MANUFACTURING & SERVICE OPERATIONS MANAGEMENT
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 1523-4614 | EISSN 1526-5498 | 00 | 0000 | 0001

INFORMS
DOI 10.1287/xxxx.0000.0000

Dynamic Call Center Routing Policies Using Call Waiting and Agent Idle Times Online Supplement

Wyean Chan
DIRO, Université de Montréal, C.P. 6128, Succ. Centre-Ville, Montréal (Québec), H3C 3J7, CANADA,
chanwyea@iro.umontreal.ca

Ger Koole
Department of Mathematics, VU University Amsterdam, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands,
koole@few.vu.nl

Pierre L'Ecuyer
DIRO, Université de Montréal, C.P. 6128, Succ. Centre-Ville, Montréal (Québec), H3C 3J7, CANADA,
lecuyer@iro.umontreal.ca

1. Additional comments on the routing rules

This section provides some additional comments on the routing policies P, PD and WR.

1.1. Priority Routing (P)

A common situation where priority routing is used in practice is when an agent group is given a primary skill and secondary skills. The priorities will be set such that an agent from this group will first try to serve a call corresponding to his primary skill before serving any call corresponding to one of his secondary skills.

1.2. Priorities with Delays (PD)

If the objective function is $F_{\mathrm{S}}(\pi)$, i.e., involves only the SL, then according to intuition we may want to impose a delay slightly smaller than the AWT for certain call types, to increase their chance of abandonment before the AWT in case they have to wait. This would in turn improve the quality of service of the remaining customers in terms of SL and average waiting times, and improve the objective function. In particular, if we consider the possibility of *balking*, where the patience time has a mass at zero (a customer hangs up immediately upon entering a waiting queue), and if the abandonments are not penalized, then it is often optimal to impose very short delays for all calls. We do not consider balking in our numerical examples.

1.3. Weight-based Routing (WR)

To better understand the relationship between the WR policy and the more traditional priority lists, and convince ourselves that the WR policy is more general and flexible than the G, P and PD policies, we now examine how these policies can be approximated arbitrary closely by the WR policies. The global FCFS routing can be approximated by setting $q_{k,g} = 0$, $a_{k,g} = 1$, and $b_{k,g} = \epsilon > 0$, for all pairs (k, g), where ϵ is arbitrarily small.

If we have a policy P that uses only the type-to-group priorities, or only the group-to-type priorities, or if the priority lists are symmetric (the primary call type of a group and the primary group of a call type are the same pair, which is often the case in practice), then it suffices either to set the $q_{k,g} \geq 0$ according to the priorities and $a_{k,g} = 1$, or to set $a_{k,g} > 0$ accordingly to the priorities and $q_{k,g} = 0$, with $b_{k,g} = \epsilon$ in both cases.

If the priority lists are not symmetric, then we can use $q_{k,g}$ to reproduce the type-to-group priorities and $a_{k,g}$ for the group-to-type priorities, and $b_{k,g} = \epsilon$. The parameters $q_{k,g}$ must be set by small increments while $a_{k,g}$ must be set by large increments. When a call arrives, its waiting time is 0, so $a_{k,g}$ has no influence in the selection of the group. But when an idle agent chooses a call in the queue, $q_{k,g}$ must be negligible compared to $a_{k,g}$. The same setting applies to approximate the policy PD with time delays. One can use the $q_{k,g}$ for the type-to-group priorities and for the delays, since both cannot be applicable simultaneously. If a call is delayed by a group, then the type-to-group priority for that group is meaningless and vice versa. Similarly, one can set the $a_{k,g}$ according to the group-to-type priorities and the delays.

Typically, the group-to-type priorities dominate when the call center is overloaded (most calls have to wait) and the type-to-group priorities dominate when the call center is underloaded (most calls do not wait). In these cases, one may still obtain a good approximation by converting only the group-to-type priorities in the high-traffic situation and the type-to-group priorities in the low-traffic case. We can simplify the weight policy to approximate the priority rules with $q_{k,g}$ only, using WR-sep.

2. Details on the MGA for routing optimization

This section describes how we apply the modified genetic algorithm (MGA) to optimize the parameters of the routing policies considered in this paper.

For the WR and LG$c\mu$ policies, for all initial parameters $\boldsymbol{\theta}^{(0)}$, we start with mean 0 and a large variance. Depending on the time unit used, choosing the right scale for $q_{k,g}, a_{k,g}$ and $b_{k,g}$ will improve the starting solutions. For example, if the time unit is in second, then we set the variance of $q_{k,g}$ to be 100 times larger than those of $a_{k,g}$ and $b_{k,g}$. In the case of hours, we set the variance of $q_{k,g}$ to be 100 times smaller.

For WR-neg, we perform a pre-selection of the solutions such that any solution for which $a_{k,g} < 0$ and $b_{k,g} < 0$ for any pair (k, g) is automatically rejected (because such rule is not realistic), and a new one is generated. That is, at each iteration of the MGA, random solutions are generated from $\Phi(\boldsymbol{\theta})$ until there are P admissible solutions, and these P solutions are then evaluated by simulation. This pre-selection procedure is particularly useful at the beginning of the algorithm, where many bad candidate solutions are rejected without requiring any simulation.

We also use the normal distribution to generate the parameters of LG$c\mu$, the delays for the policies PD and PDT, and the idle agent thresholds for the policies PT and PDT. The negative delays and agent thresholds are replaced by 0. For these policies, we set the initial parameters with small positive mean and variance. In the numerical experiments, we use initial means of 10 seconds and one idle agent, and the same for the initial standard deviations.

Optimizing the priority lists with MGA is more complicated, because it is a combinatorial problem. Here, we assume that the skill sets $\mathcal{S}_g$ and $h_g = |\mathcal{S}_g|$ are fixed, and we only decide on the priorities. To generate a solution, we generate the type-to-group and the group-to-type priority lists independently. We describe how to create the preference list $\mathcal{L}_g$ of group g, with skill set $\mathcal{S}_g$. The type-to-group priority list $\mathcal{G}_k$ of call type k is generated in a similar way. Note that even in very large call centers, there may be many priority lists, but these lists are usually small.

If $h_g = 1$, then the preference list is trivial and there is nothing to do, so we now assume that $h_g > 1$. The distribution used to generate the list $\mathcal{L}_g$ has parameters $\alpha_{g,k} > 0$ for $k \in \mathcal{S}_g$ and $\beta_{g,n} \in [0.05, 0.95]$ for $n = 1, \ldots, h_g - 1$. We initialize $\alpha_{g,k} = 1$ for all k and $\beta_{g,n} = 0.5$ for all n. The priority list is generated in two independent steps:

1. We generate a permutation $\mathcal{P} = (p_1, p_2, \ldots, p_{h_g})$ such that $p_i \in \mathcal{S}_g$, for all i and $p_i \neq p_j$ for $i \neq j$. To do this, we generate a random variate $z_{g,k}$ uniformly distributed over the interval $[0, \alpha_{g,k}]$, for each $k \in \mathcal{S}_g$, and the permutation is obtained by sorting these $z_{g,k}$'s in increasing order. Note that the $h_g!$ permutations do not have all equal probabilities. Call types k with smaller $\alpha_{g,k}$ have more chance to appear earlier in the list and to have higher priorities.

2. Given the permutation, we now have to decide on the subsets $\mathcal{L}_g^{(\cdot)}$ of call types having equal priorities in this list. That is, we must decide between "equal" and "higher than" for the priority relationship between any two successive positions in the permutation $\mathcal{P}$. There are $h_g - 1$ relationships to determine, and the nth one is chosen to be "equal" with probability $\beta_{g,n}$, for $n = 1, \ldots, h_g - 1$. More precisely, call types p_n and p_{n+1} have the same priority with probability $\beta_{g,n}$, else call type p_n has higher priority. We bound $\beta_{g,n}$ away from 0 and 1 to allow for a minimum of diversification.

The distribution parameters $\alpha_{g,k}$ and $\beta_{g,n}$ are updated at each iteration as follows. For $\alpha_{g,k}$, we simply take the average rank of call type k in the priority lists of group g over the $\hat{P}$ elite solutions. Call types with equal priorities have the same rank. Those with the highest priority have rank 1, those with the next priority have rank 2, and so on. To update $\beta_{g,n}$, we compute the proportion of call types at position n having the same priority as the call types at position $n+1$ in the $\hat{P}$ elite solutions. For the stopping criterion based on the variance of $\Phi(\boldsymbol{\theta})$, we only look at the variance of the ranks used to update $\alpha_{g,k}$, from the elite sample, for each g and k.

For the policy PD, we optimize both the priority rules and delays simultaneously for small size call centers. For larger call centers, we found that a 2-stage approach where the only priority rules are optimized first and the delays after is more efficient (although suboptimal) when the execution time is limited. We also use a 2-stage approach to optimize the policy PT for large call centers.

Combining the priorities, delays and agent thresholds in the policy PDT quickly increases the number parameters and optimizing all these parameters simultaneously becomes very difficult. Instead, we optimize subsets of parameters iteratively with the MGA, using a multi-stage approach (itMGA stages). We always start the first stage with the priority rules, since they tend to have a bigger impact on the PMs. For the itMGA-1 remaining stages, we optimize subsets chosen randomly from: (1) priorities, (2) priorities and delays, (3) priorities and thresholds, and (4) delays. A tabu list is used to store the subsets that give no improvement and these subsets cannot be chosen while they are in the list. This algorithm for PDT is very time-consuming, more than for WR and LG$c\mu$, because of the multiple executions of the MGA. We allow larger execution times for this algorithm.

3. Additional comments on the numerical examples

In this section, we give further comments on the numerical results of the X and N models. We also give additional details on the model parameters of the real-life based example with 6 call types and 8 agent groups. Finally, we include a new example with 8 call types and 10 agent groups.

3.1. Comments on the solutions for the X-model

We now discuss further how the routing solutions for the X-model look like. Among the 5 retained solutions for each policy, we always report and discuss the best one. We first compare the solutions for the objective function F_{S}. For policy P, group i gives high priority to call type i, for $i = 1, 2$, and for both call and agent selections. Note that this is not equivalent to a fastest-server-first rule, because here $\mu_{1,2} = \mu_{2,2}$. For PD, the best solution sets the same group-to-type priorities as for P and sets the delays to $d_{1,2} = 16.1$ and $d_{2,1} = 2.7$ seconds. Note that if we had $d_{k,g} > \tau_k = 20$, then all calls of type k served by group g would have a "bad service". PD has more abandonments than P overall, because call type 1 has 10 times more volume than type 2 and the penalties are not weigthed by call volumes; so it is easier to improve the SL of type 2 (with the same staffing vector). For PT,

the type-to-group priority lists are identical to P, and the group-to-type lists are $\mathcal{L}_1 = (\{2\}, \{1\})$ and $\mathcal{L}_2 = (\{1, 2\})$, with thresholds $m_{1,2} = 0.95$, $m_{2,1} = 1.62$, and $m_{2,2} = 0.09$. The small positive value of $m_{2,2}$ is only an artefact of the "incomplete" convergence of MGA. Replacing $m_{2,2}$ with 0 reduces the penalty by less than 0.2. The PDT solution is mixed: the priorities are the same as for P, the delays are $d_{1,2} = 14.7$ and $d_{2,1} = 7.8$ seconds, and the thresholds are $(m_{1,1}, m_{1,2}) = (0.63, 0.80)$ and $(m_{2,1}, m_{2,2}) = (1.64, 0.004)$. The threshold $m_{2,2}$ is practically 0, but $m_{1,1}$ is not (putting it from 0.63 to 0 increases the cost from 4.9 to 8.45; with 0.63, there are 0.05% more abandonments and the average waiting times are 0.2 seconds larger, but S_1 is 0.8% larger and S_2 is 0.2% smaller). The policy LG$c\mu$ often has lower SL and higher abandonments than WR and WR-idnum, and this makes it more expensive. For WR, the PMs for F_{S} are just slightly below those of PD. The large $q_{k,g}$ gives more weight on the faster group, as expected in this inefficient-sharing context. The coefficient $a_{1,2}$ is relatively large (and larger than $a_{2,2}$), but this is to compensate for the negative $q_{1,2}$ and to create dynamic delays. Observe in Table 3 of the main paper that for all solutions of WR, the parameter $q_{1,2}$ is negative, and the parameters $q_{1,1}$ and $q_{2,2}$ are positive.

There are similarities between the parameters $q_{k,g}$ of WR-sep and WR. The solution for WR-sep allows the possibility of delay for the pair $(k = 1, g = 2)$, just like WR. The best solution of WR-sep2 is: $(q_1, q_2) = (575, 797)$, $(a_1, a_2) = (15.1, 0.028)$, $(r_1, r_2) = (374, -783)$, and $(b_1, b_2) = (12.4, 0)$. Group 2 will prioritize call type 1 once the waiting time is greater than 14.7 seconds. The PMs are similar to those of LG$c\mu$.

The best solution with WR-neg for $F_{\mathrm{SA}}^{\lambda}$ has negative coefficients and significantly smaller cost than WR. Like for WR, we have $q_{1,2} < 0$ and $a_{1,2} > 0$, which creates "delays" that decrease as time increases. But there are also opposite cases with $q_{2,1}, q_{2,2} > 0$ and $a_{2,1}, a_{2,2} < 0$, where delays increase (or are initiated) as wait times increase! The small call type 2 is even less important with WR-neg and $F_{\mathrm{SA}}^{\lambda}$: the agents are *discouraged* to serve call type 2 if it has waited too long, unless the queue of type 1 is empty and an agent has been idle for a long time.

Since agents can serve all call types in the X-model, it is not hard to have fair agent occupancy; even policy G has occupancy levels $\mathbf{O} = (95.9, 96.1)$. The difficult part is to reduce the SL penalties in F_{SO}.

3.2. Additional details on the example with 6 call types and 8 groups

The service time distribution parameters are estimated from approximately one year of data. For *each* of the nearly 1500 agents, the data contains her daily average service time for each call type, and the daily number of calls of each type she served. Considering only the agents that belong to one of the 8 selected groups, we estimate the service time distribution of a group by aggregating the agents of the same group, weighted by the number of served calls. After this aggregation, we have 240 to 252 days of service time observations for 17 skill pairs; this corresponds approximately to a period of one year excluding weekends. The other skills have smaller numbers of daily observations, with the smallest being 62 days. Many reasons can explain this difference of observed days: the routing has changed over time, agents have moved to different groups, some groups do not work everyday, etc.

The service times of the 25 skill pairs are modeled as lognormal distributions, and Table 1 shows, for each pair, the mean μ and standard deviation σ of the underlying normal distribution, followed by the mean and standard deviation of the lognormal distribution, in minute units.

3.3. An example with 8 call types and 10 agent groups

We consider a larger example with 8 call types and 10 agent groups. The arrival process is Poisson, and the service and patience times are exponential with rates $(\lambda_1, \ldots, \lambda_8) = (250, 200, 100, 80, 50, 20, 15, 10)$, $(\mu_1, \ldots, \mu_8) = (10, 6, 6, 10, 6, 6, 8, 10)$, and $(\nu_1, \ldots, \nu_8) = (10, 8, 10, 12, 6, 10, 12, 10)$, all per hour. The staffing vector is $(y_1, \ldots, y_{10}) = (21, 12, 14, 8, 16, 5, 3, 7, 8, 9)$ for a

Table 1 The lognormal distribution parameters of the service times for each of the 25 skills in the example with 6 call types and 8 groups.

Call type	Group	μ	σ	mean	Std. dev.	Call type	Group	μ	σ	mean	Std. dev.
1	1	2.126	0.278	8.7	6.1	3	8	1.862	0.451	7.1	11.5
1	2	2.352	0.275	10.9	9.3	4	1	2.140	0.322	9.0	8.8
1	5	2.343	0.326	11.0	13.5	5	1	1.924	0.334	7.2	6.2
1	6	2.260	0.322	10.1	11.1	5	3	1.826	0.498	7.0	13.9
1	8	2.116	0.374	8.9	11.8	5	4	1.949	0.343	7.5	6.9
2	2	2.293	0.273	10.3	8.2	5	5	1.958	0.376	7.6	8.8
2	6	2.386	0.268	11.3	9.5	5	6	2.117	0.338	8.8	9.4
3	1	2.036	0.312	8.0	6.6	5	7	1.898	0.397	7.2	8.9
3	3	1.749	0.614	6.9	22.0	5	8	1.887	0.399	7.2	8.8
3	4	1.921	0.390	7.4	8.9	6	4	1.514	0.497	5.1	7.4
3	5	1.965	0.458	7.9	14.7	6	7	1.640	0.413	5.6	5.9
3	6	2.172	0.357	9.4	11.9	6	8	1.768	0.445	6.5	9.2
3	7	1.899	0.499	7.6	16.2						

total of 103 agents. The skill sets are $\mathcal{S}_1 = \{1,4\}$, $\mathcal{S}_2 = \{2,5\}$, $\mathcal{S}_3 = \{3,4,7\}$, $\mathcal{S}_4 = \{4,6,8\}$, $\mathcal{S}_5 = \{2,5\}$, $\mathcal{S}_6 = \{6,7,8\}$, $\mathcal{S}_7 = \{1,3,7\}$, $\mathcal{S}_8 = \{2,4,8\}$, $\mathcal{S}_9 = \{1,3,4,8\}$ and $\mathcal{S}_{10} = \{2,7,8\}$. The average agent occupancy is around 90%, but the average occupancy per group can vary between 75% and 95%. Compared to the example with 6 types and 8 groups, this model has 3 more skills, but the stochastic processes are simpler. The SL targets are $t_k = 80\%$ with $\tau_k = 20$ seconds for all call types. The simulation parameters are $T = 100$ hours and $n = 10$ replications.

We execute the MGA with $P = 400$, $\hat{P} = 20$ and $\mathsf{maxIt} = 40$. For PDT, we take $\mathsf{maxIt} = 20$, $P = 200, \hat{P} = 10$ and we set the multi-step parameter $\mathsf{itGMA} = 10$. We optimize the parameters of PD and PT with a two-stage MGA. The first stage optimizes the priorities with $\mathsf{maxIt} = 30$ and the second stage optimizes the delays or thresholds with $\mathsf{maxIt} = 10$. An optimization run takes roughly 14 hours. Contrary to the real-life based example, the solution costs do not improve much further with additional CPU time.

Table 2 Average and lowest cost estimated out-of-sample for the retained policy, for the 3 runs, for the large example with 8 call types and 10 groups.

Objective	G	P	PD*	PT*	PDT	LG$c\mu$	WR	WR-sep	WR-idnum	WR-neg
F_{S}	616	231	192	147	165	209	107	108	**96**	**61**
F_{S}^{*}	616	211	176	137	156	206	104	102	**95**	**57**
$F_{\mathrm{SA}}^{\lambda}$	26.3	14.4	13.3	11.3	11.9	13.6	9.4	9.1	**8.6**	**7.7**
$F_{\mathrm{SA}}^{\lambda *}$	26.3	14.3	13.0	10.8	11.7	13.5	9.4	9.1	**8.6**	**7.2**

Table 2 reports the average costs (rows F_S and $F_{\mathrm{SA}}^{\lambda}$) and lowest costs (rows F_S^* and $F_{\mathrm{SA}}^{\lambda *}$) of 3 optimization runs, evaluated out-of-sample. The lowest costs of each objective function are shown in bold font. PD* and PT* in the table denote PD and PT policies whose parameters have been optimized via two-stage MGA (the solutions from the single-stage MGA are worse than P in this case). The WR-idnum policy clearly wins, followed by WR and WR-sep. If we allow negative coefficients, then WR-neg gives even lower costs.

3.4. N-model with penalties on expected waiting times only

We now examine an example of an N-model under a heavy-traffic regime with penalty costs based on expected waiting times only, where either a $c\mu$-type rule or a priority rule (such as policy P), depending on the asymptotic regime and the waiting (holding) costs, is known to be asymptotically optimal, as shown in Tezcan and Dai (2010). Our aim is to examine how well WR and LG$c\mu$, combined with MGA, perform in those types of situations.

There are two call types and two agent groups. Group 1 can serve only call type 1, while group 2 can serve both types 1 and 2. The arrivals are Poisson with rates $\lambda_1 = 5.7$ and $\lambda_2 = 0.6$ call per second for types 1 and 2 respectively. There are no abandonments. The service times are exponential, with means that depend only on the agent group (not on the call type); the mean is 10 minutes for agents of group 1 and 9 minutes for agents of group 2. There are $y_1 = 2850$ and $y_2 = 580$ agents, and the average agent occupancy is 99% (a heavy-traffic situation). This system evolves over successive independent "days" of length T (the time horizon). The penalty function involves only the expected waiting times:

$$F_{\mathrm{W}}(a) = a\mathbb{E}[W_1] + \mathbb{E}[W_2],$$

where W_k is the waiting time of a call of type k selected randomly over all days and a is a penalty weight on call type 1. To relate with the objective function (8) considered by the G$c\mu$ rule in the main paper, we can rewrite $F_{\mathrm{W}}(a)$ as the following expected cumulative weighted sum of waiting times over one day:

$$F_{\mathrm{W}}(a) = \mathbb{E}\left[a\sum_{n=1}^{N_1}\frac{W_{1,n}}{\lambda_1 T} + \sum_{n=1}^{N_2}\frac{W_{2,n}}{\lambda_2 T}\right],$$

where N_k is the number of arrivals of type k during the day and $W_{k,n}$ is the waiting time of the n-th call of type k.

We report experiments with two values of a, namely $a = 3$ and $a = 20$. The optimal policy is (at least to a very close approximation) a fixed priority rule (policy P) when $a = 3$, and a $c\mu$-type rule when $a = 20$. Note that these two rules can be mimicked by both LG$c\mu$ and WR for this N-model. We also tested non-heavy-traffic settings, with arrival rates 100 times smaller and with smaller staffing, $y_1 = 30$ and $y_2 = 9$, for which the average occupancy is approximately 94%. For each routing policy and each of these four settings, we optimized the routing parameters for $F_{\mathrm{W}}(a)$ with MGA, and we did this 3 times independently. The costs of the best solutions (evaluated out-of-sample) are presented in Table 3. The results agree with the theory: policy P is nearly-optimal for $a = 3$ and LG$c\mu$ gives a nearly-optimal policy when $a = 20$. Overall, LG$c\mu$, WR and WR-idnum perform well in all cases.

Table 3 Comparing the best costs out of 3 runs for the N-model example, with objective function $F_{\mathrm{W}}(a)$.

Heavy-traffic	a	G	P	LG$c\mu$	WR	WR-idnum	WR-idt
Yes	3	143	117	117	118	116	123
	20	746	493	407	419	410	466
No	3	691	582	582	581	581	584
	20	3360	3212	2684	2670	2666	2678

References

Tezcan, T., J. G. Dai. 2010. Dynamic control of N-systems with many servers: Asymptotic optimality of a static priority policy in heavy traffic. *Operations Research* **58**(1) 94–110.