A Staffing Algorithm for Call Centers with Skill-Based Routing

Ward Whitt

IEOR Department, Columbia University

http://www.columbia.edu/∼ww2040

Joint work with:

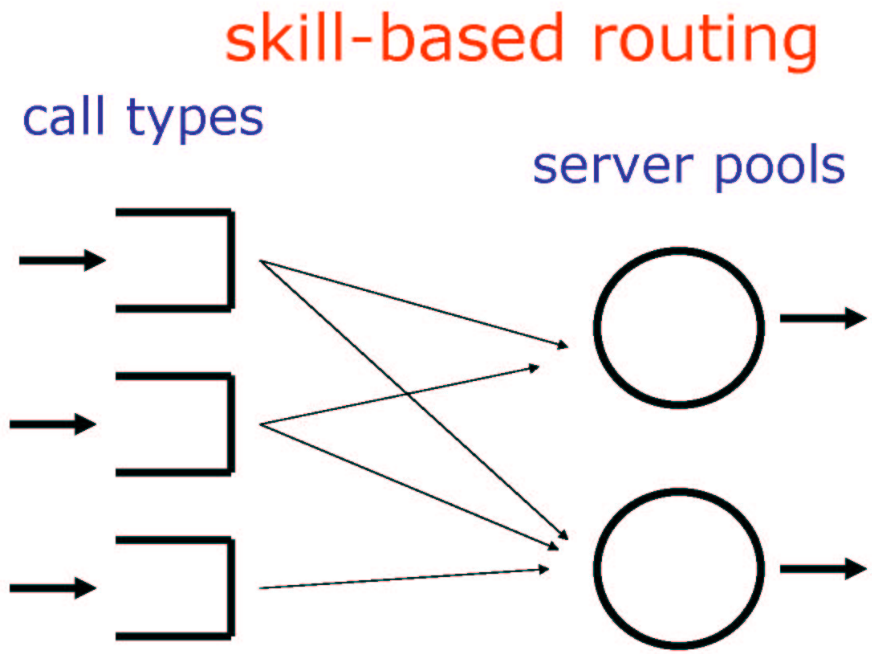
Rodney B. Wallace

IBM and George Washington University

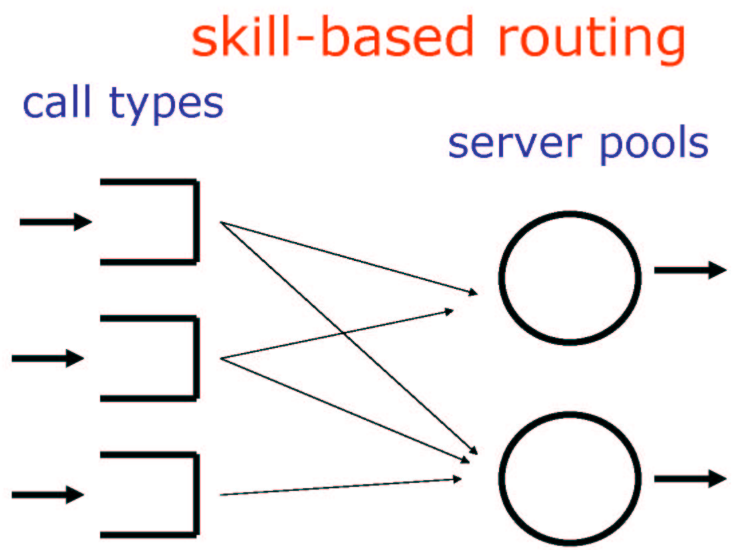
Thesis: Performance Modelling and Design of Call Centers with Skill-Based Routing

Advisors: William A. Massey (Princeton), Thomas A. Mazzuchi (GW) and Ward Whitt (Columbia)

Multiple Types of Calls and Agents



Multiple Types of Calls and Agents



Special case: The service-time distribution does not depend on the call type or the agent.

First Contribution:

Routing and Provisioning Algorithm

Minimize the Required Staff and Telephone Lines

While Meeting the Service level Agreement (SLA)

$$\mathbf{P(Delay} \leq 30 \ \mathbf{seconds)} \geq 0.80$$

$$\mathbf{P(Blocking)} \leq 0.005$$

(service level may depend on call type)

Second Contribution:

Demonstrate Resource-Pooling Phenomenon

A small amount of cross training (multiple skills) produces almost the same performance as if all agents had all skills (as in the single-type case).

Simulation Experiments

Precedents

"A little bit of flexibility goes a long way."

Joining One of Many Queues

- Azar, Broder, Karlin and Upfal (1994)
- Vvedenskaya, Dobrushin and Karpelovich (1996)
- Turner (1996, 1998)
- Mitzenmacher (1996) and
- Mitzenmacher and Vöcking (1999)

Flexible Manufacturing: Chaining

- Jordan and Graves (1995)
- Aksin and Karaesman (2002)
- Hopp and Van Oyen (2003)
- Jordan, Inman and Blumenfeld (2003)
- Gurumurthi and Benjaafar (2004)

Outline

1. **SBR Call-Center Model**
2. **Resource-Pooling Experiment**
3. **Provisioning Algorithm**
4. **Simulation to Show Performance**

$M_n/M_n/C/K/NPrPr$ **SBR Call Center**

1. C agents, $C + K$ telephone trunklines, and n call types.

2. *Non-preemptive Priorities (NPrPr)* - Calls are processed in priority order. Calls are worked to completion once they are handed to an agent.

3. *Longest-Idle-Agent Routing (LIAR) Policy* - Calls are forwarded to the agent who has been waiting the longest since his last job completion and has the highest skill to handle the request.

Agent-Skill Matrix - $C \times n$

4. *Agent-Skill Profile* - Predefined in an agent-skill matrix $A \equiv (a_{ij})$ as

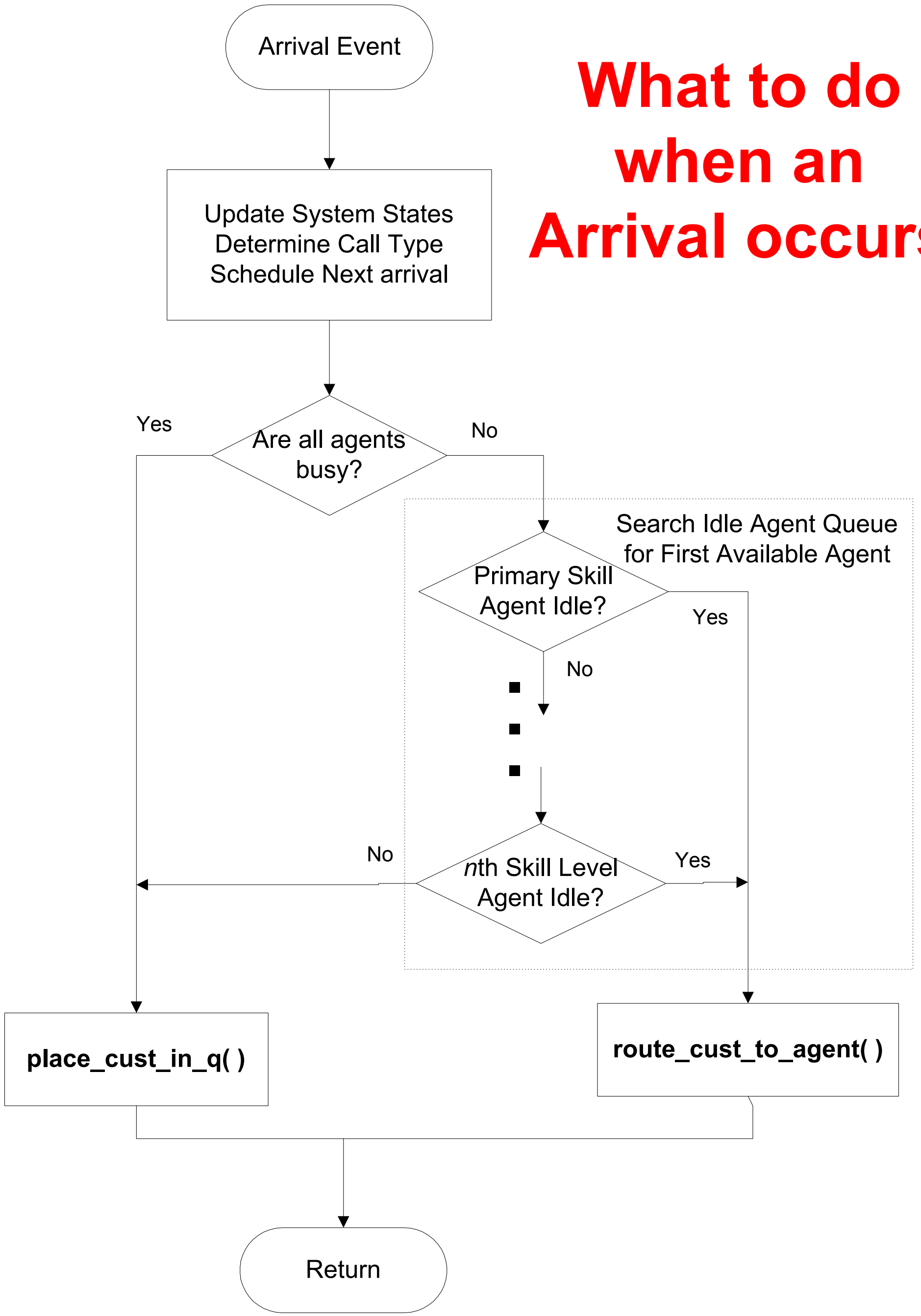
$$a_{ij} = \begin{cases} k & \text{when agent } i \text{ supports call type } k \\ & \text{ at priority level } j \text{ (primary, secondary, etc),} \\ 0 & \text{otherwise.} \end{cases}$$

where $i = 1, \ldots, C$, $1 \leq k \leq n$, and $1 \leq j \leq n$.

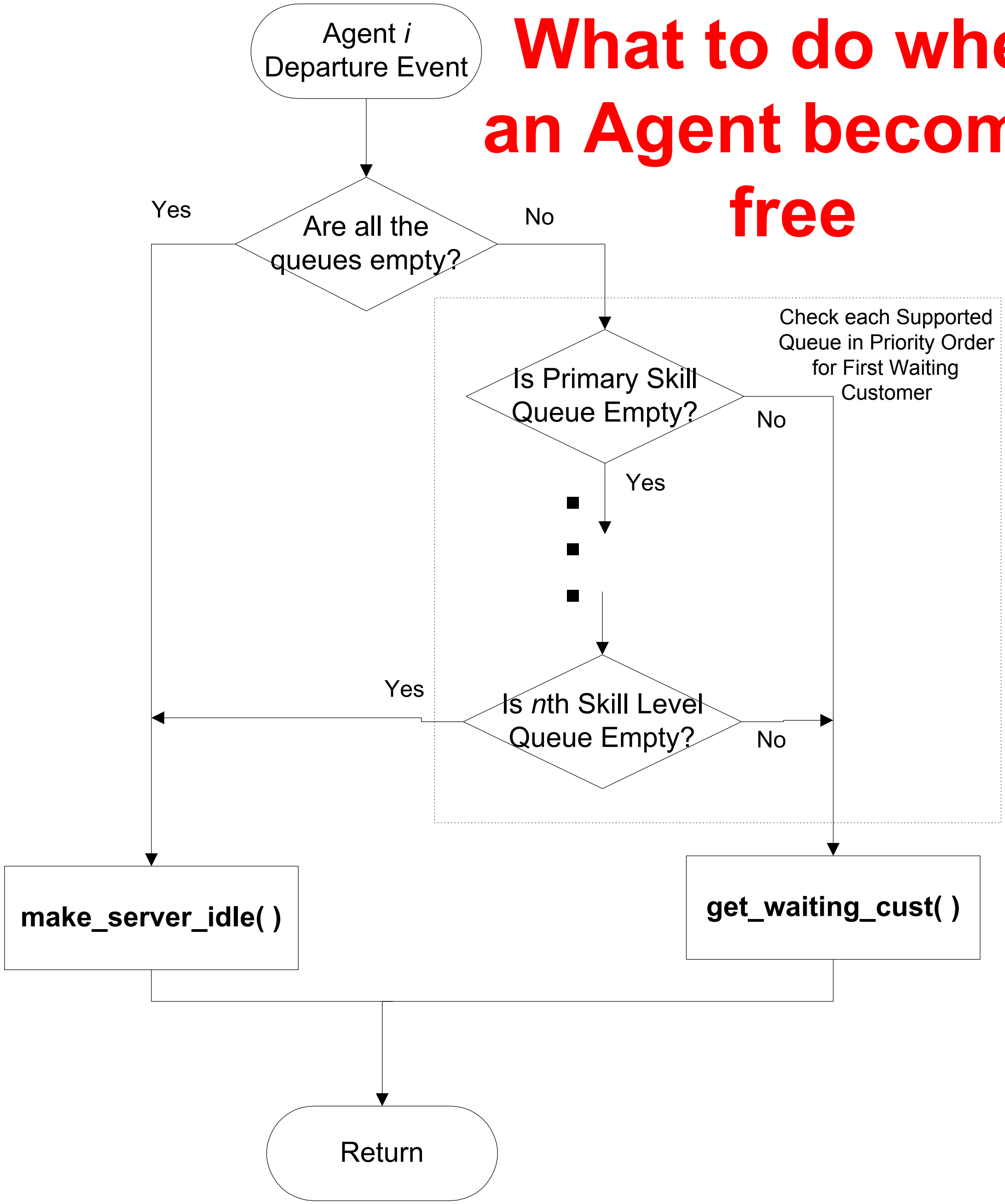
Examples:

$$\mathbf{A}_{5\times 1} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad \mathbf{A}^{(1)}_{3\times 2} = \begin{pmatrix} 1 & 0 \\ 2 & 0 \\ 2 & 0 \end{pmatrix}, \quad \mathbf{A}_{4\times 2} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 2 & 1 \\ 2 & 1 \end{pmatrix}, \quad \mathbf{A}_{6\times 4} = \begin{pmatrix} 3 & 4 & 1 & 0 \\ 1 & 4 & 0 & 0 \\ 2 & 3 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 3 & 1 & 2 & 4 \\ 1 & 0 & 4 & 0 \end{pmatrix}$$

What to do when an Arrival occurs



What to do when an Agent becomes free



Resource-Pooling Experiment

Model Assumptions

1. **Arrival Process** - n types of calls arrive at the call center according to n mutually independent Poisson processes with rate λ_i, $1 \leq i \leq n$.
   [$n = 6$, $\lambda_i = 1.40$ for all i]

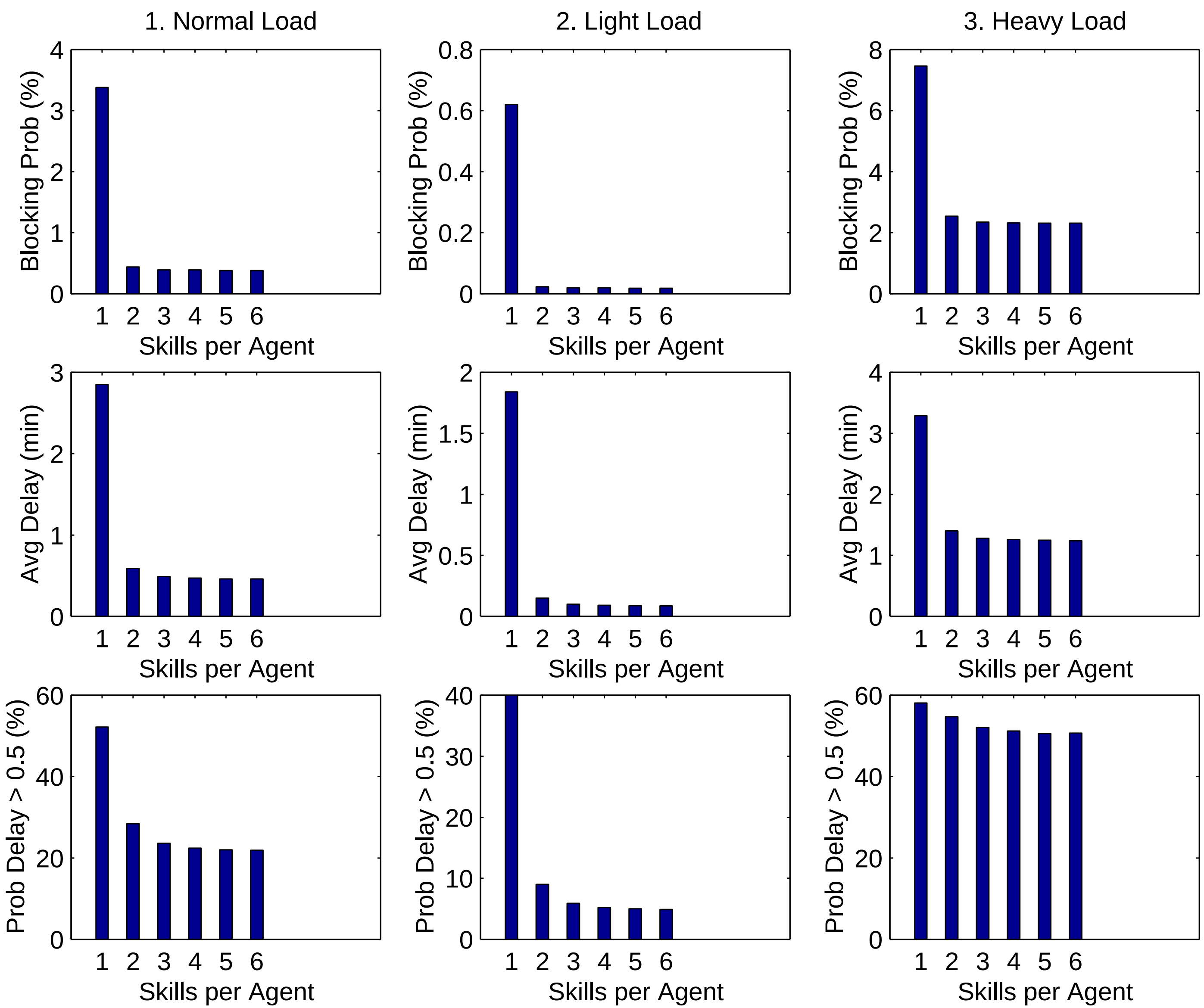
2. **Service Time Process** - Call holding (service) times are mutually independent exponential random variables with mean $1/\mu_i$ which are independent of the arrival process, $1 \leq i \leq n$.
   [$1/\mu_i = 1/\mu = 10$ minutes for all i]

3. **Offered Loads** - $\alpha_i = \lambda_i/\mu_i$
   [$\alpha_i = 14$ for all i, so the total offered load is $\alpha = 84$]

4. **Agents and Telephone Lines**
   [$C = 90$ and $K = 30$ $(C + K = 120)$]

Agents are given k skills, $1 \leq k \leq 6$

Three Loads: Normal (84), Light (77.4), Heavy (90)


1. Normal Load
2. Light Load
3. Heavy Load
Blocking Prob (%)
Avg Delay (min)
Prob Delay > 0.5 (%)
Skills per Agent

Provisioning Algorithm

Find C, K and A

So that each agent has at most 2 skills and all performance constraints are met.

How do we know it works?

The optimal values of C and K

are almost the same as for M/M/C/K which occurs with a single call type.

Balanced Example

M/M/C/K: $C = 90$ and $K = 19$

SBR: $C = 91$ and $K = 20$

SBR Balanced Provisioning Example

- Call volume is $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5 = \lambda_6 = 1.375$,

- Service times are $1/\mu_1 = \ldots = 1/\mu_6 = 10$ mins
- Agents Skill Profile: Agents have 2 skills each.
- Service level targets
  1. Blocking service level target is 0.5%.
  2. 80% of the calls are answered within $\tau = 0.5$ minute.
- Square-root safety method for distributing agents into work groups is used.
- It is known that the total number of agents required is between 90 (best-case) and 106 (worse-case). Similarly, the the telephone trunkline capacity is between 111 and 156.

Unbalanced Example

M/M/C/K: $C = 90$ and $K = 19$

SBR: $C = 91$ and $K = 21$

SBR Unbalanced Provisioning Example

- Call volume is $\lambda_1 = \lambda_2 = 0.425$, $\lambda_3 = 1.05$, $\lambda_4 = 1.375$, $\lambda_5 = 1.925$, and $\lambda_6 = 3.05$ calls/min.

- Service times are $1/\mu_1 = \ldots = 1/\mu_6 = 10$ mins

- Agents Skill Profile: Agents have 2 skills each.

- Service level targets
  1. Blocking service level target is 0.5%.
  2. 80% of the calls are answered within $\tau = 0.5$ minute.

- Square-root safety method for distributing agents into work groups is used.

- It is known that the total number of agents required is between 90 (best-case) and 106 (worse-case). Similarly, the the telephone trunkline capacity is between 111 and 156.

Unbalanced SBR Provisioning Example Summary

	Best Case	Actual Perf.	Worst Case
$(C, C + K)$	(90, 109)	(91, 111)	(106, 156)
Workgroup 1 C_1		7	7
Workgroup 2 C_2		7	7
Workgroup 3 C_3		13	14
Workgroup 4 C_4		15	18
Workgroup 5 C_5		21	24
Workgroup 6 C_6		28	36

SBR Provisioning

- Solves the problem of determining the minimum number of agents C and the minimum number of telephone trunklines $C + K$ needed to meet service level targets.

- Exploits resource pooling results.

- Exploits $M/M/C/K$ results to determine initial estimate for (C, K).

- Uses fair agent skill assignment scheme to construct agent skill matrix satisfying general agent skill profile.

- Simulation runs are performed to make improvements on the initial assignment using a heuristic search algorithm.

Determining Primary Skills

$$C_k = \alpha_k + x\sqrt{\alpha_k}$$

$$x = \frac{(C-\alpha)}{\sum_{i=1}^{n}\sqrt{\alpha_i}}$$

and round

Determining Secondary Skills

$$C_{i,k} = \frac{C_i C_k}{C - C_i}$$

and round

STOP

Initial SBR Provisioning Algorithm				
	Number of Iterations (Agents)			
Performance Measure	**1 (90)**	**2 (91)**	**3 (92)**	**4 (93)**
1. Blocking (%)	0.53	0.42	0.36	0.30
4. $\mathcal{P}(\text{Delay} \leq 0.5 \mid \text{entry})$	81.3	83.9	86.5	88.8
5. $\mathcal{P}(\text{Delay}_1 \leq 0.5 \mid \text{entry})$	68.3	75.5	78.4	80.5
5. $\mathcal{P}(\text{Delay}_2 \leq 0.5 \mid \text{entry})$	65.2	74.9	77.8	80.3
5. $\mathcal{P}(\text{Delay}_3 \leq 0.5 \mid \text{entry})$	79.7	81.8	84.7	88.0
5. $\mathcal{P}(\text{Delay}_4 \leq 0.5 \mid \text{entry})$	82.0	83.6	86.5	88.8
5. $\mathcal{P}(\text{Delay}_5 \leq 0.5 \mid \text{entry})$	83.4	86.2	87.8	89.8
5. $\mathcal{P}(\text{Delay}_6 \leq 0.5 \mid \text{entry})$	84.4	85.8	88.7	90.9

Refined SBR Provisioning Algorithm

Performance Measure	Number of Iterations (Agents) 4 (93)	5 (92)	6 (92)	7 (91)	**8 (91)**	9 (90)
1. Blocking (%)	0.30	0.35	0.36	0.43	**0.44**	**0.54**
4. $\mathcal{P}$(Delay $\leq$ 0.5|entry)	88.8	86.5	86.2	83.4	82.9	**79.8**
5. $\mathcal{P}$(Delay$_1$ $\leq$ 0.5|entry)	80.5	78.0	81.6	78.6	**82.6**	80.0
5. $\mathcal{P}$Delay$_2$ $\leq$ 0.5|entry)	80.3	77.6	81.4	78.6	**81.9**	79.7
5. $\mathcal{P}$Delay$_3$ $\leq$ 0.5|entry)	88.0	86.1	85.8	83.6	**83.4**	78.6
5. $\mathcal{P}$Delay$_4$ $\leq$ 0.5|entry)	88.8	87.2	87.0	83.2	**82.6**	80.5
5. $\mathcal{P}$Delay$_5$ $\leq$ 0.5|entry)	89.8	87.7	86.7	84.6	**83.1**	79.4
5. $\mathcal{P}$Delay$_6$ $\leq$ 0.5|entry)	90.9	88.0	86.9	84.1	**82.9**	80.3