2156

■

Optimizing the staffing and routing of small-size
hierarchical call centers

Philippe Chevalier and Jean-Christophe Van den Schrieck

CORE
Voie du Roman Pays 34
B-1348 Louvain-la-Neuve, Belgium.
Tel (32 10) 47 43 04
Fax (32 10) 47 43 01
E-mail: corestat-library@uclouvain.be
http://www.uclouvain.be/en-44509.html

PRODUCTION AND OPERATIONS MANAGEMENT
Vol. 17, No. 3, May–June 2008, pp. 306–319
ISSN 1059-1478 | EISSN 1937-5956 | 08 | 1703 | 0306
DOI 10.3401/poms.1080.0033

Optimizing the Staffing and Routing of Small-Size Hierarchical Call Centers

Philippe Chevalier, Jean-Christophe Van den Schrieck
Louvain School of Management–Center for Operations Research and Econometrics, Université catholique de Louvain, 1348 Louvain-la-neuve, Belgium {philippe.chevalier@uclouvain.be, jc.vandenschrieck@uclouvain.be}

Multiple-skill call centers propagate rapidly with the development of telecommunications. An abundance of literature has already been published on call centers. Here, we want to focus on centers that would typically occur in business-to-business environments; these are call centers that handle many types of calls but where the arrival rate for each type is low. To find an optimal configuration, the integrality of the decision variables is a much more important issue than for larger call centers. The present paper proposes an approach that uses elements of combinatorial optimization to find optimal configurations. We develop an approximation method for the evaluation of the service performance. Next, we search for the minimum-cost configuration subject to service-level constraints using a branch-and-bound algorithm. What is at stake is to find the right balance between gains resulting from the economies of scale of pooling and the higher cost or cross-trained agents. The article shows that in most cases this method significantly decreases the staffing cost compared with configurations with only cross-trained or dedicated operators.

Key words: call centers; staffing; cross-training; branch and bound; loss model
History: Received: March 2005; Revised: December 2005 and January 2006; Accepted: February 2006 by Costis Maglaras.

1. Introduction

The development of telecommunications has been phenomenal in the last decade. The possibilities of new telecommunication technologies to provide services remotely have led to the development of many new business opportunities. Demand for such services has boomed in the last decade in the business-to-consumer (B2C) sector as well as in the business-to-business (B2B) sector.

An important way of providing these types of services is through call centers where clients call to obtain some kind of service (e.g., after sales, helpdesk, financial, ...). The growing importance of call centers for operations management in services has led to a growing interest in this topic for researchers. Almost all articles focus on the very large call centers, where there can be several hundreds of agents connected at any time. Given the economies of scale that arise by pooling the stochastic demand for services, large call centers are more frequently used in B2C environments, where such pooling is possible. In this article we focus on small-size call centers that are more frequently used in B2B environments. Typical situations where one would find such a call center are

- helpdesks for very specialized services, where agents might need special tools or software programs to solve the client's problems remotely (e.g., mobile phone operator);
- management of large accounts. To better serve the particular needs of its clients, a company might want to have operators specially trained to handle the specific problems that might arise with its large accounts (e.g., bank);
- a hotline for retailers. In some situations the retailers provide part of a service (e.g., activation of mobile phone subscription): when a customer has an exceptional demand, the retailer must be able to call on very specialized agents to quickly find a solution (e.g., mobile phone operator).

These types of services are very specialized and will not generate a large demand justifying many agents. Often, companies try to group services that require skills relatively close to each other and where it is then possible to increase pooling through cross-training. Given the specialized nature of the services, cross-training can be expensive. Therefore it is necessary to find the best trade-off between the cost of cross-training and the economies of pooling. In the context of small call centers, this requires combining the stochastic aspects of arrival and service processes with the combinatorial aspects of the staffing decisions.

The contribution of this article is twofold: it is to our knowledge one of the first articles focusing on small-size multiple-skill call centers, and we present an original method to combine the stochastic and combinatorial aspects in an optimization model.

In the following section we present how the problem studied here relates to earlier published work. We then formulate the problem in a more formal way. This is followed by a section that presents a branch-and-bound type algorithm to find an optimal configuration. Then, some numerical experiments are presented to illustrate the usefulness of our method, and the results of a simulation experiment are given to validate the main hypothesis of our model. We end with a concluding section.

2. Relation with Earlier Work

To operate a multiple-skill call center in an efficient way, the manager has to decide on a routing policy for incoming calls and on a staffing level for each group of operators. Gans et al. (2003) and Aksin et al. (2007) present very well-structured and exhaustive introductions and literature reviews on models for the management of call centers.

Assigning an incoming call to a group of operators that can handle the corresponding call type is referred to as skill-based routing. Optimizing the routing policy is a difficult problem, and it is only recently that researchers have started investigating this domain. Garnett and Mandelbaum (2001) present an introduction to the different types of routing schemes and their complexity. Örmeci (2004) shows the form of the optimal routing policy when there are two skills. In the general case with an arbitrary number of skills, Bhulai (2005) derives a heuristic rule to route calls based on the state of the system, and Borst and Seri (2000) propose a heuristic rule to route calls based on a credit scheme that depends on allocations made previously. Sisselman and Whitt (2007) extend the classical skill-based routing by introducing value-based routing. The idea is to add some quality measurement in the model: instead of simply maximizing the proportion of calls that are answered, value-based routing tries to optimize an expected revenue that is an indirect measure of qualitative elements such as first-call resolution or the preference of the operators. The authors show that it is possible to use value-based routing by adapting the skill-based routing algorithm presented in Wallace and Whitt (2005). There is a large literature on the staffing of call centers, but most of the articles are about the staffing of systems with operators that all have the same skill. The general trend in those articles is to show that different variants of what has become known as the "square root staffing rule" enables to reach the best trade-off between service quality and staffing cost.

In multiple-skill environments, there are some articles about call centers with two skills. For example, Stanford and Grassmann (2000) study a call center with a group of monolingual agents handling the majority of the calls and a group of bilingual agents that are able to answer calls of the minority language as well (see also Shumsky 2004 for another approach to a similar setting).

For the staffing in an environment with an arbitrary number of skills, to our knowledge, all articles focus on large call centers and, as a result, neglect the integrality constraints on the staffing levels (at any time one can only have an integer number of agents in each group). Wallace and Whitt (2005) study the case where all agents master the same number of skills. They show that there is a staffing rule where all operators have two skills that already enables to capture most of the efficiency gains that can be obtained from cross-training. Using a fluid model approach, Harrison and Zeevi (2005) develop a way to compute the optimal staffing when most of the traffic randomness comes from the nonstationarity over time of the arrival process. Whitt (2006b) and Green et al. (2007) are other recent references of models that are dealing with variability on a long-term scale. Chevalier et al. (2005) show that in a similar (but somewhat simpler) setting to the one studied here, a very simple heuristic gives results close to optimality. This heuristic essentially says that when only specialists and fully polyvalent operators are available, then 20% of the budget of the call center should be allocated to cross-trained operators. This confirms a general finding of articles in this area: that a little flexibility goes a long way. The same type of conclusion can be drawn from several articles that analyze call centers using heavy traffic models (see, e.g., Gans and van Ryzin 1997 or Harrison and López 1999).

The methods developed in these articles are not applicable in small-size call centers where the combinatorial side of the problem is predominant.

Because the optimal staffing depends on the routing policy, we analyze the two decisions together. For the routing policies, we restrict ourselves to a type of policy that is often used in practice because it is both simple and intuitive. We call this type of routing policy "hierarchical routing." The principle is to route arriving calls to the group with an available agent that has the smallest level of cross-training (with the necessary skill to handle the call). The idea is to try to keep the more flexible servers free to be able to handle more calls in the future (the heuristic proposed by Borst and Seri 2000 is based on very much the same idea).

Our objective is to find the optimal number of servers in each group and the corresponding routing rule to meet a service quality target at the minimum operating cost. Our service quality indicator is the proportion of lost calls if the system were a pure loss system (that is, when all operators are busy, callers do not have the opportunity to stay on hold). Although in practice almost all call centers provide the opportunity for callers to wait on hold, a loss model will provide a good approximation of the *relative* performance of different configurations in many situations. In particular, when callers are impatient and soon renege, or when the call center wants to provide small waiting times to their callers, it has been shown that the loss probability is a very good indicator of the call center performance (see, e.g., Tabordon 2002, Chevalier et al. 2005). Whitt (2006a) also finds that in a queueing model with abandonments, the rate of abandonment has a much smaller influence on performance than the arrival and/or processing rate.

Even for a small-size center (less than 10 agents) with few different skills (three or four), the number of possible configurations is very large (several hundred millions with three skills and several hundred billions with four skills). To find an optimal configuration it is necessary to compute bounds on a set of configurations. To estimate the performance of a configuration, we use an overflow model where arriving calls are overflowed from one agent group to another. We use the so-called Hayward approximation to compute loss probabilities. The first publication of this method is by Fredericks (1980). In Tabordon (2002) this approximation method was adapted to model call centers. In this paper we go one step further and develop a method to optimize the staffing of a call center. In particular, we show that it is possible to combine the hierarchical structure of the routing policy with the characterization of the call overflows to derive useful bounds that significantly speed up the search for an optimal configuration. Because we use a combinatorial optimization approach based on partial enumeration of the solution space, the size of the problems we can handle is limited.

3. Problem Formulation

We call C the set of all different call types. We assume that the arrival process for calls of each type is a Poisson process with parameter λ_i ($i \in C$). Each call type corresponds with one skill for the agents: to answer a call of type i an agent needs to have the skill i. Since agents can have multiple skills, we call a group of agents all the agents that have the same skill set. Formally, group g has the skills $S(g) \subset C$, which means that all agents of group g can answer any call of type i: $\forall i \in S(g)$.

We suppose that the processing time for calls handled by all groups is exponential with rate μ. The hypothesis that service times are exponential is not so important for the loss probability as it is well known that the Erlang B formula, on which our approximation scheme is based, is valid for any service time distribution. Simulation results suggest the insensitivity of the service time distribution holds even for the non-Poisson overflow processes; details of the simulation tests are not reported because they are beyond the scope of this paper. Note that waiting times, on the other hand, are strongly influenced by the service time distribution. The other limitation of this hypothesis is that the service time distribution is identical for all groups and all call types. This is, in fact, not such a bad approximation, as there is little benefit to pool calls that would have very different processing time distributions. van Dijk (2004) showed that efficiency could actually decrease in such circumstances. In practice, we observe that call center operators try to pool call types that are relatively similar.

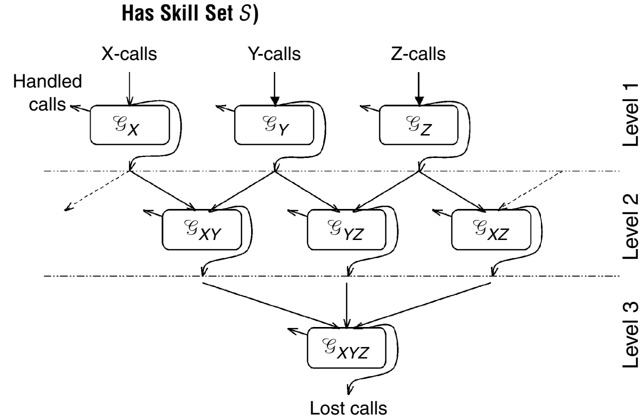
As already mentioned, we study a hierarchical routing, where arriving calls are routed to the group with an available agent that has the smallest set of skills (that includes the needed skill for the call). This policy tries to keep the more flexible servers free to be able to handle more calls in the future.

In this context, we call the first level the set of all agent groups that are dedicated to one type of call. The second level contains all agent groups that have two skills. In general, the mth level contains the groups of agents that have m skills (all groups g where $|S(g)| = m$). We call $G(m)$ the set of all groups of level m:

$$G(m) = \{g\colon |S(g)| = m\}.$$

The principle of the hierarchical routing policies is then to route arriving calls preferably to a group of agents with the adequate skill belonging to the smallest level where that skill is available. If no agent is available in that group, then the call is "overflowed" to a group of agents from the second-smallest level that has the adequate skill. If all agents from that group are busy, the call is overflowed again to further levels until either a group is found where a suitable agent is available, or the call is overflowed from the last level. Figure 1 shows the structure of the hierarchical routing in the case of a call center handling

Figure 1 **Illustration of the Hierarchical Routing Concept (Group $\mathcal{G}_S$ Has Skill Set S)**



three different types of calls. For analytical tractability we do not allow a "horizontal" routing between two groups at the same level. Although a system with such horizontal routing should be efficient, when the routing probabilities are optimized, it seems, from our simulation section (see §6), that the gain is not very important.

The hourly cost of an agent is a function of the number of skills she has. We normalize the hourly cost of an agent with a unique skill to 1. An agent with m skills has a cost of $1+(m-1)p$. This corresponds to a "premium" of p for each additional skill. This cost structure has been observed in several call centers we studied.

Our objective is to find the configuration that minimizes the total operating cost subject to a given service-level constraint. We express the service-level constraint in terms of the loss probability that the system would incur if calls could not wait at all. We choose this performance measure because it is much easier to compute than the expected waiting time or than a given percentile for the waiting times. As already mentioned, earlier work has shown that there is a very good correlation between these performance measures (especially for small loss probabilities/small waiting times). Another reason to study this system is for loss systems, where waiting is not possible at all (for example, paying services). Given the very high combinatorial complexity of the problem, it is important to have a performance measure that is not too hard to compute, lest there is no hope to be able to find optimal solutions.

The maximum loss probability for any incoming call is β. This corresponds to the probability that an entering call will find no group with an available agent and will be overflowed from successive levels until it is finally overflowed from the last level and thus rejected. We impose this loss probability constraint for each type of call to ensure an identical quality of service. We could just as easily impose a distinct loss probability constraint for each call type, but we must remember that the loss probability is only a proxy for the service performance of the call center. If it is clear that imposing the same loss probability would lead to a similar service quality for each type of call, we have no good estimate as to how a measure of service quality (e.g., average waiting time) would vary between the different call types as a function of the distinct loss probabilities for each call type.

$$\min \sum_g c_g n_g \tag{1}$$

$$\Psi_i(\mathbf{n},\mathbf{q}) \le \beta \quad \forall\, i \tag{2}$$

$$\sum_{g\in G(m):\, i\in S(g)} q_{img} = 1 \quad \forall\, i, m \tag{3}$$

$$n_g \in \mathbb{N} \quad \forall\, g, \tag{4}$$

where

- c_g is the server cost for group g;
- n_g is the number of servers in group g;
- $\Psi_i(\mathbf{n},\mathbf{q})$ is the proportion of type i calls that are lost;
- β is the maximum proportion of lost calls;
- q_{img} is the proportion of type i calls routed to group g at level m; and
- $\mathbf{n}$, $\mathbf{q}$ stand, respectively, for the vector of decision variables n_g and the matrix of decision variables q_{img}.

The loss probability $\Psi_i(\mathbf{n},\mathbf{q})$ is computed using the so-called Hayward approximation method, which was first developed in Fredericks (1980) (see also Whitt 1984 and Guerin and Lien 1990 for evaluations of this method). It was further developed in the framework of call centers in Tabordon (2002). Here, we present only a brief summary of the computation method. For more details, the interested reader is referred to the articles cited above or to Chevalier and Tabordon (2003), where an exposition is made of how to adapt the Hayward approximation method in a call center setting. Franx et al. (2006) have recently developed a method to estimate overflows for call centers using a hyperexponential model for flows of calls. It would probably not be very difficult to use that method to compute an alternative approximation to $\Psi_i(\mathbf{n},\mathbf{q})$.

The hierarchical routing policy makes it possible to view the system as if an arriving call was first sent to the corresponding group with one skill, and then, if all agents of that group were busy, the call would be overflowed to a suitable group of agents with two skills, and so forth, as long as all agents in the group reached were busy (see Figure 1). To compute the loss probability of a call, we model the system as a queueing network with flows of calls representing the arrival processes and the overflows between the different agent groups. The flows of calls are modeled using a two-parameter approximation, one being the flow rate and the other being the peakedness, which is a measure of the "burstiness" of the flow. It is defined as the ratio of (i) the variance in the number of servers needed to serve all calls (assuming an $M/M/\infty$ queue), and (ii) the average number of servers needed to serve all calls. Note that a Poisson process has a peakedness equal to one. Overflows tend to be quite bursty, and this has to be taken into account to compute the loss probabilities with a correct precision.

The Hayward approximation is based on the Erlang B formula. If the arrival rate to a group of agents is λ, and the peakedness of the arrival process to that group is z, then the overflow probability for a call reaching that group is

$$B(n,\lambda,z) = B_E\left(\frac{n}{z}, \frac{\rho}{z}\right),$$

where

- n is the number of agents of the group,
- $\rho = \lambda/\mu$ and μ is the service rate of that group, and
- $B_E()$ is the Erlang B formula.

The peakedness of the overflow of calls from that group is then

$$z^{out} = z\left(1 - \frac{\rho}{z}B(n, \lambda, z) + \frac{\rho}{n + z - \rho(1 - B(n, \lambda, z))}\right).$$

When n flows of calls are merged together, the peakedness of the merged flow is

$$z = \frac{\sum_{i=1}^{n} \lambda_i z_i}{\sum_{i=1}^{n} \lambda_i},$$

where λ_i and z_i are, respectively, the arrival rate and peakedness of the merging flows. When a flow (with rate λ and peakedness z) is split according to probabilities $\{p_i\}$, then $p_i\lambda$ and z are, respectively, the arrival rates and peakedness of the resulting flows.

In our model we consider that after each level all calls of the same type are first merged together and then, if need be, split again among the different pools handling this particular type of calls. When such a split is done, the different fractions are the decision variables q_{img}.

We make the simplifying assumption that when a group handles different call types, the overflow probability and the peakedness of the resulting overflow is identical for each call type. Simulation experiments show that the loss probability and peakedness of the overflow of a particular type of call will both be higher if the incoming peakedness of the flow of calls of that type is higher. To the best of our knowledge there has not been any publication about a model for this relation. Moreover, our simulation experiments as well as those performed earlier in Tabordon (2002) suggest that this assumption does not create a large distortion in the approximated loss for the system.

4. Finding an Optimal Configuration

Because we want to find the optimal staffing for call centers with small loads, it is likely that some of the groups of agents (especially the more flexible ones) will be of very small size. Therefore, it is important to take the integrality constraints into account. In all but the smallest instances the number of possible configurations is tremendously large. To be able to find an optimal configuration, we developed a branch-and-bound-type search algorithm. In the following subsections we first show how it is possible to limit the search space in the branching process and then how the Hayward approximation scheme can be used to derive bounds that are useful to restrict the search even further. Finally, we present two different search strategies in the branch-and-bound tree. The ideas implemented in this algorithm are quite similar to the ideas used in integer linear programming (see Wolsey 1998 for a good presentation of branch-and-bound techniques in combinatorial optimization).

4.1. Branching

To enumerate the different configurations, we gradually fix the sizes of the groups and the routing probabilities starting from the first level and then proceeding level by level until we reach the last level. The routing probabilities are enumerated with a step size of 0.1.

At level m, each group has m skills, so there are $\binom{N}{m}$ possible groups at that level (N is the total number of call types). This can be a very large number, and having operators in each of these groups will completely ruin the pooling gains. Because of the concave nature of the Erlang B formula, it is always optimal to have the smallest number of groups possible at any level (given that, for any given level, the cost is identical for all groups). To determine the number of groups that might be needed at level m, we note that we must have enough groups to keep N degrees of freedom to determine an adequate service level for each type of call at that level. With k different groups at level m, we have km flows of calls entering the different groups (remember that at level m, all groups can handle m different call types). This gives $km - N$ independent decision variables because, for each call type, we know that the sum of the routing probabilities at level m should be one (see (3)). Practically, this removes N degrees of freedom. In addition, we have k variables for the sizes of the groups. In total, we have $km - N + k = k(m+1) - N$ independent decision variables. This number should be larger than N to be able to decide independently on the service level for each call type. Note that we neglect that two configurations yielding the same service levels for each call type could have overflows with a different peakedness. In practice, with small configurations, the difference in peakedness that could be observed for configurations with equivalent loss probabilities will only have a second-order effect on the performance of the call center.

This implies that $k \geq 2N/(m+1)$. For any configuration with more than $M(m) = \lceil 2N/(m+1) \rceil$ groups, it is possible to find a combination yielding the same service levels with a smaller number of groups. In our branching procedure we use this fact that whenever at level m, we have $M(m)$ groups with a positive number of agents; we immediately fix the number of agents to zero for all other possible groups of that level (and for all the corresponding routing variables).

It can easily be checked that $M(m) \leq \binom{N}{m}$ for all values of m except the last level, where only one group is present (fully cross-trained agents). Note that $M(N) = 2$, which is the number of groups that would be needed to be able to assign independent service levels for each call type. Such a configuration is beyond the scope of our model. It seems, anyway, that the diseconomies of scale linked to splitting a group in two would almost never make such configuration economically interesting.

Remember that, in our approximation method for computing the loss probabilities, all different types of calls handled by a group have the same overflow probability. Since at the last level we can have only one group of (fully cross-trained) operators, the search space is significantly reduced. Indeed, the loss probability for a type of call is the product of its overflow probabilities across all levels. Consequently, at the next-to-last level, we need only to consider configurations with identical loss probabilities.

4.2. Bounding

Given the decision variables that were fixed, we compute a lower bound on the cost of the possible configurations that would satisfy the service constraint. For this, we solve a relaxation of the problem where we suppose that

—all remaining flows (overflows of agent groups and arrival processes) can be treated together without having to incur the cost of fully cross-trained agents, and

—the service constraint has only to be satisfied globally for all call types and not individually for each one.

Because we do not know how the different flows of calls will be combined, we must be careful in the way we treat the peakedness of the different flows. Remember that in the Hayward approximation the loss probability is computed by dividing the arrival rate and the number of servers in the Erlang B formula by the peakedness. The concavity of the Erlang B formula implies that for a given load a lower peakedness will always result in a smaller loss probability (or equivalently, for a given loss probability, a lower peakedness will require a smaller number of agents). So, here, we start by "normalizing" the flows of calls, i.e., by dividing their rate by their peakedness. We can then merge the different flows as if their peakedness was one. Using the Erlang B formula, we compute the number of servers needed to reach the desired loss probability. This is the global loss probability for all call types determined by dividing (i) the maximum rate of calls overflowed from the system when the service constraint is reached for all call classes ($\beta \sum \lambda_i$) by (ii) the sum of the flow rates of all overflows at the current partial configuration. The number of servers found is then divided between the different flows in proportion to their normalized rates. All parts are multiplied by the peakedness of their corresponding flows. We finally obtain the lower bound by summing them. When searching for the minimal number of agents to satisfy the global loss probability constraint, we use values for the number of servers that (after multiplying back by the different peakedness) give an integer number of agents.

This procedure determines a lower bound because we compute the number of servers needed with the full benefit of pooling (all types of calls are handled by a single group), but the peakedness of each stream of calls is kept separate. Indeed, when joining two flows of calls, the peakedness is the weighted average of their peakedness. In the Hayward approximation the total flow is divided by this average peakedness. Dividing each flow by its own peakedness (normalizing as we call it in our procedure) is equivalent to joining the flows and computing the peakedness as the weighted harmonic average of the flows. Since the weighted harmonic average is always lower than the weighted (arithmetic) average, and since a lower peakedness is always favorable, this will result in a smaller number of agents needed to reach a target loss rate.

Example 1. To illustrate the bounding procedure, we present a small numerical example. Consider two flows, A ($\lambda_A = 3$, $z_A = 1.5$) and B ($\lambda_B = 5$, $z_B = 2$), that would be the overflows resulting from the groups that were already fixed. The required global loss probability (GLP) is 0.2.

The first step to find the bound is to normalize the flows, i.e., to divide both flows by their peakedness. We obtain for A $\bar{\lambda}_A = \lambda_A / z_A = 3/1.5 = 2$ and for B $\bar{\lambda}_B = \lambda_A / z_A = 5/2 = 2.5$. The total normalized flow, $\bar{\lambda}_{tot}$, is thus equal to $\bar{\lambda}_A + \bar{\lambda}_B = 4.5$.

The next step is to determine the number of agents that are required. First, we compute $\bar{s}$, the number of agents required to reach the required global loss probability with the normalized loads. After that, $\bar{s}$ is modified to get s, the equivalent number of operators that takes into account the different peakednesses. This is done by multiplying a proportion of $\bar{s}$ by z_A and the remaining part by z_B. This proportion corresponds to the weight of $\bar{\lambda}_A$ in the total normalized flow. Therefore, we have $s = ((\bar{\lambda}_A/\bar{\lambda}_{tot})\bar{s})z_A + ((\bar{\lambda}_B/\bar{\lambda}_{tot})\bar{s})z_B = ((\bar{\lambda}_A/\bar{\lambda}_{tot})z_A + (\bar{\lambda}_B/\bar{\lambda}_{tot})z_B)\bar{s} = (\frac{2}{4.5}1.5 + \frac{2.5}{4.5}2)\bar{s} = 1.78\bar{s}$. Note that 1.78 corresponds to the weighted harmonic average of the peakedness of the flows ($1.78 = (\lambda_A + \lambda_B)/(\lambda_A/z_A + \lambda_B/z_B)$).

However, we wish to have s as an integer. This implies that we cannot take whatever value of $\bar{s}$. We should choose it accordingly. From $s = 1.78\bar{s}$, we immediately have $\bar{s} = \frac{1}{1.78}s = 0.5625s$. Consequently, to have s integer, we should choose for $\bar{s}$ to be a multiple of 0.5625.

Table 1 Global Loss Probabilities as a Function of s

s	1	2	3	4	5	6	7	8	9	10
$\bar{s}$	0.563	1.125	1.688	2.25	2.813	3.375	3.938	4.5	5.063	5.625
GLP	0.897	0.796	0.67	0.608	0.521	0.439	0.365	0.297	0.237	0.185

Table 1 gives the loss probabilities for integer values of s. We see that with one operator we would have a global loss probability of 0.8965. By gradually increasing $\bar{s}$ by multiples of 0.5625 we find that $s = 10$ is the smallest value for which the global loss probability is smaller than 0.2. $s = 10$ is therefore the value of the bound.

4.3. An Improved Bound

Until now, we have not taken into account the cost of the different groups of agents. We can improve our lower bound by noting that for each flow there is a lower bound on the cost of agents that will handle the overflow. For example, if one of the merged flows is an overflow from a group at the second level, then, with the hierarchical routing policy, these calls can be handled in groups belonging to level 3 or higher. Because all agents at the nth level have n different skills, they also all have the same cost. If we suppose that all calls from the overflow of level 2 are handled by agents with a level 3 cost, this is a lower bound (as some calls might actually be overflowed again and be handled by agents in even higher levels).

In the preceding bounding procedure, after having multiplied the parts of operators by their peakedness, we also multiply these parts by the cost of agents at the next level. We then add up the costs of the different flows.

To illustrate this, let us continue the example of §4.2.

Example 2. We calculated that the bound must be equal to 10 with $\lambda_A = 3$, $z_A = 1.5$, $\lambda_B = 5$, $z_B = 2$ and with a required global loss probability of 0.2. We suppose now that A is a second level overflow and B a first level overflow and that the premium p is equal to 0.1. We wish to compute $c(s)$, the bound that takes the cost of the different groups into account.

Remember that to have $s = 10$ as a bound we needed $\bar{s} = 5.625$ (see Table 1). $c(s)$ is also computed from $\bar{s}$, but instead of computing $s = ((\bar{\lambda}_A/\bar{\lambda}_{tot})\bar{s})z_A + ((\bar{\lambda}_B/\bar{\lambda}_{tot})\bar{s})z_B$, we have to weigh each term with the corresponding cost. Therefore, we have $c(s) = ((\bar{\lambda}_A/\bar{\lambda}_{tot})\bar{s})z_A(1+2p) + ((\bar{\lambda}_B/\bar{\lambda}_{tot})\bar{s})z_B(1+p) = ([\frac{2}{4.5}1.5]1.2 + [\frac{2.5}{4.5}2]1.1)\bar{s}$. With $\bar{s} = 5.625$ (this corresponds to $s = 10$), we have $c(s) = 11.5125$. It is the new bound.

Note that this is a simplified example. If there really were only two flows of calls, one could assume that they can only be merged at the third level and use the cost of 1.2 for both flows. When there are more than two flows, we obtain a lower bound by assuming that each call type will be handled by the first possible level. This is what is done in our small example.

4.4. Branch-and-Bound Tree Search

One of the questions that need to be solved when implementing a branch-and-bound approach is to choose a search strategy in the enumeration tree. The most common search strategies are depth-first search (dfs) and a best-bound search (bbs). The former proposes to explore the tree by selecting the node that has the most variables already fixed (to obtain as soon as possible a feasible solution that constitutes an upper bound). The latter suggests to select the node that has the best lower bound. The dfs strategy tends to minimize the number of active nodes at any time by improving the upper bound. The bbs strategy, on the other hand, tends to minimize the total number of nodes to explore by improving the lower bound. To try to combine the advantages of both approaches, we implemented a mixed strategy where we use bbs when fixing the number of operators for the groups of the first level. When exploring a node where all decision variables of the first level are fixed, we switch to dfs to find the best feasible solution corresponding to the given first-level configuration.

5. Numerical Experiments

We have implemented our optimization algorithm in Java and tested it on a variety of small three- and four-skill examples. We implemented several programs that gradually include the different ideas of §4.

In the following subsections, we analyze the computational performance of our optimization procedure for three- and four-skill call centers. We then analyze the type of characteristics of the optimal configurations found on a sample of 100 randomly generated three-skill examples.

5.1. Three-Skill Call Centers

For systems with three types of calls, there are five parameters to be chosen. The three first are the arrival rates for each call type (we normalize the time scale by fixing $\mu = 1$). For these three we considered five combinations. They are presented in Table 2. The fourth parameter is the value of the cost premium p, and the fifth is the value of β, the maximum loss probability. We took three values for the premium, namely, 0.1, 0.15, and 0.25 and three values for the loss

Table 2 Description of the Five Combinations of Flows Considered for the Three-Call Type Systems

Example	Flow X	Flow Y	Flow Z
1	10	2	1
2	4	4	4
3	8	7	3
4	3	2	1
5	5	4	2

probability, 0.05, 0.1, and 0.2. This gave us 45 cases to analyze. We tried to make a balanced and representative choice. That is why we tested some cases where all three flows are the same, other cases where two flows are similar while the third is quite different, or cases where all flows are different.

We solved these examples using various programs. The first one (R.Met) consists of an algorithm that makes an exhaustive search using the branching rule of §4.1 and only uses upper bounds to limit its search of possible configurations. The starting bound is the minimum between the cost of a configuration with only dedicated agents and a configuration with only fully cross-trained agents.

In the second one (L.Bou.), we include the bounding procedure of §4.2 with a dfs search strategy.

In the next program (N.Bou.), we use the same branching with the improved bound according to the method described in §4.3 with the same search strategy.

Finally, we tested all examples using a program (BBS) that uses the same branching and bounding strategy as (N.Bou) with our mixed-search strategy presented in §4.4.

To check how effective the programs were, we measured the time it took to solve each instance using each program. The results are summarized in Figure 2. This figure shows, for each improvement of the algorithm, the number of examples for which the improvement in solution time lies in each percentage range compared with the exhaustive search method.

The results confirm that each successive improvement in our resolution method improves the speed to find the optimal solution. The average time reductions are 32%, 46%, and 50% for the L.Bou, N.Bou, and BBS programs, respectively (compared with the time to perform an exhaustive search).

We also evaluated the performance of the programs using another measure: we counted the number of configurations that had to be completely evaluated. The results were very similar to the time measures. That is why they are not presented here.

To evaluate our method, we also investigated the cost savings of our proposed solution compared with "naive" configurations such as using either only fully cross-trained agents or only dedicated agents. To check this, we compared the optimal configuration obtained with the method proposed here to both extreme configurations, i.e., a configuration that satisfies the loss probability constraint with only dedicated operators (DED) and with only polyvalent (i.e., fully cross-trained) operators (POL). Figure 3 shows the number of examples for each range of percentage cost improvement of the optimal solution compared with the two naive policies. Over all 45 examples we observed an average cost reduction of 14% compared with the DED staffing and 11% compared with the POL staffing.

5.2. Four-Skill Call Centers

We now investigate the performance of our procedure for centers with four types of calls. As the benefit of the improved bound over the basic bound appeared almost constant in our instance with three types of calls, we did not include the (L.Bou) program in our comparison. Because the number of possible configurations increases tremendously, we had to restrict

Figure 2 Time Improvements Brought by Each Algorithm Improvement for the Three-Call Type Systems

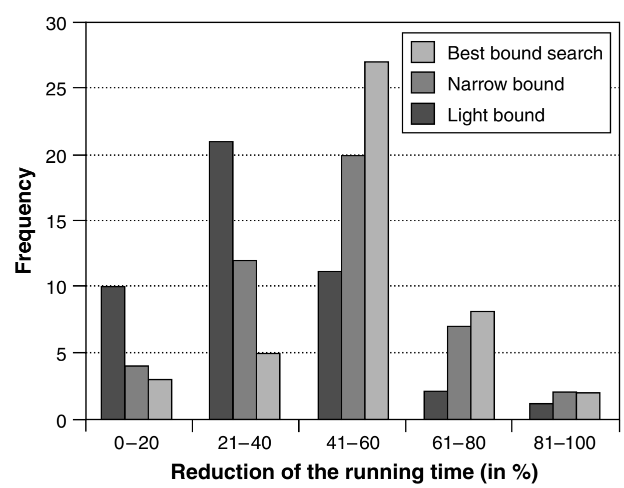


Figure 3 Percentage Reduction of the Total Cost Compared with Configurations with POL or with DED

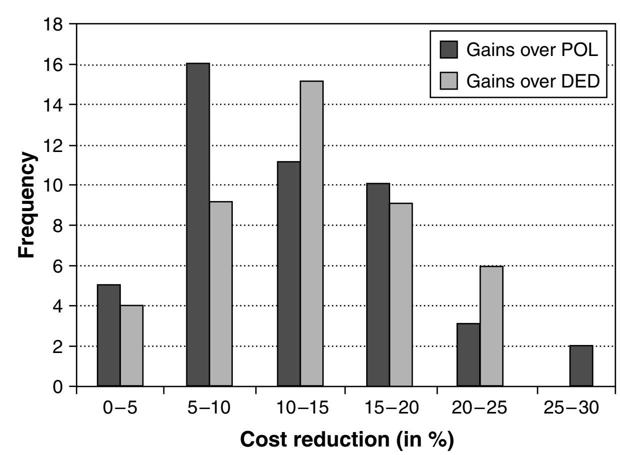


Table 3 Description of the Test Set for the Four-Call Type Systems

Example	Flow A	Flow B	Flow C	Flow D	Premium	β
1	1	1	1	1	0.05	0.05
2	2	1	1	1	0.05	0.05
3	3	2	1	1	0.05	0.05
4	0.5	4	1	0.5	0.15	0.05
5	2	1	1	1	0.2	0.05

ourselves to smaller loads than the loads chosen in the three-call type systems to keep the resolution times manageable (especially of the exhaustive search program). For the same reason, we only work with five examples. The parameters of each of them are presented in Table 3.

As expected, the computation times were much longer than for the systems with three types of calls. The results are presented in Table 4: the first column gives the time duration for the exhaustive search algorithm, the other ones present the time duration for each improvement in a normalized way.

Here, we see that our procedure really makes a huge difference for the computation time. We now have a reduction in computation time of at least 88% in all instances and a reduction of more than 95% in all but one instance. However, for the three first examples, it may look like the BBS program is not improving as much since the processing time is similar or even longer with BBS than with N.Bou. The three first examples consider a relatively low premium (see Table 3). The consequence is that the optimal configuration in two of the three cases is a configuration with only fully cross-trained operators (see Table 5, where we may observe no cost decrease for these two examples). Since the initial bound (the one used for starting the algorithm) is the minimum between the two extreme configurations (POL and DED), the algorithm begins with a bound that is the best configuration. The consequence is that, in this case, the mixed-search strategy cannot find a better upper bound faster. This is supported by the fact that we did not observe any difference in the number of configurations evaluated using both methods. When we look at example nr.5, which is exactly the same as example nr.2 except for the higher premium, we see that the mixed-search strategy still provides a nonnegligible improvement.

Table 4 Time Improvements Brought by Each Algorithm Improvement for the Four-Call Type Systems

Example	Time	R.Met	N.Bou	BBS
1	6,921,172	100	2.97	3.06
2	8,734,740	100	3.16	3.15
3	29,316,375	100	3.71	3.78
4	21,111,376	100	12.39	11.48
5	33,257,332	100	7.08	4.91

Table 5 Comparison of the Staffing Costs of the Optimal and the DED and POL Policies for the Four-Call Type Systems

Example	DED	POL	OPT
1	16.0	9.2	8.8
2	17.0	10.35	10.35
3	20.0	12.65	12.65
4	18.0	14.5	13.1
5	17.0	14.4	12.4

Looking at Table 5, we see that, again, the optimal configuration brings important gains relative to both extreme configuration. These gains tend to be higher if the premium is large, as the comparison between examples nr.2 and nr.5 suggests.

5.3. Analysis of Optimal Configurations for Three-Skill Call Centers

We used our optimization procedure to further investigate the structure of the optimal configurations. To do that, we randomly generated 100 instances. In each of these instances, the incoming flows were randomly chosen in an interval between 0 and 10, the premium for cross-training was between 0.05 and 0.3, and the loss probability could go from 0.01 to 0.25. For each case, the optimal configuration was found. When analyzing the results we could observe some interesting facts, mainly concerning the second level. As shown in Figure 4, the large majority of optimal configurations had only one pool of two cross-trained operators. In our sample, only four optimal solutions out of 100 have a second pool in level 2. Moreover, that second pool consists, in all four cases, of only one operator, suggesting that the flow passing through it is minor. Actually, this is not surprising: it tends to show that dividing flows in multiple parts goes against optimality. This is coherent with the well-known phenomenon of economies of scale.

A second important point to notice is that the 80%–20% rule stated in Chevalier et al. (2005) does not appear to work in our case. As Figure 5 shows, the proportion of the budget allocated to the dedicated operators is lower than one could expect. Still, the general finding that "a little flexibility goes a long

Figure 4 Percentage of Cases with Zero, One, or Two Groups in the Optimal Solution at Level 2

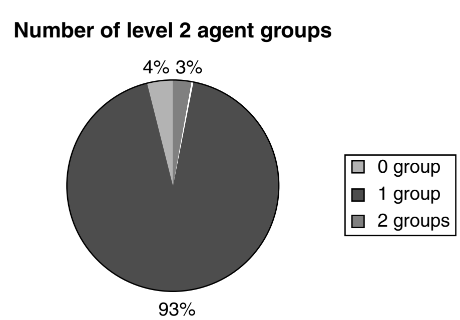
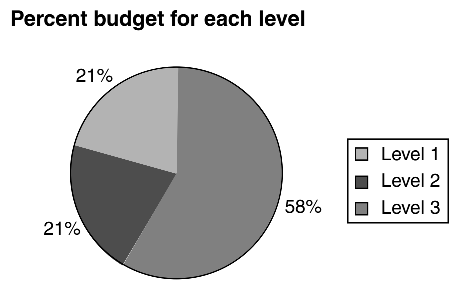


Figure 5 Optimal Repartition of Operators Across the Different Levels



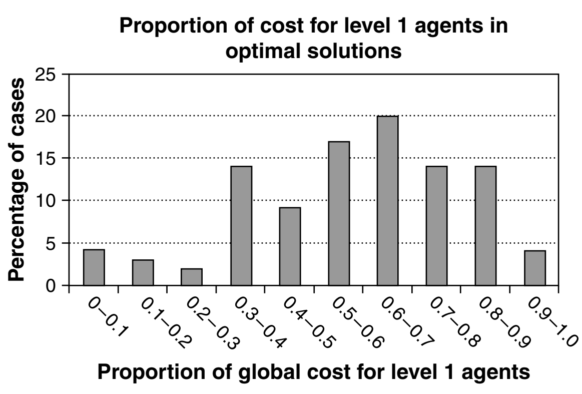
way" can be observed here again with an average of only approximately 20% of the staffing cost used for second-level and third-level operators. This could be compared, for example, with the observations of Wallace and Whitt (2005), where there is no cost for additional skills and operators all have the same number of skills. In that setting the authors find that most benefits from cross-training are obtained by having agents with two skills. Here we see that, when there is a cost for additional skills but it is possible to have agents with different levels of skills, the number of additional skills in the best solutions is again very limited.

A closer analysis shows that there are huge differences between optimal configurations (see Figure 6) across our sample of instances. The proportion of dedicated operators, operators who can deal only one type of call, may go from 0% to 100%). It provides actually a justification to the specific study of call centers of small size: the integrality condition may change the optimal solution significantly.

6. Simulation Study

For analytical tractability, many simplifying assumptions have to be made. We would now like to investigate their impact on the results we obtain. Thus, we ran a simulation experiment. First, we compared the loss probability computed using our model with the loss probabilities observed in the simulations. Second, we evaluated the relevance of our model for situations where waiting is allowed. Finally, we checked whether allowing horizontal routing makes a large difference.

Figure 6 The Optimal Proportion of the Total Cost That Is Spent for Level 1 Agents



6.1. Validation of the Results

We start this simulation study by examining whether the loss probabilities estimated with our model match the loss probabilities observed in a simulation experiment. Because the objective of our model is to find the best configuration, we focus our investigation on how well the relative performance of different configurations is estimated. The test consists in taking a series of relatively similar configurations with the same combination of incoming flows. We took examples with three call types and arrival rates of 8, 7, and 3. We started with the optimal configuration with a cost premium of 0.25 and maximum loss probability of 0.1 found in §5.1. We performed an exhaustive search and selected 29 other configurations that, according to our model, give loss probabilities very close to 0.1 for each call type. In Figure 7, we compare the computed and simulated loss probabilities for those 30 examples.

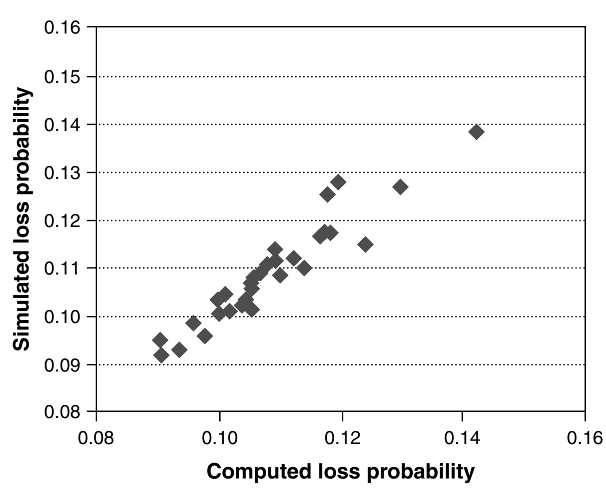
This figure confirms that there is a very good agreement between the simulated and computed loss probabilities.

6.2. Evaluation of the Queueing Approximation

For analytical tractability we cannot explicitly consider queues. We argued in §3 that the results obtained with a loss model can be a good proxy for models with waiting. In other words, our idea is that the best configurations in the blocking model are also among the best configurations in a queueing model.

To verify that, we took the set of 30 configurations we used in §6.1. We now compare the results we get

Figure 7 Correlation Between Computed Loss Probability and Simulated Loss Probability



when simulating the configurations with waiting and the results we compute using our model. For each configuration, we consider two cases.

First, we introduce queues of limited sizes. More precisely, we allow a maximum of two calls to be put on hold for each of the two first types of calls and a maximum of one call for the third flow. This gives queues of relatively comparable sizes proportional to the arrival rates. All callers are considered to have infinite patience. In the present case, it is possible to compare the computed loss probability to two performance measures. Because some calls are still lost we can compare the computed loss probability with the simulated loss probability like in §6.1. The second performance measure is the average waiting time. Figure 8 summarizes the results when the size of the queues is restricted.

From Figure 8 it appears that there is a remarkably strong correlation between the performance in both systems. Consequently, using the loss probability as the performance when waiting is possible is not bad at all.

Figure 8 Correlation Between Computed Loss Probability and (a) Simulated Loss Probability and (b) Average Waiting Time

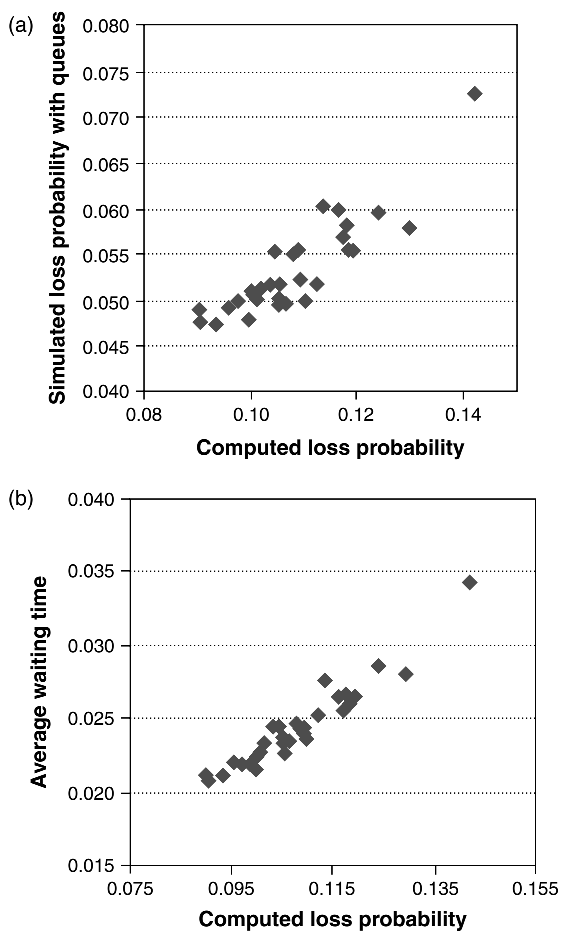
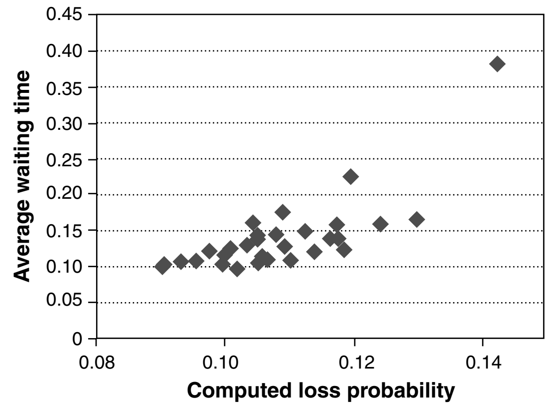


Figure 9 Correlation Between Computed Loss Probability and Simulated Loss Probability with Queues



Second, we consider queues of infinite size. Again, callers never become impatient. The consequence is that a call is never lost and the blocking probability is thus equal to zero. That is why we only compare the computed loss probability with the average waiting time. Figure 9 summarizes the results when the size of the queue is unlimited.

The correlation is not as strong in this case. Still, there clearly is a correlation. Moreover, one must keep in mind that the configurations we try to compare are configurations that have quite similar performances.

In conclusion, it is clear that call centers where waiting is possible are much more prevalent than loss systems, yet a loss model can give good performance indications.

6.3. Horizontal Routing

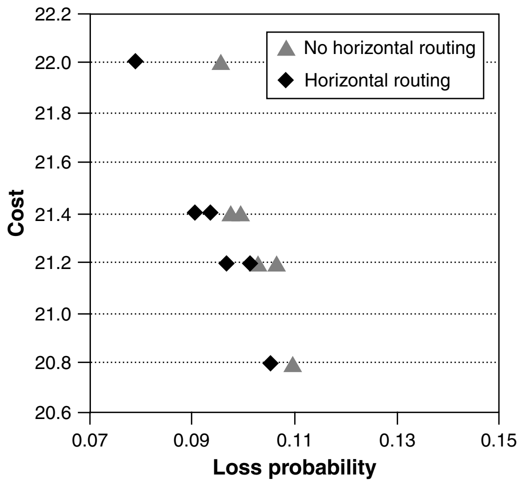
For analytical tractability in our model, we do not allow horizontal routing. When flows have the possibility to be routed from one pool to another pool of the same level, the computation of the incoming flows at each pool becomes much more complicated. Deriving a bound for the branch-and-bound search would also be much harder.

From the results of §5.3 we could assume that very often there is only one group present at the second level as a consequence of the strictly hierarchical routing. Perhaps, with the possibility of horizontal routing between groups at the second level, it would be interesting to have more groups at that level.

The queueing simulation gives an indirect answer to the question already. Indeed, when waiting is allowed, a call will not be put on hold if there is a pool that can handle it, so horizontal routing must be allowed. We saw that this did not prevent a very good agreement between the simulated and computed performance.

The following simulation experiment is meant to have more direct answer to this question. We study three different cases of call centers with three call

Figure 10 Cost vs. Average Loss Trade-Off for Policies With and Without Horizontal Routing (Example with Arrival Rates = 5 for Each Type)



types, two cases with identical arrival rates for each call type and one case with one call type having a much smaller arrival rate than the two others. These seem to be the cases where horizontal routing would be most beneficial. The justification would be for the case of equal arrival rates that horizontal transfers could make a symmetrical configuration more attractive and, for the case with a much smaller demand for one type, that a second-level server that can handle calls of the infrequent type would have spare capacity, which is easier to use with horizontal transfers.

For each case, we simulated more or less all configurations that had a global loss probability close to 0.1 with not too large an imbalance between the three classes.

Figure 11 Cost vs. Average Loss Trade-Off for Policies With and Without Horizontal Routing (Example with Arrival Rates = 6 for Each Type)

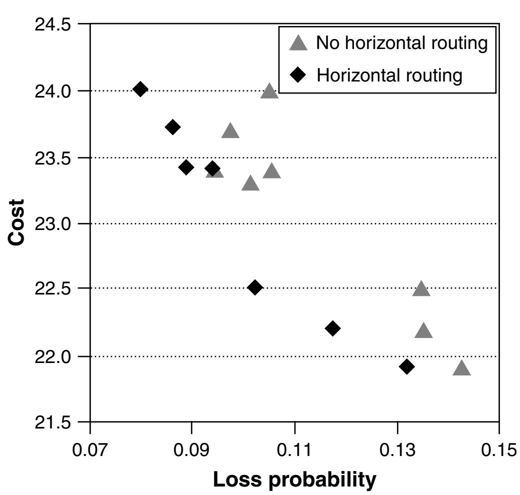
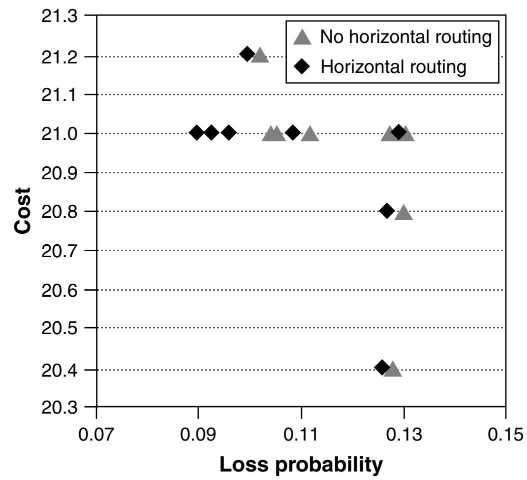


Figure 12 Cost vs. Average Loss Trade-Off for Policies With and Without Horizontal Routing (Example with Arrival Rates = {7,7,1}, Respectively)



In Figures 10, 11, and 12 we observe that horizontal routing reduces the loss probability, especially when there are numerous operators in the second level. Generally speaking, the gains do not appear to be large: to get an important gain from horizontal routing, we have to change the optimal configuration significantly to get a large number of second-level operators, and this results in a degradation of the performance seldom compensated by the horizontal routing. The fact that we have a discrete problem implies, nevertheless, that there are large jumps between configurations and that there are points where the difference can be significant. The details can be found in Tables 6, 7, and 8.

7. Conclusion and Further Research

In this article, we presented a method to optimize small multiskill call centers typical of B2B environments. We could show that in such environments, unless both the cross-training cost and the loads are very low, optimizing the configuration of the call center yields significant gains over simpler all-dedicated or all-cross-trained configurations. Moreover, we could observe that in settings with small loads, the combinatorics of the problem significantly influences the solution.

To obtain an efficient optimization algorithm, we translated the main ideas of branch-and-bound techniques for a stochastic model. This seems to be a promising way to handle stochastic models where there are some discrete decision variables.

Our optimization procedure led us to the findings of §5.3. They tend to show an interesting property: it is, in general, optimal not to divide overflows in multiple parts. This property has, so far, only been empirically observed, but it is quite logical: division

Table 6 Comparison of Configurations With and Without Horizontal Routing When the Incoming Arrival Rates Are All Equal to 5 and the Cost Premium Is Equal to 0.2

Example	Sx	Sy	Sz	Sxy	Sxz	Syz	Sxyz	Horiz. rout.	LPx	LPy	LPz	LPtot	Cost
1	4	4	4	2	2	2	2	1	0.0829	0.0744	0.0799	0.079	22
2	4	4	4	2	2	2	2	0	0.0942	0.092	0.1008	0.0957	22
3	5	5	5	1	1	1	2	1	0.0968	0.0917	0.0922	0.0936	21.4
4	5	5	5	1	1	1	2	0	0.1002	0.0926	0.1012	0.098	21.4
5	4	4	4	1	1	0	5	1	0.0967	0.0795	0.097	0.0911	21.4
6	4	4	4	1	1	0	5	0	0.1018	0.0988	0.0979	0.0995	21.4
7	5	5	6	1	0	0	3	0	0.1025	0.1061	0.0839	0.0975	21.4
8	6	4	6	1	1	0	2	1	0.0818	0.1278	0.08181	0.0971	21.2
9	6	4	6	1	1	0	2	0	0.0816	0.1465	0.0811	0.1031	21.2
10	5	6	5	1	1	0	2	1	0.1212	0.063	0.1208	0.1017	21.2
11	5	6	5	1	1	0	2	0	0.1256	0.0746	0.1199	0.1067	21.2
12	6	5	6	1	1	0	1	1	0.1038	0.1135	0.0997	0.1057	20.8
13	6	5	6	1	1	0	1	0	0.1045	0.1271	0.097	0.1095	20.8

Table 7 Comparison of Configurations With and Without Horizontal Routing When the Incoming Arrival Rates Are All Equal to 6 and the Cost Premium Is Equal to 0.1

Example	Sx	Sy	Sz	Sxy	Sxz	Syz	Sxyz	Horiz. rout.	LPx	LPy	LPz	LPtot	Cost
1	2	2	2	4	4	4	4	1	0.0773	0.0811	0.0812	0.0799	24
2	2	2	2	4	4	4	4	0	0.1084	0.1064	0.1004	0.1051	24
3	4	4	4	1	1	1	7	1	0.0875	0.0848	0.087	0.0864	23.7
4	4	4	4	1	1	1	7	0	0.096	0.0982	0.0983	0.0975	23.7
5	4	4	4	3	3	0	4	1	0.0939	0.0694	0.103	0.0888	23.4
6	4	4	4	3	3	0	4	0	0.0989	0.0878	0.0973	0.0947	23.4
7	4	4	4	2	2	2	4	1	0.0948	0.0924	0.0944	0.0939	23.4
8	4	4	4	2	2	2	4	0	0.1054	0.1045	0.1061	0.1053	23.4
9	4	4	6	3	0	0	5	0	0.1043	0.1119	0.0885	0.1016	23.3
10	3	3	3	3	3	3	3	1	0.1049	0.1029	0.1004	0.11027	22.5
11	3	3	3	3	3	3	3	0	0.1316	0.1379	0.1341	0.1345	22.5
12	4	4	4	2	2	2	3	1	0.1161	0.1193	0.1171	0.1175	22.2
13	4	4	4	2	2	2	3	0	0.1329	0.1335	0.1392	0.1352	22.2
14	5	5	5	1	1	1	3	1	0.1304	0.1322	0.1341	0.1322	21.9
15	5	5	5	1	1	1	3	0	0.1388	0.1412	0.148	0.1427	21.9

Table 8 Comparison of Configurations With and Without Horizontal Routing When the Arrival Rates Are Equal to $\{7, 7, 1\}$, Respectively, and the Cost Premium Is Equal to 0.2

Example	Sx	Sy	Sz	Sxy	Syz	Sxz	Sxyz	Horiz. rout.	LPx	LPy	LPz	LPtot	Cost
1	7	7	2	0	1	1	2	1	0.1015	0.1071	0.0375	0.0998	21.2
2	7	7	2	0	1	1	2	0	0.108	0.1027	0.0602	0.1023	21.2
3	6	6	0	1	1	2	3	1	0.0978	0.0768	0.1267	0.0899	21
4	6	6	0	1	1	2	3	0	0.1104	0.0882	0.172	0.1042	21
5	6	6	0	1	3	0	3	1	0.1204	0.061	0.1246	0.093	21
6	6	6	0	1	3	0	3	0	0.1274	0.0839	0.0999	0.1053	21
7	5	5	0	2	3	3	1	1	0.0967	0.084	0.0976	0.0963	21
8	5	5	0	2	3	3	1	0	0.1306	0.1278	0.1485	0.1305	21
9	5	5	2	4	0	0	3	0	0.1068	0.1073	0.0665	0.1043	21
10	5	5	0	0	4	4	1	1	0.1136	0.0508	0.1125	0.1089	21
11	5	5	0	0	4	4	1	0	0.1087	0.1679	0.1065	0.1116	21
12	2	3	1	0	4	5	3	1	0.1377	0.1341	0.043	0.1297	21
13	2	3	1	0	4	5	3	0	0.1308	0.1312	0.0826	0.1278	21
14	3	3	1	0	4	4	3	1	0.1325	0.1339	0.0416	0.1271	20.8
15	3	3	1	0	4	4	3	0	0.1312	0.1349	0.0899	0.1302	20.8
16	4	4	1	0	3	3	3	1	0.1312	0.1323	0.0503	0.1263	20.4
17	4	4	1	0	3	3	3	0	0.1305	0.1294	0.0996	0.1279	20.4

of flow goes against the benefit of pooling. This property should be further investigated as it could lead to the development of faster heuristics to optimize the configuration of larger call centers.

Other ideas for improving the optimization procedure could be to study whether it would be possible to develop a tighter bound, or develop a more sophisticated search strategy. This would, in turn, make it possible to use this approach for larger call centers.

Other questions that remain to be studied are how to extend the routing policies, how to have a better estimation of the performance measures that are related to waiting times, and how could nonexponential service times be handled.

References

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

Bhulai, S. 2005. Dynamic routing policies for multi-skill call centers. Technical Report WS2004-11, Vrije Universiteit, Amsterdam.

Borst, S., P. Seri. 2000. Robust algorithms for sharing agents with multiple skills. Technical report, Bell Laboratories, Murray Hill, NJ.

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

Chevalier, P., R. Shumsky, N. Tabordon. 2005. Routing and staffing in large call centers with specialized and fully flexible servers. Technical report, Université catholique de Louvain, Louvain, Belgium.

Franx, G. J., G. Koole, A. Pot. 2006. Approximating multi-skill blocking systems by hyperexponential decomposition. *Performance Eval.* **63**(8) 799–824.

Fredericks, A. A. 1980. Congestion in blocking systems—A simple approximation technique. *Bell System Tech. J.* **59**(6) 805–827.

Gans, N., G. van Ryzin. 1997. Optimal control of a multiclass, flexible queueing system. *Oper. Res.* **45**(5) 677–693.

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

Garnett, O., A. Mandelbaum. 2001. An introduction to skills-based routing and its operational complexities. Technical report, Technion, Technion City, Haifa, Israel.

Green, L. V., P. J. Kolesar, W. Whitt. 2007. Coping with time-varying demand when setting staffing requirements for a service system. *Production Oper. Management* **16**(1) 13–39.

Guerin, R., L. Y.-C. Lien. 1990. Overflow analysis for finite waiting room systems. *IEEE Trans. Comm.* **38**(9) 1569–1577.

Harrison, J., A. Zeevi. 2005. A method for staffing large call centers based on stochastic fluid models. *Manufacturing Service Oper. Management* **7**(1) 20–36.

Harrison, J. M., M. J. López. 1999. Heavy traffic resource pooling in parallel-server systems. *Queueing Systems* **33**(4) 339–368.

Örmeci, E. L. 2004. Dynamic admission control in a call center with one shared and two dedicated service facilities. *IEEE Trans. Automatic Control* **49**(7) 1157–1161.

Shumsky, R. A. 2004. Approximation and analysis of a queueing system with flexible and specialized servers. *OR Spectrum* **26**(3) 307–330.

Sisselman, M. E., W. Whitt. 2007. Value-based routing and preference-based routing in customer contact centers. *Production Oper. Management* **16**(3) 277–2919.

Stanford, D., W. K. Grassmann. 2000. Blingual server call centers. D. R. McDonald, S. R. E. Turner, eds. *Analysis of Communication Networks: Call Centers, Traffic and Performance*, Vol. 28. Fields Institute Communications Series, American Mathematical Society, Providence, RI, 31–48.

Tabordon, N. 2002. Modeling and optimizing the management of operator training in a call center. PhD thesis, Institut D'Administration et de Gestion, Louvain, Belgium.

van Dijk, N. M. 2004. Should we pool or not for call centers. *First Conf. POMS College of Service Oper., New York*.

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

Whitt, W. 1984. Heavy-traffic approximations for service systems with blocking. *AT&T Bell Laboratories Tech. J.* **63**(5) 689–708.

Whitt, W. 2006a. Sensitivity of performance in the Erlang-a queueing model to changes in the model parameters. *Oper. Res.* **54**(2) 247–260.

Whitt, W. 2006b. Staffing a call center with uncertain arrival rate and absenteeism. *Production Oper. Management* **15**(1) 88–102.

Wolsey, L. A. 1998. *Integer Programming*. John Wiley & Sons, New York.

CORE Reprints -- Recent titles

2107. David DE LA CROIX and Clara DELEVALLADE. Growth, public investment and corruption with failing institutions. *Economics of Governance*, 10, 187-219, 2009.
2108. David DE LA CROIX. Adult longevity and economic take-off from Malthus to Ben-Porath. In Neri Salvadori (eds.), *Institutional and Social Dynamics of Growth and Distribution*. Edward Elgard, 172-190, 2009.
2109. Carmen HERRERO and Juan D. MORENO-TERNERO. Opportunity analysis of newborn screening programs. *Review of Economic Design*, 12, 259-277, 2008.
2110. Juan D. MORENO-TERNERO. Interdependent preferences in the design of equal-opportunity policies. In Emerald Group Publishing Limited (eds.), Inequality and Opportunity: Papers from the Second ECINEQ Society Meeting, *Research on Economic Inequality*, 16, 49-65, 2008.
2111. Carmen HERRERO and Juan D. MORENO-TERNERO. Estimating production costs in the economic evaluation of health-care programs. *Health Economics*, 18, 21-35, 2009.
2112. Jozef KONINGS and Hylke VANDENBUSSCHE. Heterogeneous responses of firms to trade protection. *Journal of International Economics*, 76, 371-383, 2008.
2113. Juan D. MORENO-TERNERO. The proportional rule for multi-issue bankruptcy problems. *Economics Bulletin*, 29(1), 483-490, 2009.
2114. Raouf BOUCEKKINE and Marc GERMAIN. The burden sharing of pollution abatement costs in multi-regional open economies. *The B.E. Journal of Macroeconomics*, 9(1), 1-32, 2009.
2115. Théophile T. AZOMAHOU, Raouf BOUCEKKINE and Bity DIENE. A closer look at the relationship between life expectancy and economic growth. *International Journal of Economic Theory*, 5, 201-244, 2009.
2116. Raouf BOUCEKKINE, Fernando DEL RIO and Blanca MARTINEZ. Technological progress, obsolescence, and depreciation. *Oxford Economic Papers*, 61, 440-466, 2009.
2117. Jean J. GABSZEWICZ, Didier LAUSSEL and Michel LE BRETON. The mixed strategy Nash equilibrium of the television news scheduling game. *The Journal of Industrial Economics* (JIE) Notes and Comments page (http://www.essex.ac.uk/jindec/notes.htm).
2118. Giorgia OGGIONI and Yves SMEERS. Evaluating the impact of average cost based contracts on the industrial sector in the European emission trading scheme. *Central European Journal of Operations Research*, 17, 181-217, 2009.
2119. Paul BELLEFLAMME and Eric TOULEMONDE. Negative intra-group externalities in two-sided markets. *International Economic Review*, 50(1), 245-272, 2009.
2120. Thierry BRECHET and Henry TULKENS. Beyond BAT: Selecting optimal combinations of available techniques, with an example from the limestone industry. *Journal of Environmental Management*, 90, 1790-1801, 2009.
2121. Yuri YATSENKO and Natali HRITONENKO. Technological breakthroughs and asset replacement. *The Engineering Economist*, 54, 81-100, 2009.
2122. Paul BELLEFLAMME. How efficient is the Patent System? A general appraisal and an application to the pharmaceutical sector. In A. Gosseries, A. Marciano and A. Strowel (eds.), *Intellectual Property and Theories of Justice*. Palgrave Macmillan, 210-229, 2008.
2123. Luc BAUWENS and Nikolaus HAUTSCH. Modelling financial high frequency data using point processes. In T.G. Andersen, R.A. Davis, J.-P. Kreiss, and T. Mikosch (eds.), *Handbook of Financial Time Series*. Springer-Verlag Heidelberg, 953-979, 2009.
2124. Per J. AGRELL and Jerome POUYET. Regulatory competition in network interconnection pricing. *Review of Network Economics*, 7(1), 111-135, 2008.
2125. Luc BAUWENS and Giuseppe STORTI. A component GARCH model with time varying weights. *Studies in Nonlinear Dynamics & Econometrics*, 13(2), 1-31, 2009.
2126. Thierry BRECHET, Stéphane LAMBRECHT and Fabien PRIEUR. Intertemporal transfers of emission quotas in climate policies. *Economic Modelling*, 26, 126-134, 2009.
2127. Filippo L. CALCIANO. Reforming European competition policy. *European View*, 8, 125-131, 2009.
2128. Filippo L. CALCIANO and Mario TIRELLI. Public versus private old-age pensions in Europe. *European View*, 7, 277-286, 2008.
2129. Louis EECKHOUDT and Pierre PESTIEAU. A note on longevity enhancing investment. *Economic Letters*, 101, 57-59, 2008.
2130. Luc CHAMPARNAUD, Victor GINSBURGH and Philippe MICHEL. Can public arts education replace arts subsidization? *Journal of Cultural Economics*, 32(2), 109-126, 2008.

2131. Jan FIDRMUC, Victor GINSBURGH and Shlomo WEBER. Voting on the choice of core languages in the European Union. *European Journal of Political Economy*, 25, 56-62, 2009.
2132. Helena BELTRAN, Alain DURRE and Pierre GIOT. Volatility regimes and order book liquidity: Evidence from the Belgian segment of Euronext. *Global Finance Journal*, 20, 80-97, 2009.
2133. Pierre GIOT. L'irrésistible ascension de la finance comportementale. *Bank- en Financiewezen/Revue bancaire et financière*, 156-158, 2009.
2134. Kirill BORISSOV and Stéphane LAMBRECHT. Growth and distribution in an AK-model with endogenous impatience. *Economic Theory*, 39, 93-112, 2009.
2135. Michel BEINE, Sébastien LAURENT and Franz C. PALM. Central bank FOREX interventions assessed using realized moments. *Journal of International Financial Markets, Institutions & Money*, 19, 112-127, 2009.
2136. Jean-Yves GNABO, Sébastien LAURENT and Christelle LECOURT. Does transparency in central bank intervention policy bring noise to the FX market? The case of the Bank of Japan. *Journal of International Financial Markets, Institutions & Money*, 19, 94-111, 2009.
2137. Marco MARINUCCI. Optimal ownership in joint ventures with contributions of asymmetric partners. *Journal of Economics*, 97, 141-163, 2009.
2138. Erik SCHOKKAERT. Critical notice: Macrojustice as a research programme. *Economics and Philosophy*, 25, 69-84, 2009.
2139. Erik SCHOKKAERT and Carine VAN DE VOORDE. Direct versus indirect standardization in risk adjustment. *Journal of Health Economics*, 28, 361-374, 2009.
2140. Koen DECANCQ, André DECOSTER and Erik SCHOKKAERT. The evolution of world inequality in well-being. *World Development*, 37(1), 11-25, 2009.
2141. Marc FLEURBAEY and Erik SCHOKKAERT. Unfair inequalities in health and health care. *Journal of Health Economics*, 28, 73-90, 2009.
2142. V. EKSLER, S. LASSARRE and I. THOMAS. Regional analysis of road mortality in Europe. *Public Health*, 122, 826-837, 2008.
2143. Dominique VANNESTE, Isabelle THOMAS and Lieve VANDERSTRAETEN. The spatial structure(s) of the Belgian housing stock. *Journal of Housing and the Built Environment*, 23, 173-198, 2008.
2144. Giacomo VALLETTA. A fair solution to the compensation problem. *Social Choice and Welfare*, 32, 455-478, 2009.
2145. Yves POCHET and Laurence A. WOLSEY. Single item lot-sizing with non-decreasing capacities. *Mathematical Programming Ser. A*, 121, 123-143, 2010.
2146. Michele CONFORTI, Marco DI SUMMA, Friedrich EISENBRAND and Laurence A. WOLSEY. Network formulations of mixed-integer programs. *Mathematics of Operations Research*, 34(1), 194-209, 2009.
2147. Helmuth CREMER, Philippe DE DONDER, Dario MALDONADO and Pierre PESTIEAU. Forced saving, redistribution, and nonlinear social security schemes. *Southern Economic Journal,* 76(1), 86-98, 2009.
2148. Natali HRITONENKO and Yuri YATSENKO. From linear to nonlinear utility in vintage capital models. *Mathematical Population Studies,* 15, 230-248, 2008.
2149. Natali HRITONENKO and Yuri YATSENKO. Integral equation of optimal replacement: Analysis and algorithms. *Applied Mathematical Modelling,* 33, 2737-2747, 2009.
2150. Pierre PESTIEAU and Motohiro SATO. Long-term care: the state, the market and the family. *Economica,* 75, 435-454, 2008.
2151. Joe THARAKAN and Jean-Philippe TROPEANO. On the impact of labor market matching on regional disparities. *Journal of Regional Science,* 49(1), 57-80, 2009.
2152. P. Jean-Jacques HERINGS and Arnald J. KANNING. Harmonization of private law on a global level. *International Review of Law and Economics,* 28, 256-262, 2008.
2153. P. Jean-Jacques HERINGS and Kirsten I.M. ROHDE. On the completeness of complete markets. *Economic Theory,* 37, 171-201, 2008.
2154. P. Jean-Jacques HERINGS, Gerard van der LAAN, Dolf TALMAN and Zaifu YANG. A fixed point theorem for discontinuous functions. *Operations Research Letters,* 36, 89-93, 2008.
2155. P. Jean-Jacques HERINGS, Gerard van der LAAN and Dolf TALMAN. The average tree solution for cycle-free graph games. *Games and Economic Behavior,* 62, 77-92, 2008.
2156. Philippe CHEVALIER and Jean-Christophe VAN den SCHRIECK. Optimizing the staffing and routing of small-size hierarchical call centers. *Production and Operations Management,* 17(3), 306-319, 2008.