

# Call Center Routing Policy Using Call Waiting and Agent Idle Times Online Supplement

(Authors' names blinded for peer review)

In this online supplement, we give additional descriptions of the implementation of the optimization algorithms presented in the paper. Then, we give the solutions that we obtained for each routing policy for the V, X and M models with objective function  $F_3^1$ .

## Appendix A: Routing Optimization

### A.1. Stochastic Gradient Descent (SGD)

We present the SGD framework in Algorithm 1. The inputs are the initial solution  $\mathbf{x}^{(0)}$ , the initial step size  $\delta_0$ , the maximum number of iterations `maxIt`, and the number of restarts `maxRestart`. Each variable is allowed to be bounded to a given interval. A variable is truncated to the nearest bound when the next iteration attempts to send it outside of its interval. The parameters  $\epsilon_1, \epsilon_2, \epsilon_3$  are used to define a stopping criterion in the inside loop of the algorithm.

We now describe our implementation of the *quasi-Newton* (QN) method. We use the Broyden-Fletcher-Goldfarb-Shanno (BFGS) formula to approximate the inverse of the Hessian matrix. Let  $\mathbf{x}^{(k)}$  be the solution at iteration  $k$ ,  $\mathbf{g}^{(k)}$  the gradient at  $\mathbf{x}^{(k)}$ ,  $\mathbf{Z}^{(k)}$  the inverse of the Hessian matrix,  $\alpha^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}$  and  $\gamma^{(k)} = \mathbf{g}^{(k)} - \mathbf{g}^{(k-1)}$ . The BFGS formula is

$$\mathbf{Z}^{(k)} = \mathbf{Z}^{(k-1)} + \left( 1 + \frac{(\gamma^{(k)})^T \mathbf{Z}^{(k-1)} \gamma^{(k)}}{(\alpha^{(k)})^T \gamma^{(k)}} \right) \frac{\alpha^{(k)} (\alpha^{(k)})^T}{(\alpha^{(k)})^T \gamma^{(k)}} - \frac{\alpha^{(k)} (\gamma^{(k)})^T \mathbf{Z}^{(k-1)} + \mathbf{Z}^{(k-1)} \gamma^{(k)} (\alpha^{(k)})^T}{(\alpha^{(k)})^T \gamma^{(k)}}.$$

The search direction is  $\mathbf{s}^{(k)} = -\mathbf{Z}^{(k-1)} \mathbf{g}^{(k-1)}$ . In our implementation,  $\mathbf{g}^{(k)}$  is actually a noisy estimate of the gradient rather than its exact value, as in the previous algorithm. The golden section search is again used for the line search.

Algorithm 2 presents the modifications to Algorithm 1 to implement the QN method. For the initialization, we assume that  $\mathbf{x}^{(-1)}$  and  $\mathbf{g}^{(-1)}$  are zero vectors, and  $\mathbf{Z}^{(-1)} = \mathbf{I}$ , the identity matrix.

### A.2. A Modified Genetic Algorithm (MGA)

In Algorithm 3, we present the framework of the MGA described in the paper. The parameter `maxIt` determines the maximum number of iterations,  $\gamma$  is used for the smoothing of the parameters of the distributions and the other parameters are described in the main text.

```

Input:  $\mathbf{x}^{(0)}$ ,  $\delta_0$ , maxIt, maxRestart
Output: the best solution found,  $\mathbf{x}^*$ 
for  $m = 1$  to maxRestart do
  for  $k = 1$  to maxIt do
     $\delta \leftarrow \delta_0/k$ 
     $\mathbf{g}^{(k-1)} \leftarrow$  a finite-difference gradient estimate at  $\mathbf{x}^{(k-1)}$ , with step size  $\delta$ 
     $\bar{\lambda} \leftarrow \arg \min_{\lambda \geq 0} \hat{f}(\mathbf{x}^{(k-1)} - \lambda \mathbf{g}^{(k-1)})$ , computed by a golden section search
     $\mathbf{x}^{(k)} \leftarrow \mathbf{x}^{(k-1)} - \bar{\lambda} \mathbf{g}^{(k-1)}$ 
    if  $\hat{f}(\mathbf{x}^{(k)}) < \hat{f}(\mathbf{x}^*)$  then
       $\mathbf{x}^* \leftarrow \mathbf{x}^{(k)}$ 
    end
    if  $\left( \left| \frac{x_i^{(k)} - x_i^{(k-1)}}{x_i^{(k-1)}} \right| < \epsilon_1 \text{ for all } i \text{ and } |\mathbf{g}| < \epsilon_2 \text{ and } d < \epsilon_3 \right)$  then
       $k \leftarrow \text{maxIt} + 1$  // exit the loop
    end
  end
   $\mathbf{x}^{(0)} \leftarrow \mathbf{x}^*$  // prepare for restart
end

```

**Algorithm 1:** The SGD algorithm used in this paper.

```

for  $k = 1$  to maxIt do
  ...
   $\mathbf{g}^{(k-1)} \leftarrow$  finite-difference gradient estimate at  $\mathbf{x}^{(k-1)}$ , with step  $\delta$ 
   $\mathbf{Z}^{(k-1)} \leftarrow$  update of the inverse Hessian, based on the BFGS formula
   $\mathbf{s}^{(k)} \leftarrow -\mathbf{Z}^{(k-1)} \mathbf{g}^{(k-1)}$  // the new search direction
   $\bar{\lambda} \leftarrow \arg \min_{\lambda \geq 0} \hat{f}(\mathbf{x}^{(k-1)} + \lambda \mathbf{s}^{(k)})$ , computed by a golden section search
   $\mathbf{x}^{(k)} \leftarrow \mathbf{x}^{(k-1)} + \bar{\lambda} \mathbf{s}^{(k)}$ 
  ...
end

```

**Algorithm 2:** Quasi-Newton method, showing the changes to Algorithm 1.

The choice of population sizes  $P$  and  $\hat{P}$  can have a significant impact on the convergence speed and the quality of the final solution. When  $P$  is large and  $\hat{P}$  is small, the MGA generally converges faster to a local minimum, whereas if  $\hat{P}$  is large, convergence is slower but often yields better solutions. For our experiments, we made preliminary tests to select good values.

Various heuristics can be used to further improve the performance of the MGA. For example, the simulation time can be reduced by rejecting early (without doing all the planned simulation runs) the solutions whose

**Input:**  $\text{maxIt}$ ,  $P$ ,  $\hat{P}$ ,  $\gamma$ ,  $\epsilon$ ,  $\boldsymbol{\theta}^{(0)}$

**Output:** the best solution found,  $\mathbf{x}^*$ , and its estimated cost  $f^*$

$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^{(0)}$

**for**  $i = 1$  **to**  $\text{maxIt}$  **do**

**for**  $p = 1$  **to**  $P$  **do**

generate a solution  $\mathbf{x}^{(p)}$  from the distribution  $\Phi(\boldsymbol{\theta})$

$f^{(p)} \leftarrow \hat{f}(\mathbf{x}^{(p)})$ , estimated by simulation

**end**

sort and relabel  $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(P)}$  by order of estimated cost  $f^{(1)}, \dots, f^{(P)}$

**if**  $f^{(1)} < f^*$  **then**

$\mathbf{x}^* \leftarrow \mathbf{x}^{(1)}$  **and**  $f^* \leftarrow f^{(1)}$

**end**

compute estimator  $\tilde{\boldsymbol{\theta}}$  (e.g., by max. likelihood) from the set of elite solutions  $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(\hat{P})}$

$\boldsymbol{\theta} \leftarrow \gamma \tilde{\boldsymbol{\theta}} + (1 - \gamma) \boldsymbol{\theta}$

**if**  $\Phi(\boldsymbol{\theta})$  has small enough generalized variance, say  $< \epsilon$ , **then**

**stop**

**end**

**end**

**Algorithm 3:** A modified genetic algorithm.

estimated cost already appears too high after a few runs. In our implementation, we used the following simple procedure. We started by simulating the  $P$  solutions with a small number of runs, say  $r_0$ , then we simulated the  $M = \min\{2\hat{P}, P\}$  solutions with the smallest empirical cost for another  $r_1$  runs, and we finally selected the  $\hat{P}$  (empirically) best solutions among those  $M$  to form the elite population. More sophisticated *ranking and selection* procedures (Goldsman et al. 2002, Benson et al. 2006) could have been used at this stage, but here the probability of correct selection does not need to be high, so these procedures are not necessarily useful. Another idea that we used is to give more weight to the best solutions in the elite population when computing the maximum likelihood estimator. This often speeds up the convergence to a local minimum, and can be useful especially when the computing budget is limited.

In our implementation, for the weight parameters in the WR policy, we used the normal distribution. These parameters can then take any real value, and the maximum likelihood estimator is easy to compute. For each of those weight parameters, there are two coordinates in  $\boldsymbol{\theta}$ : the mean and the variance of the normal distribution. Any solution for which  $a_{k,g} < 0$  and  $b_{k,g} < 0$  for any pair  $(k, g)$  is rejected immediately, and a new one is generated. At each iteration of the MGA, random solutions are generated from  $\Phi(\boldsymbol{\theta})$  until there are  $P$  admissible solutions. This procedure is particularly useful at the beginning of the algorithm, where many bad candidate solutions are rejected.

We also use the normal distribution to generate the delays for the policies PD and PDT, and the idle

agent thresholds for the policies PT and PDT. The negative values are replaced by 0, and the thresholds are rounded to the nearest integer.

Optimizing the priority lists with the MGA is more complicated, because it is a combinatorial problem. Here, we assume that the skill sets  $\mathcal{S}_g$  and their cardinalities 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 generate 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 a call type  $k$  is generated in a similar way. Note that even in very large call centers, although there may be many priority lists, 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  $m_{g,k} > 0$  for  $k \in \mathcal{S}_g$  and  $q_{g,n} \in [0.05, 0.95]$  for  $n = 1, \dots, h_g - 1$ . We initialize  $m_{g,k} = 1$  for all  $k$  and  $q_{g,n} = 0.5$  for all  $n$ . The priority list is generated in two independent steps:

1. We generate a permutation  $\mathcal{P} = (p_1, p_2, \dots, p_{n_g})$  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  $[0, m_{g,k}]$ , 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 a smaller  $m_{g,k}$  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  $\mathcal{L}_g^{(n)}$  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  $n$ th one is chosen to be ‘‘equal’’ with probability  $q_{g,n}$ , for  $n = 1, \dots, h_g - 1$ . We bound  $q_{g,n}$  away from 0 and 1 to allow for a minimum of diversification. Call types  $p_n$  and  $p_{n+1}$  have the same priority if  $q_{g,n} = 1$ , otherwise call type  $p_n$  has higher priority.

The distribution parameters  $m_{g,k}$  and  $q_{g,n}$  are updated at each iteration as follows. For  $m_{g,k}$ , 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  $q_{g,n}$ , 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  $\Phi(\theta)$ , we only look at the variance of the ranks used to update  $m_{g,k}$ , from the elite sample, for each  $g$  and  $k$ .

## Appendix B: Solutions to the numerical examples

We report the actual solutions for the V, X and M models with the objective function  $F_S^1$  from the numerical section, to give an idea of how the policy parameters look like. The complete solutions for all the examples are available on the third author's web site, in ‘‘publications’’. The delays for the policies PD and PDT are given in seconds. For the WR policy, the call waiting times  $w_k$  and agent idle times  $v_g$  are measured in hours. To save space, we do not report the trivial cases where the type-to-group and group-to-type priority

**Table 1** Solutions for the V-model examples under the objective function  $F_S^1$ .

Case	Routing solutions
1	<p><b>P:</b> <math>\mathcal{L}_1 = (\{2\}, \{1\})</math>.</p> <p><b>PD:</b> <math>\mathcal{L}_1 = (\{2\}, \{1\}), d_{2,1} = 5.9</math>.</p> <p><b>PT:</b> <math>\mathcal{L}_1 = (\{1\}, \{2\}),</math> No threshold.</p> <p><b>PDT:</b> <math>\mathcal{L}_1 = (\{1\}, \{2\}), d_{1,1} = 5.8,</math> No threshold.</p> <p><b>WR:</b> <math>q_{1,1} = 10, q_{2,1} = 10, a_{1,1} = -114, a_{2,1} = -128, b_{1,1} = 220, b_{2,1} = 62</math>.</p> <p><b>WR*:</b> <math>q_{1,1} = q_{2,1} = 1, a_{1,1} = a_{2,1} = -180, b_{1,1} = b_{2,1} = 627</math>.</p>
2	<p><b>P:</b> <math>\mathcal{L}_1 = (\{2\}, \{1\})</math>.</p> <p><b>PD:</b> <math>\mathcal{L}_1 = (\{2\}, \{1\}), d_{1,1} = 19.9</math>.</p> <p><b>PT:</b> <math>\mathcal{L}_1 = (\{1\}, \{2\}), m_{1,1} = 1</math>.</p> <p><b>PDT:</b> <math>\mathcal{L}_1 = (\{1, 2\}), d_{2,1} = 10.4, m_{1,1} = 1</math>.</p> <p><b>WR:</b> <math>q_{1,1} = -0.09, q_{2,1} = 13, a_{1,1} = -81, a_{2,1} = 125, b_{1,1} = 189, b_{2,1} = 105</math>.</p>
3	<p><b>P:</b> <math>\mathcal{L}_1 = (\{2\}, \{1\})</math>.</p> <p><b>PD:</b> <math>\mathcal{L}_1 = (\{2\}, \{1\}), d_{1,1} = 19.9</math>.</p> <p><b>PT:</b> <math>\mathcal{L}_1 = (\{1\}, \{2\}), m_{1,1} = 1</math>.</p> <p><b>PDT:</b> <math>\mathcal{L}_1 = (\{2\}, \{1\}), d_{1,1} = 19.9,</math> No threshold.</p> <p><b>WR:</b> <math>q_{1,1} = -0.013, q_{2,1} = 1.0, a_{1,1} = -493, a_{2,1} = 10, b_{1,1} = 1710, b_{2,1} = 10</math>.</p>

lists have a single group or a single call type. We also do not report the delay when  $d_{k,g} = 0$ , nor the idle agent threshold when  $m_{k,g} = 0$ . We mention “No delay” if all  $d_{k,g} = 0$  and “No threshold” if all  $m_{k,g} = 0$ . The parameters  $q_{k,g}$ ,  $a_{k,g}$  and  $b_{k,g}$  for the WR policy are not reported when their values are 0. Refer to the descriptions of the routing policies for the definitions of the parameters.

## References

- Benson, K. C., D. Goldsman, Amy R. Pritchett. 2006. Ranking and selection procedures for simulation. *Proceedings of the 2006 Winter Simulation Conference*. IEEE Press, 179–185.
- Goldsman, D., S.-H. Kim, W. S. Marshall, Barry L. Nelson. 2002. Ranking and selection for steady-state simulation: Procedures and perspectives. *INFORMS Journal on Computing* **14**(1) 2–19.

**Table 2** Solutions for the X-model examples under the objective function  $F_S^1$ .

Case	Routing solutions
1	<p><b>P:</b> <math>\mathcal{G}_1 = \mathcal{G}_2 = (\{1, 2\})</math>, <math>\mathcal{L}_1 = (\{2\}, \{1\})</math>, <math>\mathcal{L}_2 = (\{1\}, \{2\})</math>.</p> <p><b>PD:</b> <math>\mathcal{G}_1 = (\{1, 2\})</math>, <math>\mathcal{G}_2 = (\{2\}, \{1\})</math>, <math>\mathcal{L}_1 = \mathcal{L}_2 = (\{1\}, \{2\})</math>, <math>d_{1,1} = 13</math>, <math>d_{1,2} = 2.1</math>, <math>d_{2,1} = 0.2</math>.</p> <p><b>PT:</b> <math>\mathcal{G}_1 = \mathcal{G}_2 = (\{1\}, \{2\})</math>, <math>\mathcal{L}_1 = (\{2\}, \{1\})</math>, <math>\mathcal{L}_2 = (\{1\}, \{2\})</math>, No threshold.</p> <p><b>PDT:</b> <math>\mathcal{G}_1 = \mathcal{G}_2 = (\{1, 2\})</math>, <math>\mathcal{L}_1 = \mathcal{L}_2 = (\{1\}, \{2\})</math>, <math>d_{1,1} = 7.8</math>, <math>d_{1,2} = 9.3</math>, No threshold.</p> <p><b>WR:</b> <math>q_{1,1} = 5.0</math>, <math>q_{1,2} = 16</math>, <math>q_{2,1} = 4.1</math>, <math>q_{2,2} = 17</math>, <math>a_{1,1} = -209</math>, <math>a_{1,2} = -68</math>, <math>a_{2,1} = -122</math>, <math>a_{2,2} = -120</math>, <math>b_{1,1} = 125</math>, <math>b_{1,2} = 334</math>, <math>b_{2,1} = 137</math>, <math>b_{2,2} = 113</math>.</p> <p><b>WR*:</b> <math>q_{1,1} = q_{1,2} = q_{2,1} = q_{2,2} = 1</math>, <math>a_{1,1} = a_{1,2} = a_{2,1} = a_{2,2} = -180</math>, <math>b_{1,1} = b_{1,2} = b_{2,1} = b_{2,2} = -700</math>.</p>
2	<p><b>P:</b> <math>\mathcal{G}_1 = \mathcal{G}_2 = (\{1\}, \{2\})</math>, <math>\mathcal{L}_1 = \mathcal{L}_2 = (\{2\}, \{1\})</math>.</p> <p><b>PD:</b> <math>\mathcal{G}_1 = \mathcal{G}_2 = (\{1\}, \{2\})</math>, <math>\mathcal{L}_1 = \mathcal{L}_2 = (\{2\}, \{1\})</math>, <math>d_{1,1} = 16</math>, <math>d_{1,2} = 19.9</math>.</p> <p><b>PT:</b> <math>\mathcal{G}_1 = \mathcal{G}_2 = (\{1\}, \{2\})</math>, <math>\mathcal{L}_1 = (\{1\}, \{2\})</math>, <math>\mathcal{L}_2 = (\{2\}, \{1\})</math>, <math>m_{1,1} = 1</math>.</p> <p><b>PDT:</b> <math>\mathcal{G}_1 = \mathcal{G}_2 = (\{1\}, \{2\})</math>, <math>\mathcal{L}_1 = (\{1, 2\})</math>, <math>\mathcal{L}_2 = (\{2\}, \{1\})</math>, No delay, <math>m_{1,1} = 1</math>.</p> <p><b>WR:</b> <math>q_{1,1} = 0.34</math>, <math>q_{1,2} = 1.0</math>, <math>q_{2,1} = 16</math>, <math>q_{2,2} = 14</math>, <math>a_{1,1} = -59</math>, <math>a_{1,2} = -149</math>, <math>a_{2,1} = 115</math>, <math>a_{2,2} = 74</math>, <math>b_{1,1} = 266</math>, <math>b_{1,2} = 201</math>, <math>b_{2,1} = 11</math>, <math>b_{2,2} = 230</math>.</p>
3	<p><b>P:</b> <math>\mathcal{G}_1 = (\{1, 2\})</math>, <math>\mathcal{G}_2 = (\{2\}, \{1\})</math>, <math>\mathcal{L}_1 = (\{2\}, \{1\})</math>, <math>\mathcal{L}_2 = (\{2\}, \{1\})</math>.</p> <p><b>PD:</b> <math>\mathcal{G}_1 = (\{1, 2\})</math>, <math>\mathcal{G}_2 = (\{2\}, \{1\})</math>, <math>\mathcal{L}_1 = \mathcal{L}_2 = (\{2\}, \{1\})</math>, <math>d_{1,1} = 19.9</math>, <math>d_{1,2} = 1</math>.</p> <p><b>PT:</b> <math>\mathcal{G}_1 = \mathcal{G}_2 = (\{1\}, \{2\})</math>, <math>\mathcal{L}_1 = \mathcal{L}_2 = (\{2\}, \{1\})</math>, No threshold.</p> <p><b>PDT:</b> <math>\mathcal{G}_1 = (\{1, 2\})</math>, <math>\mathcal{G}_2 = (\{1\}, \{2\})</math>, <math>\mathcal{L}_1 = \mathcal{L}_2 = (\{2\}, \{1\})</math>, <math>d_{1,1} = 15</math>, <math>d_{1,2} = 19.7</math>, <math>d_{2,2} = 0.01</math>, No threshold.</p> <p><b>WR:</b> <math>q_{1,1} = 0.071</math>, <math>q_{1,2} = 0.47</math>, <math>q_{2,1} = 0.33</math>, <math>q_{2,2} = 1.1</math>, <math>a_{1,1} = -9.0</math>, <math>a_{1,2} = -77</math>, <math>a_{2,1} = -9.1</math>, <math>a_{2,2} = 11</math>, <math>b_{1,1} = 42</math>, <math>b_{1,2} = 73</math>, <math>b_{2,1} = 10</math>, <math>b_{2,2} = 7.2</math>.</p>

**Table 3** Solutions for the M-model examples under the objective function  $F_S^1$ .

Case	Routing solutions
1	<p><b>P:</b> <math>\mathcal{G}_1 = (\{1\}, \{3\})</math>, <math>\mathcal{G}_2 = (\{2\}, \{3\})</math>, <math>\mathcal{L}_3 = (\{2\}, \{1\})</math>.</p> <p><b>PD:</b> <math>\mathcal{G}_1 = (\{1, 3\})</math>, <math>\mathcal{G}_2 = (\{2\}, \{3\})</math>, <math>\mathcal{L}_3 = (\{1\}, \{2\})</math>, <math>d_{1,3} = 18.2</math>, <math>d_{2,3} = 19.6</math>.</p> <p><b>PT:</b> <math>\mathcal{G}_1 = (\{1\}, \{3\})</math>, <math>\mathcal{G}_2 = (\{2\}, \{3\})</math>, <math>\mathcal{L}_3 = (\{2\}, \{1\})</math>, No threshold.</p> <p><b>PDT:</b> <math>\mathcal{G}_1 = (\{1, 3\})</math>, <math>\mathcal{G}_2 = (\{3\}, \{2\})</math>, <math>\mathcal{L}_3 = (\{1\}, \{2\})</math>, <math>d_{1,3} = 18.2</math>, <math>d_{2,3} = 17.2</math>, No threshold.</p> <p><b>WR:</b> <math>q_{1,1} = 13</math>, <math>q_{1,3} = 1.3</math>, <math>q_{2,2} = 12</math>, <math>q_{2,3} = 1.6</math>, <math>a_{1,1} = 1475</math>, <math>a_{1,3} = -49</math>, <math>a_{2,2} = 1465</math>, <math>a_{2,3} = -62</math>, <math>b_{1,1} = 2007</math>, <math>b_{1,3} = 60</math>, <math>b_{2,2} = 2008</math>, <math>b_{2,3} = 69</math>.</p>
2	<p><b>P:</b> <math>\mathcal{G}_1 = (\{3\}, \{1\})</math>, <math>\mathcal{G}_2 = (\{2\}, \{3\})</math>, <math>\mathcal{L}_3 = (\{1\}, \{2\})</math>.</p> <p><b>PD:</b> <math>\mathcal{G}_1 = (\{1, 3\})</math>, <math>\mathcal{G}_2 = (\{2, 3\})</math>, <math>\mathcal{L}_3 = (\{1\}, \{2\})</math>, <math>d_{1,3} = 0.2</math>, <math>d_{2,3} = 19.9</math>.</p> <p><b>PT:</b> <math>\mathcal{G}_1 = (\{3\}, \{1\})</math>, <math>\mathcal{G}_2 = (\{2\}, \{3\})</math>, <math>\mathcal{L}_3 = (\{1\}, \{2\})</math>, No threshold.</p> <p><b>PDT:</b> <math>\mathcal{G}_1 = (\{1, 3\})</math>, <math>\mathcal{G}_2 = (\{2, 3\})</math>, <math>\mathcal{L}_3 = (\{1\}, \{2\})</math>, <math>d_{1,3} = 0.3</math>, <math>d_{2,3} = 19.5</math>, No threshold.</p> <p><b>WR:</b> <math>q_{1,1} = 16</math>, <math>q_{1,3} = 1.1</math>, <math>q_{2,2} = 23</math>, <math>q_{2,3} = 1</math>, <math>a_{1,1} = 91</math>, <math>a_{1,3} = -30</math>, <math>a_{2,2} = 46</math>, <math>a_{2,3} = -129</math>, <math>b_{1,1} = 130</math>, <math>b_{1,3} = 60</math>, <math>b_{2,2} = 491</math>, <math>b_{2,3} = 75</math>.</p>
3	<p><b>P:</b> <math>\mathcal{G}_1 = (\{1\}, \{3\})</math>, <math>\mathcal{G}_2 = (\{2, 3\})</math>, <math>\mathcal{L}_3 = (\{2\}, \{1\})</math>.</p> <p><b>PD:</b> <math>\mathcal{G}_1 = (\{1\}, \{3\})</math>, <math>\mathcal{G}_2 = (\{2, 3\})</math>, <math>\mathcal{L}_3 = (\{2\}, \{1\})</math>, <math>d_{1,3} = 19.9</math>, <math>d_{2,3} = 0.01</math>.</p> <p><b>PT:</b> <math>\mathcal{G}_1 = (\{1\}, \{3\})</math>, <math>\mathcal{G}_2 = (\{2, 3\})</math>, <math>\mathcal{L}_3 = (\{2\}, \{1\})</math>, No threshold.</p> <p><b>PDT:</b> <math>\mathcal{G}_1 = (\{1\}, \{3\})</math>, <math>\mathcal{G}_2 = (\{2, 3\})</math>, <math>\mathcal{L}_3 = (\{2\}, \{1\})</math>, <math>d_{1,1} = 2</math>, <math>d_{1,3} = 19.9</math>, No threshold.</p> <p><b>WR:</b> <math>q_{1,1} = 12</math>, <math>q_{1,3} = 0.5</math>, <math>q_{2,2} = 15</math>, <math>q_{2,3} = 2.6</math>, <math>a_{1,1} = 1461</math>, <math>a_{1,3} = -80</math>, <math>a_{2,2} = 1743</math>, <math>a_{2,3} = -29</math>, <math>b_{1,1} = 2127</math>, <math>b_{1,3} = 25</math>, <math>b_{2,2} = 2359</math>, <math>b_{2,3} = 144</math>.</p>