

# To Pool or Not to Pool in Call Centers

Nico M. van Dijk, Erik van der Sluis

Faculty of Economics and Business, University of Amsterdam, 1018 WB Amsterdam, The Netherlands  
 {n.m.vandijk@uva.nl, h.j.vandersluis@uva.nl}

Should service capacities (such as agent groups in call centers) be pooled or not? This paper will show that there is no single answer. For the simple but generic situation of two (strictly pooled or unpooled) server groups, it will provide (1) insights and approximate formulae, (2) numerical support, and (3) general conclusions for the waiting-time effect of pooling. For a single call type, this effect is clearly positive, as represented by a pooling factor. With multiple job types, however, the effect is determined by both a pooling and a mix factor. Due to the mix factor, this effect might even be negative. In this case, it is also numerically illustrated that an improvement of both the unpooled and the strictly pooled scenario can be achieved by simple overflow or threshold scenarios. The results are of both practical and theoretical interest: practical for awareness of this negative effect, the numerical orders, and practical scenarios in call centers, and theoretical for further research in more complex situations.

*Key words:* call centers; queueing; pooling; overflow; thresholds

*History:* Received: April 2005; Revised: January 2006; Accepted: May 2006 by Costis Maglaras.

## 1. Introduction

### 1.1. Motivation

Whether agents should be pooled or not is a question of general interest for call center management. The general perception seems to exist that pooling is always beneficial in terms of performance (waiting times) and agent capacity. Clearly, one large agent or call center group is more efficient than separate ones by the rationale of load balancing. This perception seems to be supported by simple queueing results. Nevertheless, the general validity of this perception seems to be false, as can be concluded from the literature (as specified in §1.3). However, no numerical results, general expressions, or conclusions for pooling appear to be reported for the simple case of two agent groups, which is of practical call center interest. Several questions on pooling that are of practical call center interest therefore remain open, such as:

1. Do these “counterintuitive” results also apply to larger server numbers, say, at realistic call center levels such as with 10, 20, 50, or more agents?
2. Are there simple expressions for pooling at a practical level?
3. Are there simple insights and conclusions?
4. And last but not least: can we do better?

### 1.2. Objectives

The objective of this paper therefore is to study and compare the pooled and unpooled scenarios as well as some other scenarios for the generic situation of two agent groups, so as to provide

1. insights and approximate formulae,
2. extensive numerical support, and
3. some general conclusions.

## 1.3. Results

**1.3.1. Insights and Formulae.** First, by means of an instructive example and by standard queueing results, one basic insight provided is that pooling is not necessarily beneficial. Next, approximate formulae will be developed for the general case of two agent groups.

These formulae rely on no more than the Pollaczek-Khintchine’s (PK)-formula as an approximation and a (so-called) pooling factor for doubling two standard exponential multiserver ( $M/M/s$ ) queues. Although rather straightforward, these “approximate” formulae are detailed later in this paper. The formulae show that the effect of pooling essentially (Result 3.1) factorizes in:

- a pooling factor (for pooling exponential queues);
- a mix factor (for mixing different services).

The pooling factor can be approximated by a simple analytic function (Result 3.2; see §3.2) for improving the mean waiting time when mixing does not take place. The mix factor, in turn, which can be computed directly from the call characteristics, may lead to an effect in the opposite direction. This factor also shows that the effect of pooling is insensitive for the actual call distributions when similar call distributions are mixed (Result 3.3).

**1.3.2. Numerical and Practical Support.** To illustrate that the simple insights provided (as based on “theoretical” queueing results and a simple example) also apply at the realistic call center level, numerical support is provided (with agent numbers of 5, 10, 20, or 50). Results are shown for performance (mean

waiting time) improvements, capacity gains, and sensitivity effects.

For exactness, these numerical results, despite the accuracy of the approximate formulae, are obtained by simulation. Such numerical results, other than for single-server examples, seem to be unreported.

**1.3.3. Main Conclusions.** The main conclusions are as follows:

1. Pooling is indeed beneficial for the situation of a single call type (with gains of over 50% for mean waiting times).
2. With different call types, in contrast, the effect of pooling can be expressed in a pooling and a mix factor. The mix factor will have an opposite effect. A trade-off may take place.
3. In this case, pooling is far less advantageous and might not even be profitable at all. Its effect depends on both the arrival and service mix ratios and traffic loads. In a practical situation, this effect is to be determined numerically (by approximate expressions, by numerical computation, or simulation).
4. The pooling effect is (approximately) insensitive for service distributional forms.
5. More sophisticated scenarios, such as threshold policies with overflow and prioritization by call type, may prove more superior than both the pooled and unpooled scenarios.

## 1.4. Literature

A first formal treatment on pooling for the general multiserver exponential ( $M/M/s$ ) case can be found in Smith and Whitt (1981) and in Wolff (1989). In this case it is shown that pooling always leads to a mean delay reduction. Both references also present a counterintuitive example for the simple case of a single server. These single-server examples illustrate an opposite effect of pooling for the average waiting time when different services are involved, as was also recently readdressed in Cattani and Schmidt (2005). A discussion on such results can also be found in Rothkopf and Rech (1987). Pooling is elegantly addressed from a psychological point of view in Larson (1987).

More recently, Mandelbaum and Reiman (1998) consider the consequences of pooling in a general setting of exponential queueing networks. Based on Pollaczek-Khinchine's formula (as will also be used extensively in the present paper), the effect is characterized by a utilization and a variability index (related to the pooling and mix factor in this paper), which indicates that pooling is not necessarily beneficial. General results are concluded for both light and heavy traffic situations and a variety of configurations. However, the simple generic structure of main interest in the present paper (that is, of parallel

groups of servers) is only briefly dealt with in this reference (see §5.2), with the present situation of mixed services (heterogeneous servers referred to in §5.3) without explicit expression. Numerical results are not reported.

Most recently, in Wallace and Whitt (2005), in the setting of skill-based routing, resource pooling is used to minimize staffing capacities. Based on the well-known square-root relationship for capacity computations, their findings show that agents should limit the multiskill functionalities, perhaps to only two skills. This result is also in line with a main conclusion from the present paper: *to be aware of and avoid (too much) mix variability by pooling*.

In Borst et al. (2004) this square-root staffing principle is also exploited and extended in a number of directions for dimensioning large call centers. Capacity savings by pooling (such as of two agents groups) for exponential queues with one type of call are hereby easily concluded. In Gans et al. (2003) a figure is provided by which the effect of scaling (and thus also pooling) can be provided for different performance targets but again for a single call type. The latter surveying reference includes a long list of queueing references on call centers, multiskill functionalities, or overflow somewhat related to pooling (e.g., Stanford and Grassmann 1993, Borst and Seri 1997, Chevalier and Tabordon 2003). None of these references, however, provides numerical results for the quantitative consequences of pooling two different call or agent groups. In a recent survey by Aksin et al. (2007) some other primarily nonquantitative reflections, such as more flexible service on pooling in call centers, have briefly been touched upon. The recent results in Sisselman and Whitt (2007) show other directions of interest for selective routing and pooling so as to maximize a total call preference value.

## 1.5. Outline

First, in §2 an instructive example of two parallel queues is presented. Next, in §3 approximate formulae will be developed for the effect of pooling two agent groups. Section 4 provides numerical support, which leads to a number of general conclusions. Finally, in §5, the instructive example is revisited to argue further improvements by more sophisticated overflow scenarios.

## 2. An Instructive Example

### 2.1. Two First Basic Queueing Insights

Pooling two separate queues is generally perceived to be efficient. Indeed, pooling two identical exponential servers leads to a mean delay reduction of roughly 50%. More precisely, with  $W_p$  and  $W_A$  the mean waiting time for the pooled and unpooled case,

respectively, and  $\rho = \lambda/\mu$  the traffic load per server, by straightforward calculations from standard  $M/M/s$  expressions, pooling two parallel exponential servers would lead to a reduction factor of at least 50% (since  $\rho < 1$ ) for the mean waiting time as follows:

$$\frac{W_p}{W_A} = \frac{\tau\rho^2/(1-\rho^2)}{\tau\rho/(1-\rho)} = \frac{\rho}{1+\rho}. \tag{2.1}$$

This result, however, relies on the implicit assumption of identical servers or, rather, service characteristics—most notably, identical means. However, when different services are involved, the advantage of pooling becomes questionable. Here, a second basic queueing result is to be realized, a result that seems hardly realized in practical call center environments. This result is known as the PK-formula (e.g., Cooper 1981) and is only exact for the single-server case. It expresses the effect of service variability by

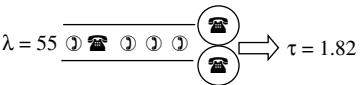
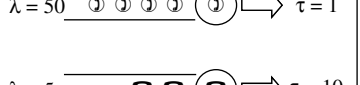
$$W_G = (1/2)(1+c^2)W_E \quad \text{with} \quad c^2 = \sigma^2/\tau^2, \quad \text{and}$$

$W_G$ : the expected mean under a general service distribution  $G$  (and  $E$  representing the exponential case) with mean  $\tau$  and standard deviation  $\sigma$ . (2.2)

By pooling two separate servers, one dedicated to services (calls) of type 1 and one dedicated to services (calls) of type 2, and assuming that each server can handle both types of services (calls), extra service variability will be brought in as a next service request at one, and the same server can then be either of type 1 or 2. By regarding the pooled service system as a server that is twice as fast as one separate server, or rather by assuming that the PK-formula also applies (approximately) for a two-server system, by virtue of the PK-formula this extra variability will lead to an effect in the opposite direction and may even lead to an increase of the mean delay (and waiting time).

**2.1.1. A Numerical Example.** In Figure 1 the situation of two job (call) types 1 and 2 with mean service (call) durations  $\tau_1 = 1$  and  $\tau_2 = 10$  minutes but arrival rates  $\lambda_1 = 50$  and  $\lambda_2 = 5$ , respectively, is illustrated. (In what follows, we will refer to this situation as by a mix ratio  $k = 10$ .) The traffic load  $\rho = \lambda_1\tau_1 = \lambda_2\tau_2$  is set at 83%, so that both call types bring in an equal workload.

Figure 1 Two-Server Example ( $k = 10$ ;  $\rho = 0.83$ )

| Pooled system   | $W_A = 6.15$ | Unpooled system  | $W_A = 4.55$                 |
|---|--------------|--|------------------------------|
|  | $W_p = 6.15$ |  | $W_1 = 2.50$<br>$W_2 = 25.0$ |

The results show that the unpooled case is still more efficient, at least for the average waiting time and particularly for the mean waiting time of type 1 calls, the vast majority (91%) of all calls. Due to the variability, it thus seems preferable to keep the servers separate despite the inefficiency, since a server might be idle when there is still a call waiting (at the other server).

**2.2. Balanced Case: Mix Ratio and Coefficient**

More generally, consider two types of calls with arrival rate  $\lambda_i$  and deterministic mean service time  $\tau_i$  for type  $i$  ( $i = 1, 2$ ). Let the mix ratio  $k$  be defined by  $\lambda_1 = k\lambda_2$  and  $\tau_2 = k\tau_1$ , so that the workloads  $\rho_i = \lambda_i\tau_i$  are assumed to be the same (in the example,  $k = 10$ ).

Throughout the paper,  $\bar{\tau}$  and  $c_{mix}^2$  denote the mean service time and mix coefficient for the pooled case, respectively, as computed by

$$\bar{\tau} = p_1\tau_1 + p_2\tau_2 \quad \text{with} \quad p_i = \lambda_i/(\lambda_1 + \lambda_2), \quad i = 1, 2,$$

$$c_{mix}^2 = \frac{p_1(\tau_1 - \bar{\tau})^2 + p_2(\tau_2 - \bar{\tau})^2}{\bar{\tau}^2}$$

$$= p_1\left(\frac{\tau_1}{\bar{\tau}}\right)^2 + p_2\left(\frac{\tau_2}{\bar{\tau}}\right)^2 - 1. \tag{2.3}$$

Now let  $W_i$  be the mean waiting time for type  $i$ ,  $W_A$  ( $=p_1W_1 + p_2W_2$ ) the average mean waiting time in the unpooled case,  $W_p$  the mean waiting time for the pooled case, and  $W_E(s, \rho, \tau)$  the mean waiting time for an exponential server group with  $s$  servers, traffic load  $\rho$  per server, and mean service time  $\tau$ .

Then, by straightforward calculations, the following expressions can be obtained by the standard  $M/M/1$  and  $M/M/2$  expressions and using the PK-formula for both the unpooled case (which is exact) and the pooled case (as an approximation). (This approximation will be shown to be fairly accurate in §3.1 and is indicated by the notation  $\triangleq$ . The expressions will be generalized in §3.) With

$$c_{mix}^2 = \frac{k}{k+1} \left(\frac{k+1}{2k}\right)^2 + \frac{1}{k+1} \left(\frac{k+1}{2}\right)^2 - 1 = \frac{(k-1)^2}{4k}$$

$$\frac{W_p}{W_A} \triangleq \frac{(1/2)(1+c_{mix}^2)W_E(2, \rho, \bar{\tau})}{(1/2)W_E(1, \rho, \bar{\tau})}$$

$$= (1+c_{mix}^2) \left[ \frac{\rho}{1+\rho} \right] = \frac{(k+1)^2}{4k} \left[ \frac{\rho}{1+\rho} \right] \tag{2.4}$$

$$\begin{aligned} \frac{W_p}{W_1} &\triangleq \frac{(1/2)(1 + c_{\text{mix}}^2)W_E(2, \rho, \bar{\tau})}{(1/2)W_E(1, \rho, \tau_1)} \\ &= (1 + c_{\text{mix}}^2) \frac{2k}{k+1} \left[ \frac{\rho}{1+\rho} \right] \\ &= (1/2)(k+1) \left[ \frac{\rho}{1+\rho} \right]. \end{aligned} \quad (2.5)$$

From these expressions we can directly draw the following conclusions, which illustrate how the effect of pooling is determined by both the mix ratio and traffic load.

**CONCLUSION 2.1.** For the deterministic two-server case,

- (i) Pooling is necessarily beneficial for all types only for  $k \leq 3$ .
- (ii) There can be a possible increase for  $k/(k+1) \cdot 100\%$  of the calls for  $k > 3$ .
- (iii) Pooling is, on average, beneficial for a mix ratio  $k \leq 5$ , but not necessarily for  $k > 5$ .

### 2.3. Unbalanced Case (Unequal Traffic Loads)

As illustrated by the example above, in the balanced case with  $\rho = \rho_1 = \rho_2$ , the question as to whether pooling is advantageous comes down to a trade-off between the traffic load  $\rho$  and the mix ratio  $k$ . More generally, with possibly unequal traffic loads, it will be determined by

- the traffic loads  $\rho_1$  and  $\rho_2$ ,
- the arrival ratio  $k_a = \lambda_1/\lambda_2$ , and
- the service ratio  $k_s = \tau_2/\tau_1$ .

For the two-server example, again by straightforward expressions for the  $M/M/1$  and  $M/M/2$  and using the PK-formula (as an approximation in the  $M/G/2$  case), this can be expressed more explicitly by the following expression. (Herein, to let the dependencies on the different traffic loads and the ratios come out most explicitly, service times are assumed to be deterministic.)

Clearly, trade-off conclusions as given above for the balanced case can hereby be determined easily. The average waiting time in the pooled and unpooled cases becomes

$$\begin{aligned} W_p &\triangleq (1/2)(1 + c_{\text{mix}}^2)W_E(2, \bar{\rho}, \bar{\tau}) \\ &= (1/2)(1 + c_{\text{mix}}^2) \frac{\bar{\tau}\bar{\rho}^2}{1 - \bar{\rho}^2} \\ W_A &= \left[ \frac{k_a}{(1 + k_s/k_a - 2\bar{\rho})} + \frac{k_s}{(1 + k_a/k_s - 2\bar{\rho})} \right] \frac{\bar{\tau}\bar{\rho}}{(k_s + k_a)} \end{aligned}$$

where

$$c_{\text{mix}}^2 = \frac{k_a(k_s - 1)^2}{(k_a + k_s)^2}, \quad \bar{\tau} = \frac{(k_a + k_s)}{(k_a + 1)}\tau_1, \quad \bar{\rho} = \frac{(k_a + k_s)}{2k_a}\rho_1. \quad (2.6)$$

## 3. Approximate Formulae

In line with the instructive example from §2, in this section we will develop approximate formula for the effect of pooling server groups of arbitrary size. These formulae are primarily meant to be indicative rather than accurate to provide global insights as well as orders of magnitude for practical call center engineering.

First, the PK-formula is argued as a reasonable approximation for larger server numbers. Next, simple approximate expressions are developed. These basically show that the effect of pooling can be factorized in two simple factors: one factor as if for identical services and one factor purely based on different service characteristics. In addition, some (approximate) insensitivity results are concluded. First, the more representative balanced case is dealt with in §3.2. Next, the general case with arbitrary parameters is covered in §3.3.

### 3.1. PK-Formula for Larger Groups

For the instructive two-server example, an expression was obtained by using the Pollaczek-Khintchine formula as an approximation for the pooled case to capture the effect of mixing different services. For larger server numbers, say, with  $s = 5, 10, 20, 50$ , or even 100 agents, this seems reasonable.

For example, the approximate formula (3.1) below for the mean waiting time of  $M/G/c$  queues (see Cosmetatos 1976) is known to be fairly accurate, in the order of a few percentages, for a wide spectrum of natural situations, for example, with  $\rho = 0.7, 0.8$ , or  $0.9$ , and  $0 \leq c^2 \leq 2$  (e.g., see Tijms 1994, pp. 297–300):

$$\begin{aligned} W_G &= [(1 - c^2)\mathbf{F} + c^2]W_E \quad \text{with} \\ \mathbf{F} &= (1/2) + \frac{(s-1)(1-\rho)(\sqrt{4+5s}-2)}{s\rho 32}. \end{aligned} \quad (3.1)$$

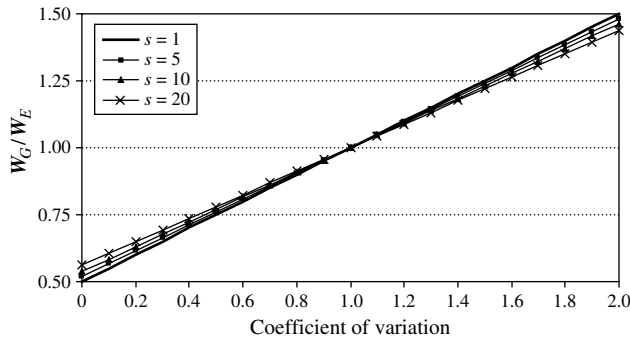
This approximation is illustrated in Figure 2, which shows the deviation from the PK-formula by the straight line  $(1/2)(1 + c^2)$  for up to  $s = 20$ . To obtain approximate and indicative expressions for the effect of pooling, we will also use the PK-formula  $W_G = (1/2)(1 + c^2)W_E$  when  $s$  is a larger number.

### 3.2. Balanced Case: Approximate Results

Let us first consider the situation of two agent groups  $i$  ( $i = 1, 2$ ) with an equal number of agents  $s$ , different call types 1 and 2 at agent groups 1 and 2, and for type  $i$  the characteristics

- $\lambda_i$ : arrival rate,
- $G_i$ : service distribution,
- $\tau_i, \sigma_i^2$ : mean service time and variance,
- $c_i^2 = \sigma_i^2/\tau_i^2$ : squared coefficient of variation, and
- $\rho_i = \lambda_i\tau_i$ : the traffic loads, assuming  $\rho_1 = \rho_2 = \rho$ .

**Figure 2** PK-Formula for Different Group Sizes  $s$  and  $c^2$



To study the option of pooling, we also implicitly assume that any agent can handle any type of call. (Recall that  $\bar{\tau} = p_1\tau_1 + p_2\tau_2$ .) As argued in §3.1, for both the pooled and unpooled case we use the PK-formula (as an approximation) to conclude

$$W_P \triangleq (1/2)(1 + c_{\text{pooled}}^2)W_E(2s, \rho, \bar{\tau}) \quad \text{with} \quad (3.2)$$

$$c_{\text{pooled}}^2 = p_1 \left(\frac{\tau_1}{\bar{\tau}}\right)^2 c_1^2 + p_2 \left(\frac{\tau_2}{\bar{\tau}}\right)^2 c_2^2 + c_{\text{mix}}^2 \quad (3.3)$$

$$W_A \triangleq p_1(1/2)(1 + c_1^2)W_E(s, \rho_1, \tau_1) + p_2(1/2)(1 + c_2^2)W_E(s, \rho_2, \tau_2). \quad (3.4)$$

Here,  $c_{\text{pooled}}^2$  represents the squared coefficient of variation for the pooled case as obtained by the standard variance relation for conditional expectations. Result 3.1 below then directly follows from combining (3.2) and rewriting (3.4) for the unpooled case (by scaling  $\tau_i$  to  $\bar{\tau}$  while assuming that the traffic loads are kept the same) as:

$$\begin{aligned} W_A &\triangleq p_1(1/2)(1 + c_1^2) \frac{\tau_1}{\bar{\tau}} W_E(s, \rho, \bar{\tau}) \\ &\quad + p_2(1/2)(1 + c_2^2) \frac{\tau_2}{\bar{\tau}} W_E(s, \rho, \bar{\tau}) \\ &= (1/2) \left[ 1 + p_1 \frac{\tau_1}{\bar{\tau}} c_1^2 + p_2 \frac{\tau_2}{\bar{\tau}} c_2^2 \right] W_E(s, \rho, \bar{\tau}) \\ &= (1/2) [1 + (1/2)c_1^2 + (1/2)c_2^2] W_E(s, \rho, \bar{\tau}). \end{aligned} \quad (3.5)$$

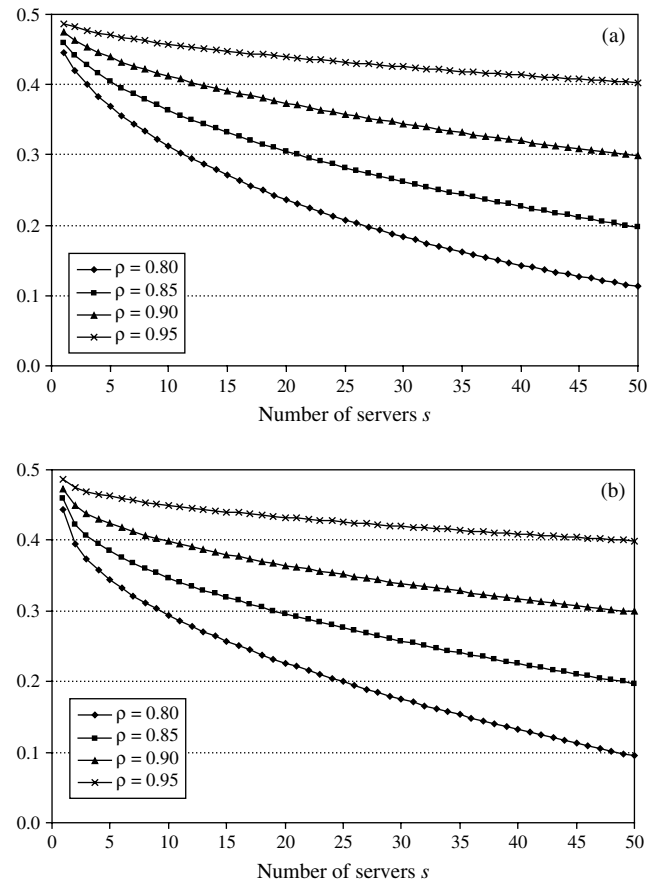
**RESULT 3.1 (POOLING EFFECT).** With  $P(2s, G_1, G_2)$ , the effect of pooling two equal agent groups with call distributions  $G_1$  and  $G_2$  is as follows:

$$\begin{aligned} P(2s, G_1, G_2) &\equiv \frac{W_P}{W_A} \triangleq \left[ \frac{W_E(2s, \rho, \bar{\tau})}{W_E(s, \rho, \bar{\tau})} \right] \\ &\quad \cdot \frac{[1 + p_1(\tau_1/\bar{\tau})^2 c_1^2 + p_2(\tau_2/\bar{\tau})^2 c_2^2 + c_{\text{mix}}^2]}{[1 + (1/2)c_1^2 + (1/2)c_2^2]}. \end{aligned} \quad (3.6)$$

**CONCLUSION 3.1.** The effect of pooling is determined by two factors:

- a pooling factor (dependent on the size of the group and the traffic load  $\rho$ ), and

**Figure 3** (a) Pooling Factor  $P(2s, \rho)$ ; (b) Approximate Formula (3.7)



- a mix factor (dependent on the mix ratio and the squared coefficient of variation).

Clearly, the pooling factor can be computed by standard analytical expressions for  $M/M/s$  queues. Also, an analytic approximation has been found that appears to fit the pooling factor reasonably well, as illustrated in Figure 3 and given in Result 3.2 below, which indicates the role of both the traffic load  $\rho$  and the number of agents  $2s$ .

**RESULT 3.2 (POOLING FACTOR).**

$$\begin{aligned} P(2s, \rho) &= \frac{W_E(2s, \rho, \tau)}{W_E(s, \rho, \tau)} \\ &\approx \left[ \frac{\rho}{(1 + \rho)} - 1/4(1 - \rho)\sqrt{s - 1} \right]^+. \end{aligned} \quad (3.7)$$

As for the mix factor, the following result is of interest, which can be obtained directly by combining Equations (3.2), (3.4), and (3.6). It expresses the effect of pooling, when the service distributional forms, or rather the squared coefficients of variation  $c_1^2$  and  $c_2^2$ , are the same. In this case, the effect equals that as for the exponential case, expressed by merely the pooling and mix factors. It can be regarded accordingly as an insensitivity result with respect to the service distributional forms.

RESULT 3.3 (INSENSITIVITY RESULTS). With  $c_1^2=c_2^2$ :

- (1)  $\tau_1 = \tau_2$ :  $\mathbf{P}(2s, G_1, G_2) \triangleq \mathbf{P}(2s, \rho)$ ;
- (2)  $\tau_1 \neq \tau_2$ :  $\mathbf{P}(2s, G_1, G_2) \triangleq (1 + c_{\text{mix}}^2)\mathbf{P}(2s, \rho)$ .

As a special corollary and similar to expression (2.5), Result 3.3 results to the following pooling effect for type 1 calls; that is, for  $k/(k+1)100\%$  of all calls:

COROLLARY 3.4 (TYPE 1 CALLS). With  $c_1^2 = c_2^2$ :

$$\frac{W_p}{W_1} \triangleq (1 + c_{\text{mix}}^2) \frac{2k}{k+1} \mathbf{P}(2s, \rho) = (1/2)(k+1)\mathbf{P}(2s, \rho). \quad (3.8)$$

### 3.3. Unbalanced Case

As in §2.3, for the instructive example the expressions (3.2) and (3.4) can be extended to the more general case with unequal traffic loads and unequal server numbers, which illustrates the effect of pooling and its dependence on

- the server numbers  $s_1$  and  $s_2$  at groups 1 and 2;
- the arrival and service mix ratios  $k_a = \lambda_1/\lambda_2$  and  $k_s = \tau_2/\tau_1$ ;
- the service and mix variabilities as expressed by  $c_1^2$ ,  $c_2^2$ , and  $c_{\text{mix}}^2$ ; and
- the traffic loads  $\rho_1$  and  $\rho_2$  and the average traffic load  $\bar{\rho} = (s_1\rho_1 + s_2\rho_2)/(s_1 + s_2)$ .

With  $c_{\text{pooled}}^2$  defined as in (3.3), and again by rescaling  $\tau_1$  and  $\tau_2$  to the average service time  $\bar{\tau} (=p_1\tau_1 + p_2\tau_2)$ , the effect of pooling an  $s_1$ -server group with service distribution  $G_1$  and an  $s_2$ -server group with service distribution  $G_2$  can then be expressed (by also approximating an  $M/M/s$  service system by a single server that is  $s$  times faster) by the approximate expression

$$\begin{aligned} & \mathbf{P}(s_1 + s_2, G_1, G_2) \\ & \triangleq \frac{(1 + c_{\text{pooled}}^2)W_E(s_1 + s_2, \bar{\rho}, \bar{\tau})}{p_1(1 + c_1^2)W_E(s_1, \rho_1, \tau_1) + p_2(1 + c_2^2)W_E(s_2, \rho_2, \tau_2)} \\ & \approx (1 + c_{\text{pooled}}^2) \left[ \frac{2\bar{\rho}^3}{1 - \bar{\rho}^2} \right] \\ & \cdot \left[ (1 + c_1^2) \frac{\rho_1^2}{(1 - \rho_1)} + (1 + c_2^2) \frac{\rho_2^2}{(1 - \rho_2)} \right]^{-1}. \quad (3.9) \end{aligned}$$

This expression can still be regarded as a factorization in a pooling factor, which depends only on the traffic loads and server numbers, and a mix factor  $(1 + c_{\text{pooled}}^2)$ , which depends purely on the mix ratios and the service variabilities.

## 4. Numerical Results and Conclusions

In this section, in line with §§2 and 3, we will provide some numerical support and conclusions for the question of pooling two agent groups. In the case of identical calls, or calls for which the mean and variance are more or less equal, pooling is indeed beneficial

(regardless of the call center sizes and regardless of whether the calls are exponential or not), as expressed by the pooling factor  $\mathbf{P}(2s, \rho)$ .

We will therefore restrict the presentation to the more practical situation of different call types. A variety of different situations and aspects can be investigated such as with different agent numbers, traffic loads, mixtures of call types, and different call types (such as deterministic, exponential, or with other squared coefficients of variation). We will limit the presentation to the most generic case of two equally sized agent groups each with one type of call. For accuracy, from hereon all numerical results are obtained by simulation. (One may note that Erlang-C is no longer sufficient and only approximate computations can be executed as in Equation (3.6).)

### 4.1. Balanced Case

As illustrated and argued in §§2 and 3, even with unchanged capacities the effect of pooling is no longer obvious and depends on the mix variability introduced by pooling, as made explicit by expressions (3.6) and (3.9). Here, we recall the mix ratio  $k$  (with  $\lambda_1 = k\lambda_2$  and  $\tau_2 = k\tau_1$ ) and implicitly assume that each agent can handle a call of either type.

**4.1.1. Mean Waiting Times.** In Figure 4, a numerical illustration is provided for the effect of pooling on average waiting times. In Figure 4(a) we study the situation for different traffic loads (for the exponential case). (One may recall the approximate insensitivity Result 3.3.) In Figure 4(b) we also study the effect for different values of the mix ratio  $k$ . Let  $\mathbf{P}$  denote the number  $s$  at which pooling becomes beneficial for the mean waiting time. The results show that the pooling point  $\mathbf{P}$  might not be or just barely be reached in realistic situations, such as with a  $\rho$  of 80% to 90% and  $s$  between 5 and 20.

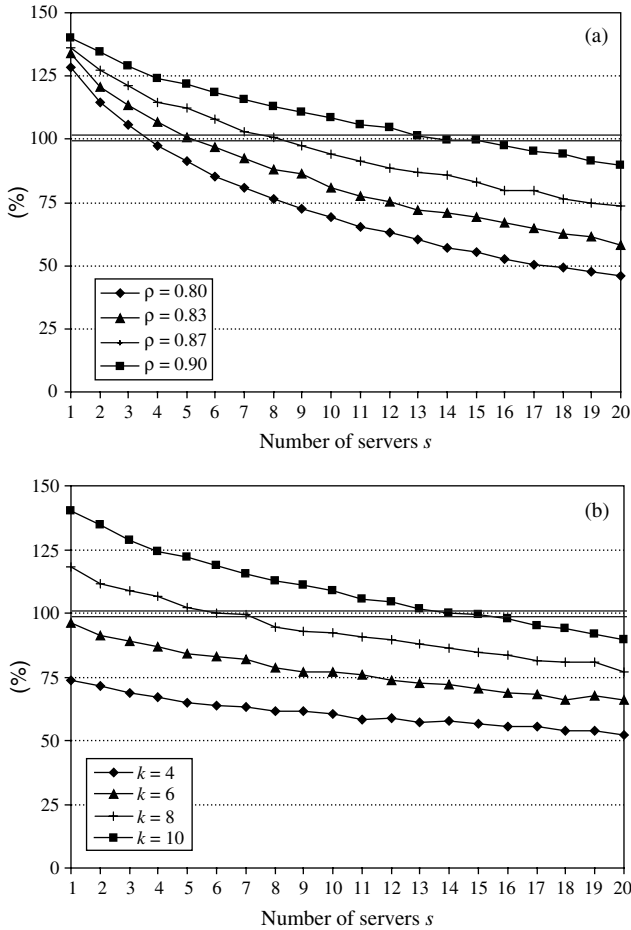
**CONCLUSION 4.1.** Pooling two agent groups of size  $s$  with unequal call types is not beneficial for agent numbers  $s < \mathbf{P}$  and

- $\mathbf{P} \uparrow \rho$
- $\mathbf{P} \uparrow k$ .

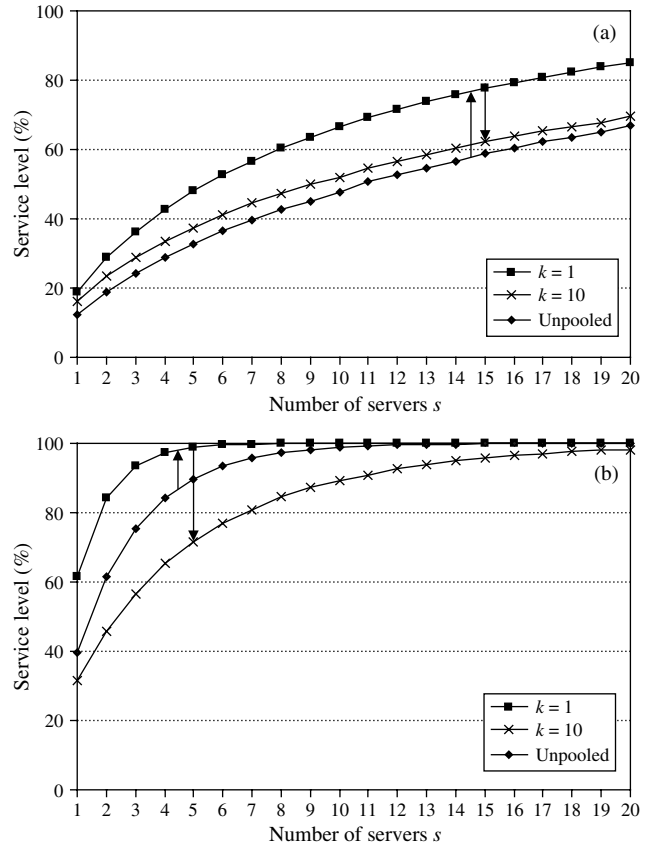
**REMARK 4.1.** In practical call center situations, one often determines the capacity based on a fixed waiting time target, for example, a mean waiting time or service level (as in the next section). In that case, the actual pooling point  $\mathbf{P}$  will increase (shift to the right).

**4.1.2. Service Levels.** Rather than mean waiting times, service levels can also be considered. Let  $\mathbf{t}$  be the threshold level for which the service level is computed. For standard service levels  $\mathbf{t} = \tau/4$ , as shown in Figure 5(a), pooling can be advantageous if the call types are equal (upper curve in Figure 5(a)). However, as illustrated in Figure 5(a) by the middle curve, the

**Figure 4** (a)  $W_p/W_A$  for Different Levels of  $\rho$  ( $k = 10$  and Exponential Call Durations), and (b)  $W_p/W_A$  for Different Mix Ratios  $k$  ( $\rho = 0.9$  and Exponential Call Durations)



**Figure 5** Service Levels for Unpooled Case and Pooled Case as if One Type ( $k = 1$ ) and Effectively ( $k = 10$ ) with (a)  $t = \tau/4$  and (b)  $t = 4\tau$



mix variability leads to a substantial quality reduction. For example, for  $s = 15$ , the service level of 60% for separate agents groups is shown to be increased by pooling to almost 80% when the calls are identical ( $k = 1$ ). However, the service level drops down to nearly 60% again (in the case of  $k = 10$ ) due to mixing calls with different characteristics. For threshold levels substantially larger—for example,  $t = 4\tau$ —pooling may even become negative, as shown in Figure 5(b). For example, for  $s = 5$ , instead of an increase from 89% to 99%, the service level decreases (in the case of  $k = 10$ ) to 71%.

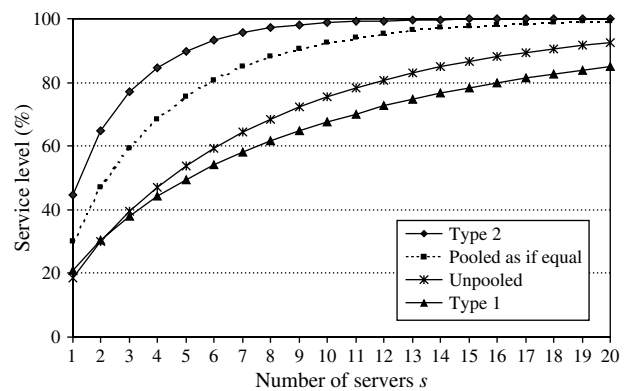
**CONCLUSION 4.2.** *The pooling effect for service levels of unequal call types can be substantially less than generally computed by M/M/s (Erlang-C or Workforce Management) calculations. The effect still seems generally beneficial for standard threshold levels  $t$  but could become negative for larger threshold levels.*

Furthermore, each call type might have its own service level. In Figure 6 the opposite effect on the service level for either types is illustrated with threshold

levels of  $t_i = \tau_i$  for both types. If the calls had been identical, then the service levels would increase up to the dotted curve in Figure 6 (this holds for both types, as  $\rho_1 = \rho_2$ ). However, due to mixing calls with different characteristics, the effect for type 1 becomes negative (lower curve). In contrast, the service levels for type 2 improve even more (upper curve).

**4.1.3. Type 1 Calls.** As expressed by Result 3.2 and illustrated in the two-server example, even

**Figure 6** Service Levels for Types 1 and 2 Before and After Pooling ( $t_i = \tau_i$ )



**Table 1**  $W_p/W_i$  for Different Mix Ratios  $k$  and Traffic Loads  $\rho$

| $s$ | $k = 4$      |              | $k = 6$      |              | $k = 8$      |              | $k = 10$     |              |
|-----|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|     | $\rho = 0.8$ | $\rho = 0.9$ | $\rho = 0.8$ | $\rho = 0.9$ | $\rho = 0.8$ | $\rho = 0.9$ | $\rho = 0.8$ | $\rho = 0.9$ |
| 1   | 1.11         | 1.18         | 1.56         | 1.66         | 2.00         | 2.13         | 2.44         | 2.61         |
| 2   | 0.99         | 1.12         | 1.38         | 1.57         | 1.78         | 2.02         | 2.17         | 2.47         |
| 3   | 0.93         | 1.10         | 1.31         | 1.53         | 1.68         | 1.97         | 2.06         | 2.41         |
| 4   | 0.89         | 1.08         | 1.25         | 1.51         | 1.61         | 1.94         | 1.97         | 2.37         |
| 5   | 0.86         | 1.06         | 1.21         | 1.48         | 1.55         | 1.91         | 1.89         | 2.33         |
| 10  | 0.74         | 1.00         | 1.03         | 1.40         | 1.33         | 1.79         | 1.62         | 2.19         |
| 15  | 0.64         | 0.95         | 0.90         | 1.33         | 1.16         | 1.71         | 1.42         | 2.09         |
| 20  | 0.57         | 0.91         | 0.79         | 1.28         | 1.02         | 1.64         | 1.25         | 2.01         |
| 25  | 0.50         | 0.88         | 0.70         | 1.23         | 0.90         | 1.58         | 1.10         | 1.93         |
| 30  | 0.44         | 0.85         | 0.61         | 1.19         | 0.79         | 1.53         | 0.96         | 1.86         |

though the overall effect of pooling might be positive, it remains to be realized whether it may lead to a (substantial) increase for the (majority) of customers with the shortest call duration. This is illustrated in Table 1. Pooling only appears to be advantageous for type 1 calls for small mix ratio  $k$  and for sufficiently large  $s$ . Otherwise, its effect can be negative (noted in gray) for type 1 calls (the majority).

**4.2. Unbalanced Case**

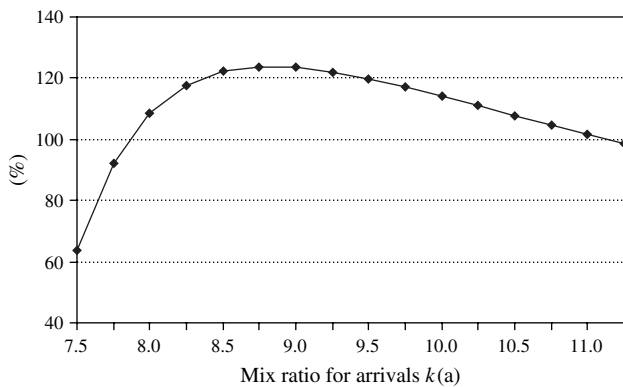
In Figure 7 below, for the simple case of two single servers with  $\rho_1 = 80\%$ ,  $k_s = 9$ , and by varying  $k_a$  from 7.5 to 11.5, hence with a traffic load for server 2 ranging from 96% to 64%, it is shown that the negative pooling effect is maximal for the balanced case ( $k_a = k_s$ ). Nevertheless, the data show that pooling remains unbeneficial over a wide range of arrival ratios (or traffic loads for server 2).

**5. Can We Do Better?**

**5.1. The Instructive Example Revisited**

So far, we have only considered the scenarios of fully pooled or strictly unpooled groups. However, other

**Figure 7**  $W_p/W_A$  for Different Arrival Ratios  $k_a$  for the Two-Server Example ( $\rho_1 = 0.8$  and  $k_s = 9$ )



scenarios can also be thought of to exploit the advantage of either scenario, such as:

- a minimum service variability as for the unpooled case, or
- a minimum idleness as for the pooled case.

To this end, let us reconsider the instructive example with two parallel servers, one for call type 1 and one for call type 2, with equal traffic loads  $\rho_1 = \rho_2 = 5/6$ , but a traffic ratio of  $k = 10$  (see §2). To avoid mix variability, the servers should still be kept devoted for type 1 and 2 calls. However, to avoid idleness, server 2 should also take a type 1 call if there is no type 2 call waiting and vice versa. The results for this two-way overflow system, as shown in Figure 8, indicate an improvement over both the pooled and the unpooled cases. (Note, however, that the unpooled case is still by far more efficient for type 1 calls.)

In this two-way overflow scenario there is no idleness at all (no server will be idle while there is still a call waiting). However, the overflow of type 2 calls may imply long disturbances (and thus a large variability) at server 1. It might thus be advantageous to allow some idleness at server 1. A one-way overflow can therefore be suggested in which type 1 calls may overflow while type 2 calls may not. This scenario not only shows a further reduction of the waiting time all over but also improves the unpooled scenario for type 1 (91%) calls. (The price to pay here is a waiting time increase for the small percentage of type 2 calls.)

**5.2. Priority and Overflow Scenario**

These simple overflow scenarios can be seen as extreme forms for avoiding idleness and providing more weight to short services. Based on these results, one might expect that further improvements can be achieved by allowing overflow at some specific threshold level (rather than only allowing servers to become idle) and prioritizing specific call types. A general threshold scenario  $\text{Thr}(\theta_1, \theta_2, \Omega_1, \Omega_2)$  can thus be defined as

$\text{Thr}(\theta_1, \theta_2, \Omega_1, \Omega_2)$ : when a server of server group  $j$  ( $j = 1, 2$ ) becomes available, it will give priority to a job of type  $i$ , where

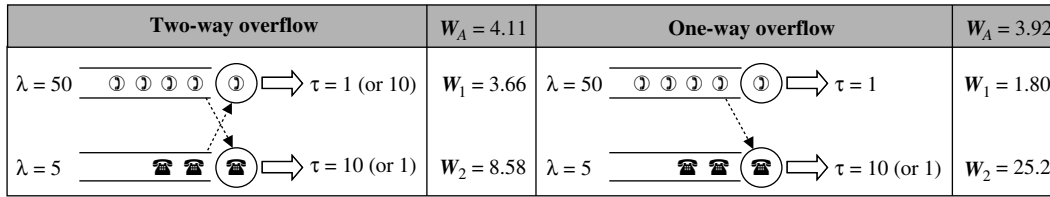
$$i = \begin{cases} 3-j & (m_{3-j} \geq \theta_{3-j} \wedge m_j < \theta_j) \vee (m_j = 0 \wedge m_{3-j} \geq \Omega_{3-j}) \\ j & \text{otherwise.} \end{cases}$$

By the  $\theta_i$ -values, calls are thus given priority in case their queue length becomes too large.

By the  $\Omega_i$ -values, overflow to an idling server is allowed when queues are long. When both call types have priority (both queues are long) or not (both queues are short), the server takes the next call from its own queue. Note that the simple two-way and one-way overflow scenarios as in the instructive example



Figure 8 Two-Server Example ( $k = 10; \rho = 0.83$ )



are covered by

- $\text{Thr}(\infty, \infty, 1, 1)$ : two-way overflow scenario.
- $\text{Thr}(\infty, \infty, 1, \infty)$ : one-way overflow scenario.

Among these dynamic (or queue length-dependent) scenarios  $\text{Thr}(\theta_1, \theta_2, \Omega_1, \Omega_2)$ , a scenario  $\text{Thr}(\text{Opt})$  is sought, which has a minimal average waiting time:

$$W_A[\text{Thr}(\text{Opt})] = \min_{\theta_1, \theta_2, \Omega_1, \Omega_2} W_A[\text{Thr}(\theta_1, \theta_2, \Omega_1, \Omega_2)]. \quad (5.1)$$

For the instructive example, this gives an optimal scenario with  $\theta_1 = 3, \theta_2 = \infty, \Omega_1 = \Omega_2 = 1, W_A = 2.95$ , and overflow percentages of 32% for type 1 and 33% for type 2. Table 2 summarizes the results of the five scenarios for the two-server example.

CONCLUSION 5.1. A substantial improvement over both the unpooled and pooled cases can be obtained by threshold policies with prioritization and overflow.

REMARK 5.1. Conclusion 5.1, or rather the minimal average waiting time as obtained by threshold policies, is in line with (and in fact extends) threshold policy results as shown in Bell and Williams (2001) and Harchol-Balter et al. (2005). Both references study a beneficiary-donor model with two parallel servers, with type 1 overflow at some threshold (thus without overflow by type 2 jobs) and for specific reasons: for dealing with high-traffic situations (Bell and Williams) and, somewhat related, for robustness purposes (Harchol-Balter et al. 2005). Furthermore, in neither of these references nor related references therein do numerical results seem to be reported for larger server numbers as in §5.3.

### 5.3. Larger Groups

The insights and results for the instructive two-server example also apply to realistic situations with a larger numbers of agents, say, up to  $2s = 100$  agents. The results are shown in Table 3.

Table 2 Summary of Five Scenarios for the Two-Server Example

|          | $W_1$ | $W_2$ | $W_A$ |
|----------|-------|-------|-------|
| Pooled   | 6.15  | 6.15  | 6.15  |
| Unpooled | 2.50  | 25.00 | 4.55  |
| Two-way  | 3.66  | 8.58  | 4.11  |
| One-way  | 1.80  | 25.20 | 3.92  |
| Thr(opt) | 2.24  | 10.03 | 2.95  |

CONCLUSION 5.2. Conclusion 5.1 also applies to server numbers of realistic call center order.

REMARK 5.2. Similar optimization results can be obtained for other performance targets, such as a service level, and for other weights for type 1 and 2 calls other than by their arrival ratios.

### 5.4. Case Example (Dutch AAA)

As another and real-life example for restricted overflow instead of pooling, the Dutch AAA (called ANWB) considered pooling (or virtualizing) four regional call centers into one (virtual) call center by instantaneous overflow upon waiting (see Table 4).

Instead of a 73% service level within 30 seconds when the call centers were kept separate, the overflow increased the level to 98%, where 60% of all calls were overflowed. Besides additional telecommunication costs, the overflowed calls were connected to “less-suited” agents from other regions with lesser knowledge of the specific region. Alternatively, by letting calls overflow after just 28 seconds of waiting (so that they could still be counted as successful within 30 seconds if an available agent in another region could be found), the service level was just slightly reduced to 96%. The overflow percentage, however, drastically dropped to 20%. Clearly, a variety of practical reasons (costs, skills, recognition, handling, responsibility) must be taken into account for trading off pooling.

Table 3 Simulation Results of the Three Variants for Pooling ( $k = 10; \rho = 0.9$ )

| s  | Pooled $W_p$ | Unpooled |       |       | Thr(Opt) |       |       | Thresholds |            |            |            |
|----|--------------|----------|-------|-------|----------|-------|-------|------------|------------|------------|------------|
|    |              | $W_1$    | $W_2$ | $W_A$ | $W_1$    | $W_2$ | $W_A$ | $\theta_1$ | $\theta_2$ | $\Omega_1$ | $\Omega_2$ |
| 1  | 11.53        | 4.49     | 45.24 | 8.18  | 2.75     | 20.56 | 4.37  | 3          | $\infty$   | 1          | 1          |
| 2  | 5.27         | 2.14     | 21.40 | 3.89  | 1.59     | 9.15  | 2.28  | 4          | $\infty$   | 1          | 1          |
| 3  | 3.33         | 1.38     | 14.15 | 2.51  | 1.10     | 5.68  | 1.52  | 4          | $\infty$   | 1          | 1          |
| 4  | 2.34         | 1.01     | 10.24 | 1.83  | 0.83     | 4.05  | 1.13  | 4          | $\infty$   | 1          | 1          |
| 5  | 1.76         | 0.78     | 7.57  | 1.41  | 0.67     | 3.02  | 0.88  | 5          | $\infty$   | 1          | 1          |
| 10 | 0.71         | 0.34     | 3.52  | 0.63  | 0.28     | 1.33  | 0.38  | 1          | $\infty$   | 1          | 1          |
| 15 | 0.40         | 0.21     | 2.15  | 0.38  | 0.15     | 0.87  | 0.21  | 1          | $\infty$   | 1          | 1          |
| 20 | 0.26         | 0.15     | 1.44  | 0.26  | 0.09     | 0.59  | 0.14  | 1          | $\infty$   | 1          | 1          |
| 30 | 0.13         | 0.08     | 0.81  | 0.15  | 0.05     | 0.32  | 0.07  | 1          | $\infty$   | 1          | 1          |
| 40 | 0.08         | 0.06     | 0.54  | 0.10  | 0.03     | 0.20  | 0.04  | 1          | $\infty$   | 1          | 1          |
| 50 | 0.05         | 0.04     | 0.38  | 0.07  | 0.02     | 0.13  | 0.03  | 1          | $\infty$   | 1          | 1          |

**Table 4** Three Scenarios for the Dutch AAA

|                         | Unpooled (%) | Pooled (%) | After 28 sec. (%) |
|-------------------------|--------------|------------|-------------------|
| Service level (30 sec.) | 73           | 98         | 96                |
| Percentage of overflow  | —            | 60         | 20                |

### 5.5. Evaluation

In practical call center environments, the general perception seems to exist that pooling agent groups is beneficial. This first intuition is correct for a single call type. With different call types, in contrast, opposite results, might apply because of mix variability. Indeed, by both approximate expressions and numerical results, it is shown that the effect of pooling (also for situations with realistic numbers of agents and traffic loads) may even turn out to be negative. *An awareness of its effect (as due to mixing different call characteristics) should therefore be present in call center practice.*

Alternatively, different threshold scenarios with prioritization and overflow may prove more efficient. Whether one should pool or not thus remains a question for which both theoretical and practical research remain required.

### Acknowledgments

The authors are grateful for the comments of the handling guest editor and the referees, which helped improve the presentation.

### References

Aksin, Z., M. Armory, V. Mehrotra. 2007. The modern call center: A multi-disciplinary perspective on operations management research. *Production Oper. Management* 16(6) 665–688.

Bell, S. L., R. J. Williams. 2001. Dynamic scheduling of a system with two parallel servers in heavy traffic with resource pooling: Asymptotic optimality of a threshold policy. *Ann. Appl. Probab.* 11(3) 608–649.

Borst, S. C., P. F. Seri. 1997. Robust algorithms for sharing agents with multiple skills. Technical Memorandum BL011212-970912-16TM, Bell Laboratories, Lucent Technologies, Murray Hill, NJ.

Borst, S. C., A. Mandelbaum, M. I. Reiman. 2004. Dimensioning large call centers. *Oper. Res.* 52(1) 17–34.

Cattani, K., G. M. Schmidt. 2005. The pooling principle. *INFORMS Trans. Ed.* 5(2). <http://ite.pubs.informs.org/Vol5No2/CattaniSchmidt/>.

Chevalier, P., N. Tabordon. 2003. Overflow analysis and cross-trained servers. *Internat. J. Production Res.* 85(1) 47–60.

Cooper, R. B. 1981. *Introduction to Queueing Theory*. North-Holland, Amsterdam.

Cosmetatos, G. P. 1976. Some approximate equilibrium results for the multi-server queue M/G/r. *Oper. Res. Quart.* 27(3) 615–620.

Gans, N., G. Koole, A. Mandelbaum. 2003. Telephone call centers: Tutorial, review and research prospects. *Manufacturing Service Oper. Management* 5(2) 79–141.

Harchol-Balter, M., T. Osogami, A. Scheller-Wolf. 2005. Robustness of threshold policies in beneficiary-donor model. *SIGMETRICS Performance Eval. Rev.* 33(2) 36–38.

Larson, R. C. 1987. Perspectives on queues: Social justice and the psychology of queueing. *Oper. Res.* 35(6) 895–905.

Mandelbaum, A., M. I. Reiman. 1998. On pooling in queueing networks. *Management Sci.* 44(7) 971–981.

Rothkopf, M. H., P. Rech. 1987. Perspectives on queues: Combining queues is not always beneficial. *Oper. Res.* 35(6) 906–909.

Sisselman, M. E., W. Whitt. 2007. Value-based routing and preference-based routing in customer contact centers. *Production Oper. Management* 16(3) 277–291.

Smith, D. R., W. Whitt. 1981. Resource sharing for efficiency in traffic systems. *Bell System Tech. J.* 60(1) 39–55.

Stanford, D. A., W. K. Grassmann. 1993. The bilingual server systems: A queueing model featuring fully and partially qualified servers. *INFOR* 31(4) 261–277.

Tijms, H. C. 1994. *Stochastic Models: An Algorithmic Approach*. Wiley, Chichester, UK.

Wallace, R. B., W. Whitt. 2005. A staffing algorithm for call centers with skill-based routing. *Manufacturing Service Oper. Management* 7(4) 276–294.

Wolff, R. W. 1989. *Stochastic Modelling and the Theory of Queues*. Prentice-Hall, Englewood Cliffs, NJ.