

In a single-skill $M/M/s$ setting, it is easy to compute a service level because the distribution of the waiting time is known. Conditionally on the fact that an arrival has to wait the waiting time is exponentially distributed (see f.i. [Khintchine, 1960] or [Gross and Harris, 1998]). So we have:

$$Pr[WT < t | WT > 0] = 1 - e^{-(s\mu - \lambda)t} \quad (12)$$

With this formula, it is easy to compute the service level. As the probability of waiting is given by the Erlang-C, the total proportion of calls that are answered within a time t , is the product of the Erlang-C and the conditional probability of equation (12) plus the proportion of calls that are answered immediately. In short:

$$Pr[WT < t] = WP(1 - e^{-(s\mu - \lambda)t}) + 1 - WP \quad (13)$$

$$= 1 - WPe^{-(s\mu - \lambda)t} \quad (14)$$

If we analyse these formulas, we see that the parameter of the exponential distribution is $s\mu - \lambda$, which is the idleness rate of the servers. Using the

idleness rates we computed in the previous section we can thus build an approximation for the service level.

$$\hat{P}_r[WT < t | WT > 0] = 1 - e^{-(s\mu - \lambda)t} \hat{P}_r[WT_i < t] = 1 - \hat{W} P_i e^{(\sum_{j \in G_i} I_j)t} \quad (15)$$

The approximation has been tested on the same data set as in the preceding sections. In order to test the validity of the conditional waiting probability approximation, we first present simulation results for the conditional service level. This is equivalent to testing formula (15). Figure (8 i. to v.) present the results for maximum waiting time of 5, 10, 25, 50 and 100 percent of the average service time.

As it may be observed, the results are particularly good for smaller maximum waiting times. There are some deviations at the higher ones (Cases iv and v) for the smallest probabilities. Again this corresponds to the heavily loaded systems: the waiting time is usually very high in these cases, resulting in a small proportion of calls answered within the proposed bounds.

In Figures (9 i. to iii.), we present the service level as it is approximated.

We observe that although the approximation is accurate in most cases, there are an several cases for which the approximation is of lesser quality. This is once again the more heavily loaded cases. A comparison of figures (8) and (9) reveals that most of the difference comes from the earlier approximation on the waiting probability. We should note however that the approximation tend to underestimate the service level compared to what the results observed by simulation.

7 Conclusion

In this paper a method was presented to approximate the most important performance measures of multi-class queueing systems based on equivalent loss systems. We successively developed approximations for the waiting

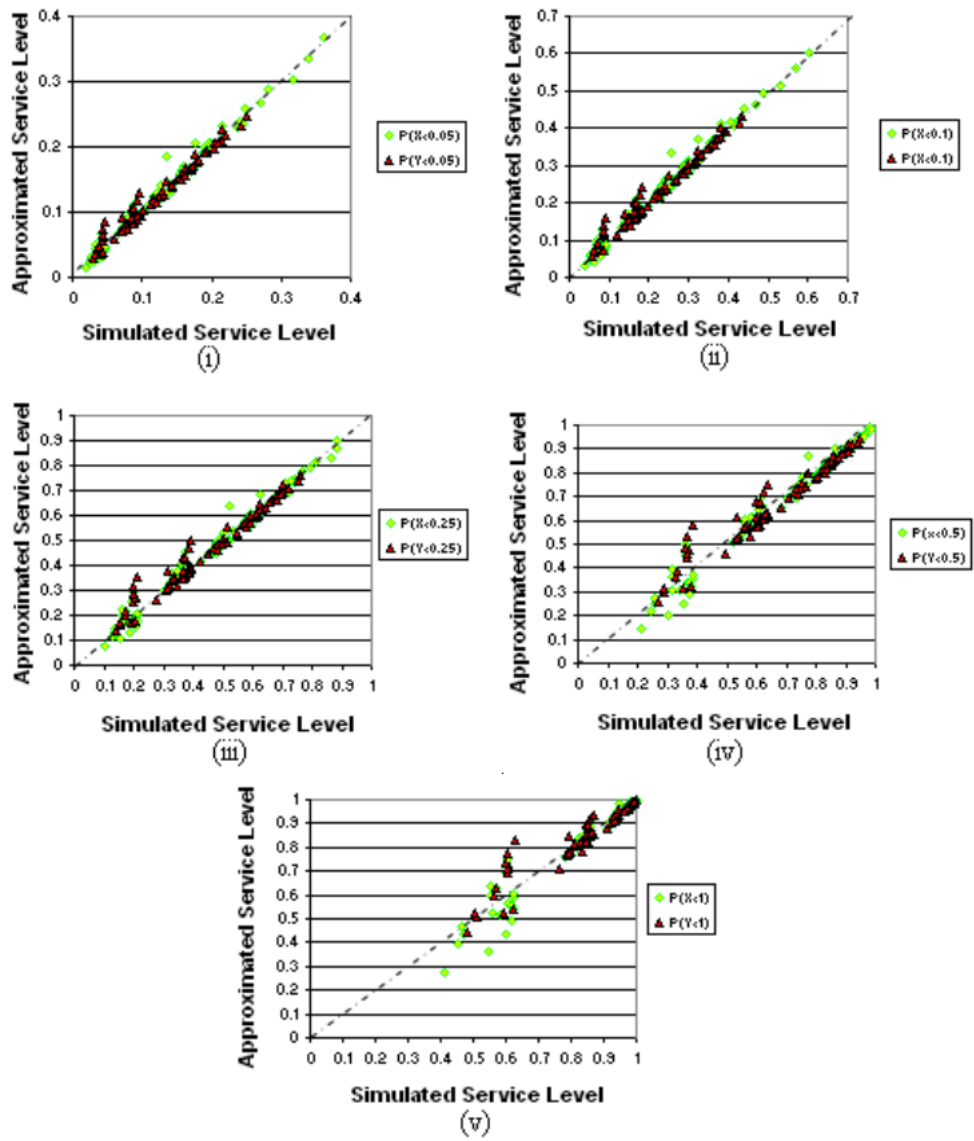


Figure 8: : Approximation of the conditional probability for five different maximum waiting times

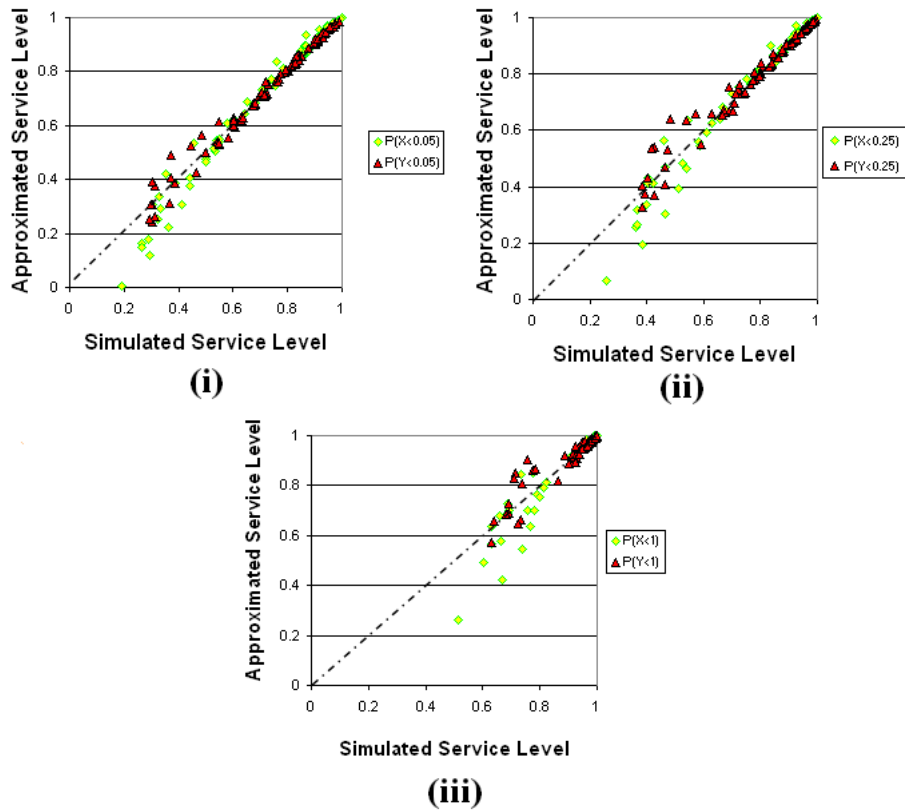


Figure 9: : Approximation of the service level for three different maximum waiting times

probability, the average waiting time and the service level. Our approximations were validated using a series of simulations. Along the way we presented some interesting insights into some similarities between queueing systems and equivalent loss systems that have to our knowledge never been reported in the literature.

The accuracy of our approximations is generally quite good, one should nevertheless be aware that the quality of the approximations degrades for heavily loaded systems and for longer waiting times. Although many call centers work close to saturation, which are cases where we observed some

deviation, the methods provide fairly good approximations even in these cases. More importantly, in terms of relative performance the approximations presented here perform particularly well. Another important point is that all methods are quite easy to compute. These observations make our method quite appropriate to be used in practice. The relative error of our method is well within the precision of the estimates that can often be obtained for the arrival rate or service rate.

There are many possible extensions to the work presented here. We really believe that the approximation methodology developed in this article could be applicable in many situations. Indeed, loss models seem in general to be easier to analyse than queueing models. It would be worthwhile investigating whether the same type of close relations can be exploited to develop an approximation for other complex queueing models.

In the context of call centers, we see the following possible extensions. First one should investigate whether the results presented here could be used in systems with limited queues and/or impatient customers. Secondly, one could investigate the method a step further for more complicated settings. In particular, we think about imposing less restrictions on the service time distributions.

References

- [Aksin et al., 2007] Aksin, Z., Armony, M., and Mehrotra, V. (2007). The Modern Call Center: A Multi-Disciplinary Perspective on Operations Management Research. *Production and Operations Management*, 16(6):665–688.
- [Avramidis et al., 2008] Avramidis, N., Chan, W., and L’Ecuyer, P. (2008). Staffing multi-skill call centers via search methods and a performance approximation. *IIE Transactions*.

- [Chevalier et al., 2005] Chevalier, P., Shumsky, R., and Tabordon, N. (2005). Routing and Staffing in Large Call Centers with Specialized and Fully Flexible Servers. Technical report, Université catholique de Louvain.
- [Chevalier and Tabordon, 2003] Chevalier, P. and Tabordon, N. (2003). Overflow analysis and cross-trained servers. *International Journal of Production Economics*, 85:47–60.
- [Chevalier and Van den Schrieck, 2006] Chevalier, P. and Van den Schrieck, J.-C. (2006). Optimizing the staffing and routing of small-size hierarchical call centers. *Production and Operations Management on Service Operations*.
- [Cooper, 1972] Cooper, R. B. (1972). *Introduction to Queueing Theory*. North Holland, 2nd edition.
- [Franx et al., 2006] Franx, G. J., Koole, G., and Pot, A. (2006). Approximating multi-skill blocking systems by HyperExponential Decomposition. *Performance Evaluation*, 63(8):799–824.
- [Fredericks, 1980] Fredericks, A. A. (1980). Congestion in Blocking Systems - A Simple Approximation Technique. *The Bell System Technical Journal*, 59(6):805–827.
- [Gans et al., 2003] Gans, N., Koole, G., and Mandelbaum, A. (2003). Telephone Call Centers: Tutorial, Review and Research Prospects. *MSOM*, 5(2):79–141.
- [Gross and Harris, 1998] Gross, D. and Harris, C. M. (1998). *Fundamentals of Queueing Theory*. John Wiley & Sons, INC., 3rd edition.
- [Hasija et al., 2008] Hasija, S., Pinker, E. J., and Shumsky, R. A. (2008). Call Center Outsourcing Contracts Under Information Asymmetry. *Management Science*, forthcoming.

- [Jagerman et al., 1997] Jagerman, D. L., Melamed, B., and Willinger, W. (1997). Stochastic Modeling of Traffic Processes. Technical Report 7, Rutgers Center for Operations Research (RUTCOR).
- [Khintchine, 1960] Khintchine, A. Y. (1960). *Mathematical methods in the theory of queueing*. Griffin.
- [Koole and Pot, 2006] Koole, G. and Pot, A. (2006). An overview of Routing and Staffing algorithms in Multi-Skill Contact Centers. Technical report, Departement of Stochastics, Vrije Universiteit Amsterdam.
- [Koole et al., 2003] Koole, G., Pot, A., and Talim, J. (2003). Routing heuristics for multi-skill call centers. In *Proceedings of the 2003 Winter Simulation Conference*, pages 1813–1816.
- [Mandelbaum and Reiman, 1996] Mandelbaum, A. and Reiman, M. I. (1996). On pooling in queueing networks. *Management Science*, 44:971–981.
- [Shumsky, 2004] Shumsky, R. A. (2004). Approximation and Analysis of a Queueing System with Flexible and Specialized Servers. *OR Spectrum*, 26(3):307–330.
- [Tabordon, 2002] Tabordon, N. (2002). *Modeling and Optimizing the Management of Operator Training in a Call Center*. PhD thesis, Institut D’Administration et de Gestion.
- [Wallace and Whitt, 2005] Wallace, R. B. and Whitt, W. (2005). A Staffing Algorithm for Call Centers with Skill-Based Routing. *MSOM*, 7(4):276–294.
- [Wilkinson, 1956] Wilkinson, R. (1956). Theories for toll traffic engineering in the u.s.a. *Bell System Technical Journal*, 35(2):421–514.