Online Supplement to "A Flexible Architecture for Call Centers with Skill-Based Routing"

Benjamin Legros • Oualid Jouini • Yves Dallery

Laboratoire Génie Industriel, Ecole Centrale Paris, Grande Voie des Vignes, 92290 Châtenay-Malabry, France

benjamin.legros@centraliens.net • oualid.jouini@ecp.fr • yves.dallery@ecp.fr

International Journal of Production Economics, 159:192-207, 2015

This is an online supplement to the main paper, with the same title.

1 Optimization Heuristic

A coherent comparison between chaining and single pooling requires first the optimization of their total cost. In this section, we develop a greedy heuristic for the optimization step of the two models, and prove their efficiency. The heuristic is a simulation based optimization method. Recall that for each model, we optimize the total staffing cost under the constraints $W_i \leq W_i^*$, for $i = 0, 1, ..., n$.

The question addressed here is how can we compute the optimal number of agents in each team? In some particular cases the answer is simple. For example when the arrival and service rates are identical for all skills and when all skills have the same costs, we would create teams with the same number of agents. A more difficult situation is in the case of asymmetric arrival or service rates. For the simulation based optimization considered here, some information about the simulation process are as follows. For a given simulation with a given set of parameters, we consider a single replication that we run for a sufficiently long time. The lengths of the confidence intervals for the different performance measures derived by simulation are in the order of 10^{-4}. To obtain such confidence intervals, we simply gradually increase the replication length up to the point that ensures the accuracy objective. This implies that the simulation length may vary from one set of parameters to another. The total number of generated calls varies and is in the order of tens of millions. The delay to run a simulation also varies and is in the order of several minutes.

1.1 Single Pooling

In what follows we present three staffing heuristics, and then select the best one. The heuristics consist of adaptations of greedy and local search algorithms.

Algorithm 1: Decreasing Greedy in Team 0

Without customers 0, single pooling is simply an FD model. A first idea of staffing is then to use a decreasing greedy algorithm as follows. We start such that we have a collection of $n+1$ independent M/M/s queues. In each team i, the number of agents is the minimum required one to reach $W_i \leq W_i^* = 0.2$, for $i = 0, 1, ..., n$. In each iteration, we decrement the number of agents in team 0 by one, and evaluate all W_i, for $i = 0, 1, ..., n$. We stop the algorithm when all the service levels are no longer reached for the first time. We then consider the results of the before last iteration. Table 7 presents the results of the decreasing greedy algorithm in a single pooling model with 3 customer types, and compare it with the those of FF and FD models. The staffing level in team i is denoted by s_i, $i = 0, 1, 2$.

Table 1: Decreasing greedy in team 0 for single pooling ($n = 2$, $\mu_i = \mu_0 = 0.2$, $W_i^* = W_0^* = 0.2$, $i = 1, 2$)

					Single pooling			Total
λ_1	λ_2	λ_0	FF	FD	s_1	s_2	s_0	$s_0 + s_1 + s_2$
1	0.5	0.2	13	19	9	6	0	15
0.2	0.5	1	13	19	4	6	5	15
1	1	1	20	27	9	9	5	23
3	2	1	36	44	20	15	5	40
2	1	3	36	44	15	9	18	42
0.5	0.2	0.1	8	13	6	4	0	10
0.1	0.2	0.5	8	13	3	4	2	9
10	5	15	151	171	57	31	83	171
10	15	5	151	171	57	83	30	170
10	10	10	151	171	57	57	57	171

Algorithm 2: Increasing Greedy

Another idea is to proceed by introducing customers 0 in the system step by step. We start from an FD model with no customers 0, and we define the staffing level in team i such that $W_i \leq 0.2$, $i = 1, ..., n$ (team 0 is being empty). In each iteration, we increase λ_0 by a given small step value (we have chosen in the experiments a sufficiently small step of $\lambda_0/100$). If $W_0 \geq W_0^*$, we add one agent in team 0. If $W_0 \leq W_0^*$ and $W_i > W_i^*$ for some customer types, we then add an agent in the team with the highest W_i. We stop the algorithm once λ_0 reaches its value and the constraints $W_i \leq W_i^*$ are all satisfied, for $i = 0, 1, ..., n$. Table 8 provides the simulation results of this algorithm.

Algorithm 3: Increasing Greedy with No Agents in Team 0

The algorithm is identical to the previous one, expect that we force team 0 to be empty. Table 9 presents the simulated results for this algorithm.

From the results of all algorithms, we observe that the decreasing greedy algorithm (algorithm 1) is the worst. The reason is that it is not possible to increase or decrease the number of agents in

Table 2: Increasing greedy for single pooling ($n = 2$, $\mu_i = \mu_0 = 0.2$, $W_i^* = W_0^* = 0.2$, $i = 1, 2$)

λ_1	λ_2	λ_0	FF	FD	Single pooling s_1	s_2	s_0	Total $s_0 + s_1 + s_2$
1	0.5	0.2	13	19	9	6	0	15
0.2	0.5	1	13	19	6	7	0	15
1	1	1	20	27	10	10	1	21
3	2	1	36	44	21	16	1	38
2	1	3	36	44	17	12	8	37
0.5	0.2	0.1	8	13	6	4	0	10
0.1	0.2	0.5	8	13	4	5	0	9
10	5	15	151	171	61	36	64	161
10	15	5	151	171	60	85	18	163
10	10	10	151	171	60	61	41	162

Table 3: Increasing greedy with no Agents in team 0 for single pooling ($n = 2$, $\mu_i = \mu_0 = 0.2$, $W_i^* = W_0^* = 0.2$, $i = 1, 2$)

λ_1	λ_2	λ_0	FF	FD	Single pooling s_1	s_2	s_0	Total $s_0 + s_1 + s_2$
1	0.5	0.2	13	19	9	6	0	15
0.2	0.5	1	13	19	6	7	0	15
1	1	1	20	27	11	10	0	21
3	2	1	36	44	22	16	0	38
2	1	3	36	44	21	16	0	37
0.5	0.2	0.1	8	13	6	4	0	10
0.1	0.2	0.5	8	13	4	5	0	9
10	5	15	151	171	93	68	0	161
10	15	5	151	171	70	93	0	163
10	10	10	151	171	81	81	0	162

a regular team i ($i = 1, ..., n$). Many effective configurations could not then be reached under this algorithm. The other two algorithms are equivalent in terms of the total number of agents in our simulation experiments. We have chosen to use algorithm 2 in the experiments of Section 5 of the main paper.

We go further in order to check the quality of algorithm 2. In Table 4, we provide optimization results using algorithm 2 and also using other configurations with one agent in less. We observe that these other configurations do not allow to satisfy all the constraints $W_i \leq W_i^* = 0.2$, for $i = 0, 1, ..., n$, which proves the efficiency of algorithm 2.

1.2 Chaining

We also use a greedy algorithm to optimize the staffing cost of chaining. The simulation results reveal that increasing and decreasing greedy algorithms are efficient and very similar if we start the optimization heuristic with a good initialization of the team sizes. We choose to use a decreasing greedy algorithm since it is faster than an increasing greedy one (no need to increase the λ_0 with a high number of small steps).

Table 4: Efficiency of algorithm 2 ($n = 2$, $\mu_i = \mu_0 = 0.2$, $W_i^* = W_0^* = 0.2$, $i = 1, 2$)

	s_1	s_2	s_0	W_1	W_2	W_0	Constraints
$\lambda_1 = 1$,	9	6	0	0.123	0.126	0.019	Satisfied (algorithm 2)
$\lambda_2 = 0.5$,	8	6	0	0.143	0.407	0.057	Not satisfied (one agent in less)
$\lambda_0 = 0.2$	7	7	0	0.898	0.049	0.036	Not satisfied (one agent in less)
	9	5	0	0.338	0.141	0.051	Not satisfied (one agent in less)
$\lambda_1 = 3$,	21	16	1	0.174	0.154	0.110	Satisfied (algorithm 2)
$\lambda_2 = 2$,	21	16	0	0.204	0.191	0.205	Not satisfied (one agent in less)
$\lambda_0 = 1$	22	15	0	0.145	0.281	0.209	Not satisfied (one agent in less)
	20	16	1	0.290	0.177	0.183	Not satisfied (one agent in less)
	21	15	1	0.195	0.269	0.195	Not satisfied (one agent in less)
$\lambda_1 = 10$,	60	85	18	0.152	0.171	0.167	Satisfied (algorithm 2)
$\lambda_2 = 15$,	60	85	17	0.166	0.207	0.177	Not satisfied (one agent in less)
$\lambda_0 = 5$	60	84	18	0.171	0.218	0.212	Not satisfied (one agent in less)
	59	85	18	0.205	0.172	0.180	Not satisfied (one agent in less)
	59	86	17	0.210	0.168	0.178	Not satisfied (one agent in less)

The method is as follows. In each team i ($i = 0, 1, ..., n$), we start with the worst (over-estimated) staffing level s_i computed from an FD model. In order to take into account the chaining configuration, i.e., the fact that customers type $i-1$ can be routed to team i and customers type i can be routed to team $i+1$, we adjust the initial staffing levels from s_i to s'_i for team i ($i = 0, 1, ..., n$). We use the method suggested by Wallace and Whitt (2005). The corrected staffing level s'_i is given by

$$s'_i = s_i - R_{i,i+1} + R_{i-1,i}, \tag{1}$$

for $i = 0, 1, ..., n$, where $R_{i,j} = \frac{s_i s_j}{s - s_i}$. This number is that of agents of team i who could go to team j, $i, j = 0, 1, ..., n$ and $i \neq j$. Using Equation (1), s'_i may not be an integer. We then round it to the nearest integer above.

2 A Fixed Point Approximation for SP

We develop in this section an approximate numerical method for a particular case of single pooling. We consider Markovian assumptions with an arbitrary number of skills. There are $n+1$ customer types (type 0, and types $1, 2, ..., n$), n teams (no team 0), $n \geq 1$. The arrival rates are λ_0 and $\lambda_i = \lambda$ for $i = 1, ..., n$, and the service rates are $\mu_i = \mu$ for $i = 0, 1, ..., n$. Since the configuration is symmetric, we consider the same staffing level s in each team. In what follows, we develop an approximation to compute the expected waiting time of regular customers type i, for $i = 1, ..., n$. The approximation is based on a Markov chain approach and a fixed point algorithm.

One can see that our model can be divided into n identical sub-systems. It suffices then to focus

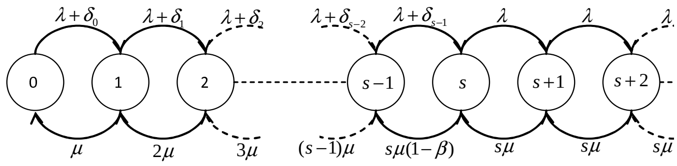


Figure 1: Markov chain associated to a sub-system of single pooling

on the performance analysis of one of these sub-systems. A sub-system is a simple queueing system with s servers and an infinite queue. Two types of customers arrive to this sub-system: customers type i with a Poisson process with rate λ and customers type 0 with a general arrival process with mean arrival rate $\frac{\lambda_0}{n}$. (The arrival process of customers 0 to the whole system is Poisson. However, it becomes a general process at each sub-system because of the routing rules.)

Recall that customers 0 wait in their own queue before being routed to one of the sub-systems for an immediate processing. Because of the routing rule, customers 0 can be routed to a sub-system only if the number of customers in the sub-system is less or equal to $s-1$. Also since we route customers 0 to the one of the less busiest sub-systems (with an equiprobable choice), the arrival rate of customers 0 is decreasing in the number of busy servers in a sub-system and it becomes 0 when all the s servers become busy.

Let us now define, for a sub-system, the stochastic process $\{E(t), t \geq 0\}$, where $E(t)$ denotes the number of customers in the system (queue + service). Note that the customers in the queue are only the regular customers, and those in service can be both regular or type 0 customers. We approximate customers 0 inter-arrival times by an exponential distribution with state-dependent rates. Since inter-arrival and service times are Markovian, $\{E(t), t \geq 0\}$ is a Markov chain as shown in Figure 1. The arrival rate δ_k denotes the state-dependent arrival rate of customers 0 when the number of customers in the sub-system is k, for $k = 0, ..., s-1$ (no customers 0 arrive at the sub-system for $k \geq s$).

Assume that exactly s customers are in the sub-system and that a service completion occurs first before the next arrival epoch of a regular customer at this sub-system (Figure 1). Therefore, two possibilities may happen. The first possibility corresponds to the case of an empty queue 0. We then move to state $s-1$. The second one corresponds to the case of a non-empty queue 0. We then stay in state s, because the server who just became idle immediately takes the customer 0 in the head of queue 0 into service. Let us denote by β the probability that queue 0 is not empty. Then the rate to move from state s to sate $s-1$ in the Markov chain is $s\mu(1-\beta)$.

Let us now assume that the stability condition of a sub-system holds, i.e., $\lambda + \frac{\lambda_0}{n} < s\mu$, and denote the stationary probabilities of the system states by π_k, for $k \geq 0$. We may then write

$$\pi_k = \frac{\prod_{i=0}^{k-1}(\lambda + \delta_i)}{k!\mu^k}\pi_0, \tag{2}$$

for $1 \leq k \leq s-1$, and

$$\pi_{s+k} = (\frac{\lambda}{s\mu})^k \frac{\prod_{i=0}^{s-1}(\lambda + \delta_i)}{s!\mu^s(1-\beta)}\pi_0, \tag{3}$$

for $k \geq 0$. Since all probabilities sum up to one, we obtain

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{s-1} \frac{\prod_{i=0}^{k-1}(\lambda+\delta_i)}{k!\mu^k} + \frac{\prod_{i=0}^{s-1}(\lambda+\delta_i)}{s!\mu^s(1-\beta)} \frac{1}{1-\frac{\lambda}{s\mu}}}. \tag{4}$$

The difficulty to compute the stationary probabilities is that we do not have the values of δ_k ($k = 0, ..., s-1$) and β. We use a fixed point algorithm to jointly compute them with the stationary probabilities. Let us now write δ_0, the arrival rate of customers 0 at a given sub-system when this sub-system is empty, as a function of the stationary probabilities of this sub-system. We use here a second approximation. We indeed assume that the states of the sub-systems are independent, which is not true. Assume that our sub-system is the only one that is empty, i.e., each one of the other $n-1$ sub-systems have at least one customer in the system (queue + service). Using the approximation this occurs with probability $(1-\pi_0)^{n-1}$, then δ_0 is simply λ_0 in that case. Assume now that our sub-system and only another one are empty. Then δ_0 is $\frac{\lambda_0}{2}$ (equiprobable routing of customers 0 to one of the less busiest sub-systems). This occurs with probability $\pi_0(1-\pi_0)^{n-2}$ and there are $\binom{n-1}{1}$ combinations (where $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ for $0 \leq k \leq n$). Continuing with the same reasoning and averaging over all possibilities, we obtain

$$\delta_0 = \lambda_0 \sum_{j=0}^{n-1} \frac{1}{j+1}\binom{n-1}{j}\pi_0^j(1-\pi_0)^{n-1-j}. \tag{5}$$

Since $\frac{1}{j+1}\binom{n-1}{j} = \frac{1}{n}\binom{n}{j+1}$, Equation (5) becomes

$$\delta_0 = \frac{\lambda_0}{n}\sum_{j=0}^{n-1}\binom{n}{j+1}\pi_0^j(1-\pi_0)^{n-1-j} = \frac{\lambda_0}{n\pi_0}\sum_{j=1}^{n}\binom{n}{j}\pi_0^j(1-\pi_0)^{n-j},$$

which leads to

$$\delta_0 = \lambda_0\frac{1-(1-\pi_0)^n}{n\pi_0}. \tag{6}$$

In the same way, we obtain

$$\delta_k = \lambda_0 \frac{\left(1 - \sum_{j=1}^{k-1} \pi_j\right)^n - \left(1 - \sum_{j=1}^{k} \pi_j\right)^n}{n \pi_k}, \tag{7}$$

for $1 \leq k \leq s-1$. Let us now give the expression of β as a function of the stationary probabilities π_k, $k \geq 0$. Since the mean arrival rate of customers 0 at our sub-system is $\frac{\lambda_0}{n}$, we have

$$\sum_{k=0}^{s-1} \pi_k \delta_k + \beta s \mu = \frac{\lambda_0}{n}, \tag{8}$$

which implies

$$\beta = \frac{\frac{\lambda_0}{n} - \sum_{k=0}^{s-1} \pi_k \delta_k}{s\mu}. \tag{9}$$

In summary, from the one hand, Equations (2)-(4) give the stationary probabilities π_k ($k \geq 0$) as a function of δ_k ($0 \leq k \leq s-1$) and β. From the other hand, Equations (6), (7) and (9) give δ_k ($0 \leq k \leq s-1$) and β as a function of π_k ($k \geq 0$). As a consequence, we have a fixed point. We propose the following fixed point algorithm to compute it. In the first iteration, we choose $\delta_0 = \frac{\lambda_0}{n}$, $\delta_k = 0$ for $1 \leq k \leq s-1$, and $\beta = 0$. Then we compute π_k ($k \geq 0$) using Equations (2)-(4). From these π_k, we next compute the new values of δ_k ($0 \leq k \leq s-1$) and β using Equations (6), (7) and (9). In the second iteration, we use the latter values of δ_k and β to compute π_k. From these new π_k, we compute the new values of δ_k and β. We do the same in the third iteration, and so on. We stop the algorithm when the values of π_k ($k \geq 0$), δ_k ($0 \leq k \leq s-1$) and β converge to their limits with a given predefined precision (we have chosen a precision of 10^{-6} in the numerical experiments below). Proposition 1 proves the convergence of the fixed point algorithm.

Proposition 1 *The fixed point algorithm always converges.*

Proof. We use the Brouwer's theorem to prove the convergence. The Brouwer's theorem states that any continuous function from a convex compact subset K of an Euclidean space to itself has at least one fixed point. In what follows, we prove that the conditions of the Brouwer's theorem hold in our context.

After k iterations, the fixed point algorithm gives the vector $(\pi_0, \pi_1, \pi_2, \cdots, \pi_c)_k$ belonging to a convex compact, $[0;1]^{s+1}$, that is included in an Euclidean space, $\mathbb{R}^{s+1}$. From Equations (2)-(9), it is obvious to see that the function that allows to calculate $(\pi_0, \pi_1, \pi_2, \cdots, \pi_c)_{k+1}$ (iteration $k+1$) as a function of $(\pi_0, \pi_1, \pi_2, \cdots, \pi_c)_k$ is continuous (combination of continuous functions), for $\pi_k \neq 0$ ($k = 0, ..., s-1$). In what follows, we prove that this function is continuous in $\pi_k = 0$

$(k = 0, ..., s-1)$ by prolongation. From Equations (6) and (7), we have

$$\delta_k = \lambda_0 \frac{\left(1 - \sum_{j=1}^{k-1} \pi_j\right)^n - \left(1 - \sum_{j=1}^{k} \pi_j\right)^n}{n\pi_k} = \lambda_0 \frac{\left(1 - \sum_{j=1}^{k-1} \pi_j\right)^n}{n\pi_k} \left(1 - \frac{\left(1 - \sum_{j=1}^{k-1} \pi_j\right)^n}{\left(1 - \sum_{j=1}^{k-1} \pi_j\right)^n}\right),$$

for $k = 0, ..., s-1$, where by convention an empty sum is equal to 0. Calculating further, we obtain

$$\delta_k = \lambda_0 \frac{\left(1 - \sum_{j=1}^{k-1} \pi_j\right)^n}{n\pi_k} \left(1 - \left(1 - \frac{\pi_k}{1 - \sum_{j=1}^{k-1} \pi_j}\right)^n\right),$$

for $k = 0, ..., s-1$. The Taylor expansion of δ_k as a function of π_k in the neighborhood of 0 is

$$\delta_k = \lambda_0 \frac{\left(1 - \sum_{j=1}^{k-1} \pi_j\right)^n}{n\pi_k} \left(1 - (1 - n\, o(\pi_k))\right) = \lambda_0 \left(1 - \sum_{j=1}^{k-1} \pi_j\right)^n + o(1),$$

where $o(1)$ is a function that converges to a finite limit as π_k goes to 0, for $k = 0, ..., s-1$. Since $\lambda_0 \left(1 - \sum_{j=1}^{k-1} \pi_j\right)^n$ is finite, δ_k is continuous by prolongation in $\pi_k = 0$, for $k = 0, ..., s-1$.

It remains now to focus on the issue for $\beta = 1$ in Equation (4). This case of $\beta = 1$ can not happen. The proof is as follows. Assume that $\beta = 1$. Equation (8) thus leads to $s\mu = \frac{\lambda_0}{n} - \sum_{k=0}^{s-1} \pi_k \delta_k$. Since δ_k and π_k $(0 \leq k \leq s-1)$ are positive, $s\mu \leq \frac{\lambda_0}{n}$. As a consequence the sub-system is unstable, which is absurd. This completes the proof of the convergence of the fixed point algorithm. □

Having in hand the stationary probabilities, we next compute for the regular customers the expected waiting time in the queue and the probability of delay. Recall that all sub-systems are identical because of the symmetry in the parameters. Using Little's law, the expected waiting time of a regular customer type i $(i = 1, ..., n)$ is given by

$$W_i = \frac{1}{\lambda} \sum_{k=1}^{\infty} k\pi_{s+k}, \tag{10}$$

for $i = 1, ..., n$, and its probability of delay denoted by $P_{D,i}$ is

$$P_{D,i} = \sum_{k=1}^{\infty} \pi_{s+k}, \tag{11}$$

for $i = 1, ..., n$. The approximation for both W_i and $P_{D,i}$ works very well for the regular customer types, however it does not for customers 0 because of their complex routing. The comparison between the approximate results using the fixed point algorithm and the exact ones using simulation

Table 5: Fixed point approximation, $\mu = 0.2$

	λ	λ_0	s	$\frac{(\lambda+\lambda_0)/n}{s\mu}$	W_i Simulation	W_i Approximation	P_D Simulation	P_D Approximation
$n = 1$	0.35	0.35	5	70%	0.581	0.581	37.78%	37.78%
	0.475	0.475	5	95%	1.672	1.672	87.78%	87.78%
	1.4	1.4	20	70%	0.035	0.035	9.36%	9.36%
	1.9	1.9	20	95%	0.359	0.359	75.54%	75.54%
	3.8	0	20	95%	3.777	3.777	75.54%	75.54%
$n = 2$	0.35	0.7	5	70%	0.436	0.435	29.74%	28.28%
	0.475	0.95	5	95%	1.623	1.623	87.51%	85.23%
	1.4	2.8	20	70%	0.003	0.0027	0.74%	0.72%
	1.9	3.8	20	95%	0.320	0.290	61.03%	60.98%
	3.8	0	20	95%	3.777	3.777	75.54%	75.54%
$n = 5$	0.35	1.75	5	70%	0.290	0.288	19.06%	18.76%
	0.475	2.375	5	95%	1.556	1.550	81.84%	81.38%
	1.4	7	20	70%	0.001	0.001	0.28%	0.27%
	1.9	9.5	20	95%	0.205	0.204	42.99%	42.97%
	3.8	0	20	95%	3.777	3.777	75.54%	75.54%
$n = 10$	0.35	3.5	5	70%	0.259	0.252	17.21%	16.39%
	0.475	4.75	5	95%	1.516	1.504	79.07%	78.96%
	1.4	14	20	70%	0.0001	0.0001	0.23%	0.23%
	1.9	19	20	95%	0.167	0.167	35.23%	35.21%
	3.8	0	20	95%	3.777	3.777	75.54%	75.54%

are given in Table 13. Note that in the extreme situations of $n = 1$ or $\lambda_0 = 0$, our method gives the exact results. Table 13 reveals that our approximation yields very accurate estimates for W_i and $P_{D,i}$.

3 Impact of Abandonment

The experiments of Tables 6-9 are associated to Figures 10(a)-10(d) of the main paper, respectively. The experiments of Tables 10-13 are associated to Figures 11(a)-11(d) of the main paper, respectively.

Table 6: Impact of p ($\mu_i = \mu_0 = 0.2$, $W_0^* = W_i^* = 0.2$, $\sum_{i=0}^{4} \lambda_i = 8$, $\gamma_i = \gamma_0 = \gamma$, $i = 1, ..., 4$, $p' = 20\%$, $U = V = 1$)

	p	Chaining t=0%	t=5%	t=10%	t=25%	t=50%	SP	Crossing value (Chaining = SP)
$\gamma = 0$	0%	49	50.95	52.9	58.75	68.5	60	t=28.21%
	10%	49	50.7	52.4	57.5	66	56	t=20.58%
	25%	48	49.3	50.6	54.5	61	52	t=15.38%
	50%	49	49.9	50.8	53.5	58	52	t=16.67%
	75%	51	51.55	52.1	53.75	56.5	51	t=0%
	90%	51	51.3	51.6	52.5	54	51	t=0%
$\gamma = 0.1$	0%	47	48.55	50.1	54.75	62.5	56	t=29.03%
	10%	46	47.5	49	53.5	61	52	t=20.00%
	25%	46	47.3	48.6	52.5	59	52	t=23.08%
	50%	46	46.9	47.8	50.5	55	51	t=27.78%
	75%	48	48.5	49	50.5	53	50	t=20.00%
	90%	49	49.35	49.7	50.75	52.5	49	t=0.00%
$\gamma = 0.2$	0%	44	45.5	47	51.5	59	52	t=26.67%
	10%	42	43.35	44.7	48.75	55.5	48	t=22.22%
	25%	44	45.25	46.5	50.25	56.5	48	t=16.00%
	50%	44	44.8	45.6	48	52	48	t=25.00%
	75%	45	45.4	45.8	47	49	48	t=37.50%
	90%	45	45.2	45.4	46	47	48	t=75.00%

4 Impact of the Number of Skills

The experiments of Tables 14 and 15 are associated to Figures 11(a) and 11 (b) of the main paper, respectively.

5 Impact of the Agent Costs

We provide here the details of the analysis related to Section 5.8 of the main paper. We change the cost framework such that the cost of regular agents are no longer identical. Our objective is to examine the impact of an asymmetry in the costs on the comparison between the staffing costs of SP and chaining. Consider a call center with $n + 1$ skills; the easy skill 0 and the regular skills i, for $i = 1, ..., n$. We define a new parameter c for the cost framework. The higher is c, the higher is the asymmetry in the agent costs, and viceversa.

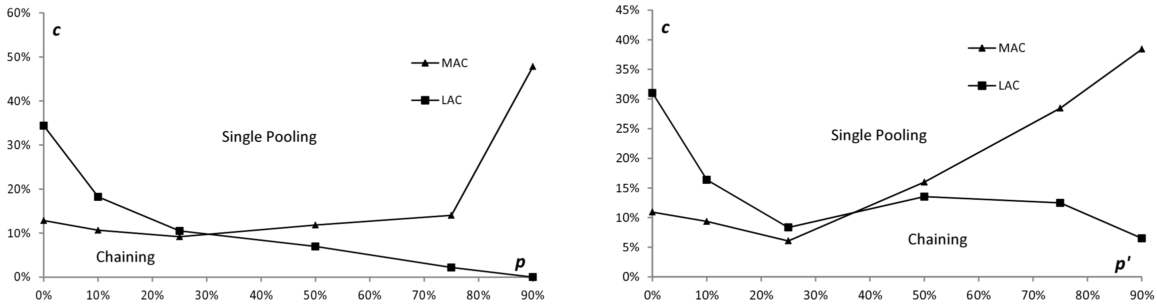
In SP, an agent has skills i and 0, for $i = 1, ..., n$. We assume that the cheapest agents is that with skills n and 0. She costs 1. Then, an agent with skills $n-1$ and 0 costs $(1+c)$, ..., and an agent with skills 1 and 0 costs $(1 + c)^{n-1}$. In chaining, an agent has skills i and j, for $i \neq j \in \{0, ..., n\}$. An agent with skills i and 0 costs, as in SP, $(1 + c)^{n-i}$, for $i = 1, ..., n$. An agent with skills i and j costs $(1 + c)^{n-i} \times (1 + c)^{n-j}$, for $i \neq j \in \{1, ..., n\}$. In the experiments below, we consider a call center with $n = 4$ regular skills and skill 0.

With this new cost framework, the choice of the two skills in chaining teams is very important. It

Table 7: Impact of p' ($\lambda_i = \lambda_0 = 2$, $\sum_{i=0}^{4} \frac{1}{\mu_i} = 25$, $W_0 = W_i^* = 0.2$, $i = 1, ..., 4$, $p = 20\%$, $U = V = 1$)

	p'	Chaining t=0%	t=5%	t=10%	t=25%	t=50%	SP	Crossing value (Chaining = SP)
$\gamma = 0$	0%	60	62.45	64.9	72.25	84.5	72	t=24.49%
	10%	59	60.95	62.9	68.75	78.5	67	t=20.51%
	25%	58	59.65	61.3	66.25	74.5	62	t= 12.12%
	50%	60	61.05	62.1	65.25	70.5	65	t=23.81%
	75%	61	61.6	62.2	64	67	68	t=58.33%
	90%	65	65.25	65.5	66.25	67.5	69	t=80.00%
$\gamma = 0.1$	0%	57	59.1	61.2	67.5	78	67	t=23.81%
	10%	57	59.05	61.1	67.25	77.5	65	t=19.51%
	25%	57	58.75	60.5	65.75	74.5	61	t=11.43%
	50%	59	60.2	61.4	65	71	64	t=20.83%
	75%	60	60.75	61.5	63.75	67.5	64	t=26.67%
	90%	61	61.25	61.5	62.25	63.5	62	t=20.00%
$\gamma = 0.2$	0%	55	57.05	59.1	65.25	75.5	60	t=12.20%
	10%	55	56.95	58.9	64.75	74.5	60	t=12.82%
	25%	55	56.75	58.5	63.75	72.5	58	t=8.57%
	50%	55	56.1	57.2	60.5	66	59	t=18.18%
	75%	57	57.7	58.4	60.5	64	60	t=21.43%
	90%	59	59.25	59.5	60.25	61.5	59	t=0.00%

has however no impact on SP. In chaining, the choice of one team skills may create various situations with different asymmetry levels in the cost parameters. For instance, the most asymmetrical case would be with the chain $3-1-2-4-0$ where the individual costs vary from 1 to $(1+c)^5$. The most symmetrical one would be with the chain $2-3-4-1-0$ where the individual costs vary from $(1+c)$ to $(1+c)^3$. in what follows, we compare between the staffing costs of SP and chaining. For the latter, we consider both cases, the most asymmetric one referred to as MAC, and the less asymmetric one referred to as LAC. The results are given in Tables 16-18 and Figure 2.



(a) Preference zone ($\mu_0 = \mu_i = 0.2$ for $i = 1, ..., 4$, $\sum_{i=0}^{4} \lambda_i = 8$, $U = V = 1$, $n = 4$, $W_0 = W_i^* = 0.2$)

(b) Preference zone ($\lambda_i = \lambda_0 = 2$, $\sum_{i=0}^{4} \frac{1}{\mu_i} = 25$, $W_0 = W_i^* = 0.2$, $i = 1, ..., 4$, $p = 20\%$, $U = V = 1$, $n = 4$)

Figure 2: Preference zone

In Tables 16 and 17, we give the overall staffing costs as a function of c for SP, MAC and LAC for different values of p and p', respectively. We also detail the team staffing levels for the case $c = 10\%$. For $c = 20\%$ and $c = 30\%$ the detailed staffing levels are almost identical to the those

Table 8: Impact of V ($\lambda_0 = 2$, $\mu_0 = \mu_i = 0.2$, $\sum_{i=0}^{4} \lambda_i = 8$, $W_0 = W_i^* = 0.2$, $i = 1, ..., 4$, $p = 25\%$, $p' = 20\%$, $U = 1$)

	V	Chaining t=0%	t=5%	t=10%	t=25%	t=50%	SP	Crossing value (Chaining = SP)
$\gamma = 0$	1	48	49.3	50.6	54.5	61	52	t=15.38%
	2	49	50.3	51.6	55.5	62	53	t=15.38%
	3	49	50.25	51.5	55.25	61.5	52	t=12.00%
	5	50	51.25	52.5	56.25	62.5	52	t=8.00%
$\gamma = 0.1$	1	46	47.3	48.6	52.5	59	52	t=23.08%
	2	46	47.25	48.5	52.25	58.5	51	t=20.00%
	3	46	47.15	48.3	51.75	57.5	51	t=21.74%
	5	46	47.1	48.2	51.5	57	51	t=22.73%
$\gamma = 0.2$	1	44	45.25	46.5	50.25	56.5	48	t=16.00%
	2	45	46.25	47.5	51.25	57.5	51	t=24.00%
	3	45	46.15	47.3	50.75	56.5	51	t=26.09%
	5	46	47.05	48.1	51.25	56.5	52	t=28.57%

Table 9: Impact of U ($\mu_0 = 0.2$, $\lambda_0 = 4$, $\lambda_i = 1$, $W_0 = W_i^* = 0.2$, $i = 1, ..., 4$, $\sum_{i=0}^{4} \frac{1}{\mu_i} = 25$, $p' = 20\%$, $p = 50\%$, $V = 1$)

	U	Chaining t=0%	t=5%	t=10%	t=25%	t=50%	SP	Crossing value (Chaining = SP)
$\gamma = 0$	1	49	50.25	51.5	55.25	61.5	52	t=12.00%
	2	49	49.75	50.5	52.75	56.5	53	t=26.67%
	3	50	51.65	52.3	54.25	57.5	52	t=7.69%
	5	52	52.65	53.3	55.25	58.5	52	t=0.00%
$\gamma = 0.1$	1	46	46.9	47.8	50.5	55	51	t=27.78%
	2	48	48.9	49.8	52.5	57	51	t=23.53%
	3	50	50.85	51.7	54.25	58.5	51	t=5.88%
	5	51	51.75	52.5	54.75	58.5	51	t=0.00%
$\gamma = 0.2$	1	44	44.8	45.6	48	52	48	t=25.00%
	2	47	47.85	48.7	51.25	55.5	48	t=5.88%
	3	49	49.8	50.6	53	57	49	t=0.00%
	5	49	49.65	50.3	52.25	55.5	49	t=0.00%

for the case $c = 10\%$. We observe that in SP we do not have enough flexibility to act on the team staffing levels so as to reduce the overall costs. The explanation is that regular customers have access to only one team.

For Chaining, MAC and LAC configurations have different team staffing levels and total staffing costs. We observe that MAC is better when p or p' is high (high predominance of the easy skill or slow served customers 0). When p or p' is high, the easy skill requires an important number of agents. It is then interesting to organize the teams such that the cheap skills are handled by the biggest teams (as MAC allows to do). This restricts the staffing levels of the expensive teams. We also observe that LAC is better when p or p' corresponds to a symmetrical situation ($p = 25\%$ or $p' = 25\%$). Under a symmetrical situation of arrival and service rates, the team staffing levels are likely to be balanced. Since $(1 + c)^n$ is convex in n, a symmetrical situation of costs is then

Table 10: Impact of p ($\mu_i = \mu_0 = 0.2$, $W_0^* = W_i^* = 0.2$, $\sum_{i=0}^{4} \lambda_i = 8$, $i = 1, ..., 4$, $p' = 20\%$, $U = V = 1$)

	p	Chaining t=0%	t=5%	t=10%	t=25%	t=50%	SP	Crossing value (Chaining = SP)
$\gamma_0 = 0.1$ $\gamma_i = 0$	0%	49	50.95	52.9	58.75	68.5	60	t=28.21%
	10%	48	49.6	51.2	56	64	58	t=31.25%
	25%	48	49.25	50.5	54.25	60.5	56	t=32.00%
	50%	48	48.9	49.8	52.5	57	54	t=33.33%
	75%	48	48.6	49.2	51	54	52	t=33.33%
	90%	49	49.3	49.6	50.5	52	51	t=33.33%
$\gamma_i = \gamma_0 = 0.1$	0%	47	48.55	50.1	54.75	62.5	56	t=29.03%
	10%	46	47.5	49	53.5	61	52	t=20.00%
	25%	46	47.3	48.6	52.5	59	52	t=23.08%
	50%	46	46.9	47.8	50.5	55	51	t=27.78%
	75%	48	48.5	49	50.5	53	50	t=20.00%
	90%	49	49.35	49.7	50.75	52.5	49	t=0.00%
$\gamma_0 = 0$ $\gamma_i = 0.1$	0%	47	48.55	50.1	54.75	62.5	56	t=29.03%
	10%	48	49.3	50.6	54.5	61	54	t=23.08%
	25%	48	49.25	50.5	54.25	60.5	52	t=16.00%
	50%	48	48.85	49.7	52.25	56.5	52	t=23.53%
	75%	49	49.75	50.5	52.75	56.5	51	t=13.33%
	90%	49	49,3	49,6	50,5	52	49	t=0,00%

preferred to another with a mix of expensive and cheap teams.

In Figure 2 and Table 18, we present the switching curves under which chaining (MAC or LAC) is less expensive than SP, and above which the opposite is true. Figure 2(a) reveals that SP is better than LAC as p increases. This agrees with the results in the main paper where also the cost framework is symmetrical. We also observe that MAC is preferred to SP under asymmetrical situations of arrival rates. The reason is again related to the flexibility of MAC for creating large cheap teams.

As for the impact of p', we observe from Figure 2(b) the opposite conclusions as those drawn above for p. The switching curve for the comparison between SP and MAC is similar to the one obtained in the main paper. The explanation is still related both to the blocking effect and the preference for asymmetrical cost framework for MAC, when the service rates are different. We also observe that the switching curve for the comparison between SP and LAC is not regular. The reason is that there are two competing phenomena. From the one hand, the blocking effect due to customers 0 implies a preference for LAC. From the other hand, the performance of chaining deteriorates under a symmetrical cost framework and asymmetrical service rate situations.

Table 11: Impact of p' ($\lambda_0 = \lambda_i = 2$, $W_0^* = W_i^* = 0.2$, $\sum_{i=0}^{4} \frac{1}{\mu_i} = 25$, $i = 1, ..., 4$, $p = 20\%$, $U = V = 1$)

	p'	Chaining t=0%	t=5%	t=10%	t=25%	t=50%	SP	Crossing value (Chaining = SP)
$\gamma_0 = 0.1$, $\gamma_i = 0$	0%	60	62.25	64.5	71.25	82.5	72	t=26.67%
	10%	58	60	62	68	78	66	t=20.00%
	25%	58	59.5	61	65.5	73	65	t=23.33%
	50%	58	59.2	60.4	64	70	65	t=29.17%
	75%	58	58.6	59.2	61	64	66	t=66.67%
	90%	59	59.25	59.5	60.25	61.5	68	t=180.00%
$\gamma_0 = \gamma_i = 0.1$	0%	57	59.1	61.2	67.5	78	67	t=23.81%
	10%	57	59.05	61.1	67.25	77.5	65	t=19.51%
	25%	57	58.75	60.5	65.75	74.5	61	t=11.43%
	50%	59	60.2	61.4	65	71	64	t=20.83%
	75%	60	60.75	61.5	63.75	67.5	64	t=26.67%
	90%	61	61.25	61.5	62.25	63.5	62	t=20.00%
$\gamma_0 = 0$, $\gamma_i = 0.1$	0%	57	59.15	61.3	67.75	78.5	66	t=20.93%
	10%	56	57.85	59.7	65.25	74.5	61	t=13.51%
	25%	57	58.35	59.7	63.75	70.5	61	t=14.81%
	50%	58	59.05	60.1	63.25	68.5	63	t=23.81%
	75%	60	60.75	61.5	63.75	67.5	65	t=33.33%
	90%	63	63.25	63.5	64.25	65.5	65	t=40.00%

References

Wallace, R. and Whitt, W. (2005). A Staffing Algorithm for Call Centers with Skill-Based Routing. *Manufacturing & Service Operations Management*, 7:276–294.

Table 12: Impact of V ($\lambda_0 = 2$, $W_0^* = W_i^* = 0.2$, $\mu_i = \mu_0 = 0.2$, $i = 1, ..., 4$, $p = 25\%$, $p' = 20\%$, $U = 1$)

		Chaining					SP	Crossing value
	V	t=0%	t=5%	t=10%	t=25%	t=50%		(Chaining = SP)
$\gamma_0 = 0.1$, $\gamma_i = 0$	1	48	49.4	50.8	55	62	53	t=17.86%
	2	49	50.3	51.6	55.5	62	53	t=15.38%
	3	50	51.15	52.3	55.75	61.5	53	t=13.04%
	5	52	53.05	54.1	57.25	62.5	54	t=9.52%
$\gamma_0 = \gamma_i = 0.1$	1	46	47.3	48.6	52.5	59	52	t=23.08%
	2	46	47.25	48.5	52.25	58.5	51	t=20.00%
	3	46	47.15	48.3	51.75	57.5	51	t=21.74%
	5	46	47.1	48.2	51.5	57	51	t=22.73%
$\gamma_0 = 0$, $\gamma_i = 0.1$	1	47	48.35	49.7	53.75	60.5	52	t=18.52%
	2	47	48.15	49.3	52.75	58.5	51	t=17.39%
	3	47	48	49	52	57	51	t=20.00%
	5	47	47.85	48.7	51.25	55.5	52	t=29.41%

Table 13: Impact of U ($\lambda_0 = 4$, $W_0^* = W_i^* = 0.2$, $\mu_0 = 0.2$, $i = 1, ..., 4$, $p = 50\%$, $p' = 20\%$, $V = 1$)

		Chaining					SP	Crossing value
	U	t=0%	t=5%	t=10%	t=25%	t=50%		(Chaining = SP)
$\gamma_0 = 0.1$, $\gamma_i = 0$	1	47	47.95	48.9	51.75	56.5	52	t=26.32%
	2	48	48.8	49.6	52	56	53	t=31.25%
	3	48	48.85	49.7	52.25	56.5	52	t=23.53%
	5	48	48.9	49.8	52.5	57	52	t=22.22%
$\gamma_i = \gamma_0 = 0.1$	1	46	46.9	47.8	50.5	55	51	t=27.78%
	2	48	48.9	49.8	52.5	57	51	t=23.53%
	3	50	50.85	51.7	54.25	58.5	51	t=5.88%
	5	51	51.75	52.5	54.75	58.5	51	t=0.00%
$\gamma_0 = 0$, $\gamma_i = 0.1$	1	48	48.85	49.7	52.25	56.5	52	t=23.53%
	2	48	48.7	49.4	51.5	55	49	t=7.14%
	3	49	49.9	50.8	53.5	58	50	t=5.56%
	5	50	51	52	55	60	51	t=5.00%

Table 14: Impact of the number of skills ($\mu_i = \mu_0 = 0.2$, $W_0^* = W_i^* = 0.2$, $\sum_{i=0}^{n} \lambda_i / N = 2$, $i = 1, ..., n$, $U = V = 1$)

	p	Chaining t=0%	t=5%	t=10%	t=25%	t=50%	SP	Crossing value (Chaining = SP)
$N = 3$	0%	36	37.8	39.6	45	54	40	t=11.11%
	10%	37	38.05	39.1	42.25	47.5	40	t=14.29%
	25%	37	37.75	38.5	40.75	44.5	39	t=13.33%
	50%	37	37.4	37.8	39	41	37	t=0.00%
	75%	36	36.15	36.3	36.75	37.5	36	t=0.00%
	90%	36	36.05	36.1	36.25	36.5	36	t=0.00%
	100%	36	36	36	36	36	36	t=0.00%
$N = 4$	0%	48	49.95	51.9	57.75	67.5	54	t=15.38%
	10%	48	49.45	50.9	55.25	62.5	52	t=13.79%
	25%	47	48.15	49.3	52.75	58.5	50	t=13.04%
	50%	48	48.8	49.6	52	56	48	t=0.00%
	75%	48	48.5	49	50.5	53	48	t=0.00%
	90%	47	47.25	47.5	48.25	49.5	47	t=0.00%
	100%	47	47	47	47	47	47	t=0.00%
$N = 5$	0%	60	62.55	65.1	72.75	85.5	68	t=15.69%
	10%	59	61	63	69	79	67	t=20.00%
	25%	58	59.6	61.2	66	74	64	t=18.75%
	50%	59	60.1	61.2	64.5	70	61	t=9.09%
	75%	60	60.75	61.5	63.75	67.5	61	t=6.67%
	90%	61	61.3	61.6	62.5	64	61	t=0.00%
	100%	57	57	57	57	57	57	t=0.00%
$N = 10$	0%	116	121	126	141	166	144	t=28.00%
	10%	115	119.6	124.2	138	161	135	t=21.74%
	25%	115	118.8	122.6	134	153	126	t=14.47%
	50%	117	119.75	122.5	130.75	144.5	117	t=0.00%
	75%	120	121.65	123.3	128.25	136.5	114	t=-18.18%
	90%	122	122.95	123.9	126.75	131.5	110	t=-63.16%
	100%	109	109	109	109	109	109	t=0.00%

Table 15: Impact of p, t and N on the staffing cost ($\mu_i = \mu_0 = 0.2$, $W_0^* = W_i^* = 0.2$, $\sum_{i=0}^{n} \lambda_i = 8$, $i = 1, ..., n$, $U = V = 1$)

	p	Chaining t=0%	t=5%	t=10%	t=25%	t=50%	SP	Crossing value (Chaining = SP)
$N = 3$	0%	47	49.35	51.7	58.75	70.5	52	t=10.64%
	10%	47	48.45	49.9	54.25	61.5	49	t=6.90%
	25%	47	47.95	48.9	51.75	56.5	48	t=5.26%
	50%	47	47.5	48	49.5	52	47	t=0.00%
	75%	47	47.2	47.4	48	49	47	t=0.00%
	90%	47	47.05	47.1	47.25	47.5	47	t=0.00%
	100%	47	47	47	47	47	47	t=0.00%
$N = 4$	0%	48	49.95	51.9	57.75	67.5	54	t=15.38%
	10%	48	49.45	50.9	55.25	62.5	52	t=13.79%
	25%	47	48.15	49.3	52.75	58.5	50	t=13.04%
	50%	48	48.8	49.6	52	56	48	t=0.00%
	75%	48	48.5	49	50.5	53	48	t=0.00%
	90%	47	47.25	47.5	48.25	49.5	47	t=0.00%
	100%	47	47	47	47	47	47	t=0.00%
$N = 5$	0%	49	51.3	53.6	60.5	72	60	t=23.91%
	10%	49	50.7	52.4	57.5	66	56	t=20.59%
	25%	48	49.3	50.6	54.5	61	52	t=15.38%
	50%	49	49.9	50.8	53.5	58	52	t=16.67%
	75%	51	51.55	52.1	53.75	56.5	51	t=0.00%
	90%	51	51.3	51.6	52.5	54	51	t=0.00%
	100%	47	47	47	47	47	47	t=0.00%
$N = 10$	0%	58	60.65	63.3	71.25	84.5	72	t=26.42%
	10%	55	57.2	59.4	66	77	72	t=38.64%
	25%	55	56.85	58.7	64.25	73.5	63	t=21.62%
	50%	56	57.45	58.9	63.25	70.5	60	t=13.79%
	75%	57	57.95	58.9	61.75	66.5	56	t=-5.26%
	90%	56	56.6	57.2	59	62	55	t=-8.33%
	100%	47	47	47	47	47	47	t=0.00%

Table 16: Impact of c ($\mu_0 = \mu_i = 0.2$ for $i = 1, ..., 4$, $\sum_{i=0}^{4} \lambda_i = 8$, $U = V = 1$, $n = 4$, $W_0 = W_i^* = 0.2$)

Single Pooling									
	$c = 10\%$					$c = 0\%$	$c = 10\%$	$c = 20\%$	$c = 30\%$
p	s_1	s_2	s_3	s_4	s_0	Total staffing cost			
0%	15	15	15	15	0	60	69.615	80.520	92.805
10%	14	14	14	14	0	56	64.974	75.152	86.618
25%	13	13	13	13	0	52	60.333	69.784	80.431
50%	13	13	13	13	0	52	60.333	69.784	80.431
75%	12	13	13	13	0	51	59.002	68.056	78.234
90%	12	13	13	13	0	51	59.002	68.056	78.234

Most Asymmetrical Chaining (MAC)									
	$c = 10\%$					$c = 0\%$	$c = 10\%$	$c = 20\%$	$c = 30\%$
p	s_1	s_2	s_3	s_4	s_0	Total staffing cost			
0%	6	12	11	16	4	49	66.631	89.663	119.337
10%	5	12	9	13	10	49	64.449	84.638	110.645
25%	8	9	6	11	14	48	60.798	77.256	98.148
50%	12	7	3	8	19	49	59.732	73.236	90.015
75%	12	3	2	6	28	51	58.845	68.573	80.498
90%	14	2	1	3	31	51	57.803	66.148	76.253

Less Asymmetrical Chaining (LAC)									
	$c = 10\%$					$c = 0\%$	$c = 10\%$	$c = 20\%$	$c = 30\%$
p	s_1	s_2	s_3	s_4	s_0	Total staffing cost			
0%	10	11	13	15	0	49	90.922	74.928	90.922
10%	10	11	11	12	5	49	92.716	75.984	92.716
25%	12	7	10	9	10	48	90.402	74.208	90.402
50%	15	5	7	6	16	49	93.769	76.656	93.769
75%	20	3	5	2	20	50	95.225	78.000	95.225
90%	23	2	2	2	22	51	98.592	80.448	98.592

Table 17: Impact of c ($\lambda_i = \lambda_0 = 2$, $\sum_{i=0}^{4} \frac{1}{\mu_i} = 25$, $W_0 = W_i^* = 0.2$, $i = 1, ..., 4$, $p = 20\%$, $U = V = 1$, $n = 4$)

Single Pooling									
	$c = 10\%$					$c = 0\%$	$c = 10\%$	$c = 20\%$	$c = 30\%$
p'	s_1	s_2	s_3	s_4	s_0	Total staffing cost			
0%	18	18	18	18	0	72	83.538	96.624	111.366
10%	16	17	17	17	0	67	77.566	89.528	102.982
25%	15	15	16	16	0	62	71.715	82.720	95.105
50%	16	17	17	17	0	67	77.566	89.528	102.982
75%	17	17	17	17	0	68	78.897	91.256	105.179
90%	17	17	17	18	0	69	79.897	92.256	106.179
Most Asymmetrical Chaining (MAC)									
	$c = 10\%$					$c = 0\%$	$c = 10\%$	$c = 20\%$	$c = 30\%$
p'	s_1	s_2	s_3	s_4	s_0	Total staffing cost			
0%	5	17	15	17	6	60	82.272	111.696	149.963
10%	9	13	12	14	11	59	78.278	103.529	136.117
25%	11	11	9	13	14	58	74.971	96.932	124.970
50%	17	7	6	8	22	60	74.219	92.341	115.139
75%	21	3	4	5	28	61	72.835	87.662	106.007
90%	26	2	1	2	34	65	75.565	88.444	103.927
Less Asymmetrical Chaining (LAC)									
	$c = 10\%$					$c = 0\%$	$c = 10\%$	$c = 20\%$	$c = 30\%$
p'	s_1	s_2	s_3	s_4	s_0	Total staffing cost			
0%	6	15	20	14	5	60	74.514	91.392	110.838
10%	10	13	14	12	10	59	74.085	91.680	111.995
25%	13	10	13	10	12	58	72.622	89.616	109.174
50%	20	7	8	6	19	60	75.592	93.696	114.504
75%	25	4	4	4	24	61	77.242	96.096	117.754
90%	32	1	2	2	28	65	82.181	102.048	124.787

Table 18: Crossing value of c ($U = V = 1$, $W_0 = W_i^* = 0.2$, for $i = 1, ..., 4$)

Impact of p ($\mu_0 = \mu_i = 0.2$ for $i = 1, ..., 4$, $\sum_{i=0}^{4} \lambda_i = 8$)			Impact of p' ($\lambda_i = \lambda_0 = 2$ for $i = 1, ..., 4$, $\sum_{i=0}^{4} \frac{1}{\mu_i} = 25$)		
p	$SP = MAC$	$SP = LAC$	p'	$SP = MAC$	$SP = LAC$
0%	12.84%	34.39%	0%	10.93%	31.02%
10%	10.64%	18.23%	10%	9.35%	16.37%
25%	9.16%	10.49%	25%	6.06%	8.34%
50%	11.82%	6.96%	50%	15.98%	13.53%
75%	14.01%	2.18%	75%	28.46%	12.47%
90%	47.82%	0.00%	90%	38.41%	6.51%