Additional Supplement
Capacity Planning for Resource Turnaround Operations

1. Break and End-of-Shift Service Assumption

Table 9 Example system with one arrival in period 1, one departure in period 1, one server working only in period 1, another server working only in period 6, and a cleaning duration of $H = 2$ periods ($d_0 = a_0 = I_0 = V_0 = 0$).

Time Period	t	1	2	3	4	5	6	7	8
Departures	d_t	1	0	0	0	0	0	0	0
Arrivals	a_t	1	0	0	0	0	0	0	0
Active Servers	z_t	1	0	0	0	0	1	0	0
Available Servers	R_t	1	-1	1	-1	1	0	0	0
Vacant Dirty Resources	V_t	1	0	1	0	1	0	0	0
Resources for which Service Starts	S_t	1	-1	1	-1	1	0	0	0
Inventory Position	I_t	-1	-1	0	-1	0	-1	0	0
Waiting Customers	W_t	1	1	0	1	0	1	0	0

To avoid complicating our state equations to handle a server's break or shift end when no other servers are available, we assume (5) is the number of available servers in any period. Table 9 shows how this assumption works for a simple system with one arrival and one departure and a resource with a turnaround time of $H = 2$ periods. One server provides one of the two required time units of service to the resource in period 1 and is no longer available. Another server is available in period 6 and completes the service so that the resource can be allocated at the beginning of period 7. Our assumption means that the number of available servers oscillates between -1 and 1 from periods 2 to 5, and the number of waiting customers oscillates between 1 and 0. Thus, a waiting cost is assessed for half of the periods during which the resource is waiting for service to resume.

We offer an alternate model that more explicitly models server availability during breaks and at the end of a shift in the additional supplement. While we do not have analytical results for this model, our numerical testing leads us to believe that the alternate model has the same properties as the model presented in this section. Also, only negligible differences exist for the two models in their estimation of policy performance.

This approach represents a compromise in which the key performance metrics are bounded by two policies: servers do not begin servicing a resource if they cannot complete it before a break or shift completion, and servers delay their break to complete servicing the resource or stay overtime and complete their work. In effect, any rooms not completed by the break are fractionally allocated to a guest until sufficient capacity is available again.

2. Alternate Model for End-of-Shift and Break Policies

We describe the state equations for an alternate model of room cleaning that avoids the assumption stated in (5). In particular, this model explicitly tracks the number of time periods of cleaning required for each vacant dirty resource and does not allow a resource to be allocated until it is fully cleaned. We assume that resources with the fewest number of periods of cleaning remaining preempt cleaning of any other resources; e.g., if the server cleaning a resource with only 1 period of cleaning remaining goes on break, a server working on a resource with many more periods remaining will switch to that resource. While this may not exactly reflect common practice — i.e., a hotel room attendant might just complete a room before going on break — this assumption allows for relatively simple state equations that avoid dynamic decision making about break timing. If the server were to wait to take a break until the resource is completely serviced, the model's estimate of resource availability would be slightly conservative.

Let $V_{t,k}(\boldsymbol{y})$ represent the number of vacant dirty resources in period $t = 0,\ldots,T$ with $k = 1,\ldots,H$ periods of cleaning remaining until it can be allocated to a new customer. Also, we use $S_{t,k}(\boldsymbol{y})$ to denote the number of resources with k servicing periods remaining that are serviced in period t. The $z_t(\boldsymbol{y})$ active servers may each perform one period of servicing; i.e., $\min\left\{z_t(\boldsymbol{y}), \sum_{k=1}^{H} V_{t,k}(\boldsymbol{y})\right\}$ resources receive servicing so that their state advances from k to $k-1$ in period t. Resource service priority increases as k decreases so that, in each period $t = 1,\ldots,T$ it can be defined recursively as

$$S_{t,k}(\boldsymbol{y}) := \min\left\{V_{t,k}(\boldsymbol{y}), z_t(\boldsymbol{y}) - \sum_{n=1}^{k-1} S_{t,n}(\boldsymbol{y})\right\} \tag{49}$$

for $k = 1,\ldots,H$. Furthermore, we also have the following recursive state equations for each period t:

$$V_{t,H}(\boldsymbol{y}) := V_{t-1,H}(\boldsymbol{y}) + d_t - S_{t-1,H}(\boldsymbol{y}), \tag{50}$$

$$V_{t,k}(\boldsymbol{y}) := V_{t-1,k}(\boldsymbol{y}) + S_{t-1,k+1}(\boldsymbol{y}) - S_{t-1,k}(\boldsymbol{y}), \qquad k = 1,\ldots,H-1, \tag{51}$$

$$I_t(\boldsymbol{y}) := I_{t-1}(\boldsymbol{y}) - a_t + S_{t-1,1}(\boldsymbol{y}). \tag{52}$$

All other system definitions from Section 3 continue to apply.

3. Additional Results for Workers with Short Shifts

In our conversations with hotel workforce strategists, they identified workers interested in shifts lasting less than 8 hours as a labor pool that hotels might need to utilize. For example, they envision high school or college students as labor sources for working shorter shifts at hotels in urban markets. Alternatively, part-time workers in this model could represent cross-trained workers who clean rooms for half of their shift and fulfill some other role for the remainder of their shift. In theory, using part-time workers should be advantageous for hotels due to the ability to schedule capacity when it is most needed. For our comparison, part-time room attendants have a shift cost and timing structure that maintains the same cost per room cleaned as full-time workers when fully utilized: $k_p = \$120$ for a four-hour shift, which includes 12.5 minutes of start-up time before cleaning the first room and no breaks during the four-hour shift. Thus, two part-time room attendants have the same pay and cleaning capacity as one full-time room attendant.

We next consider a model in which the hotel uses room attendants who work four-hour shifts. For convenience, we refer to the room attendants who only work four-hour shifts as *part-time* workers. While using only part-time workers is implausible in practice, it points to the usefulness of alternate labor sources for hotels procured either through gig platforms or by cross-training other workers on the property.

Table 10 shows the optimal schedule with the option to choose full-time or part-time room attendants and 1,300 rooms to clean. Highlighting the value of flexible capacity that can be employed with more precision, the optimal solution chooses 200 part-time room attendants and 0 full-time room attendants. This result is expected as two part-time workers have the same cleaning capacity and cost as one full-time worker but allow for added flexibility. Because it is possible to achieve a solution with negligible waiting using only full-time workers, the benefits of part-time workers are not immediately obvious for this specific scenario. However, under a more realistic scenario with a maximum of 50 part-time workers, part-time workers can sometimes help the hotel reducing the number of full-time equivalent workers through better targeting capacity allocation. In particular, Table 11 shows a decrease in

Table 10 **Optimal schedules with part-time room attendants working 4-hour shifts and 1,300 rooms to clean.**

		Shift Start Time											Total
		8 AM	9 AM	10 AM	11 AM	12 PM	1 PM	2 PM	3 PM	4 PM	5 PM	6 PM	
Current Practice		0	90	0	0	0	0	0	0	10	0	0	100
Optimal Full-time		34	23	22	3	5	5	5	0	0	1	1	100
Optimal with Part-time	Part-Time	5	74	9	22	19	0	1	2	51	2	15	200
	Full-Time	0	0	0	0	0	0	0	0	0	0	0	0

Table 11 **Optimal schedules with a maximum of 50 part-time room attendants available to work 4-hour shifts.**

Number of Rooms to Clean		Schedule											Total
		8 AM	9 AM	10 AM	11 AM	12 AM	1 PM	2 PM	3 PM	4 PM	5 PM	6 PM	
1600	Part-Time	9	0	0	0	0	0	0	0	21	3	17	50
	Full-Time	15	26	42	8	6	4	0	0	0	0	0	101
1900	Part-Time	3	0	0	0	0	0	0	7	18	1	21	50
	Full-Time	17	44	42	13	9	0	0	0	0	0	0	125

Table 12 **Solution accuracy based on the number of sample paths (solving and testing) and 100 replications.**

Evaluation Design	Performance Measure		
Sample Paths	MSE	MAPE	Optimal Rate
50	\$0.79	0.0006%	98%
100	\$0.26	0.0003%	100%
200	\$0.25	0.0003%	100%

the required staffing level for the scenarios with a high volume of rooms to clean. For the scenarios with 1,600 and 1,900 rooms to clean, using part-time workers can reduce the staffing level needed by the equivalent of 3 and 6 full-time workers, respectively. In both cases, the waiting costs remain less than \$5. Part-time workers appear to be particularly valuable in the late afternoon and evening. This analysis shows the usefulness of part-time workers or alternate sources of shorter-term capacity in helping a hotel to clean all rooms in time to avoid customer waiting.

4. Sample Average Approximation Accuracy Test

We numerically determine the size of the sample paths required to solve our problem using sample average approximation. method numerically. For different sizes of solving sample path, we generate 100 sets of solving and testing sample paths. Then, we find the optimal solution for the solving sample paths and evaluate the resulting solution on the testing sample paths. Finally, we solve the testing sample path optimally to check if it corresponds to the solution given by the solving sample path. Table 12 confirms that 100 solving sample paths and 100 testing sample paths are sufficient to solve the problem: the in-sample solution is 100% optimal when evaluated on the testing sample paths. Also, the in-sample average total cost is nearly identical to that of the testing sample paths. Therefore, we set the solving and testing sample-paths size to be 100 for our analysis.