Chapter 8
Simulation-Based Optimization: A Case Study for Airline's Cargo Service Call Center

Tiravat Assavapokee[1] **and Ibrahim Mourtada**[2]

Abstract In this chapter, we introduce the basic concept of the simulation-based optimization and illustrate its usefulness and applicability for generating the manpower planning of airline's cargo service call center. Because of the continuous increase in oil prices, and combined with many other factors, the airline industry is currently facing new challenges to keep its customers satisfied. One of the most important drivers of the customer satisfaction is the customer service. The excellent customer service can give an airline company the edge over its competitors. Airline companies need to insure the appropriate level of staffing at their service call centers in order to maintain a high level of customer satisfaction with the appropriate level of the overall cost. With the high level of uncertainty in the customer demand and a number of complicated factors in the problem, it becomes necessary to apply the simulation-based optimization technique to help managers generate the efficient staffing policy for the airline's cargo service call center. In this work, the technique called reinforcement learning and Markov decision process are used to build and solve the mathematical model to determine the appropriate staffing policy at the airline's cargo service call center on the monthly basis. Simulation and optimization models are incorporated together so as to solve the overall problem. The results of the case study are thoroughly analyzed, discussed, and compared with the current staffing policies. All results illustrate the impressive performance of the recommended staffing policies over the current staffing policies.

8.1 Introduction

Simulation and optimization are clearly two of the most widely implemented operation research and management science techniques in practice. There have been several obstacles that limit acceptance and usefulness of simulation and optimization techniques in the past. For example, developing simulation and optimization

[1] Assistance Professor, Department of Industrial Engineering, University of Houston, Houston, Texas 77204, USA

[2] Manufacturing Engineer, Cameron Measurement Systems, Houston, Texas, 77032, USA

A.K. Kamrani, E.S. Abouel Nasr (eds.) *Collaborative Engineering: Theory and Practice,*
doi: 10.2007/ 978-0-387-47321-5, © Springer Science + Business Media, LLC 2008

models for the large-scale real-world systems tends to be a very complex task. In addition, writing the computer code to execute and solve these models can also be another difficult and time-consuming process. Because of the recent advance in computer technology and recent development of modern simulation and optimization software, these obstacles have been significantly reduced (if not eliminated). Complex simulation and optimization models can now be developed much easier in recent years by utilizing the modern software packages that conveniently provide many of the features required to develop these models. In addition, one can now run the simulation and optimization models of complex systems much faster as computers become much more powerful.

Simulation refers to the broad concepts of operation research methodologies and techniques that imitate the behavior of the real-world system. Simulation is usually used to study and to improve the performance of the existing system or to design a new system under uncertainty without actually experimenting with the actual physical system. This feature makes simulation a very powerful operation research technique in practice because it is often too difficult and costly to perform physical studies on the actual system. Simulation is often used as evaluation tools to answer many important “what if” questions that decision makers may have about the system. For example, decision makers can use simulation to answer the question such as: “what would happen to the performance of the factory if the layout is changed?” Even though, simulation can be used to efficiently evaluate the system performance for a given solution, it is not capable of recommending the best solution for the complex decision-making problems by itself.

Optimization refers to the broad concepts of operation research methodologies and techniques that model the complex decision-making problems and recommend the best solution to these problems. Optimization is certainly one of the most powerful operation research techniques and it pervades the fields of engineering, science, and business. To apply optimization techniques, decision makers have to first formulate the mathematical models that capture the decision-making problems. The appropriate optimization techniques are then applied to find the solutions to these models. The general goal of optimization is to find the solution that yields the best value of the performance criterion under some restrictions in the decision-making problems. In many cases, the real-world decision-making problems cannot be fully represented by the mathematical models. Decision makers are often required to make a number of assumptions in order to construct the appropriate mathematical models for these problems. As a consequence of making these assumptions, the solution obtained by solving these mathematical models may not be fully applicable for some real-world decision-making problems.

Because of the usefulness and applicability of these two powerful operation research techniques, researchers and practitioners have always been trying to combine simulation and optimization techniques into an even more powerful decision-making tool. In fact, simulation-based optimization is not a new topic in operation research and management science literature. Since the time that the computer systems was invented and started making an impact on practical decision-making processes and scientific researches, researchers and practitioners have always wanted to optimize their decision-making systems by utilizing simulation models. However, it is only recently that remarkable success in realizing this objective has been seen in practice due to the dramatic increase in the power of computer systems over the years. Simulation-based optimization now has so much potential in almost every area of decision-making processes under uncertainty.

In Sect. 8.2, we briefly review the literature in the areas of simulation-based optimization and service call center staff planning. In Sect. 8.3, we discuss the basic reinforcement learning (RL) methodology. In Sect. 8.4, the case study from the real airline industry is discussed and the results from the case study are thoroughly analyzed and illustrated. We then conclude the chapter and give the summary of the overall work in Sect. 8.5.

8.2 Literature Review

In this section, we summarize a number of literatures related to simulation-based optimization techniques and service call center staff planning.

8.2.1 Literature Review for Simulation-Based Optimization and RL

As discussed earlier, simulation is a very powerful decision-making tool to perform "what if" analysis of the complex systems. Recent research discovery illustrates that simulation can be coupled with powerful optimization algorithms to solve complex real-world problems. The effectiveness of this approach depends on the quality of the simulation model that represents the real-world system. A high degree of understanding of the system being studied is often required. The book written by Gosavi [1] gives a good introduction to the topics of simulation-based optimization and RL techniques. Kleinman et al. [2] show that reductions in the cost of the airline delay can be obtained by using a simulation optimization procedure to process delay cost measurements. They discuss how the optimization procedure called simultaneous perturbation stochastic approximation (SPSA) can be used to process delay cost measurements from air traffic simulation packages and produce an optimal gate holding strategy. Rosenberger et al. [3] developed a stochastic model for airline operations by using a simulation package called SIMAIR. The developed model is not modular and does not allow other recovery procedures to be integrated. Lee et al. [4] used their model to propose a modular method of approaching the problem that can deal with different recovery procedures from different airlines.

Even though there are dramatic advances in the field of operation research and computer science over the past decade, there are still lots of work to be done to come up with the efficient methodologies and software to solve the complicated real-life problems. Many of these problems are currently unsolvable, not because current computer systems are too slow or have too little memory, but simply because it is too difficult to determine what the computer program should do to solve these complicated problems. If the computer program could learn to solve the problems by itself, this would result in a great contribution in the field of operation research and computer science. RL is one such approach that makes the computer program to learn while trying to solve the complex decision-making problems. RL dates back to the early days of cybernetics and work in statistics, psychology, neuroscience, and computer science. In the last decade, it has rapidly attracted increasing interest in the machine learning and artificial intelligence communities. RL has significant potential in advancing parameters and policy optimization techniques. Sutton and Barto [5] and Bertsekas and Tsitsiklis [6] provide an excellent background reading for this field. Comprehensive literature surveys of pre-1996 research have been published by Kaelbling et al. [7] and Mahadevan [8]. Creighton and Nahavandi [*9*] developed a MATLAB toolbox to allow an RL agent to be rapidly tuned to optimize a multipart serial line. Aydin and Oztemel [10] successfully applied RL agents to dynamic job-shop scheduling problems. Other agent-based work in the job scheduling field has also been completed by Jeong [11], Zhang and Dietterich [12], Reidmiller and Reidmiller [13], and Schneider et al. [14]. Several research groups have recently focused on RL agent applications in manufacturing. Paternina-Arboleda and Das [15] used a SMART algorithm on a serial production line to optimize the preventative maintenance in a production inventory system. Mahadevan et al. [16] used this same algorithm and touched upon the integration of intelligent agents using RL algorithms with commercial DES packages. Mahadevan and Theocharous [17] also examined a manufacturing application using RL technique.

8.2.2 Literature Review for Service Call Center Staff Planning

Service call centers are the common way for many companies to communicate with their customers. In the customer point of view, the quality of service at the service call center usually reflects the operational efficiency of the company. Thus, the performance of the service call center is very essential for the survival of the company within our highly competitive service-driven economy. One important issue that many companies have to face is staff planning at their customer service call centers. At a service call center, hundreds of agents may have to answer to several thousands of telephone calls per hour. In addition, the number of calls is usually uncertain and is quite hard to predict from one time period to the next. The design of such an operation has to be based on solid scientific principles. Sze [18] discusses a queuing model of telephone operators at the Bell Communication Research Company, Inc. The queuing model is used to approximate the effects of several features such as general service times, abandonment, reattempts, etc. The results have proved to be quite useful in planning and managing the operator staffing for the service call center. Andrews and Parsons [19] have developed an economic-optimization model for telephone agent staffing at L. L. Bean. The model provides half an hour target staffing levels to an automated scheduler, which generates the specific on duty tours for each individual telemarketing operator. Chen and Henderson [20] discuss difficulties in using historical arrival rates to determine the staffing levels for a call center with priority customers. Fukunaga et al. [21] describe a staff scheduling system for contact centers called Blue Pumpkin Director. Borst et al. [22] use an *M*/*M*/*N* queuing model to build a model for staffing large service call centers with a large number of agents (*N*). Atlason et al. [*23*] use simulation and an iterative cutting plane method to find the staffing plan that minimizes the overall cost of a service system subject to a certain service level over multiple time periods. Atlason et al. [24] use simulation and analytic center cutting plane method to find the staffing plan that minimizes the overall staffing cost in an inbound call center subject to a certain service level. Deslauriers et al. [25] consider a blend call center with both inbound and outbound calls. They present a continuous time Markov chain models to solve the problem. Mourtada [26] considers the staffing problem at the Continental airline service call center and uses the RL technique to solve the problem.

8.3 Simulation-Based Optimization: Rl Technique

In our everyday life, we have to make many decisions. For each decision that we make, we can observe the immediate impact of that decision. It may not be a smart idea to use the immediate consequence of the decision as the only measurement for the quality of that decision. In fact, many decisions that we make have both the immediate consequence and the long-term consequences. By not properly accounting for the relationship between immediate and long-term consequences when making the important decisions, the resulting decisions may not have the good overall performance. For example, in a marathon racing, a racer who runs with the full speed at the beginning may be the leader in the initial phase of the race (good immediate consequence). Unfortunately, this may result in depleting the reserved energy very quickly and finally may result in a very poor finish (poor overall performance).

In this section, we first discuss the theoretical concepts and the general mathematical notations, formulations, and solution methodology of the sequential decision-making problems under uncertainty such that both immediate and long-term consequences have to be considered when making the decision. We will also discuss the difficulties in formulating and solving these models for the real-world decision-making problems. We then introduce the general concepts of RL, which properly combines simulation and optimization techniques to solve these complex decision-making problems under uncertainty.

8.3.1 Sequential Decision-Making System and Markov Decision Process

Figure 8.1 illustrates the general framework of the sequential decision-making system. At a particular point in time before making the decision, hereafter called *decision*

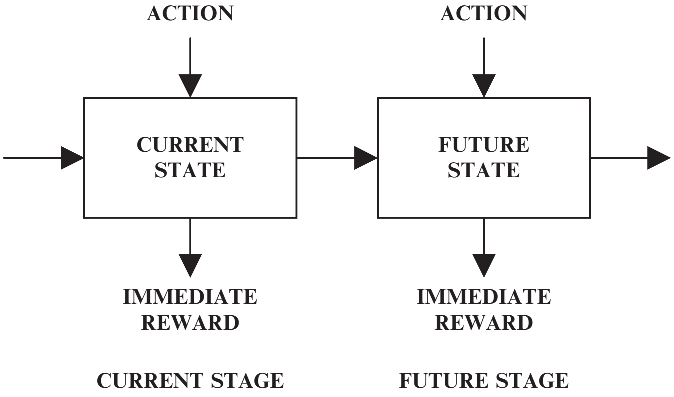


Fig. 8.1 General framework of sequential decision-making systems

stage, the decision maker has to carefully observe the information about the surrounding environment. This information will be hereafter called *system state*. Based on the system state information, the decision maker selects a possible decision, hereafter called *action*. After the appropriate action is chosen, decision maker receives the immediate consequence, hereafter called *immediate reward*, and the system stochastically evolves with some probability distributions, hereafter called *transition probability*, to a new system state at the next decision stage. At this decision stage, the decision maker again faces a similar decision-making problem.

Let us now define the general mathematical notations for the sequential decision-making problems. Let T denote the set of all possible decision stages. Let S denote the set of all possible system states. If at a particular decision stage, the decision maker observes that the system is in the state $s \in S$, he or she may select an action a from the set of all possible actions in the system state s, A_s. Let $A = \cup s \in sA_s$ denote the set of all possible actions. As the result of selecting an action $a \in A_s$ in the system state $s \in S$ at the decision stage $t \in T$, the decision maker receives an immediate reward of $r_t(s,a)$ and the system state at the next decision stage is determined by the transition probability $p_t(\cdot|s,a)$. In this section, we assume that the set S and A_s and the values of $r_t(s,a)$ and $p_t\ (\cdot|s,a)$ do not vary with different decision stages. Because of these assumptions, we will use the notations $r(s,a)$ and $p(\cdot \backslash s,a)$ instead of $r_t(s,a)$ and $p_t\ (\cdot|s,a)$ respectively for the rest of this chapter. We also assume that sets S and A_s are finite and the reward $r(s,a)$ is bounded for all system states and actions. The collection of objects $[T,S,A_s,\ p(\cdot|s,a),\ r(s,a)]$ is referred to as a *Markov decision process* (MDP). To formulate the mathematical models for sequential decision-making problems under uncertainty, decision makers have to properly define this collection of objects. The book written by Puterman [27] summarizes the detailed methodologies and theoretical concepts about MDP.

The solutions of the sequential decision-making problems under uncertainty are represented as *policies*. A *policy* normally refers to the set of selected actions for each state of the system. Without loss of generality, we assume that the decision makers are searching for the policy that maximizes the expected value of the overall reward of the system. Let v(s) denote the maximum expected value of the overall reward of the system when the system is initially in the system state *s*. Based on these notations, we can solve for the optimal policy for a given sequential decision-making problem by solving the following set of equations, hereafter called *optimality equations*:

$$v(s) = \max_{a \in A_s} \left\{ r(s,a) + \lambda \sum_{j \in S} p(j|s,a) v(j) \right\} \forall s \in S \tag{8.1}$$

where $\lambda \in (0,1)$ represents the discounting factor per each decision stage for the future rewards. If the optimality equations can be solved, the optimal policy for each system state *s* is

$$a^*(s) \in \arg\max_{a \in A_s} \left\{ r(s,a) + \lambda \sum_{j \in S} p(j|s,a) v(j) \right\} \tag{8.2}$$

Once all elements of MDP are identified and the optimality equations are constructed, we can apply the following algorithm called value iteration algorithm to find an ε-optimal policy and the approximated value of $v(s) \forall s \in S$.

8.3.1.1 Value Iteration Algorithm

Step 1: Select arbitrary real values for $v^0(s) \forall s \in S$, specify $\varepsilon > 0$, and set $n = 0$.

Step 2: For each $s \in S$, compute $v^{n+1}(s)$ by

$$\mathrm{v}^{n+1}(s) = \max_{a \in A_s} \left\{ r(s,a) + \lambda \sum_{j \in S} p(j|s,a) v^n(j) \right\} \tag{8.3}$$

Step 3: If $\|\vec{V}^{n+1} - \vec{V}^n\| < \varepsilon(1-\lambda)/2\lambda$, go to step 4. Otherwise increase the value of n by 1 and return to step 2. Note that V^n is a vector of size $|S|$ containing $\vec{v}^n(s) \forall s \in S$ as its elements.

Step 4: For each $s \in S$, choose

$$a^*(s) \in \arg\max_{a \in A_s} \left\{ r(s,a) + \lambda \sum_{j \in S} p(j|s,a) v^{n+1}(j) \right\} \forall s \in S \tag{8.4}$$

and stop.

After the algorithm terminates, the resulting values of $v^{n+1}(s)$ and $a^*(s)$ $\forall s \in S$ represent the optimal expected values of the overall reward and the ε-optimal policy of the considered problem, respectively.

Unfortunately, formulating and solving the real-world decision-making problems as a MDP is not an easy task. In many cases, obtaining the complete information on $r(s,a)$ and $p(\cdot \mid s,a)$ is a very difficult and time-consuming process. This process may involve a number of complex mathematical terms consisting of the joint probability distribution of many random variables. Furthermore, many unrealistic assumptions may have to be made in the process of obtaining this information. This phenomenon is hereafter called the *curse of modeling* of the MDP. If we can solve the sequential decision-making problems with the efficient methodology that does not require the exact close-form formulation of $r(s,a)$ and $p(\cdot \mid s,a)$, this methodology would be really attractive and would really be applicable to solve many complex real-world problems. In fact, RL is one of the methodologies that have the promising potential to perform this task. In the following subsection, we will discuss the RL technique and how to apply the technique to solve the complex sequential decision-making problems under uncertainty.

8.3.2 RL Technique

Because MDP is seriously cursed by the curse of modeling for some real-world decision-making problems, the methodology such as RL, which does not require the close-form formulations of rewards and transition probabilities, is of our interest in this subsection. It is worth noting that unlike the solution obtained from MDP, which is guaranteed to be optimal, the resulting solution obtained from the RL may only be just suboptimal. RL nicely combines the simulation technique with the solution methodology of MDP and normally produces a high quality solution to the problem.

The key idea of RL is to approximately solve the optimality equations, which may not be represented in the close-form formulations by utilizing the simulation models. Let us introduce the notation $Q(s,a) \forall s \in S$, $\forall a \in A_s$ such that

$$\begin{aligned} Q(s,a) &= r(s,a) + \lambda \sum_{j \in S} p(j \mid s,a) v(j) \quad \forall s \in S, \forall a \in A_s \\ &= \sum_{j \in S} p(j \mid s,a)(r(s,a,j) + \lambda\, v(j)) \quad \forall s \in S, \forall a \in A_s \end{aligned} \tag{8.5}$$

where $r(s,a,j)$ represents the immediate reward by making the action a in the system state s and the next system state is j. By using this notation of $Q(s,a)$, the optimality equations can be rewritten as

$$v(s) = \arg\max_{a \in A_s} \{Q(s,a)\} \forall s \in S \tag{8.6}$$

$$a^*(s) = \arg\max_{a \in A_s} \{Q(s,a)\} \forall s \in S \tag{8.7}$$

These equations imply that if we can calculate the value of $Q(s,a)$ $\forall s \in S$, $\forall a \in A_s$, we can easily obtain the value of $v(s)$ and $a^*(s)$ $\forall s \in S$, which are the decided solutions of the problem. We will now concentrate on the methodology for approximating the value of $Q(s,a)$ $\forall s \in S$, $\forall a \in A_s$. By using the definition of $Q(s,a)$, we can obtain the following equations:

$$Q(s,a) = \sum_{j \in S} p(j|s,a)\left(r(s,a,j) + \lambda \max_{b \in A_s} \{Q(j,b)\} \right) \quad \forall s \in S, \forall a \in A_s$$

$$= E\left(r(s,a,j) + \lambda \max_{b \in A_i} \{Q(j,b)\} \right) \quad \forall s \in S, \forall a \in A_s \tag{8.8}$$

As this equation indicates, calculating the value of $Q(s,a)$ $\forall s \in S$, $\forall a \in A_s$ involves the expectation operation, which can be obtained by using the simulation model and the result from the following Robbins-Monro algorithm. The Robbins-Monro algorithm is the algorithm developed in 1951 by Robbins and Monro [28] for estimating the population mean of a random variable from the sample. Let X denote the considered random variable and let xi denote the value of the ith independent sample of X. Let X^n denote the value of the sample average of x_i from $i = 1$ to n. From the strong law of large number, we can obtain the following relationship between $E(X)$, X^n, and x_i:

$$E(X) = \lim_{n \to \infty} \left(\sum_{i=1}^{n} \mathrm{x}_i / n \right)$$

$$= \lim_{n \to \infty} \left(X^n \right) \tag{8.9}$$

The Robbins-Monro algorithm utilizes the relationship between X^n and X^{n+1} and suggests the iterative procedure for calculating the value of $E(X)$. The relationship between X^n and X^{n+1} can easily be derived as follows where $\alpha^n = 1/n$:

$$X^{n+1} = \sum_{i=1}^{n+1} x_i / (n+1) \tag{8.10}$$

$$X^{n+1} = \left(\sum_{i=1}^{n} X_i + X_{n+1} \right) / (n+1)$$

$$X^{n+1} = \left(nX^n + X_{n+1} \right) / (n+1)$$

$$X^{n+1} = \left(n/(n+1) \right) X^n + (1/(n+1)) X_{n+1}$$

$$X^{n+1} = \left(1 - \alpha^{n+1} \right) X^n + (\alpha^{n+1}) X_{n+1} \tag{8.11}$$

By using this relationship, we can iteratively calculate the value of $X^1, X^2, \ldots, X^N$ after obtaining the sample information about the random variable X and can use the value of X^N as the approximation to $E(X)$ if N is a significantly large number. It is worth mentioning that the sample information of the random variable can be generated by using the simulation model and this is exactly the idea of RL. RL uses the basic idea of Robbins-Monro algorithm in calculating the expected value of the random variable to iteratively calculate the value of $Q(s,a)$ $\forall s \in S$, $\forall a \in A_s$ and finally obtain the values of $v(s)$ and $a^*(s)$ $\forall s \in S$. The algorithm iteratively calculates the value of $Q(s,a)\ \forall s\in S,\ \forall a\in A_s$ by generating a series of numbers $Q^1(s,a), Q^2(s,a),\ldots,Q^N(s,a)$ by utilizing the following relationship:

$$Q^{n+1}(s,a) = \left(1-\alpha^{n+1}\right)Q^n(s,a) + \left(\alpha^{n+1}\right)\left(r(s,a,j) + \lambda \max_{b\in A_j}\left\{Q^n(j,b)\right\}\right) \tag{8.12}$$

This calculation will be executed each time the action a is made in the system state s and the system evolves into the system state j. This relationship allows us to calculate the value of $Q(s,a)\ \forall s\in S,\ \forall a\in A_s$ without knowing the close-form formulation of rewards and transition probabilities because the value of $r(s,a,j)$ can be obtained from the simulation model. By utilizing this idea, the basic procedure of RL can be summarized as follows.

8.3.2.1 Basic RL Procedure for Discounted MDP

Step 1: Initialize the values of $Q(s,a) = 0$ and $N(s,a) = 0\ \forall s\in S,\ \forall a\in A_s$. Set $i = 0$ and N = maximum number of iterations (large integer number).

Step 2: Let s denote the current state of the system (from the simulation model). Randomly select an action from set A_s, each with equal probability. Let a denote the selected action.

Step 3: By selecting this action a in the system state s, the simulation model will be used to determine the next state of the system in the following decision stage. Let j denote this next system state. In addition, the simulation model will also be used to determine the value of $r(s,a,j)$

Set

$$N(s,a) \leftarrow N(s,a) + 1,\ i \leftarrow i + 1, \text{ and } \alpha = 1/N(s,a).$$

Step 4: Update the value of $Q(s,a)$ by using the following relationship.

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + (\alpha)\left(r(s,a,j) + \lambda \max_{b\in A_j}\left\{Q(j,b)\right\}\right)$$

Step 5: If $i<N$, update the current system state $s = j$ and return back to step 2. Otherwise proceed to step 6.

Step 6: Calculate and return the following values of $v(s)$ and $a^*(s)\ \forall s\in S$:

$$v(s) = \max_{a\in A_s}\left\{Q(s,a)\right\} \quad \forall s\in S \quad \textit{and} \quad a^*(s) = \arg\max_{a\in A_s}\left\{Q(s,a)\right\} \quad \forall s\in S$$

Figure 8.2 illustrates the general framework of this RL algorithm.

Note that more sophisticated methods of selecting the action can be implemented to improve the overall performance of the algorithm. In this subsection, we only present the basic idea of the algorithm that randomly selects an action for each iteration. In the following section, we apply the RL technique to the staff planning

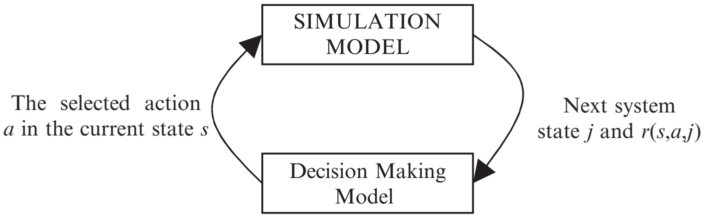

SIMULATION MODEL
Next system state j and $r(s,a,j)$
Decision Making Model
The selected action a in the current state s

Fig. 8.2 General framework of the reinforcement learning algorithm

problem of the airline's service call center. All results illustrate the very promising potential of the algorithm to solve this complex real-world problem.

8.4 Case Study on Airline's Cargo Service Call Center Planning

In today's business, many companies are aggressively racing to improve their customer service to increase their customer satisfaction in order to survive in the current highly competitive business environment. The Airline industry is no exception. Airline companies are constantly looking for new and innovative ways to keep their customers satisfied and to stay in the market. To do so, airline companies must ensure a high level of customer service 24 h a day, 7 days a week. This requires hard work and dedication from their employees at every level. Although employees do not lack any dedication, it is the correct staffing policy that poses a challenge for the managers at the airline's service call center. Efficient staff planning could make all the difference between success and failure in managing the customer service call center. Staffing managers are facing the challenge of deciding on the number of customer service agents required for each month to properly answer incoming customer calls in order to meet the certain service level with the minimum overall cost possible.

In this section, we apply the RL technique to the staff planning problems by using the real data obtained from one of the largest airline companies in USA. One of this airline's service centers is the Cargo Service Center (CSC). The CSC provides cargo booking and tracking services. The CSC handles 10 different types of customer calls that are divided as follows: (1) international (general service calls); (2) animal; (3) elite; (4) mortuary; (5) globalink; (6) SAS; (7) service recovery; (8) JFK; (9) Spanish; and (10) AMS.

In this chapter, we will concentrate our attention only on four major types of calls at the CSC, namely international, animal, elite, and mortuary, which comprise over 90% of the overall number of calls. The objective of this work is to decide on the number of agents required for each month at each of the four different types of customer calls. It is necessary to mention that both international and animal calls at CSC are currently handled by the same group of agents. This means that the data for both international and animal calls can be consolidated to create one set of data for this study. The airline company would like to set the service level for these four different types of customer calls as follows. For animal and international calls, 80% of all calls should be answered within 20 s of their arrival. For elite calls, 80% of all calls should be answered within 20 s of their arrivals. For mortuary calls, 70% of all calls should be answered within 20 s of their arrivals. To meet these service-level requirements, the number of agents on duty must be carefully decided and allocated. To gain better understanding of the system, multiple observational visits are made to the CSC. Observations included listening to the four different types of calls and observing their processes. The managers of the CSC are also of great help for us to understand the overall system. After acquiring enough information about the overall system and its processes, the system is then translated into a high-level flowchart, which is eventually transformed into the detailed simulation model. As a call enters the system, it will be classified as animal, international (GS), elite, or mortuary call. The call will then be answered immediately if there is at least one available agent at the time of its arrival, or else it will wait in the split specific queue. Each call split has its own queue and its own 1–800 number. The airline company has a policy that if a call arrives in its specific queue and there are seven calls already waiting in that queue, then the call will be rolled over to a different available queue. This is to keep customers' wait times at minimum and ensure that all agents are properly utilized, since some call splits have lower volumes than the others. The call will then wait in the next queue, given that it has less than seven calls waiting in it already, until the next agent becomes available and the call will be answered. Finally, once the call has been answered, it exits the system. If all queues are full, the incoming calls will not be answered. Figure 8.3 illustrates the

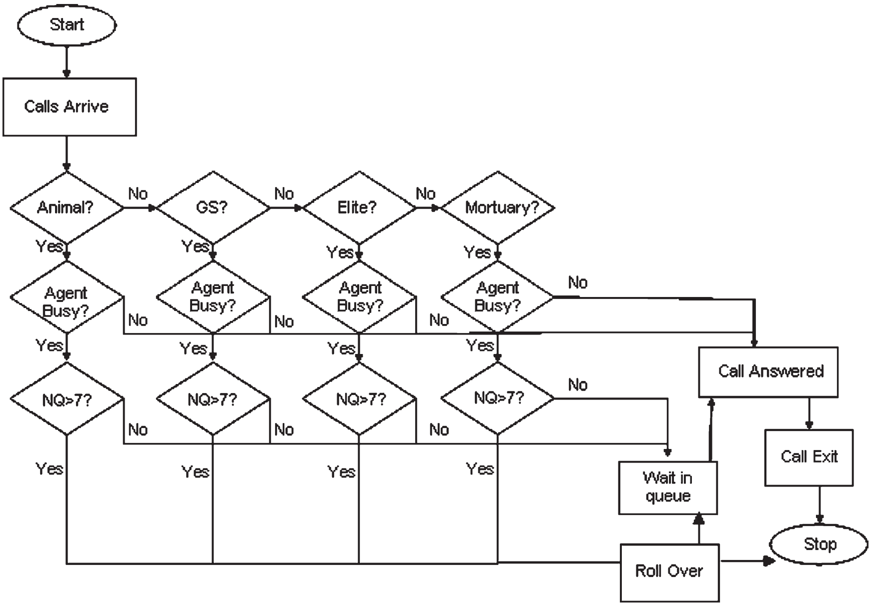


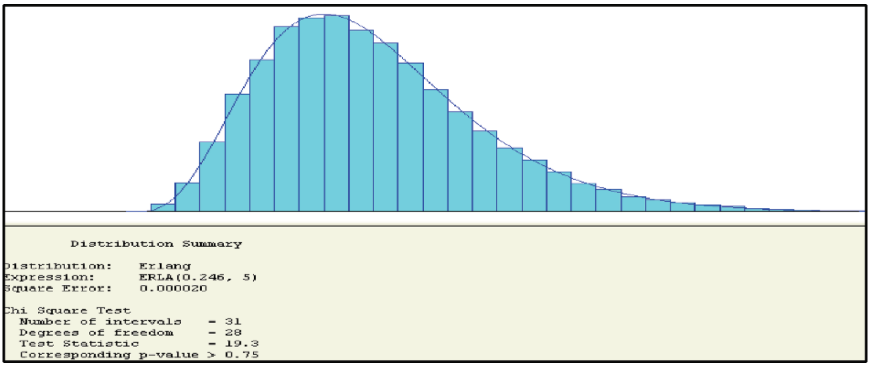
Fig. 8.3 Flowchart of customer calls routing at the CSC

flowchart of the customer call routing at the CSC where the notation NQ denotes the number of calls waiting in the queue.

8.4.1 Data Collection and Analysis for Constructing the Simulation Model

Accurate data analysis is the key fundamental in developing any simulation model. The performance of the simulation model can only be as good as the accuracy of the input data. With that in mind, the data collection and analysis is one of the most important tasks of this research. For this work, real data for an entire year are used to construct the simulation model of the service call center. The data in this study are obtained by using the historical information from the airline company. The airline company records these data for the different call splits and stores them in the company database. Note that the data used for this research are the year 2005 data. The data used in constructing the simulation model include (1) the interarrival time for each type of calls on each day of each month for the entire year and (2) the service time of each type of calls on each day of each month for the entire year.

Once the data had been collected and analyzed, appropriate probability distributions of these parameters are determined by utilizing the ARENA 10.0 input analyzer [29]. Input analyzer is a statistical analysis program included in the simulation software package called ARENA. This program takes a set of raw data as its input and generates a list of probability distributions that best fits the data. Figure 8.4 illustrates an example output of ARENA input analyzer. Once all required probability distributions of the model parameters are obtained, the detailed simulation model of the entire system is then developed by utilizing the simulation software package ARENA 10.0.



```
        Distribution Summary

Distribution:   Erlang
Expression:     ERLA(0.246, 5)
Square Error:   0.000020

Chi Square Test
  Number of intervals   = 31
  Degrees of freedom    = 28
  Test Statistic        = 19.3
  Corresponding p-value > 0.75
```

Fig. 8.4 An example output of ARENA input analyzer

8.4.2 RL Model for the Service Call Center Problems

After the simulation model of the service call center has been developed, some components of MDP have to be determined in order to implement the RL technique. These components are (1) the state space (S); (2) the action space for each possible system state s (A_s); (3) the reward structure; and (4) the decision stage (T). In this problem, the state information consists of the number of calls from the previous month and the current calendar month. For example, in the month of May, one of the possible states is s = (12,000 calls, May) if the number of calls in April was 12,000. After the system state information is observed, the possible action is basically the number of agents available to work in the current month. The reward structure of this problem is the numerical quantity that indicates how well a certain policy performs under certain circumstances. Deciding on the structure of the reward is somewhat challenging when modeling a service call center. The reward has to be measured in terms of the number of answered calls, the number of dropped calls, the number of calls with long queue waiting time, the hiring and firing costs, and the number of agents working at the service call center. In this model, the following formulation is used to calculate the reward value of making a certain action in a particular state.

Reward = [(profit per call) × (number of answered calls)] - [(monthly salary per agents) × (number of agents)] − [(penalty) × (number of calls that do not meet the required service level)] − [(Hiring cost per agent) × (the number of new agents)] - [(Firing cost per agent × the number of agents fired)]

This reward value can easily be obtained from the simulation model. Finally, the decision stage is the time period between each pair of the decision-making processes. In this work, the decision stage is the beginning of each month when the decision maker is required to decide on the number of working agents for each type of calls. Once all these components are identified, the RL technique is then applied to solve the considered decision problem. The simulation and decision-making models of the RL are executed on a Windows XP-based Pentium(R) 4 CPU 3.60 GHz personal computer with 4.00 GB RAM using Arena 10.0 and Visual Basic for Application (VBA) programming language. MS-Excel is used for the case study input and output database. Table 8.1 summarizes the recommended staffing policy for international or animal type of calls.

Table 8.1 Recommended staffing policy for the international or animal call split

Month (s)	No. of last month calls	No. of agents
February	Any value	31 agents
January, March, April, June, September	Any value	33 agents
May, July, August	Any value	35 agents
October, November, December	Any value	37 agents

Table 8.2 Recommended staffing policy for the elite call split

Month (s)	No. of last month calls	No. of agents
February, March, April, June, October, November, December	Any value	7 agents
January, August	Any value	8 agents
May, September	<4,700 calls	7 agents
May, September	≥4,700 calls	8 agents
July	<5,100 calls	8 agents
July	≥5,100 calls	9 agents

Table 8.3 Recommended staffing policy for the mortuary call split

Month (s)	No. of last month calls	No. of agents
April, May	All value	4 agents
June, August, September, October	All value	5 agents
January	<2,700	4 agents
January	≥2,700	5 agents
February	<2,400	4 agents
February	≥2,400	5 agents
March	<1,900	4 agents
March	≥1,900	5 agents
July	<2,250	4 agents
July	≥2,250	5 agents
November	<2,200	5 agents
November	≥2,200	6 agents
December	<2,450	5 agents
December	≥2,450	6 agents

Tables 8.2 and 8.3 illustrate the recommended staffing policies for the elite call split and the mortuary call split, respectively.

The current staffing policy at the CSC is to use 34.5, 7, and 8 full-time equivalents (FTEs) working on answering the international or animal, elite, and mortuary types of calls, respectively. An FTE consists of either a full-time employee or two part time employees. In the following subsection, we will compare the performance of these recommended solutions with the performance of the current policy used by the airline company. All results illustrate the improvements in the system performances resulting form the recommended solutions over the current policy.

8.4.3 Case Study Result and Performance Comparison

In this subsection, our goal is to statistically compare the performances of the policies recommended by the RL model and the performances of the current policy utilized by the airline company (original). To do so, another simulation model is developed to read a specific staffing policy as the input. This simulation model will then evaluate the input policy and will calculate a number of important performance measures of the system as the output. These results of each policy are analyzed and statistically compared. In this research, the following characteristics are used to measure the performance of the service call center: (1) the average number of calls that do not meet the required service level; (2) the average number of calls that are dropped; (3) the average utilization of agents; (4) the average waiting time in queue of each call; and (5) the overall cost per month of the system.

Based on the results obtained from 100 simulation years run, the values of these characteristics are calculated and recorded for each policy. After obtaining the values of these characteristics, statistical hypothesis testing procedures are performed in order to analyze and compare the performances of these two policies. These statistical hypotheses are summarized in the Tables 8.4 and 8.5. The mean values of these characteristics generated by the simulation model are compared between these two policies by utilizing the standard *t*-test. Note that the *t*-test is very robust for testing these hypotheses even if the data are not normally distributed when the sample sizes are large, which is the case for the examined data sets in this research.

If the null hypothesis contained in Table 8.4 is rejected for a specific performance measure, then we can conclude that the RL solution performs better in that characteristic. If the null hypothesis contained in Table 8.5 is rejected for a specific performance measure, we can conclude that the solution generated by the current plan performs better in that characteristic. If we fail to reject the hypotheses in both Tables 8.4 and 8.5 for a specific performance measure, we can conclude that there is no statistical difference between the two policies in that characteristic. Before applying the *t*-test to test these hypotheses, *F*-test is first used to check for the equality of variances between the two data sets: The null hypothesis (HO) of the F-test states that the variances of these two data sets are equal, while the alternative hypothesis (Ha) of the F-test states that the variances of these two data sets are different. The results from the *F*-test will determine the type of *t*-test to be used. The detailed information about statistical hypothesis testing with the *t*-test and the *F*-test can be studied in the book written by Johnson [*30*].

Table 8.4 The first set of statistical hypotheses for performance comparison

Characteristic	H_0	H_a
Mean number of bad calls	Means are equal	Original > RL
Mean number of dropped calls	Means are equal	Original > RL
Mean utilization of agents	Means are equal	RL > Original
Mean queue time	Means are equal	Original > RL
Mean monthly cost	Means are equal	Original > RL

Table 8.5 The second set of statistical hypotheses for performance comparison

Characteristic	H_0	H_a
Mean number of bad calls	Means are equal	Original < RL
Mean number of dropped calls	Means are equal	Original < RL
Mean utilization of agents	Means are equal	RL < Original
Mean queue time	Means are equal	Original < RL
Mean monthly cost	Means are equal	Original < RL

Table 8.6 Summary of the performance comparison for animal or GS call type

	Animal or GS type calls				
	No. of bad calls	Average queue time	No. of dropped calls	Average utilization	Average monthly cost
January	Δ	Δ	Δ	Δ	O
February	X	X	Δ	O	O
March	Δ	Δ	Δ	O	O
April	Δ	Δ	Δ	O	O
May	Δ	Δ	O	X	Δ
June	Δ	Δ	Δ	O	O
July	Δ	Δ	O	X	Δ
August	Δ	Δ	O	X	Δ
September	Δ	Δ	Δ	Δ	O
October	O	O	O	X	Δ
November	O	O	O	X	Δ
December	O	O	O	X	O
Overall	Δ	Δ	O	Δ	O

Table 8.7 Summary of the performance comparison for elite call type

	Elite type calls				
	No. of bad calls	Average queue time	No. of dropped calls	Average utilization	Average monthly cost
January	O	O	O	X	O
February	Δ	Δ	Δ	Δ	Δ
March	Δ	Δ	Δ	Δ	Δ
April	Δ	Δ	Δ	Δ	Δ
May	Δ	Δ	Δ	X	Δ
June	Δ	Δ	Δ	Δ	Δ
July	O	O	O	X	O
August	O	O	O	X	O
September	Δ	Δ	Δ	Δ	Δ
October	Δ	Δ	Δ	Δ	Δ
November	Δ	Δ	Δ	Δ	Δ
December	Δ	Δ	Δ	Δ	Δ
Overall	O	O	Δ	Δ	Δ

After performing the hypothesis testing procedures with the value of the type I error probability of 0.05, the results are obtained and summarized for each call type (animal or GS, elite, and mortuary). Tables 8.6–8.8 contain the summary information on the test results for animal or GS, elite, and mortuary call types, respectively, for each month and for the overall year. The following notations are used in these tables for ease in interpreting these results.

X: This notation indicates that the mean of the RL model was statistically significantly worse than the mean of the original model.

O: This notation indicates that the mean of the RL model was statistically significantly better than the mean of the original model.

Table 8.8 Summary of the performance comparison for mortuary call type

	Mortuary type calls				
	No. of bad calls	Average queue time	No. of dropped calls	Average utilization	Average monthly cost
January	O	D	X	O	D
February	Δ	Δ	Δ	O	Δ
March	O	Δ	Δ	O	Δ
April	Δ	Δ	Δ	O	Δ
May	O	Δ	Δ	O	Δ
June	Δ	Δ	Δ	Δ	Δ
July	O	Δ	X	O	Δ
August	Δ	Δ	Δ	Δ	Δ
September	Δ	Δ	Δ	Δ	Δ
October	Δ	Δ	Δ	Δ	Δ
November	O	O	O	X	Δ
December	O	O	O	X	Δ
Overall	O	O	Δ	O	Δ

Δ : This notation indicates that there is no statistically significant difference between the mean of the RL model and the mean of the original model.

Keeping in mind the results from the overall performance comparisons, we can come to the following conclusions. For the animal or GS call type, the staffing policy generated by the RL technique statistically outperforms the current staffing policy in the average number of dropped calls and the average monthly cost criteria. There are no statistically significant differences between the performances of these two policies for other criteria. For the elite call type, the staffing policy generated by the RL technique statistically outperforms the current staffing policy in the average number of bad calls and the average waiting time in queue criteria. There are no statistically significant differences between the performances of these two policies for other criteria. For the mortuary call type, the staffing policy generated by the RL technique statistically outperforms the current staffing policy in the average number of bad calls, the average waiting time in queue, and the average agent utilization criteria. There are no statistically significant differences between the performances of these two policies for other criteria.

8.5 Summary

Simulation and optimization are clearly two of the most powerful fields in the study of operation research and management science. Combining these two techniques together is definitely a promising concept for solving the real-world complex decision-making problems. In this chapter, the basic concepts of the simulation-based optimization technique, namely the RL, are explained and discussed in detail. We then apply the RL technique to determine the staffing policy for the airline service call center. Statistical hypothesis testing procedures are used to perform the performance comparisons between the recommended policy and the current policy. All results illustrate that the policy generated by the RL is superior to the current policy in a number of performance measures. This illustrates the promising potential of the simulation-based optimization techniques in generating the high quality solution for the complex decision-making problems in practice.

References

1. Gosavi, A., Simulation-Based Optimization: Parametric Optimization Techniques & Reinforcement Learning, Berlin Heidelberg New York, Springer, 2003.
2. Kleinman, N.L., Hill, S.D., and Ilenda, V.A., Simulation optimization of air traffic delay cost, Proceedings of the Winter Simulation Conference, Washington, DC, 1177–1181, 1998.
3. Rosenberger, J.M., Schafer, A.J., Goldsman, D., Johnson, E.L., Kleywegt, A.J., and Nemhauser, G.L., SIMAIR: A stochastic model of airline operations, Proceedings of the Winter Simulation Conference, 2000.
4. Lee, L.H., Hunag, H.C., Lee, C., Chew, E.P., Jaruphongsa, W., Yong, Y.Y., Liang, Z., Leong, C.H., Tan, Y.P., Namburi, K., Johnson, E., and Banks, J., Discrete event simulation model for airline operations: SIMAIR, Proceedings of the Winter Simulation Conference, 2003.
5. Sutton, R.S. and Barto, A.G., Reinforcement learning: An introduction, Cambridge, MA, MIT Press, 1998.
6. Bertsekas, D.P. and Tsitsiklis, J.N., Neuro-Dynamic Programming, Optimization and Neural Computer Series, 3, Belmont, MA, Athena Scientific, 1996.
7. Li Littman, M.L., Moore, A.W., and Kaelbling, L.P., Reinforcement Learning: A Survey, Brown University/Carnegie Mellon University, Providence, RI/Pittsburgh, PA.
8. Mahadevan, S., Average reward reinforcement learning: foundations, algorithms, and empirical results, Machine Learning, 22 (1), 159–195, 1996.
9. Creighton, D.C. and Nahavandi, S., Optimizing discrete event simulation models using a reinforcement learning agent, Winter Simulation Conference, Vol. 2, pp. 1945–1950, 2002.
10. Aydin, M.E., and Öztemel, E., Dynamic job shop scheduling using reinforcement learning agents, Robotics and Autonomous Systems, 33, 169–178, 2000.
11. Jeong, K., Conceptual frame for development of optimized simulation-based scheduling systems, Expert Systems with Applications, 18, 299–306, 2000.
12. Zhang, W. and Dietterich, T.G., A reinforcement learning approach to job-shop scheduling, Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, 1114–1120, 1995.
13. Reidmiller, S. and Reidmiller, M., A neural reinforcement learning approach to learn local dispatching policies in production scheduling, Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, 1999.
14. Schneider, J.G., Boyan, J.A., and Moore, A.W., Value function based production scheduling, Proceedings of the Fifteenth International Conference on Machine Learning, 24–27 July 1998.
15. Paternina-Arboleda, C.D. and Das, T.K., Intelligent dynamic control policies for serial production lines, IIE Transactions, 33, 65–77, 2001.
16. Mahadevan, S., Marchalleck, N., Das, T.K., and Gosavi A., Self-improving factory simulation using continuous-time average-reward reinforcement learning, Proceedings of the Fourteenth International Conference on Machine Learning, 202–210, 1997.
17. Mahadevan, S. and Theochaurus, G., Optimizing production manufacturing using reinforcement learning, Proceedings of the Eleventh International FLAIRS Conference, 372–377, 1998.

18. Sze, D.Y., A queuing model for telephone operator staffing, Operation Research, 32, 229–249, 1993.
19. Andrews, B. and Parsons, H., Establishing telephone-agent staffing levels through economic optimization, Interfaces, 23, 14–20, 1993.
20. Chen, B.P.K. and Henderson, S.G., Two issues in setting call centre staffing levels, Annals of Operations Research, 108, 175–192, 2001.
21. Fukunaga, A., Hamilton, E., Fama, J., Andre, D., Matan, O., and Nourbakhsh, I., Staff Scheduling for Inbound Call Centers and Customer Contact Centers, Blue Pumpkin Software, Sunnyvale, California, 2002.
22. Borst, S., Mandelbaum, A., and Reiman, M.I., Dimensioning large call centers, Operation Research, 52, 17–34, 2004.
23. Atlason, J., Epelman, M.A., and Henderson, S.G., Call center staffing with simulation and cutting plane methods, Annals of Operations Research, 127, 333–358, 2004.
24. Atlason, J., Epelman, M.A., and Henderson, S.G., Optimizing call center staffing using simulation and analytic center cutting plane methods, Department of Industrial and Operations Engineering, University of Michigan/School of Operations Research and Industrial Engineering, Cornell University, Ann Arbor, MI/Ithaca, NY, 2005.
25. Deslauriers, A., Ecuyer, P.L., Pichitlamken, J., Ingolfsson, A., and Avramidis, A.N., Markov chain models of a telephone call center with call blending, Computer and Operations Research, 34, 1616–1645, 2005.
26. Mourtada, I., Reinforcement Learning Model for Continental Airline’s Service Center, M.S. Thesis, Department of Industrial Engineering, the University of Houston, May 2007.
27. Puterman, M., Markov Decision Processes: Discrete Stochastic Dynamic Programming, Wiley Series in Probability and Mathematical Statistics, New York, Wiley, 1994.
28. Robbins, H. and Monro, S., A stochastic approximation method, Annals of Mathematical Statistics, 22, 400–407, 1951.
29. Kelton, W.D., Sadowski, R.P., and Sturrock, D.T., Simulation with Arena, 4th edition, New York, NY, Mc Graw Hill, 2007.
30. Johnson, R.A., Miller & Freund’s Probability and Statistics for Engineers, 7th edition, Upper Saddle River, NJ, Pearson Prentice Hall, 2005.