The Pennsylvania State University

The Graduate School

The Mary Jean and Frank P. Smeal College of Business

MANAGING SERVICE CAPACITY UNDER UNCERTAINTY

A Thesis in
Business Administration and Operations Research
by
Thomas R. Robbins

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2007

The thesis of Thomas R. Robbins was reviewed and approved* by the following:

Terry P. Harrison
Professor of Supply Chain and Information Systems
Thesis Adviser
Chair of Committee

Susan Xu
Professor of Management Science and Supply Chain Management

Douglas Thomas
Associate Professor of Supply Chain Management

Tom Cavalier
Professor of Industrial Engineering

John E. (Gene) Tyworth
Chair, Department of Supply Chain & Information Systems

*Signatures are on file in the Graduate School.

Abstract

This dissertation addresses the issue of capacity management in a professional services context; specifically a call center based support operation with contractually committed Service Level Agreements (SLAs). The focus of this research is on capacity planning in the face of uncertainty. I investigate the impact of uncertainty on the capacity management decision and develop models that explicitly incorporate uncertainty in the planning process. A short term scheduling model develops detailed staffing plans given variable and uncertain demand patterns. A medium term hiring model seeks the optimal hiring level for the start up of a new project with learning curve effects. A cross training model seeks to determine the best number of agents to cross train on multiple projects. The analysis employs stochastic programming, discrete event simulation, and a simulation based optimization heuristic.

This dissertation is very much an applied OR analysis. The research focuses not on developing new theory or methodology, but on applying existing methods to a real problem. In the process I create several new and unique models that contribute to the literature. The research is motivated by work I performed with an IT Support outsourcing company. That company was kind enough to give me access to a great deal of data upon which to base my analysis.

I find that incorporating uncertainty into the planning process yields solutions with better outcomes and also provides better insight into key management tradeoffs. The short term scheduling model shows that hedging against arrival rate uncertainty lowers the total cost of operation by improving the probability of SLA attainment. It also shows that increasing the flexibility of the staffing model, by scheduling even a few part time resources, can significantly lower costs. I also find that increasing the probability of achieving the service level goal becomes increasingly expensive. The medium term hiring model shows that learning curve issues during start-up have a significant impact on total costs. The cross training model shows that adding even a moderate amount of flexibility into the workforce can significantly lower costs through the dynamic reallocation of capacity.

Table of Contents

List of Figures

List of Tables

1 Introduction

The professional services industry is a rapidly growing segment of both developed and developing economies. For example, U.S. based multinationals such as IBM, HP and GE are increasingly reliant on professional services. In 2004, IBM Global Services revenue grew 8 percent to $46.2 billion, 48% of the company's total (IBM 2004). Also in 2004, the consulting firm Accenture grew revenue by 15% to 13.6 billion, and increased headcount to over 100,000 people (Accenture 2004). The global services company Capgemini lost 20% of its staff in 2004 through attrition and layoffs yet grew its total headcount by 6.7% through new hires and outsourcing transactions. In the same year that the firm hired over 9,000 people, they terminated over 2,300. Developing nations such as India and China are also rapidly growing IT and Business Process Outsourcing operations. In 2004 India based Infosys lost over 2,700 employees to attrition, yet grew its headcount 37%, hiring 11,597 new employees from an applicant base of over 13 million. (Infosys 2005)

For our purposes professional services includes business such as traditional management consulting, outsourced design, technical support, call center operations, IT implementation, and IT outsourcing (Dietrich and Harrison 2006). The detailed analyses in this dissertation are based on a Business Process Outsourcing/Call Center operation, but they are easily extended to any service operation where load is uncertain and service levels are important.

While many management practices from traditional industrial businesses are applicable, the professional services sector presents some unique challenges. Unlike traditional manufacturing operations, the delivery capacity of a professional services organization is largely dependent on the quantity and skills of human resources available to the organization, either through direct employment, subcontracting, or through partnerships with other professional service firms. The heavy reliance on human resources as the primary determinant of productive capacity implies that a key challenge for managers is the strategic and tactical planning of acquisition, training, and termination of resources (Dietrich and Harrison 2006). The literature generally refers to this as the Manpower Planning Problem.

Manpower planning models have been analyzed in the operations literature since the emergence of the field. Dantzig first formulated the staff scheduling problem as a linear program in 1954 (Dantzig 1954). Holt *et al.* published their text on aggregate planning models in 1960 (Holt, Modigliani *et al.* 1960) and Bartholomew and Forbes published a text devoted to the application of stochastic modeling to manpower planning models in the 1970s (Bartholomew and Forbes 1979). However, much of this work has evaluated manpower planning in the context of a manufacturing enterprise and much of the remaining research fails to address issues critical to large scale professional service operations. Subtle yet important differences in the professional service environment call for modifications to these models.

Manpower planning problem can be categorized using the following framework.

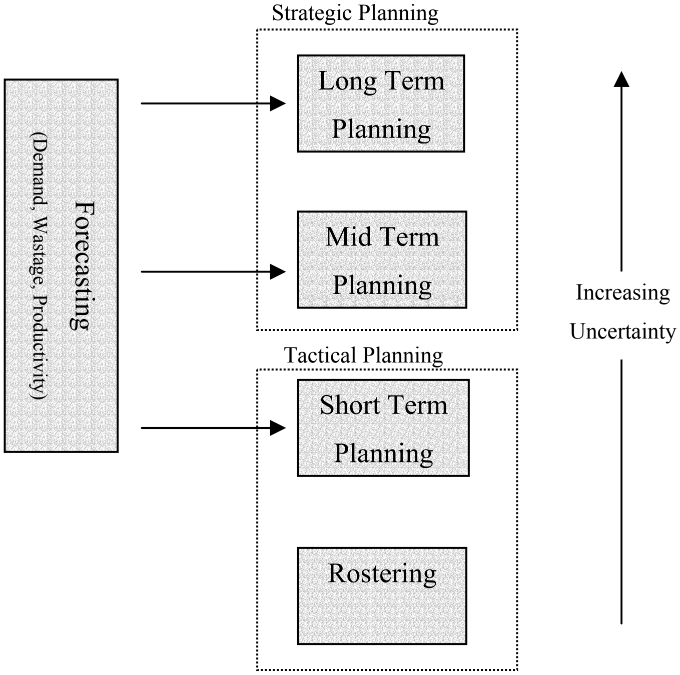


Figure 1-1 Manpower Planning Framework

In tactical planning the workforce is fixed and the challenge is to efficiently assign resources to satisfy demand. Strategic manpower planning on the other hand addresses workforce management over a longer term planning horizon where the size and skills of the workforce is variable. Over the strategic horizon an organization must plan for recruiting, hiring, and training of new resources to facilitate growth and compensate for unplanned turnover. In the long term horizon individuals progress through different skill or grade level changing their productivity as well as their cost. In the strategic timeframe organizations may also need to separate workers either to lower costs or to *reskill* the organization.

In this dissertation I develop three related models primarily focused on short and mid term planning:

- **Short Term Scheduling Model**: The short term model addresses short term (weekly) scheduling of resources. I address this problem in the context of a call center operation, building a model that generates a shift schedule that explicitly accounts for uncertainty in arrival rates. This analysis shows that explicitly recognizing uncertainty in the load leads to verifiably superior solutions; that is solutions with a lower expected cost of operation.
- **Medium Term Hiring Model**: The medium term model is designed to address a 3-4 month planning horizon. The model identifies hiring levels for a new outsourcing project given uncertain demand and learning curve effects. The model characterizes the optimal level of spare capacity to hire under various conditions.
- **Cross Training Model**: This model analyzes the impact of cross training in BPO/call center operations with uncertain arrival rates and attempts to find the optimal level of project cross training. The analysis shows that in general a small level of cross-training provides substantial benefit. In this model I develop a heuristic algorithm to find near-optimal cross training levels given nonstationary and uncertain demand.

In Section 2 of this dissertation I discuss detailed work with a provider of outsourced technical support services. My work with this company provides the context and motivation for the models in this dissertation. In Section 3 I review the relevant literature. Sections 4-6 present the three models included in this dissertation. Section 7 provides general conclusions.

2 Industry Context

2.1 Overview

My concept of professional services is an organization providing business-to-business services where the services are knowledge intensive and are delivered by highly skilled resources. Common examples of professional services include management consulting, IT outsourcing, legal services, or investment banking. The models in this dissertation will be focused on a BPO/Call center operation. This research is motivated in part by the author's previous work experience in management consulting, as well as two recent projects conducted during the 2nd half of 2006. The projects address capacity management in two professional service organizations. The first project examined short-medium term capacity management issues within IBM's Business Consulting Services operation. The second involved an analysis of tactical and strategic capacity management at a provider of outsourced customer support. While the models are easily extended to general service applications, in this dissertation I will focus on call center operations. This decision is based largely on the availability of detailed data and a specific problem context.

2.2 Call Center Operations

Call centers are a critical component of the worldwide services infrastructure and are often tightly linked with other large scale services. Many outsourcing arrangements, for example, contain some level of call center support, often delivered from offshore locations. A call center is a facility designed to support the delivery of some interactive service via telephone communications; typically an office space with multiple workstations manned by *agents* who place and receive calls (Gans, Koole *et al.* 2003). Call centers are a large and growing component of the U.S. and world economy (Gans, Koole *et al.* 2003). In 1999 an estimated 1.5 million workers were employed in call centers in the US alone[1] and call center operations represent a major portion of the offshore BPO market. Large scale call centers are technically and managerially sophisticated operations and have been the subject of substantial academic research. Call center applications include telemarketing, customer service, help desk support, and emergency dispatch.

[1] This figure is somewhat dated but it is the most recent figure I could find in the academic literature.

Even a moderately sophisticated call center is equipped with advanced computer and telecommunications equipment. An inbound call typically connects from the public service telephone network (PSTN) to the call center's switch, the private branch exchange (PBX), over a number of owned or leased trunk lines. Callers may initially connect to an Interactive Voice Response unit (IVR) where the caller can use her keypad to select options and potentially provide data input to call center system. When callers need to speak to an agent, the call is handled by the Automated Call Distributor (ACD). The ACD routes calls internal to the call center and is responsible for monitoring agent status, collecting data, managing on hold queues, and making potentially complex routing decisions. For example, in call centers that employ *skills based routing*, a complex decision process is used to match callers and agents based on multiple criteria concerning both the callers and the agents. In centers that perform outbound calling a Predictive Dialer is often used to perform anticipatory dialing. In addition to the telephone system, a call center agent usually has a computer terminal connected into one or more enterprise applications; these are typically classified under the general category of Customer Relationship Management (CRM).

A general architecture is depicted in the figure below:

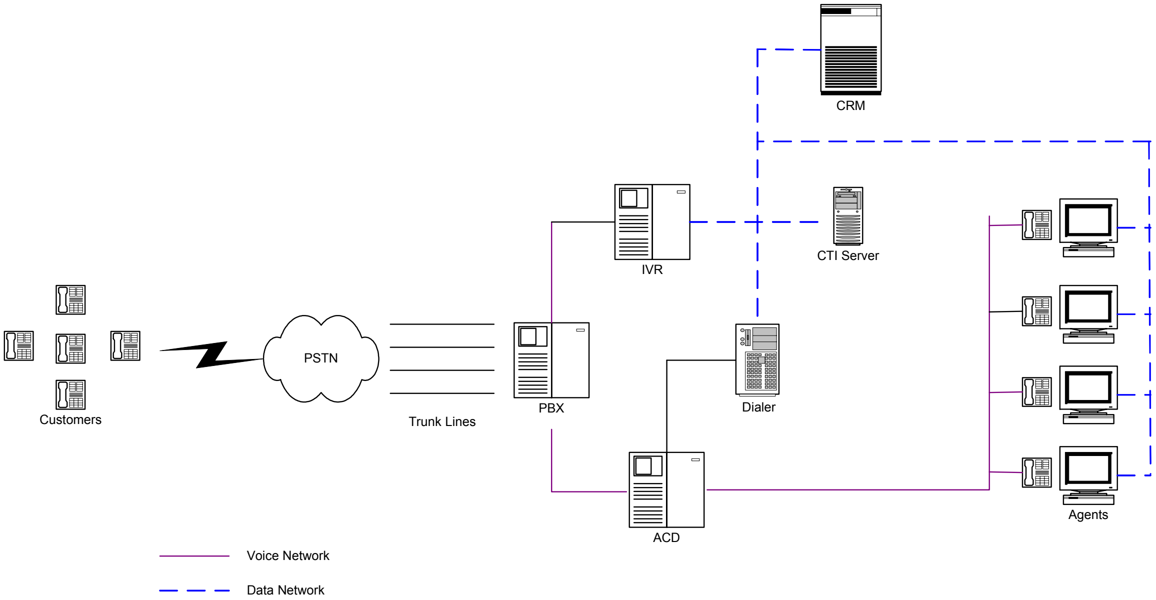


Figure 2-1 Prototypical Call Center Technology Architecture

From a queuing perspective a general model of a call center is the Erlang A model, depicted schematically in the following figure.

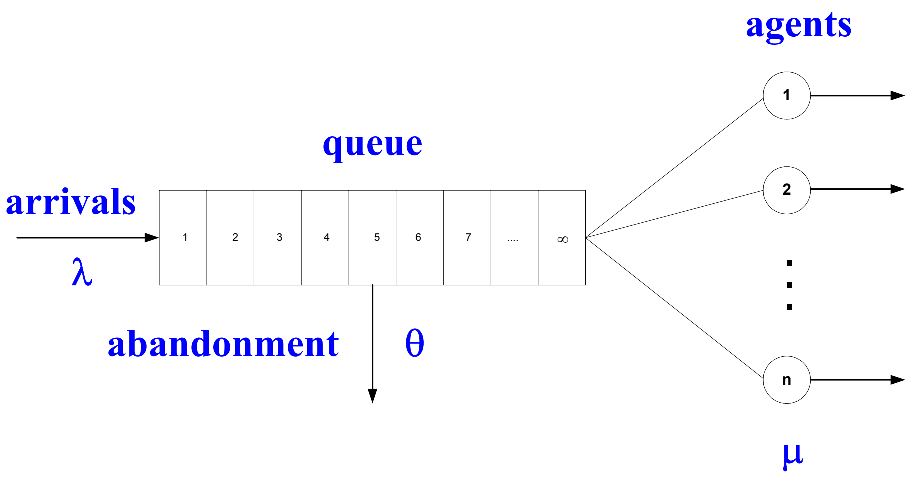


Figure 2-2 Call Center Queuing Architecture

In the Erlang A model calls arrive at a rate λ, with interarrival times that are independent and exponentially distributed. If no agents are available calls are placed into an infinite capacity queue to await service on a First Come First Serve (FCFS) basis. Calls have an exponentially distributed talk time with mean $1/\mu$. Associated with each call is a *patience time*; if callers are forced to wait longer then their patience time, they abandon the queue (hang up.) Patience time is exponentially distributed and average time to abandon is $1/\theta$ and the corresponding individual abandonment rate is θ (Mandelbaum and Zeltyn 2004).

Erlang A is more complicated and less widely applied then the Erlang C model[2]. However, as I will show, abandonment is an important consideration in this environment and the use of the Erlang A model is warranted.

[2] Erlang C is identical to the M/M/n queue. The Erlang C name is widely used in call center applications.

2.3 Subject Firm

My research in this area is motivated in part by my recent work with a medium sized (approximately $160 million/year public company) provider of technical support. The company provides both tier 1 (help desk) and tier 2 (deskside[3]) support. The bulk of their business, and the focus or my research, is on the inbound call center operation. This operation involves providing help desk support to large corporate and government entities[4]. While the scope of services varies from account to account, many accounts are 24 x 7 support and virtually all accounts are subject to some form of Service Level Agreement (SLA). There are multiple types of SLAs, but the most common specifies a minimum level of the Telephone Service Factor (TSF). A TSF SLA specifies the proportion of calls that must be answered within a specified time. For example, an 80/120 SLA specifies that 80% of calls must be answered within 120 seconds. A very important point is that the service level applies to an extended period, typically a month. The SLA does not define requirements for a day or an hour. The desk is typically staffed so that at some time the service level is underachieved, sometimes overachieved, and is on target for the entire month.

2.4 Arrival Rate Uncertainty

2.4.1 Overview

The key challenge involved with staffing this call center is a fixed SLA with a variable and uncertain arrival rate pattern. Inbound call volume is highly variable with multiple sources of uncertainty. In the following analysis I consider variability at multiple levels of aggregation; weekly, daily, and by half hour period.

2.4.2 Weekly Arrivals

In the short to medium term the major seasonality pattern occurs at the weekly level. For the purpose of this analysis we ignore unusually slow periods, such as the week between Christmas and New Years, and examine the variability of call volume at the weekly level. The following chart summarizes four months of call volume data for 11 US based projects. I list the average

[3] Deskside support involves physically dispatching technicians to the customer's location to perform detailed troubleshooting or configuration.

[4] The company provides national or global support to multiple Fortune 100 companies. They have multiple call centers in North America, Western Europe, and Eastern Europe.

weekly inbound call volume, the standard deviation of call volume, and the corresponding coefficient of variation.

	Average		
	Weekly	Std. Dev.	Coefficient of
	Volume	of Volume	Variation
Project 1	248.1	81.7	0.330
Project 2	291.9	92.7	0.318
Project 3	516.7	283.2	0.548
Project 4	560.7	175.0	0.312
Project 5	1,442.9	460.2	0.319
Project 6	1,545.0	504.1	0.326
Project 7	2,599.3	809.5	0.311
Project 8	3,336.9	986.7	0.296
Project 9	4,386.8	1,664.4	0.379
Project 10	7,566.9	3,493.9	0.462
Project 11	8,221.9	1,586.9	0.193

Table 2-1 Variability of Weekly Call Volume

This table shows that volume varies considerably from week to week. It also shows that the degree of variability varies considerably from project to project, with coefficients of variation as low as .193 and as high as .548.

2.4.3 Daily Arrivals

Call volumes exhibit a strong seasonality pattern over the course of a week. In the following figure we see daily call volume for a typical project shown over a 3 month period.

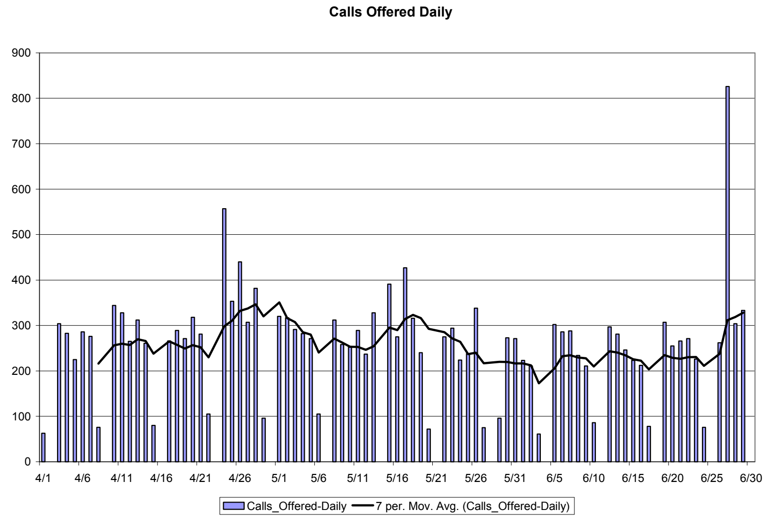
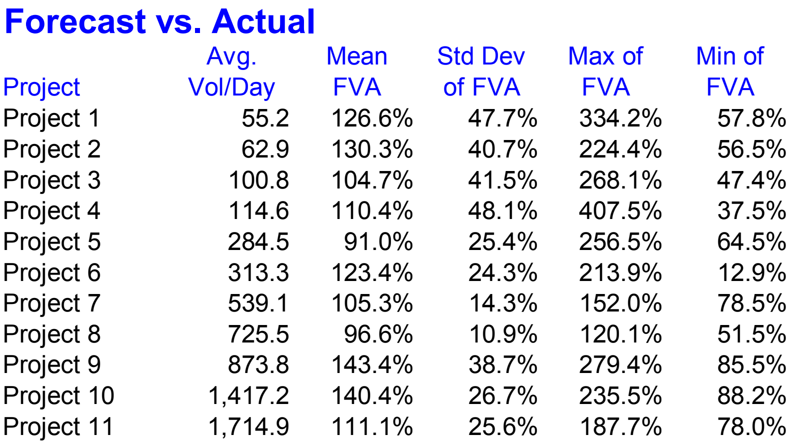


Figure 2-3 Sample Daily Call Volume

This graph shows strong "seasonal" variation over the course of a week. Monday's tend to be the highest volume days with volumes dropping off over the course of the week. Volume on Saturdays is a small fraction of the weekday volume and this particular desk is closed on Sunday. The graph also reveals significant stochastic variability. Tuesdays are, for example, often higher volume then Wednesdays but this is not always the case. During the weeks of 4/26 and 5/16 we see larger volumes on Wednesday then Tuesday. We also see the issue of unanticipated spikes in demand, often referred to as *significant events*. This is an extremely common event in support desk operations. A downed server, for example, will generate a large call volume. While some contracts provide SLA relief in the case of significant events, in general the desk must meet SLA even when significant events occur. The large volume of calls during a significant event not only result in poor performance, but also create a large proportion of the total calls making it more difficult to achieve a specific percentage of "within SLA" calls.

The following chart summarizes the problem of daily volume variability. The average (M-F) daily call volume for each project is listed along with summary statistics for the daily Forecast vs. Actual measure[5].



Forecast vs. Actual

Project	Avg. Vol/Day	Mean FVA	Std Dev of FVA	Max of FVA	Min of FVA
Project 1	55.2	126.6%	47.7%	334.2%	57.8%
Project 2	62.9	130.3%	40.7%	224.4%	56.5%
Project 3	100.8	104.7%	41.5%	268.1%	47.4%
Project 4	114.6	110.4%	48.1%	407.5%	37.5%
Project 5	284.5	91.0%	25.4%	256.5%	64.5%
Project 6	313.3	123.4%	24.3%	213.9%	12.9%
Project 7	539.1	105.3%	14.3%	152.0%	78.5%
Project 8	725.5	96.6%	10.9%	120.1%	51.5%
Project 9	873.8	143.4%	38.7%	279.4%	85.5%
Project 10	1,417.2	140.4%	26.7%	235.5%	88.2%
Project 11	1,714.9	111.1%	25.6%	187.7%	78.0%

Table 2-2 Forecast vs. Actual Call Volume

[5] I define FVA as the ratio of actual calls presented to the forecasted calls presented. Each project generates daily forecasts of volume. These forecasts attempt to account for day of week effects, trends, and any special events such as holidays or changes in scope. This table is based on 4 months of daily data and uses only data for Monday through Friday because many projects have very low weekend volume.

The table reveals the challenge related to accurately forecasting volume. Most projects systematically underestimate volume. The standard deviation of the forecast error is large and the range of observed values is substantial. It is also worth noting that in general smaller (mid-market) projects are more difficult to forecast than larger projects[6].

2.4.4 Intraday Variability

In addition to day-of-week seasonality these call centers also experience very significant time-of-day seasonality. The following figure shows the average call volume presented per ½ hour period to a particular corporate help desk.

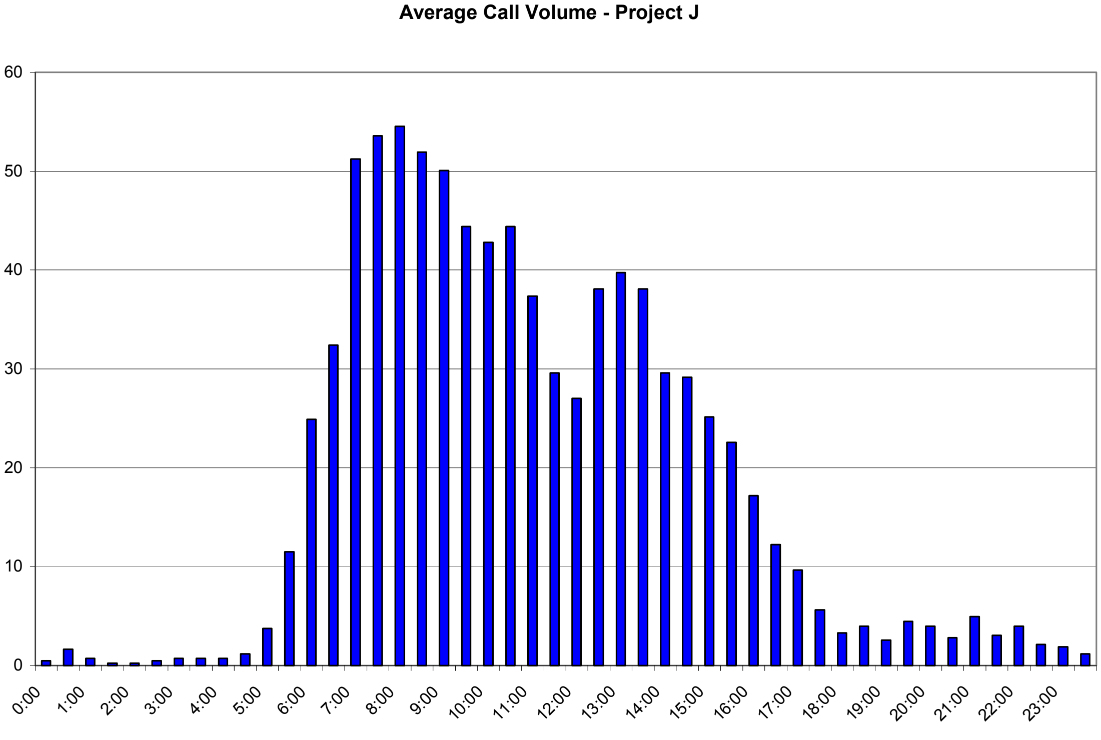


Figure 2-4 Sample Average Daily Arrival Pattern

This particular desk operates 24x7 and we see that the volume during the overnight hours is quite low. Volume ramps up sharply in the morning with a major surge of calls between 7 and 11 AM.

[6] Projects 9 and 11 represent 2 divisions of the same company, a company in the middle of a merger integration effort. The disruptions related to the merger are a major contributor to the variability of this project. Projects 7 and 8 represent two long term Fortune 500 level companies that are relatively stable; with a standard deviation of forecast error less then 15%.

Volume tends to dip down around the lunch break, but a second peak occurs in the afternoon; though the afternoon peak is typically lower volume then the morning peak.

While this basic arrival pattern exists on most business days, there is significant stochastic variability in the call pattern from day to day. The following graph shows call volume over an 8 week period for a particular project. The inner region represents the minimum volume presented in each period, while the overall envelope is the maximum volume presented in each period. The outer region then represents the variability over this eight week period.

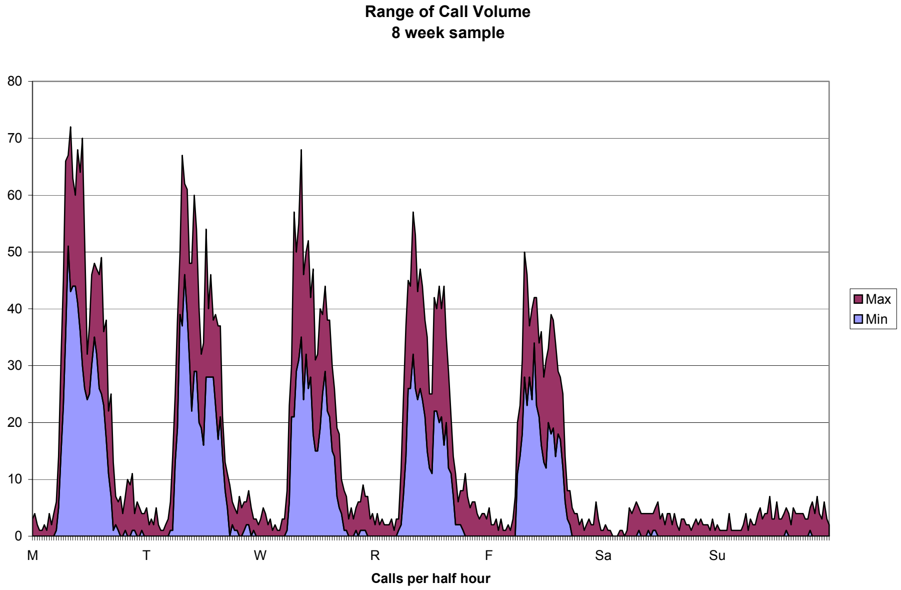


Figure 2-5 Range of Call Volume

This graph shows that while there is significant variability in call volume, a strong pattern exists.

2.4.5 A Statistical Model of Call Arrivals

In numerical analysis of stochastic systems we have three options to represent variability (Law 2007).

- Empirical (trace) data
- Empirical distribution
- Theoretical distribution

The preferred approach is to find a theoretical distribution that provides a reasonable fit to empirical data and to sample from that distribution (Law 2007). In this section I develop a relatively simple model of call arrivals and show that this model provides a reasonable fit to the observed data. The selected model is a two stage, hierarchal model of arrivals. This type of model is commonly employed in call center operations (Gans, Koole *et al.* 2003). In the first level we model daily call volume. In the second level we model the distribution of daily calls across 30 minute time periods.

2.4.5.1 Daily Call Volume

The goal of the top level model is to develop a statistical distribution for daily call volume. Daily call volume can vary for multiple reasons; including changes in the scope of support, holidays, annual seasonality, and stochastic variability. Since our concern is with stochastic variability, I ignore structural and long range seasonality issues and focus on day to day stochastic variability.

The assumption is that daily calls are generated by a stationary process with day of week and holiday effects. Mathematically I assume a model with the following form

$$\hat{y} = \overline{y} + b_M d_M + b_T d_T + b_R d_R + b_F d_F + b_{SA} d_{SA} + b_{SU} d_{SU} + b_H d_H + \varepsilon \qquad (2.1)$$

$\hat{y}$ represents the call volume predicted on any given day. $\overline{y}$ is the overall average call volume. We have dummy variables ($d_M - d_{SU}$) for day of week effects (I arbitrarily select Wednesday as the baseline day). Each dummy represents the average change in daily volume relative to Wednesday, i.e d_M represents the average difference in calls between a Wednesday and a Monday. d_H is a similar dummy variable for holiday effects. I fit a model using ordinary least squares to approximately 6 months of data and obtained the following results.

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	727.24	17.14887	42.41	<.0001
Mon	141.12421	24.69112	5.72	<.0001
Tue	69.88	24.25217	2.88	0.0045
Thu	-45.32	24.25217	-1.87	0.0634
Fri	-109.5453	24.29783	-4.51	<.0001
Sat	-669.84	24.25217	-27.62	<.0001
Sun	-651.1874	24.06047	-27.06	<.0001
Holiday	-610.3674	37.22337	-16.40	<.0001

Source	Nparm	DF	Sum of Squares	F Ratio	Prob > F
Mon	1	1	240177.6	32.6679	<.0001
Tue	1	1	61040.2	8.3024	0.0045
Thu	1	1	25673.8	3.4920	0.0634
Fri	1	1	149438.9	20.3260	<.0001

Source	Nparm	DF	Sum of Squares	F Ratio	Prob > F
Sat	1	1	5608570.3	762.8532	<.0001
Sun	1	1	5385360.5	732.4932	<.0001
Holiday	1	1	1976798.6	268.8755	<.0001

RSquare	0.936968
RSquare Adj	0.934357
Root Mean Square Error	85.74437
Mean of Response	525.0113
Observations (or Sum Wgts)	177

Table 2-3 Regression Model Results

This model provides an excellent overall fit to the data with a very high R^2 and adjusted R^2 value. Each dummy variable, other then Thursday, is significant at the .01 level, with Thursday being significant at the .06 level. This supports the notion of strong day of week seasonality effects[7].

Based on this regression an initial model for this call arrival process is as follows:

$$\hat{y} = 727 + 141 d_M + 69 d_T - 45 d_R - 109 d_F - 670 d_{SA} - 651 d_{SU} - 610 + \varepsilon \qquad (2.2)$$

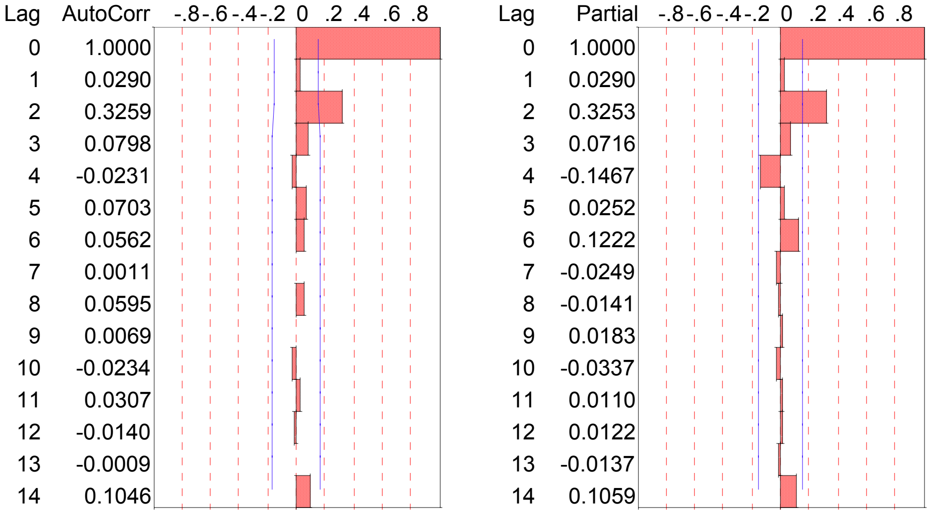
with a standard deviation of $\sigma = 85.74$. I now need to test the underlying assumptions of the linear regression model; specifically I wish to confirm that the residuals are independent and normally distributed.

2.4.5.1.1 Autoregressive analysis

An assumption of the linear regression model is independence of the residuals, but it is reasonable to assume that the arrival data may exhibit an autoregressive dependence. To test for this effect, I performed a time series analysis on the residuals from the basic regression model. The classic reference for time series analysis is (Box, Jenkins *et al.* 1994); Chapter 6 of this text outline a procedure for fitting time series models.

[7] Formally we reject the null hypothesis that day of week effects are zero.

The following chart shows the Autocorrelation and Partial Autocorrelation plots for the residuals.



Lag	AutoCorr	Lag	Partial
0	1.0000	0	1.0000
1	0.0290	1	0.0290
2	0.3259	2	0.3253
3	0.0798	3	0.0716
4	-0.0231	4	-0.1467
5	0.0703	5	0.0252
6	0.0562	6	0.1222
7	0.0011	7	-0.0249
8	0.0595	8	-0.0141
9	0.0069	9	0.0183
10	-0.0234	10	-0.0337
11	0.0307	11	0.0110
12	-0.0140	12	0.0122
13	-0.0009	13	-0.0137
14	0.1046	14	0.1059

Figure 2-6 Autocorrelation and Partial Correlation Plots

The graph shows generally low levels of correlation with lagged observations. There is a slight positive correlation with a 2 day lag. I speculate that this lag is related to call backs on previously unresolved problems. A standard diagnostic test for autocorrelation is the Durbin-Watson test (Kutner, Nachtsheim *et al.* 2005). From the original regression analysis the Durbin-Watson statistic is 1.68, which allows us to conclude that autocorrelation is not significant.

Because the autocorrelation effect is quite small, adds significant complexity to the model, and is of limited value[8], we choose to ignore it and to continue with a non-autoregressive model in future analysis.

[8] In most of our analysis staffing decisions must be made at least a week in advance. A small adjustment to the forecast two days into the future is of limited practical value.

2.4.5.1.2 Residual Distribution

Another important assumption of the linear regression model is constant variance of the residual term. The following figure plots the residuals by day of week.

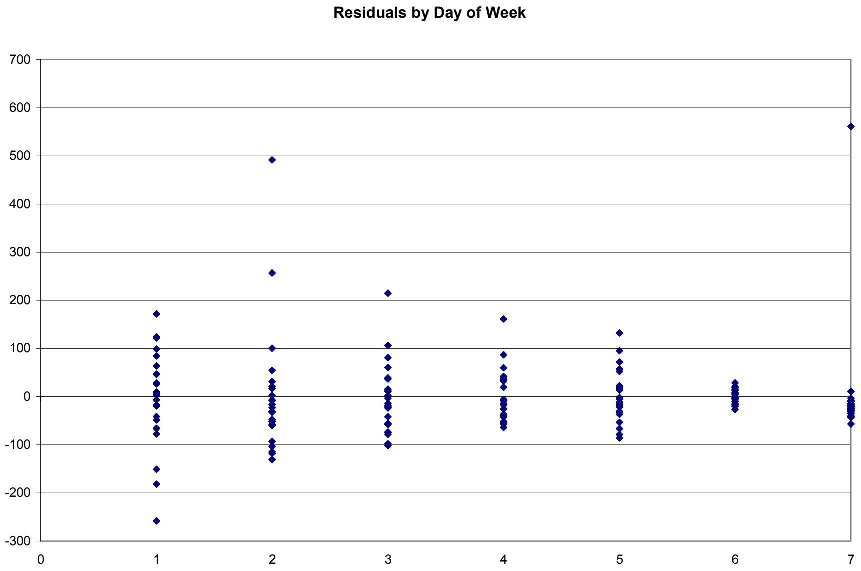


Figure 2-7 Residuals by Day of the Week

This figure clearly shows that that the residual variance is not independent of the day of the week. Saturdays, for example, have a much lower average volume and a much lower variance then Mondays. This implies that the assumptions of the standard linear regression model are not satisfied. This should be obvious from the regression output where the standard deviation of volume (85) is greater then the average volume (58) on Saturdays. This implies that negative calls can occur with positive probability, which is clearly not the case.

This analysis shows that the standard linear regression model is not valid, but we have determined the following:

- Day of week effects are statistically significant for each day of the week.
- Day of week effects explain a significant proportion of the variability in call volume.
- After considering day of week effects, the autoregressive effects are minimal and can be ignored.

- The variance in call volume is dependent on the day of week.

There are several remedial measures which can be utilized to address the variance issue, such as data transformation or weighted least squares (Kutner, Nachtsheim *et al.* 2005). However, our data suggests a straightforward approach. We assume that volume on each day is an independent random draw from a normal distribution with a mean and standard deviation specific to that day of the week. Since we have show autoregressive effects are minimal the assumption of independence is justified. Since each day's volume is generated by an independent process the variance of volume on each day can be unique.

2.4.5.2 Intraday Arrivals

Having developed a model of daily call volumes, we must now develop a model of how these calls arrive over the course of the day. It is clear from Figure 2-5 that the arrival rate varies considerably over the course of the day and we can not assume a constant arrival rate. A common approach in practice is to assume a fixed shape of the demand pattern that simply shifts vertically with volume. Mathematically this implies that the call volume in each is a fixed proportion of the daily volume. However the graph in Figure 2-6 shows that per period volume is more volatile than this assumption would indicate. In the following figure I show the average proportion of daily call volume in each 30 minute period, along with the coefficient of variation of that proportion.

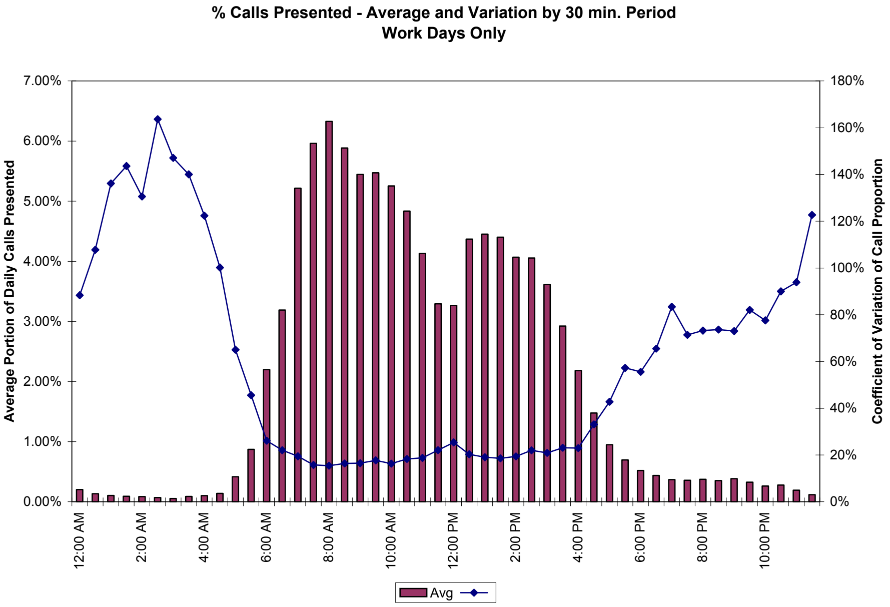
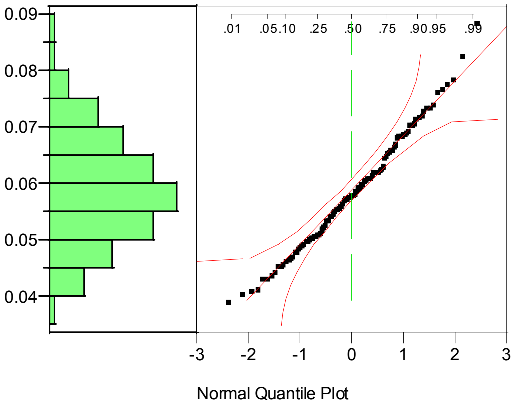


Figure 2-8 Per period Variation in Call Volume

This graph clearly shows that the call volume proportion in each period is not fixed. While the CV is roughly flat across the busy period of the day, it is in the range of .2 -.3 indicating considerable stochastic variability. During slower periods volumes are very low and highly volatile.

While the data clearly refutes the fixed proportion assumption, we may hypothesize that the call volume in any period *t*, is a proportion of total daily volume p_t, where p_t is a random variable. My analysis shows that during the busy hours (6 AM to 5 PM) the proportion of call volume presented in any ½ period has an approximately normal distribution with a coefficient of variation of approximately .2.

The following chart illustrates the distribution of call proportion in the period between 8:00 and 8:30AM for one particular project



% Arrivals at 8 AM

Figure 2-9 Arrival Proportion Normal Plot

The left panel shows a histogram of the proportion of daily volume received in this period for each observation. The right panel of the graph is a normal quantile plot of the same data. The histogram and quantile plot show that normal distribution is a reasonable approximation to the distribution of call proportion in this time frame[9]. Further analysis shows similar results for busy periods during the workday. Call volume during off peak hours is generally independent of total daily volumes. During evening hours the volume of calls presented is approximately normally distributed with no significant day of week effects. In the overnight hours calls are a rare event (many periods have no calls at all.)

We can reasonably assume a model where the call proportion of daily volume is normally distributed during busy hours. During slow hours this assumption finds less support in the data. However, practical considerations show that this assumption will introduce little error in our model. Recall that our primary motivation is to develop a model of call volume that can be used for scheduling purposes. Standard operating procedures require that a minimum of two agents

[9] An alternative distribution is the Beta distribution which unlike the normal distribution has a finite support. Call proportion can obviously not be negative and an infinite support distribution such as the normal can generate negative volumes with positive probability. Given the mean and and sd. of the proportion during high volume hrs. this is a low probability event. My analysis indicates that the normal distribution approximation is reasonable for high volume periods.

are staffed in all periods, which implies abundant capacity in the slow periods. In addition, service level agreements are based on answering a specified proportion of calls within a specified time. Since the total proportion of calls received in the slow periods is very low, small errors in volume during these periods will have limited impact on the aggregate service level attainment.

Based on these considerations we will utilize a statistical model that assumes that the proportion of total daily call volume presented in any half hour period is a normally distributed random variable. We will estimate the parameters of this random variable by calculating the proportion of volume received in each period across all the weekdays in our data set. I then calculate the average and standard deviation of this sample.

2.4.5.3 Simulating Call Arrival Patterns

This simplified model developed above presents a reasonable approximation of a stochastic process that generates call arrival patterns. An algorithm for generating a week of simulated calls is provided in the following figure

```
For d = 1 to 7
   Read $DA_d$,$DS_d$                    ' Read daily average and sd
   $DV_d$ = RndNorm($DA_d$,$DS_d$)       ' Generate random volumes
Next
For d = 1 to 7                           ' Gen initial proportions
   For t = 1 to 48
      Read $TA_t$,$TS_t$                 ' Read period average and sd
      $TP_t$ = min[RndNorm($TA_t$,$TS_t$),0]  ' Calc initial proportion
      $SP_d$ = $SP_d$ + $TP_t$           ' Sum up proportions
   Next
Next
For d = 1 to 7                           ' Normalize proportions
   For t = 1 to 48
      $TV_{dt}$ = $TP_t$*$DV_d$/$SP_d$   ' Calculate period volume
      $LAM_{dt}$ = 2* $TV_{dt}$          ' Calculate arrival rate
   Next
Next
```

Figure 2-10 Simulated Call Generation Algorithm

This algorithm has 3 loops. In the first loop we calculate daily call volumes by generating random normal variates with the daily mean and standard deviation. In the second loop we calculate the preliminary proportion of daily volume realized in each 30 minute time period. The third loop normalizes the proportion and calculates the average realized volume in each period and the associated arrival rate.

2.5 Abandonment

An important consideration in call center operations is abandonment, the proportion of callers who decide to hang up prior to being serviced. The abandonment rate is a key parameter tracked in most call centers. In this context the firm makes a distinction between negative abandonment and positive abandonment. Positive abandonment is the proportion of callers who hang up without waiting on hold for an extended period of time. The rationale is that when a known problem is identified a recorded message is usually played to all new calls stating the problem and the expected resolution time. Callers who hear this message and hang up are assumed to have been serviced as they learned that their problem is known and learned of the expected completion time. Positive abandonment is therefore not considered a problem. Formally, positive abandonment is usually calculated as the number of callers who abandon with a wait time of 30 seconds or less.

Negative abandonment on the other hand occurs when a caller chooses to hold through this initial period, but ultimately hangs up before they are serviced. Abandonment rates tend to vary widely and are correlated to wait time. When queues build up wait time increases and callers are more likely to abandon[10]. Abandonment rate therefore tend to be the highest when volumes are high and capacity tight.

[10] Abandon calls count against the service level, in that an abandon call was not serviced but counts as a received call. In this sense one minus the abandonment rate is an upper limit on the service level.

The following graph illustrates the daily abandonment rate for a particular project over a three month period.

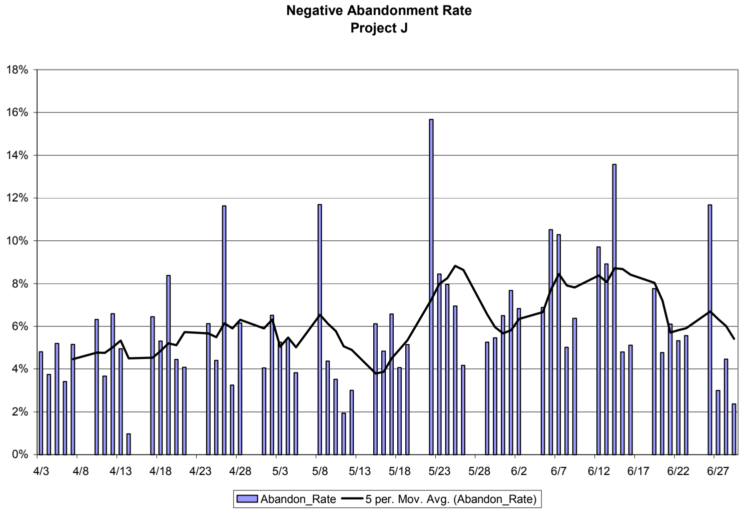


Figure 2-11 Abandonment Rate - Stable Project

The abandonment rate is seen to vary considerably and spike up on busy days. The overall abandonment rate for this project is typically in the range of 4%-6%, rarely below 2% and as high as 15%. The particular project shown here is a relatively stable project with relatively low abandonment.

The abandonment rate for a more volatile project is show in the following figure:

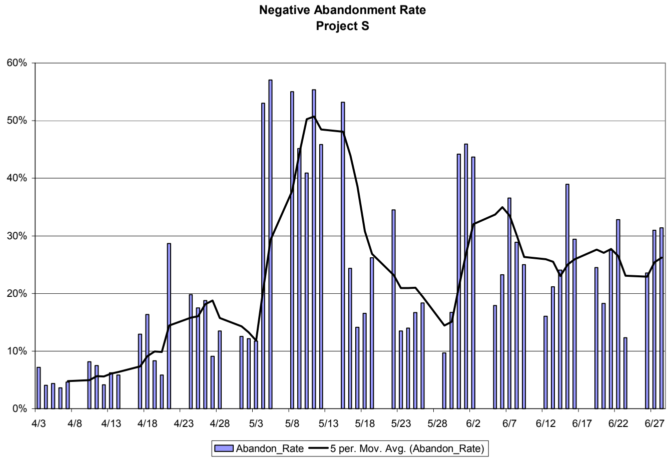


Figure 2-12 Abandonment Rate - Unstable Project

This particular problem encountered serious service quality problems in early May as changes in scope caused call volume to increase faster than capacity. Average waiting time increased to over 20 minutes and significant number of callers decided to abandon. Abandonment rates peaked at over 50% for several days, and even after capacity adjustment were made abandonment rates remained in the 20%-30% range.

Over this time period the average abandonment rate for the 11 projects listed in Table 2-2 varied between 2.5% and 22%. Overall this analysis shows that abandonment is an important consideration that needs to be considered in any planning model.

2.6 New Project Launch

One of the key challenges in this business model is the new project launch process. A significant problem is determining the appropriate number of agents to hire and train. Because of the substantial (project specific) training investment required for new hires, management is reluctant to hire extra workers. Standard operating procedures call for hiring to the projected steady state level based on expected call volumes[11]. As in the case of the short term scheduling problem the decision is complicated by uncertainty. Attrition levels are again uncertain, as is demand. The level of demand uncertainty is very high prior to the *go live* event because accurate counts of current call volumes are often extremely difficult to obtain. Business process changes involved with the transition, such as call center consolidation, changes in hours of operation, or changes in the type of support provided, often make previous data of limited value, even if known. Another major complicating factor is the evolving level of productivity due to learning curve effects. Talk times tend to decline during a launch as agents become more familiar with the environment and the project knowledge base becomes better populated. Variability in talk time is subject to institutional learning curve effects, individual learning curve effects, and stochastic variability. A final complicating factor is the lead time required to add new capacity. Recruiting new hires can take time, but the biggest factor is training time. Since agents must provide detailed technical support they require extensive training before they can be deployed on the help desk. Training times are project dependent and vary from two weeks to three months.

[11] Accounting issues are also important as projects are generally expected to be profitable from launch.

The following graph shows the average talk time over the first 3 months of a major launch that occurred in 2005.

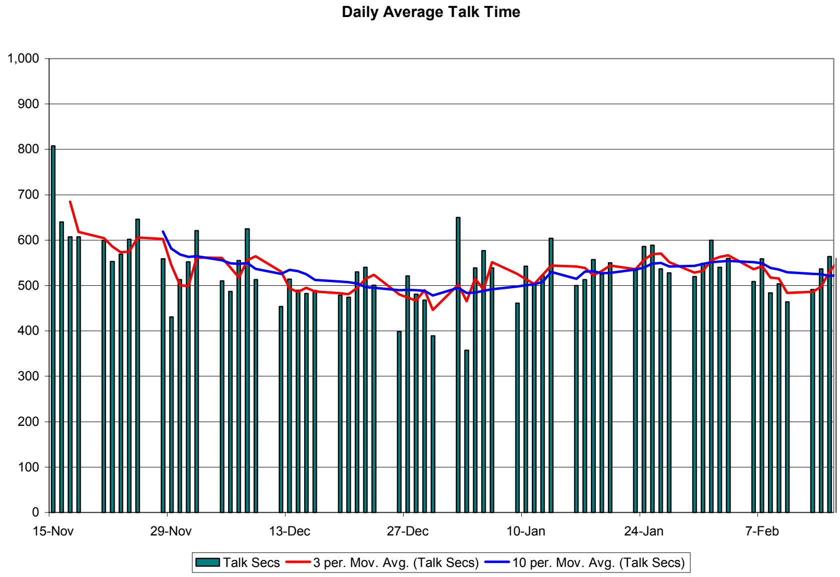


Figure 2-13 Talk Time Evolution during Startup

This graph reveals a general decline in talk time (increase in productivity) during the first several weeks of the launch, followed by a leveling off and a slight increase in the third month. The increase in talk time in January is due, at least in part, to the addition of new hires made to replace resigning workers[12]. This particular project involved a phased deployment where large groups of agents were added at various time through the extended launch period. If we plot the average talk time over a longer period, we can clearly see the impact of new agents and the learning curve effect.

[12] Six new hires were made in December for a project with a total headcount at that time of about 35. These individuals began taking calls in early January. I don't have individual talk time data for this project.

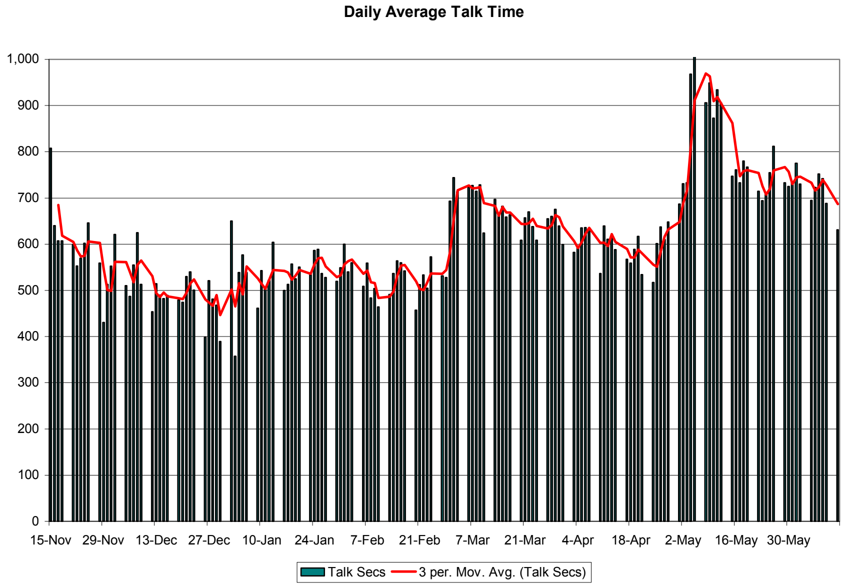


Figure 2-14 Talk Time Shocks

The start up problem has is illustrated by a recent launch of the company's single largest customer. Based on the scope of this launch the decision was made to conduct a phased launch effort, adding new users every few weeks over an extended period. Unfortunately this created multiple forecasting challenges. As the following graph shows, the inability to ramp up capacity along with demand led to extremely poor quality of service over an extended period of time.

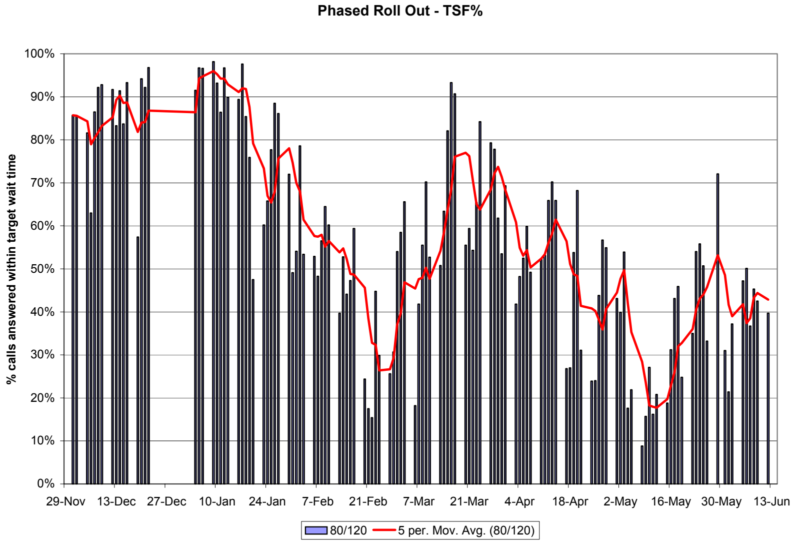


Figure 2-15 Service Levels During rollout

The challenge of forecasting the new demand with each subsequent roll out, coupled with learning curve issues as new agents were added with each rollout, created acute mismatches in capacity and demand. The service level target for this project was 80%, but as the graphic shows actual service levels were well below this target for extended periods with several periods of extremely poor performance[13].

2.6.1 A Statistical Model of Learning Curve Effects

To develop a statistical model of learning curve effects I collected individual agent scorecards from a sample project. These scorecards are prepared for each front line worker each month and assess the worker on a number of key operating metrics including talk time, wrap time, first tier closure rate, inside call volume, and monitoring scores[14]. The data set included scorecards for 2004 through November 2006. The data was pulled from the individual scorecards and arranged by month of service. I included only agents where I had at least 3 months of contiguous data. The resulting data set had measure for 53 agents with length of service ranging from 3 months to 19 months.

As a proxy for agent productivity I examined talk time, first tier closure rate, and inside call volume. My hypothesis is that as agents learn they will resolve more problems, in less time, with less help from other agents.

[13] The problems associated with this launch have provided the motivation for the company to rethink its launch process.

[14] Talk time is the average time the agent spends on the phone per call, while wrap time is the post call time spent processing data from the call. First tier closure rate is the proportion of calls resolved directly by the agents, as opposed to escalation to a tier 2 agent. Inside call volume are calls placed by the agent to other agents seeking help to resolve difficult problems. Monitoring score are the scores given by QA personnel who anonymously monitor a portion of calls and grade agents against a broad list of subjective performance measures.

The following graph shows the reported monthly average talk time for each agent as a function of their month of service.

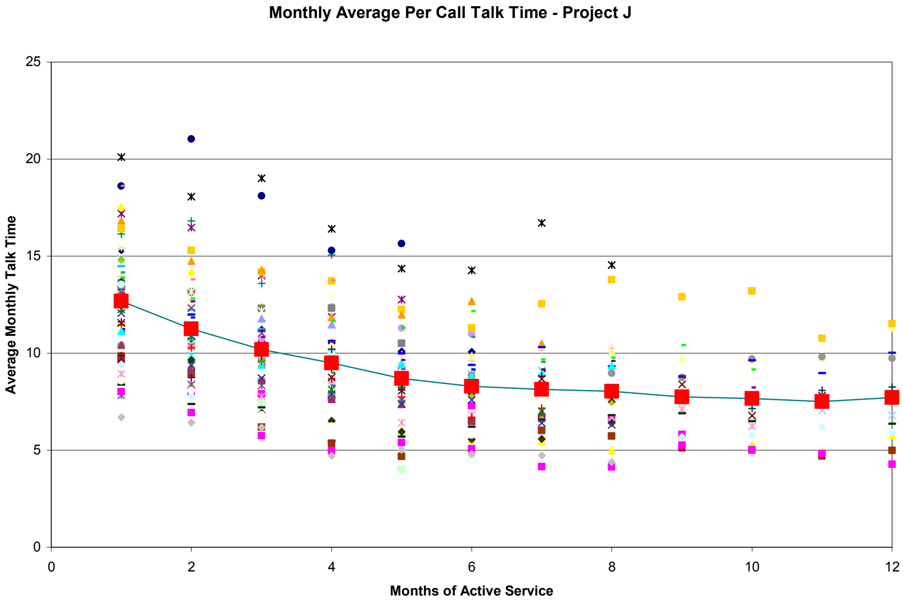


Figure 2-16 Monthly Average Talk Time

Each point on this graph represents an individual agent wmployed for a month. The data reveals a general decline in average talk time over the first several months of service as expected. Talk time declines from an average of 12.7 mins in the first month, to 8.3 mins in the 6th month. The standard deviation of talk time ranges between 3.0 and 2.5 over this period. However, when I examined first tier closure rate and inside call volume the picture became a little more complicated.

The following graph shows that First Tier Closure rate decreases over the first few months of service.

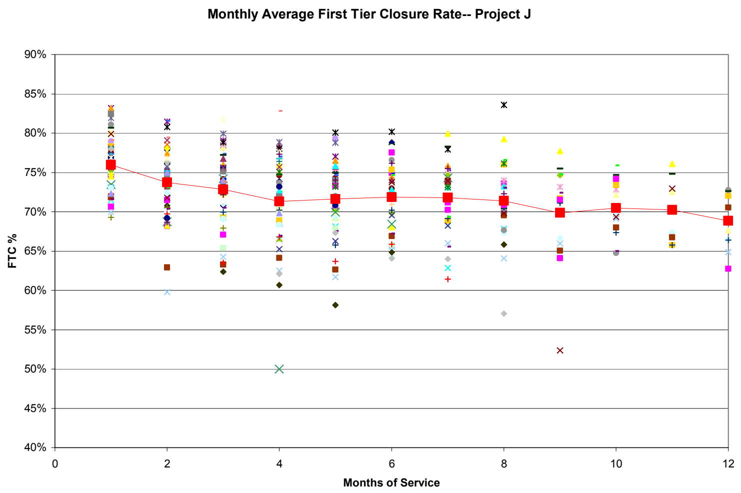


Figure 2-17 Monthly Average First Tier Closure Rate

This unexpected result is partially explained by the Inside Call Volume statistic. Shown in the following graph:

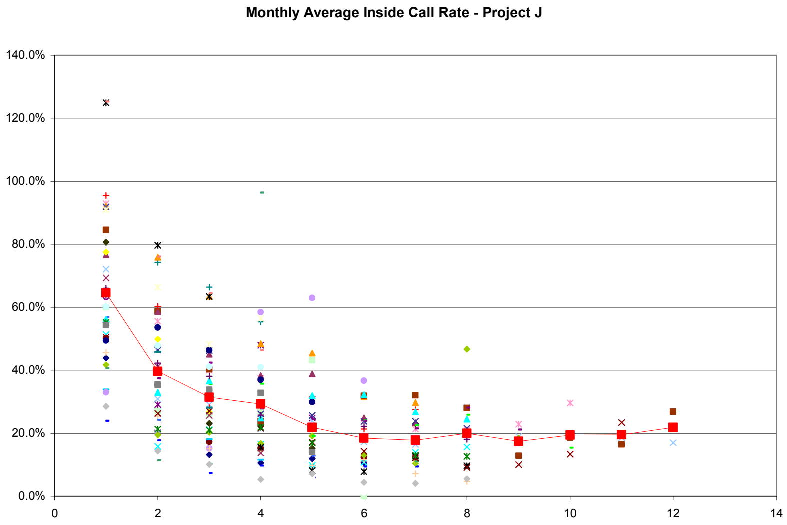


Figure 2-18 Monthly Average Inside Call Rate

During the first few months when agent experience is low they draw upon more experienced agents to help them solve a large portion of their calls and because of this they are able to close a relatively high percentage of calls without escalation.[15] So as agents progress they are able to resolve calls faster with less outside support. As a first preliminary measure of productivity we use the talk time measure as a surrogate of productivity.

I first compare the average talk time for all agents in their nth month of service to the average of the n-1th month of service. Using a standard T-test I find the reduction is statistically significant at the .1 level through the first 5 months of service. Average talk time continues to decline through the first 11 months of service, but the month to month changes are statistically significant only at the .5 level. If we evaluate the 2 month improvement the improvement is significant at the .01 level through the 6th month.

This analysis shows us that average talk time for more experienced agents is lower than talk time for less experienced agents; but does not give us conclusive evidence as to why talk time decreases. Two explanations are possible; first agents learn and become more productive and second slower agents are removed from the system. To verify that individual agents become more productive I examine the one month difference for individual agents.

The data is summarized in the following graph

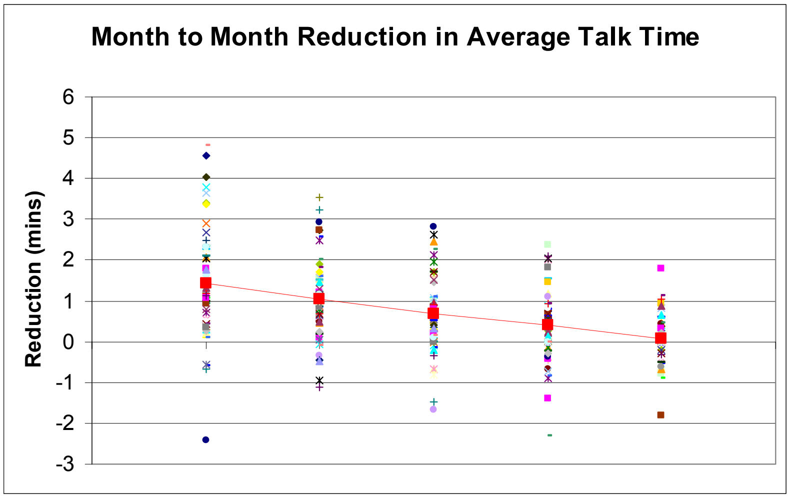


Figure 2-19 Monthly Reduction in Talk Time

[15] Management policy prevents agents on this project from escalating tickets without concurrence of another more experienced agents until they earn their escalation rights, typically sometime in the 2nd or 3rd month of service.

The graph shows positive reductions (improvements) in average talk time through the first 5 months. The improvement in month 6 is not statistically significant.

All of this data suggests that a standard learning curve model is appropriate. Given the data I have, I developed a learning curve model based on months of service rather than cumulative call volume. I fit a model of the form

$$T_t = T_0\left(1+e^{-\alpha n}\right), n \geq 1 \tag{2.3}$$

where T_t is the average talk time in period t, T_0 is the average talk time for an experienced agent (6 months +), n is the month of service, and α is the learning curve rate parameter. I fit a curve to the total average talk time measure[16] and calculated an α value of 0.4605.

The curve gives a reasonably good fit as the following graph illustrates:

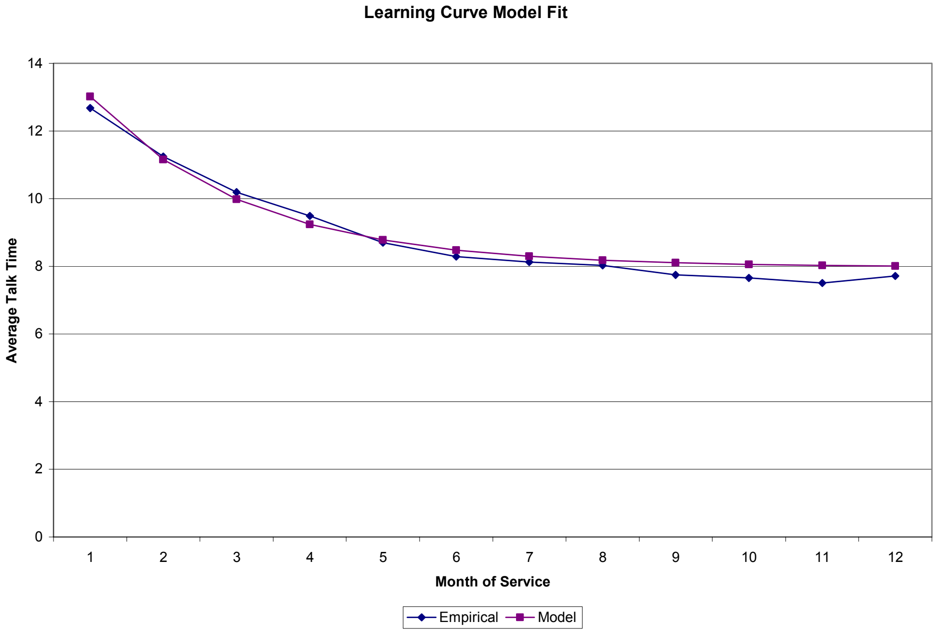


Figure 2-20 Learning Curve Model Fit

I was able to collect similar data for a second project and performed a similar analysis. I obtained slightly different, but similar results. On this project the improvement is statistically significant at

[16] I fit the curve using Excel's solver to find the value at minimizes the sum of squared errors.

the .2 level only through the 4th month of service, indicating a somewhat faster learning process. The corresponding alpha value for this project is .703.

The learning curve model expressed in (2.3) is quite flexible and can be used to represent a wide range of learning curve effects as the following graph illustrates.

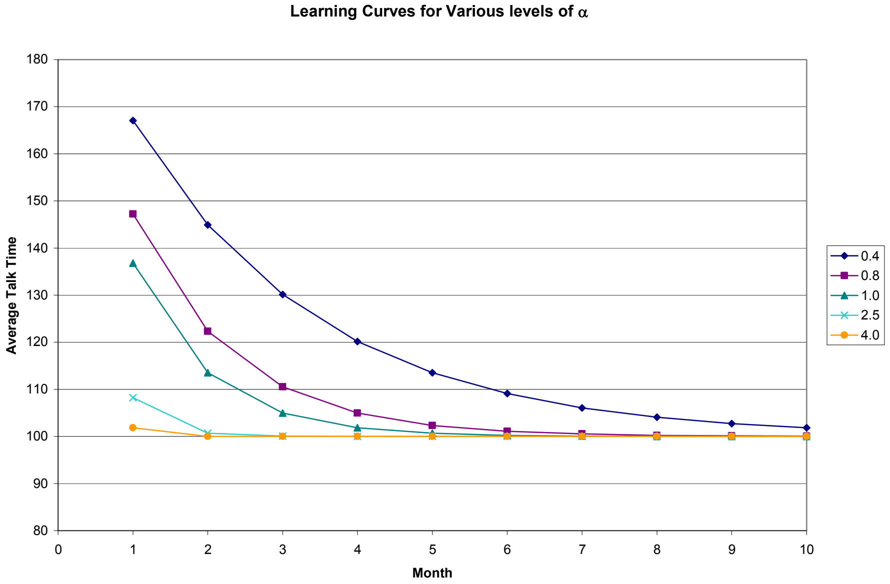


Figure 2-21 Family of Learning Curves

The curve is valid for any $\alpha \in (0, \infty)$. For large values of α the curve is relatively flat; new agents perform nearly as well as experienced agents and any gap is quickly closed. As the value of α decreases, the curve becomes steeper and the learning effect is more pronounced. The limitation of this functional form is that α must be strictly greater then zero and the initial effort can be no more then twice the long run effort. This limitation is easily overcome by adding a 2nd scaling parameter to (2.3), but that is not necessary to fit our data.

2.6.1.1 Relative Productivity

In Figure 2-19 we show the impact of learning on talk time. An alternative way to think of learning is the impact on relative productivity or capacity. As learning occurs agents become more productive and able to handle more calls and increase their effective capacity. Based on equation (2.3) average talk time will settle in at the value of T_0. If we equate a relative productivity level of one with a talk time of T_0 then we can define the relative productivity index ρ as the ratio of the average talk time of the inexperience agent with that of the experience agent $\left(T_0 / T_t\right)$ or

$$\rho_t = \frac{1}{1+e^{-\alpha n}} \tag{2.4}$$

The relative productivity will evolve as shown in the following graph

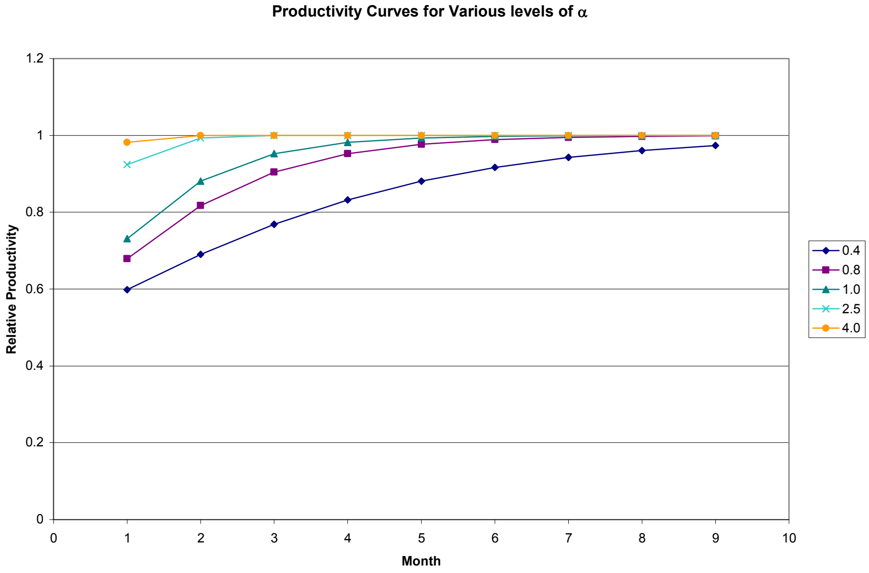


Figure 2-22 Relative Productivity Curves

This graph shows that with an α value of .8 a new agent can handle approximately 68% as many calls as a fully experienced agent, based simply on talk time. A more complete analysis would take a broader measure of agent productivity then just talk time. Based on the inside call volume statistic shown in Figure 2-16 , new agents place a high burden on experienced agents by asking them questions. While we have a measure of the number of inside calls made we have no data on the duration of calls. Qualitatively, the data indicates that the burden placed on experienced agents decreases with time and we can conclude that the productivity curves of Figure 2-20 understate the productivity improvement associated with learning.

2.7 Turnover Issues

As is often the case in call center environments, turnover in this company is a significant issue. The following graph shows the month by month annualized turnover rate over an approximately 28 month period.

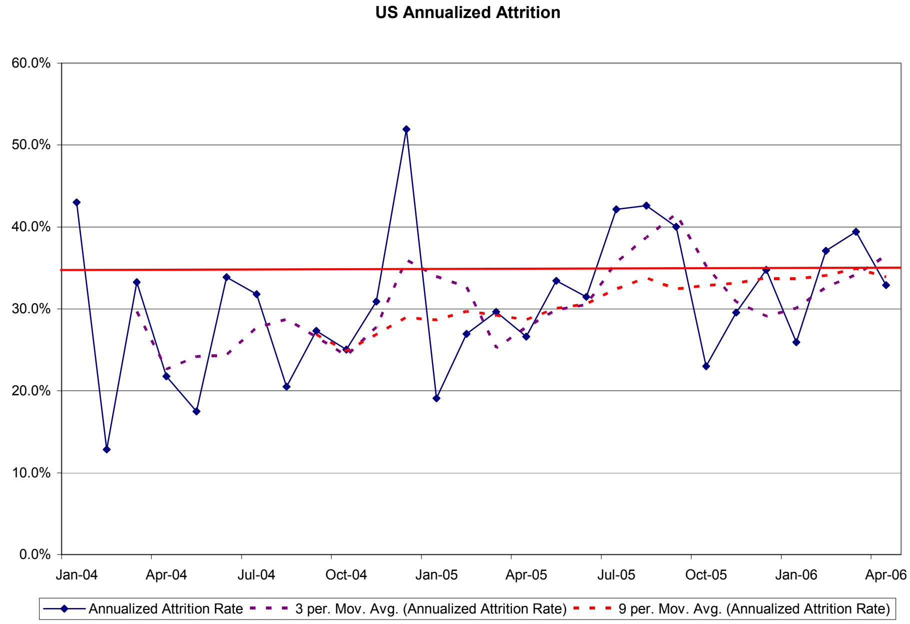


Figure 2-23 Annualized Attrition Rates by Month

We see that turnover varies significantly from month to month, with the 9 month moving average in the range of 25-35% per year. A widely used model for employee attrition estimates the attrition probability as a function of length of service. (See for example (Bartholomew and Forbes 1979)).

I collected detailed termination data on the 1,400 terminations (voluntary and involuntary) that occurred between January 2001, and May 2006. I used this data to estimate a hazard rate function for the probability of quitting.

The data was fit to a Weibull distribution with shape parameter equal to .918 and scale parameter equal to .0309. The hazard rate function derived from that distribution is shown in the following graph:

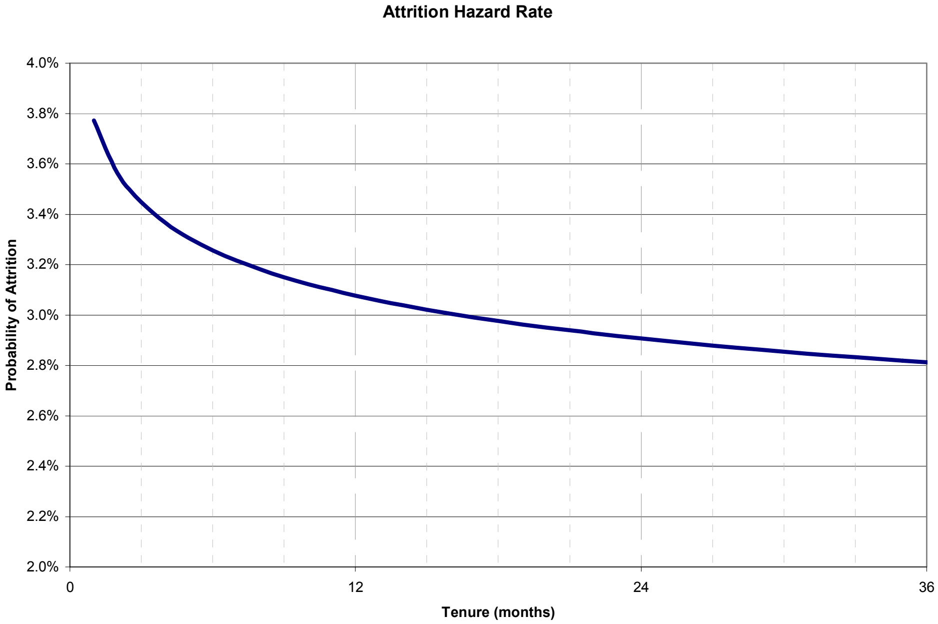


Figure 2-24 Attrition Hazard Rate

The analysis reveals a decreasing failure rate function; that is the probability of quitting declines with length of service[17]. This is consistent with summary data that shows that over this period approximately 15% of new hires quit within the first three month of employment. We termed this the *washout rate*. The observations that are relevant for the analysis in this dissertation are the following. First, attrition rates of 3.0%-3.5% per month are realistic, and second the probability of quitting declines with length of service. These observations, and parameter values, will become important when we analyze new project start ups. The high attrition rate, especially among new hires, implies that we must consider attrition when staff planning for a new project.[18]

[17] A Weibull distribution will have a decreasing failure rate if the shape parameter a is less then 1 and an increasing failure rate if the shape parameter is greater then 1. If the shape parameter is equal to 1 the Weibull is equal to the exponential distribution and the failure rate is constant.

[18] Interestingly, the company currently does not factor attrition levels into new project start ups. Hiring is currently capped at the number of agents specified in the long run cost model for the project.

2.8 Capacity Management Practices

The primary organizational structures at this firm are the *account* and the outsourcing *project*. For the most part each account/project is managed as a stand alone operation with its own management structure and its own profit and loss statement. Based on this decentralized management structure each project team has wide latitude in terms of how it performs capacity management functions.

While the firm owns a sophisticated Work Force Management (WFM) tool that can perform complex scheduling tasks, the tool is used by very few project teams. Most teams perform scheduling tasks in a semi-automated fashion utilizing Excel spreadsheets. Managers collect historical call volumes from the ACD and use that data to develop a call volume forecast; typically for each 30 minute period over the course of a week. Most managers use the average per period over a six to eight week period, and then manually adjust that schedule based on holidays or other known changes in scope. The forecast is used to drive overall staffing requirements. Several managers use Erlang C calculators, simple applications that calculate the staffing level required to achieve a specified service level in each time period, to create a candidate staffing profile. However, because of the large peak in volume during the 8-12 AM time period, most managers do not staff to this level. Instead they under staff during the morning peak, and overstaff during the afternoon period. The objective is to achieve a service level below target in the morning, above target in the afternoon and on target overall. Balancing over and under staffing is done heuristically, based on experience and intuition. Staffing plans also account for other constraints, such as the general requirement to always have at least two agents staffed during any period.

While most managers perform these tasks using spreadsheets, a small number use the automated WFM application. The general process here is the same but the tool provides automated support. The tool collects ACD statistics which the manager can manually adjust. A menu of feasible schedules and other side constraints can be established and an automated schedule generated. Manual adjustment of the schedule can be performed after optimization. The tool's documentation is quite vague about the nature of the scheduling algorithm. The scheduling algorithm does provide a slider control that allows the planner to select between 1 - Minimize spikes in service level, and 2 - Maximize overall (weekly) Service Level. The tool can account for abandonment either by allowing the user to enter an abandonment rate forecast or a patience factor[19].

For the most part agents are scheduled to full time schedules (40 hours per week) that remain constant from week to week. Some projects have implemented flex scheduling; whereby an agent's scheduled hours may be different every day. This practice has been controversial and is believed by some to lead to increased attrition rates.

The WFM tool is widely used in planning new project launches. Forecasts of daily arrival rates are collected from whatever sources are available and allocated to 30 minute time periods by applying a standard seasonality model, based on other projects. The WFM staffing model is run from this point estimate of per period arrivals to calculate the number of agents required to achieve the targeted service level. The estimated number of agents required are then recruited, with no allowance made for learning and or turnover. All agents are hired to a specific project requisition in a hire to order process.

2.9 Model Projects

Throughout this dissertation we will analyze decision models developed in the context of multiple *model projects*. These model projects are based on real projects currently in operation. I selected these projects so that I could analyze the behavior of the models under various realistic operating conditions. Each project has unique characteristics that create operational challenges. Taken together this set of projects provides a broad test bed upon which to evaluate the models.

The projects I review are summarized as follows:

- **Project J**: an IT help desk for a large US based corporation.
- **Project S**: an IT help desk that supports store operations of a large US based retail chain.
- **Project O**: an IT help desk that supports corporate and store based operations for a medium sized retail chain.

[19] It is not clear if the patience is a average figure from some distribution or a deterministic point estimate. Online documentation asks the user to enter the Patience - *This represents the length of time a caller will wait before hanging up*, which seems to imply a point estimate. Knowledge of the technical details of the scheduling algorithm even by those who actively use the tool is very limited.

2.9.1 Project J

Project J provides Help Desk and desk side support for approximately 30,000 users in 13 locations across the United States. The project has approximately 98 dedicated staff members, of which approximately 45 are dedicated to help desk support. The help desk receives approximately 16,000 calls per month. The project is subject to an 80/60 TSF SLA, as well as a 75% first tier closure rate SLA. Talk time on this project averages 12 minutes. Daily call volume for this project is relatively stable as the following graph illustrates:

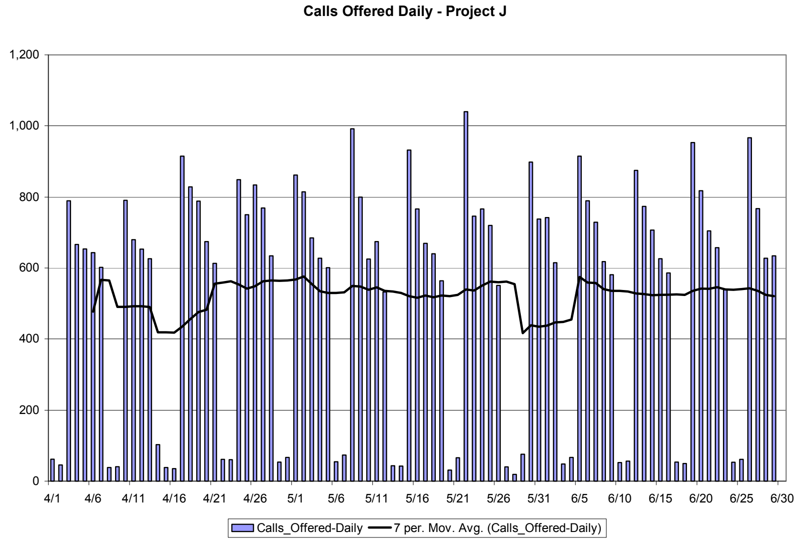


Figure 2-25 Project J Daily Arrivals

Since this project primarily supports corporate users weekend call volume tends to be quite low. A typical weekend day has 40-60 calls, while on weekdays call volume is in typically in the range of 500-800 calls, although volumes can rise higher on busy days. This project exhibits a typical weekly seasonality pattern, with volumes generally declining during the working week.

The intraday pattern also follows a standard corporate project pattern; an early morning peak, followed by a lunch lull and a second smaller afternoon peak. The following graph illustrates.

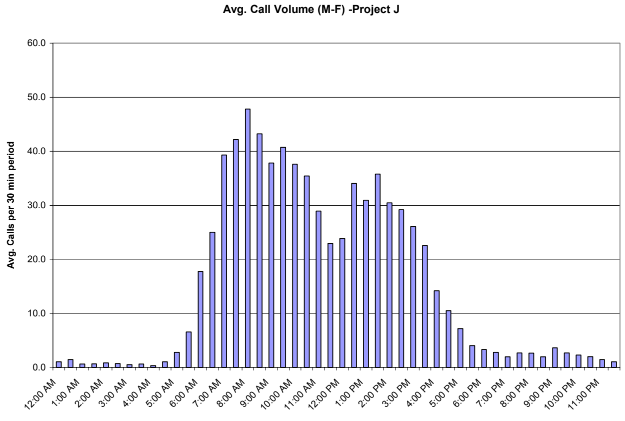


Figure 2-26 Project J Average Intraday Arrivals

2.9.2 Project S

Project S provides support for a large retail chain that is undergoing significant disruption in operations due to a merger. Volume tends to be very high and unpredictable. The project supports approximately 350,000 end users at 3,400 stores. The project generates between 40,000 and 45,000 calls a month. Approximately 125 staff members are dedicated to this project. The following graph illustrates daily volume during a period that includes the final roll out in a phased roll out process.

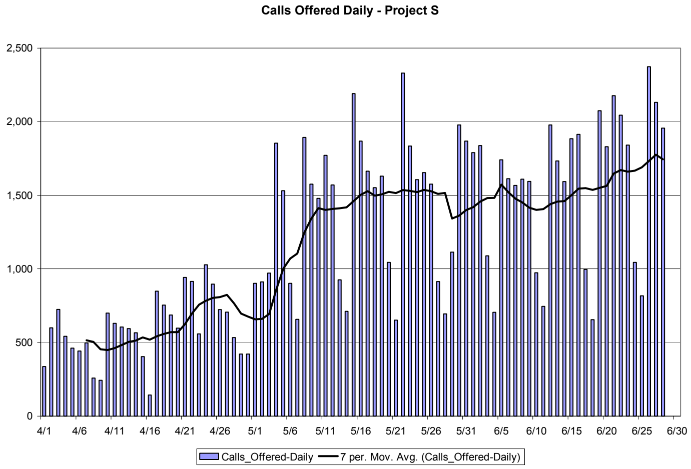


Figure 2-27 Project S Daily Arrivals

We see that call volume tends to be more volatile. We can also see that because stores are open 7 days a week weekend volumes tend to be much higher then on a corporate project. The intraday seasonality pattern is also different for this project as the following graphic illustrates.

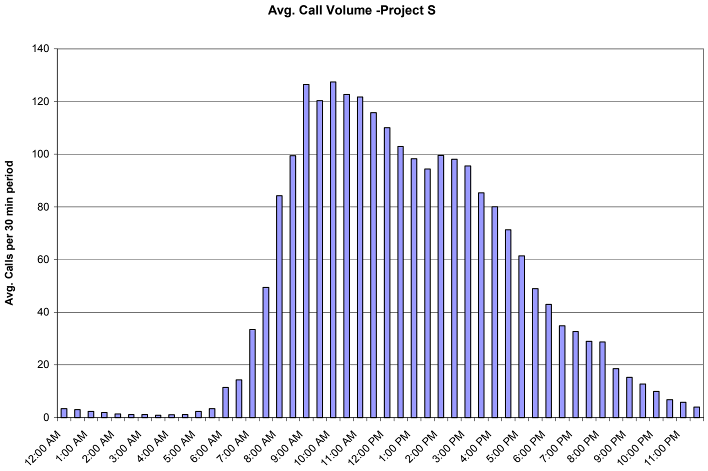


Figure 2-28 Project S Average Intraday Arrivals

The double-hump pattern of the corporate project is much less pronounced and the evening trail off of call volume is much more gradual. This project is subject to an 80/120 TSF SLA. Talk time on this project averages 13.5 minutes.

2.9.3 Project O

Project O provides support to a medium sized retail chain. The project supports approximately 10,000 users at 1,070 retail locations plus the corporate office. The project generates about 15,000 calls per month and has approximately 40 dedicated staff members.

As the following graph illustrates, the project is considerably smaller than project S and overall it is less volatile, though it subject to very large spikes. The day to day pattern is similar to Project S.

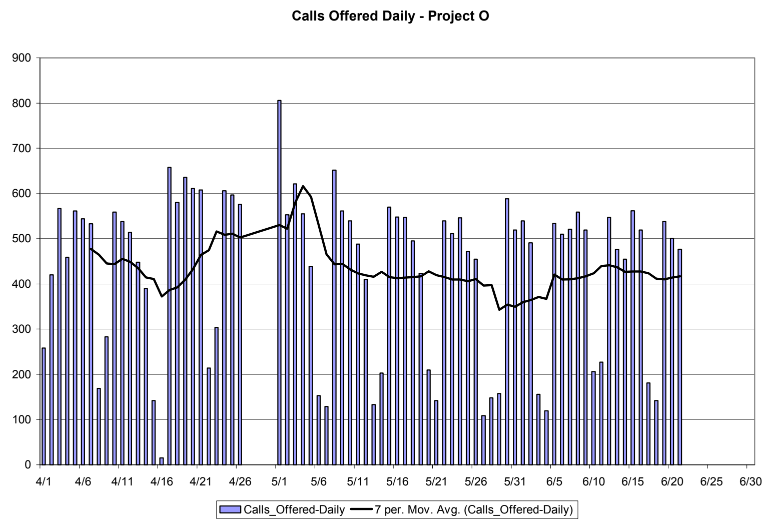


Figure 2-29 Project O Daily Arrivals

The daily pattern has much less seasonality then other projects. The morning and afternoon peaks present in other projects is very limited on this project.

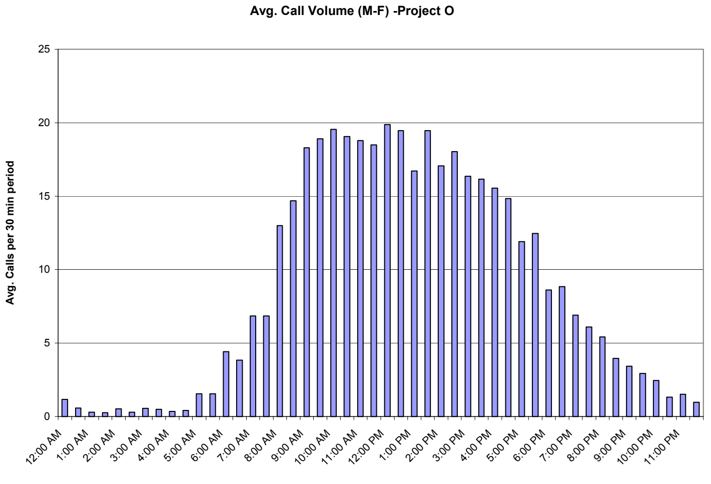


Figure 2-30 Project O Average Intraday Arrivals

2.9.4 Statistical Models of Model Projects

Statistical models were developed for each of these projects using the approach outlined in section 2.4.5. For each project I eliminated holidays from the data set and identified shock days. The day of week effect was then calculated by estimating the mean and standard deviation of arrivals on each "normal" day. I then estimated the proportion of calls received in each 30 minute period along with the associated standard deviation. Finally I estimated the probability, mean and standard deviation of shocks. A summary of the data for each of these model projects is shown in the following table:

	Project J	*Project S*	*Project O*
Support Base	Corporate	Retail	Corp/Retail
Hours of Operation	24x7	24x7	24x7
SLA	80/60	80/120	80/120
Average Weekly Volume	3,825	10,600	3,000
Talk Time	12	13.5	14
Shock Probability	0	3.0%	1.3%
Mean Shock Volume	0	792	267
Shock Standard Deviation	0	72	20

Table 2-4 Model Project Summary

While these estimates are based on several months of data, a more accurate model fitting would require a larger data set. Estimating shock parameters in particular was challenging given the length of this data set. My intent is not to develop specific forecasting models for these projects; rather it is to develop representative and realistic models of projects that can be used to validate the decision models, and to generate insight into the operating characteristics of different classes of projects.

2.9.5 Summary

The analysis of this company's operations provides the motivation, and the data, to support the development of several capacity management models. The data demonstrates the significant variability in load and illustrates the capacity management challenge. The model projects selected also demonstrate the relative scope of operations inherent in support desk operations.

2.10 Summary of Research Problem and Objectives

In this section I presented an empirical analysis of data associated with an IT support outsourcing company. The purpose of this review is to highlight some of the key operational challenges associated with this type of operation, to provide representative data for future analysis, and to motivate a set of optimization problems. While this data has been collected from a single company, it in fact represents operational data from a number of different companies and government agencies around the world and is therefore quite general.

Important observations we can make from this analysis include the following:

- Arrival rates are highly variable.
- Arrival rates exhibit day of week and time of day seasonality.
- Forecasting arrivals is very difficult and prone to substantial error.
- Hiring and training costs for support agents is costly.
- Attrition rates are high.
- Learning curve effects are significant and vary from project to project.

From this data we identify several research questions of theoretical and practical value. These questions include the following:

- How should we hedge against uncertainty when scheduling call center agents?
- What impact do uncertainty and variability have on optimal staffing levels?
- How can we create operating systems more robust to uncertainty and variability?

I address these questions by developing three specific models for call center capacity management.

- **Short Term Scheduling Model**: a model that accounts for variability and uncertainty when scheduling agents.
- **Medium Term Hiring Model**: a model that considers uncertainty, variability, and learning effects when setting initial hiring levels for a new outsourcing project.
- **Cross Training Model**: a model that examines how cross training can be used to create lower cost systems that are more robust to uncertainty.

In the next section we review the existing academic literature relevant to these problems. In sections 4 – 6 we develop and analyze each of these models. In section 7 we summarize major conclusions and discuss directions for future research.

3 Literature Review

3.1 Introduction

In this section I review and summarize some of the key literature relevant to this dissertation. I review literature in the following areas:

- **Manpower Planning**: literature related to manpower capacity planning. I review tactical and strategic models, wastage analysis, and models of manpower planning systems in practice.
- **Call Centers**: literature specifically addressing call center operations.
- **Stochastic Optimization**: literature that addresses methodological issues related to optimization under uncertainty.
- **Design of Statistical Experiments**: literature that addresses methodological issues related to the design, execution, and analysis of statistical experiments.

3.2 Manpower Planning

3.2.1 Overview

Manpower planning became a very popular topic in the early years of Operations Research. Many research papers and multiple texts (Holt, Modigliani *et al.* 1960; Charnes, Cooper *et al.* 1978; Bartholomew and Forbes 1979) addressed aspects of the manpower planning problem. As shown in Figure 1-1 of the introduction, it is useful to separate the manpower planning problem based on the length of the planning horizon.

In tactical planning, workforce capacity is considered fixed and the objective is to develop efficient schedules that balance firm and individual goals and constraints. Short term planning involves determining time phased resource requirements, while rostering involves assigning individuals to specific schedules. In strategic planning, workforce capacity is a decision variable. In medium term planning we seek to make near term capacity adjustments through new hiring and termination. Long term planning involves shaping the workforce over an extended period of time and considers issues such as career progression and skill shifting. In the following sections we review the literature on tactical and strategic manpower planning. We also review the practice-oriented literature, papers that consider how manpower planning models have been applied in real world industrial or governmental organizations.

3.2.2 Tactical Manpower Planning

The tactical planning problem deals with the assignment of a specific number of resources to detailed schedules. This problem has been analyzed in the literature extensively, dating back to the set covering problem originally modeled by Dantzig in 1954 (Dantzig 1954). Dantzig's model was outlined in a short 'letter to the editor' and formulated tactical scheduling as a linear program. His model assumes the work day can be divided into a number of discrete periods, say 15 or 30 minute buckets, and that the required number of resources for each time period can be specified. Dantzig's model also assumed that a number of standard work patterns or shifts could be defined, specifying the starting and ending period of work, along with any breaks. Mathematically the Dantzig model can be expressed as

$$\text{Minimize}\sum_{j=0}^{n} x_j \tag{3.1}$$

Subject to

$$\sum_{t=0}^{n} a_{tj} x_j \geq b_t, x_j \geq 0 \tag{3.2}$$

Where b_t represents the number of workers required in period t and the decision variable x_t denotes the number of workers assigned to shift j. The coefficient a_{tj} is equal to one if shift j works in period t and is zero otherwise. In this simple form all shifts are equally costly and the objective is to minimize the total number shifts scheduled. A simple extension adds a cost coefficient c_j to the shift which facilitates shifts of different length, or shifts with differential wage rates. Dantzig's model requires one decision variable for each shift corresponding to the number of workers assigned, and one constraint for each time period.

This model seems quite straight forward on the surface, but is in fact rather powerful, and unfortunately can become rather complex computationally. There are three features of this model that can make it computationally difficult. First, is the practical requirement for all decision variables to be integer valued[20]. Second is the issue of continuous scheduling; that is 24 hour operation with no down time. Finally and perhaps most significantly, is the issue of explicitly scheduled breaks.

Consider a more significant problem addressed in Henderson and Berry (1976). They evaluate a phone company operator scheduling problem. Operators are scheduled for 8 hour shifts, with a variable length lunch break and 15 minute rest break explicitly scheduled. The model addresses only the peak demand hours of 6:00 AM to 12 midnight. Because of the requirement to schedule a 15 minute break that planning horizon is divided into 72 fifteen minute periods. Staffing requirements are defined exogenously and vary over time as in the following example:

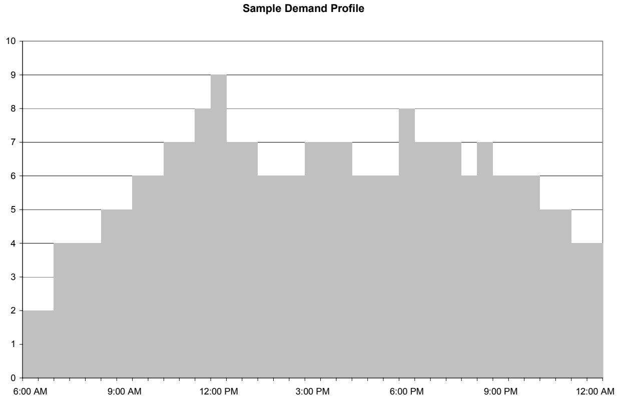


Figure 3-1 Sample Agent Requirements

Given the various options for starting times and the possible combinations of break times, the model includes 7,120 different decision variables. The model also includes 72 constraints to account for the demand in each period.

The requirement to explicitly model breaks is costly. Without breaks there are at most 72 different full time shifts, one for each discrete time period. So fully 98% of the decision variables are a direct result of the explicit break scheduling problem. It is worth noting that the Henderson and Berry model simplifies the problem to avoid the continuous (24 hr) staffing problem. Their model schedules only the peak hours of 6:00 AM to midnight which allows them to treat each day

[20] Dantzig's original model was solved by hand as an LP. He suggested rounding non-integer solutions, while rounding is a viable approach for scheduling a toll booth; it is not practical for a call center with hundreds, or thousands of potential shift patterns.

as a separate scheduling problem. Without this simplification the number of schedule periods would increase by a multiplicative factor equal to the number of days in the planning horizon.

The staff scheduling problem is a special case of the set covering problem in which the objective is to minimize the weighted sum of set coverings, with the weights being the cost of each shift. The Staff Scheduling problem is known to be NP Complete unless it possesses the cyclic 1's property (Garey and Johnson 1979); that is unless each shift is continuous with no breaks. NP Completeness implies the lack of a polynomial time solution algorithm.[21] Practically this means staff scheduling problems that include explicit breaks are inherently unscalable. The majority of the research related to staff scheduling is related either directly or indirectly, to this scalability problem.

3.2.2.1 Continuous Shifts

A number of papers analyze staff scheduling problems that do not explicitly schedule breaks. Without this requirement the problem is no longer NP Complete and a number of polynomial time approaches are available. Segal shows that without breaks the problem can be modeled as a network flow problem (Segal 1974). Baker examines the problem of scheduling full time nurses to meet a deterministic staffing requirement in a series of papers. The first model (Baker 1974a) considers only full time workers and attempts to find an optimal allocation of days off during a weekly (7 day) cyclical schedule where each employee is scheduled for two consecutive days off. Another paper extends this model to allow for the added flexibility of scheduling part-time employees along with full time employees (Baker 1974b). A third model (Baker and Magazine 1977) again considers only full time employees, but evaluates a number of different day off policies. This series of papers shows that efficient algorithms can be developed for specific cases, obviating the need to solve integer programs. The nurse scheduling problem is further analyzed in another set of papers by Warner *et al*. The model in (Warner and Prawda 1972) also avoids the explicit break problem, but introduces other complications that also make the problem computationally difficult. An important feature of this model is the ability to substitute one class of nurse for another with some proportional efficiency. For example, an LPN may be schedule in

[21] It has not been proven that no polynomial time algorithm exists for NP Complete problems, but it is widely believed to be true. If a polynomial time algorithm were to exist for the staff scheduling problem it must be true that a polynomial time algorithm exists for all NP Complete problems.

place of an RN, but provides only 70% of the RN's productivity. (Warner 1976) builds on the nurse scheduling problem and addresses the issue of rostering, the assignment of specific individuals to shifts. A detailed and up to date review of the nurse rostering problem is provided in (Burke, De Causmaecker *et al.* 2004)

3.2.2.2 Fixed Break Scheduling

The introduction of explicit break time into the scheduling problem adds considerable computational complexity, making the problem intractable for reasonably sized problems. Researchers have addressed this problem in several ways; through problem simplification, heuristic methods, and alternative algorithms. The Henderson and Berry model (Henderson and Berry 1976) applies two type of heuristics. The first heuristic reduces the number of shift types, scheduling against only a reduced set of schedules referred to as the *working subset*. The second approximation is the scheduling algorithm; the authors use 3 different scheduling heuristics.

An alternate stream of research attacks the problem using an implicit scheduling approach. Implicit scheduling models generally use a two-phased approach, generating an overall schedule in the first phase, and then placing breaks in the second phase. Implicit scheduling approaches are addressed in (Bechtold and Jacobs 1990), (Thompson 1995) and (Aykin 1996). (Thompson 1995) includes a summary of related papers and then develops a Doubly-Implicit Shift Scheduling Model (DISSM). (Aykin 1996) Several other papers addresses related problems (Brusco and Johns 1996; Brusco and Jacobs 1998; Brusco and Jacobs 2000).

A succinct overview of a two-stage approach to scheduling in a call center environment is provided in section 12.7 of (Pinedo 2005). This model is motivated by a call center application where resource requirements are defined exogenously and breaks must be scheduled.

Pinedo summarizes his approach in the following figure

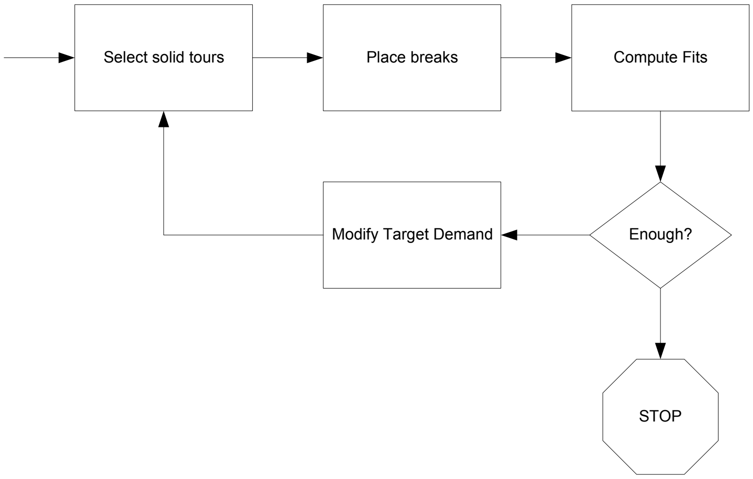


Figure 3-2 Iterative Scheduling Approach

The Select Solid Tours step uses a math programming approach to fit schedules without break considerations to Target Demand. Target Demand is the overall requirements inflated to be somewhat higher than actual demand to account for loses due to breaks. The break placement step uses heuristics to schedule breaks into the solid tours. The compare fitness step calculates a fitness function with the following form

$$\Im = \psi^{-}\sum_{t=1}^{H} e^{-}(t) + \psi^{+}\sum_{t=1}^{H} e^{+}(t) \tag{3.3}$$

In this calculation the difference between the required staffing level and the scheduled staffing level is denoted $e(t)$; $e^{+}(t)$ is the overstaffing level and ψ^{+} is the overstaffing penalty. The algorithm seeks to minimize the total cost measure

$$\mathrm{C} = \Im + \sum_{j} c_j x_j \tag{3.4}$$

An overall fit measure is defined as the smoothness

$$L = \sum_{t=1}^{H} e(t)^2 \tag{3.5}$$

which is calculated as the sum of squared deviations from requirements.

(Cezik and L'Ecuyer 2007) solve a global service level problem using simulation and integer programming. They use simulation to estimate service level attainment and integer programming to generate the schedule. The IP model generates cuts via sub-gradient estimation calculated via simulation. The model solves the sample average problem and therefore ignores arrival rate uncertainty, but it does allow for multiple skills. This model is a an extension of the model presented in (Atlason, Epelman *et al.* 2004). In a related paper (Avramidis, Chan *et al.* 2007) use a local search algorithm to solve the same problem. A related model is presented in (Avramidis, Gendreau *et al.* 2007).

3.2.3 Strategic Manpower Planning

The strategic capacity planning literature is in general divided into two complementary approaches. In one approach the evolution of the workforce is modeled as a stochastic process that evolves over time (Bartholomew and Forbes 1979; Bartholomew 1982). This approach explicitly models the stochastic nature of hiring, turnover, skills acquisition and demand. An alternate approach is based on an optimization paradigm in which the objective is to make a set of control decisions over time that optimize some measure of system performance, such as total cost, deviation from staffing plan, or expected profit (Holt, Modigliani *et al.* 1960). More recent work has attempted to integrate uncertainty and optimization, which is the focus of my research.

3.2.3.1 Workforce Capacity as a Stochastic Processes

Stochastic models of manpower systems focus on the uncertainty inherent in the system. Bartholomew provides a general review of the application of stochastic modeling to social systems in (Bartholomew 1982), and a more specific application of these principals to the manpower planning problem in (Bartholomew and Forbes 1979). A basic model incorporates a number of discrete manpower grades. The state of the system is then defined as the number of employees currently in each grade. If we make the standard Markov assumptions then the system can be modeled as a Discrete Time Markov Chain (DTMC).

Many papers have built on this simple Markov model to analyze manpower systems, introducing various control objectives into the process. Grinold develops a stochastic model motivated by the demand for naval aviators (Grinold 1976). The environment evolves as a Markov process and the demand for aviators therefore has a definable probability distribution. The control objective is then to find the optimal accession policy that governs new entrants into the system, and the continuation policy that governs movement through the system. A useful feature of the model is the ability to distinguish between gross headcount and *qualified* headcount and an important implication of this model is that changes in capacity are not instantaneous but rather are driven by the delays required to train new recruits. This issue is further addressed in (Anderson 2001). In this model demand is driven by a continuous nonstationary seasonal process meant to approximate a business cycle. The model explicitly assumes employees progress at differential rates, unlike the deterministic rates in Grinold. The objective trades off the discounted cost of meeting demand requirements with a penalty term for abrupt changes in the employee stock. Based on this objective Anderson uses a dynamic programming approach to define optimal control policies. A similar model that focuses on cohort analysis is developed in (Gaimon and Thompson 1984). The Gaimon and Thompson model postulates that the effectiveness of an individual can be defined exogenously as a function of organizational age and grade. Effectiveness may be defined to increase throughout the individual's career, or it may be specified to peak at some organizational age and begin to decline in an environment with rapid technological change.

A number of other papers examine the strategic staffing problem using a stochastic setting. (Gans and Zhou 2002) develop a model with learning curve and stochastic turnover issues. (Gaimon 1997) examines manpower planning in the context of knowledge intensive IT workers. (Bordoloi and Matsuo 2001) also examine a knowledge intensive work environment with stochastic turnover.

3.2.3.2 Strategic Manpower Planning as an Optimization Problem

An alternative approach to manpower planning is based on optimization theory. The theoretical foundations of the optimization approach to manpower were developed in Holt *et al.* (Holt, Modigliani *et al.* 1960) Holt evaluates manpower as a component of the productive capacity of a manufacturing enterprise, evaluating staffing decisions in an aggregate planning context. Holt develops a quadratic cost model that includes both the costs of maintaining a workforce and the cost of changing the workforce. Holt's quadratic cost model is converted to a linear cost model in (Hanssmann and Hess 1960) and solved via linear programming. The Holt model is also extended in (Ebert 1976) with the inclusion of time varying productivity. Ebert uses the quadratic cost model directly from Holt, but allows productivity to vary over time as learning takes place. Ebert solves this non-linear program using a search heuristic. An alternative formulation that also includes learning curve effects is presented in (Harrison and Ketz 1989). This model is non-linear but is solved via successive linear programming.

The two approaches to manpower planning outlined above emphasize different aspects of the system and as such have different applications. The stochastic models are generally high level abstractions useful for identifying system phenomenon or developing general policies. The optimization models on the other hand are often crafted to identify specific management actions but tend to ignore the variability in the system. Variable parameters are typically modeled with their expected values yielding what is known as the *mean value problem* which may result in solutions that are far from optimal (Birge and Louveaux 1997). Modeling variability in optimization problems is likely to yield solutions that are superior to the deterministic counterparts, but solutions to these stochastic programs are difficult to find.

3.2.4 Wastage Analysis

"*Of all the flows in a manpower system, wastage is the most fundamental for manpower planning*", so states (Bartholomew and Forbes 1979). In this context wastage refers to the total loss of individuals from the manpower system regardless of the reason. Wastage has two basic components, voluntary attrition or turnover (quitting) and involuntary terminations (firing.) In either case headcount, and capacity, is removed from the system.

Call centers are notoriously difficult work environments and high turnover rates are common. A number of papers in the Management and Organization/Human Resources literature address the work environment and turnover problem in general, and several address call center specific issues. (Witt, Andrews *et al.* 2004) examine issues of emotional exhaustion in a survey of 92 call center agents. Specifically they examine the relationship between exhaustion and performance. (Singh, Gollsby *et al.* 1994) survey 377 agents and find burnout levels among customer service agents are high relative to other high stress occupations. (Singh 2000) examines how burnout affects productivity and quality. (Cordes and Dougherty 1993) review the literature on job burnout. (Holman 2002) surveys 577 call center agents to assess several measures of employee well being. (Cotton and Tuttle 1986) perform a meta-analysis of the papers available at the time that examined turnover across industries. (Abelson and Baysinger 1984) develop a conceptual model of turnover and argue that the optimal level of turnover balances retention and turnover costs. Their argument is that some positive level of turnover is desirable, but they provide no hard data as to what level is optimal.[22]

My focus in this dissertation is not to analyze the causes of turnover, or to address the issue of reducing turnover, my concern here is simply to develop models of turnover that are useful for planning purposes[23].

Basic models of wastage are developed in (Bartholomew and Forbes 1979). A common objective is to develop models that describe turnover patterns and facilitate forecasting of future wastage rates. These models typically include some independent variables that segregate the employee pool to homogeneous groups. The independent variables could include age, job level, gender, or length of service. A common choice is to separate the workforce based on date of hire into cohorts; employee groups with roughly the same hire date (Bartholomew and Forbes 1979).

(Bartholomew 1971) describes the propensity to leave as a function of length of service via the *force of separation* function $\phi(x)$ defined as

$$\Pr\left\{\begin{array}{c}\text{individual leaves with length of service in}\\ (x, x+\delta x)\text{ given he survives to x}\end{array}\right\} = \phi(x)\delta x \tag{3.6}$$

This is mathematically equivalent to the survivor function $G(x)$ which is probability that an individual survives (remains employed) for a time *x*. Its complement $F(x)$ is the distribution of the completed length of service, which in a continuous representation may have a density function $f(x)$. In a continuous mode, equation (3.6) is equivalent to the *hazard rate* (aka the *failure rate*) function defined as follows (Ross 2003).

$$r(t) = \frac{f(t)}{1-F(t)} \tag{3.7}$$

The three alternative representations (density function, survivor function, hazard function) are all mathematically equivalent; given one the other two can be derived.

[22] Presumably the 30%+ level observed is above optimal.

[23] In my work with this company we developed several strategies for reducing turnover, but that work is outside the scope of this dissertation.

While an empirical representation of the survivor data may be sufficient in some situations, a common objective is to fit some theoretical distribution to the data. An important consideration in selecting a distribution is the shape of the failure rate function. With an increasing failure rate the propensity to quit increases with length of service. With a decreasing failure rate the propensity to quit declines with length of service. A constant failure rate implies the probability of quitting is independent of length of service. Most empirical analysis of turnover suggest a decreasing failure rate (Bartholomew and Forbes 1979).

Many statistical distributions have been proposed to model wastage rates. (Bartholomew and Forbes 1979) discuss the exponential model which has a constant failure rate. While the constant failure rate simplifies analysis, it is not always a good fit to observed data. (Bartholomew and Forbes 1979) also present a mixed exponential model, along with the lognormal model. The lognormal model is attractive because it can match a commonly observed pattern where the mode of the distribution is separate from, but near the origin, and the distribution has a long tail. They argue that the lognormal model is a good empirical fit to many length-of-service data sets, but that the theoretical arguments as to why the distribution should be lognormal are weak. Though not discussed in the text, the lognormal distribution has a failure rate curve that may be decreasing, but may be increasing for a period and then decreasing.

Another distribution commonly applied in reliability analysis is the Weibull distribution, which is perhaps the most widely used distribution for lifetime analysis (Lawless 2003). The Weibull distribution has two parameters, β and λ[24]. The density function of the Weibull distribution is

$$f(t) = \lambda\beta\left(\lambda t\right)^{\beta-1} e^{-(\lambda t)^{\beta}} \qquad t > 0 \tag{3.8}$$

The corresponding hazard rate function

$$h(t) = \lambda\beta\left(\lambda t\right)^{\beta-1} \tag{3.9}$$

The Weibull distribution provides a flexible model for reliability since with a β greater than one the failure rate is increasing, with β less than one the failure rate is decreasing, and when β equals

[24] The literature presents the Weibull distribution in a variety of forms and notations; I adopt the form presented in Lawless (2003) p. 18. Many authors write the density with a scale parameter $\alpha = \lambda^{-1}$.

one the failure rate is constant. Note that when β equals one, the Weibull distribution simplifies to the exponential distribution.

3.2.4.1 Model Fitting

Various methods exist for fitting historical data. While the cohort approach is appealing, we have the difficulty associated with censoring; the fact that we do not have survival times for those employees who have not yet quit. To avoid this problem some authors rely instead on cross sectional or census data. Both methods are used to fit non-parametric models in (Forbes 1971). An application of the census method is presented in (Price 1978) where a non-parametric survivor function is estimated from data supplied by a large Canadian organization. Our focus will be on the estimation of parametric models from (censored) cohort data.

The key issue associated with fitting a distribution to survivor is the issue of data censoring; specifically right censoring of the data. Suppose we have resource data that provides hire dates for all employees hired over some period of time, along with the associated termination date for those employees that have separated. For each separating employee we have a lifetime observation. But for those still employed we have only a lower bound on lifetime; we know they have survived to their current length of service but don't know how much longer they will survive.

A standard approach for dealing with censored data is presented in (Lawless 2003) whereby we fit an empirical distribution to the survivor data, adjusted for censoring, then fit a distribution to the empirical model to develop a parameterized model. If no censoring exists in a sample of size *n*, then the empirical survivor can be calculated as

$$\hat{S}(t) = \frac{\text{Number of observations } \geq \text{t}}{n} \qquad t \geq 0 \tag{3.10}$$

This is simply a step function that decreases by *d/n* at each observed lifetime, where *d* is the number of observations at a given lifetime.

Now consider the case where we have censored observations. The Kaplan-Meier estimate (KM) (aka the Product-Limit estimate) calculates the empirical survivor function as follows:

$$\hat{S}(t) = \prod_{j:t_j<t} \frac{n_j - d_j}{n_j} \tag{3.11}$$

where d_j is the number of failures observed to time t_j, and n_j is the number of individuals at risk at time *t*, i.e. the number of samples uncensored and alive at time t. Equation (3.11) is a nonparametric, maximum likelihood estimate (MLE) of the survivor function, and again is a decreasing step function. This plot is easily generated from many statistical analysis packages, an example is shown below.

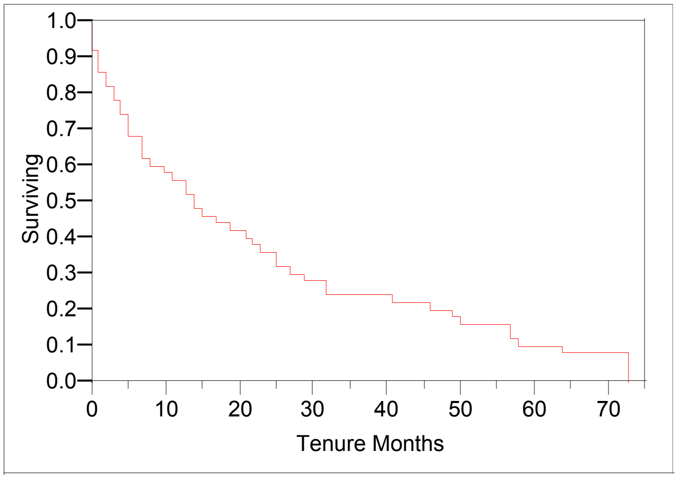


Figure 3-3 Sample Empirical Survivor Function

A distribution, such as the Weibull, can then be fit to the data using a Maximum Likelihood (MLE) estimation processes, either with or without explanatory covariates. (Lawless 2003) provides an overview of model generation and inference.

3.2.5 Manpower Planning in Practice

There are a number of papers in the literature describing the implementation of manpower planning systems in practice. Many of these papers are focused on the tactical scheduling problem. (Schindler and Semmel 1993) describes an application used to schedule airline station agents. (Mason, Ryan *et al.* 1998) addresses a related problem, scheduling customs officials at the Aukland, New Zealand airport. (Gaballa and Pearce 1979) study telephone agent scheduling at Quantas. (Andrews and Parsons 1993) develop a model that determines the required number of agents to schedule at a call center based on an optimization of staffing and customer service costs. (Saltzman and Mehrotra 2001) also examine the issue of call center staffing. (Yu, Pachon *et al.* 2004) describe a system developed for Continental Airlines that includes an interesting mix of short and long term planning decisions.

There are several other papers in the literature that address the strategic manpower planning in practice, primarily in the context of the United State military. A long rang planning system for the U.S. Army is described in 2 papers, (Holz and Wroth 1980) and (Gass, Collins *et al.* 1988). (Bres, Burns *et al.* 1980) describe a model was developed for the U.S Navy in the 1970s. (Shrimpton and Newman 2005) describe a model used to designate career fields for officers in the US Army. (Krass, Pinar *et al.* 1994) develop a model for allocating personnel to combat units for the US Navy. (Morey and McCann 1980) analyzes the issue of resource allocation toward recruiting.

3.2.6 Manpower Planning Summary

A summary of the research related to strategic manpower planning is provided in the following table.

	(Bartholomew 1982)	(Bartholomew and Forbes 1979)	(Grinold 1976)	(Anderson 2001)	(Gaimon and Thompson 1984)	(Gans and Zhou 2002)	(Gaimon 1997)	(Bordoloi and Matsuo 2001)	(Grinold and Stanford 1974)	(Holt, Modigliani et al. 1960)	(Hanssmann and Hess 1960)	(Ebert 1976)	(Harrison and Ketz 1989)	(Yu, Pachon et al. 2004)	(Gass, Collins et al. 1988)	(Shrimpton and Newman 2005)	(Holz and Wroth 1980)	(Bres, Burns et al. 1980)
Planning Approach																		
Stochastic Modeling	X	X	X	X	X	X	X	X	X									
Optimization									X	X		X	X	X	X	X	X	X
Foundation																		
Theory	X	X	X	X	X	X	X	X	X	X		X	X					
Practice														X	X	X	X	X
Resource Indexing																		
Skill	X	X				X		X						X	X	X		X
Grade	X	X			X										X		X	
Length of Service	X	X	X		X										X			X
Geography														X				
Trained				X														
Transitions																		
Hiring	X	X	X	X	X	X	X	X	X	X		X		X	X		X	X
Voluntary Attrition	X	X	X	X	X	X	X	X		X		X			X		X	X
Layoffs	X	X		X	X		X			X		X						
Skill Redeployment														X	X			
Promotions	X	X							X						X			
Demand																		
External-deterministic						X				X				X	X	X	X	X
External-variable												X	X					
Stochastic-stationary					X		X		X									
Stochastic-nonstationary				X														
Markov Process			X															
Miscellaneous																		
Recruitment Source																		X
Training						X												
Learning Curve												X	X					
Forecasting		X								X							X	

Table 3-1 Strategic Manpower Planning Summary

A summary of the research related to tactical manpower planning is provided in the following table.

	Dantzig (1954)	Henderson and Berry (1976)	Baker (1974 a)	Warner and Prawda (1972)	Bechtold and Jacobs (1990)	Thompson (1990)	Thompson (1995)	Aykin (1996)	Brusco and Jacobs 1988	Brusco and Johns 1996	Brusco and Jacobs 2000	Segal (1974)	Koole and Sluis 2003	Pinedo (2005)	Andrews and Parsons 1993	Mason et al. 1998	Saltzman and Mehrotra 2001
Optimization Approach																	
Linear Programming	X					X						X		X			
Integer Programming				X	X		X	X	X	X	X					X	
Optimal	X		X	X	X		X	X			X		X				
Nonlinear programming				X													
Simulation																X	X
Heuristic		X							X	X		X		X	X	X	X
Foundation																	
Theory	X	X	X	X	X	X	X	X	X	X	X	X	X	X			
Practice															X	X	X
Considerations																	
No breaks	X		X	X								X	X				
Explicit breaks		X															
Implicit Breaks					X	X	X		X	X	X			X			

Table 3-2 Tactical Manpower Planning Summary

Many of the practice papers dealing with strategic manpower planning deal with military sources. The military is an interesting special case with several unique properties. For the most part all new personnel enter the system at the entry level and promotions come from within. Attrition or reenlistment rates are critical variables in the model, although they occur with more regularity (at the end of fixed enlistment periods) then in the commercial sector. In some cases, such as the Army's COMLIP system, the military can set fixed recruiting levels through the compulsory draft. But perhaps the most significant difference between the military and the professional services environment is the manner in which manpower requirements are defined. In the military context manpower levels are established and fixed external to the system by legislation. The planning objective is to meet the externally defined goal. In the professional services environment demand occurs randomly in response to market fluctuations.

3.3 Call Center Operations

3.3.1 Overview

There is a relatively large body of literature that addresses call center operations, directly or indirectly. (Gans, Koole *et al.* 2003) provides a detailed tutorial on call center operations and extensive survey of the academic literature. A more recent but unpublished literature review is provided in (Aksin, Armony *et al.* 2007). The Gans *et al.* paper provides a general framework for call center research that categorizes research into the following categories:

- Queuing performance models
- Queuing control models
- Human resource issues
- Service quality and customer/agent behavior
- Statistical analysis of call centers

(Gans, Koole *et al.* 2003) documents a series of industry standard performance measures used to evaluate call center performance. We summarize the key measures here:

- **Telephone Service Factor (TSF):** also called the "service level", TSF is the fraction of calls presented which are eventually serviced and for which the delay is below a specified level. For example, a call center may report the TSF as the percent of callers on hold less then 30 seconds.
- **Average Speed of Answer (ASA)**: this is the average time calls spend on hold, waiting for an agent.
- **Abandonment Rate:** callers that are put on hold and hang up while in queue are said to have *abandoned* the system. The proportion of all calls that abandoned is known as the abandonment % and is a key metric in most call centers.

The paper also outlines a set of call center regimes, three basic categories that describe the staffing/customer service objectives of the call center.

- **Quality Driven Regime**: customer waiting costs are assumed to dominate the cost of capacity and the objective is to serve the majority of customers without delay. Staffing levels are increased linearly with offered load. Average utilization in this regime is typically low, on the order of 65-75% and average customer wait time is also low.
- **Efficiency Driven Regime**: staffing costs are assumed to dominate the cost of customer delay and the operational objective in this regime is to maximize the efficiency of the operation.
- **Quality Efficiency Driven (QED)**: an operational environment that attempts to strike some balance between efficiency and customer service is the QED regime. Unlike the quality regime where the fraction of delayed customer is near zero, or the efficiency regime where the fraction delayed is near one, the QED regime balances costs and

attempts to achieve some steady delay proportion between 0 and 1. QED operations are facilitated by economies of scale, because call center staffing is subject to square root staffing it is possible to achieve high levels of utilization and low probability of wait if the scale of the call center is large.

A number of papers in the literature utilize this framework to categorize their research.

3.3.2 Queuing Models

A substantial amount of research addresses basic queuing models of call centers. Three basic queuing models are examined in the literature, the Erlang C, Erlang B, and Erlang A models. We review these models and the relevant literature briefly.

3.3.2.1 Erlang C Model

The Erlang C model is identical to the M/M/N queue and is widely used in workforce management (WFM) systems (Gans, Koole *et al.* 2003; Mandelbaum and Zeltyn 2004). The Erlang C model assumes a Poisson arrival process with constant rate λ, independent and exponentially distributed service times with mean $1/\mu$, and a pool of n homogeneous (statistically identical) agents. The system is assumed to have an infinite number of trunk lines and an infinite capacity queue so no callers are ever blocked. Furthermore, the model assumes that all callers who enter the queue are eventually served so the model allows for no abandonment. This model yields relatively straightforward formulas for key performance measures.

Following the notation and terminology in (Gans, Koole *et al.* 2003) I define the *offered load* as

$$R_i \equiv \lambda_i / \mu_i = \lambda_i E\left[S_i\right] \tag{3.12}$$

and the *traffic intensity* (aka *utilization*) as

$$\rho_i \equiv \lambda_i / \left(N\mu_i\right) = R_i / N \tag{3.13}$$

Given the no abandonment and steady state assumptions of the Erlang C model, the traffic intensity must be strictly less than one for system stability.

The steady state probability that all N agents are busy is then given by

$$C(N,R_i) \triangleq 1-\left(\sum_{m=0}^{N-1}\frac{R_i^m}{m!}\right)\Bigg/\left(\sum_{m=0}^{N-1}\frac{R_i^m}{m!}+\left(\frac{R_i^m}{N!}\right)\left(\frac{1}{1-R_i/N}\right)\right) \qquad (3.14)$$

This is identical to the fraction of customers that must wait to be served

$$P\{Wait>0\}=C(N,R_i) \qquad (3.15)$$

The "*Poisson Arrivals See Time Averages*" (PASTA) principle developed in (Wolff 1982; Wolff 1989) implies that the conditional delay in queue has an exponential distribution with mean $\left(N\mu_i-\lambda_i\right)^{-1}$. The service level metric can then be expressed as

$$\begin{aligned} TSF &\triangleq P\{Wait\le T\}=1-P\{Wait>0\}\cdot P\{Wait>T\,|\,Wait>0\} \\ &= 1-C(N,R_i)\cdot e^{-N\mu_i(1-\rho_i)T} \end{aligned} \qquad (3.16)$$

The speed to answer metric is also easily defined as

$$\begin{aligned} ASA &\triangleq E\left[Wait\right]=P\{Wait>0\}\cdot E\left[Wait>T\,|\,Wait>0\right] \\ &= C(N,R_i)\cdot\left(\frac{1}{N}\right)\cdot\left(\frac{1}{\mu_i}\right)\cdot\left(\frac{1}{1-\rho_i}\right) \end{aligned} \qquad (3.17)$$

An important paper that analyzes the Erlang C model in the context of the QED regime is (Halfin and Whitt 1981). This paper develops the well known square root staffing principle that sets the number of agents to be

$$N=R+\beta\sqrt{R} \qquad (3.18)$$

where β is a non-negative quantity that specifies the service grade. The importance of the Halfin-Whitt approximation is the notion that for a fixed quality of service, staffing requirements increase with the square root of offered load. This simple principle highlights the inherent economies of scale in call center staffing.

3.3.2.2 Erlang B

Erlang B is a *loss model*, a model that assumes that the number of available trunk lines is exactly equal to the number of agents. In this model callers are either serviced immediately, or blocked from the queue (they face a busy signal). Erlang B, also known as the *Erlang Loss Formula*, is often used to calculate the number of trunk lines required in order to achieve a desired blocking probability. The Erlang Loss Formula is presented in (Hall 1991) as

$$P(\text{lost customer}) = \frac{r^m / m!}{\sum_{i=1}^{m} r^i / i!} \tag{3.19}$$

Although originally derived by Erlang under the assumption of exponential service time, equation (3.19) was later shown to apply to any talk time distribution.

Given that telecommunication costs are relatively low, most call center staffing models will assume sufficient trunk capacity so that no calls are blocked and the loss function is not used. The Erlang Loss function is sometimes used in other types of queuing models where blocking is important, for example trauma centers. See (Cachon and Terwiesch 2006) for some examples.

3.3.2.3 Erlang A

The relative simplicity of the Erlang C is a directly result of the rather strong assumption of zero abandonment. As we showed in section two, abandonment rates in support oriented call centers are non-trivial and ignoring abandonment can introduce significant error into the model. Staffing models that ignore abandonment will tend to overstaff the call center for a targeted level of system performance.

The Erlang A model allows callers who enter the queue to abandon the queue if their wait exceeds their patience. Specifically, the model assumes that each caller has an exponentially distributed patience with a mean of $1/\theta$. A caller presented with a wait time in excess of their patience hangs up rather then waiting in queue.

Several papers address the Erlang A model. (Gans, Koole *et al.* 2003) discuss the model briefly, while a more complete overview of the model is provided in (Mandelbaum and Zeltyn 2004)[25]. The issue of parameter estimation and sensitivity is addressed in (Whitt 2006a). A thorough comparison of the Erlang A and Erlang C models is provided in (Garnett, Mandelbaum *et al.* 2002). This paper develops steady state staffing heuristics for each staffing regime that are analogous to the Halfin-Whitt square root staffing heuristic for the Erlang C model. This paper also presents diffusion approximations for key performance metrics in the Erlang A model.

Formulas for the Erlang A are significantly more complicated then those for the Erlang C model. Steady state probabilities for the distribution of the number of callers in the system were provided in Palm's paper and are reproduced in an appendix to (Mandelbaum and Zeltyn 2004). (An abbreviated exposition is provided in section 5.5 of (Riordan 1962).)

The steady state distribution for the Erlang-A model is given by

$$\pi_j = \begin{cases} \pi_n \cdot \dfrac{N!}{j!R_i^{N-j}}, & 0 \le j \le N \\ \\ \pi_n \cdot \dfrac{(\lambda_i / \theta_i)^{j-n}}{\prod_{k=1}^{j-n}\left(\dfrac{N\mu}{\theta} + k\right)}, & j \ge N+1 \end{cases} \tag{3.20}$$

where

$$\pi_n = \frac{E_{1,n}}{1 + \left[A\left(\dfrac{N\mu}{\theta}, \dfrac{\lambda}{\theta}\right) - 1 \right] E_{1,n}} \tag{3.21}$$

$$A(x,y) \triangleq \frac{xe^y}{y^x}\gamma(x,y) \tag{3.22}$$

and

$$\gamma(x,y) \triangleq \int_o^y t^{x-1}e^{-t}dt, \;\; x > 0, y \ge 0 \tag{3.23}$$

[25] The model was originally introduced by Palm in a paper written in 1946 and published in Swedish. The paper is based on pioneering work done by Agner Krarup Erlang at the Copenhagen Telephone Exchange.

is the incomplete Gamma function. The expression $E_{1,n}$ represents the blocking probability from the Erlang B model.

$$E_{1,n} = \frac{R_i^m / N!}{\sum_{j=0}^{m} R_i^j / j!} \tag{3.24}$$

The probability a caller has to wait is given by

$$P\{Wait > 0\} = \frac{A\left(\frac{N\mu}{\theta}, \frac{\lambda}{\theta}\right) E_{1,n}}{1 + \left[A\left(\frac{N\mu}{\theta}, \frac{\lambda}{\theta}\right) - 1 \right] E_{1,n}} \tag{3.25}$$

The expected wait time for a queued customer is

$$E[W \mid W > 0] = \frac{1}{\theta} \left[\frac{1}{\rho A\left(\frac{N\mu}{\theta}, \frac{\lambda}{\theta}\right)} + 1 - \frac{1}{\rho} \right] \tag{3.26}$$

Numerical evaluation of these expressions is difficult. Useful approximations based on diffusion limits are developed in (Garnett, Mandelbaum *et al.* 2002). I summarize these calculations below.

Let $\phi(x)$ represent the standard normal density function and $\Phi(x)$ the cumulative distribution.

$$\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \tag{3.27}$$

$$\Phi(x) = \int_{-\infty}^{x} \phi(y) dy \tag{3.28}$$

The hazard rate function is then defined as

$$h(x) = \frac{\phi(x)}{1 - \Phi(x)} = \frac{\phi(x)}{\Phi(-x)} \tag{3.29}$$

Now define the following 2 expressions

$$w(x,y)=\left[1+\frac{h(-xy)}{yh(x)}\right]^{-1} \tag{3.30}$$

$$\Psi(x,y)=\frac{\phi(x)}{1-\Phi(x+y)} \tag{3.31}$$

Also define the service grade β as

$$\beta=\frac{N-R_i}{\sqrt{R_i}} \tag{3.32}$$

The (Garnett, Mandelbaum *et al.* 2002) paper goes on to show how approximations for key performance metrics can be derived from these functions. In particular we have the following expressions.

The probability a caller must wait is:

$$P\left\{W>0\right\}\approx w\left(-\beta,\sqrt{\mu/\theta}\right) \tag{3.33}$$

The probability that the wait is greater then T is:

$$P\left\{W>T\right\}\approx w\left(-\beta,\sqrt{\frac{\mu}{\theta}}\right)\cdot\frac{h\left(\beta\sqrt{\mu/\theta}\right)}{\Psi\left(\beta\sqrt{\mu/\theta},\sqrt{N\mu\theta t}\right)}\cdot e^{-\theta t} \tag{3.34}$$

The probability of abandonment is:

$$P\left\{Ab\right\}\approx\left[1-\frac{h\left(\beta\sqrt{\mu/\theta}\right)}{h\left(\beta\sqrt{\mu/\theta}+\sqrt{\theta/\left(N\mu\right)}\right)}\right]\cdot w\left(-\beta,\sqrt{\frac{\mu}{\theta}}\right) \tag{3.35}$$

From these expressions we can write a formula for our key metric the Telephone Service Factor (TSF). TSF is defined in practice as the proportion of all incoming calls that are serviced within T seconds.

$$TSF\triangleq P\left\{Wait\le T\mid Sr\right\}=\left(1-P\left\{Ab\right\}\right)\cdot E\left\{Wait>T\right\} \tag{3.36}$$

While these expressions are rather complicated, they are relatively straightforward to code and are used in numerical calculations developed in this dissertation.

3.3.2.4 Server Sizing Models

A number of call center papers address what I call the server sizing problem; finding the minimum number of servers (agents) required to meet a defined service objective given specified arrival and service characteristics. Note that these models typically ignore all shift constraints and implicitly assume a two stage process where a set covering approach is used in the second stage to satisfy the server requirements defined in the first stage.

The (Halfin and Whitt 1981) paper develops server sizing heuristics for the Erlang C model based on the square root staffing principle. Other papers that build on this analysis and develop staffing models for inbound centers are (Borst, Mandelbaum et al. 2004) and (Jennings and Mandelbaum 1996). A more general queuing model without abandonment is examined in (Whitt 1989). Server sizing when abandonment is allowed is addressed in (Garnett, Mandelbaum *et al.* 2002). (Whitt 2006a) examines the sensitivity of the Erlang A model to parameter misspecification. (Whitt 2006b) addresses server sizing when arrival and absenteeism rates are uncertain. (Koole and van der Sluis 2003) attempt to develop a staffing model that optimizes a global objective, i.e. an average performance metric over a longer time period.

3.3.2.5 Nonstationary Models

Basic queuing models are built on the assumption of long term steady state behavior with stationary arrival and service processes. As shown in section two, this is often not the case in practice. In (Robbins, Medeiros *et al.* 2006) we use simulation to show that arrival rate uncertainty can cause significant deviation from targeted call center performance metrics. The issue with modeling time varying arrival rates is the system will never achieve steady state. We seek conditions and models that are reasonable approximations and the system can then be said to be in *quasi-steady state* (Hall 1991). There are several relatively simple approximations that can be considered with time varying arrival rates (Jennings, Mandelbaum *et al.* 1996).

- Simple Stationary Approximation (SSA) – uses the stationary model with the long-run average arrival rate.
- Pointwise Stationary Approximation (PSA) - uses the stationary model with instantaneous arrival rate *t*.

- Stationary Independent Period by Period (SIPP) – applies the stationary model to discrete planning periods, often 15 or 30 minute segments of the day.

The SSA model is a very simple approximation that is useful when arrival rates vary rapidly relative to service time, but is nearly constant over the long run, (i.e. stochastic variability around a steady state level.)

3.3.2.5.1 PSA

The PSA model is useful in models with arrival rates that vary slowly as compared to service levels but can be expressed (or approximated) as a function of time, for example a sinusoidal varying arrival rate. The PSA model calculates a stationary approximation at each point in time and integrates to calculate the average. A single server example is presented in section 6.3 of (Hall 1991), a more detailed analysis is presented in (Green and Kolesar 1991). In this paper the authors assume that arrival rates vary periodically and therefore the system will develop a periodic steady state behavior. Consider the average queue length L_q given by

$$L_q = \frac{1}{T}\int_0^T \left(\sum_{n=s}^{\infty} (n-s) p_n(t) dt \right) \tag{3.37}$$

where s is the number of servers.

Now define the pointwise stationary approximation as

$$L_q^{\infty} = \frac{1}{T}\int_0^T L_q(\lambda(t))dt \tag{3.38}$$

where $L_q(\lambda(t))$ is the queue length in a stationary M/M/s queue with arrival rate $\lambda(t)$. The authors assume an arrival rate that can be expressed as

$$\lambda(t) = \lambda + A\cos\left(2\pi t / 24\right) \tag{3.39}$$

They solve these equations numerically and further restrict themselves to the case where traffic intensity is less then 1 and show that under these conditions the PSA is an upper bound on the actual performance.

(Green and Kolesar 1997) also examines a variant of the PSA model that the authors call the Lagged PSA model. The lagged PSA is motivated by the fact that the actual peak congestion will occur at a point later then predicted by the PSA. By considering only the current traffic when calculating congestion, the model ignores the fact that a time delay exists between the point of peak traffic and peak congestion. In this paper the authors develop a method for estimating the delay in the peak, and recalculating congestion at the adjusted peak.

3.3.2.5.2 SIPP

The PSA and lagged PSA methods create accurate estimations of nonstationary system performance, given the assumptions of no abandonment, utilization less than one, and sinusoidal arrival rates. Insight gleaned from these models can be then used to foster intuition and generate heuristics for planning purposes. However, in practice a more commonly implemented approach is the Stationary Independent Period by Period (SIPP) approach. In this approach the day is divided into buckets; often 15 or 30 minute periods, and an average arrival rate is calculated for each period. Staffing plans are then based on the assumption that steady state is achieved in each of those periods, typically using an Erlang model[26].

The accuracy of this method is analyzed in (Green, Kolesar *et al.* 2001). In this paper the authors again assume a model of sinusoidal varying arrival rates with no abandonment. The authors perform detailed numerical analysis and show that SIPP can lead to poor approximations in cases where the relative amplitude of the sine wave is large, planning periods are long, service rates are low, or the system is large.

The authors propose several variants of the SIPP method to reduce errors.

- SIPP Max: implements SIPP using the maximum arrival rate for the period, rather then the average. SIPP Max is more conservative then SIPP and clearly leads to higher proposed staffing levels.
- SIPP Mix: implements average arrival rate for periods where arrival rate is increasing and the Max for periods where it is decreasing.

The paper also analyzes each of these approaches with a Lag option, applying a lagged estimate similar to that developed in their previous paper, creating the Lag Avg, Lag Max, and Lag Mix

[26] Erlang C models are often used in practice, in spite of the no abandonment assumption of that model.

methods. No approach is found to be superior in all circumstances and some general heuristic are developed.

The Lagged SIPP approach is further analyzed in (Green, Kolesar *et al.* 2003). This paper maintains the assumption of a sinusoidal varying arrival rate as in (3.39) but now assumes that the call center operates on less then a 24 hour clock; therefore it starts and ends empty. The paper shows that modifications to the SIPP approach consistently achieve the service level target with only modest increases in staffing.

3.3.2.5.3 Other Approaches to Time Varying Arrivals

Several other papers propose variations or combinations of these models. (Feldman, Mandelbaum *et al.* 2005) evaluate the applicability of PSA under conditions of predictable variability and stochastic variability, but do not address the issue of uncertainty. They develop a Simulation-Based Iterative Staffing Algorithm (ISA). (Whitt 2006b) considers both arrival rate uncertainty and staff absenteeism to address the server sizing problem. (Jennings, Mandelbaum *et al.* 1996) looks for a compromise between the PSA and ISA model based on infinite server approximations. This paper examines the sinusoidal case as well as start up case, volume ramping up from zero at the start of business.

Time varying arrivals are examined in a more general case in a working paper (Green, Kolesar *et al.* 2005). The model in this paper allows for more general arrival patterns and allows for abandonment, but again this model only addresses the staffing requirements (server sizing) decision. The *scheduling* decision is solved separately, presumably by solving an integer program as specified in Dantzig's model.

3.3.2.6 Skills Based Routing

In many models agents are assumed to be equally skilled and statistically identical, but in some cases this assumption can not be made. Consider a case where the call center must support multiple languages. Typically not all agents are able to speak each of the supported languages, but many agents may speak multiple languages. The problem of routing calls to the appropriately skilled agents is often referred to as *skills based routing*. A high level review of skills based routing literature is provided in section 5.1 of (Gans, Koole *et al.* 2003). The problem is examined in more detail in (Koole and Pot 2005). (Gans and Zhou 2007) examine a series of routing schemes for call center outsourcing.

A relevant paper that examines staffing under a skills based routing concept is (Wallace and Whitt 2005). (I'll refer to this paper as W&W) In the W&W model there are 6 call types and every agent is trained to handle a fixed number of those types. The authors use a simulation based optimization model to find the ideal cross training level. The paper's key insight is that a low level of cross training provides "most" of the benefit. Specifically, they find that training every agent in 2 skills provides the bulk of the benefit, while additional training has a relatively low payoff. W&W show that adding a second skill gives most of the value, but they don't analyze the cost associated with cross training. Additionally, W&W examine cross training only in steady state, where arrival rates and staff levels are fixed. At a detailed level the W&W model ignores abandonment, an important consideration in our situation.

3.3.2.7 Empirical Analysis

In addition to the large body of theoretical/analytic models addressing call centers, I know of two papers that explicitly analyze the statistical data generated in a call center. (Mandelbaum A., Sakov A. *et al.* 2001) and (Brown, Gans *et al.* 2002; Brown, Gans *et al.* 2005) are papers that provide a statistical analysis of the same set of call center data gathered from a bank's call center[27]. Among other things they find is that call time has a lognormal distribution. In contrast, most analytical work makes the simplifying assumption that talk time is exponential. The analysis also examines abandonment rate. The authors find that that hazard rate for abandonment (the time phased probability for abandoning) is bi-modal. A large peak occurs a few seconds after the caller realizes they must wait, while a second peak occurs 60 seconds later after a *please wait* message is played. The analysis shows that in the tail the hazard rate become approximately constant, supporting the concept of exponential patience for those willing to wait at least a moderate time.

[27] The earlier Brown, Gans, *et.al.* paper is a working paper significantly more detailed then the final version published in JASA.

(Avramidis, Deslauriers *et al.* 2004) develop a series of stochastic models to generate simulated call volumes based on empirical analysis of call center data. They develop a set of models that factor in the correlation structure of intraday arrivals.

3.3.2.8 Practice Papers

In addition, a few papers have been written on the call center operations in practice. A series of papers discuss call center operations at L. L. Bean. (Andrews and Parsons 1989; Quinn, Andrews *et al.* 1991; Andrews and Parsons 1993; Andrews and Cunningham 1995). The authors discuss issues related to forecasting, resource allocation and scheduling. The theme that runs through these papers is the challenge related to determining the appropriate number of agents to staff given the trade off of operational costs and customer service. A practice paper that highlights the use of simulation in call center planning is (Saltzman and Mehrotra 2001). In this paper the authors document the use of a call center simulation model to help a software company determine approximate staff level requirements for a new service offering.

3.4 Stochastic Optimization

3.4.1 Overview

Math programs which explicitly incorporate variability in parameter values are known as *stochastic programs*. The notion of stochastic programming was first introduced in the 1950s (Beale 1955; Dantzig 1955). Although the concept of stochastic programming has been in the literature for over 60 years, the application of stochastic programming has been relatively rare. This is due in large part to the computational challenges associated with stochastic programming.

At the most basic level stochastic programs come in two varieties, *chance constrained programs* and *recourse programs*. Chance constrained programs implement a confidence level type constraint; for example specifying a positive inventory position with 95% confidence. Recourse programs on the other hand recognize two types of decisions; decisions that occur before uncertainty is revealed – stage one decisions, and decisions that occur after uncertainty has been revealed – recourse decisions. In this review we restrict our analysis to recourse problems. A comprehensive review of chance constrained programs is provided in (Prekopa 1995).

3.4.2 Two Stage Stochastic Recourse Problems

Recourse problems have been widely analyzed in the literature. A brief tutorial type introduction is provided in (Higle 2005). Several excellent texts are also available that outline the structure and solution approaches for stochastic programming. (Kall and Wallace 1994) is an excellent introduction that includes a survey of various solution techniques and algorithms. (Birge and Louveaux 1997) is a thorough review of linear and non-linear stochastic programming, while (Kall and Mayer 2005) focuses strictly on stochastic linear programs.

Adopting the notation from (Birge and Louveaux 1997), the general stochastic linear programming problem can be expressed as

$$\min \boldsymbol{c}^T \boldsymbol{x} + E_\xi \left[\min \boldsymbol{q}(\omega)^T \boldsymbol{y}(\omega) \right] \tag{3.40}$$

$$s.t.\ \ A\boldsymbol{x} = \boldsymbol{b} \tag{3.41}$$

$$T(\omega)\boldsymbol{x} + W\boldsymbol{y}(\omega) = \boldsymbol{h}(\omega) \tag{3.42}$$

$$\boldsymbol{x} \geq 0, \boldsymbol{y}(\omega) \geq 0 \tag{3.43}$$

The objective of the stochastic linear program (3.40) is to minimize the cost of the stage one decision, plus the expected cost of the stage two decisions. The optimization is constrained by a set of constraints (3.41) that depend only on the deterministic stage one variables, and a set of constraints (3.42) that depend on the recourse decisions ($y(\omega)$) and may have random components. The stochastic program is typically solved relative to a finite set of scenarios, sample draws of the random vector ξ. If the number of sample outcomes is denoted as K, then we can write the stochastic program in extensive form as

$$\min \quad c^T x + \sum_{k=1}^{K} p_k q_k^T y_k \tag{3.44}$$

$$s.t. \quad Ax = b \tag{3.45}$$

$$T_k x + W y_k = h_k, \quad k = 1,...,K \tag{3.46}$$

$$x \geq 0, \quad y_k \geq 0, \quad k = 1,...,K \tag{3.47}$$

The extensive form program (3.44) - (3.47) is the deterministic equivalent of (3.40) - (3.43) with a finite set of outcomes, and as such can be written as a large linear program. The program can then be solved using the standard simplex algorithm for linear programs. However, as the number of realizations increases the size of the program can be quite large and difficult to solve.

The L-Shaped decomposition algorithm is a commonly used method for solving large scale stochastic programs. The algorithm is discussed in detail in (Birge and Louveaux 1997) as well as (Kall 1976; Kall and Wallace 1994; Kall and Mayer 2005). The L-Shaped method is a variation of the Dantzig-Wolfe or Bender's decomposition methods. (For details on these approaches see (Lasdon 2002)). In the L-Shaped method a sub problem is solved for each scenario, where a scenario is a sample realization of the problem's random vector. Using the dual variables from the solution of the subproblems a piece wise linear approximation of the recourse function can be generated. The L-Shaped algorithm proceeds iteratively, adding cuts in each major iteration. The standard L-Shaped method adds one cut per iteration, while the Multicut L-Shaped method adds one cut per scenario in each iteration (Birge and Louveaux 1997).

An alternative approach is the Stochastic Decomposition (SD) method. Whereas the L-Shaped approach begins with a fixed set of scenarios, SD uses a variable number of scenarios. The SD algorithm adds new scenarios; creating new cuts and updating existing cuts until some stopping criteria is reached. The SD approach is described in detail in (Higle and Sen 1996).

3.4.3 The Benefit of Stochastic Programming

Although they are difficult to solve, stochastic programs create models that are more realistic representations of the phenomenon under study. (Rarely do we know the precise value of important parameters such as demand.) In addition, the solution to the stochastic program is in generally quantitatively superior to the alternative mean valued solution; the solution to the LP when variable parameters are represented by their means.

The notion of the stochastic program was first developed in the mid 1950s (Beale 1955; Dantzig 1955). These papers develop some important properties of stochastic programs that are useful for proving results related to the value of information. I restate several properties here without proofs. To simplify notation I define

$$z(x,\xi) = c^T x + \min[q(\omega)y(\omega) \mid T(\omega) + Wy(\omega) = h(\omega), y(\omega) \geq 0] \qquad (3.48)$$

I then restate the stochastic program as follows:

$$Min\ z(x,\xi) \tag{3.49}$$

$$s.t.\ A\boldsymbol{x} = \boldsymbol{b} \tag{3.50}$$

$$\boldsymbol{x} \geq 0 \tag{3.51}$$

The first result is that $E[z(x,\omega)]$ is a convex function of x. The second result states that for a given x, z is a convex function of ω.

A summary of the research on the value of information is stochastic programming is provided in (Birge and Louveaux 1997). Early work on the topic is developed in (Madansky 1960)[28]. Mandansky describes the stochastic recourse problem as the *here and now* problem, indicating the problem that must be solved before uncertainty can be resolved. He contrasts that to the *wait and see* problem, the problem that would result if we were somehow able to wait for uncertainty to be resolved before making our stage one decision. The here and now/recourse problem (RP) can be stated mathematically as follows:

$$RP = \min_x E_\xi z(x,\xi) \tag{3.52}$$

The problem is to find the set of decisions that minimizes the expectation of objective function (3.48). Alternatively, the wait and see problem is expressed as

$$WS = E_\xi \left[\min_x z(x,\xi) \right] \tag{3.53}$$

The wait and see solution is the expectation, taken over all realizations of ξ, of the optimal decision x for each realization of ξ. In other words, the average outcome if we were able to observe ω before making the stage one decisions.

As we discussed previously, the recourse problem (3.52) is often difficult to solve so a natural approximation is found by replacing the random vector ξ, by its expectation $\bar{\xi}$

$$EV = \min_x z(x,\bar{\xi}) \tag{3.54}$$

[28] I will attempt to trace out the development of the subject across multiple papers, but for consistency will rely primarily on the notational conventions from Birge and Louveaux (1997).

We call the solution to this problem the *mean value solution*, and denote it as $\bar{x}(\bar{\xi})$. I have argued that this approximation may not always provide a good answer to the underlying stochastic problem. To quantify this argument I introduce a quantity that measures result of using the mean value solution. We denote the expected result of using the EV solution as

$$EEV = E\left[z(\bar{x}(\bar{\xi}), \xi) \right] \tag{3.55}$$

The EEV represents the expected result of applying the mean value solution in stage one, then allowing the decision maker to make optimal recourse decisions after uncertainty has been revealed. As such it represents the average payoff that would occur if decisions are based on the solution of (3.54).

Mandansky shows that since z is a continuous, convex function, a straightforward application of Jensen's inequality gives the following inequality[29]

$$EEV \geq RP \geq WS \geq EV \tag{3.56}$$

Equation (3.56) provides the foundation for much of what follows so it is worth a close examination. Assume a decision maker is faced with some stochastic optimization problem which he solves using the (standard) approach of replacing random variables with their expectation. He solves the problem and estimates a minimal cost objective of EV. We say that that the mean value solution is *biased*, in that it is less then (or equal to) the expected outcome that would occur from implementing that decision. The *Mean Value Bias* is defined as

$$MVB = EEV - EV \tag{3.57}$$

In other words the expected outcome of applying this solution is at least as great as the mean value problem predicts but may be larger. We see that solving the recourse problem is no more biased than the mean value problem. In other words the solution found by solving the problem can be in the worst case as biased as the mean value problem, but no more so. On the other hand it may be less biased.

The solution WS represents an important benchmark as it tells us the objective value we would expect if we could resolve uncertainty before making our decision. Equation (3.56) tells us this value is bounded between the solutions to the recourse and mean value

[29] This is equation (8) in Birge (1982) stated using his notation. The original statement is in Mandansky as an unnumbered theorem, using expectation notation.

problems $RP \geq WS \geq EV$. The expected value problem gives us an estimate at least as low, and perhaps lower, then what we could achieve even if we can perfectly predict the random outcome. The recourse problem on the other hand is no better then what we can achieve with knowledge of the random outcome. In this sense the recourse problem is a more conservative estimate, and likely a more realistic estimate.

We now consider the concept of the *Expected Value of Perfect Information* (EVPI). The notion of EVPI in stochastic programs is developed in (Avriel and Williams 1970). EVPI tells us how much better our decision making would be if we had perfect insight into the outcome of the uncertainty in our decision making problem, or in Madansky's terminology the improvement we could receive if somehow we could make the wait and see decision instead of the here and now decision. We can therefore define EVPI, again using the notation from (Birge and Louveaux 1997) as

$$EVPI = RP - WS \tag{3.58}$$

Avriel and Williams develop bounds on EVPI in stochastic programs. Their theorem 1 provides a lower bound

$$EVPI \geq 0 \tag{3.59}$$

and follows directly from the definition and from (3.56). The result is quite intuitive, we do worse (in expected terms) if we have perfect knowledge of how uncertainty will be resolved. An upper bound is also easily calculated as:

$$EVPI \leq EV - RP \tag{3.60}$$

The upper bound tells us that the difference between the expected value problem and recourse problem is an upper bound on the value we can achieve from perfect information about the problem's uncertainty. Avriel and Williams develop a more general form of (3.60) that can be calculated for any realization of ξ, but prove that the choice of $\overline{\xi}$ provides the tightest bound. They also apply these bounds to a more general class of stochastic programs that include quadratic recourse functions. A tighter set of bounds are developed in (Huang, Vertinsky *et al.* 1977).

The notion of EVPI information concerns that value that accrues to a decision maker if uncertainty can be resolved prior to decision making. A related concept is the value that accrues to the decision maker by explicitly recognizing parameter uncertainty when making a decision. In our context this relates to the value of using the more complicated stochastic program.

3.4.3.1 Value of the Stochastic Solution

The concept of quantifying the value of using a stochastic programming formulation, as opposed to a mean value formulation, was first developed (Birge 1982) and later addressed in (Birge and Louveaux 1997).

Birge (1982) introduces the key concept used to measure the benefit of using stochastic programming, the Value of the Stochastic Solution (VSS) which he defined as

$$VSS = EEV - RP \qquad (3.61)$$

The VSS is a simple measure of the improvement in the expected objective that arises from using a recourse formulation. Recall that the term EEV is the expected result of using the mean value solution while RP is the expected outcome of using a stochastic formulation. A simple lower bound on VSS is easily derived from (3.56)

$$VSS \geq 0 \qquad (3.62)$$

This simple result is quite important; it tells us we can not do any worse by explicitly considering variability when developing our solution. However, since we know the cost of computing the stochastic solution is high, we want to know when we can do better, and by how much.

Birge presents a straightforward upper bound that applies to both VSS and EVPI for stochastic programs with fixed recourse and fixed objective coefficients

$$EVPI \leq EEV - EV \qquad (3.63)$$

$$VSS \leq EEV - EV \qquad (3.64)$$

The right hand side of these equations represents the bias of the mean value solution; that is the degree to which the mean value solution overstates the actual expectation of the outcome. Equation (3.63) tells us that this bias is an upper limit on how much better we can do with perfect information, while equation (3.64) tells us this bias is also a limit on how much better we can do using a stochastic formulation. If the mean value solution is unbiased ($EEV = EV$), then we receive no benefit from perfect information or from stochastic programming. This will occur for example, if the optimal solution to the problem is independent of the model's uncertainty. For example, if the distribution of crop yields in the farmer's problem is such that the same allocation is optimal for all realizations of yield, then the farmer receives no value from perfect information and similarly no value from stochastic programming. The clear implication is that when considering stochastic programming, we can restrict ourselves to considering the variability of parameters whose realization has a material effect on the decision vector.

While $EVPI$ and VSS have the same upper and lower bounds, it is important to note that they are not equivalent. In general we would expect that $EVPI$ will be greater than VSS, this is not always the case. Birge (1982) presents an illustration of a problem where perfect information has nonzero value, but the value of the stochastic solution is zero[30]. This result is not surprising, it tells us that there are problems where knowing the true outcome of uncertainty is valuable, but explicit modeling of the variability does not help us. The more surprising result is a similar example that shows $EVPI = 0$ while $VSS > 0$. The particular example is a case where the problem has multiple optimal solutions. The argument being that the linear program solver may return a solution that is optimal for the mean value problem, but sub-optimal to the recourse problem. The alternative solution is optimal for both the mean value and recourse problem. Since we find the right optimum in the recourse problem, but not the mean value problem, the VSS is positive. However, knowing the outcome with certainty yields the same answer as the recourse program so the value of prefect information is zero. Birge makes that the claim that "because linear programs often include multiple optimal solutions, this type of solution is far from exceptional.[31]"

[30] The same example is present in Birge (1982) Appendix A and Birge and Louveaux (1997) pp. 142-144

[31] Birge and Louveaux (1997) p. 143

The following diagram summarizes the relationship between the various quantities we have discussed.

Relative Solutions for a Minimization Problem Under Uncertainty

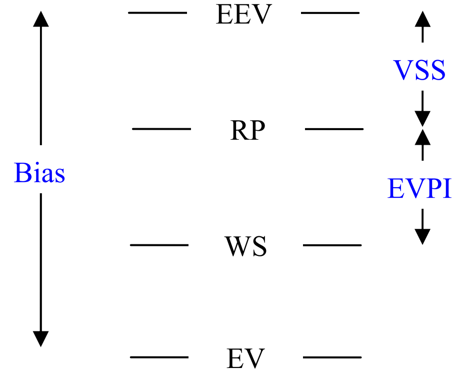


Figure 3-1

The expected value solution (EV) yields the best predicted result, while the actual expected value of implementing the expected value solution (EEV) is the worst result. The difference between these two quantities is the bias of the mean value solution. The recourse problem (RP) provides the best expected result that can actually implemented. The improvement from the using the recourse formulation relative to the expected outcome of the expected value solution is the Value of the Stochastic Solution (VSS). The additional benefit that can be achieved by resolving uncertainty prior to decision making is the Expected Value of Perfect Information (EVPI), the difference between RP and the Wait and See solution (WSS).

This graphic illustrates what I'll call the paradox of the mean value problem; *the mean value problem gives the <u>best</u> objective value, but yields the <u>worst</u> expected result.* We are faced with making a recommendation to implement a decision model that yields a worse objective value, but a better expected result. This occurs because the objective of the recourse problem is an expected outcome; the objective of the mean value problem is not.

3.4.4 Monte Carlo Methods

For most problems it is not practical to solve the stochastic optimization problem against the complete set of possible realizations of the random vector and the problem is typically solved for some sample of possible outcomes. Since we are solving against a sample of possible outcomes, the calculated optimal solution is a statistical estimate of the true solution, subject to statistical sampling error. A growing body of literature addresses the statistical properties of sampled stochastic programs under the general category of Monte Carlo Methods. An overview of Monte Carlo methods is provided in (Birge and Louveaux 1997) Chapter 10. Monte Carlo methods for stochastic programming borrow many concepts from the simulation literature and the interface between these two methodologies is discussed in (Pflug 1996).

An important consideration in Monte Carlo methods is the distinction between the true problem and the sampled problem. The true problem can be written as[32]:

$$z^* = \min_{\boldsymbol{x}\in X} E\left[f(\boldsymbol{x},\tilde{\xi}) \right] \tag{3.65}$$

$$\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}\in X} E\left[f(\boldsymbol{x},\tilde{\xi}) \right] \tag{3.66}$$

We seek to find the decision vector $\boldsymbol{x}^*$, and the corresponding objective value z^* that minimize the expected value of the objective function $f(\boldsymbol{x},\tilde{\xi})$, where the expectation is taken over the support of the random vector $\tilde{\xi}$. The actual problem we solve is the sample path approximation problem, an optimization over a finite set of samples $\tilde{\xi}^i$. The sample path approximation problem can then be written as:

$$z_n^* = \min_{\boldsymbol{x}\in X} \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x},\tilde{\xi}^i) \tag{3.67}$$

$$\boldsymbol{x}_n^* \in \arg\min_{\boldsymbol{x}\in X} \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x},\tilde{\xi}^i) \tag{3.68}$$

An important issue in stochastic programming is the statistical convergence of the sample path solution solution z_n^* to the true solution z^*. We are interested in how close our approximate solution is to the true solution. Conversely, we may be interested in determining how many samples (scenarios) are required in order to achieve a desired level of confidence. An important

[32] The exact notation varies from paper to paper. Here I adopt a notation used in Mak, Morton, and Wood (1999).

paper on this topic is (Dupacova and Wets 1988). This paper establishes the basic convergence properties of the sampled stochastic program proving that the sample path solution converges to the true solution as the sample size goes to infinity. Other papers on this topic include (Shapiro 1991; King and Rockafeller 1993).

While the sampled problem converges to the true solution, it generates a biased estimated of the true solution for finite samples. (Mak, Morton *et al.* 1999) show that the expected outcome of the sampled problem is optimistically biased and that the bias is decreasing in the number of samples. The convergence properties of the solution vector x are discussed in the most detail in (Shapiro and Homem-de-Mello 2000). The paper shows that in the case of a two stage linear stochastic program with a discrete distribution, the optimal solution to the approximating problem will be exactly equal to the solution of the true problem for a large enough *N*. They also show that the rate of convergence is exponential in the number of samples. An empirical assessment of sampling bias is provided in (Freimer, Thomas *et al.* 2006; Linderoth, Shapiro *et al.* 2006).

A method for developing a confidence interval on the sampled problem is developed in (Mak, Morton *et al.* 1999). The solution to any sampled problem provides a point estimate on the lower bound of a stochastic minimization problem. Assume we choose to solve the approximation several times to improve our estimate. Let $\tilde{\xi}^{i1},....,\tilde{\xi}^{in}, i=1,...,n_\ell$ be a set of n_ℓ different batches of scenarios, each of which has n observations.

Define

$$z_n^{*i} = \min_{\boldsymbol{x}\in X}\left[\frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x},\tilde{\xi}^{ij})\right] \tag{3.69}$$

to be the objective value found by solving the sample path approximation problem against the ith batch of sample scenarios.

We can then define our estimate lower bound as

$$\overline{L}(n_\ell) = \frac{1}{n_\ell}\sum_{i=1}^{n_\ell} z_n^{*i} \tag{3.70}$$

The lower bound is set as the average over the n_ℓ different batches of scenarios solved.

To calculate an upper bound, assume we have a candidate solution $\hat{\boldsymbol{x}}$. This solution might be the result of solving the mean value problem, or it could be the lower bound found by solving the sample approximation problem. We can therefore calculate an upper bound by finding the expected cost of implementing the solution $\hat{\boldsymbol{x}}$.

$$\bar{U}(n) = \frac{1}{n}\sum_{i=1}^{n} f(\hat{\boldsymbol{x}}, \xi^i) \tag{3.71}$$

The solution to the reference problem provides an unbiased estimate of the expected cost of implementing the stage one decision $\hat{\boldsymbol{x}}$.

We can then define an approximate $(1-2\alpha)$ confidence interval on the optimality gap as

$$\left[0, \left[\bar{U}(n_u) - \bar{L}(n_\ell)\right]^+ + \tilde{\varepsilon}_u + \tilde{\varepsilon}_\ell\right] \tag{3.72}$$

where $(\tilde{\varepsilon}_u, \tilde{\varepsilon}_\ell)$ are standard errors estimated for the upper and lower bounds.

These results suggest a general procedure for estimating a set of bounds on the optimal solution to a stochastic optimization problem.

1) Determine the number of batches (n_ℓ) to be solved and the number of scenarios (n) to be used in each batch.
2) Solve each of the (n_ℓ) problems to optimality using any algorithm
3) Calculate a point estimate for the lower bound using (3.70) (average objective value)
4) Calculate the sample variance of lower bound. Use this statistic to estimate the standard error on the lower bound ε_ℓ.
5) Calculate a candidate solution $\hat{\boldsymbol{x}} = n_\ell^{-1}\sum_{i=1}^{n_\ell} \boldsymbol{x}_n^{*i}$ where $\boldsymbol{x}_n^{*i}$ are the solutions to the individual batch problems.
6) Generate a set of n_u scenarios to be used for the upper bound estimate. Note that it often the case that the number of scenario used for the upper bound is much larger

then the number of scenarios used in the lower bound calculation.

7) Calculate an estimate on the upper bound by solving the reference problem with the stage 1 decision fixed at $\hat{\boldsymbol{x}}$. Note that this solution finds the optimal recourse for each scenario given $\hat{\boldsymbol{x}}$ and takes the average. Since the master problem is not optimized we need only solve the subprogram n_u times which can be done relatively fast.
8) The optimal solution to the reference problem is the expected cost of implementing $\hat{\boldsymbol{x}}$ and is our point estimate for the upper bound.
9) Calculate the sample variance of the upper bound and use that to calculate the standard error.
10) Calculate a point estimate of the optimality gap $\left[\bar{U}(n_u)-\bar{L}(n_\ell)\right]^+$.
11) Calculate a confidence interval on the optimality gap using (3.72).

Figure 3-4 Stochastic Bounding Procedure

This bounding technique forms the basis for a solution approach known as Sample Average Approximation which is developed in (Kleywegt, Shapiro *et al.* 2001). The basic idea of SAA, is that instead of solving the problem with a large set of samples we solve the problem multiple times with smaller samples and examine the statistical properties of the resulting solutions. This procedure is analogous to the multiple runs concept in discrete simulation.

The approach is a loose framework that does not specify specific solution algorithms, but rather a general iterative approach. I present a slightly simplified version of the algorithm presented in Kleywegt's section 3.5

1) Choose an initial sample size N and N', a tolerance level ε, and a number of batches M
2) For each batch $m=1,...M$ perform the following
   a) Generate a sample of size N and solve the SAA problem with objective value $\hat{v}_N^M$ and ε optimal solution $\hat{x}_N^M$
   b) Estimate the optimality gap and the variance of the gap estimator
   c) If the optimality gap and the variance of the gap estimator are sufficiently small go to step 4.
3) If the optimality gap or the variance of the estimate is too high increase the sample size and go to step 1

```
4) Choose the best solution $\hat{x}$ among all candidate solutions
   using a screening and selection process.  Stop
```

Figure 3-5 General Sample Average Approximation Procedure

3.4.5 Multistage Stochastic Programs

3.4.5.1 Overview and Formulation

Unlike two stage models, the multistage models allows for a series of decisions evolving over some arbitrary (usually finite) time horizon. A general formulation of the multi stage model is as follows:

$$\min z = c^1 x^1 + E_{\xi^2}\left[\min c^2(\omega)x^2(\omega^2) + ... + E_{\xi^H}\left[\min c^H(\omega)x^H(\omega^H)\right]\right] \tag{3.73}$$

$$s.t. \quad W^1 x^1 = h^1 \tag{3.74}$$

$$T^1(\omega)x^1 + W^2 x^2(\omega^2) = h^2(\omega) \tag{3.75}$$

$$\vdots$$

$$T^{H-1}(\omega)x^{H-1} + W^2 x^2(\omega^H) = h^2(\omega) \tag{3.76}$$

$$x^1 \geq 0, x^t(\omega^t) \geq 0, t = 2, ..., H \tag{3.77}$$

3.4.5.2 Scenario Modeling and Nonanticipativity

A general overview of scenario trees for multistage stochastic programs is given in (Dupačová, Consigli *et al.* 2000). In a multistage stochastic program we the decisions made at each stage, $t = 1, 2, ..., T$ are based on the observed realizations of the random variables made in all proceeding stages, $\omega^{T-1} = \{\omega_1, \omega_2, ...\omega_{T-1}\}$. A common way to represent these realizations is via a scenario tree; an oriented graph that begins with a single root node at level 0, and branches into a series of nodes at level 1, each node corresponding to a possible realization of ω in period one. The tree continues to branch up to the nodes at level T. Each node in the tree has a single predecessor and a finite number of descendants corresponding to the possible realization of the random vector at that stage. If the scenario tree is constructed such that the number of descendants is identical for each non-leaf node, then the tree is said to be *balanced*.

An example of an unbalance tree is shown in the following figure

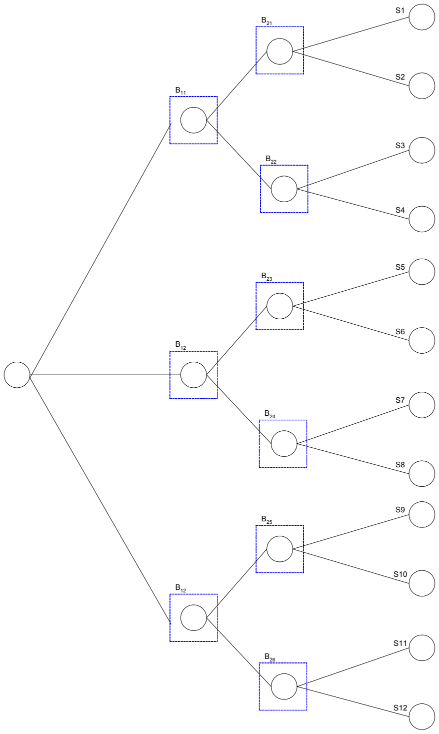


Figure 3-6 Multistage Scenario Tree

In this example we have three stages, with three realizations at the first stage, and two realizations in the subsequent stages, yielding a total of 12 scenarios.

An important consideration is multistage stochastic programs in *nonanticipativity*. An overview of the nonanticipativity problem is presented in (Higle 2005). Simply stated nonanticipitavity requires that decisions are based only on information available at the current stage of the decision process. (Decisions may not *anticipate* future outcome of the random vector.) The formulation (3.73) -(3.77) did not explicitly define nonanticipitavity constraints, these constraints were implicit; i.e. the assumption was made that nonanticipitativity is enforced by the definition of the scenarios.

Consider the scenario tree in Figure 3-3. Each leaf node in the scenario tree is associated with a single scenario, while earlier nodes include multiple scenarios. We refer to all the collection of scenarios that pass through a particular node as a *bundle*. In Figure 3-3 the bundles are indicated by squares. Consider the bundle labeled B_{11} which includes scenarios one through four. Nonanticipativity requires that the decision made at bundle B_{11} is the same for all scenarios that pass through that bundle.

In general let N define the set of all nodes, and let $B(n)$ be the set of be the set of scenarios for each $n \in N$. Formally, nonanticipativity requires that for each node there exists an x_n such that

$$x_{(n)\omega} - x_n = 0 \quad \forall \omega \in B(n) \tag{3.78}$$

Or in words, at intermediate nodes all decision variables must take the same value for each scenario that passes through that node. The multistage stochastic program is often written with implicit nonanticipativity constraints, as in (3.73) - (3.77), but nonanticipativity becomes an important consideration when trying to formulate and solve problem instances.

3.4.5.3 Scenario Growth and Generation

A significant problem in multistage stochastic programming is the rapid growth in the size of the scenario tree. For a problem with T stages, with R_t realizations at each node in stage, the total number of scenarios is

$$N = \prod_{t=1}^{T} R_t \tag{3.79}$$

In a balanced tree with R realizations per stage, the number of scenarios is

$$N = R^T \tag{3.80}$$

The number of scenarios in a multistage problem can easily grow quite large if either the number of stages or the number of realizations becomes large. Since the size of a stochastic linear program grows non-linearly with the number of scenarios, a great deal of attention has been placed on efficient scenario modeling.

Techniques for efficient scenario generation are provided in (Dupačová, Consigli *et al.* 2000; Hoyland and Wallace 2001; Pflug 2001). An approach for selecting a subset of scenarios is presented in (Dupačová, Gröwe-Kuska *et al.* 2003). A method based on *Importance Sampling* is presented in (Dantzig and Infanger 1993). Importance Sampling is based on the idea that that certain values of the random vector have more impact on the parameter being estimated than others. If the sampling approach is modified so that these "important" values sampled more frequently, then the estimator variance can be reduced (Wikipedia 2007).

A post optimality approach, in which the problem is solved for an initial set of scenarios, and then test for sensitivity to out of sample scenarios is presented in (Dupačová 1995). This method supports the development of bounds on the optimal solution of the problem.

3.4.5.4 Solution Algorithms

Solving multistage problems is much more difficult than two stage problems. (Higle 2005) provides a brief summary, and a few algorithms are reviewed in (Birge and Louveaux 1997). Most methods are based on some form of decomposition. Nested Decomposition (Birge 1985) and MSLip (Gassman 1990) decompose the problem by decision stage. An alternative strategy decomposes the problem by scenario. The Progressive Hedging algorithm (Rockafellar and Wets 1991) relaxes the nonanticipativity constraints and applies a Lagrangian relaxation based algorithm. (Higle and Sen 2006) review the duality properties of multistage problems and (Higle, Rayco et al. 2004) applies the stochastic decomposition method to the dual of the multistage problem where the variables correspond to multipliers associated with the nonanticipativity constraints of the primal problem.

3.4.6 Simulation Based Optimization

An alternative approach to stochastic optimization involves the use of discrete event simulation. Overviews of simulation based optimization is presented in Chapter 12 of (Law 2007) and in (Fu 2002). Simulation Based Optimization (SBO) has the same goal as stochastic programming, finding the decision vector that optimizes some performance measure of a stochastic system. However SBO uses a very different approach.

At the most general level an optimization algorithm has two basic components; generating candidate solutions, and evaluating candidate solutions. In stochastic programming we assumed the candidate solution could be evaluated via some closed form, typically linear, objective function. In a decomposition algorithm that linear function is evaluated multiple times, once for each scenario and the expected outcome is a weighted combination of those outcomes. To find the next candidate solution in a stochastic program we perform a simplex pivot if solving the extensive form, or by solving the updated master program in a decomposition approach.

In SBO the solution evaluation step is performed by executing a discrete event simulation (DES) model. Using DES allows us to evaluate a very general model of our stochastic system. We may, for example, drop the simplifying assumptions in a queuing model that allowed us to generate analytical expression for system behavior. The literature on DES is vast; popular texts include (Law and Kelton 2000; Banks 2005; Law 2007). In an SBO the bulk of the computational effort is spent on the evaluation step, but from an algorithm design perspective the challenge is developing a method to find the next candidate solution. Because we do not have an analytical model of the objective function, mechanisms that use that function in a deterministic environment (gradient search) are difficult to implement in a stochastic setting. Some methods such as Response Surface Methodology attempt to estimate a model of the objective function and use that in the search process.

More common are mechanisms that treat the objective function (simulation model) as a black box and simply search the feasible space for better solutions. These search methods often employ randomization in the search process. There are a wide range of search methodologies available that are generally classified in the general category of metaheuristics (Fu 2002; Law 2007). Meta heuristics are "solution methods that orchestrate an interaction between local improvement procedures and higher level strategies to create a process capable of escaping from local optima and performing a robust search of a solution space" (Glover and Kochenberger 2003). Comprehensive reviews of various metaheuristics are provided in (Glover and Kochenberger 2003; Burke and Kendall 2005). Metaheuristics have been widely applied in deterministic combinatorial optimization problems (Nemhauser and Wolsey 1988; Papadimitriou and Steiglitz 1998; Wolsey 1998). An introductory review of their application to SBO is provided in (Fu 2002). Search methodologies include genetic algorithms (Reeves 2003; Sastry, Goldberg *et al.* 2005), Tabu search (Gendreau and Potvin 2005), and simulated annealing (Henderson, Jacobson *et al.* 2003; Aarts, Korst *et al.* 2005).

Most metaheuristics implement some form of a neighborhood based search. Given a candidate solution x, the neighborhood $N(x)$ is a set of feasible points that are close in some sense to x.

Formally, for an optimization problem with feasible set X, a neighborhood is a mapping $N : X \to 2^X$ (Papadimitriou and Steiglitz 1998). In a continuous problem where $X = \mathbb{R}^n$ a natural neighborhood is the set of all points within some fixed Euclidean distance of x.

In combinatorial problems the choice of neighborhood is typically problem dependent. A standard example is the Traveling Salesman problem for which we can define the 2-change neighborhood as any tour that can be formed from the current tour by replacing 2 edges. The choice of neighborhood is an important consideration when designing a local search algorithm. A smaller neighborhood can be more thoroughly searched, but makes escaping a local optimum more difficult.

A metaheuristic that attempts to address this is the Variable Neighborhood Search (Hansen and Mladenovic 2001; Hansen and Mladenovic 2005). In a Variable Neighborhood Search (VNS) we define not one, but multiple neighborhoods. In many implementations of VNS, neighborhoods are nested, such that

$$N_1(x) \subset N_2(x) \subset \ldots \subset N_{k_{Max}}(x) \quad \forall x \in X \tag{3.81}$$

The search algorithm begins by searching the closest neighborhood, iterating x each time an improving solution is found. When a neighborhood search fails to find an improving solution the search is expanded to the next largest neighborhood and the search continues. Each time an improving solution is found, the search returns to the smallest neighborhood. VNS has the advantage that the neighborhood is kept small as long as improvements are available, but it becomes large when a local optimum is established, providing the opportunity to search outside the current valley.

VNS is a very general framework into which virtually any search approach can be incorporated. Simulated Annealing, for example, can be easily incorporated into VNS by allowing non-improving changes to be accepted with some time varying probability.

3.5 Design of Statistical Experiments

3.5.1 Overview

Throughout this paper I perform a number of computational experiments and I attempt to use efficient formal experimental designs throughout the analysis. An overview of experimental design is found in many general statistical texts such as (Kutner, Nachtsheim *et al.* 2005). A more thorough analysis of experimental design issues is presented in Box, Hunter and Hunter, often referred to as BH^2. (Box, Hunter *et al.* 1978). BH^2 provides a thorough analysis of full and factorial designs and an introduction to response surface methods. A more detailed discussion of Response Surface Methods (RSM) is provided in (Box and Draper 1987). The application of Experimental Design to computer experiments creates a number of interesting challenges. A detailed analysis is provided in (Santner, Williams *et al.* 2003). The application of full and fractional factorial designs to discrete event simulation models is provided in (Law 2007).

3.5.2 Full and Fractional Factorial Designs

In a statistical experiment we control a set of *factors*, and measure the impact on one or more *responses*. The experimental design specifies a number of distinct factor settings referred to as *design points*. In an experiment subject to statistical error we often conduct multiple *replications* of the experiment at each design point.

A popular approach for designing numerical experiments is the factorial design (Box, Hunter *et al.* 2005; Law 2007). In a standard application of the factorial design each factor is set to one of two levels; a high value (+) and a low value (-). In the DOE literature factors are typically coded to a standardized form. For example, suppose we are interested in some variable ξ_i over the interval $\left[\xi_L, \xi_H\right]$. We can define the coded variable x_i as

$$x_i = \frac{\xi_i - \xi_l}{(\xi_H - \xi_L)/2} \tag{3.82}$$

With this coding the low value corresponds to -1 and the high value corresponds to +1 (Box and Draper 1987).

In a full factorial design a design point is specified at every 2^k possible combination of the experimental factors taken only at their low and high values. The design table for the experiment is often expressed using standardized values of + and – for each factor, so a full factorial design in four factors has eight design points and can be represented by the following design matrix

DP	Factor 1	Factor 2	Factor 3	Response
1	-	-	-	R1
2	+	-	-	R2
3	-	+	-	R3
4	+	+	-	R4
5	-	-	+	R5
6	+	-	+	R6
7	-	+	+	R7
8	+	+	+	R8

Table 3-3 Three Factor Full Factorial Design of Experiments

For illustration purposes the entries in the design matrix are typically presented as a + or -, representing the values ± 1.

In the full factorial design the factors are orthogonal (totally uncorrelated) and it is quite easy to fit a linear model to the *k* factors, the *k-1* first level interactions, and all higher level interactions. (A model with *k* factors has a total of *k-1* interaction terms each with *k-1* terms.) The problem with the full factorial design is that the number of design points becomes quite large as the number of factors increases.

An alternative is the Fractional Factorial Design. This design maintains an orthogonal design with a smaller number of design points by sacrificing the independent estimation of higher level interaction terms. Higher level interactions are said to be *confounded* with single factor responses. Since in many linear models high level interactions are negligible little precision is lost. For example a full factorial design in 8 factors requires 256 separate design points, while a fractional (2^{8-4}) design requires only 16 design points. This particular fractional design can estimate main (single factor) effects and first level interactions, but no higher level interactions.

The ability to discriminate between effects is determined by the *resolution* of the design. The higher the resolution of the design the more finely we can discriminate between different types of effects. Standard notation specifies resolution with roman numerals based on the definitions in the following table[33]:

Resolution	Definition
III	No main effect is confounded with any other main effect, but main effects are confounded with two-way interactions and some two-way interactions may be confounded with each other.
IV	No main effect is confounded with any other main effect or with any two-way interactions, but two way interactions are aliased with each other.
V	No main effect or two-way interaction is confounded with any other main effect or two-way interaction.

Table 3-4 Definitions for Fractional Factorial Resolutions

The general notation for a fractional design is 2^{k-p}, where k is the original number of factors and 2^{k-p} is the total number of design points, or runs, required in the experiment. The first k factors of the design are generated using a standard full factorial design in all k factors. The remaining $k-p$ factors are created by multiplying selected entries in the design matrix (equal to ± 1) to obtain the new columns. The values assigned to these columns are derived based on the defining relationship of the design. For example, a 2^{4-1}_{IV} design has a defining relationship equal to 4 = ± 123, meaning that the fourth columns is found by multiplying the ± 1 values of columns 1,2 and 3 together. This results in the following experimental design

DP	Factor 1	Factor 2	Factor 3	Factor 4	Response
1	-	-	-	+	R1
2	+	-	-	-	R2
3	-	+	-	-	R3
4	+	+	-	+	R4
5	-	-	+	-	R5
6	+	-	+	+	R6
7	-	+	+	+	R7
8	+	+	+	-	R8

Table 3-5 Three Factor Fractional Factorial Design of Experiments

[33] This information is presented in one form or another in most Design of Experiment texts. I've used the notation from Law (2007). See section 12.3 and tables 12.11 and 12.12 on page 638.

Higher level designs can be generated based on careful selection of the defining relationships. The defining relationships used to generate a number of fractional designs for various levels of resolution and design points is presented in Table 3-6, adapted from Law (2007).

An important point to note is that although the $k - p$ columns in a properly designed fractional design are linear combinations of the first k columns, the design remains orthogonal. This means that the input factors are perfectly uncorrelated. Again the orthogonality of the independent variables totally eliminates any issue of multicollineartiy when fitting a regression model. Practically speaking, this means that the values of the regression coefficients found by fitting a least squares model are unaffected by the choice of which factors are included in the model.

	Factors (k)						
Runs	**3**	**4**	**5**	**6**	**7**	**8**	**9**
4	2_{III}^{3-1} 3 = ± 12						
8		2_{IV}^{4-1} 4 = ± 123	2_{III}^{5-2} 4 = ± 12 5 = ± 13	2_{III}^{6-3} 4 = ± 12 5 = ± 13 6 = ± 23	2_{III}^{7-4} 4 = ± 12 6 = ± 13 6 = ± 23 7 = ± 123		
16			2_{V}^{5-1} 4 = ± 1234	2_{IV}^{6-2} 5 = ± 123 6 = ± 234	2_{IV}^{7-3} 5 = ± 123 6 = ± 234 7 = ± 134	2_{IV}^{8-4} 5 = ± 123 6 = ± 234 7 = ± 134 8 = ± 124	2_{IV}^{9-5} 5 = ± 123 6 = ± 234 7 = ± 134 8 = ± 124 8 = ± 1234
32					2_{IV}^{7-2} 6 = ± 1234 7 = ± 1245	2_{IV}^{8-3} 6 = ± 123 7 = ± 124 8 = ± 2345	2_{IV}^{9-4} 6 = ± 2345 7 = ± 1345 8 = ± 1245 9 = ± 1235
64						2_{V}^{8-2} 7 = ± 1234 8 = ± 1256	2_{IV}^{9-3} 7 = ± 1234 8 = ± 1356 9 = ± 3456

Table 3-6 Defining Relationships for Common Fractional Designs

3.5.3 Space Filling Designs

Since the standard factorial designs use only 2 levels per factor, only linear models can be estimated. In the factorial model all design points are on the boundary of the experimental region and the implicit assumption is that any response variable behaves linearly within the design region. An alternative experimental design approach that still attempts to economize on the number of design points, but allows for multiple levels of each factor are so called *space-filling designs* (Santner, Williams *et al.* 2003). While the factorial design evaluated point on the boundary of the design space, space filling designs evaluate points on the interior of the design space thus allowing nonlinear models to be fit. In general space filling designs seek in some sense to place design points evenly throughout the design space.

A popular space filling design is the Latin Hypercube design (Santner, Williams *et al.* 2003). In general the Latin Hypercube approach divides the experimental region into a number of fixed hypercubes and then selects design points randomly from each hypercube. Another space filling approach is known as the Uniform Design (Fang, Lin *et al.* 2000; Fang and Lin 2003; Santner, Williams *et al.* 2003). While the Latin Hypercube (LH) approach has deterministic and random components, the uniform design (UD) approach is purely deterministic[34]. A UD places the design points in a manner that minimizes the *discrepancy* between the empirical distribution of the sample points and a uniform density function, where the discrepancy is some measure of the departure from perfect uniform spacing.

The problem of selecting the uniform design can be stated as follows. For a given set of s parameters, and the corresponding s-dimensional space, find the set of n points $P_n^* = \left\{x_1, ..., x_n\right\} \subset \mathbb{R}^s$ such that the discrepancy measure $D(P_n)$ is minimized. To define the discrepancy measure we first define the empirical distribution of P_n as

$$F_n(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} I\left\{\boldsymbol{x}_i \leq \boldsymbol{x}\right\} \tag{3.83}$$

[34] Although selection of the uniform design points is deterministic, the UD is in general not unique in that multiple designs may exist with the same discrepancy measure.

Where $I\{\cdot\}$ is the indicator function and the inequality is with respect to the componentwise order of $\mathbb{R}^s$. From this function we define a class of discrepancy measures, the L_p discrepancy measures.

$$D_p(P_n) = \left[\int_{\mathbb{R}^s} \left| F_n(x) - F(x) \right|^p dx \right]^{1/p} \tag{3.84}$$

Different value of p create different discrepancy measures. A common choice is to select p equal to ∞, the resulting L_∞ discrepancy (also referred to as the star discrepancy or simple the discrepancy) is then

$$D(P_n) = \sup_{\boldsymbol{x} \in \mathbb{R}^n} \left| F_n(x) - F(x) \right| \tag{3.85}$$

A problem with the Uniform Design is that the selection of P_n^* is itself a difficult problem. (Fang and Lin 2003) provides an overview of design generation techniques. For low dimension - low replication problems, tables are available, see for example (Wang, Lin *et al.* 1995). For moderately large designs, tables can be generated interactively from the uniform design website[35]. Larger designs can be generated using the Uniform Design software application (Fang and Du 1998).

Figure 3-6 shows examples of the UDs generated in $\mathbb{R}^2$ for sample sizes of 100 and 200 using the Uniform Design software.

[35] The Uniform Design website (http://www.math.hkbu.edu.hk/UniformDesign/) is associated with Hong Kong Baptist University.

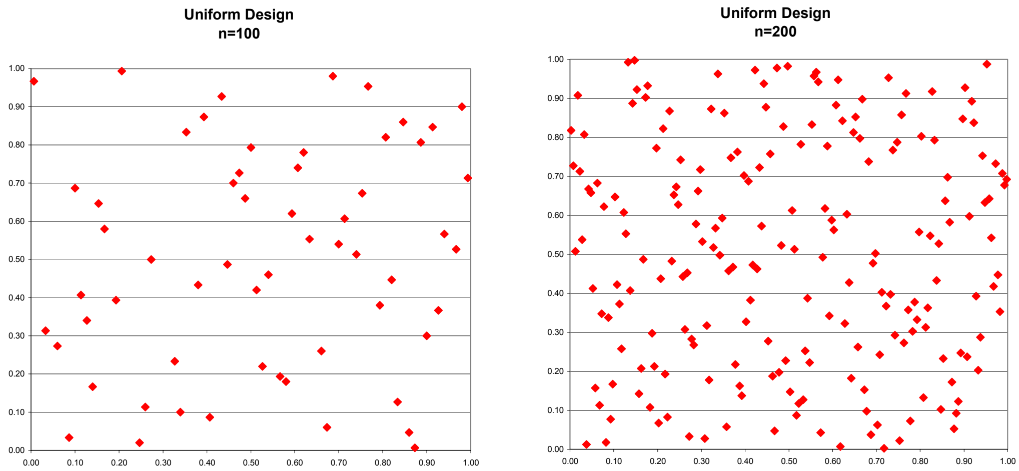


Figure 3-7 Example of Two Dimensional Uniform Designs

We see that the design places points throughout the region in a pattern that is space filling and lacks any obvious pattern[36]. For a given vector dimension and sample size the Uniform Design creates a deterministic approximation of the *n*-dimensional uniform density[37]. Transformation to other densities is straight forward, see for example (Wang, Lin *et al.* 1995).

3.5.4 Applications

Our use of experimental design is two fold. First, we wish to analyze the model's response to changes in deterministic parameters. We therefore utilize formal experimental designs to develop an efficient sensitivity analysis. Secondly, we are interested in adopting experimental design techniques to create efficient representations of the random parameters in our experiments. A straightforward mechanism for representing the vector of random parameters is through random sampling via simulation. Techniques for generating sample paths via simulation are widely discussed in the simulation literature (Johnson 1987; Law and Kelton 2000). Random samples are easy to generate and have the desirable property that the discrete approximation error tends to zero as sample size goes becomes large (Dupacova and Wets 1988). However, the cost of solving

[36] Issues related to observable patterns in uniform designs is discussed in Santer and Willaims et. al (2003) p. 147-148. The UD software eliminates these issues by only considering design matrices of rank *d*, a restriction not required by the definition of UD.

[37] The current version of the software generates designs with up to 30 factors and 600 observations.

the stochastic program increases non-linearly with sample size so the random sample approach may not be very efficient.

One alternative is to approximate the probability density function with a small sample discrete mass function. For example, a method based on Gaussian Quadrature may be used to generate a 3 point approximation that preserves the mean and variance of the density function (Miller and Rice 1983). Alternatively, some researchers have implemented quasi-random techniques such as Latin Hypercube Sampling to create efficient samples. (McKay, Beckman *et al.* 1979) use experimental designs in the evaluation of a continuous simulation experiment. (Simpson, Lin *et al.* 2001) investigates LH and UD designs in the evaluation of two engineering simulations. Several authors have extended this concept to use space filling experimental designs to create efficient scenario sets for stochastic programs. Linderoth *et al*. perform a detailed empirical test of several stochastic programs using random and LH samples (Linderoth, Shapiro *et al.* 2006). (Freimer, Thomas *et al.* 2006) compare random samples with Latin Hypercube samples, and antithetical variates in the context of stochastic programs. (Mo, Harrison *et al.* 2006) utilize UD in a stochastic facility location model.

3.5.5 Summary

In the review above I summarize the literature in the key fields related to this dissertation. The dissertation that follows will pull together these fields and apply them to the call center capacity management problem. I have presented several conference papers that include portions of the work presented in this dissertation or are based on the work in this dissertation. In (Robbins, Medeiros *et al.* 2006) we use simulation to show that arrival rate uncertainty can cause significant deviation from targeted call center performance metrics. That paper describes a call center simulation model that is adapted for use in this dissertation. (Robbins and Harrison 2006) is a stochastic hiring model that is adapted in Chapter 6 of this dissertation. (Robbins, Medeiros *et al.* 2007) summarizes a portion of Chapter 7 and has been submitted to the 2007 Winter Simulation conference. A high level summary of Chapters 2, 5 and 7 is presented in (Robbins 2007) which has been submitted to the 2007 IEEE/INFORMS International Conference on Service Operations and Logistics, and Informatics.

4 The Short Term Scheduling Model

4.1 Introduction

The objective of the short term scheduling model is to determine optimal schedules given uncertainty about arrival patterns. The model develops a week long schedule designed to achieve an aggregate Telephone Service Factor (TSF) based service level agreement (SLA) by assigning a financial penalty based on the probability of failing to achieve the SLA target. It is formulated and solved as a stochastic program, and is optimized over a series of simulated weekly arrival patterns. While the explicit objective of this model is to develop a specific schedule for a given situation, a secondary objective is to study the impact that uncertainty has on the scheduling process. Since most scheduling models ignore arrival rate uncertainty, we seek to understand how uncertainty impacts the optimal schedule and its associated cost.

In Section 4.2 I develop a formulation of the model and discuss the solution algorithm. Section 4.3 examines the various assumptions used in model formulation and assesses how accurate the model is in estimating the aggregate TSF. Section 4.4 discusses the issue of cost and service level tradeoffs. Section 4.5 examines the impact of variability on the optimal schedule and compares the stochastic model to the deterministic mean value model. Section 4.6 examines the impact of staffing flexibility and assesses the impact that part time resources can have on the overall cost of service delivery. Section 4.6 summarizes the results of the analysis and assesses the contributions of the model and its managerial implications.

4.2 Problem Formulation and Solution Approach

4.2.1 Overview

In this model I attempt to find a minimal cost staffing plan that satisfies a global service level requirement. The model estimates the number of calls that meet the service level requirement in each period by making a piecewise linear approximation to the TSF curve; the curve that relates the number of agents to a given service level for a given arrival rate. I generate the linear approximation of the TSF curve based on an Erlang A model which estimates abandonment rates based on an exponential patience distribution. Since the model allows for abandonment, it remains valid if the arrival rate exceeds the service rate. The Erlang C model on the other hand becomes undefined in this condition and the queue size become infinite. Because of the high level of variability in the support desk environment, arrival rates will often exceed service rates, at least temporarily. This happens if we experience unplanned spikes in arrivals. It may also happen by design for short periods of (known) high demand.

I formulate the model as a two stage mixed integer stochastic program. In the first stage staffing decisions are made and in the second stage call volume is realized and we calculate SLA attainment.

4.2.2 Formulation

I formulate a model with the following definitions:

Sets

I: time periods
J: possible schedules
K: scenarios
H: points in a linear approximation

Deterministic Parameters

c_j: cost of schedule j
a_{ij}: indicates if schedule j is staffed in time i
g: global SLA goal
m_{ikh}: slope of piecewise TSF approximation h in period i of scenario k
b_{ikh}: intercept of piecewise TSF approximation h in period i of scenario k
p_k: probability of scenario k
μ_i: minimum number of agents in period i
η_j : maximum number of agents that can be assigned to schedule j
r: per point penalty cost of TSF shortfall

Decision Variables

x_j: number of resources assigned to schedule j

State Variables

y_{ik}: number of calls in period i of scenario k answered within service level
S_k: TSF shortfall in scenario k

Stochastic Parameters

n_{ik}: number of calls in period i of scenario k

The model can then be expressed as

$$\min \sum_{j\in J} c_j x_j + \sum_{k\in K} p_k r S_k \tag{4.1}$$

subject to

$$y_{ik} \le m_{ikh} \sum_{j\in J} a_{ij} x_j + b_{ikh} \qquad \forall i\in I, k\in K, h\in H \tag{4.2}$$

$$\sum_{i\in I} n_{ik} S_k \ge \sum_{i\in I} (g n_{ik} - y_{ik}) \qquad \forall i\in I, k\in K \tag{4.3}$$

$$y_{ik} \le n_{ik} \qquad \forall i\in I, k\in K \tag{4.4}$$

$$\sum_{j\in J} a_{ij} x_j \ge \mu_i \qquad \forall i\in I \tag{4.5}$$

$$x_j \le \eta_j \qquad \forall j\in J \tag{4.6}$$

$$x_j \in \mathbb{Z}^+, y_{ik} \in \mathbb{R}^+, S_k \in \mathbb{R}^+ \qquad \forall i\in I, k\in K, h\in H \tag{4.7}$$

The objective of this model is to minimize the total cost of staffing plus the expected penalty cost associated with failure to achieve the desired service level. The optimization occurs over a set K of sample realizations of call arrivals. These samples are called scenarios. Constraint (4.2) defines the variable y_{ik} as the number of calls answered within SLA in period *i* of scenario *k* based on a convex linear approximation of the TSF curve. Constraint (4.3) calculates the TSF proportional shortfall; the maximum of the percentage point difference between the goal TSF and achieved TSF and zero. Constraint (4.4) limits the calls answered within the SLA target to the total calls received in the period. Constraint (4.5) defines the minimum number of agents in any period. Constraint (4.6) sets an upper limit on the number of agents assigned to each schedule. Constraint (4.7) defines the non-negativity and integer conditions for program variables.

This model is derived from the basic set covering formulation in models such as (Dantzig 1954) but with the following extensions:

- The server sizing and staff scheduling steps are combined into one optimization program.
- The model is explicitly defined for a queuing process with a non homogeneous arrival rate.
- The model explicitly recognizes that the arrival rate is a random variable.
- This model specifies both a per-period performance constraint and a global performance constraint.
- The model uses a piecewise linear approximation for the TSF curve derived from an Erlang A model.

The size of the model, and therefore the computation effort required to solve it, is driven in large part by two factors; the number of potential schedules (J) and the number of scenarios (K). The number of integer variables is equal to the number of schedules, while the number of continuous variables is equal to the product of the number of scenarios and the number of time periods, plus the number of scenarios.

In this analysis we are creating schedules for a week (with explicit breaks between shifts, but not within shifts.) In simple cases where I allow only five day a week eight hour shifts, the number of possible schedules is 576. In more complex cases where we have a wider range of full and part time schedule options we have 3,696 schedules. I investigate the number of scenarios required in the next section, but 50 scenarios is not unreasonable. This implies the requirement to solve models with 3,696 integer variables and over 16,000 continuous variables.

4.2.3 Scenario Generation

This program (4.1) -(4.7) is solved over some set of sample outcomes from the statistical model of call arrival patterns. An algorithm for generating simulated calls was presented in Figure 2-10 and is repeated here for convenience.

```
For d = 1 to 7
   Read $DA_d, DS_d$                   ' Read daily average and sd
   $DV_d$ = RndNorm($DA_d, DS_d$)      ' Generate random volumes
Next
For d = 1 to 7                         ' Gen initial proportions
   For t = 1 to 48
      Read $TA_t, TS_t$                ' Read period average and sd
      $TP_t$ = min[RndNorm($TA_t, TS_t$),0] ' Calc initial proportion
      $SP_d = SP_d + TP_t$             ' Sum up proportions
   Next
Next
For d = 1 to 7                         ' Normalize proportions
   For t = 1 to 48
      $TV_{dt} = TP_t*DV_d/SP_d$       ' Calculate period volume
      $LAM_{dt} = 2* TV_{dt}$          ' Calculate arrival rate
   Next
Next
```

Figure 4-1 Simulated Call Generation Algorithm

This algorithm is coded in Visual Basic.Net and is used to generate scenarios for the optimization algorithm. The program is designed to generate scenarios in batches of variable size. Each batch is fully independent of other batches; i.e. the first scenario in a batch of 50 scenarios is different from the first scenario in a batch of 100 scenarios. The algorithm writes call volumes to a file that can be read directly by the GAMS system[38]. This algorithm uses a relatively simple approach to generate simulated call volumes. The objective here is to generate a reasonable set of call arrival patterns to test the optimization model. A more detailed set of generation algorithms is provided in (Avramidis, Deslauriers *et al.* 2004).

[38] GAMS is the front end system used to define the model and code the solution algorithm. GAMS calls on CPLEX which actually solves the integer and linear programs.

4.2.4 Solution Algorithm

This model is formulated as a MIP and as such can be solved by an implicit enumeration (branch and bound) algorithm. Branch and bound works well for smaller problems, but tends to bog down as the number of scenarios increases. To facilitate the solution of large scale problems I implemented a version of the L-Shaped decomposition algorithm (Birge and Louveaux 1997). My decomposition method is a straight forward implementation of this method, adapted for a discrete first stage. I decompose the problem into a master problem where the staffing decision is made, and a series of sub-problems where the TSF shortfall is calculated for each scenario.

Letting v denote the major iterations of the algorithm, θ^v the approximated recourse cost, and E_{ik}^v and e_{ik}^v the coefficient of the recourse function *cuts*, the master problem can be defined as

$$\min \sum_{j \in J} c_j x_j + \theta^v \tag{4.8}$$

subject to

$$\theta^v \geq \sum_{k \in K} p_k E_{ik}^v \sum_{j \in J} a_{ij} x_j + e_{ik}^v \qquad \forall i \in I, v \tag{4.9}$$

$$\sum_{j \in J} a_{ij} x_j \geq \mu_i \qquad \forall i \in I \tag{4.10}$$

$$x_j \leq \eta_j \qquad \forall j \in J \tag{4.11}$$

$$x_j \in \mathbb{Z}^+, \theta^v \in \mathbb{R}^+ \qquad \forall j \in J \tag{4.12}$$

In this problem θ^v represents an estimate of the TSF shortfall penalty term. Let (x^v, θ^v) be an optimal solution.

For each realization of the random vector $k = 1, ..., K$ we then solve the following subproblem

$$\min rS_k \tag{4.13}$$

subject to

$$y_{ik} \leq m_{ikh} \sum_{j \in J} a_{ij} x_j^v + b_{ikh} \qquad \forall i \in I, k \in K, h \in H \tag{4.14}$$

$$\sum_{i\in I} n_{ik} S_k \ge \sum_{i\in I}(gn_{ik} - y_{ik}) \qquad k \in K \qquad (4.15)$$

$$y_{ik} \le n_{dik} \qquad \forall i \in I, k \in K \qquad (4.16)$$

$$x_j^v \in \mathbb{Z}^+, y_{ij} \in \mathbb{R}^+, S_k \in \mathbb{R}^+ \qquad \forall i \in I, k \in K, h \in H \qquad (4.17)$$

I use the dual variables from the solution of the set of subproblems to improve the approximation of the penalty term. Let $\pi 1_{ikh}^v$ be the dual variable associated with(4.14), $\pi 2_k^v$ the dual variable associated with (4.15), and $\pi 3_{ik}^v$ the dual variable associated with (4.16). I then calculate the following parameters used for cut generation:

$$E_{ik}^{v+1} = \sum_{i\in I}\sum_{h\in H} \pi 1_{ikh}^v m_{ikh} \sum_{j\in J} a_{ij} x_j^v \qquad (4.18)$$

$$e_k^{v+1} = \sum_{i\in I}\left[\pi 3_{ik}^v n_{ik} + \sum_{h\in H} \pi 1_{ikh}^v b_{ikh} n_{ik}\right] - \pi 2_k^v g \sum_{i\in I} n_{ik} \qquad (4.19)$$

I use these values to generate a constraint of the form (4.9). Set $v = v+1$, add the constraint to the master and iterate. The algorithm solves the master program then solves each sub-program for the fixed staffing level defined in the master solution. Based on the solution of the sub-problems, each iteration adds a single cut to the master problem. These cuts create an outer linearization of the penalty function (Geoffrion 1970).

The solution of the master problem provides a lower bound on the optimal solution, while the average of the subproblem solutions provides an upper bound. In my implementation I solve the LP relaxation of the master until an initial tolerance level on the optimality gap is achieved and I then reapply the integrality constraints. I continue to iterate between the master MIP and the subprogram LPs until a final tolerance gap is achieved.

Whereas the branch and bound approach solves a single large MIP, the decomposition solves a large number of relatively small LPs and a single moderately sized MIP[39]. The advantage of the

[39] A representative instance with 100 scenarios required 30 major iterations, thereby requiring the solution of the master problem 30 times, and the subproblem 3,000 times. The master was solved as an LP relaxation 26 times and as a MIP 4 times.

decomposition approach is that solution time will tend to increase as an approximately linear function of the number of scenarios, while the branch and bound algorithm will increase as a non-linear function of the number of scenarios.

The following graph shows the results of solving an instance with a moderate number of schedules (384), and a variable number of scenarios using a branch and bound solution of the extensive form, and the decomposition algorithm. In each case I solved five instances with a randomly generated set of scenarios.

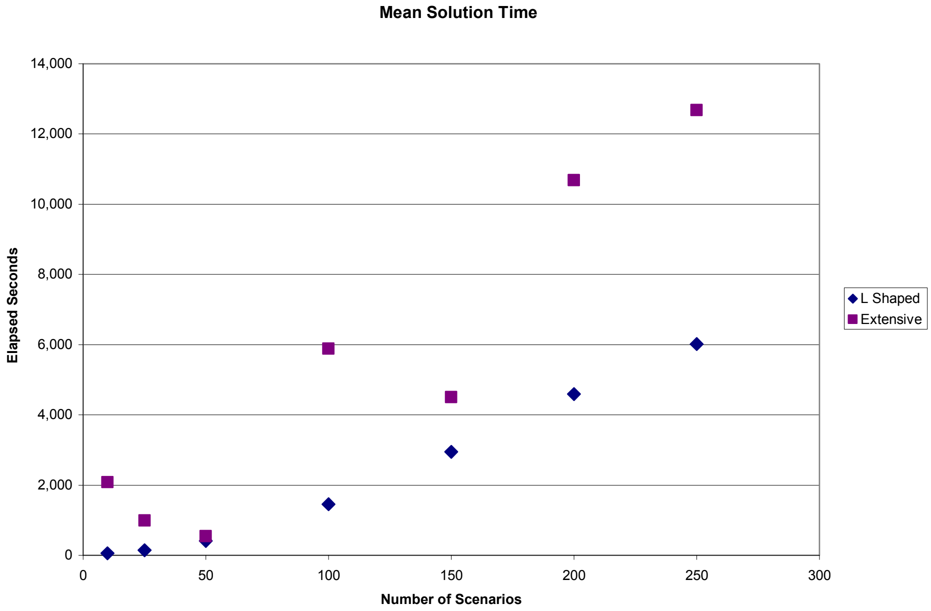


Figure 4-2 Mean Solution Times

The graph shows that the solution time for the L-Shaped method does increase in an approximately linear fashion in the number of scenarios. Average solution time for the branch and bound algorithm is larger in each case, though it is somewhat erratic; the average solution time for 150 scenarios is, for example, smaller then the average solution time for 100 scenarios. The following graphs illustrate individual solution times:

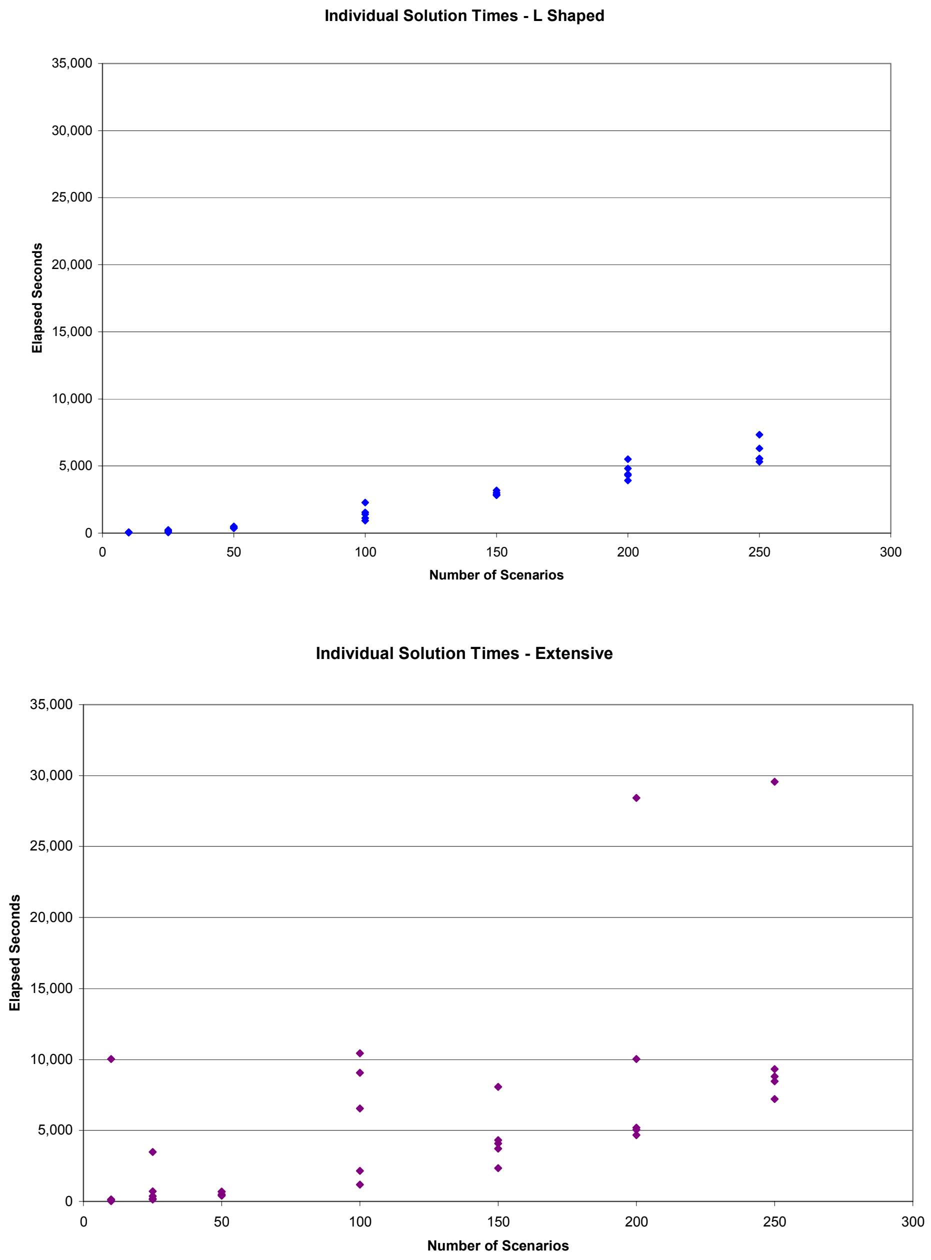


Figure 4-3 Individual Solution Times

The variance of solution times for the L Shaped method is much lower than for the branch and bound method. The average performance of the branch and bound method is highly influenced by the worst case solution time. Note that these solution times are for instances with only a moderate number of schedules (384). In later experiments we increase the number of schedules to well over 2,000 as we introduce more flexible staffing options. Given that each schedule option creates an integer variable solving these larger problems by branch and bound will be extremely difficult[40].

The following graph illustrates the convergence of the L-Shaped decomposition algorithm[41].

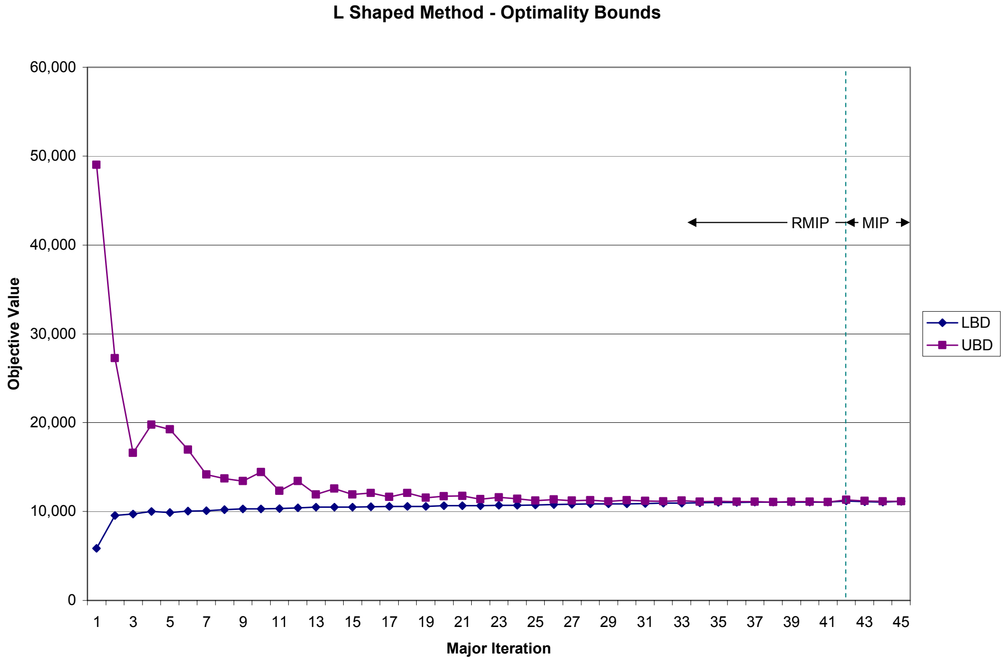


Figure 4-4 Convergence of the L-Shaped Algorithm

As is the case with a branch and bound algorithm relatively good bounds are found in the first few iterations. Convergence then slows as each successive iteration cuts a smaller area from the feasible region of (4.8) - (4.12). In this particular case the relaxation was solved 41 times and the MIP was solved 4 times. A slight shift in the bounds occurs when the integrality constraints are reapplied in iteration 42. The bump that occurs when integrality constraints are reapplied in this, and many other instances, is quite small. I believe this is due to two facts. First, by the time integrality constraints are reapplied, a large number of cuts have been applied, narrowing the search to a relatively small region. Second, an instance the weighted set covering problem, with many schedule options, has a large number of nearly identical solutions. In some instances where

[40] I have solved instances of the problem with 500 scenarios using the L Shaped method and there seems to be no upper limit above which the problem will not solve as is often the case in branch and bound.
[41] This particular instance had 384 schedules and 100 scenarios.

the algorithm switches to MIP mode with fewer cuts, such as when the penalty rate is set to zero, the bump is more significant and the time to solve the final MIP can be much longer.

This analysis indicates that the decomposition method provides a generally superior approach for solving most instances of this problem. It is more scalable and the solution time tends to be less variable. The remainder of this analysis is based on a decomposition algorithm.

4.2.5 Post Optimization Analysis

The solution of (4.1) - (4.7) is the optimal solution of the sample path problem. We denote this solution as z_n^*, where n is the number of scenarios used to calculate the solution. This is a biased estimate of the solution true problem; that is the problem evaluated against the continuous distribution of arrival rates. I denote the true solution as z^*. (Mak, Morton *et al.* 1999) show that the expected bias in the solution is decreasing in sample size

$$E[z_n^*] \le E[z_{n+1}^*] \le z^* \tag{4.20}$$

From a practical perspective a key decision is determining the number of scenarios to use in our optimization. As I increase the number of scenarios the solution becomes a better approximation of the true solution, but the computational cost of finding that solution increases.

To aid in this process I perform a post optimization evaluation of the candidate solution using a Monte Carlo bounding process described in (Mak, Morton *et al.* 1999). Denote the solution to the sample problem as $\hat{x}$. I then solve the subprogram (4.13) to (4.17) using $\hat{x}$ as the candidate solution, to obtain the expected cost of implementing this solution. In this analysis I solve the subprogram with n_u equal 500 scenarios generated independently from the scenarios used in the optimization. The solution to the subprogram gives us an upper bound on the true solution, while the solution to the original problem z_n^* is a lower bound.

To obtain better bounds on the true optimal solution we may choose to solve the original problem multiple times, each with independently generated scenarios. Denote the number of batches (sets of scenarios) used to solve the original problem as n_ℓ and the sample variance of the objective as

$s_\ell(n_\ell)$. Similarly I calculated the sample variance of the expected outcome of the candidate solution against the n_u evaluation scenarios. We can then define the following standard errors

$$\tilde{\varepsilon}_u = \frac{t_{n_u-1,\alpha} s_u(n_u)}{\sqrt{n_u}} \tag{4.21}$$

$$\tilde{\varepsilon}_\ell = \frac{t_{n_\ell-1,\alpha} s_\ell(n_\ell)}{\sqrt{n_\ell}} \tag{4.22}$$

Where $t_{n_u-1,\alpha}$ is a standard *t*-statistic, *i.e.* $P\left\{T_n \le t_{n_u-1,\alpha}\right\} = 1-\alpha$. We can now define an approximate $(1-2\alpha)$ confidence interval on the optimality gap as

$$\left[0, \left[\bar{U}(n_u) - \bar{L}(n_\ell)\right]^+ + \tilde{\varepsilon}_u + \tilde{\varepsilon}_\ell\right] \tag{4.23}$$

Note that we take the positive portion of the difference between the upper and lower bounds because it is possible, due to sampling error, that this difference is negative. This procedure allows us to generate a statistical bound on the quality of our solution, i.e. the potential distance from the true optimal.

4.3 TSF Approximation and SIPP

4.3.1 Overview

This model attempts to generate a schedule that meets a Service Level Agreement (SLA) at a minimal cost. For the sake of this analysis, I assume that the SLA is defined based solely on the TSF. In order to do so effectively the optimization program must estimate the service level that will be achieved for any staffing plan for each realization of calls. In this section I outline the approach used to estimate the TSF and document the assumptions used in developing this estimate. I then attempt to validate the estimate using a discrete event simulation model.

4.3.2 Basic TSF Calculations

The basic model used to estimate the service level in this analysis is the Erlang A model. The Erlang A model is a widely accepted model for call center systems with non-negligible abandonment rate. Details of the Erlang A model are presented in section 3.3.2.3 and summarized here. Erlang A assumes calls arrive via a Poisson process with rate λ and are served by a set of homogeneous agents with an exponentially distributed service time with mean $1/\mu$. If no agent is available when the call arrives it is placed in an infinite capacity queue where it waits for the next available agent. Each caller has a patience level which are iid draws from an exponential distribution with mean $1/\theta$. If a caller is not served by the time her patience expires she hangs up. The call center is also assumed to have infinite capacity so no calls are blocked.

In steady state, the staffing decision then involves forecasting the arrival rate λ_i and setting the staff level based on equation (3.36). The Erlang A model is difficult to calculate and I use a series of approximations defined explicitly in equations (3.20) - (3.36). The result is a non-linear S-shaped curve that for a fixed arrival rate, relates the achieved service level to the number of agents staffed. The following figure shows an example.

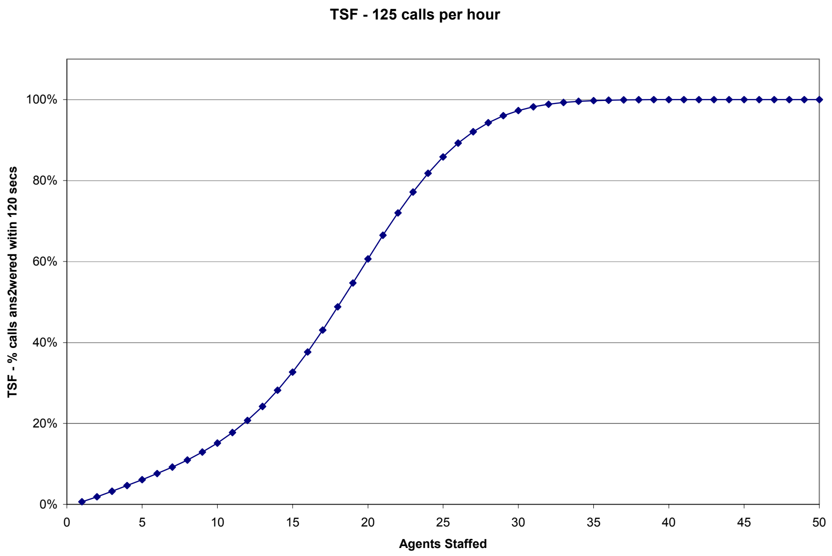


Figure 4-5 TSF Curve for a Fixed Arrival Rate

4.3.3 Piecewise Linear Approximation

It is obvious from this graphic that the TSF curve is neither convex nor concave over the full range of staffing. For very low staffing levels, where performance is very poor, the curve is convex and we experience increasing efficiency from incremental staffing. For higher staffing levels the curve becomes concave and the impact of incremental staffing becomes decreasing.

Note that the area of convexity corresponds to very poor system performance; an area where we do not plan to operate. In addition, embedding this function in our optimization model creates a non-convex optimization problem.

To address this problem I create a piecewise linear, convex approximation to the TSF curve as shown in the following figure:

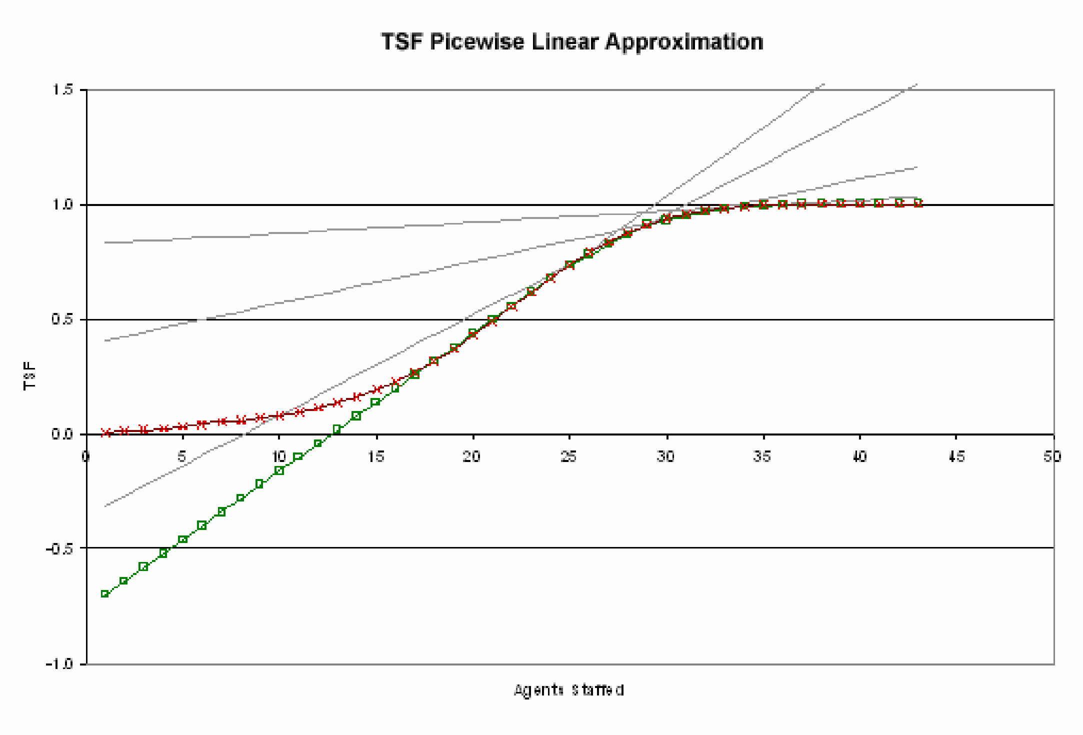


Figure 4-6 Piecewise Approximation of TSF

In this graph the straight lines represent the individual constraints, and the piecewise linear function is my approximation of the nonlinear curve[42]. The piecewise linear approximation and the true TSF curve are very close for staffing levels above 15 for this data[43]. For very low staffing levels the linear approximation will overly penalize performance, potentially calculating a negative TSF level. My assumption is that we are almost always operating in the higher performance region; I constrain the problem so that expected performance in any period is over

[42] This graph has five linear segments, including a horizontal segment at a service level of 100%. The optimization model requires that the TSF is less than each line segment. The optimization process will force these constraints to be binding.

[43] In general the piecewise approximation will provide a good approximation if staffing levels are large enough; that is if the staff levels are above the lower inflection point of the TSF curve.

some minimal threshold level of say 50%. Only in the case of very large shocks will we ever be driven into the poor performance region.

4.3.4 Non Stationary Arrivals

I estimate the TSF in each period using equations (3.20) - (3.36). However, these equations are based on limiting behavior in steady state. For the most part, our analysis is concerned with nonstationary transient behavior. In my analysis I use a Stationary Independent Period by Period approximation. The SIPP approach is described in more detail in (Green, Kolesar *et al.* 2001) and is reviewed in section 3.3.2.5 of this thesis. Essentially in this approach I divide each day into 48 periods of 30 minutes each. I then estimate the average number of calls received in that period, set the arrival rate appropriately, and assume steady state behavior is quickly achieved in that period. I therefore assume that equations (3.20) - (3.36) can be used to estimate system performance in each 30 minute period using average arrivals. In applying the SIPP approach the arrival rate is assumed to change discontinuously at the start of each 30 minute as shown in the following figure

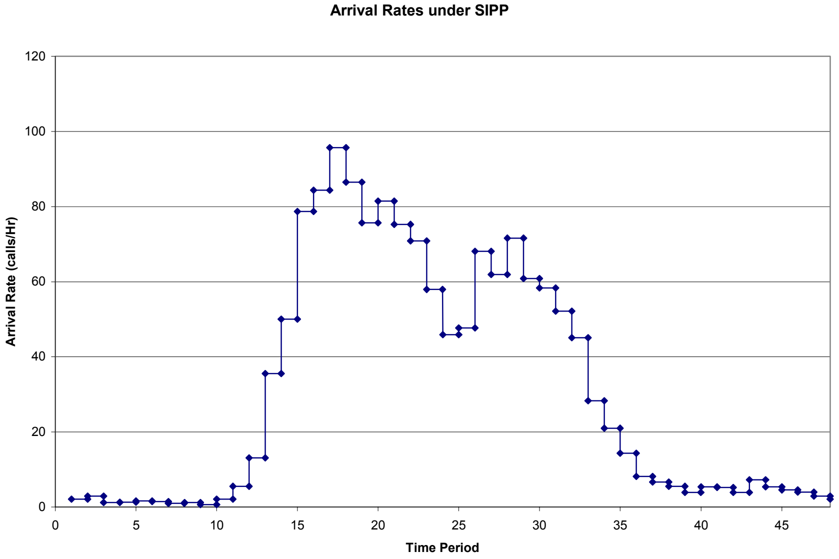


Figure 4-7 Arrival Rates for the SIPP Approximation

Clearly these assumptions have the potential to introduce significant error and the literature suggests several modifications of this approach, namely the SIPP Max and SIPP Mix approaches, both of which attempt to adjust for performance bias in the standard SIPP approach. We can define the three alternative approaches as follows. Let $n(t)$ be the simulated arrivals that occur in the (thirty minute) period t, and let $\lambda(t)$ denote the arrival rate used to calculate the service level.

In the standard SIPP approach the arrival rate is

$$\lambda(t) = 2n(t) \tag{4.24}$$

The SIPP Max approach uses the maximum arrival rate over the course of the period. I implement this as the maximum of the arrival rate in the current period, the average of the current and preceding rates, and the average of the current and succeeding rates.

$$\lambda(t) = 2 \cdot \max\left[(n(t-1)+n(t))/2, n(t), (n(t)+n(t+1))/2\right] \tag{4.25}$$

Finally the SIP Mix approach uses the current arrival rate when rates are increasing, but the average of the preceding and current rates when rates are declining.

$$\lambda(t) = \begin{cases} 2n(t) & n(t) > n(t-1) \\ n(t)+n(t-1) & n(t) \le n(t-1) \end{cases} \tag{4.26}$$

4.3.5 Scenario Based TSF Approximation

The TSF calculations defined above are based on the call volume in each 30 minute period, n_{ik}, which is a random variable. The TSF calculations are therefore dependent on the sample path and must be included in the scenario generation algorithm. A comprehensive algorithm for scenario generation is then provided by the following algorithm:

1. Generate a week of call volume using the algorithm shown in Figure 4-1.
2. Based on the SIPP method calculate the per period arrival rate using equations (4.24), (4.25) or (4.26).
3. For a given call volume, select $h+1$ probability levels for estimating points on the TSF curve[44].
4. Calculate the staff level required to achieve the target probabilities defined in Step 3 using equations (3.20) - (3.36).
5. Recalculate the TSF for the integral staffing level calculated in Step 4. We now have $h+1$ staff level probability pairs on the TSF curve.

[44] In practice I use values of .3, .72, .9, .98, and .995 for all periods with call volumes of at least 5. Different values are used for lower call volumes to maintain a convex approximation.

```
6. Calculate the slope ($m_{ikh}$) and intercept ($b_{ikh}$) for each
   pair of adjacent points found in Step 5.
7. Generate a scenario that includes the per period call
   volumes ($n_{ik}$) and $h$ pairs of slope and intercept
   parameters for each period in the planning horizon.
```

Figure 4-8 Scenario Based TSF Approximation Approach

In addition to the individual scenario information, parameters for the minimum agent level constraint (4.5) must be generated. This is a straightforward procedure as follows:

```
1. Define w, the worst case acceptable expected service
   level, and $n_{min}$ the overall minimum number of agents to
   be staffed at any time[45].
2. Repeat Steps 2 – 6 for each period i
3. Determine the expected call arrival rate
4. Calculate the staff level $n_w$ required to achieve the
   worst case expected service level defined in Step 1
   using equations (3.20) - (3.36).
5. Calculate $\mu_i = \lceil \min(n_w, n_{\min}) \rceil$, the minimum agents to staff
   in period i.
6. Write out $\mu_i$ in a GAMS compatible format.
```

Figure 4-9 Minimum Staff Level Constraint Generation

The scenario generation algorithm described above is written in VB.Net. It generates scenario files in a format that can be read by GAMs and are used to generate CPLEX models. A 100 scenario file is generated in a few seconds on a desktop computer. Overall, the scenario generation time is negligible as compared to solution time for the stochastic program.

4.3.6 SIPP Testing and Model Validation

4.3.6.1 Overview

A number of approximations go into calculating the service level in this stochastic optimization model. Since the TSF level is the key driver of the staffing level, it is reasonable to question the accuracy of these approximations and to consider the SIPP adjustments discussed above. In this section I perform a numerical experiment to test the accuracy of each version of the SIPP model.

[45] Throughout this dissertation, unless stated otherwise I use a worst case TSF of 50% and a minimum staffing level of 2.

I test each SIPP version against models based on 3 of the model projects[46] described in Section 2-7.

4.3.6.2 Experimental Approach

As outlined above, the basic process involved with solving the stochastic program is to solve the model (4.1) - (4.7) against a set of scenarios, simulated realizations of call volume. During this process I calculate a TSF level and objective value, both of which are biased estimates of the true values. I then perform a post optimization analysis which tests the candidate solution against a set of evaluation scenarios. In this process I calculate an expected outcome (service level and cost) that is an unbiased estimate of the true solution[47]. The objective of this validation is to determine how well the unbiased, post optimization analysis is at predicting the actual realized service level. Note that since arrivals are random, the realized service level will be random.

In order to make this assessment I turn to Discrete Event Simulation (DES). DES is a well established methodology for examining complex queuing systems like this one. Using DES we can more closely model the specific behavior of the system to specific realized call patterns. The DES model used in this analysis uses the same algorithm shown in Figure 4-1 to generate a nonstationary call pattern. The model then generates individual simulated calls which are processed using the same theoretical distributions used in the Erlang A model. The simulation approach allows us to run the model for a large number of simulated arrival patterns and to calculated statistical bounds on key performance metrics such as TSF. See (Banks 2005) or (Law 2007) for a detailed description of the simulation process. Assuming that the DES model is a valid representation of the non-stationary Erlang-A queuing model, we can use this model to assess the accuracy of the TSF calculation in the optimization program.

The validation processed is outlined below:

1. Generate a set of 100 scenarios and use these to solve the stochastic optimization problem (4.1) - (4.7).
2. Using the solution found in step 1 as the candidate solution, perform a post optimization evaluation

[46] The fourth project outlined in section 2.7 is too small to be off interest in the scheduling model. Given its very low volumes the project is almost always staffed at the minimal staffing level of two agents. I examine this project in the final model of this thesis which addresses project pooling.

[47] The post optimization process is discussed in more detail in the next session.

against 500 independently generated scenarios to find the expected service level associated with the candidate solution.

3. Use the period by period staffing plan developed in step 1 to create the resource profile in a discrete event simulation model with an identical statistical distribution of call volumes.
4. Perform 50 replications of the DES model to calculate a point estimate of the expected TSF from implementing the solution found in step 1 using SIPP, SIPP Max, and SIPP Mix.
5. Compare the results found in step 2 to those found in step 4 to assess the error associated with each SIPP approach.

Figure 4-10 TSF Validation Approach

In the following table I summarize the results from applying this approach to compare the three SIPP models to Project J.

		SIPP Method	
Optimization	Std	Max	Mix
Scheduled Hours	1,160	1,200	1,200
Expected TSF	83.2%	81.0%	83.2%
Std. Dev of TSF	2.6%	3.0%	2.6%
Simulation	Std	Max	Mix
Expected TSF	81.50%	84.00%	84.29%
Std. Dev of TSF	2.70%	2.87%	2.46%
Bias (Opt-DES)	-1.72%	3.00%	1.07%
Error in Std Dev of Sim TSF	-0.64	1.05	0.43

Table 4-1 TSF Validation – Project J

As predicted by theory, the standard SIPP model overestimates the expected service level and under staffs the call center. However, at least in this case the error is rather low. The TSF estimated in the optimization program is only 1.72% above what is estimated by the DES model. Furthermore in the DES model the standard deviation of the TSF measure is 2.70%, so the estimate is within .64 standard deviations. Both the SIPP Max and SIPP Mix models use a more conservative estimate of the service level attained and as a result calculate a higher staffing level. The SIPP Max is the most conservative and underestimates the service level by over 3%. SIPP Mix is less conservative and underestimates the service level by 1.07%. SIPP Mix is arguably a better fit in this case as the error is slightly smaller and in a conservative direction. If I apply the same analysis to projects S and O we obtain the following results.

	Project O			Project S		
	SIPP Method			SIPP Method		
Optimization	Std	Max	Mix	Std	Max	Mix
Scheduled Hours	1,080	1,160	1,120	2,760	2,880	2,880
Expected TSF	81.5%	82.8%	81.9%	78.2%	78.3%	76.2%
Std. Dev of TSF	2.9%	2.8%	3.0%	9.7%	10.4%	11.0%
Simulation						
Expected TSF	80.99%	84.70%	84.30%	79.33%	81.80%	83.28%
Std. Dev of TSF	3.22%	3.02%	3.40%	4.74%	4.80%	3.64%
Error (Opt-DES)	-0.54%	1.91%	2.45%	1.15%	3.50%	7.08%
Error in Std Dev of Sim TSF	-0.17	0.63	0.72	0.24	0.73	1.95

Table 4-2 TSF Validation – Projects O and S

Results from these two projects again show that the SIPP Standard method is the least conservative, but in the case of project O it overestimates the service level. The Standard SIPP model is in general the most accurate and I will utilize this approach in the remainder of this analysis.

4.4 Cost and Service Level Tradeoffs

In a deterministic optimization approach to call center scheduling we set a performance target for some metric and then find the minimal cost schedule that satisfies that constraint. In a stochastic setting the solution criteria is more complex. Given that call volume, and therefore service level is random, the performance target can only be expressed in probabilistic terms. The resulting schedule will achieve the stated performance target with some probability. I call this probability the *confidence level*. Given the nature of arrival variability it is not practical, or desirable, to generate a schedule that will always achieve the service level target as this schedule would be prohibitively expensive.

In my formulation I express the degree of certainty indirectly by assigning a financial penalty to the probability of missing the performance target. By adjusting the performance penalty factor r, I adjust the degree of certainty associated with meeting the target. We now analyze the relationship between the penalty rate, the cost of service delivery, and the confidence associated with the performance target. This model applies two performance constraints. Constraint (4.5) defines a minimum staff level in each period, which in my test cases I set to the minimum of a global minimum staffing level and the staffing level required to achieve some minimal performance level at expected volumes.[48] If the penalty rate is set to zero the penalty term drops out of the objective function and constraint (4.5) becomes binding. As I increase the penalty rate the scheduled staff levels will increase to balance the cost of staffing and the expected penalty cost associated with TSF shortfalls.

In the following tables I show the result of an experiment to evaluate the impact of various penalty rates. For each project I evaluate the schedule at eight design points (DPs) and in each case we solve the stochastic problem five times, each with an independent batch of 50 scenarios. I then evaluate that solution against an independently generated set of 500 scenarios to determine the expected outcome of implementing the candidate solution. The model is solved with the constraint that all schedules are full time (40 hours)[49].

		Average				Standard Deviation			
	Penalty	Labor	Expected	Average		Labor	Expected	Average	
DP	Rate	Cost	Outcome	TSF	Confidence	Cost	Outcome	TSF	Confidence
1	0	8,800	8,800	60.5%	0.0%	0	0	0.00%	0.00%
2	25,000	10,800	11,008	80.6%	61.6%	0	18	0.16%	2.73%
3	50,000	10,880	11,249	81.0%	65.7%	179	40	1.16%	12.71%
4	75,000	11,120	11,332	82.6%	82.9%	179	28	1.11%	11.35%
5	100,000	11,120	11,419	82.7%	83.1%	179	127	1.11%	11.74%
6	150,000	11,200	11,458	83.1%	87.9%	0	36	0.30%	2.74%
7	200,000	11,200	11,504	83.1%	88.8%	0	56	0.23%	2.36%
8	250,000	11,200	11,597	83.1%	89.0%	0	72	0.31%	2.30%

Table 4-3 Cost and Service Level Tradeoffs – Project J

		Average				Standard Deviation			
	Penalty	Labor	Expected	Average		Labor	Expected	Average	
DP	Rate	Cost	Outcome	TSF	Confidence	Cost	Outcome	TSF	Confidence
1	0	20,880	20,880	52.5%	0.0%	179	179	0.82%	0.00%
2	25,000	22,880	26,869	64.1%	1.9%	179	23	0.71%	1.00%
3	50,000	26,160	29,280	75.2%	41.1%	358	31	1.07%	7.26%
4	75,000	26,800	30,677	77.0%	53.2%	283	59	0.71%	4.76%
5	100,000	27,920	31,801	79.5%	67.3%	769	118	1.42%	6.45%
6	150,000	29,040	33,554	81.5%	76.1%	1,152	89	1.72%	5.03%
7	200,000	30,480	34,801	83.7%	80.9%	1,481	343	2.20%	6.47%
8	250,000	31,920	35,662	85.7%	84.4%	1,559	392	2.26%	4.23%

Table 4-4 Cost and Service Level Tradeoffs – Project S

[48] In our test problems we require that at least 2 agents are scheduled at all times. We also require that at expected volumes we achieve a minimum 50% TSF in each period.
[49] This issue is addressed thoroughly in section 4.6 Here I use schedule B.

			Average				Standard Deviation		
	Penalty	Labor	Expected	Average		Labor	Expected	Average	
DP	Rate	Cost	Outcome	TSF	Confidence	Cost	Outcome	TSF	Confidence
1	0	8,240	8,240	54.2%	0.0%	219	219	1.49%	0.00%
2	25,000	10,800	11,705	76.8%	27.2%	0	37	0.17%	1.52%
3	50,000	11,360	12,294	79.9%	62.0%	219	37	0.97%	11.80%
4	75,000	11,600	12,736	80.6%	71.6%	0	58	0.33%	3.72%
5	100,000	11,600	13,022	80.9%	74.2%	0	46	0.21%	1.89%
6	150,000	12,000	13,595	82.5%	86.2%	0	21	0.17%	2.49%
7	200,000	12,000	14,127	82.4%	86.0%	0	112	0.36%	3.40%
8	250,000	12,320	14,591	83.1%	89.3%	179	72	0.71%	2.30%

Table 4-5 Cost and Service Level Tradeoffs – Project O

The following figures show the same data graphically. In the first set of graph I show how confidence and average service level vary with the penalty rate.

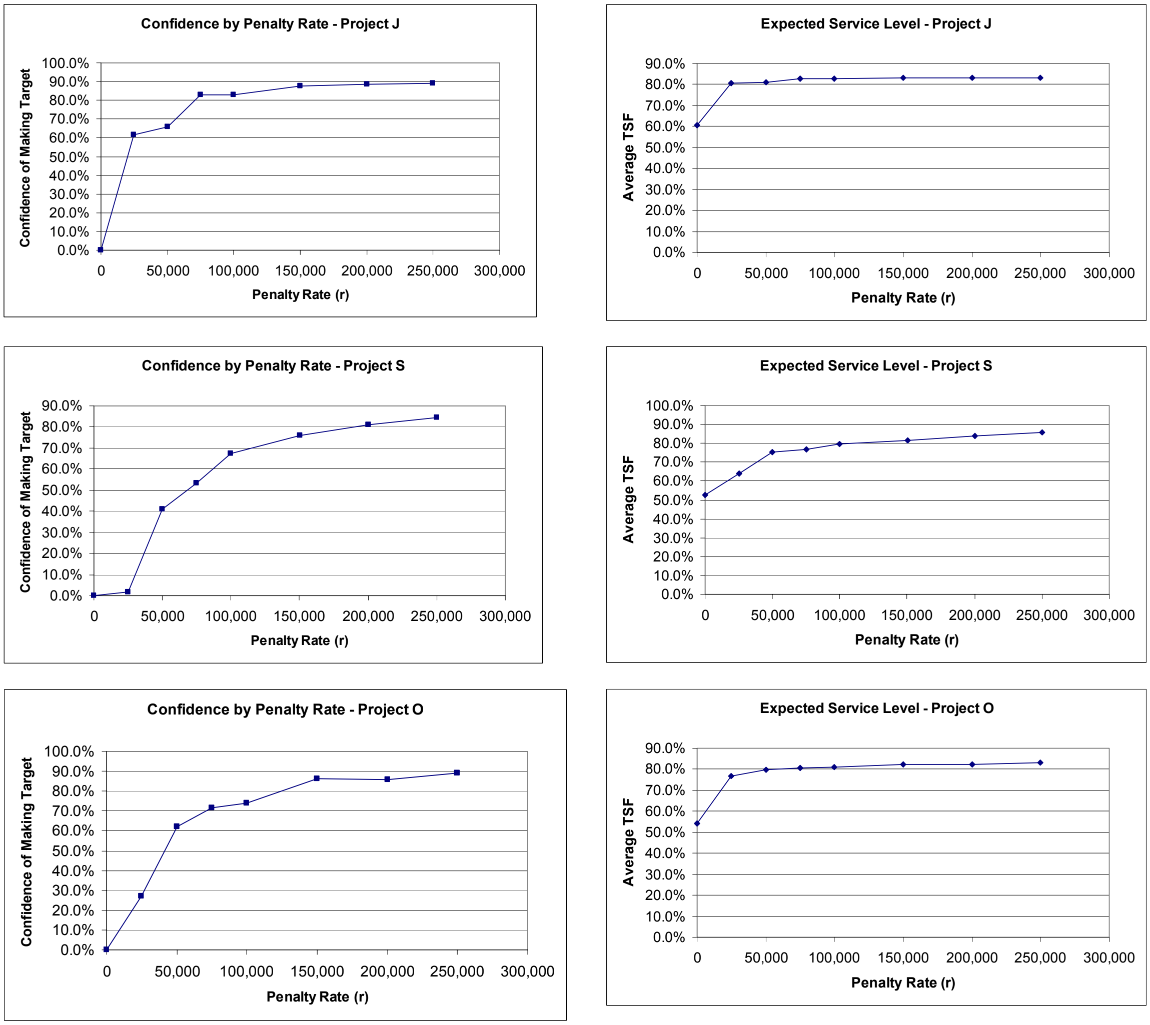


Figure 4-11 Confidence and Expected Service Level as a function of Penalty Rate

For each project the panel on the left shows the confidence level of the resulting solution, i.e. the proportion of the evaluation scenarios in which the performance target was achieved. The panel on the right shows the corresponding expected service level associated with the candidate solution. These plots imply that an *efficient frontier* exists, an optimal region that balances the labor cost and penalty cost for a given vector of labor and penalty rates.

In all cases low penalties result in a zero confidence and an expected TSF near 60%[50]. As the penalty rate increases the expected TSF begins to increase as additional staffing is added to offset shortfall penalties. Both factors increase rapidly and then level off as it becomes increasingly expensive to meet the service levels in the tail of the arrival rate distribution. It is interesting to note that each project requires a different penalty rate to achieve a desired confidence level. Project S which has the largest staff levels and a high degree of variability, requires penalty rates in the range of $200,000 (2,000 per percentage point shortfall) to schedule with 80% plus confidence. Project O, a smaller project with moderate variability, plateaus with penalty rates around 100,000. Project J is a relatively predictable project and the level of confidence stabilizes with penalty rates above 75,000.

The call center manager seeks to minimize the cost of staffing, while maximizing the probability of achieving the target service level. These two goals are clearly in conflict and the manager must decide how to balance cost and risk; *a decision obscured in a deterministic optimization approach*.

In the following graphs I recast the data from Figure 4-10 to illustrate this tradeoff. On the left side we see the confidence level of achieving the performance target as a function of staffing cost, and on the right we see the expected service level as a function of staffing cost.

[50] This model requires that the service level is at least 50% in every period based on expected volumes. In order to achieve that level in the busiest period staffing is set such that the service level is above 50% in subsequent periods. This is due to the constraint of scheduling agents to full time shifts.

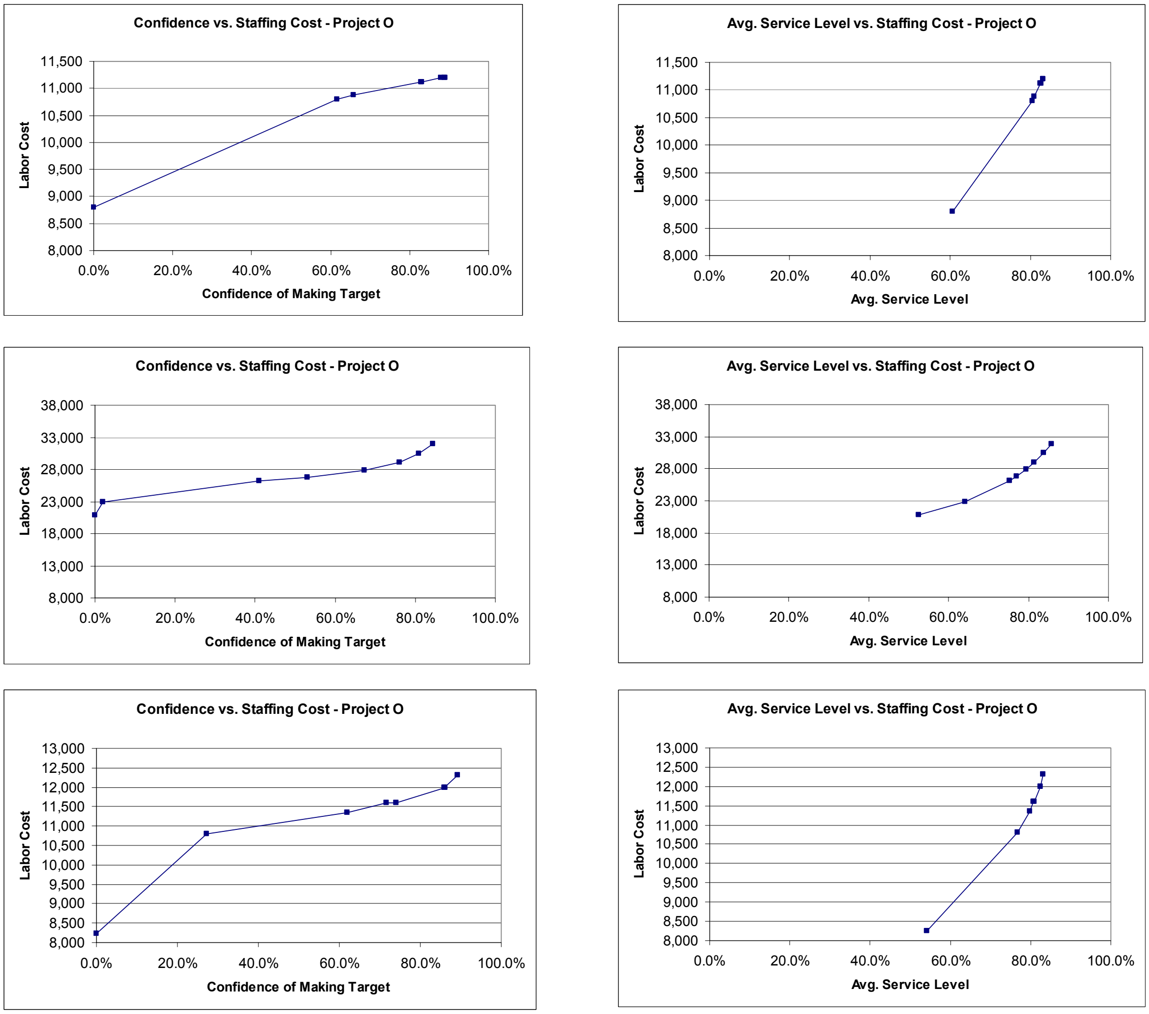


Figure 4-12 Confidence and Expected Service Level

The managerial implications here are important. When making day to day staffing decisions managers must make decisions about how much risk of missing the service level they are willing to tolerate. Conversely, they decide how much insurance to buy in the form of excess capacity. In most situations managers must make these decision based on intuition. The model operationalizes this decision by assigning a financial penalty to the possibility of failing to meet the service level target.

4.5 The Impact of Variability and VSS

4.5.1 Overview

As discussed previously, the solution of the mean value program generates a biased estimate of the true cost of implementing the proposed solution. Solving a stochastic program reduces that bias, and the bias declines with the number of scenarios, going to zero as the number of scenarios goes to infinity (Mak, Morton *et al.* 1999). The expected cost of implementing the stochastic solution is lower than the cost of implementing the mean value solution, or stated differently we can lower the expected cost of operating the system by explicitly considering variability in our optimization problem. This reduction in cost is known as the Value of the Stochastic Solution (VSS). It is easily shown that VSS is a nonnegative quantity, (Birge 1982; Birge and Louveaux 1997)[51]

The following figure depicts the relationship of the various costs.

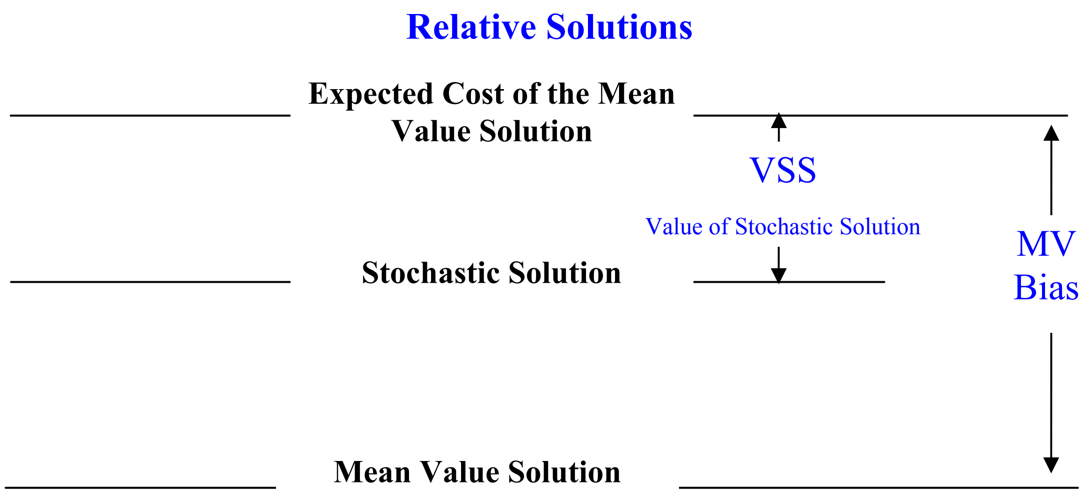

Relative Solutions
Expected Cost of the Mean Value Solution
VSS
Value of Stochastic Solution
Stochastic Solution
MV Bias
Mean Value Solution

Figure 4-13 Relative Cost of Optimal Solutions

[51] The nonnegativity of the VSS implies that we can do no worse on an expected basis by considering variability in the optimization process. VSS may be zero, so we do not necessarily do better by considering variability.

4.5.2 VSS and Solution Convergence

In this section I estimate the bias and VSS for three test projects for various scenario levels. At each scenario level I generate 5 independent batches and solve the program once for each batch. The expected outcome is found by evaluating that solution against 500 evaluation scenarios. The following table summarizes the results.

Project	Scenarios	Direct Cost	Calculated Optimum	Expected Outcome	Solution Bias	VSS	VSS %	Confidence Level
Project J	MV	10,020	10,081	12,838	2,758			1.6%
	10	10,824	10,959	11,253	295	1,585	12.3%	63.5%
	25	10,848	11,044	11,146	121	1,693	13.2%	70.6%
	50	10,868	11,044	11,108	64	1,730	13.5%	74.4%
	100	10,884	11,075	11,092	36	1,747	13.6%	76.8%
Project S	MV	23,200	23,240	34,860	11,620			14.0%
	10	25,400	25,710	28,663	2,953	6,197	17.8%	56.2%
	25	26,720	27,376	27,540	193	7,320	21.0%	84.6%
	50	26,440	27,280	27,496	303	7,364	21.1%	81.2%
	100	26,260	27,069	27,337	304	7,523	21.6%	81.5%
Project O	MV	8,820	8,820	13,855	5,035			69.9%
	10	10,488	10,717	11,079	361	2,776	20.0%	80.2%
	25	10,500	10,844	11,009	199	2,846	20.5%	80.5%
	50	10,388	10,872	10,993	125	2,862	20.7%	80.1%
	100	10,520	10,879	10,956	77	2,899	20.9%	80.8%

Table 4-6 Solution Bias and VSS

In each case I find substantial bias in the Mean Value Solution and find substantial value from implementing the stochastic solution. On the moderately variable project J the stochastic program reduces expected cost by 13%. On the more variable projects S and O, the stochastic solution reduces cost by over 20%. Also note that the stochastic solution provides a higher confidence that the performance target will be achieved.

4.5.2.1 Sampling Bounds

In Section three, I showed that the average solution to the stochastic program provides a point estimate on the lower bound on the true optimal solution, while the average expected outcome of the candidate solution forms a point estimate of the upper bound of the true optimal. In Figure 4-12 I plot the point estimate of the upper and lower solution bounds for Project J at multiple scenario levels, estimated using five batches at each scenario level.

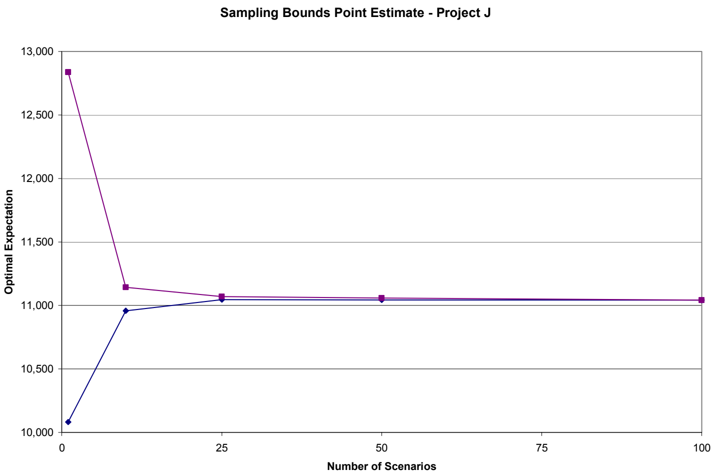


Figure 4-14 Point Estimate of Bounds

Equation (4.23) provides a mechanism to calculate a confidence interval on the optimality gap. In Figure 4-13 I plot the 90% confidence interval on the magnitude of the optimality gap

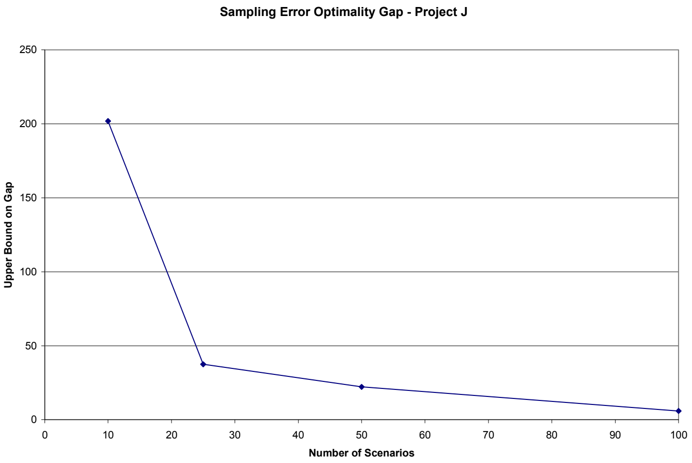


Figure 4-15 Optimality Gap

These graphs show that the mean value problem exhibits significant bias, but that even with a moderate number of scenarios, and a few batches, we are able to generate fairly tight bounds on the true optimal value. The data suggests that solving the problem with as few as 25 scenarios provides reasonably good results, while a 50 or 100 scenario model gives us a tighter bound that may be useful when trying to make detailed comparisons between alternatives.

For each project listed in table 6-1 the stochastic program lowers overall expected cost by increasing direct labor. It is somewhat paradoxical that stochastic programs provide better results by calculating worse objective functions. The intuition is however straightforward; *deterministic optimization programs assume away uncertainty and therefore do not adequately hedge for variability*.

Figure 4-16 compares the schedules generated from a mean value program and a stochastic program. The stochastic program adds incremental staffing at various points throughout the day. Figure 4-17 shows a 90% confidence interval for the calls received by period. Comparing that graph to Figure 4-16 we see that incremental staffing is added in periods with relatively high volumes and high variability.

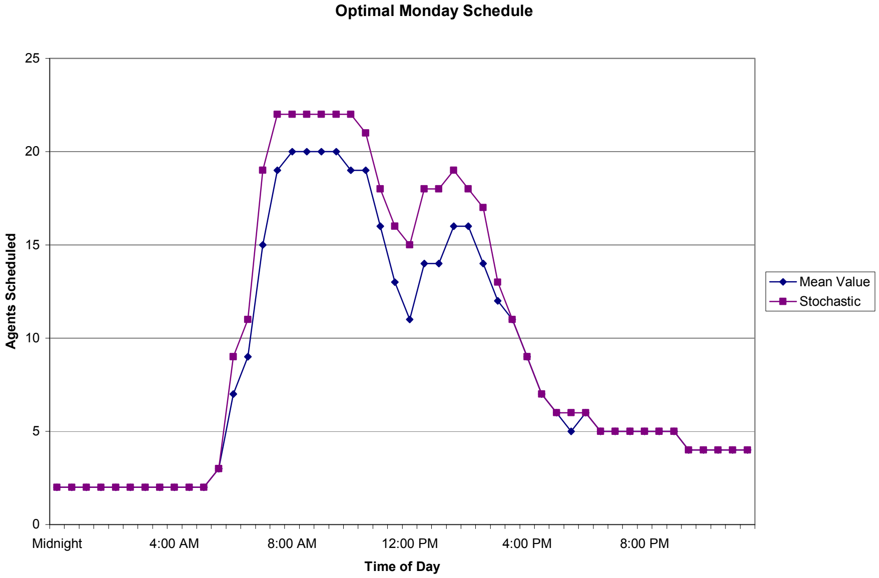


Figure 4-16 Comparison of 2 schedules

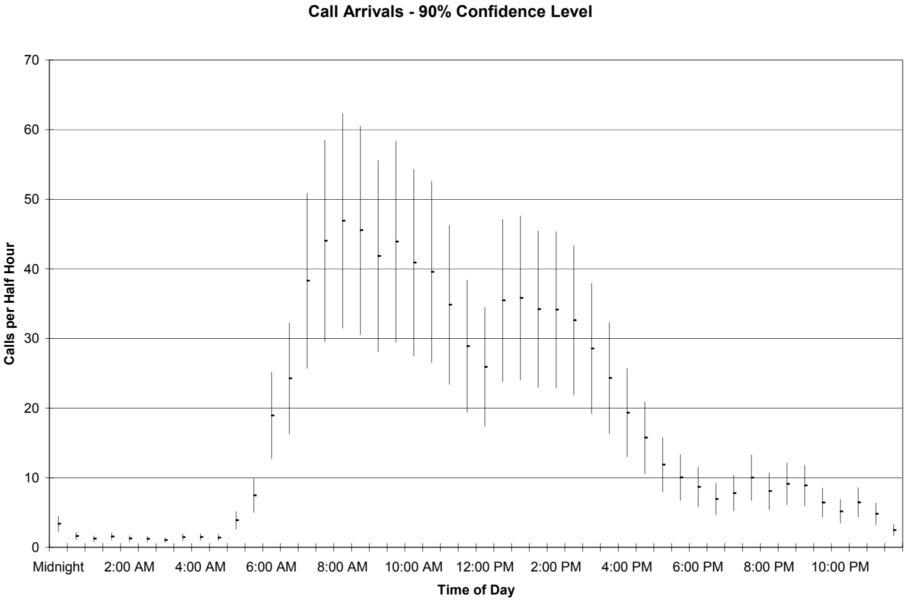


Figure 4-17 Confidence Interval for per period calls

Figure 4-16 shows the mean value and stochastic solution for Monday, the busiest day of the week. In Figure 4-18 I plot the incremental staffing generated by the stochastic solution over the course of the week. We see that the stochastic model adds incremental capacity during the busy periods of most days, but reduces staffing in some low volume periods.

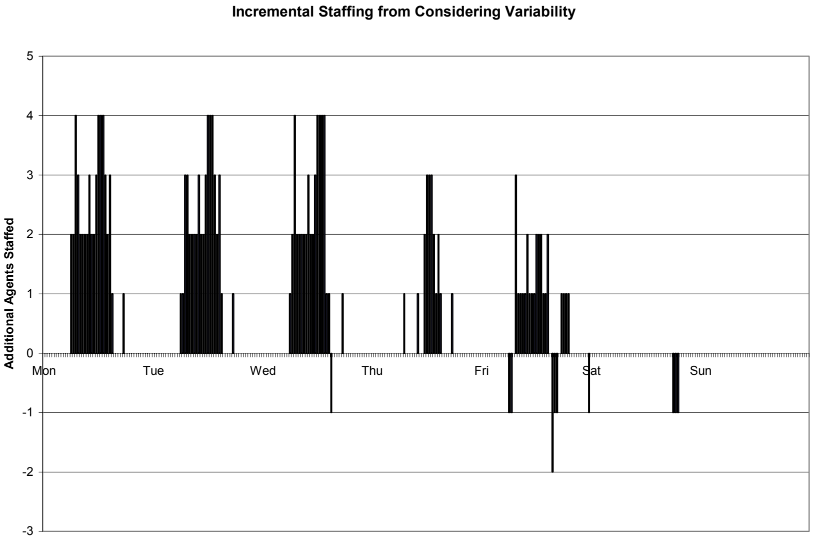


Figure 4-18 Mean Value vs. Stochastic Staffing

4.5.3 Impact of Variability

In the prior analysis I calculated schedules for models of several real world projects and examined the convergence properties of the solution. I examined the differences between the mean value solution and the stochastic solution and showed that the stochastic schedule adds extra capacity to buffer against uncertainty. The analysis showed that VSS varies from project to project and the data suggests that for projects with higher variability the stochastic solution diverges from the Mean Value Solution more significantly.

In this section I conduct a controlled experiment to assess the impact of variability more directly. While I still base the analysis on a specific project, I manipulate key parameters to determine the impact of variability on the resulting schedule. Specifically, I analyze a series of alternative project configurations for which the expected number of calls, and the average seasonality pattern are based on project J, but I manipulate key environment and policy variables.

4.5.3.1 Experimental Design

To assess the impact of variability I will conduct a controlled experiment that adjust factors related to variability as well as the required service level quality, Specifically I will conduct an experiment using the following factors

- **Daily CV Scale**: the variability of daily arrivals is adjusted by scaling the coefficient of variation for day of week effects; the mean is held constant and standard deviation is adjusted to achieve the scaled CV.
- **Time Period CV Scale**: the same variability scaling is performed on the Time of Day effect.
- **Service Level Requirement**: a *loose* SLA (70/120) and a *tight* SLA (90/30).
- **Service Level Penalty**: different penalty costs for failing to achieve the specified service level target.
- **Shock Probability**: arrivals with and without shocks. In the case of shocks I scale down the non-shock volume so that the expected call volume is constant across all design points.

I created an experiment with 16 design points as defined below

	A	B	C	D	E
1	-	-	-	-	+
2	+	-	-	-	-
3	-	+	-	-	-
4	+	+	-	-	+
5	-	-	+	-	-
6	+	-	+	-	+
7	-	+	+	-	+
8	+	+	+	-	-
9	-	-	-	+	-
10	+	-	-	+	+
11	-	+	-	+	+
12	+	+	-	+	-
13	-	-	+	+	+
14	+	-	+	+	-
15	-	+	+	+	-
16	+	+	+	+	+

Factor Definitions		-	+
A	Daily CV Scale	0.75	1.25
B	Time Period CV Scale	0.75	1.25
C	Service Level Requirement	70/120	90/30
D	Shock Probability	0%	5%
E	Service Level Penalty	50,000	150,000

Table 4-7 Impact of Variability Experimental Design

This is a 2_{V}^{5-1} fractional factorial design and contains 16 design points. This design has a resolution of V, which allows us to estimate all the main effects and all the two way interaction effects. Higher level interactions are confounded and can not be estimated independently.

4.5.3.2 Experimental Results

To conduct this experiment I generate 5 batches of 50 scenarios and a single evaluation batch of 500 scenarios at each of the 16 design points. I solve the optimization problems for each batch, calculating a candidate solution which is evaluated against the 500 scenario evaluation batch to calculate expected outcomes. Based on these solutions I calculate the following response variables:

- **Labor cost**: the cost of direct labor in the candidate solution.
- **Expected Outcome**: the labor and penalty cost found when evaluating the candidate solution.
- **TSF Cushion**: the difference between the expected TSF found when evaluating the candidate solution, and the SLA performance goal.
- **Confidence**: the proportion of evaluation scenarios for which the service level target is achieved.

This approach generates 5 samples for each response. The results of this analysis are presented in the following table. Recall that all design points in this experiment have the same expected call volume.

						Average				Standard Deviation	
						Labor	Expected	TSF		Labor	Expected
	A	B	C	D	E	Cost	Outcome	Cushion	Confidence	Cost	Outcome
1	-	-	-	-	+	8,992	9,097	3.9%	93.3%	75.6	40.9
2	+	-	-	-	-	9,048	9,210	3.9%	85.0%	85.6	20.3
3	-	+	-	-	-	9,056	9,170	2.1%	83.2%	51.8	16.1
4	+	+	-	-	+	9,524	9,616	5.6%	95.9%	91.0	51.3
5	-	-	+	-	-	12,404	12,856	-0.4%	41.7%	138.1	53.7
6	+	-	+	-	+	13,200	13,520	1.9%	84.3%	154.9	45.7
7	-	+	+	-	+	13,440	13,969	0.7%	71.2%	105.8	62.5
8	+	+	+	-	-	13,408	14,002	-0.4%	47.5%	100.6	31.7
9	-	-	-	+	-	8,836	8,942	2.5%	83.8%	43.4	15.9
10	+	-	-	+	+	9,216	9,349	5.6%	94.1%	69.9	7.0
11	-	+	-	+	+	9,128	9,347	2.9%	88.7%	22.8	20.0
12	+	+	-	+	-	9,248	9,465	3.3%	80.8%	68.7	12.0
13	-	-	+	+	+	12,748	13,027	1.2%	80.2%	136.1	56.0
14	+	-	+	+	-	12,692	13,117	0.2%	58.9%	156.6	13.3
15	-	+	+	+	-	13,168	13,481	0.0%	54.7%	128.5	47.7
16	+	+	+	+	+	13,332	13,855	-0.1%	51.8%	136.8	11.8

Table 4-8 Impact of Variability Experimental Results

4.5.3.3 Analysis of Results

The resolution V experimental design allow us to calculate the main effects; the impact of moving each factor from it's low to high value, as well as first level interaction terms; the interaction of each unique pair of factors. Given the orthogonal nature of the experimental design all factors are perfectly uncorrelated and we have no issue of multicolinearity in our analysis.

The following table summarizes the estimated main and first level interaction effects for each of the response variables. The main effects represent the average change in the response when the factor is changed from its low value to its high value. The interaction effects estimate the impact of factors that have a coupled influence upon the response beyond their main effects, they are calculated as one half of the average difference in response when both factors change together (Box, Hunter *et al.* 2005)[52]. Only those effects that are statistically significant at the 0.01 level are displayed.

[52] If both factors are at the same level, the interaction term is added to the estimated outcome. If they are at opposite levels the interaction term is subtracted from the estimated outcome. So for example, the BC

		Expected	SL	
Factor Effects	Labor	Outcome	Cushion	Confidence
Intercept	11,091	11,378	2.0%	74.4%
A - Daily CV	259	282	1.0%	
B - Time Period CV	395	474	-0.6%	-6.0%
C - Service Level Req	3,919	4,207	-3.3%	-27.4%
D - Shock Prob	-87	-104		
E - SL Penalty	211	189	1.3%	15.7%
A*B	-122	-41		-10.9%
A*C			-0.8%	
B*C	179	224		-8.3%
A*D	-126	-64		-8.3%
B*D		-45		-8.8%
C*D		-106		
A*E		-55		
B*E		-48		
C*E		34		11.8%
D*E	-99	-52	-0.9%	-12.3%

Table 4-9 Impact of Variability on Expected Outcome - Main Effects

This table presents a considerable amount of information. Some key observations include:

- The average cost of operating this call center is $11,378 per week, but the realized cost varies considerably.
- The most influential cost driver is the service level requirement, increasing the service level requirement adds about $4,000, or 50% to the cost of operations.
- Variability has a substantial impact on the cost of delivery, but the impact is influenced by the SLA regime.
  - In a loose SLA environment (C - , E -) increased daily variability increases costs by about 6%. Increased time variability increases cost by 2.3%, together they increase cost by 6.6%
  - In a tight SLA environment (C - , E -) increased daily variability increases costs by about 2.5%. Increased time variability increases cost by 8.3%, together they increase cost by 9.4%
- On average the optimal staffing decision staffs the project so that the expected service level is 2% above the requirement, which results in a 74% confidence level. However in the tight SLA regime, the cost meeting a high service level cause the cushion and confidence level to drop significantly.

interaction term increases the labor estimate by $179 if both factors have the same setting. If one is high and the other low the estimate is reduced by $179.

4.6 Staffing Flexibility

4.6.1 Overview

One of the operational challenges associated with the type of call center analyzed here is that demand is often more variable then capacity. The arrival pattern shown in Figure 2-24, for example, has a large spike in demand between 8 and 11 AM. In order to efficiently match supply and demand we would like to create a corresponding spike in capacity at the same time. Accomplishing this with full time staffing can be difficult. However, in practice some call centers are often staffed exclusively with full time resources; that is that resources scheduled to 40 hour per week schedules.

Managers have multiple reasons for hiring only full time agents. Full timers are believed to be less expensive to train because their hiring and training cost is amortized more quickly. Many managers also believe that full time agents will learn faster and thus be more productive than one working part time by being exposed to more calls. Some managers also believe that part time agents are more difficult to recruit and retain[53]. The potential savings from using part time resources is of interest, from a both a practical and research perspective. I examine that issue in this section.

In Section 4.6.2 I develop alternative scheduling patterns and develop a range of flexibility options. In section 4.6.3 I develop a conceptual framework for evaluating the cost of staffing flexibility. In section 4.6.4 I perform a numerical experiment to calculate the cost of staffing under each scheduling pattern for 3 model projects. In section 4.6.5 I perform a related experiment that look at the implications of limiting the availability of part timers.

4.6.2 Types of Staffing Flexibility

In this section I review what I mean by flexibility and define different levels of staffing flexibility. Conceptually staffing flexibility implies the ability to schedule resources as necessary to closely match capacity with demand; unrestricted by constraints on possible schedules. Constraints may include union rules on feasible schedules, restrictions on starting time or days

[53] The company I worked with uses full time agents almost exclusively for all the reasons cited above. In addition their human resource policies restrict benefits to full time agents making it hard for them to retain part time resources if they should hire them.

off. For the purpose of this analysis I focus on the constraint imposed by full time staffing. I consider a work force to be more flexible the more options we have to schedule resources to work part time shifts if that is what the demand pattern dictates.

I consider two types of part time resources:

- **Full Shifts**: full time shifts, less than five days per week
- **Partial Shifts**: shifts of less then eight hours, five days a week

Based on this I define the following potential shift patterns:

- 5 x 8: 5 days a week, 8 hours a day (40 hr week)
- 4 x 10: 4 days a week, 10 hours a day (40 hr week)
- 4 x 8: 4 days a week, 8 hours a day (32 hr week)
- 5 x 6: 5 days a week, 6 hours a day (30 hr week)
- 5 x 4: 5 days a week, 4 hours a day (20 hr week)

In each case I assume that a shift can start during any half hour period, for a total of 48 starting times per day. I also assume a full complement of daily work patterns that require a two consecutive day off policy. For five day a week schedules this implies only seven feasible day patterns. For a four day a week schedule there are 28 day patterns that satisfy the two consecutive day off constraint.

Based on this I define the following set of schedule patterns:

Pattern	Schedule Types Included	Feasible Schedules
A	5x8 only	336
B	5x8, 4x10	1,680
C	5x8, 4x10, 4x8	3,024
D	5x8, 4x10, 4x8, 5x6	3,360
E	5x8, 4x10, 4x8, 5x6, 5x4	3,696

Table 4-10 Scheduling Patterns

For patterns A-E I incrementally add more flexibility into the set of available schedules. Table 4-9 illustrates the combinatorial problem associated with evaluating multiple scheduling patterns. As we move from five day a week, 8 hour a day staffing in pattern A to the multiple options of pattern E the number of possible schedules, and the corresponding number of integer variables, increases five fold.

4.6.3 The Value of Flexibility

In the previous section I outlined a number of different scheduling options and developed a set of over 2,700 schedules from which to choose. We know from basic optimization theory that adding more schedule options can make our objective no worse, and I argued qualitatively that adding part time shifts will improve the objective function. However, the choices made in putting this list together are somewhat arbitrary. We restricted the set of schedules to those that included at least 20 hours and 4 working days, but perhaps we should consider two ten hour shifts, or six three hour shifts. The number of possible shift patterns is unlimited. In this section I attempt to define a lower bound on the cost reduction that can be achieved via flexible staffing, and in the process develop a framework for categorizing the costs associated with service delivery.

Assume that calls arrive via a non-homogeneous Poisson process with a known arrival rate in each 30 minute period. Also assume that we have the option of scheduling any integral number of servers (agents) in each 30 minute period, independent of any other 30 minute period, as if we could schedule workers to 30 minute shifts. We could then make an independent staffing decision in each period, and because of the concave nature of the TSF curve each period would be staffed to achieve a service level near the goal[54]. We call this staffing level the ideal or *maximum flexibility* staffing model. The cost of providing this staff level, in dollars or person hours, represents the minimal cost required to deliver the required service.

Now assume that we relax the assumption of known arrival rates and allow call volume to vary stochastically. Incremental staffing will be required to hedge against uncertainty and the cost of service delivery will increase. I refer to this incremental cost as the *cost of load uncertainty*.

In reality, we can not make independent staffing decisions in each period. Workers are scheduled in shifts so the staffing level in any period is not independent of the staffing level in the neighboring periods. If we now relax the assumption of maximum flexibility and instead pick scheduling pattern from table 4-5, then we have the *cost of shift constrained staffing*. I call the

[54] Because of the integrality constraint on the number of servers TSF would still vary moderately from period to period.

difference between these two staffing costs the *cost of staffing inflexibility*; it is the additional cost of service delivery due to shift constraints. The following figure illustrates this relative costs.

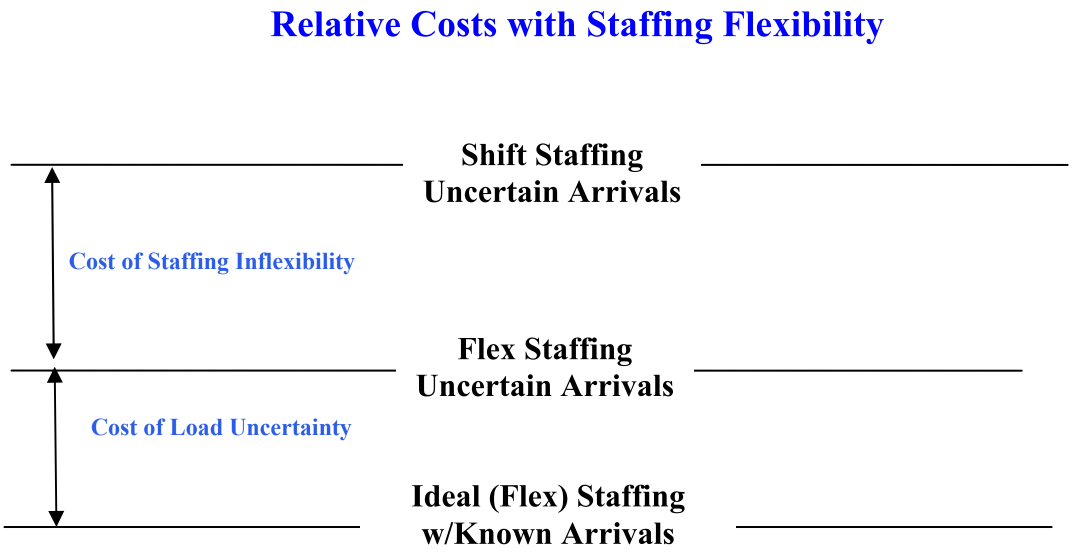


Figure 4-19 Relative Costs of Staffing Flexibility

4.6.4 Numerical Experiment – Part Time Staffing

In this section I perform a numerical experiment to estimate the cost of service delivery under various levels of staffing flexibility. I wish to examine how the schedule changes, qualitatively and quantitatively, as we increase the level of flexibility of the workforce. For each project I calculate the staffing cost and expected outcome for each scheduling option listed in Table 4-9. I evaluate the savings in service delivery cost at each level from the baseline, and also look at how each level compares to the max flex option. In this analysis I assumed agents are paid $10 per hour, regardless of the schedule to which they are assigned.

For each project and schedule set combination I optimized against five batches of 50 scenarios each and computed the average for all performance metrics. Each candidate solution was evaluated against the same set of 500 evaluation scenarios to calculate the expected outcome.

4.6.4.1 Project J

In this experiment I analyze a model based on project J. The following table summarizes the results for each scheduling option.

				Average				
	Sched	Labor	Calculated	Expected	Average			Cost of
DP	Set	Cost	Objective	Outcome	TSF	Confidence	Bias	Inflexibility
1	A	11,280	11,679	11,660	81.1%	66.1%	-19.6	696
2	B	10,800	11,204	11,239	80.4%	59.8%	34.9	275
3	C	10,944	11,197	11,235	81.3%	71.0%	37.3	271
4	D	10,844	11,083	11,103	81.5%	73.2%	20.6	139
5	E	10,720	10,976	11,019	81.3%	70.5%	42.6	55
6	MF	10,677	10,867	10,964	81.2%	70.8%	96.6	-
7	MF-MV	9,845	9,859	12,544	74.6%	3.6%	2,685	-
	Cost of Load Uncertainty			1,105				

	% Savings by Flexibility			% of Max Savings Achieved		
Sched	Labor	Calculated	Expected	Labor	Calculated	Expected
Set	Cost	Objective	Outcome	Cost	Objective	Outcome
A						
B	4.3%	4.1%	3.6%	79.6%	58.5%	60.5%
C	3.0%	4.1%	3.6%	55.7%	59.3%	61.1%
D	3.9%	5.1%	4.8%	72.3%	73.5%	80.0%
E	5.0%	6.0%	5.5%	92.9%	86.6%	92.1%
MF	5.3%	7.0%	6.0%	100.0%	100.0%	100.0%

Table 4-11 Impact of Flexible Scheduling- Project J

The data shows for this arrival pattern flexibility can lower the cost of service delivery considerably. Simply adding full time 4x10 shifts lowers total cost by 4% as the weekly seasonal pattern is better matched. Adding part time shifts allows time of day seasonality to be better matched and lowers cost by an additional 1.8%. The max flex schedule is 6.1% less expensive then the 5x8 schedule, but most of that savings can be achieved with less flexible options. Over half the total possible savings are achieved simply with 4x10 schedules, and in schedule set E we are able to achieve fully 96% of the total possible savings.

To gain insight into how flexibility alters the optimal schedule we can examine the results graphically. In the following graphics I plot the Monday schedule for each shift pattern. (Note: the schedule was calculated by optimizing over the full week, we show only the Monday schedule in this graphic. On the next page I show the full week.)

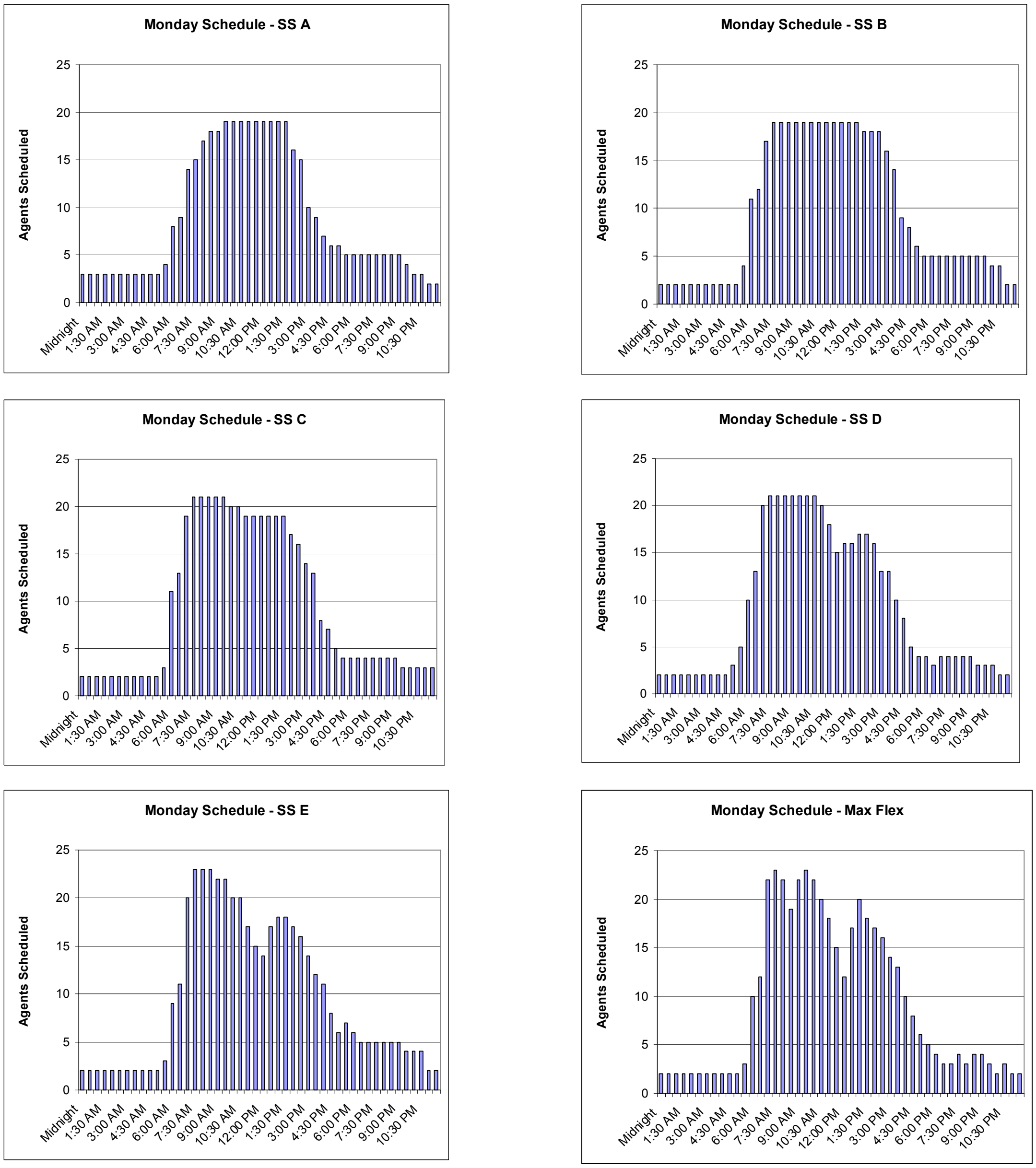


Figure 4-20 Impact of Part Timers on Daily Schedule - Project J

We can make some observations about the evolution of the optimal schedule as we add more flexibility in to the set of possible schedules:

- With 5x8 staffing only we maintain a relatively flat level of staffing throughout the busy period. The staffing level is a compromise between the busier morning and slower afternoon, set to balance out to a service level 80% over the course of the week. We will tend to be understaffed in peak periods and overstaffed in slower periods during the primary busy period.
- With full shift schedules (B,C) we see little change in this pattern and staffing remains relatively flat over the course of the busy period.
- With the introduction of shorter schedules (D,E) capacity becomes more variable throughout the day. With 6 hour shifts we start to see the double hump pattern that characterizes arrivals repeated in the capacity plot. With 4 hour shifts the pattern becomes more pronounced as morning staffing increases and afternoon staffing decreases.
- With maximum flexibility shifts in staffing become more pronounced as the model attempts to match the shape of the arrival pattern as closely as possible.

In the following graphic we look at the schedule over the course of the week to see how we are able to address weekly seasonality.

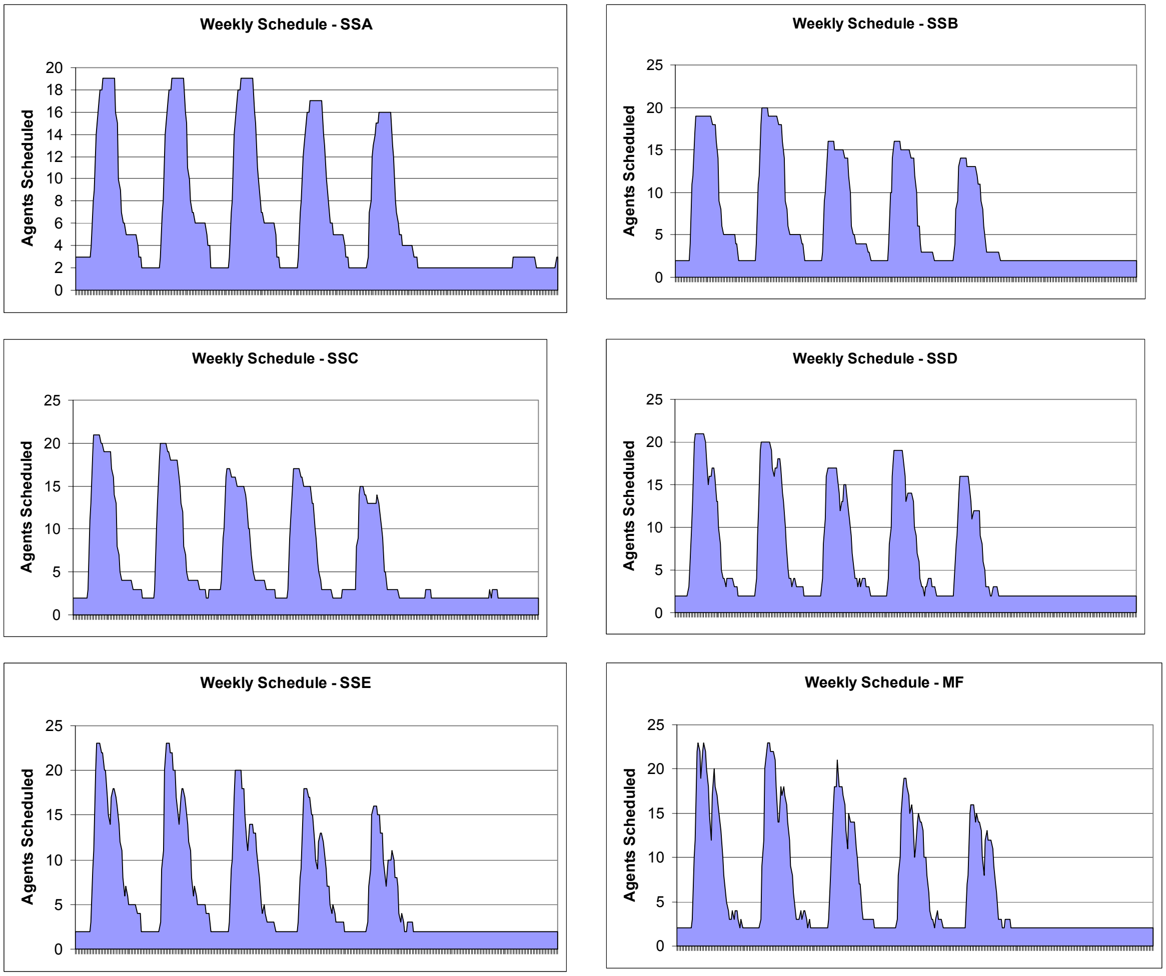


Figure 4-21 Impact of Part Timers on Weekly Schedule Project J

- With 5x8's only weekly staffing is fairly constant over the course of the week with only a small number of resources peeled off of Wednesday's and Thursdays to meet weekend demand.
- 4x10 staffing allows a better matching of the weekly pattern and we see an uneven staffing profile over the course of the week.
- With 4x8 staffing (C) the weekly staffing pattern better matches the arrival patterns and capacity declines steadily throughout the week.

- Adding short shifts (D-E) makes little change in the aggregate weekly capacity profile, the changes are primarily made to better match the intraday seasonality.

4.6.4.2 Project S

I then repeated the analysis for project S.

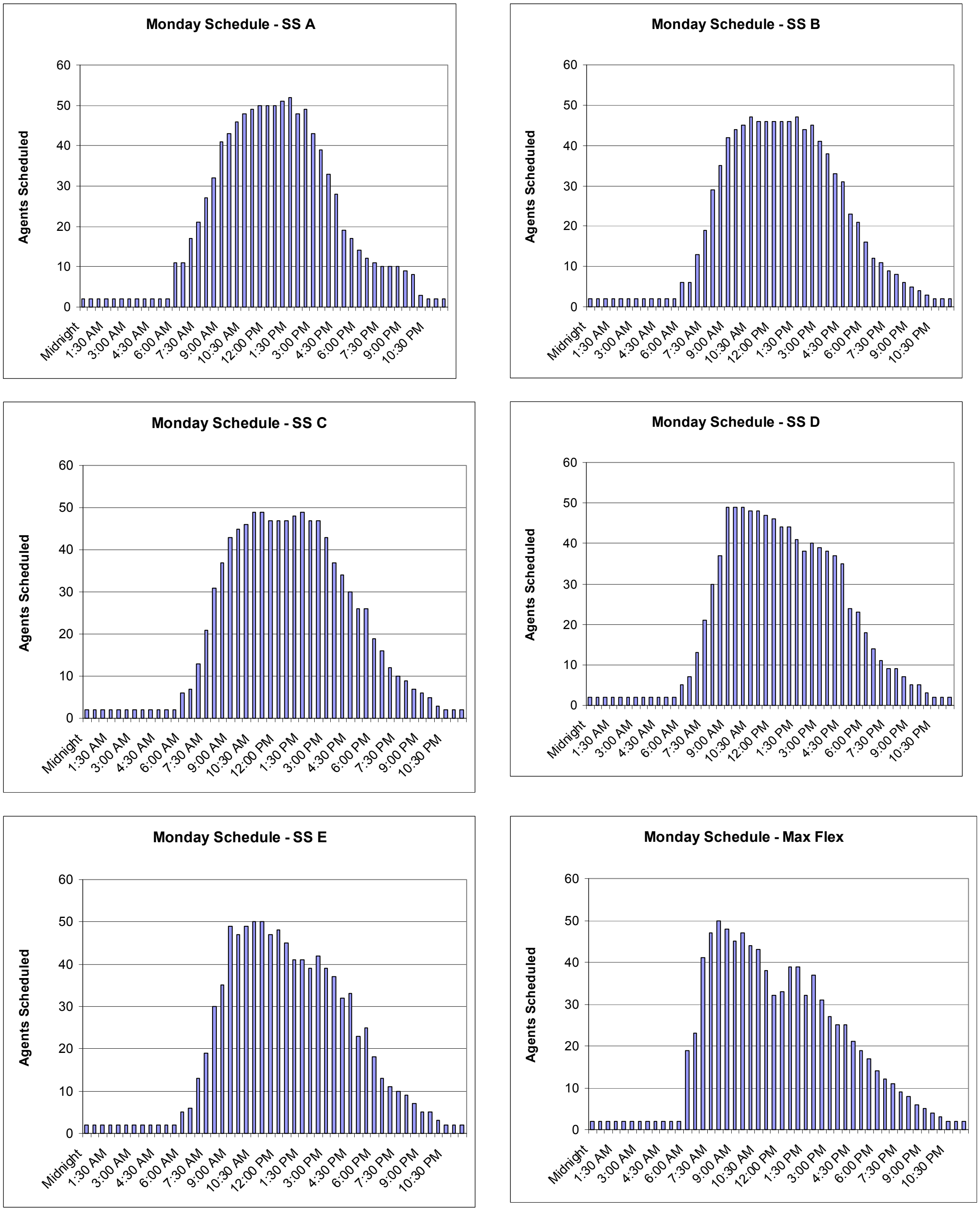
				Average				
	Sched	Labor	Calculated	Expected	Average			Cost of
DP	Set	Cost	Objective	Outcome	TSF	Confidence	Bias	Inflexibility
1	A	30,960	34,238	35,305	83.2%	80.5%	1,067	6,369
2	B	30,320	33,597	34,728	83.7%	81.3%	1,132	5,793
3	C	30,384	33,639	34,733	83.6%	81.0%	1,094	5,797
4	D	30,092	33,398	34,585	83.5%	80.6%	1,187	5,649
5	E	30,096	33,407	34,595	83.5%	80.2%	1,189	5,659
6	MF	25,427	27,458	28,936	74.3%	36.0%	1,478	-
7	MF-MV	24,040	24,079	33,654	60.8%	0.2%	9,575	-

Cost of Load Uncertainty 4,857

% Savings by Flexibility			% of Max Savings Achieved		
Labor Cost	Calculated Objective	Expected Outcome	Labor Cost	Calculated Objective	Expected Outcome
2.1%	1.9%	1.6%	11.6%	9.5%	9.0%
1.9%	1.7%	1.6%	10.4%	8.8%	9.0%
2.8%	2.5%	2.0%	15.7%	12.4%	11.3%
2.8%	2.4%	2.0%	15.6%	12.3%	11.1%
17.9%	19.8%	18.0%	100.0%	100.0%	100.0%

Table 4-12 Impact of Flexible Scheduling- Project S

Shown below are the Monday schedules for this project.



Monday Schedule - SS A

Monday Schedule - SS B

Monday Schedule - SS C

Monday Schedule - SS D

Monday Schedule - SS E

Monday Schedule - Max Flex

Figure 4-22 Impact of Part Timers on Daily Schedule - Project S

Shown here are the weekly patterns.

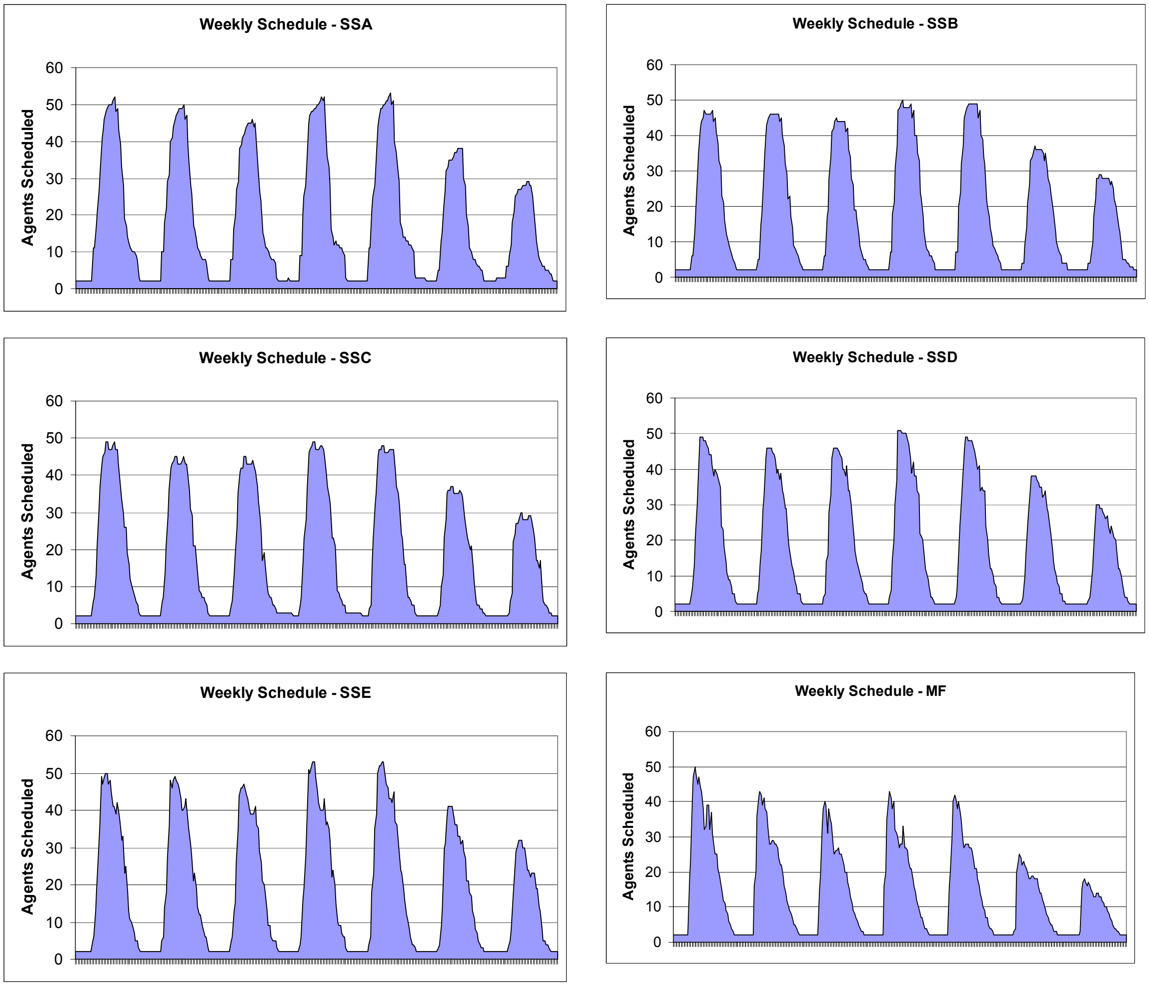


Figure 4-23 Impact of Part Timers on Weekly Schedule - Project S

Both the data and the graphs and the numbers reveal that part time staffing is less effective for this project. Some possible reasons are summarized below:

- The project has a less well defined seasonal pattern.
- Greater call volume on weekends allows better tailoring of the mid week staffing profiles with full time resources.
- The highly volatile nature of the project makes capacity shaping less effective.

4.6.4.3 Project O

Finally I examined a third project patterned off of Project O. The results are similar to project S Here are the financial results

				Average				
	Sched	Labor	Calculated	Expected	Average			Cost of
DP	Set	Cost	Objective	Outcome	TSF	Confidence	Bias	Inflexibility
1	A	11,600	12,254	12,443	80.2%	66.3%	188.8	236
2	B	11,360	12,020	12,257	80.1%	64.5%	236.9	50
3	C	11,296	12,044	12,278	79.5%	58.4%	233.9	71
4	D	11,352	11,967	12,210	80.2%	66.9%	243.2	3
5	E	11,316	11,951	12,226	79.9%	62.8%	274.6	19
6	MF	11,287	11,856	12,207	79.8%	62.0%	351.2	-
7	MF-MV	9,275	9,275	15,662	67.2%			-
	Cost of Load Uncertainty			2,932				

% Savings by Flexibility			% of Max Savings Achieved		
Labor Cost	Calculated Objective	Expected Outcome	Labor Cost	Calculated Objective	Expected Outcome
2.1%	1.9%	1.5%	76.7%	58.6%	78.6%
2.6%	1.7%	1.3%	97.1%	52.7%	69.9%
2.1%	2.3%	1.9%	79.2%	72.0%	98.6%
2.4%	2.5%	1.7%	90.7%	76.1%	92.1%
2.7%	3.2%	1.9%	100.0%	100.0%	100.0%

Table 4-13 Cost of Schedule Patterns – Project O

Again it would appear that the lack of a strong seasonal pattern, either weekly or daily; makes the part time strategy less effective. The strategy does however provide non-trivial benefit, boosting utilization and cutting costs.

4.6.5 Incremental Value of Part-timers

In this next experiment I examine the benefit potential from flexible staffing if the number of part time workers is limited; either by policy or availability. I continue to solve the stochastic optimization program but with an additional constraint that limits the number of agents assigned to a shift of less then 40 hours to some parameter. I then vary that parameter from 0 to 20, running 5 batches at each level and computing the resulting costs.

In the following graph I plot the expected cost of operation as a function of the maximum number of allowable part time workers.

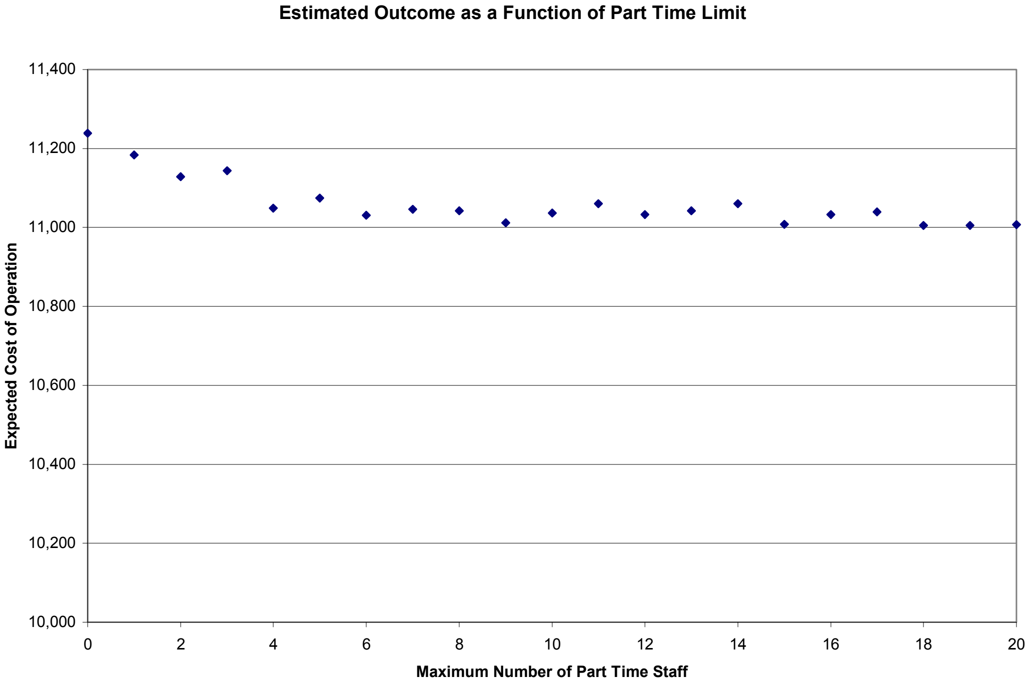


Figure 4-24 Incremental Value of Part Time Staff

The graph reveals that the benefit of part time staffing comes from the first few part timers. With zero part timers this schedule is equivalent to schedule B with an estimated cost of operation of about $11,300. As part timers are added the schedule evolves toward schedule E and an estimated cost of operation of approximately $11,000. The graph shows that the full benefit is achieved with about 5 part time workers, or approximately 16% of the workforce. Beyond 5 workers the change in the estimated outcome is statistical noise.

This illustrates an important point I will return to in the conclusions in Chapter 7. While flexibility greatly improves the efficiency of the system, only a limited amount of flexibility is needed. A small number of flexible workers is all that is required to achieve the bulk of the benefit.

4.7 Comparison with the Common Practice

4.7.1 Introduction

Throughout this chapter we have analyzed a model that utilizes the Erlang A model, a model that includes abandonment, and arrival rate uncertainty. Neither of these conditions are included in industry standard models; "*common practice uses the M/M/N (Erlang C) queuing model to estimate the stationary system performance of short – half hour or hour – interval.*" (Gans, Koole *et al.* 2003) p.92. Furthermore, standard industry practice is to make staffing decisions based on a period by period (local) service level requirement; "*each half hour interval's forecasted λ_i and μ_i give rise to a target staffing level for the period. ... determination of optimal set of schedules can then be described as the solution to an integer program*" (Gans, Koole *et al.* 2003) p.93.

In section 4.5.2 I showed that ignoring arrival rate uncertainty leads to verifiably more expensive solutions, on an expected cost basis, than models which account for variability. In this section I compare the stochastic Erlang–A model to the commonly applied known arrival rate Erlang C model.

4.7.2 Weighted Set Covering Model

The standard approach described above generates a set of fixed staffing requirements in each period, and then attempts to find the lowest cost schedule to satisfy these requirements. The resulting integer program is a standard weighted set covering problem which can be expressed as

$$\min \sum_{j \in J} c_j x_j \tag{4.27}$$

subject to

$$\sum_{j \in J} a_{ij} x_j \geq b_i, \quad \forall i \in I \tag{4.28}$$

$$x_{ij} \in \mathbb{Z}^{+} \tag{4.29}$$

Where c_j is the cost of the j^{th} schedule, x_j is the number of resources assigned to the j^{th} schedule, and a_{ij} is the mapping of schedules to time periods.

4.7.3 Locally Constrained Erlang C Model

I refer to the standard approach described in (Gans, Koole *et al.* 2003) as the locally constrained Erlang C model because it uses Erlang C to generate a local constraint in each period[55]. I construct the locally constrained Erlang C schedule using the following process:

1. Calculate the average volume in each 30 minute period of the week.
2. Using the volumes calculated in step 1, determine the number of agents required to achieve the target service level in each 30 minute period by performing a search using equation (3.16).
3. Set the period staffing requirement to the maximum of the number calculated in step 2 and the global minimal staffing requirement.
4. Using the resulting vector of staffing requirements as the requirement parameter b_i in the IP (4.27) - (4.29).

The general problem with this approach is the constraint created by the per period service level requirement, coupled with the requirement to schedule resources in shifts. The peak staffing level is set by the peak arrival period, and depending on the length of the arrival peak, and the length of the flexibility of the staffing model, a substantial amount of excess capacity may be created in other periods. I refer to the extra capacity created in other periods as the *deadweight loss*[56], the extra man-hours scheduled due to the shift constraint. The magnitude of the deadweight loss will be a factor of the flexibility of the available set of schedules. With more flexible staffing options, the weighted set covering algorithm can match the requirement more closely.

Consider the examples shown in the following two graphs. In each graph the inner region defines the requirements generated for the set covering problem. The envelope of the graph represents the total staffing assigned by solving the set covering problem. The outer region therefore represents the excess capacity assigned above and beyond what was specified.

[55] In this context local refers to a period by period constraint, and global refers to a constrained applied over a longer period of say a week or a month. This is generally accepted terminology, see for example Koole, van der Sluis (2003).

[56] The term deadweight loss is borrowed from the economics literature where it refers to the loss of economic efficiency from an equilibrium that is not Pareto efficient. It is often used to quantify the loss of efficiency created by taxes. In this case I use the term to refer to the loss of efficiency from the optimal solution created by shift constraints.

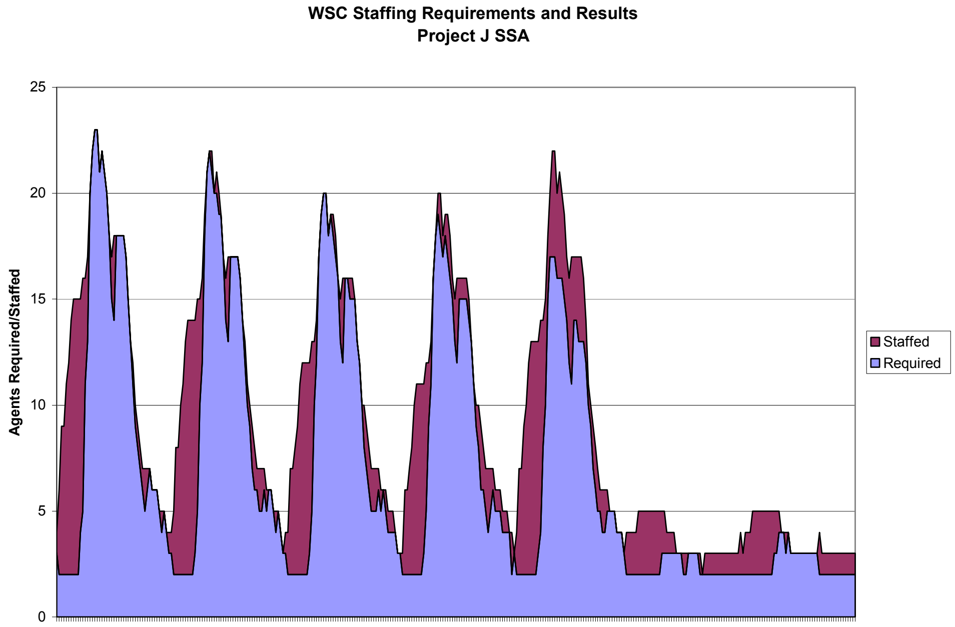


Figure 4-25 WSC Excess Staffing – Project J -Schedule Set A

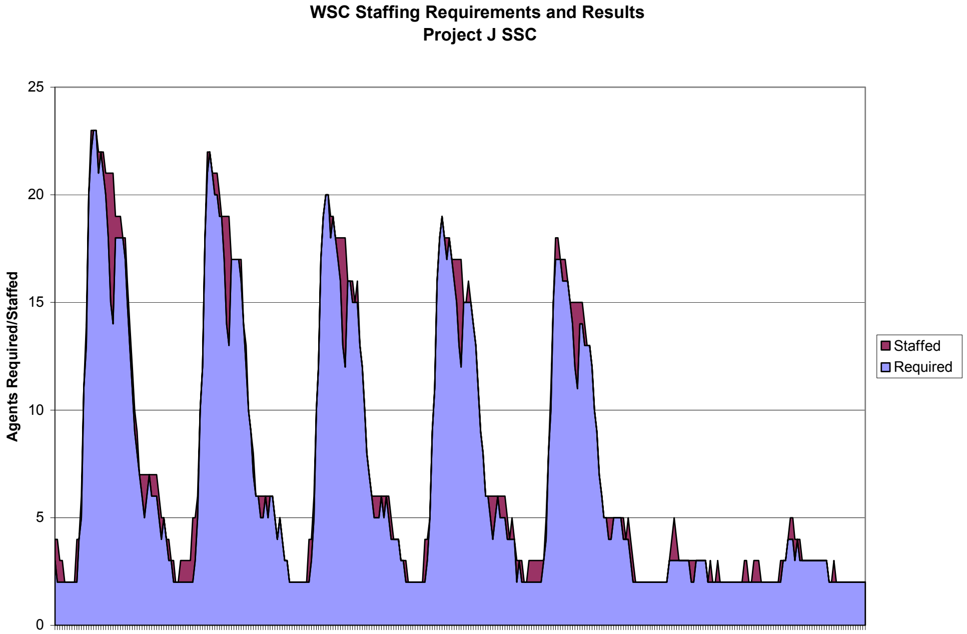


Figure 4-26 WSC Excess Staffing – Project J - Schedule Set C

In Figure 4-24 we can only assign resources to full time 4x8 schedules and so the set covering is poor. The graph shows a significant amount of overstaffing throughout the course of the week. In Figure 4-25 we have the option of 4x10 and 4x8 shifts so we can match the required demand much more closely.

To quantify the impact I ran a locally constrained Erlang C model for each of the three test projects for each of the 5 schedule sets. The per-period constraints are set so that the service level with expected volumes is at least 80%. In the following table I compare the results of this analysis with the results of the stochastic schedules generated in section 4.6.

	Locally Constrained Erlang C						SCCS - Erlang A								
	Direct	Expected	Expected	Average		DWL	Direct	Expected	Expected	Average	Direct Labor		Expected	
	Labor	Penalty	Outcome	TSF	DWL	%	Labor	Penalty	Outcome	TSF	Savings		Savings	
Project J														
Sched A	16,000	0	16,000	91.8%	4,055	34%	11,280	380	11,660	81.1%	4,720	29.5%	4,340	27.1%
Sched B	13,200	0	13,200	91.0%	1,255	11%	10,800	439	11,239	80.4%	2,400	18.2%	1,961	14.9%
Sched C	12,880	0	12880	90.4%	935	8%	10,944	291	11,235	81.3%	1,936	15.0%	1,645	12.8%
Sched D	12,500	0	12500	89.5%	555	5%	10,844	259	11,103	81.5%	1,656	13.2%	1,397	11.2%
Sched E	12,300	0	12300	89.2%	355	3%	10,720	299	11,019	81.3%	1,580	12.8%	1,281	10.4%
Project S														
Sched A	38,000	1,565	39,565	91.6%	8,340	28%	30,960	4,345	35,305	83.2%	7,040	18.5%	4,260	10.8%
Sched B	32,800	3,847	36,647	88.0%	3,140	11%	30,320	4,408	34,728	83.7%	2,480	7.6%	1,919	5.2%
Sched C	32,320	4,184	36,504	87.4%	2,660	9%	30,384	4,349	34,733	83.6%	1,936	6.0%	1,772	4.9%
Sched D	30,900	4,820	35,720	86.1%	1,240	4%	30,092	4,493	34,585	83.5%	808	2.6%	1,135	3.2%
Sched E	30,980	4,796	35,776	86.2%	1,320	4%	30,096	4,499	34,595	83.5%	884	2.9%	1,181	3.3%
Project O														
Sched A	13,600	384	13,984	85.7%	2,180	19%	11,600	843	12,443	80.2%	2,000	14.7%	1,542	11.0%
Sched B	12,400	514	12,914	83.4%	980	9%	11,360	897	12,257	80.1%	1,040	8.4%	656	5.1%
Sched C	12,160	544	12,704	83.0%	740	6%	11,296	982	12,278	79.5%	864	7.1%	426	3.4%
Sched D	11,980	592	12,572	82.4%	560	5%	11,352	858	12,210	80.2%	628	5.2%	362	2.9%
Sched E	11,880	624	12,504	82.1%	460	4%	11,316	910	12,226	79.9%	564	4.7%	278	2.2%

Table 4-14 Comparing the Stochastic and Local Erlang C Schedules

The data confirms that the excess staffing is high for 4x8 staffing but decreases quickly with more flexible scheduling options. It also shows that this is a more significant problem for project J, which has a strong seasonality pattern, that for either Project S or O. The set covering approach tends to overstaff the project and achieves expected service levels higher than those achieved in the stochastic model. However, because the set covering model considers only the expected value and not the variance of arrivals, it is less effective at hedging than the stochastic model. Consider the case of schedule D for project S. The deterministic model has an expected service level of 86.2%, versus the goal of 80%, but still an expected penalty cost of $4,700. The stochastic model on the other hand has an expected service level of 82.9%, 3.3% lower, but an expected penalty only slightly higher at 5,080.

In all cases the stochastic model yields a lower direct labor cost and a lower expected cost of operation. The benefit of using the stochastic model is most significant when arrivals have a strong seasonal pattern, as in Project J, or when workforce flexibility is low. With 4x8 only staffing the stochastic model provides at least 10.8% reduction in operating costs.

4.7.4 Globally Constrained Erlang C Model

In the previous section I showed that the stochastic model based on the Erlang A model provides lower cost solutions than the locally constrained Erlang C model discussed in the literature. An alternative approach is to use a deterministic Erlang C model, ignoring abandonment and uncertainty as in the previous model, but optimizing to a global vs. local constrained. While this approach is not presented in the literature as far as I know, it is a natural simplification of the stochastic model I have analyzed so far. Because the model is deterministic, it assumes arrival rates are known, it will in general be easier to solve then the stochastic model. Ignoring abandonment will tend to increase recommended staffing, but ignoring uncertainty will tend to decrease staffing. It may be the case that under some circumstances these errors will cancel each other out and we can achieve good solutions at a lower computational cost.

The method for formulating and solving these problems is a straightforward implementation of the model (4.1) - (4.7). I solve a mean value version of the problem. The major change is that the coefficients for constraints (4.3) and (4.5) are calculated based on the Erlang C model. I still require a minimum of two agents staffed at all times, and a minimum service level at expected volume in every period of at least 50%.

I solve a version of this problem for each of the 3 projects for each scheduling option. Since the model is deterministic there is no need to solve multiple batches. To evaluate the expected cost of implementing the solution I continue to evaluate the resulting schedule against the stochastic Erlang A model. *I assume that the Erlang A model with uncertain arrivals is the correct model ands the objective of this analysis is to determine the error introduced by using a Globally Constrained Erlang C model.*

The results of this analysis are shown in the following table:

	Globally Constrained Erlang C				SCCS - Erlang A							
	Direct Labor	Expected Penalty	Expected Outcome	Average TSF	Direct Labor	Expected Penalty	Expected Outcome	Average TSF	Direct Labor Savings		Expected Savings	
Project J												
Sched A	14,000	20	14,020	88.6%	11,280	380	11,660	81.1%	2,720	19.4%	2,360	16.8%
Sched B	12,000	2	12,002	87.1%	10,800	439	11,239	80.4%	1,200	10.0%	763	6.4%
Sched C	11,760	5	11,765	86.3%	10,944	291	11,235	81.3%	816	6.9%	530	4.5%
Sched D	11,600	7	11,607	86.3%	10,844	259	11,103	81.5%	756	6.5%	504	4.3%
Sched E	11,580	26	11,606	85.8%	10,720	299	11,019	81.3%	860	7.4%	587	5.1%
Project S												
Sched A	35,200	953	36,153	87.3%	30,960	4,345	35,305	83.2%	4,240	12.0%	848	2.3%
Sched B	30,400	5,412	35,812	84.8%	30,320	4,408	34,728	83.7%	80	0.3%	1,084	3.0%
Sched C	30,160	5,426	35,586	84.7%	30,384	4,349	34,733	83.6%	-224	-0.7%	854	2.4%
Sched D	29,340	6,080	35,420	83.6%	30,092	4,493	34,585	83.5%	-752	-2.6%	835	2.4%
Sched E	29,320	6,050	35,370	83.7%	30,096	4,499	34,595	83.5%	-776	-2.6%	775	2.2%
Project O												
Sched A	11,600	976	12,576	79.9%	11,600	843	12,443	80.2%	0	0.0%	133	1.1%
Sched B	11,200	1,305	12,505	78.5%	11,360	897	12,257	80.1%	-160	-1.4%	247	2.0%
Sched C	11,120	1,394	12,514	78.3%	11,296	982	12,278	79.5%	-176	-1.6%	236	1.9%
Sched D	10,960	1,442	12,402	78.0%	11,352	858	12,210	80.2%	-392	-3.6%	192	1.5%
Sched E	11,080	1,421	12,501	78.1%	11,316	910	12,226	79.9%	-236	-2.1%	276	2.2%

Table 4-15 Comparing the Stochastic and Global Erlang C Schedules

4.8 Fine Tuning via Simulation

4.8.1 Overview

The models analyzed throughout this chapter utilize an analytical approximation of the Erlang-A model to estimate queuing system behavior in general to estimate service level in particular. The model uses a piece wise stationary approximation to estimate nonstationary behavior; the Stationary Independent Period by Period (SIPP) approximation. In section 4.3 we examined the accuracy of the SIPP approach and found that in general it provided reasonably accurate estimate of total service level, but we also found that the accuracy of the approximation varied from project to project.

In this section I develop an extension of the scheduling process used throughout this chapter that I term simulation based fine tuning. The basic idea is to take the schedules generated by solving the stochastic scheduling model and see if a better schedule can be found via a simulation based local search heuristic.

4.8.2 Simulation Based Optimization Approach

In this approach I use Discreet Event Simulation (DES) to model the operation of the call center. Unlike the analytically based model used so far, the DES approach simulates individual call processing. The simulation model is designed to generate call arrivals using the same statistical model described in Figure 2-10. The simulation model also varies the number of agents based on the time phased staffing model generated from the scheduling model.

Basic DES can evaluate the expected outcome of the candidate schedule, but in order to find a better schedule we need to implement some form of optimization algorithm. I implement a local search algorithm that starts with the schedule generated from the stochastic optimization process and searches the neighborhood of closely related schedules. The local search algorithm is guided by a variable neighborhood search (VNS) metaheuristic. VNS is a metaheuristic that makes systematic changes in the neighborhood being searched as the search progresses (Hansen and Mladenovic 2001; Hansen and Mladenovic 2005). When using VNS a common approach is to define a set of nested neighborhoods, such that

$$N_1(x) \subset N_2(x) \subset ... \subset N_{k_{Max}}(x) \quad \forall x \in X \tag{4.30}$$

The general structure of the VNS is then as follow:

```
1. Initialization
    a. Select the set of neighborhood structures $N_k$, for
       $k = 1,...,k_{\max}$
    b. Construct an initial incumbent solution, $x_I$, using
       some heuristic procedure.
    c. Select a confidence level $\alpha$ for the selection of
       a new incumbent solution
2. Search: repeat the following until Stop=True
    a. Set $k = 1$
    b. Find $n_{k_{\min}}$ candidate solutions, $x_C$ that are
       neighbors of $x_I$
    c. Simulate the system with each candidate and
       compare the results to the incumbent using a
       pairwise T Test.
    d. If any $x_C$ is superior to $x_I$ at the $\alpha$ level then
       set $x_I = x_C^*$, where $x_C^*$ is the best candidate
       solution
       Else, set $i = n_{k_{\min}}$, set found = false, and repeat
       until ($i = n_{k_{\max}}$ or found=True)
         i. Find a new candidate $x_{k_i}$
        ii. Simulate the system with each candidate and
            compare the results using a pairwise T
            Test.
        iii. If $x_{k_i}$ is superior to $x_I$ at the $\alpha$ level then
             set $x_I = x_{k_i}$ and found = True
    e. If a no new incumbent was found in neighborhood
       $k$ then
         i. set $k = k+1$
        ii. if $k > k_{\max}$ then Stop = True
```

Figure 4-27 General VNS Search Algorithm

A common approach in VNS is to define a series of nested neighborhood structures such that

$$N_1(x) \subset N_2(x) \subset \ldots \subset N_{k_{Max}}(x) \quad \forall x \in X \tag{4.31}$$

When defining the neighborhood structure I make the distinction between the set of active schedules, those schedules with a non-zero assignment in the candidate schedule, and feasible schedules which include all schedules in the feasible schedule set. Based on this distinction I define the following neighborhoods

- $N_1(x)$: **Active 1 Change**: the set of all staff plans where an active schedule is either incremented or decremented by 1.
- $N_2(x)$: **Active 2 Change**: pick any two active schedules and independently increment or decrement each.
- $N_3(x)$: **Feasible 1 Change**: pick any feasible schedules and add an assignment.
- $N_4(x)$: **Feasible 2 Change**: pick any feasible schedule and add an assignment, pick an active schedule and decrement the number assigned. .

4.8.3 Fine Tuning Process

In this test I evaluate the base schedule developed for each of the three test projects for each of the five standard scheduling options; resulting in a total of 15 different optimization problem. In each case I begin with the results found in section 4.6.

The results for project J are shown as follows:

	Preliminary Solution				Simulation Results				Comparison		
Sched Set	Labor Cost	Expected Outcome	Average TSF	TSF SD	Labor Cost	Expected Outcome	Average TSF	TSF SD	Labor Savings	Total Savings	% Total Saving
A	11,200	12,362	78.3%	4.1%	11,600	12,105	81.0%	3.8%	-400	256	2.1%
B	10,800	11,933	78.1%	3.2%	11,200	11,611	80.6%	3.0%	-400	322	2.7%
C	10,960	11,867	78.9%	3.3%	11,360	11,697	81.5%	3.4%	-400	170	1.4%
D	10,840	11,609	79.4%	3.5%	11,060	11,380	81.2%	3.0%	-220	228	2.0%
E	10,720	11,521	78.9%	3.3%	10,920	11,402	80.3%	3.1%	-200	119	1.0%

Table 4-16 Simulation Based Fine Tuning – Project J

The results show that moderate improvement is possible in all cases. The first section lists the results found by simulating the results of the schedule generated from the stochastic optimization model. Comparing the results in this table with those in table 4-11 show that the simulation model calculates a lower service level and higher penalty then estimated in the optimization model. This is consistent with optimistic bias found in section 4-3 and summarized in table 4-1. The simulation search is then able to find lower cost schedules by adding resources. The incremental cost of staffing is offset by the lowered expected cost of the service level penalty. In this particular project the benefits are moderate in the range of 1.0% to 2.7%.

The results for projects S and O are similar and are summarized in the following tables:

	Preliminary Solution				Simulation Results				Comparison		
Sched Set	Labor Cost	Expected Outcome	Average TSF	TSF SD	Labor Cost	Expected Outcome	Average TSF	TSF SD	Labor Savings	Total Savings	% Total Saving
A	30,800	32,143	83.5%	4.5%	30,000	31,313	83.3%	4.9%	800	830	2.6%
B	30,400	31,167	84.7%	4.2%	29,200	29,866	84.0%	4.6%	1,200	1,301	4.2%
C	30,880	31,418	85.0%	4.0%	28,960	29,860	83.7%	4.7%	1,920	1,558	5.0%
D	29,860	30,759	84.4%	4.2%	29,060	29,960	84.1%	4.4%	800	799	2.6%
E	30,320	30,879	85.3%	4.1%	29,080	30,102	84.0%	4.6%	1,240	777	2.5%

Table 4-17 Simulation Based Fine Tuning – Project S

	Preliminary Solution				Simulation Results				Comparison		
Sched Set	Labor Cost	Expected Outcome	Average TSF	TSF SD	Labor Cost	Expected Outcome	Average TSF	TSF SD	Labor Savings	Total Savings	% Total Saving
A	11,600	12,244	79.9%	3.7%	11,600	12,097	80.3%	3.6%	0	148	1.2%
B	11,200	12,281	78.5%	3.6%	11,600	12,047	80.6%	3.5%	-400	234	1.9%
C	11,120	12,236	78.3%	3.5%	11,440	11,978	80.2%	3.6%	-320	258	2.1%
D	11,380	12,035	79.7%	3.4%	11,480	11,972	80.2%	3.3%	-100	63	0.5%
E	11,340	12,151	79.1%	3.3%	11,540	12,015	80.2%	3.1%	-200	136	1.1%

Table 4-18 Simulation Based Fine Tuning – Project O

The improvement for Project O is more significant than what was found for Project J and generally results from reducing labor. This is consistent with the SIPP findings for project S where the scheduling model tends to underestimate the achieved service level. The results fro Project O are similar to those for project J with savings in the range of in the range of 0.5% to 2.1% resulting from increased staffing and improved service levels.

4.8.4 Summary and Other Applications

This experiment shows two things. First the schedules generated from the stochastic optimization process are quite good, but not optimal. The addition of a simulation based fine tuning algorithm can improve the schedule slightly. In this particular case the simulation model applied the same set of assumptions as the optimization process; e.g. exponential talk time, exponential patience. Those assumptions were necessary to obtain the analytical expressions of the Erlang A model and some of them are troublesome, in particular the exponential talk time assumption[57].

These assumptions are however easily relaxed in the simulation based approach. An alternative scheduling process might involve solving the stochastic optimization model under an Erlang A assumption, then fine tuning under a different set of assumptions, e.g. lognormal talk time.

An open issue in either case is the precision applied in each step of the optimization process. One could envision a process whereby the stochastic program is solved in less time with less precision, perhaps by using fewer scenarios or a larger optimality gap on the final IP, and then a final schedule is generated by simulation based optimization. The performance tradeoffs between the computational effort expended in each stage of the process is a potential topic of future research.

4.9 Summary and Conclusions

4.9.1 Summary

In this chapter I examined the issue of short term shift scheduling for call centers for which it is important to meet a service level commitment over an extended period. While the analysis focused exclusively on a TSF based SLA, the model could easily be adapted to support other forms of an SLA; such as abandonment rate or average speed to answer. The model was designed to recognize the uncertainty in arrival rates and was formulated as a mixed integer two stage stochastic program. Although difficult to solve, I showed the model is tractable and can be solved in a reasonable amount of time. In previous chapters I showed that uncertainty is of real concern in call centers, and in this chapter I showed it has a real impact on scheduling decisions.

[57] Other analytical models are available for general talk time distributions without abandonment, but relaxing the exponential assumptions and allowing abandonment creates an analytically intractable situation. See the empirical work by Brown *et. al.* that addresses the exponential talk time assumption.

4.9.1.1 Expected Cost of Implementation

In Section 4-5 I showed the Value of the Stochastic Solution for this model is substantial; ranging from 12.3% to over 21%. The clear implication is that for this model formulation ignoring variability is a costly decision; however most models in practice ignore both uncertainty and abandonment. The implication is that one should not introduce abandonment into the model without also considering uncertainty. In section 4.7 I compared this model with the common practice of scheduling to a local Erlang C constraint; that is scheduling based on a model that ignores abandonment and uncertainty but requires the service level target is achieved in every period. Comparing my model to this common practice I again found my model achieve lower cost results; ranging from 2.4% to 27%. The basic implication here is that the Erlang C model sometime achieves good results, since the abandonment and uncertainty assumptions create counter balancing errors. However the stochastic model always achieves a better solution and in many practical cases a substantially better result. This is particularly true when the flexibility of the workforce is limited to full or near full time shifts and the set covering approach introduces considerable slack in the schedule.

Finally I compared this model to a Globally Constrained Erlang C model. Though not addressed in the literature, to my knowledge, the global Erlang C model is a simple extension of the Erlang C model that relaxes the period by period service level constraint. It's rather obvious that one should expect a better result from a global constraint. This model gives superior results as compared to the local constrained Erlang C, but again my stochastic model outperforms this model in every case, by as little as 1% but by as much as 16%.

The overall conclusion is that compared to the alternative methods analyzed here, the Stochastic Model will always give a lower cost of operation schedule, and sometime this difference can be substantial. This is a basic property of stochastic programming in general, but in this analysis I have shown that the difference is significant in real world cases.

4.9.1.2 A Probabilistic Framework

In addition to provide a lower cost solution, the model presented in this chapter addresses the scheduling problem from a fundamentally different perspective. In the standard set covering approach the service level constraint is a hard constraint, it must be satisfied and any candidate schedule either achieve the service level requirement or does not. But in reality the service level is a random variable and we will achieve the SLA target with some probability. The analysis in Section 4-4 examines this explicitly and addresses the trade offs that managers must make in terms of cost and the confidence of achieving the service level. My analysis shows how the cost of operation increases non-linearly with the desired confidence level. This trade-off is obscured in the deterministic setting.

4.9.1.3 The Value of Flexibility

In Section I examined how the cost of service delivery varies with the flexibility of staffing; that is the availability of workers to staff part-time schedules. Obviously introducing more staffing options (feasible schedules) will reduce costs and my analysis quantifies that reduction. I show directly that part time staffing can substantially lower costs when managers are faced with the types of seasonality patterns evaluated in this analysis. I also show that it is only necessary to have a few workers wiling to work part time in order to get most of the benefits. The flexibility of the workforce is also a key factor in subsequent analysis. Set covering models in particular are inefficient if the workforce is constrained to full time shifts.

4.9.2 Contributions

I believe this analysis makes several important contributions to the literature. This model is in several respects quite unique, and represents a fundamental departure from the scheduling models in the literature. As discussed previously, the scheduling problem has two basic components:

- Server Sizing: determining the number of agents required to achieve a targeted performance level. This is generally accomplished via queuing model analysis assuming some type of stationary behavior.
- Staff Scheduling: determining what schedules to assign agents to in order to satisfy the requirements established by the server sizing process. This is a combinatorial optimization problem and is typically solved via integer programming.

All of the literature I examined treats these two problems as separate and distinct. The literature on staff scheduling is rich but dated; the problem being essentially solved from a theoretical perspective. The literature on server sizing is also rich, but more recent. Many recent papers address aspects of this problem such as uncertainty, time varying rates, etc. But all of these papers find a staffing vector independent of any staffing constraints. The implicit assumption is that the output of the server sizing problem can be implemented with little loss via some set covering approach.

Perhaps the most significant contribution this analysis makes is combining these two steps into one. Throughout this analysis I have shown that no mater how accurate the server sizing calculation, staff scheduling constraints can introduce substantial slack into the schedule unless the workforce is very flexible. It is a well established principle in operations research that one will obtain sub optimal results by optimizing the components of a system instead of optimizing the system globally. This analysis verifies and quantifies that concept for this particular problem.

Once the problem is model as global optimization problem it is fairly straightforward to introduce uncertainty and formulate it as a stochastic optimization problem. This paper is to my knowledge the first application of stochastic programming to the staff scheduling problem.

4.9.3 Management Implications

4.9.3.1 Consider Variability

Managers should consider variability of arrivals when creating staffing plans and/or estimating the cost of service delivery. Periods with higher variability require extra staffing capacity to properly hedge risk. Similarly, if projects have the same average volume, but one is more variable than the other, the more variable project will be more expensive to service.

4.9.3.2 Add Flexibility

Flexibility in staffing can substantially lower the cost of service delivery. The ability to schedule resources to part time staffing allows the staffing profile to better match the demand profile. However, as the analysis in section 4-6 shows, a little bit of flexibility goes a long way. If managers can find just a few people to work part-time they can achieve mnost of the benefit that comes from part time staffing.

4.9.4 Future Research

4.9.4.1 Break Scheduling

In this analysis I focused only on scheduling what Pinedo calls *solid tours* (Pinedo 2005). A relatively straightforward extension allows breaks to be scheduled at the same time as the solid tours. Assume that for each shift we can define a *break window*, the period during which breaks may be scheduled. For an eight hour (work) schedule, this might be periods 7-12. (An eight hour shift with a one hour break will cover nine elapsed hours or 18 periods.) Further, assume that the actual break time might occur during any two consecutive periods in the window, and that all choices are equally likely. The following figure illustrates:

			Schedule			
Period	1	2	3	4	5	Avg
1	1	1	1	1	1	1.0
2	1	1	1	1	1	1.0
3	1	1	1	1	1	1.0
4	1	1	1	1	1	1.0
5	1	1	1	1	1	1.0
6	1	1	1	1	1	1.0
7	-	1	1	1	1	0.8
8	-	-	1	1	1	0.6
9	1	-	-	1	1	0.6
10	1	1	-	-	1	0.6
11	1	1	1	-	-	0.6
12	1	1	1	1	-	0.8
13	1	1	1	1	1	1.0
14	1	1	1	1	1	1.0
15	1	1	1	1	1	1.0
16	1	1	1	1	1	1.0
17	1	1	1	1	1	1.0
18	1	1	1	1	1	1.0

Figure 4-28 Implicit Break Calculations

We can then generate the shift mapping coefficients a_{ij} in (4.2) as non-integral "expected staffing levels." The TSF calculation in (4.2) and illustrated in Figure 4-6 does not require integral staffing levels. This is a direct consequence of the integration of the server sizing and staff scheduling steps into one optimization problem.

4.9.4.2 Alternative Queuing Assumptions

The service level calculation in this analysis was based on a piecewise linear approximation of the Erlang A queuing formula. While Erlang A is a reasonably good model, it does have some limitations, most notably the assumptions of exponential talk time. These assumptions are easily relaxed. One approach is the simulation based tuning approach discussed in this chapter. An alternative is to use a different queuing model to generate the service level approximation. This is unlikely to generate fundamentally different results, but it could conceivably provide a better fit if scheduling a real project.

4.9.4.3 Simulation Based Heuristic

The stochastic model presented in this chapter generates a better scheduling that the man value model. The stochastic model does however require a non-trivial amount of computer resources. It may be possible to create a faster heuristic that will generate *good* results. An algorithm that first applies a greedy heuristic to cover local constraints than performs a simulation based fine tuning is one promising possibility.

4.9.4.4 Erlang C Assumption Sensitivity

The analysis in Section 4-7 shows that the stochastic scheduling model performs consistently better than the Erlang C model. What is a somewhat surprising is how well the Erlang C model performs given the questionable assumptions. I postulate in this dissertation that error created by the abandonment and uncertainty assumptions tend to cancel. Future research could investigate the conditions under which the Erlang C model provides reasonably good results.

4.9.4.5 Algorithm Tuning

In this dissertation I have developed a model to find an optimal staffing plan when arrival rates are uncertain. I implement a version of the L-Shaped Decomposition algorithm that that provides a reasonable tractable solution. I do not however focus on algorithmic efficiency in this dissertation. I have no doubt that additional analysis could improve the efficiency of the algorithm.

5 The Medium Term Hiring Model

5.1 Overview

The objective of the medium term model is to address capacity management in the 0-3 month start up phase of a new project launch. Since the services provided are highly technical a significant training investment is required for new hires. Significant learning occurs during the start up phase and productivity increases rapidly. The outsourcing contract typically specifies a global service level agreement, but the SLA is often not strictly enforced until the third month of the launch. The management problem I address is the development of a staffing plan for a new project in the face of uncertain demand and productivity. Over hiring results in training expenses that can not be recouped on other projects, while under hiring results in poor customer service and may make it impossible to achieve the service level commitment. I seek to develop a model that finds the optimal level of hiring; that is the level of hiring with the lowest total expected cost of operation.

The level of pre-launch hiring is a critical decision in the new project launch process. The following graph outlines the basic timeline of the process.

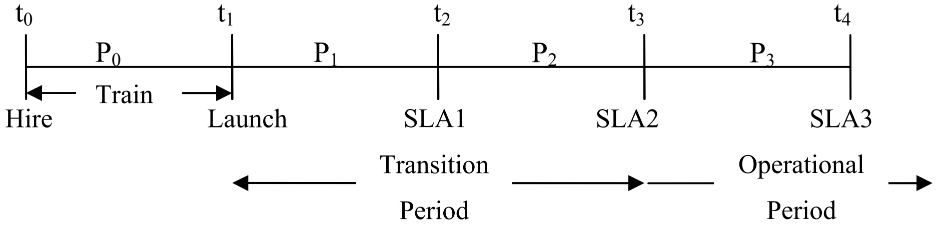


Figure 5-1 Project Launch Timeline

At time t_0 a set of resources are hired and entered into a training program. The company makes a hiring decision with an uncertain call volume and talk time. Training occurs in period zero and at time t_1 the project launches and the uncertainty in average call volume and initial talk time are revealed. In periods one through three the company provides service, measuring service levels which are reported at times t_2 through t_4. Periods one and two are considered transition periods, significant learning occurs during this period and service level agreements are often not strictly enforced. Period three is typically the first month in which service levels are contractually enforced. Turnover and learning occur throughout the launch process.

Learning occurs through individual and institutional processes. Individual learning occurs as agents become more familiar with the systems and become more productive in solving customer problems. Institutional learning occurs as the project knowledge base, the repository of known problems and solutions, is enhanced. Both types of learning result in increased agent productivity. Individual learning is lost when turnover occurs while institutional learning remains in the knowledge base. In this model I assume turnover occurs only at discrete points in time, t_1 through t_4. We assume hiring is instantaneous, but that training delays the deployment of replacement hires by one period. Average volume is revealed at time t_1, but call volumes are subject to stochastic variability in all periods.

This model is developed as a multistage decision problem. The initial decision on hiring occurs at time t_0. This decision occurs before average call volumes or learning curve effects are revealed. Recourse decisions are made at times t_1, t_2, and t_3 that include additional hiring and/or termination. The management objective is to minimize the overall expected cost of staffing such that the service level agreement is satisfied by period three.

5.2 Model Formulation

The model presented here is a refinement of the model in (Robbins and Harrison 2006), modified to address the specific issue related to a new project startup[58]. I formulate the model as a multi stage stochastic program with the following definitions:

[58] The Robbins and Harrison (2006) model examined optimal hiring in a generic professional services environment where demand and attrition where variable and hiring was limited to certain time periods. The model here tailors that model to the specifics of the project start up situation and explicitly includes learning curve and service level agreement considerations.

Sets

T: time periods
K: scenarios
H: linear segments of service level curve

Deterministic Parameters

w: wage rate
h_t: hiring and training cost
f: termination cost
g: SLA target
q_t: SLA penalty rate in period t
μ: minimum expected SLA
r: expected SLA shortfall penalty rate

Stochastic Parameters

m_{tkh}: SLA slope in period t of scenario k
b_{tkh}: SLA intercept in period t of scenario k
a_{tk}: attrition rate in period t of scenario k
γ_{tk}: institutional productivity in period t of scenario k
ρ_{jtk}: individual productivity in period t of scenario k for resources hired in period j

Variables

X_{tk}: resources in period t of scenario k
H_{tk}: hires in period t of scenario k
Y_{tk}: SLA shortfall in period t of scenario k
F_{tk}: terminations in period t of scenario k
C_{tk}: effective capacity in period t of scenario k
S_{tk}: SLA in period t of scenario k
E_t: Expected SLA penalty

Probabilities

p_k: probability of scenario k

$$Min\sum_{t\in T}\sum_{k\in K} p_k(wX_{tk} + h_t H_{gk} + fF_{tk} + q_t Y_{tk} + rE_t) \tag{5.1}$$

subject to

$$X_{tk} = X_{t-1,k} + H_{tk} - F_{tk} - a_{tk}X_{t-1,k} \qquad \forall t\in T, k\in K \tag{5.2}$$

$$C_{tk} = \gamma_{tk}\left(\sum_{j\in T, j<t} \rho_{jtk}\left(H_{jk} - F_{jk} - a_{tk}X_{t-1,k}\right)\right) \qquad \forall t\in T, k\in K \tag{5.3}$$

$$S_{tk} \le m_{tkh}C_{tkh} + b_{tkh} \qquad \forall t\in T, k\in K, h\in H \tag{5.4}$$

$$Y_{tk} \ge g - S_{tk} \qquad \forall t\in T, k\in K \tag{5.5}$$

$$E_t \ge u - \sum_{k\in K} p_k Y_{tk} \qquad \forall t\in T \tag{5.6}$$

$$X_{tk}, H_{tk}, Y_{tk}, F_{tk}, C_{tk}, S_{tk} \ge 0 \qquad \forall t\in T, k\in K \tag{5.7}$$

$$X_{0k} \in \mathbb{Z} \qquad \forall k\in K \tag{5.8}$$

The objective function (5.1) seeks to minimize the expected cost of staffing plus the penalty cost associated with failing to meet the SLA target; a penalty is applied for any scenario that does not achieve the period's SLA target. A second, large penalty is assessed of the expected service level in any period it is below some minimal threshold. This condition ensures that some minimal service level target is enforced based on expected volume. Constraint (5.2) is the staff balance constraint; it defines the staff in a period to equal the prior period staff plus new hires, less attrition and terminations. Constraint (5.3) defines the effective capacity of the current staff. For each hiring cohort the factor β_{thk} specifies the individual productivity component. Effective capacity is further adjusted by the institutional capacity factor α_{tk}. Constraint (5.4) defines the SLA achieved in period *t* based on the stochastic demand. The model is formulated so that demand is expressed in terms of the slope and intercept of the linear approximation of the TSF curve. Constraint (5.5) defines the SLA shortfall, the degree to which the realized SLA is below the target level g. Constraint (5.6) calculates the expected SLA shortfall, the degree to which the expected service level falls short of the minimum target. Constraint (5.7) defines non-negativity and conditions and constraint (5.8) forces the period 0 hiring decisions to be integral valued.

The most significant uncertainty in demand is in the first period where the overall level of demand is uncertain. After the general level of demand is revealed, period to period volume varies stochastically.

5.2.1 Detailed Decision Process Timing

The model (5.1) - (5.6) implements the decision process outlined in Figure 5-1. To further clarify how this process works, the following steps present the process in more detail.

1. At time t_0 the firm hires an initial group of agents. Those agents are trained during period P_0. During this period the agents are paid a salary and the firm makes an additional investment in training.
2. At time t_1 the project goes live and begins accepting calls. Calls are received throughout period P_1 and overall call volumes and call patterns are revealed. Throughout period P_1 agents may resign reducing the capacity of the project team.

3. At time t_2 the first period SLA is calculated and any shortfall penalty is assessed. At this time the firm may choose to hire additional agents or terminate existing agents. The firm incurs a severance cost for all terminated agents.
4. Newly hired agents are trained during period P_2 and are unavailable to take calls. A training cost is incurred for these agents and they are paid a salary. Call volume during period P_2 is handled by the original set of agents who are now more productive due to learning.
5. At time t_3 the second period SLA is calculated and any shortfall penalty is assessed. At this time the firm may again make a hire/termination decision.
6. During period P_3 agents hired at time t_3 are trained and paid a salary. Call volume is handled by the remaining agents hired at times t_0 through t_2.
7. At time t_4 the third period SLA is calculated and any shortfall penalty is assessed. At this time the firm may again make a hire/termination decision.

The detailed decision process is illustrated in the following diagram:

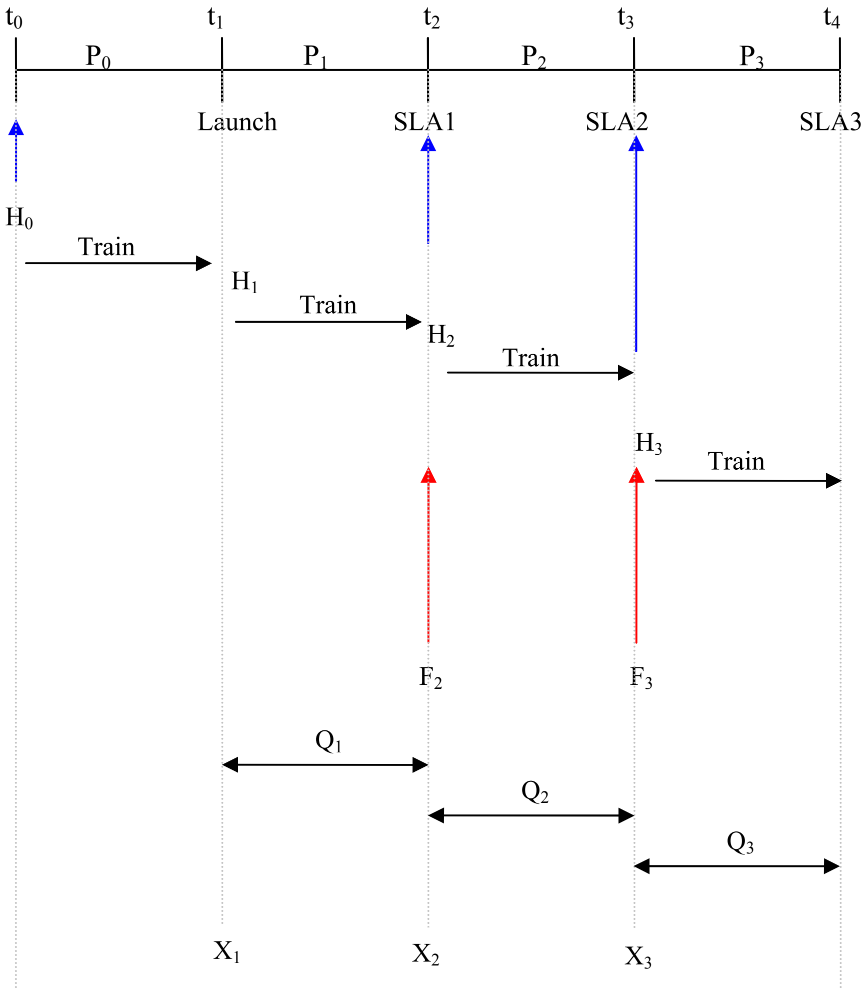


Figure 5-2 Detailed Timeline

5.2.2 Effective Capacity

An important consideration in this model is the capacity available to service calls in any time period. I define the *base capacity* as the total number of agents available in the period. The *effective capacity* is the capacity of the equivalent number of experienced agents; that is the base capacity adjusted for training and deflated by the relative productivity of the agent base.

The base capacity is impacted by hiring, firing and attrition. Net capacity is impacted by agents held for training along with learning curve issue. In any time period average agent productivity will vary based on length of service. Throughout this analysis I assume that all terminations, voluntary or involuntary, come from the initial hiring class. This is a conservative assumption; these will be the longest tenured and most productive agents. I also assume that *quits* are distributed evenly through the time period but firing occurs at the beginning of the period. Furthermore, as illustrated in figure 5-2, I assume that no firing occurs at the start of period one. In this model firing occurs only to adjust capacity and there is no reason to adjust capacity prior to launch. In reality firing may also occur because of performance issues. This is a random event and for the purposes of this model such firing can be included in the attrition parameter.

The base and effective capacity available in each time period is summarized in the following table:

Period	Base Capacity	Effective Capacity
1	$H_0 + H_1 - .5Q_1$	$\rho_{01}\left(H_0 - .5Q_1\right)$
2	$H_0 + H_1 + H_2 - .Q_1 - .5Q_2 - F_2$	$\rho_{02}\left(H_0 - .Q_1 - .5Q_2 - F_2\right) + \rho_{12}H_1$
3	$H_0 + H_1 + H_2 + H_3 - .Q_1 - Q_2 - .5Q_3 - F_1 - F_2$	$\rho_{03}\left(H_0 - .Q_1 - Q_2 - .5Q_3 - F_2 - F_3\right) + \rho_{13}H_1 + \rho_{23}H_2$

Table 5-1 Start Up Capacity by Period

5.2.3 The Multistage Decision Tree

The uncertainty associated with model parameters in the program (5.1) - (5.6) is represented by generating a set of sample paths, or scenarios, against which the optimization is conduced. However, the scenario generation problem for the multistage program is considerably more difficult than that of the two stage problem analyzed in Chapter 4. The key issues in the multistage problem are nonanticipativity and the multistage scenario explosion problem; issues I briefly review in the following sections.

In a multistage stochastic program the decisions made at each stage, $t = 1, 2, ..., T$ are based on the observed realizations of the random variables made in all proceeding stages, $\omega^{T-1} = \left\{\omega_1, \omega_2, ...\omega_{T-1}\right\}$. I represent these realizations via a scenario tree; an oriented graph that begins with a single root node at level 0, and branches into a series of nodes at level 1, each node corresponding to a possible realization of ω in period one. The tree continues to branch up to the nodes at level 3. Each node in the tree has a single predecessor and a finite number of descendants corresponding to the possible realization of the random vector at that stage. A scenario tree that is constructed such that the number of descendants is identical for each non-leaf node and the tree is said to be *balanced*. If a different number of scenarios are allowed at each stage then the tree is said to be *unbalanced*.

An example of a balanced tree is shown in the following figure

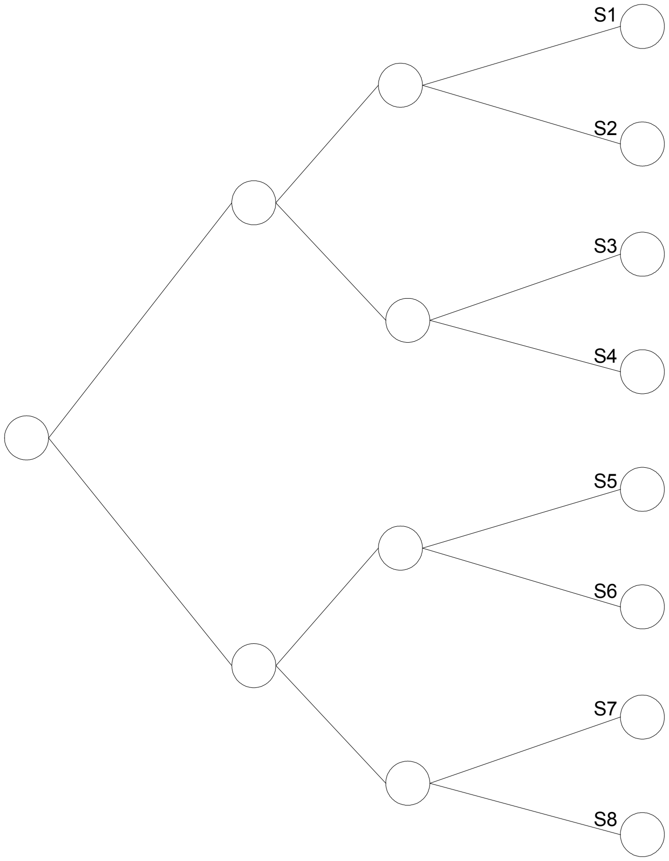


Figure 5-3 Multistage Scenario Tree

5.2.4 The Scenario Explosion Problem

This tree in Figure 5-3 is balanced with two realizations per stage. The tree has a total of eight scenarios or possible sample path outcomes. In general, for this three stage problem the number of scenarios in a with R_t realizations per stage, the number of scenarios is

$$N = R_1 R_2 R_3 \tag{5.9}$$

The total number of scenarios therefore grows in a rapid, but polynomial, fashion with the number of realizations per stage. The following graph illustrates the number of scenarios associated with various realization levels for a balanced tree.

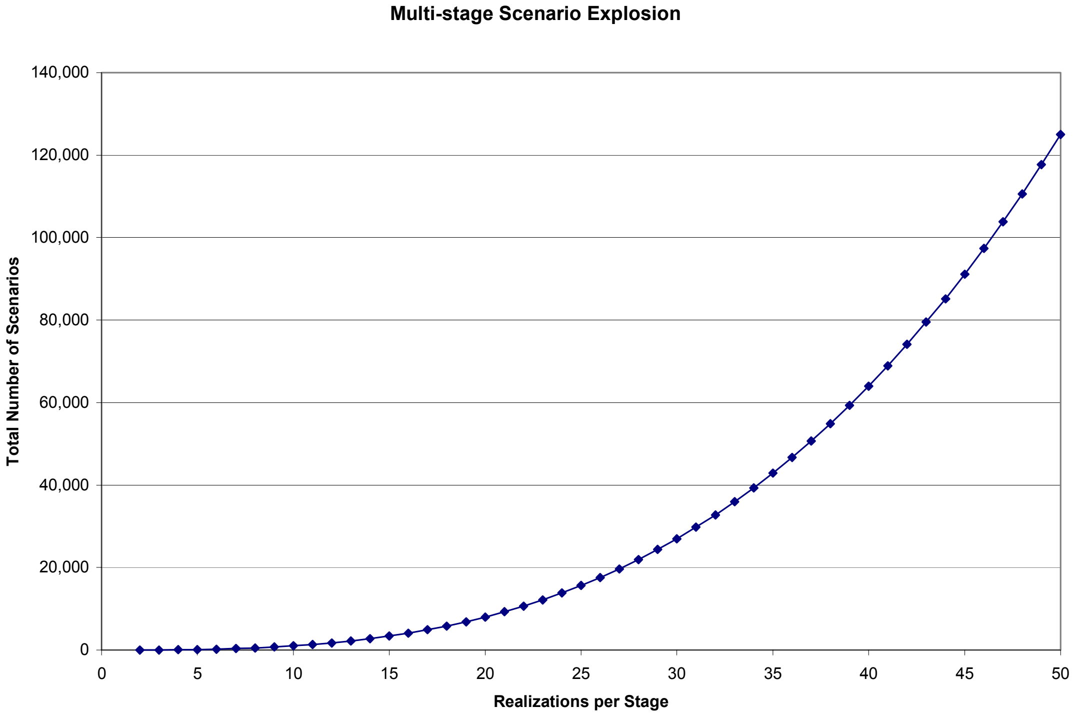


Figure 5-4 Scenario Explosion

Since the size of the multistage linear program increases nonlinearly with the number of scenarios, computational effort will clearly increase significantly as the number of realization per stage increases.

5.2.5 Information Bundles and Nonanticipativity Constraints

An important practical consideration in multistage programs is the enforcement of nonanticipativity. Simply stated nonanticipativity requires that any decisions made at any stage in the decision process are based only on the information available to the decision maker at that time; decision making can not *anticipate* future outcomes. To facilitate this restriction we need to introduce the concept of a *bundle*.

Consider the unbalanced tree depicted in the following figure.

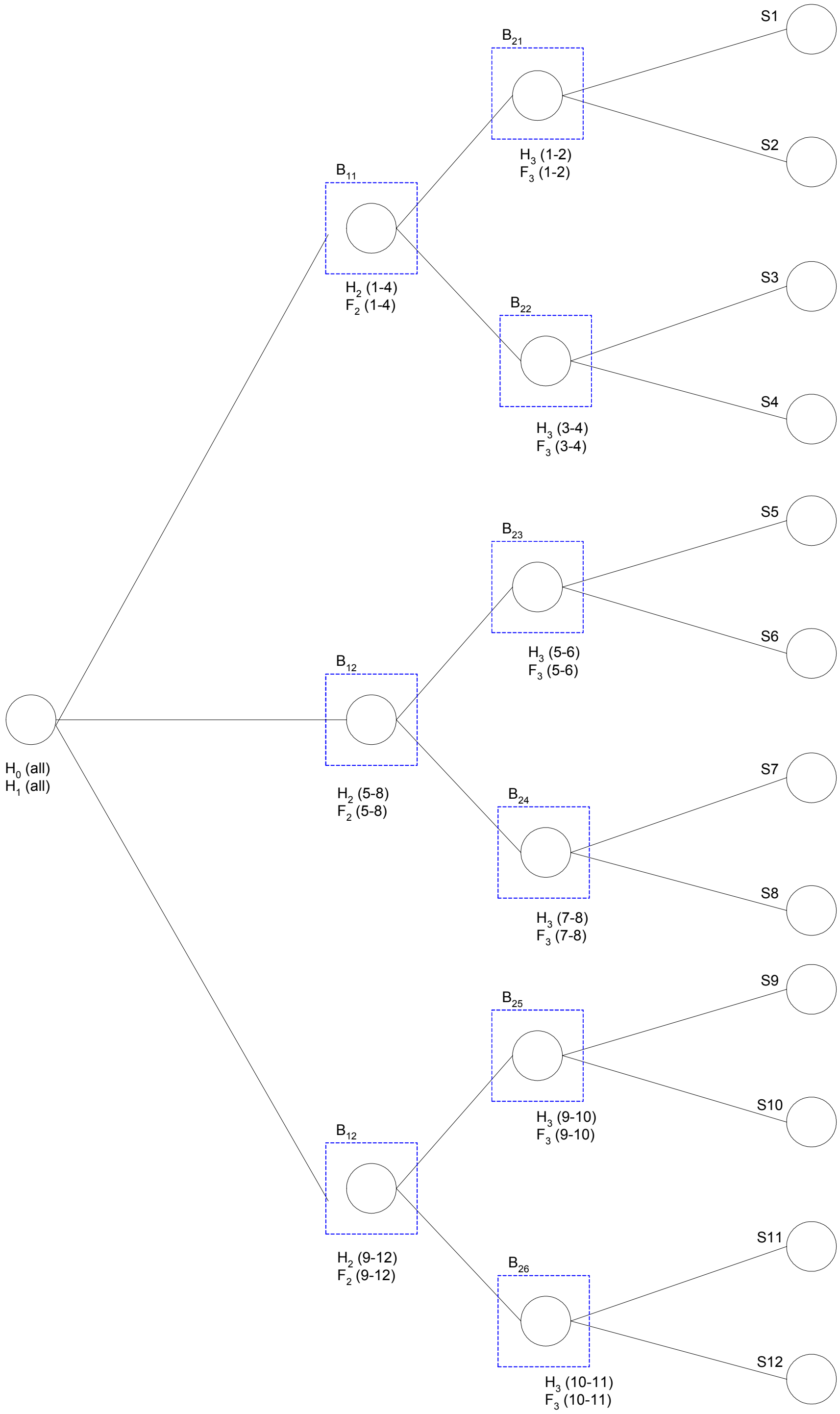


Figure 5-5 Unbalanced Scenario Tree

Each node has a number of scenarios that pass through that node. A bundle is the set of all scenarios that pass through a specified node. Bundles are indicated by squares and are denoted B_{ij}, where *i* represents the decision stage, and *j* sequentially indexes the bundles in each stage.

Consider the bundle labeled B_{11}, this bundle is the set of scenarios $\{S_1, S_2, S_3, S_4\}$. In stage 1 of the decision problem, the decision maker can not anticipate which of these four outcomes will ultimately be realized and hence his decision must be identical for each scenario. The notation below the square (*H₂(1-4)*, *F₂(1-4)*) indicates that the stage 2 hiring and firing decisions for scenarios 1-4 must be identical. Likewise the stage three decisions made at the bundle $B_{21} = \{S_1, S_2\}$ must be identical. These conditions are represented via nonanticipativity constraints. In our model we must implement nonanticipativity constraints on both the hiring and firing variables, both at stages one and two. Let N_2 denote the set of nonanticipativity constraints applied to stage two. The constraints on hiring can then be represented as

$$\sum_{k \in K} \eta 2_{nk} H_{2k} = 0 \quad \forall \eta 2 \in \mathrm{N}_2 \tag{5.10}$$

$\eta 2_{nk}$ is a coefficient with value in $\{-1, 1\}$. By selecting the appropriate pairs the constraints enforce nonanticipativity. For example, consider the bundle B_{11} in Figure 5-5 and its associated hiring constraint. This constraint can be represented as $h_{21} - h_{22} = 0$. The scenario tree in Figure 5-5 requires two nonanticipativity constraints on hiring in stage one, and six constraints in stage two. An identical number of constraints are required on firing.

5.2.6 Problem Formulation with Explicit Nonanticipativity

Given the scenario approach I will use for this problem I can now restate the complete problem with explicit nonanticipativity constraints.

$$Min \sum_{t\in T}\sum_{k\in K} p_k (wX_{tk} + h_t H_{gk} + fF_{tk} + q_t Y_{tk} + rE_t) \qquad (5.11)$$

subject to

$$X_{tk} = X_{t-1,k} + H_{tk} - F_{tk} - a_{tk} X_{t-1,k} \qquad \forall t \in T, k \in K \qquad (5.12)$$

$$C_{tk} = \alpha_{tk} \left(\sum_{j\in T, j<t} \beta_{jtk} \left(H_{jk} - F_{jk} - a_{tk} X_{t-1,k} \right) \right) \qquad \forall t \in T, k \in K \qquad (5.13)$$

$$S_{tk} \le m_{tkh} C_{tkh} + b_{tkh} \qquad \forall t \in T, k \in K, h \in H \qquad (5.14)$$

$$Y_{tk} \ge g - S_{tk} \qquad \forall t \in T, k \in K \qquad (5.15)$$

$$E_t \ge u - \sum_{k\in K} p_k Y_{tk} \qquad \forall t \in T \qquad (5.16)$$

$$\sum_{k\in K} \eta 2_{nk} H_{2k} = 0 \qquad \forall \eta 2 \in \mathrm{N}_2 \qquad (5.17)$$

$$\sum_{k\in K} \eta 3_{nk} H_{3k} = 0 \qquad \forall \eta 3 \in \mathrm{N}_3 \qquad (5.18)$$

$$\sum_{k\in K} \eta 2_{nk} F_{2k} = 0 \qquad \forall \eta 2 \in \mathrm{N}_2 \qquad (5.19)$$

$$\sum_{k\in K} \eta 3_{nk} F_{3k} = 0 \qquad \forall \eta 3 \in \mathrm{N}_3 \qquad (5.20)$$

$$X_{tk}, H_{tk}, Y_{tk}, F_{tk}, C_{tk}, S_{tk} \ge 0 \qquad \forall tk \qquad (5.21)$$

In this formulation constraints (5.17) - (5.20) enforce nonanticipativity. Constraints (5.17) and (5.18) enforce nonanticipativity on hiring while constraints (5.19) and (5.20) enforce nonanticipativity on firing.

5.3 Characterizing Uncertainty

In Chapter 4 we reviewed a scheduling model that considered variability in arrival rates. In this model the variability is stochastic in the sense that outcomes are random realizations from known probability distributions. In the start up phase we face the additional challenge of parameter uncertainty. In most cases the decision maker does not have hard data on key system parameters but must instead make subjective prior estimates.

For the sake of this analysis I will assume uncertainty in the following parameters:

- **Volume**: overall average weekly call volume.
- **Arrival Rate Variability**: the level of variability in day of week and time of time call variability.
- **Talk Time**: average service time for experienced agents.
- **Learning Curve**: the learning curve coefficients both at the individual and institutional level.

In each case I assume that the true parameter value is unknown and is drawn from some prior probability distribution. In addition to parameter uncertainty the system under analysis here faces stochastic variability in key parameters; specifically:

- **Realized Volume**: actual call volume presented by day of week and time of day.
- **Attrition**: the number of employees who resign in any time period.

For the sake of this analysis, all other parameters (e.g. hiring cost, firing cost and SLA penalties) are considered know[59]. These parameters may however vary over time; for example hiring may be less expensive in the initial period when training costs can be amortized over a large number of hires. In this model I assume that parameter uncertainty is effectively eliminated during the first operational period. During P1 managers have the opportunity to observe four weeks of data and make informed estimates of model parameters. In subsequent periods model parameters exhibit only stochastic variability.

The process of generating scenarios then proceeds as follows. For a given number of realizations at each stage, I calculate the total number of scenarios, information bundles, and nonanticipativity constraints at stages 2 and 3. For each realization at stage 1 a set of parameter values are sampled that hold for that branch of the tree. In each stage stochastic variability is added to the service level curves by adding an error term to the intercept of the service level curves. Attrition rates are

[59] I also fix the institutional productivity factor to 1 for this analysis and consider only the impact of individual productivity. While the distinction between individual and institutional productivity is theoretically appealing, I lack the data to make independent estimates. It's also apparent from the analysis that follows that the level of post launch hiring is small enough so that the distinction has no practical impact on the results.

estimated by calculating a random binomial variable based on the period average attrition and an average estimate of staffing. Individual productivity is calculated by calculating the average productivity rate from the learning curve and adding a stochastic error term.

5.3.1 Service Level Approximations

5.3.1.1 Overview

A key consideration in the practical application of this program will be the development of a set of service level approximation curves; the curves whose coefficients create the piecewise linear approximation of the service level achieved for various staffing level decisions. These curves are represented in the problem formulation as the coefficients m_{tkh} and b_{tkh} in equation (5.14).

5.3.1.2 TSF Curve Generation Process

To develop the service level approximation curves I utilize the procedure described below. Let N be the average weekly volume, T the average talk time, v_d be the daily variability scale factor, and v_t be the time period variability scale factor.

1. Identify a template project profile that has the approximate seasonality pattern of the new project.
2. Define prior probability distributions for stochastic parameters, N , T , v_d , v_t
3. Generate a uniform design for four factors and 10 design points.
4. Define S different staffing levels.
5. For each design point, set the total volume and scale the variability of arrivals appropriately. Generate five batches of 25 scenarios each.
6. For each staff level find the associated service level by solving problem.
7. Calculate a slope and intercept for each adjacent pair of staffing levels using the average of the five batches.

Figure 5-6 Service Level Approximation Process

5.3.1.3 Service Level Maximization Program

In step six of the process outlined in Figure 5-6 we must find the service level for a sample call pattern for a fixed staffing level. The scheduling algorithm in Chapter 4, (4.1) - (4.7) solves a related problem - finding the minimum cost schedule to achieve a desired service level. We can modify this program to find the maximum expected service level possible for a given staffing level. I call this program the service level maximization problem. Using the notation from Chapter 4 the model can be expressed as

$$\max \sum_{k\in K} p_k \frac{\sum_{i\in I} y_{ik}}{\sum_{i\in I} n_{ik}} \tag{5.22}$$

subject to

$$y_{ik} \le m_{ikh} \sum_{j\in J} a_{ij} x_j + b_{ikh} \qquad \forall i\in I, k\in K, h\in H \tag{5.23}$$

$$y_{ik} \le n_{ik} \qquad \forall i\in I, k\in K \tag{5.24}$$

$$\sum_{j\in J} a_{ij} x_j \ge \mu_i \qquad \forall i\in I \tag{5.25}$$

$$x_j \le m_j \qquad \forall j\in J \tag{5.26}$$

$$\sum_{j\in J} x_j \le N \tag{5.27}$$

$$x_j \in \mathbb{Z}^+, y_{ik} \in \mathbb{R}^+ \qquad \forall i\in I, j\in J, k\in K \tag{5.28}$$

The objective function (5.22) seeks to maximize the expected service level; the ratio of calls answered within service level to the total number of calls. As in the Chapter 4 program, constraints (5.23) and (5.24) create a piecewise linear approximation of the service level curve. Equation (5.25) creates a lower bound on the total number of agents scheduled in each period and this coefficient is set to achieve at least a 50% expected service level and guarantee that at least two agents are always staffed. Constraint (5.26) sets an upper limit on the total number of agents that can be scheduled to each shift, and constraint (5.28) enforces non-negativity and integrality conditions.

5.3.2 The Base Case Example

To illustrate the analysis process I'll work through an example. Assume that we are planning a launch of a new corporate support project that operates 24x7. The project is subject to an 80/60 SLA. We first pick a similar project profile, which in this case is Project J. In many cases detailed call volume data is not available, for example if multiple help desks are being consolidated. In this case assume that the best estimate is that call volume will average 5,000 calls per week, and that we are reasonably sure volume will be at least 4,000 and no more than 7,000 calls per week. With limited data a common prior distribution is the triangular distribution (Law 2007). So for planning purposes we assume that the true expected average weekly volume has a triangular distribution with parameters (4000, 5000, 7000). Talk time for corporate projects tends to be in the range of 9 to 14 minutes. Without empirical data I will assume that the true average talk time is drawn from a uniform distribution on this range. Finally we must develop a prior estimate for the variability of arrivals, relative to project J. I will assume that the scaling factor is uniformly distributed on [.75, 1.25].

Using these distributions I use a 10 point uniform design in four factors to generate 10 design points. The Uniform Design is summarized in the following table:

				Time
			Daily	Period
DP	Volume	Talk Time	Variability	Variability
1	4,387	10.75	0.88	0.98
2	5,183	11.75	1.08	0.78
3	5,775	9.75	0.83	0.83
4	5,025	12.25	0.78	1.18
5	5,357	10.25	1.23	1.03
6	5,551	13.75	0.93	1.08
7	6,452	12.75	0.98	0.93
8	6,051	11.25	1.13	1.23
9	4,671	13.25	1.18	0.88
10	4,866	9.25	1.03	1.13

Table 5-2 Uniform Design for Service Level Approximations

The Uniform Design approach ensures that the 10 points effectively fill the four dimensional design space.

At each design point I generate 25 scenarios. The service level maximization problem (5.22) - (5.28) is solved for 8 staffing levels (15,20,25,30,35,40,50,65).

Executing this process results in the service level curves shown in the following figure:

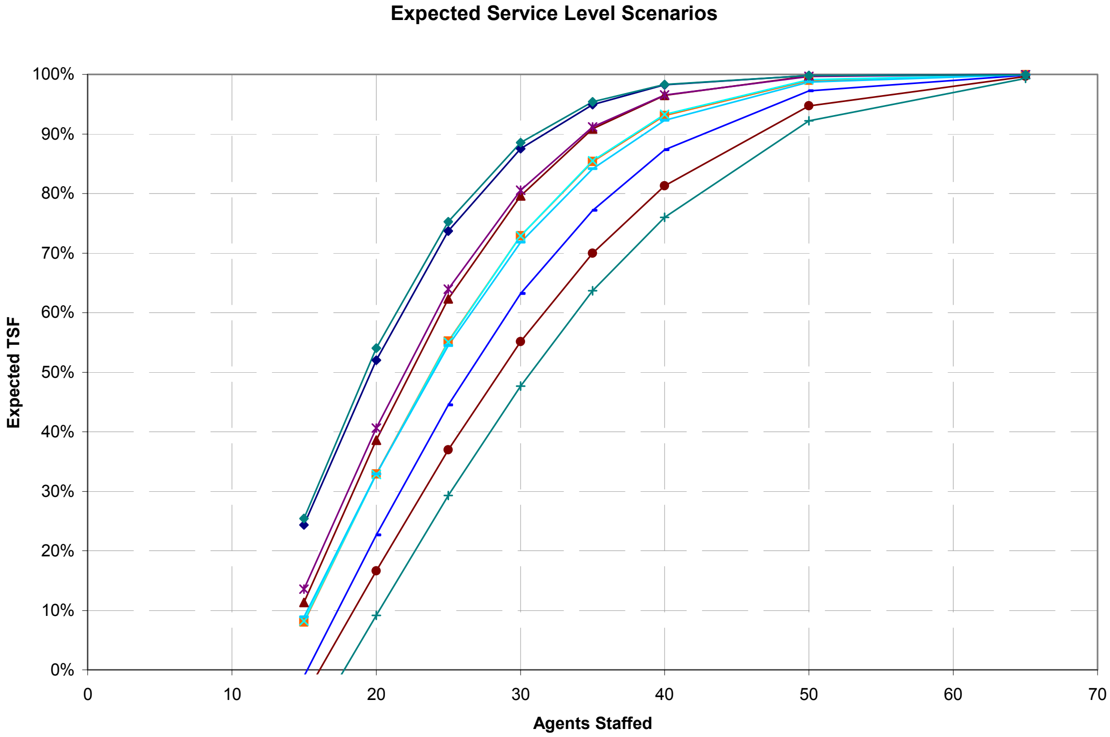


Figure 5-7 Estimated Service Level Curves

Each of the 10 lines in this figure represents, based on the assumed priors, an equally likely aggregate service level curve.

5.4 Empirical Properties of the Sampled Problem

5.4.1 Overview

Given the scenario explosion problem illustrated in Figure 5-4, a key decision will be the number of realizations to choose when solving the sample path (scenario) approximation problem. Increasing the number of scenarios is likely to yield a more accurate result but at a rapidly increasing cost. In this section I examine the properties of the sampled problem as the number of scenarios increases. Because parameter uncertainty is revealed in the first stage and subsequent stages exhibit only stochastic variability, it is reasonable to assume that stage 1 realizations have a bigger impact on the outcome than subsequent stages. For this reason I implement an unbalanced tree approach with more realization in the first stage.

In the following section I examine the sampling properties of the MIP along with the LP relaxation, i.e. a problem that allows non-integral stage 0 hiring decisions.

5.4.2 Sampling Properties of the LP Relaxation

I first examine the sampling properties of the relaxed problem. I use an unbalanced tree where the number of realizations in stage one is larger than the number of realization in stages two and three. I use the same number of realization in stages two and three, and examine two cases where each later stage has either 10 or 15 realizations.

5.4.2.1 N/10/10 Sampling

In this analysis I keep second and third stage realizations at 10 each and vary stage one realizations between 10 and 80. This results in a set of problems where the number of scenarios varies from 1,000 to 8,000. In each case I generate a batch of 15 independent scenario files. From an analysis perspective I am interested in the objective value, the stage 0 hiring decision and the solution time; each of which will be a random variable.

The results of this analysis are summarized in the following table:

	Objective		Stage 0 Hiring		Reource Use	
		Standard		Standard		Standard
Scenarios	Mean	Deviation	Mean	Deviation	Mean	Deviation
1,000	331,753	11,953	42.5	1.92	7.9	1.87
1,500	331,158	13,239	42.2	1.82	12.7	2.88
2,000	335,744	12,341	42.4	1.46	25.5	7.15
2,500	324,385	4,992	41.4	0.76	43.6	14.84
3,000	326,976	7,542	41.6	0.97	79.1	40.84
3,500	334,215	8,380	42.5	1.16	95.7	70.78
4,000	330,863	5,884	42.1	0.80	116.5	37.13
4,500	326,974	6,363	41.4	0.91	124.3	44.32
5,000	332,040	6,112	42.2	0.83	314.3	179.63
5,500	331,276	7,007	42.1	0.89	263.6	164.36
6,000	331,631	9,359	42.0	1.16	317.9	277.07
6,500	334,711	4,016	42.5	0.61	351.8	185.34
7,000	335,213	6,368	42.6	0.78	417.6	338.40
7,500	333,057	5,551	42.1	0.72	503.7	434.20
8,000	334,161	4,245	42.4	0.53	605.5	417.78

Table 5-3 Empirical Properties of N/10/10 Models

A few observations are immediately apparent. The variability of the objective and stage zero hiring decision both decrease with the number of scenarios, but do so at a relatively slow and somewhat erratic rate. The sample standard deviation sometimes increases with increased samples, and in general the difference between sampled observations of the standard deviation from adjacent rows in table 5-3 is not statistically significant at the .95 level. Similar results hold for the stage zero hiring decision. The mean values for each of these quantities also change very little with sample size and a paired T-Test fails to reject the null hypothesis that either the objective or the hiring decision means are different from row to row.

A graphical view further illustrates these results. The following graph shows the objective values for each of the 15 samples at each scenario level. The small points represent individual samples while the larger square represents the sample average.

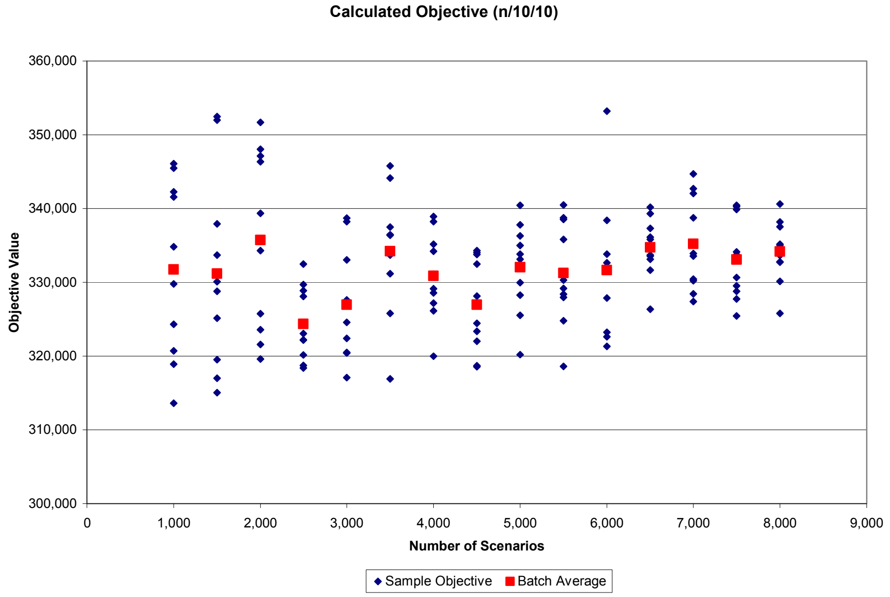


Figure 5-8 Objective Values for Sample Problems (N/10/10)

The variance of the samples is seen to decline slightly as the number of scenarios increases but the sample average seems to show only minimal variation. Note that unlike the two stage problem, the sampled problem does not necessarily exhibit a bias level decreasing in the number of scenarios. Similarly, if we examine the distribution of the hiring decision shown in the following graph we see a similar pattern.

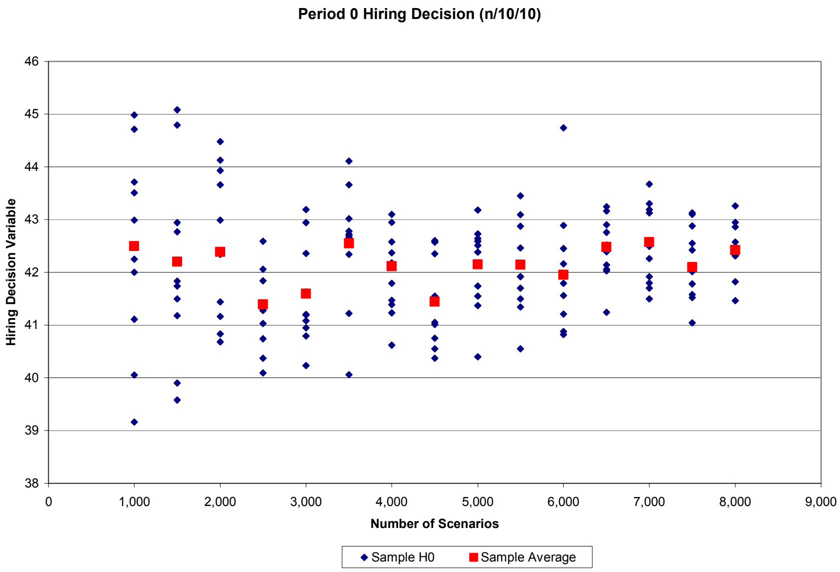


Figure 5-9 Initial Hiring for Sample Problems (N/10/10)

Again the variance is slowly decreasing with an increased number of scenarios, but the sample average values do not change in a statistically significant manner. However, if we examine the computational effort required to solve the problems, expressed as the resource usage statistic, we see a clear pattern.

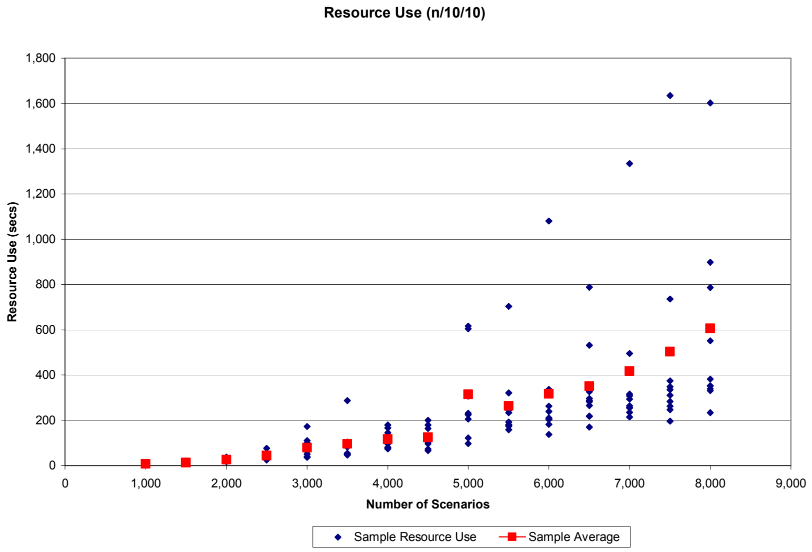


Figure 5-10 Resource Use for Sample Problems (N/10/10)

The average time required to solve the problem increases in a non-linear fashion with the number of scenarios. (Fitting a Power curve in Excel yields the function $y = 2 \cdot 10^{-6} x^{2.1736}$, where *x* is the number of scenarios and *y* is the resource use time in seconds. This curve matches the data with an R^2 value of .985.)

In summary, adding more scenarios has a significant negative impact on computational costs, along with a minor positive impact on solution quality when using 10 samples in each of the later stages. It may be possible that 10/10 sampling is not sufficient so in the next section I perform a similar test with 15 samples in each of the later stages.

5.4.2.2 N/15/15 Sampling

In this section I perform a similar test, but allow for 15 samples in each of the later stages. In this situation the total number of scenarios is equal to 225 times the number of stage one realizations. I again allow the number of first stage realizations to vary from 10 to 80, which implies the total number of scenarios varies from 2,2,50 to 18,000. The results are summarized in the follow g table.

	Objective		Stage 0 Hiring		Reource Use	
		Standard		Standard		Standard
Scenarios	Mean	Deviation	Mean	Deviation	Mean	Deviation
2,250	331,867	12,274	42.7	1.97	42.8	10.06
3,375	331,410	13,543	41.9	1.81	82.8	39.11
4,500	334,829	11,519	42.4	1.36	162.4	73.51
5,625	326,697	5,936	41.6	0.82	480.2	245.85
6,750	326,867	7,596	41.7	0.98	404.3	134.90
7,875	334,226	8,315	42.6	1.18	567.8	377.92
9,000	331,780	4,528	42.2	0.65	755.5	533.46
10,125	326,329	5,961	41.5	0.86	758.0	274.68
11,250	332,327	6,088	42.0	0.82	1,487.0	1238.63
12,375	331,347	6,709	42.2	0.84	1,188.6	501.18
13,500	330,851	9,031	42.0	1.13	1,451.1	1132.20
14,625	335,270	4,089	42.5	0.61	1,406.2	1273.69
15,750	335,468	6,085	42.5	0.76	2,232.8	1186.20
16,875	332,135	4,973	42.0	0.65	1,761.8	566.22
18,000	334,734	4,505	42.5	0.56	2,157.0	474.68

Table 5-4 Empirical Properties of N/15/15 Models

The general convergence results are similar to the previous case, but the computational costs are much higher. The general conclusion is that the additional late stage realizations add considerable cost but little in the way of precision.

5.4.3 Sampling Properties of the MIP

I now examine the empirical properties of the mixed integer problem, the problem where the stage zero decision is constrained to be integer valued. Constraining the stage 0 decision clearly leads to an actionable decision, which the relaxation does not, but presumably at a slightly higher cost. Given that the MIP has only a single variable to branch on, the increased computational cost should be minimal.

These expectations are confirmed in the following data that list the results from solving a MIP with n/10/10 scenarios.

		Objective		Stage 0 Hiring		Reource Use	
			Standard		Standard		Standard
	Scenarios	Mean	Deviation	Mean	Deviation	Mean	Deviation
A	1,000	332,356	11,694	43.0	1.66	8.0	0.52
B	1,500	331,568	13,237	42.6	1.90	22.1	7.42
C	2,000	336,400	12,314	42.8	1.49	41.8	26.53
D	2,500	325,200	4,896	42.0	0.74	63.9	33.44
E	3,000	327,674	7,619	42.1	0.99	79.5	27.64
F	3,500	334,940	8,411	43.1	1.10	106.4	58.14
G	4,000	331,553	5,934	42.7	0.85	122.5	56.37
H	4,500	327,691	6,309	42.0	0.88	161.2	41.95
I	5,000	332,677	6,133	42.7	0.84	196.1	41.94
J	5,500	331,843	7,090	42.6	0.97	311.1	100.83
K	6,000	332,105	9,353	42.3	1.17	317.7	81.36
L	6,500	335,577	3,879	43.1	0.57	1,029.8	773.18
M	7,000	335,886	6,550	43.1	0.94	391.4	158.97
N	7,500	333,817	5,590	42.7	0.79	403.9	81.45
O	8,000	334,741	4,250	42.9	0.57	610.5	365.89

Table 5-5 Empirical Properties of N/15/15 MIP Models

If we again look at the results graphically we see the following:

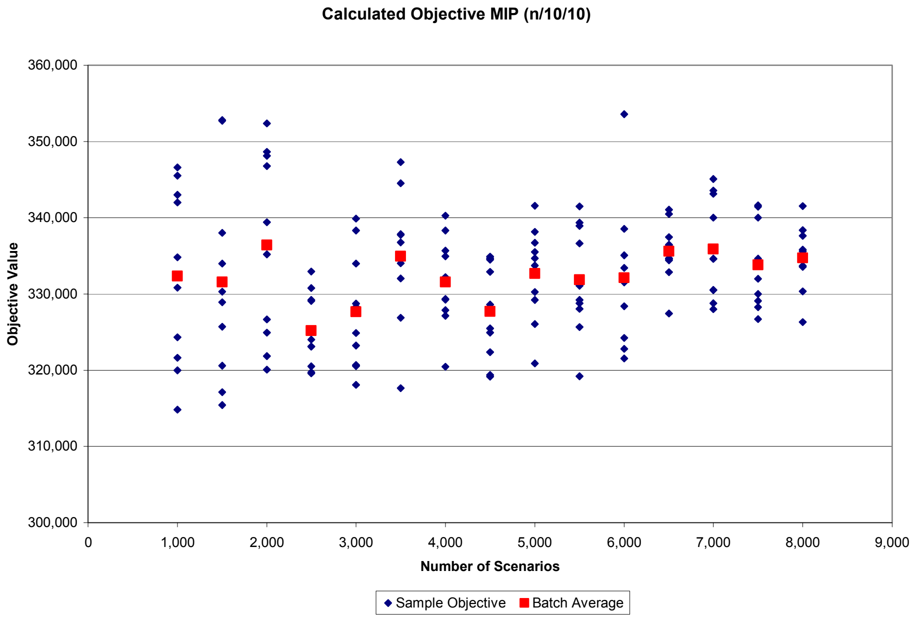


Figure 5-11 Objective Value for MIP N/10/10 Problem

The objective value is again distributed such that the variance declines slightly with the number of scenarios and the batch averages are within a fairly tight range. The next figure examines the stage 0 hiring decision.

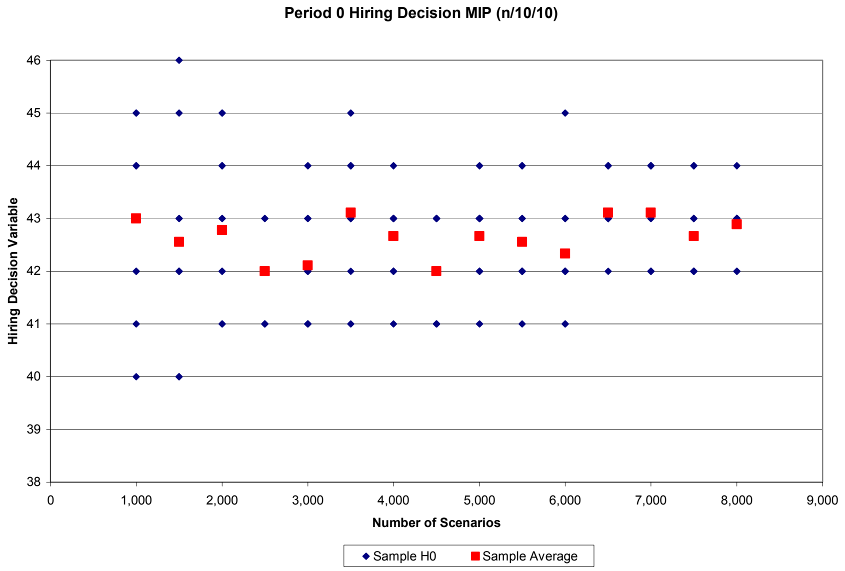


Figure 5-12 Initial Hiring Value for MIP N/10/10 Problem

In this case the graph is a bit different than the graph of the relaxation in that the hiring variables are all integer valued, while the batch average continues to be non-integer. The individual solutions include up to six discrete values in the low scenario case. Even in the high scenario situations three different candidate solutions are identified in each batch.

From a resource perspective the addition of the integrality constraint adds to the cost. However, since their is only a single variable to branch on, the additional cost is relatively small.

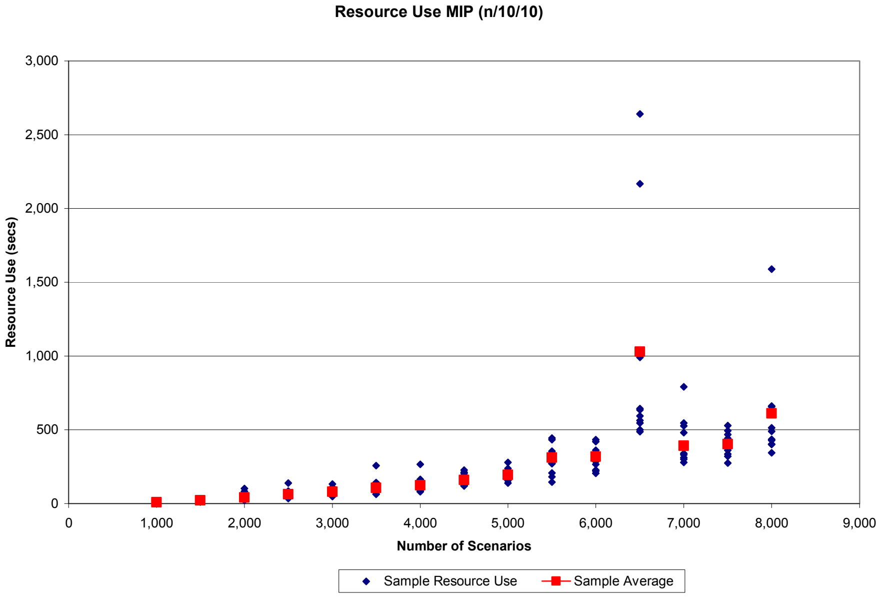


Figure 5-13 Resource Usage for the MIP n/10/10 Problem

5.4.4 SAA Based Algorithm

The results of the previous section suggest that finding a precise solution form a single optimization run may be quite expensive. They also suggest however that on average, even low scenario solutions provide good estimates. This suggests a batch solution algorithm that is a variation of the Sample Average Approximation approach and has three main steps

- **Identify Candidate Solutions**: solve a batch of sample path problems to identify one or more candidate solutions.
- **Evaluate Candidates**: calculate the expected outcome for each candidate against a reference set of scenarios. Select the candidate with the lowest expected cost.
- **Calculate Bounds**: calculate statistical bounds on the outcome and optimality gap.

The algorithm is presented in detail in the following figure

```
1. Identify Candidate Solutions
    a. Initialization
        i. Define realizations per stage $R_1, R_2, R_3$, and
           calculate the number of scenarios $N_S = R_1 R_2 R_3$
        ii. Define $N_B$ the number of batches
    b. Generate $N_B$ batches of $N_S$ scenarios each
    c. Solve $N_B$ optimization problems finding candidate
       optimal solutions $\hat{H}_0$ and objective value $\hat{z}$
2. Evaluate Candidates
    a. Generate an independent set of $N_C$ comparison
       scenarios
    b. Calculate the expected outcome of each candidate
       solution
    c. Select the best outcome
    d. If the selected outcome is superior to the second
       best at the $\alpha = .01$ level then continue, else
       repeat Step 2 with a larger sample size
3. Calculate Bounds
    a. Generate an independent set of $N_E$ evaluation
       scenarios ($N_E > N_C$)
    b. Calculate the expected outcome of selected
       solution for each $N_E$ scenario
```

Figure 5-14 Sample Average Algorithm

5.5 Numerical Analysis

5.5.1 Screening Analysis

In this section I perform a series of computational experiments to estimate the optimal level of hiring and expected cost for various project conditions. As an initial objective I seek to determine what factors have the biggest impact on the optimal level of excess hiring. To accomplish this goal I conduct a preliminary screening experiment. I use a fractional factorial experiment of resolution IV. This experiment will allow the unconfounded estimation of all main effects.

In this initial screening I consider the following seven factors:

1. **Hiring Cost**: the cost to hire new agents in period 0.
2. **Termination Cost**: the cost to terminate an employee.

3. **SLA Target**: the contractual SLA target.
4. **Operational Penalty Rate**: the penalty rate per point of SLA shortfall assigned beginning in period three.
5. **Hiring Cost Differential**: the incremental cost, relative to the period zero hiring cost, to hire agents in periods 1-3.
6. **Period Two Penalty Rate**: the penalty rate per point of SLA shortfall assigned in period two.
7. **Minimum Launch Expected SL**: the minimal expected service level allowable in the first period.

The first four factors essentially address base operating characteristics of the system, such as the cost to hire and fire along with the financial constraints placed on SLA attainment. Factors five through seven are focused on the transition phase. Factor five estimates the inefficiency from last minute hiring, while factors six and seven indicate how poor service level performance can be during the transition period; factor 6 specifies how severely shortfalls are penalized in the second month of operation and factor seven specifies the required expected performance in the first month of launch.

The experimental design is a 16 run 2_{IV}^{7-3} experiment and allows the unconfounded estimation of all main (single factor) effects.

	A	B	C	D	E	F	G
1	-	-	-	-	-	-	-
2	+	-	-	-	+	-	+
3	-	+	-	-	+	+	-
4	+	+	-	-	-	+	+
5	-	-	+	-	+	+	+
6	+	-	+	-	-	+	-
7	-	+	+	-	-	-	+
8	+	+	+	-	+	-	-
9	-	-	-	+	-	+	+
10	+	-	-	+	+	+	-
11	-	+	-	+	+	-	+
12	+	+	-	+	-	-	-
13	-	-	+	+	+	-	-
14	+	-	+	+	-	-	+
15	-	+	+	+	-	+	-
16	+	+	+	+	+	+	+

Factor Definitions	-	+
A Hiring Cost	500	1500
B Hiring Cost Differential	0%	50%
C Termination Cost	0	3200
D SLA Target	80%	90%
E Operational Penalty Rate	50,000	320,000
F Period Two Penalty Rate	0%	100%
G Min Launch Expected SL	50%	75%

Table 5-6 Screening Analysis Design of Experiment

At each design point I first solve the Mean Value problem. I then solve the stochastic problem using the process outlined in 5-13. I solve 15 instances of the problem at each design point using a 30/10/10 realization pattern for 3,000 scenarios. I then run an evaluation comparing the average hiring level, rounded to the nearest integer, with the four closest neighboring solutions. The comparison is run against a set of 9,000 scenarios generated from a 60/15/10 realization pattern. The solution with the best expected outcome is selected and the results from the 9,000 scenario run are used to calculate the statistical properties of the outcome. The mean value solution is evaluated against the same set of scenarios to estimate its expected outcome. The Value of the Stochastic Solution (VSS) is then calculated as the difference between the expected outcome of implementing the mean value solution and implementing the stochastic solution.

The results of this analysis are summarized in the following table. The table lists the coded value of each factor along with the hiring level determined by the mean value program and the hiring level selected through the evaluation process outlined above. The table also lists the VSS calculated.

								MV	Best	Best		
DP	A	B	C	D	E	F	G	Hire	Hire	Outcome	VSS	VSS %
1	-	-	-	-	-	-	-	38	38	266,460	0	0.0%
2	+	-	-	-	+	-	+	50	49	356,154	3,408	0.9%
3	-	+	-	-	+	+	-	50	52	340,442	3,360	1.0%
4	+	+	-	-	-	+	+	50	49	375,309	600	0.2%
5	-	-	+	-	+	+	+	50	50	367,397	0	0.0%
6	+	-	+	-	-	+	-	43	45	390,142	2,393	0.6%
7	-	+	+	-	-	-	+	50	49	328,474	5,435	1.6%
8	+	+	+	-	+	-	-	40	38	318,460	7,777	2.4%
9	-	-	-	+	-	+	+	54	56	387,236	900	0.2%
10	+	-	-	+	+	+	-	56	60	469,510	6,769	1.4%
11	-	+	-	+	+	-	+	50	49	338,152	2,162	0.6%
12	+	+	-	+	-	-	-	39	38	346,832	1,633	0.5%
13	-	-	+	+	+	-	-	38	38	309,384	0	0.0%
14	+	-	+	+	-	-	+	50	49	388,464	0	0.0%
15	-	+	+	+	-	+	-	54	52	401,208	623	0.2%
16	+	+	+	+	+	+	+	56	58	491,623	1,136	0.2%

Table 5-7 Experimental Results

A few important observations are apparent from this data. First and foremost, the mean value problem often provides very good results. In several cases the mean value problem finds the same hiring level as the stochastic problem and hence the VSS is zero. In cases where the stochastic model finds a different solution, the Value of the Stochastic Solution is relatively small. In the best case, the VSS represents about a 2.4% improvement over the mean value solution. It is apparent from this analysis that the main benefit from the stochastic model will not be in improving the objective, but rather in understanding the statistical distributions of the outcomes. I return to this topic later, but first analyze the impact of the experimental factors on the outcomes.

The following table summarizes the Main Factor Effects of each experimental factor on two response variables, the stage zero hiring level and the expected cost of operation over the start-up period.

		Main Effects			
	Factor Definitions	Hiring		Objective	
A	Hiring Cost	0.06		12,429	*
B	Hiring Cost Differential	0.00		180	
C	Termination Cost	-0.38		3,596	*
D	SLA Target	0.94		12,174	*
E	Operational Penalty Rate	0.56		3,344	
F	Transition Penalty Rate	2.31	*	17,828	*
G	Min Launch Expected SL	1.50	*	5,949	*
	Average	48.13		367,203	

* Indicates significance at the 95% level

Table 5-8 Main Effects on Hiring and Objective

5.5.1.1 Effects on Hiring

The data in Table 5-7 shows that for this particular volume estimates the average optimal hiring level is just under 49, but varies from 38 to 56. Table 5-8 decomposes this variability into the effect that results from each variable. The most significant factors to impact the stage zero hiring decision relate to the degree that the SLA must be met during the transition phase; factors F and G are in fact the only statistically significant factors. In a tight start up, where the transition penalty and minimum staffing levels are both high, the optimal hiring is on average higher by about 7.6 individuals, an increase of nearly 16% from the mean.

Other factors have a much smaller impact on initial hiring, and are not statistically significant. Raising the steady state service level requirement only increases hiring by about 1.8 on average.

A tighter service level requirement, expressed as a higher operational penalty rate, increase hiring by only about 1 full time equivalent. Conversely, making it more expensive to terminate employees depresses initial hiring by only about 1 FTE. Finally, the cost of hiring has very little impact on the optimal number to hire as these costs are dominated by other costs in the decision process.

5.5.1.2 Effects on Objective

The last column of Table 5-8 provides additional information about how these factors drive the expected cost of operation. Again, the service level requirements during transition have a practical and statistically significant impact on the expected outcome. Tight transition requirements add about 12.9% to the start up cost. The steady state service level requirement and penalty rate also add significantly to the overall cost.

Termination costs on the other hand have a more significant impact on the total cost than they do on the stage zero hiring decision. While a high cost of terminating employees adds to the total cost of operation, it has a relatively limited impact on the initial decision. The ability to downsize the staff once uncertainty is revealed is a valuable recourse option, even if the cost is high. Since only a few agents will be terminated the increased cost of termination does little to lower the initial hiring level.

Similarly, the cost of hiring has a significant impact on the cost of operation but almost no impact on the hiring decision; the hiring cost shifts the cost by 6.8% but the hiring decision by only .4% on average. The rationale is that while the cost to hire has a major impact on the cost to start up the project, there are no other options in this model.

5.5.2 Distribution of Outcomes

The results of the previous section indicate that solving the stochastic model may lead to moderately reduced cost launches as compared to solving the mean valued solution. However, in addition to decreasing the expected cost of operation, the stochastic model provides the important of providing insight to decision makers on the statistical distribution of outcomes. The mean value model provides a single point-estimate of model parameters while the stochastic model allows us to estimate the statistical distribution of outcomes.

In the solution process outlined in Figure 5-13 we estimate the outcome of the candidate solution by evaluating the solution against a set of evaluation scenarios. The expected outcome for any random parameter is the average result over all the evaluations scenarios. If we examine the scenario results in detail we can estimate the distribution.

As an example, the following figure plots a histogram of the objective value generated for DP1. Recall from table 5-7 that the expected outcome of this startup is $266,458.

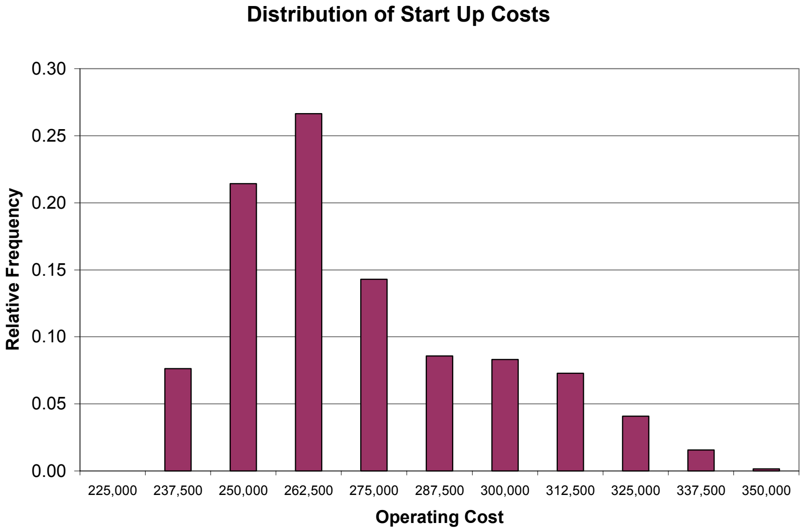


Figure 5-15 Distribution of Start Up Costs – DP1

The graph shows that the outcome is positively skewed with outcomes as much as $83,000 (31%) above the mean possible. In this particular case there is a 21% probability that start up costs will exceed $300,000. A similar analysis generates the following histogram of the ending staff level.

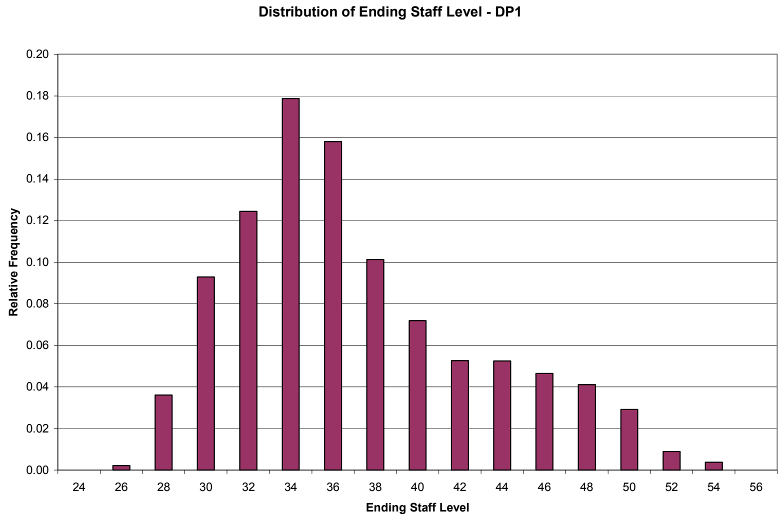


Figure 5-16 Distribution of Ending Staff Level – DP1

The mode of this distribution is 34, while the initial hiring level was 38; indicating that on average the staff level will decrease by 4 agents over the course of the start-up. In this particular case there is in fact an approximately 60% probability that the ending staff level will be less than the number originally hired. The staff level reduction occurs because of attrition, but also because of terminations. In this particular case the expected number of post launch terminations is 1.6. The logic is fairly straightforward. Given uncertainty and learning curve issues the optimal policy calls for acquiring some spare capacity prior to launch.

The following figure plots the distribution for the number of agents hired and fired post launch. It may seem odd that the expected number of hire and fires are both positive, but the specific action will depend on how demand is realized. In rare cases the model may call for post launch hiring and firing in the same scenario.

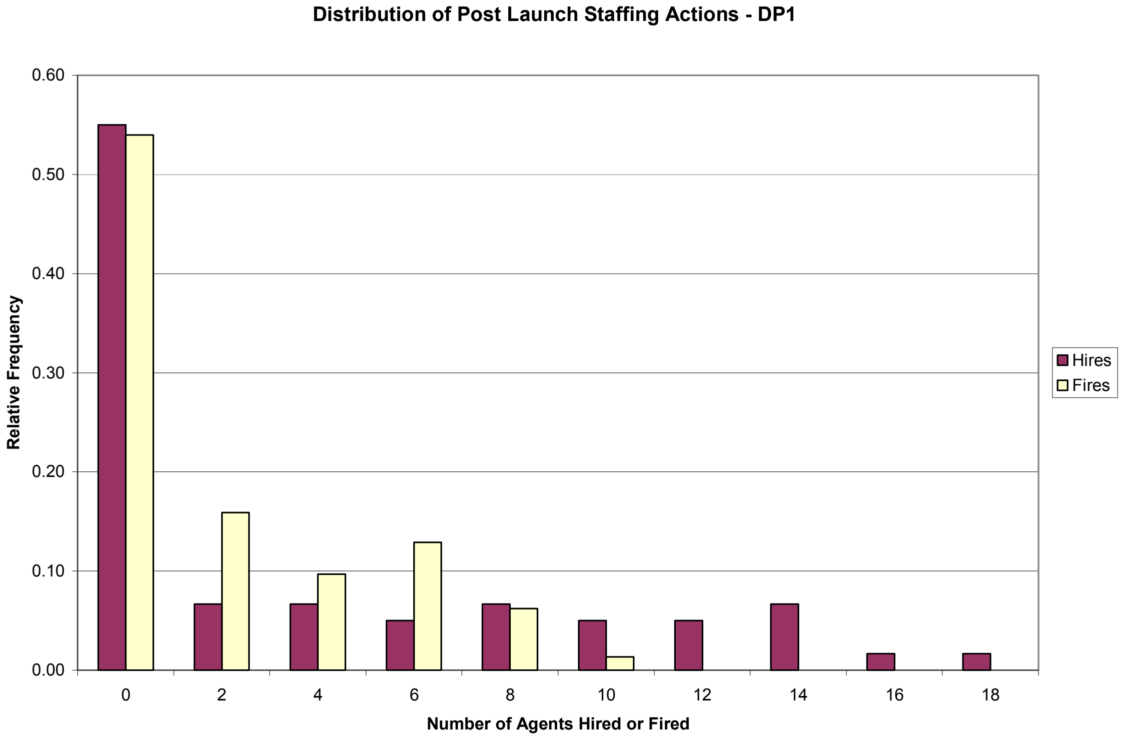


Figure 5-17 Distribution of Post Launch Hiring and Firing – DP1

The above graphs show the distribution of key outcome graphically for a single design point. To get a sense of how these distributions may vary with the model's control factors I list some summary statistics in the following tables. The first table summarizes the distribution of the objective and ending hiring level, while the second summarizes the post launch hiring and firing.

	Cost					Ending Staff				
DP	Avg	SD	Min	Max	Range	Avg	SD	Min	Max	Range
1	266,460	24,496	227,229	345,676	118,447	36.1	5.8	24.9	53.6	28.7
2	356,154	19,052	322,936	445,399	122,463	36.6	5.2	26.1	52.8	26.7
3	340,442	59,465	284,468	592,640	308,172	36.7	5.3	26.1	52.1	26.0
4	375,309	49,156	322,376	551,197	228,821	36.1	4.9	26.0	49.0	23.0
5	367,397	62,719	313,761	635,479	321,718	45.3	2.2	36.8	52.8	16.0
6	390,142	60,797	327,385	599,291	271,906	40.9	2.0	33.1	46.7	13.7
7	328,474	7,180	308,240	368,134	59,894	44.3	2.2	36.0	49.7	13.7
8	318,460	33,559	276,458	444,065	167,607	38.0	5.1	28.6	55.6	27.0
9	387,236	56,808	318,638	567,845	249,207	43.4	5.2	31.2	56.0	24.8
10	469,510	73,557	395,926	747,177	351,252	44.8	6.1	31.6	61.5	29.9
11	338,152	31,519	289,148	449,185	160,037	44.5	7.1	31.1	67.2	36.1
12	346,832	38,252	282,628	453,562	170,934	36.2	2.7	28.6	46.5	18.0
13	309,384	35,323	247,620	422,780	175,160	46.5	8.4	30.6	70.9	40.3
14	388,464	18,862	357,240	462,380	105,141	44.3	2.2	36.0	49.0	13.0
15	401,208	63,593	331,165	605,114	273,949	47.0	2.3	38.2	52.0	13.8
16	491,623	74,325	421,963	778,523	356,561	52.5	2.6	42.6	59.1	16.4

Table 5-9 Summary of Cost and Ending Staff

	Post Launch Hiring					Post Launch Firing				
DP	Avg	SD	Min	Max	Range	Avg	SD	Min	Max	Range
1	3.41	4.91	0.00	16.72	16.72	1.56	2.34	0.00	8.84	8.84
2	0.21	0.80	0.00	4.53	4.53	8.31	5.50	0.00	19.84	19.84
3	0.02	0.16	0.00	1.23	1.23	10.90	6.02	0.00	22.83	22.83
4	0.00	0.00	0.00	0.00	0.00	8.63	5.58	0.00	19.84	19.84
5	0.14	0.62	0.00	3.85	3.85	0.00	0.00	0.00	0.00	0.00
6	0.20	0.66	0.00	3.19	3.19	0.00	0.00	0.00	0.00	0.00
7	0.05	0.29	0.00	2.17	2.17	0.00	0.00	0.00	0.00	0.00
8	3.92	5.40	0.00	17.79	17.79	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	7.63	5.91	0.00	20.53	20.53
10	0.08	0.49	0.00	3.58	3.58	10.06	6.76	0.00	23.53	23.53
11	3.22	5.34	0.00	18.65	18.65	2.99	3.56	0.00	12.53	12.53
12	2.01	2.67	0.00	9.95	9.95	0.08	0.33	0.00	2.53	2.53
13	12.99	9.08	0.00	32.86	32.86	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.02	0.12	0.00	0.84	0.84	0.00	0.00	0.00	0.00	0.00
16	0.08	0.53	0.00	4.00	4.00	0.00	0.00	0.00	0.00	0.00

Table 5-10 Summary of Post Launch Staffing Actions

The data shows the wide range of outcomes possible. In many cases the cost of operation varies by more than $250,000, and final staffing varies by as much as 40.

5.6 Summary and Conclusions

5.6.1 Summary

This model examines the issue of how to staff a new call center outsourcing project in the face of uncertainty about demand. I developed a model that is motivated by the empirical analysis in Chapter 2 and extends the scheduling model developed in Chapter 4. I use a version of that model to generate an estimated aggregate service level curve for a range of possible demand outcomes. Given those estimates I develop a multistage model of the project start-up process that accounts for agent learning, and attrition.

The model is developed as a multistage stochastic problem, with an integer constrained decision in the first stage. The analysis shows that an approach, based on Sample Average Approximations, provides tractable solutions to this problem, exploiting the fact that the stage 0 decision is scalar valued.

Detailed numerical analysis shows that the stochastic formulation provides only a moderate benefit in terms of lowering the cost of the launch. It does however provide a significant qualitative benefit in terms of contingency planning by providing estimated distributions of key parameters such as total cost, hiring, or firing. Given these distributions, managers can make more informed decisions regarding pricing as well as staffing contingencies.

A key insight in this model is the importance of the transition phase of the start-up. Given the nature of the learning curve, it is extremely difficult, and expensive, to achieve targeted service levels in the first few month of the launch. Managers have responded by attempting to lower expectations for the service level over the first few months of launch. My analysis supports that strategy, but also helps to quantify the costs associated with attempting to meet service level commitments in the first few moths. The model also helps to quantify the degree to which the project should be overstaffed at start-up, a practice currently mot employed at the company I analyzed. The analysis indicates the combined effects of learning and attrition provide strong incentive to err on the side of over hiring.

5.6.2 Contributions

Like the other chapters in this dissertation, this chapter presents the development and evaluation of an applied OR model. This model is, to my knowledge, relatively unique in the literature. I have seen no other model that addresses the issue of project staffing in a start up outsourcing project. Perhaps this model is too application specific for presentation in the OR literature, but it does, in the author's opinion, synthesize multiple concepts to address an important practical issue. The model integrates research on queuing theory, call center operations, stochastic optimization and learning curves. The primary contribution of this particular chapter is the integration of these concepts into a formulation that can solve a specific problem of practical importance.

5.6.3 Management Implications

The analysis presented in this chapter identifies several issues with important managerial implications. The most important implications arise from issues related to the learning curve. At launch time all agents are inexperienced and subject to rapid productivity improvement. Failure to account for this learning when planning a startup, as is the case in the company I studied, will often lead to significant start up challenges. A second key implication is the need to plan for staffing flexibility. The analysis shows that the optimal staffing level at start-up time is not likely to be the optimal staffing level once the transition phase has been completed. Managers must maintain the flexibility to add additional staff, or if necessary to remove staff once uncertainty is revealed and learning has occurred. Lastly, is the issue of the quality of service during the transition phase. In projects where learning is significant rapid attainment of service level objectives is possible, but very expensive.

5.6.4 Future Research

Several extensions to this model are possible.

- **Alternative Utility Functions**: the model has implicitly assumed a risk neutral decision maker seeking to minimize expected cost. The stochastic model formulation allows for other utility models. An approach such as minimax that seeks to minimize the maximum cost can be implemented quite easily.
- **Investment in Learning**: the learning curve has a significant impact on the project start-up. Throughout this analysis we assumed that the learning rate was exogenous. One potential extension of this research is to examine the benefit that would accrue from

investments that increase the rate of learning, for example higher levels of investment in training or knowledgebase development.

- **Phased Start-Up**: the analysis presented here is based on a single cut over of support services. The model is however motivated in part by the service level collapse observed during a phased roll out process. Each time a new cutover occurred, things became worse as the project got further and further behind. Extending this model to look at a phased rollout would be difficult, but beneficial.

6 The Cross Training Model

6.1 Overview

The cross training model further examines the operations of a call center. I investigate the option of cross training a subset of agents so that they may serve calls from two separate projects, a process I refer to as *partial pooling*. Since queuing systems have natural economies of scale cross training agents will increase overall system performance. However, given the investment required in training, and the potential requirement to pay a differential wage, it may not be beneficial to cross train all agents. Alternatively, consider the case of a multi-lingual call center. Staffing the call center with multi-lingual agents will increase the efficiency of the center. But presumably multi-lingual agents are more difficult to find and can command a premium wage.

This model seeks to quantity the benefits of partial pooling and characterize the conditions under which pooling is most beneficial. We then determine the optimal number of agents to cross train given the training investment and incremental wage paid to cross skilled agents.

The staffing challenge in this model is to find the optimal mix of agents so as to achieve the global SLA target with a high probability and at the lowest possible cost. In partial pooling a small subset of *super agents* are cross trained to take calls from two projects. The call center can then be viewed as a skills-based routing (SBR) model with two skills. Super agents possess both skills, while *base agents* have only one skill set. It is clear that cross training all agents will increase the service level of the call center for a fixed level of staffing. My hypothesis is that cross training a small number of agents can deliver a substantial portion of the benefit and my objective is to find the level of cross training that minimizes staffing costs, while satisfying the service level constraint with high probability.

I examine the case of cross training between two projects (or two language groups) and assume that the skills based routing system is configured as follows:

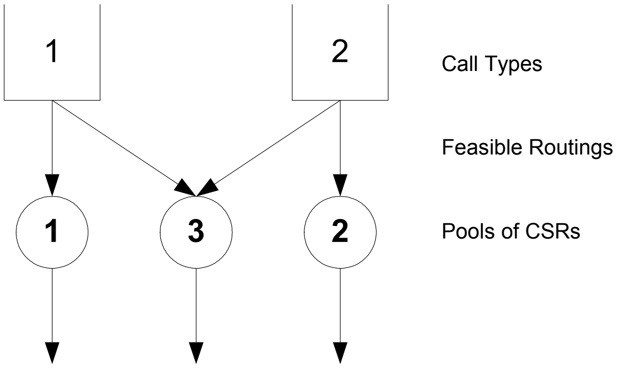


Figure 6-1 Pooling Model

We have two call types, one for each project, and three agent pools. Pool 1 has skill 1 and can service call type 1. Similarly pool 2 services call types 2. Pool 3 is cross trained (or multilingual) and can service calls from either queue.

6.2 Partial Pooling in Steady State

In this section we analyze system performance in steady state using simulation. In this analysis I assume a simple routing model. When a call arrives it is routed to an appropriately skilled base agent if one is available. Only if all base agents are busy will the call be sent to a super agent. If all super agents are busy, then the call is placed in queue to be serviced by the next available qualified agent. When a base agent becomes available, she will pull an appropriate call from the queue if available. When super agents become available they take a call from the virtual queue with the largest number of waiting calls. Other routing models are possible; in particular we may wish to route calls based on a month to date SLA achievement.

The model is analyzed using an extension of the simulation model described in (Robbins, Medeiros *et al.* 2006). This general purpose call center simulation model has been modified to support the pooling approach described in this paper, and to execute a search based optimization algorithm. The model generates two independent Poisson arrival processes and services those calls using the routing scheme described above. The model is configured to vary the number of pooled agents for any given arrival rate pair.

6.2.1 The TSF Response Function

In the single queue, single resource pool case, we have an analytical expression for the service level as a function of arrival rates and staffing (equation (3.36)) and can easily generate a plot of the TSF as a function of staffing (see Figure 4-5). In the pooling case the situation is considerably more complicated. There are no known analytical expressions available to calculate the service level. Based on intuition we expect the service level is increasing in the number of base agents and the number of super agents. To verify this intuition I use simulation to create the following graphical representation of the TSF as a function of the number of agents.

In this simulation I assume that each queue receives calls at a rate of 100 calls per hour, that in each cases talk time averages 12 minutes, callers have an average patience of 350 seconds, and the service level is based on a 120 second hold time. I vary the number of agents assigned to each base pool and the number of agents assigned to the super agent pool independently. For each staffing combination I simulate operations for two days, and perform 25 replications.

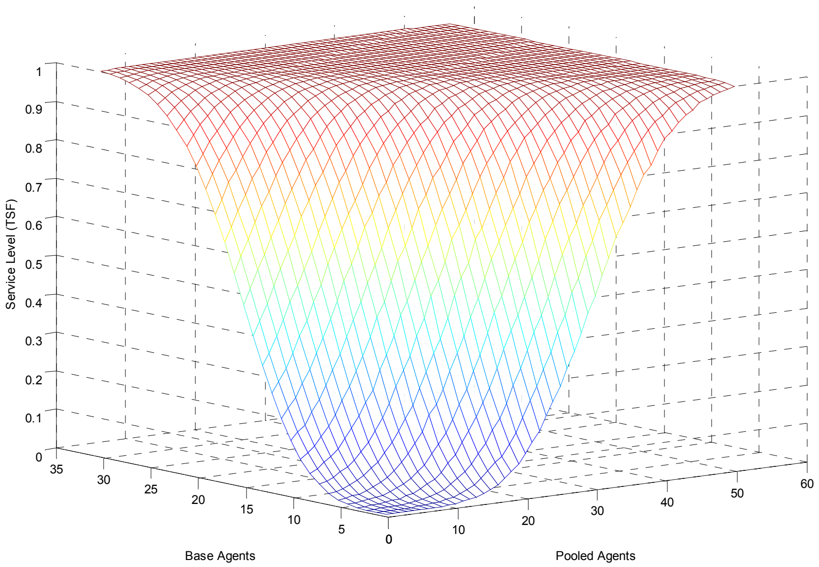


Figure 6-2 Pooled Model TSF Surface

In Figure 6-2 I show a three dimensional plot of the TSF surface. The graph illustrates a large plateau of 100% TSF when the total number of agents is large. Similarly a small plateau at 0% TSF exists when the total number of agents is small. In between the surface exhibits an S shaped profile. Figure 6-3 is a contour plot of this data in two dimensions. The contour plot shows a series of iso-service level lines, agent combinations that deliver the same service level. So for example, to achieve a 95% service level we need roughly 25 agents in each pool or 50 agents overall. However, in a pure pooled mode the same service level can be achieved with a total of only 45 pooled agents.

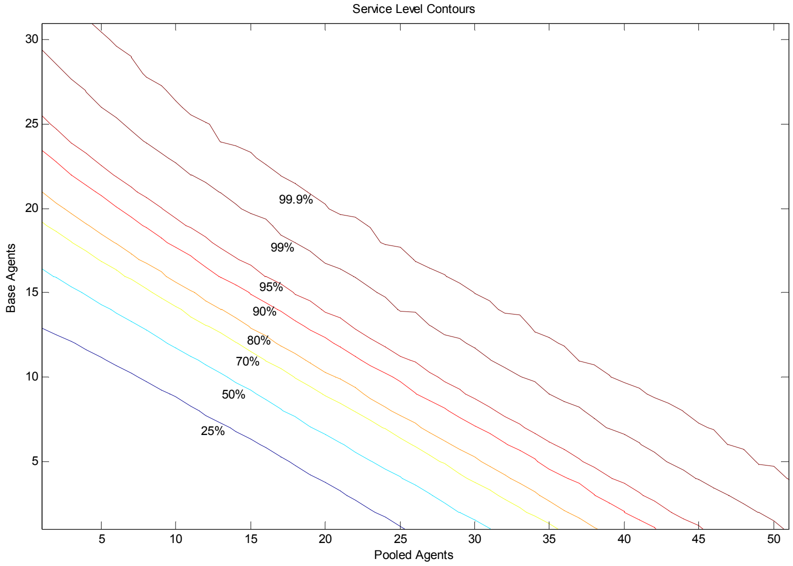


Figure 6-3 Pooled TSF Contour Diagram

Though difficult to see, close inspection reveals that the iso-service lines are not straight, but have a convex *bowed* shape. This is further illustrated in the next figure where I show the 80% TSF contour with a line connecting the end points. The convexity of the contour implies that the cost minimizing combination of pooled and base agents may be in the interior.

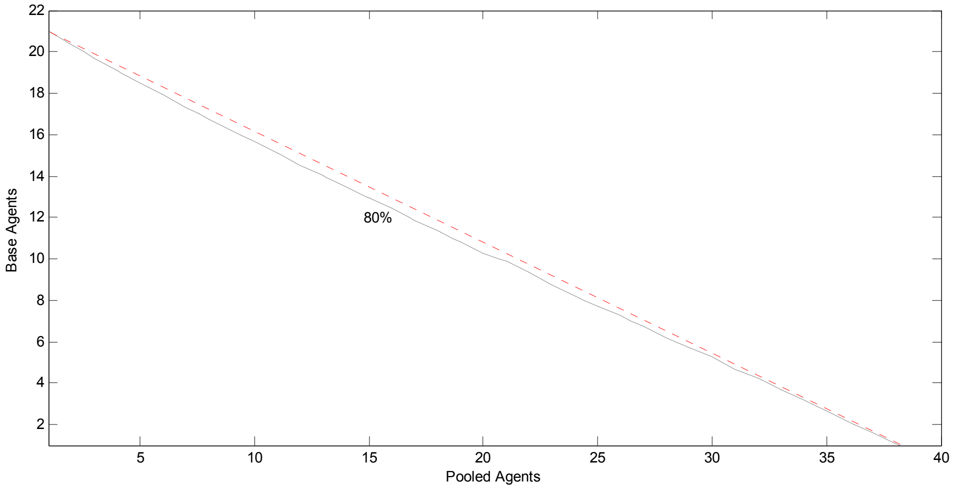


Figure 6-4 80% TSF Contour

An alternative way to look at this data is to examine the marginal impact on the service level by adding one type of agent, while holding the other agent pool fixed. This is illustrated in the following pair of graphs.

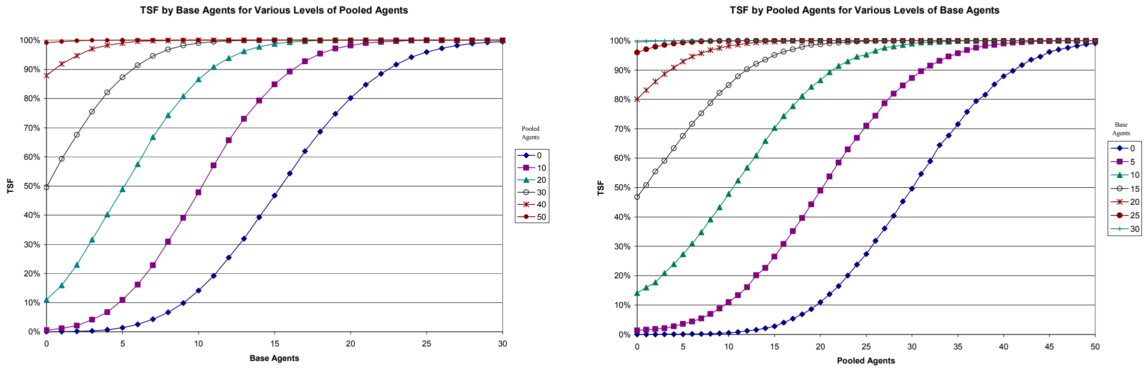


Figure 6-5 Marginal Impact on Service Level

On the left side I vary the number base agents while holding the number of pooled agents fixed. With zero pooled agents we get the standard TSF curve as seen in Figure 4-5. When pooled agents are in place the TSF curve is effectively shifted to the left and the service level for any level of base agents. The right side graph reveals a similar relationship when the number of base agents is held constant.

6.2.2 Symmetric Projects in Steady State

In this experiment I test the impact of pooling on steady state performance with symmetric projects. Consider two statistically identical projects each staffed with 36 agents and receiving calls at a constant rate λ. Talk time has an exponential distribution with mean 12 minutes and the mean time to abandon is 350 secs. The service level is measured against a two minute hold time. I evaluate the situation where the total number of agents remains constant, but each project contributes between 0 and 36 agents to the pool. The first graph shows the service level for each level of pooling when λ is 200 calls per hour. The next two graphs on the left side show the service level of arrival rates of 180 and 220. On the right hand side I plot the abandonment rate. In each case I plot TSF and abandonment rate for one of the projects. (Because of the symmetric nature of the model, each project has the same curve.) The data was generated by simulating five days of operations over 50 replications. In each curve I show the sample average along with a 90% confidence interval, where the confidence interval is calculated by

$$\overline{x}(n) \pm t_{n-1,1-\alpha/2}\sqrt{\frac{s^2(n)}{n}} \tag{6.1}$$

where n is the number of replications, $\overline{x}(n)$ is the sample average, $s^2(n)$ is the sample variance and $t_{n-1,1-\alpha/2}$ is the critical value from the Student's t-distribution.

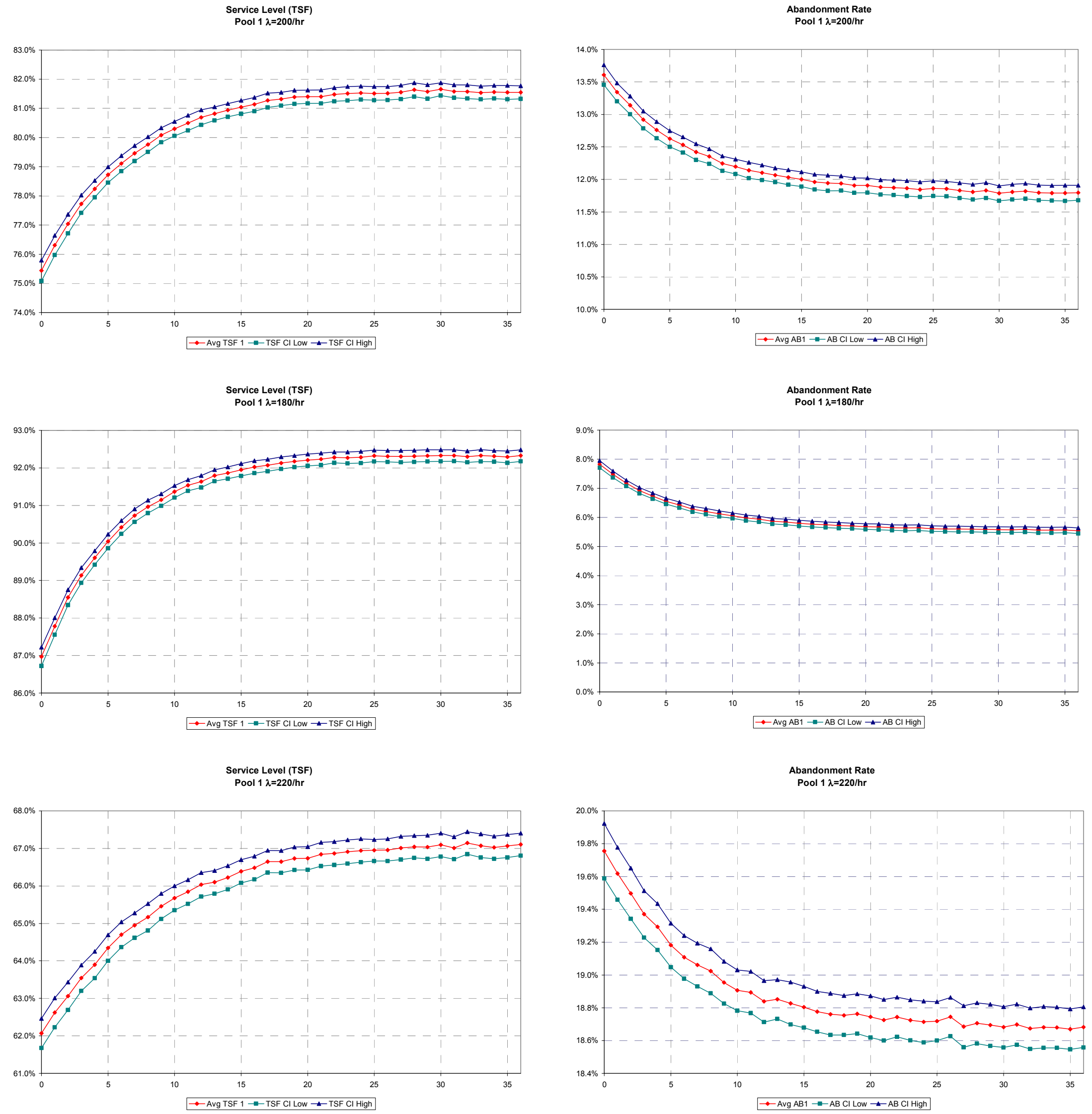


Figure 6-6 Impact of Pooling with Fixed Staffing Levels

These graphs reveal that a small level of pooling yields improvement, but that the return on cross training declines rapidly. In each case cross training 10 agents provides the bulk of the benefit and cross training beyond 15 agents provides very limited benefits. In each case cross training can boost TSF by 5%-6%, while the biggest improvement is in the medium volume (200/hr) case. Abandonment is reduced by about 1% in the high volume case, 1.8% in the middle case, and 2.3% in the slow case.

6.2.3 Steady State Differential Rates

The previous analysis reveals that moderate benefits are achieved when agents are cross trained, and the amount of improvement depends on the spare capacity in the system. However in that analysis both projects had the same arrival rate. A more interesting case occurs when the arrival rates are different as may be the case if rates are subject to forecast error. In the next analysis I allow arrival rates to vary independently from target by ± 10%. Total staffing is fixed at 72, so that in the now pooling each project has 36 agents, a staff level that results in an approximately 76% service level with no pooling. The following tables summarize the resulting TSF measures under various arrival rate combinations.

TSF Total

λ1	λ2	0	5	10	15	20	25	30	35
180	180	87.9%	90.8%	91.9%	92.5%	92.7%	92.8%	92.9%	92.8%
180	200	81.5%	85.3%	86.9%	87.6%	88.0%	88.1%	88.1%	88.0%
180	220	73.6%	78.8%	80.9%	81.9%	82.4%	82.5%	82.5%	82.5%
200	200	76.1%	79.3%	80.8%	81.6%	81.9%	82.0%	82.1%	81.9%
200	220	68.8%	72.3%	74.2%	75.0%	75.3%	75.6%	75.6%	75.5%
220	220	62.3%	65.1%	66.6%	67.5%	67.8%	68.0%	68.0%	68.1%

Δ TSF Total

λ1	λ2	5	10	15	20	25	30	35
180	180	2.8% *	1.1% *	0.6% *	0.2% *	0.1% *	0.0%	0.0%
180	200	3.8% *	1.6% *	0.7% *	0.3% *	0.1% *	0.0%	0.0%
180	220	5.2% *	2.2% *	1.0% *	0.5% *	0.1% *	0.0%	-0.1% *
200	200	3.2% *	1.5% *	0.8% *	0.3% *	0.1% *	0.1% *	-0.2% *
200	220	3.6% *	1.9% *	0.8% *	0.3% *	0.2% *	0.0%	-0.1% *
220	220	2.8% *	1.5% *	0.9% *	0.3% *	0.3% *	0.0%	0.1%

* indicates statistical significance at the .9 level

Table 6-1 Impact on Overall TSF of Pooling with Fixed Staffing Levels

TSF Pool 1

λ1	λ2	0	5	10	15	20	25	30	35
180	180	86.8%	90.4%	91.7%	92.4%	92.7%	92.8%	92.8%	92.8%
180	200	86.8%	87.5%	87.7%	87.8%	87.7%	87.6%	87.4%	87.3%
180	220	86.8%	84.7%	82.9%	82.1%	81.4%	80.9%	80.6%	80.2%
200	200	75.5%	78.9%	80.5%	81.4%	81.8%	82.0%	82.0%	82.0%
200	220	75.5%	75.3%	75.1%	74.9%	74.7%	74.4%	74.3%	74.0%
220	220	61.9%	65.0%	66.4%	67.6%	68.0%	68.1%	68.2%	68.2%

Δ TSF Pool 1

λ1	λ2	5	10	15	20	25	30	35
180	180	3.5% *	1.3% *	0.7% *	0.2% *	0.1% *	0.0%	0.0%
180	200	0.7% *	0.2% *	0.1%	0.0%	-0.1% *	-0.2% *	-0.1% *
180	220	-2.2% *	-1.7% *	-0.8% *	-0.7% *	-0.5% *	-0.3% *	-0.4% *
200	200	3.4% *	1.6% *	1.0% *	0.4% *	0.1% *	0.1%	-0.1% *
200	220	-0.2%	-0.2%	-0.3% *	-0.2% *	-0.2% *	-0.1% *	-0.2% *
220	220	3.1% *	1.5% *	1.1% *	0.4% *	0.2% *	0.1%	0.0%

* indicates statistical significance at the .9 level

Table 6-2 Impact on Low Volume Project TSF of Pooling with Fixed Staffing Levels

TSF Pool 2

λ1	λ2	0	5	10	15	20	25	30	35
180	180	89.0%	91.1%	92.1%	92.5%	92.8%	92.9%	92.9%	92.8%
180	200	76.7%	83.3%	86.1%	87.5%	88.2%	88.5%	88.7%	88.7%
180	220	62.7%	73.9%	79.3%	81.8%	83.2%	83.8%	84.1%	84.3%
200	200	76.7%	79.7%	81.0%	81.7%	82.0%	82.0%	82.2%	81.9%
200	220	62.7%	69.6%	73.4%	75.1%	75.9%	76.6%	76.8%	76.8%
220	220	62.7%	65.2%	66.7%	67.4%	67.6%	67.9%	67.8%	68.0%

Δ TSF Pool 2

λ1	λ2	5	10	15	20	25	30	35
180	180	2.1% *	0.9% *	0.4% *	0.3% *	0.1% *	0.0%	-0.1% *
180	200	6.6% *	2.8% *	1.4% *	0.7% *	0.4% *	0.1% *	0.0%
180	220	11.3% *	5.3% *	2.5% *	1.5% *	0.6% *	0.3% *	0.2% *
200	200	3.0% *	1.3% *	0.6% *	0.3% *	0.0%	0.1% *	-0.2% *
200	220	7.0% *	3.8% *	1.7% *	0.9% *	0.7% *	0.2% *	0.0%
220	220	2.6% *	1.5% *	0.6% *	0.2% *	0.3% *	-0.1%	0.2% *

* indicates statistical significance at the .9 level

Table 6-3 Impact on High Volume Project TSF of Pooling with Fixed Staffing Levels

Abandonment 1

λ1	λ2	0	5	10	15	20	25	30	35
180	180	7.9%	6.4%	5.8%	5.5%	5.4%	5.4%	5.3%	5.2%
180	200	7.9%	8.1%	8.3%	8.3%	8.4%	8.4%	8.5%	8.5%
180	220	7.9%	9.6%	10.7%	11.3%	11.7%	11.9%	12.0%	12.2%
200	200	13.6%	12.6%	12.1%	11.9%	11.8%	11.6%	11.6%	11.6%
200	220	13.6%	14.3%	14.7%	15.0%	15.1%	15.2%	15.2%	15.3%
220	220	19.7%	18.9%	18.6%	18.4%	18.2%	18.2%	18.2%	18.2%

∆ Abandonment Pool 1

λ1	λ2	5	10	15	20	25	30	35
180	180	-1.5% *	-0.6% *	-0.2% *	-0.2% *	0.0%	-0.1% *	-0.1% *
180	200	0.2% *	0.2% *	0.1% *	0.0%	0.1% *	0.1% *	0.0%
180	220	1.7% *	1.1% *	0.6% *	0.4% *	0.2% *	0.1% *	0.2% *
200	200	-1.1% *	-0.5% *	-0.2% *	-0.1% *	-0.1% *	0.0%	0.0%
200	220	0.6% *	0.4% *	0.3% *	0.1% *	0.1% *	0.0%	0.1% *
220	220	-0.8% *	-0.3% *	-0.2% *	-0.1% *	0.0%	-0.1% *	0.1% *

* indicates statistical significance at the .9 level

Table 6-4 Impact on Low Vol. Project Abandonment of Pooling with Fixed Staffing Levels

Abandonment 2

λ1	λ2	0	5	10	15	20	25	30	35
180	180	7.2%	6.2%	5.8%	5.5%	5.5%	5.4%	5.3%	5.3%
180	200	13.1%	10.7%	9.5%	8.9%	8.6%	8.4%	8.3%	8.2%
180	220	19.2%	15.2%	13.2%	12.2%	11.6%	11.3%	11.1%	11.0%
200	200	13.1%	12.5%	12.1%	11.9%	11.8%	11.7%	11.7%	11.7%
200	220	19.2%	17.0%	15.9%	15.3%	15.1%	14.8%	14.7%	14.7%
220	220	19.2%	18.8%	18.6%	18.6%	18.5%	18.4%	18.4%	18.4%

∆ Abandonment Pool 2

λ1	λ2	5	10	15	20	25	30	35
180	180	-1.0% *	-0.4% *	-0.2% *	-0.1% *	-0.1% *	-0.1% *	0.0%
180	200	-2.5% *	-1.2% *	-0.5% *	-0.4% *	-0.2% *	-0.1% *	-0.1% *
180	220	-4.0% *	-2.0% *	-1.0% *	-0.6% *	-0.3% *	-0.1% *	-0.2% *
200	200	-0.7% *	-0.3% *	-0.2% *	-0.1% *	0.0%	-0.1% *	0.0% *
200	220	-2.2% *	-1.1% *	-0.6% *	-0.3% *	-0.2% *	-0.1% *	-0.1% *
220	220	-0.4% *	-0.2% *	-0.1%	0.0%	-0.1% *	0.0%	0.0%

* indicates statistical significance at the .9 level

Table 6-5 Impact on Low Vol. Project Abandonment of Pooling with Fixed Staffing Levels

In the first set of three tables I examine the impact on the combined TSF, and the TSF of each individual project. We see that the overall TSF is always improved by pooling, and the degree of improvement is based on the amount of spare capacity in the system. When both projects are below plan the overall TSF is improved by 2.8% with just five agents. If both projects have above plan volume, the TSF also improves by 2.8%. The biggest gain comes when the projects have differential rates; when one project is low and the other high we get a 5.2% gain in overall TSF. The improvement quickly drops off with the number of agents cross trained; the most benefit comes from the first few agents. Cross training beyond 15 agents yields results that are not meaningful, and in many cases are not statistically different from zero.

The results are even more interesting when we examine the data at the individual project level. When each project has a similar arrival rate the benefits are distributed evenly. But it is when the arrival rates are different that the maximum gain occurs; and that gain accrues disproportionately to the under staffed project. When volumes are at opposite extremes, the understaffed project receives a benefit of an 11% boost in TSF from only 5 cross trained agents. Cross training of 10 agents increase TSF by another 10 points raising TSF to nearly 80%. In the case of significant mismatch the over staffed project may suffer degradation in performance, but this decline is significantly smaller then the boost to the other project and aggregate TSF always increases. The most significant case is when volumes have a maximum mismatch and the overstaffed project's TSF declines by 2.2% with 5 agents cross trained. Note however that this project had a baseline TSF of 86%, well over the standard target of 80%. This result does however raise a caution for pooling projects with very high (90%) TSF targets. In the case of a smaller mismatch the degradation was very moderate, about 0.9% with 10 agents cross trained, where the busy project may see an improvement on the order of four points from only 5 cross trained agents.

Tables 6-4 and 6-5 shows a similar analysis for the abandonment rate for each project. We see similar results; pooling reduces the maximum wait time callers face, and therefore reduces the proportion of callers kept on hold past their patience level. The improvement is the most significant when a capacity mismatch occurs.

Overall this analysis shows that partial pooling yields substantial benefits in steady state. The improvement is the greatest when a capacity mismatch occurs and the under capacity project receives the greater benefit. In the next section we examine how arrival rate uncertainty impacts the pooling analysis.

6.2.4 Steady State but Uncertain Arrival Rate

In this analysis I continue to examine the impact of pooling when projects have a constant rate, but I now allow for uncertainty in the arrival rate. Specifically I assume that the calls in each pool will arrive with a constant rate, but the realized rate is a random variable. Assume that the arrival rates are independent and identically distributed normal random variables with mean 200 and standard deviation 20. I examine how partial pooling impacts the expected TSF and abandonment rate.

The following graphs present the results of a simulation experiment

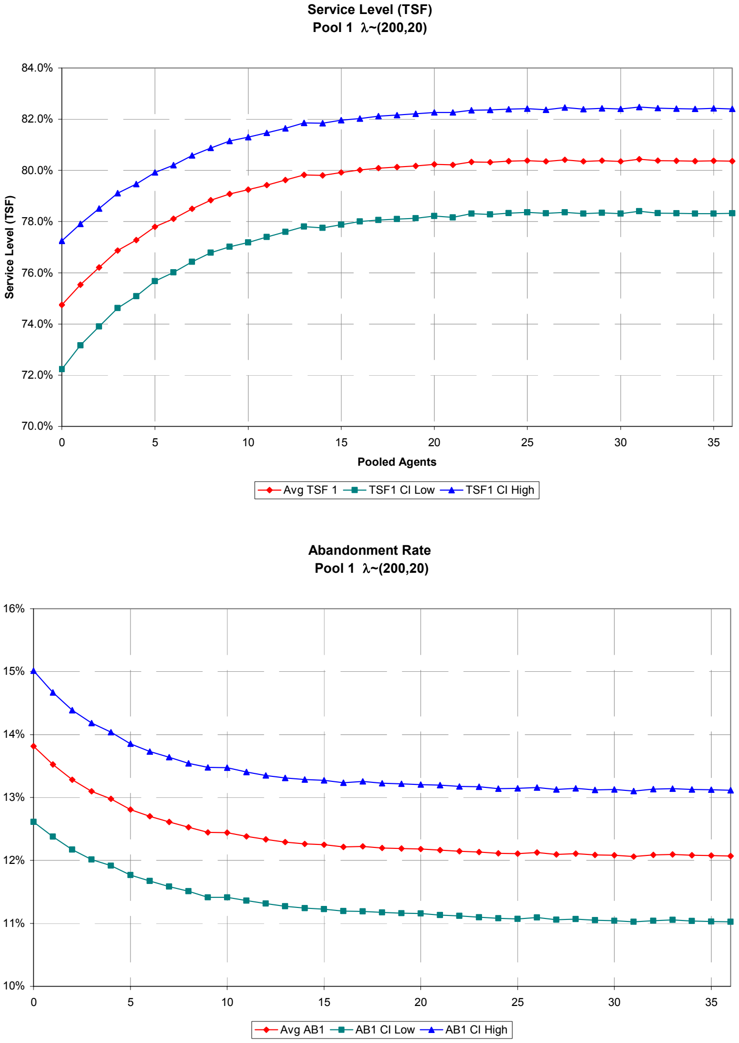


Figure 6-7 Impact of Pooling with Steady but Uncertain Arrivals

The curves in Figure 6-7 are almost identical to the plots for steady state arrivals at 200 calls/hr shown in Figure 6-1. The TSF level is slightly lower in the uncertain case; 74.2% vs. 75.4% with no pooling and 80.5% vs. 81.5% in the full pooling case. Although not a major shift, this illustrates one of the effects of arrival rate uncertainty. Because of the nature of the TSF curve the effects of volume changes is not proportional; higher volume causes a larger shift in the resulting service level then lower volume. So even if volume varies around the mean symmetrically, the resulting TSF will be lower in the uncertain case then the corresponding mean value case. An interesting phenomenon is illustrated in the following two graphs.

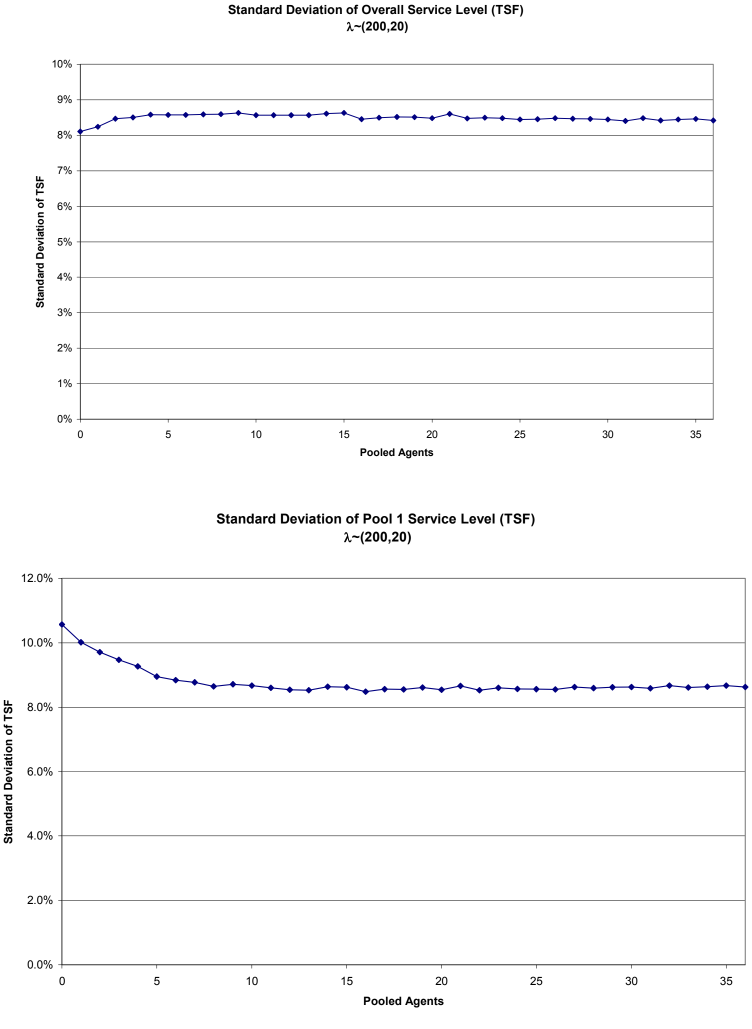


Figure 6-8 Standard Deviation of TSF with Variable Pooling

On the top we see that the standard deviation of the overall (combined) service level is essentially unaffected by pooling, remaining at a roughly constant level just over 8%. The bottom graph however reveals the standard deviation of the service level for pool one decreases as the pooling level increases, at least for the first few pooled agents. In the case of no pooling the service level in each pool is independent from the service level in the other pool. As pooling increases the service level in each pool become dependent random variables.

6.3 Optimal Cross Training in Steady State

6.3.1 Overview

In the previous section I examined the impact of varying the number of cross trained agents for a fixed pool of resources. I showed that the service level increases as agents are crossed trained, but that the incremental benefit drops off quickly. This suggests, assuming cross training is costly, that cross training more than a moderate proportion of the work force is sub optimal. In this section I examine this issue more rigorously and attempt to find the optimal level of cross training. To do this I relax the assumption of a fixed resource pool. The optimization problem then becomes selecting the staffing vector that defines the number of agents in each pool so as to minimize the expected cost of operation.

6.3.2 Operational Costs

As discussed previously, the cost of cross training is relatively high. In this model I assume that the incremental cost has two components, a training component and a wage component. The training component represents the investment in an individual agent to give her the skills necessary to handle the second project type. The wage component represents the incremental wage paid to a cross trained agent. Specifically the incremental cost of a cross trained agent is defined as

$$K = \frac{T}{\gamma} + w \tag{6.2}$$

where w is the incremental wage. T is the cost of training, and γ expected lifetime of an agent, so T/γ represents the average cost of training amortized over the agents employment life.

The second consideration is the degree of confidence sought to achieve the targeted service level. I quantify this by assigning a penalty cost proportional to the service level shortfall. If we denote the penalty rate as *r*, the goal as *g*, and the realized service level as *S*, then the penalty cost *P* can be expressed as

$$P = r(g - S)^{+} \tag{6.3}$$

The model assumes a penalty for failing to achieve the SLA, but no bonus for overachieving so the penalty cost is non-negative. Our system has three staffing levels denoted as $x_i, i = 1, 2, 3$ and the total cost of operating the system is given by

$$TC= K_1 x_1 + K_2 x_2 + r_1(g_1 - S_1)^{+} + r_2(g_2 - S_2)^{+} + w\left(x_1 + x_2 + x_3\right) \tag{6.4}$$

Our objective is to select the staffing vector that minimizes the expected cost of operating the system.

6.3.3 A Simulation Based Optimization Method

I use a simulation based local search algorithm to find the optimal cross training pattern for any given parameter setting. The local search algorithm is guided by a variable neighborhood search (VNS) metaheuristic. VNS is a metaheuristic that makes systematic changes in the neighborhood being searched as the search progresses (Hansen and Mladenovic 2001; Hansen and Mladenovic 2005). When using VNS a common approach is to define a set of nested neighborhoods, such that

$$N_1(x) \subset N_2(x) \subset \ldots \subset N_{k_{Max}}(x) \quad \forall x \in X \tag{6.5}$$

The general structure of the VNS is then as follow:

```
1. Initialization
    a. Select the set of neighborhood structures $N_k$, for
       $k = 1,\ldots,k_{\max}$
    b. Construct an initial incumbent solution, $x_I$, using
       some heuristic procedure.
    c. Select a confidence level $\alpha$ for the selection of
       a new incumbent solution
2. Search: repeat the following until Stop=True
    a. Set $k = 1$
    b. Find $n_{k_{\min}}$ candidate solutions, $x_C$ that are
       neighbors of $x_I$
    c. Simulate the system with each candidate and
       compare the results to the incumbent using a
       pairwise T Test.
```

```
d. If any $x_C$ is superior to $x_I$ at the $\alpha$ level then
   set $x_I = x_C^*$, where $x_C^*$ is the best candidate solution
   Else, set $i = n_{k_{\min}}$, set found = false, and repeat until ($i = n_{k_{\max}}$ or found=True)
      i. Find a new candidate $x_{k_i}$
     ii. Simulate the system with each candidate and compare the results using a pairwise T Test.
    iii. If $x_{k_i}$ is superior to $x_I$ at the $\alpha$ level then
         set $x_I = x_{k_i}$ and found = True
e. If a no new incumbent was found in neighborhood $k$ then
     i. set $k = k + 1$
    ii. if $k > k_{\max}$ then Stop = True
```

Figure 6-9 General VNS Search Algorithm

This algorithm searches the neighborhood of the current incumbent evaluating at least $n_{k_{\min}}$ points. If no statistically improving solution is found it continues to search until either an improving solution is found or a total of $n_{k_{\max}}$ points have been evaluated. Each time an improving solution is found the search restarts with the new incumbent. If no new incumbent is found the search continues in the next largest neighborhood. The search process continues until no improving solution is found in the largest neighborhood structure.

Two important parameters for this search process are $n_{k_{\min}}$ and $n_{k_{\max}}$, the lower and upper bounds on the number of neighbors to evaluate before moving to the next neighborhood. If the neighborhood is defined narrowly these parameters are both set equal to the total number of neighbors and the neighborhood is searched exhaustively. In larger neighborhoods an exhaustive search is not practical and solutions are selected at random. In this case $n_{k_{\min}}$ is the minimum number of neighbors to evaluate. Setting this parameter to one implements a first improving local search.

6.3.4 Optimal Cross Training with Known Arrival Rates

In the case of steady state arrivals with known rates, two different neighborhoods are defined. N_1 is the neighborhood of all *1-changes*; that is the set of all feasible solutions x_i such that one element differs from x_c by either 1 or -1. For any incumbent there are up to 6 solutions in this neighborhood. N_1 is the neighborhood of all *2-changes*; that is the set of all feasible solutions x_i such that exactly two elements differ from x_c by either 1 or -1. For any incumbent there are up to 12 solutions in this neighborhood.

In this experiment I seek to determine the optimal staffing vector for a steady state process with known arrival rates. I am interested in determining how the staffing vector is impacted by the relative arrival rates as well as management decisions related to the desired quality of service. Specifically I create a two level full factorial design in four factors as shown below.

	A	B	C	D
1	-	-	-	-
2	+	-	-	-
3	-	+	-	-
4	+	+	-	-
5	-	-	+	-
6	+	-	+	-
7	-	+	+	-
8	+	+	+	-
9	-	-	-	+
10	+	-	-	+
11	-	+	-	+
12	+	+	-	+
13	-	-	+	+
14	+	-	+	+
15	-	+	+	+
16	+	+	+	+

Variable Factor Definitions	-	+
A Arrival Rate 2	100	200
B Service Level Requirement	70/120	85/60
C Penalty Rate/hr	5	15
D Pooled wage differential	10%	40%

Constant Factors	
Arrival Rate 1	100
Talk Time (min)	12
Mean time to Abandon (sec)	350

Table 6-6 Cross Training with Steady State Known Arrivals – Experimental Design

I ran this experiment using a version of the VNS algorithm outlined in Figure 6-9. For each configuration I simulated two days of operations and performed 10 replications. The search moved to a new solution if the pairwise comparison showed an improvement at the 80% confidence level. The results of this optimization are shown in the following table.

	Factors				Staffing Vector				Metrics		
									Average	Average	Average
	A	B	C	D	N1	N2	N3	% Pooled	TSF	Total Cost	Penalty
1	-	-	-	-	17	17	2	5.6%	70.2%	17,759	383
2	+	-	-	-	17	31	4	7.7%	69.1%	25,693	541
3	-	+	-	-	21	21	3	6.7%	86.2%	21,872	128
4	+	+	-	-	21	39	4	6.3%	85.1%	31,226	314
5	-	-	+	-	17	17	3	8.1%	73.9%	17,904	0
6	+	-	+	-	17	32	4	7.5%	72.3%	25,808	176
7	-	+	+	-	21	21	3	6.7%	86.2%	22,154	410
8	+	+	+	-	21	40	4	6.2%	93.6%	32,496	1,104
9	-	-	-	+	17	17	2	5.6%	70.1%	18,120	456
10	+	-	-	+	17	32	3	5.8%	69.2%	26,082	546
11	-	+	-	+	21	21	3	6.7%	86.2%	22,297	121
12	+	+	-	+	21	40	3	4.7%	85.0%	31,718	422
13	-	-	+	+	17	17	3	8.1%	73.7%	18,336	0
14	+	-	+	+	17	32	4	7.5%	72.1%	26,338	130
15	-	+	+	+	21	21	3	6.7%	86.2%	22,584	408
16	+	+	+	+	21	40	4	6.2%	87.2%	32,025	57

Table 6-7 Cross Training with Steady State Known Arrivals – Experimental Results

This data shows that in all cases examined, partial pooling is beneficial and the optimal solution always includes some level of cross training. In this analysis the optimal number of cross trained agents covers a relatively narrow range. The optimal solution always has at least two, but no more than four cross trained agents. Cross trained agents represent between 4.7% and 8.1% of the total labor pool. The algorithm also sets staffing levels such that the service level is very close to the target level. However, because this is fundamentally a discrete optimization problem, the service level can not be set to an arbitrary level and is sometimes optimal to allow a small expected penalty cost.

6.3.5 Optimal Cross Training with Uncertain Loads

In the previous section I calculated the optimal staffing vector when arrival rates are known and constant. We found that in all cases we examined the optimal staffing choice called for some level of cross trained resources, even though those resources are more costly than base level resources. In this section I relax the assumption that arrival rates are known and examine how this impacts the optimal staffing vector.

I conducted an experiment similar to the experiment outlined in Table 6-7 with the exception that the arrivals rates are normally distributed around the original set points with a coefficient of variation of 0.1.

	Factors				Staffing Vector				Metrics		
									Average	Average	Average
	A	B	C	D	N1	N2	N3	% Pooled	TSF	Total	Penalty
1	-	-	-	-	17	17	3	8.1%	77.7%	17,904	0
2	+	-	-	-	17	32	4	7.5%	75.5%	25,632	0
3	-	+	-	-	21	21	3	6.7%	89.5%	21,744	0
4	+	+	-	-	21	40	4	6.2%	90.0%	31,392	0
5	-	-	+	-	17	17	5	12.8%	84.4%	18,960	0
6	+	-	+	-	17	32	6	10.9%	80.8%	26,688	0
7	-	+	+	-	21	20	6	12.8%	94.1%	22,848	0
8	+	+	+	-	21	39	7	10.4%	93.6%	32,496	0
9	-	-	-	+	17	17	3	8.1%	77.7%	18,336	0
10	+	-	-	+	17	33	3	5.7%	75.4%	26,016	0
11	-	+	-	+	21	21	3	6.7%	89.5%	22,176	0
12	+	+	-	+	21	40	3	4.7%	88.0%	31,308	12
13	-	-	+	+	17	17	4	10.5%	83.6%	19,488	480
14	+	-	+	+	17	33	5	9.1%	80.8%	27,360	0
15	-	+	+	+	21	21	5	10.6%	93.8%	23,520	0
16	+	+	+	+	21	40	6	9.0%	93.5%	33,312	0

Table 6-8 Cross Training with Steady State Uncertain Arrivals – Experimental Results

In the uncertain arrival case the level of cross training is in general increased, total costs in general increase, and the service level penalty is effectively eliminated. The difference between these two experiments is summarized in the following table:

	Factors				Staffing Vector				Metrics		
									Average	Average	Average
	A	B	C	D	N1	N2	N3	% Pooled	TSF	Total Cost	Penalty
1	-	-	-	-	0	0	1	2.6%	7.5%	144.6	-383.4
2	+	-	-	-	0	1	0	-0.1%	6.4%	-60.7	-540.7
3	-	+	-	-	0	0	0	0.0%	3.3%	-127.8	-127.8
4	+	+	-	-	0	1	0	-0.1%	4.9%	166.1	-313.9
5	-	-	+	-	0	0	2	4.7%	10.5%	1,056.0	0.0
6	+	-	+	-	0	0	2	3.4%	8.4%	879.5	-176.5
7	-	+	+	-	0	-1	3	6.1%	7.9%	693.8	-410.2
8	+	+	+	-	0	-1	3	4.3%	0.0%	0.0	-1,104.0
9	-	-	-	+	0	0	1	2.6%	7.7%	216.4	-455.6
10	+	-	-	+	0	1	0	-0.1%	6.2%	-66.3	-546.3
11	-	+	-	+	0	0	0	0.0%	3.3%	-120.8	-120.8
12	+	+	-	+	0	0	0	0.0%	3.0%	-410.5	-410.5
13	-	-	+	+	0	0	1	2.4%	9.9%	1,152.0	480.0
14	+	-	+	+	0	1	1	1.5%	8.7%	1,021.9	-130.1
15	-	+	+	+	0	0	2	4.0%	7.6%	936.0	-408.0
16	+	+	+	+	0	0	2	2.8%	6.3%	1,286.7	-57.3
				Average	0	0.1	1.1	2.1%	6.4%	422.9	-294.1

Table 6-9 Comparison of Known and Uncertain Arrival Experiments

There are a few key observations from this analysis:

- Uncertainty increases cost – the total cost of operation increased by an average of $422. The cost of service delivery increased significantly in the high penalty rate cases, where service level attainment is important.
- Pooling is more effective in uncertain situations – more pooling was added in the uncertain arrival cases, and the service level penalty was effectively eliminated in the uncertain case. With uncertain arrivals the probability of a capacity mismatch is higher, and therefore the benefits of dynamic capacity reallocation are higher.

6.4 Optimal Cross Training with Time Varying Arrivals

6.4.1 Overview

In the previous section I analyzed the impact of pooling on steady state stationary behavior. As described in Chapter 2, real call centers often face arrival rates that vary significantly across the course of the day and therefore must change the staff level throughout the course of the day. In the call center projects I analyzed, staffing varies from two agents overnight, to as many as 70 agents during peak hours. On a 24 hour schedule, the call center may have shifts starting during any 30 minute period. But because the vast majority of agents are scheduled to full time shifts, the call center can not vary the staff as quickly as demand varies. The call center is therefore subject to periods of tight capacity and excess capacity in any given day.

I investigate two alternative algorithms for calculating the three pool staffing plan. In the first approach (Sim-Opt) I use simulation to find the optimal staffing levels in each individual time period. This analysis generates a minimum staffing triplet in each individual time period. I then use a weighted set covering algorithm to schedule each pool independently so as to satisfy the minimum staffing requirement. The second approach (Opt-Sim) first performs a scheduling optimization on each project individually. The resulting independent schedules are then merged in a simulation based optimization effort. The two methods are outlined in more detail below, but first I review how I calculate the objective function value in a project based optimization.

6.4.1.1 Objective Function

Conceptually, the objective of this optimization problem is to find the minimal cost staffing plan that meets the service level requirement with the appropriate level of confidence. As in the models of Chapter 4, I implement the service level requirement by adding a penalty to any service level shortfall. In the math programming formulation of Chapter 4 we also added several side constraints. In particular I required that staffing was at least at a minimum level at all times (typically two agents) and that the level of staffing was such that at expected volumes we achieved some minimal service level (typically 50%). While it is possible to modify the neighborhood structure to enforce these hard constraints, a more straightforward search mechanism results if we *soften* these constraints and add them as penalty terms to the objective function.

6.4.1.2 The Simulation – Optimization Approach (Sim-Opt)

In this approach I create a firm staffing vector for each resource pool via simulation, and then use a deterministic scheduling algorithm to assign shifts. In the initial step I run the steady state simulation analysis for each time period (336 periods) to create a staffing vector triplet (b_{1i}, b_{2i}, b_{3i}), where b_{1i} represents the staffing requirement in pool 1 in period *i.*

Given these firm staffing requirements I then execute a deterministic staffing algorithm to cover these requirements. The resulting integer program is a standard weighted set covering problem which can be expressed as

$$\min \sum_{j \in J} c_j x_j \tag{6.6}$$

subject to

$$\sum_{j \in J} a_{ij} x_j \geq b_i \qquad \forall i \in I \tag{6.7}$$

Where c_j is the cost of the *j*th schedule, x_j is the number of resources assigned to the *j*th schedule, and a_{ij} is the mapping of schedules to time periods.

6.4.1.3 The Optimization-Simulation Approach (Opt-Sim)

In this approach I generate a preliminary schedule for each project independently using an optimization program and then run a local search via simulation to optimize the overall project. To implement the process I need to define an approach for generating an initial feasible solution and for selecting new candidate feasible solutions.

To develop an initial feasible solution I run the optimization program (4.1) - (4.7) from Chapter 4 for each project individually. In this instance the model is configured to generate a schedule at a lower TSF and with a minimum staffing level of one instead of two agents. This procedure creates a staff plan that is slightly understaffed. The objective is to create an initial plan where selective cross training can create rapid improvement.

To identify additional candidate solutions I implement a VNS as described in Figure 6-7. However, in this case the neighborhood structure is considerably more complex. I define a nested neighborhood structure with five individual neighborhoods.

Let J be the set of schedules to which an agent may be assigned and denote as x_j the number of agents assigned to schedule j. A staff plan is a vector of x_j values. A staff plan is feasible if every x_j is non-negative and integral valued. Assume that any complicating constraints, such as minimum staffing levels, have been moved into the objective function as a penalty term. Denote the set of feasible staff plans as X. Furthermore, define the sets $A_i \subseteq J$ as the active schedules, for resource pool *i*; that is the schedules to which at least one resource has been assigned and let $A = A_1 \cup A_2 \cup A_3$ be the set of active schedules across pools.

Now, for some arbitrary $x \in X$, define a series of K_{Max} nested neighborhood structures such that

$$N_1(x) \subset N_2(x) \subset ... \subset N_{k_{Max}}(x) \quad \forall x \in X \tag{6.8}$$

I define the following neighborhoods

- $N_1(x)$: **Active 1 Change**: the set of all staff plans where an active assignment is updated by and additive offset, $\delta_i \in \{-1,1\}$.
- $N_2(x)$: **Active 2 Change**: pick any two feasible schedules in A_i and independently update each by $\delta_i \in \{-1,0,1\}$.
- $N_3(x)$: **Feasible 1 Change**: the set of all staff plans where a feasible assignment is updated by $\delta_i \in \{-1,1\}$.
- $N_4(x)$: **Feasible 2 Change**: pick any two feasible schedules in J and independently update each by $\delta_i \in \{-1,0,1\}$.
- $N_5(x)$: **Feasible 3 Change**: pick any three feasible schedules in J and independently update each by $\delta_i \in \{-1,0,1\}$.

Figure 6-10 Project Based Neighborhood Structure

I each neighborhood a new schedules is selected randomly and a large number of alternative schedules are evaluated at each iteration of the algorithm. While a pure random search will likely find improving solutions if enough permutations are evaluated I have found that using certain heuristic methods in each neighborhood improves the rate of convergence. In this modified approach each time a new neighbor is required the algorithm picks either a heuristic or a pure random permutation.

The following table summarizes the heuristics utilized in each neighborhood:

Neighborhood	Heuristics
$N_1(x)$: **Active 1 Change**	− Pool Support: select an active schedule in Pool 1 or Pool 2 and staff an agent to the same schedule in the cross trained pool.
$N_2(x)$: **Active 2 Change**	− Cross Train: select an active schedule in Pool 1 or Pool 2 and change the agent's designation to a cross trained agent. − Untrain: select a staffed schedule in pool three and change the designation to either 1 or 2.

$N_3(x)$: **Feasible 1 Change**	− Add Max Cover: find the set of feasible schedules that covers the most short-staffed periods and schedule an agent to one of those schedules.
$N_4(x)$: **Feasible 2 Change**	− Active Time Shift: select an active schedule and shift the assignment forward or backward by one time period.
$N_5(x)$: **Feasible 3 Change**	− Two for One: pick a schedule in Pool 1 or 2, then find the closest active matching schedule in the other pool, decrement each of these assignments and staff a super agent.

Table 6-10 Neighborhood Search Heuristics

The logic behind this neighborhood structure is relatively straightforward if we recall that we start with a near optimal solution generated from an optimization program designed to slightly under staff the projects. First of all, the set of schedules selected in the optimization process will closely match the time profile of demand. The set of active schedules will typically be a small subset of the total schedules. Therefore it is reasonable to search these Active schedules first. Since the initial schedule is understaffed by design it is reasonable that additional staffing, particularly in the super agent pool, will decrease penalty costs more than the associated labor costs so it is reasonable to focus the search efforts here. Neighborhood 1 is small enough that I can search it exhaustively. In neighborhood 2 I test the benefits of changing agent's skill designations. By testing both training and untraining I make sure that the incremental cost of training is justified.

When no improvements can be found in the set of active schedules the search is expanded to the full set of feasible schedules. The heuristic in neighborhood 3 is designed to address the short staffing penalty found by not having at least 2 agents available for each project in each time period. This heuristic is designed to test all of the schedules with the max cover and will often select a super agent as these agents provide cover for both projects. In neighborhood 4 I allow for 2 changes in the feasible schedule and specifically test for the impact of shifting a schedule forward or backward by 1 time period to potentially better cover a service level gap. The logic of the neighborhood 5 schedule is based on the notion that if we have agents in each pool on the same schedule it might be beneficial to replace both of them with a single cross trained agent. This is beneficial when the service level is being met with high probability, and the penalty is low. Making a two for one swap reduces labor cost and may not have a major impact on service level penalties.

In practice the largest number of improving solutions were found in neighborhood 1. Improving solutions were found in every neighborhood, though not for every optimization. In a typical optimization process improvements are found in three to four neighborhoods, though in some cases all neighborhoods generated improvements. The number of solutions tested in each iteration clearly varies based on where an improvement is found. By design most improvements are found in the first neighborhood. In my experiment I required that at least 20 candidates were tested before the best was selected. The max number varies with the number of active schedules, as neighborhood 1 is searched exhaustively. Ina typical scenario bout 300 candidate solutions were tested in the final iteration of the algorithm, the iteration which found no improvements.

The total number of iterations until termination is also random, and depends on the number of feasible schedules. The total number of iterations tended to vary between 15 and 25. All in all this implies that an optimization effort will evaluate somewhere in the range of 500 to 1,500 different schedule combinations. It was based on this need to evaluate a large number of schedules that I made the decision to code a simulation model in VB, rather than use the previously developed Automod code.

In terms of the selection of the metaheuristic, there are a very large number of algorithms available including genetic algorithms, simulated annealing, and Tabu search as well as other approaches such as gradient based search or response surface methods. Because the problem is discrete I decided not to pursue gradient or response surface methods as these algorithms are better suited to smooth response functions. My choice of metaheuristic was driven by the combinatorial nature of the problem. Technically the feasible set for the problem is unlimited. Assume we place a practical limit of η as the total number of agents assigned to any schedule, the number of feasible staff plans is $3\mathrm{N}^{\eta}$ where N is the number of feasible schedules for the scheduling option. (see Table 4-10). The least flexible option (A) has 336 feasible schedules. If we set η as 10 then there are approximately 10^{30} feasible schedules. For option F the number expands to more than 10^{40}. I sought some algorithm that allowed other search heuristic (such as those in Table 10-10) to be embedded to

I rejected genertic algorithms because there was no obvious way to implement a crossover mechanism that would yield high quality solutions. In addition a population based approach increases the number of solutions to be tested, and the simulation process makes evaluation relatively expensive. The selection process is also more difficult when trying to select the best solution for a population vs. a sequential pairwise comparison. Tabu Search is a viable approach and could in fact be added to the current algorithm to prevent repeated evaluation of the same solution which clearly happens in this algorithm. Simulated Annealing is another alternative to facilitate the breakout from local optimum which is accomplished via expanded neighborhoods in this algorithm. My overall objective was to find a relatively easy to implement algorithm that would lead to good solutions because my objective was to determine if pooling is beneficial. Having shown that it is I may investigate algorithmic efficiency in future work.

6.4.2 Comparison of SimOpt and OptSim

In this experiment I use each method described in the previous section to find the optimal staffing plan for a pair of projects. In this analysis I examine pooling of the test Project J and O described in section 2.9 and analyzed in Chapter 3. As was shown in Chapter 4 the cost of service delivery, and the quality of the solution algorithm depends on the flexibility of the workforce. For this reason I test each approach for the five schedule options described in Chapter 4.

6.4.2.1 SimOpt Approach

To test the SIMOpt approach I ran the simulation search algorithm to find the optimal staffing triplet for each of the 336 individual time periods for a Project J and S pairing. The resulting staff plan is summarized in the following graphic:

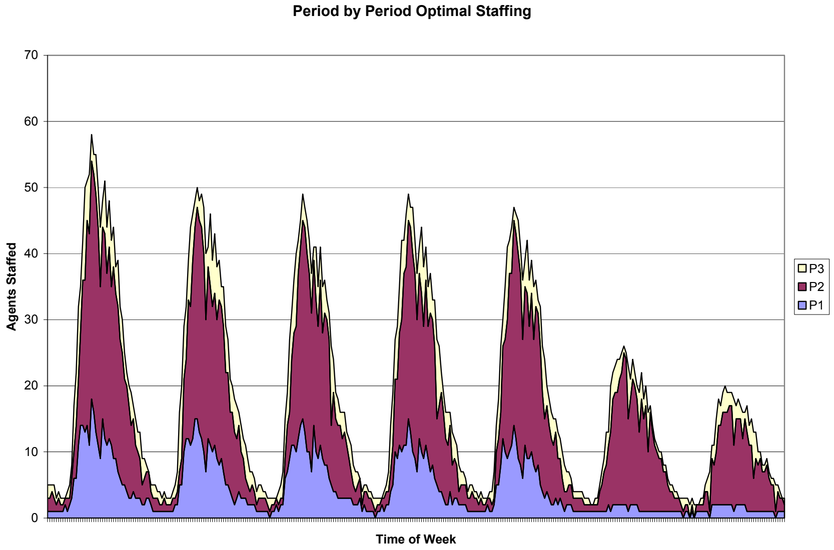


Figure 6-11 Period by Period Optimal Staffing

Visually we can see that optimizing period by period creates a highly variable staff plan. This is better illustrated in the following graph that shows only the staff plan for cross trained agents.

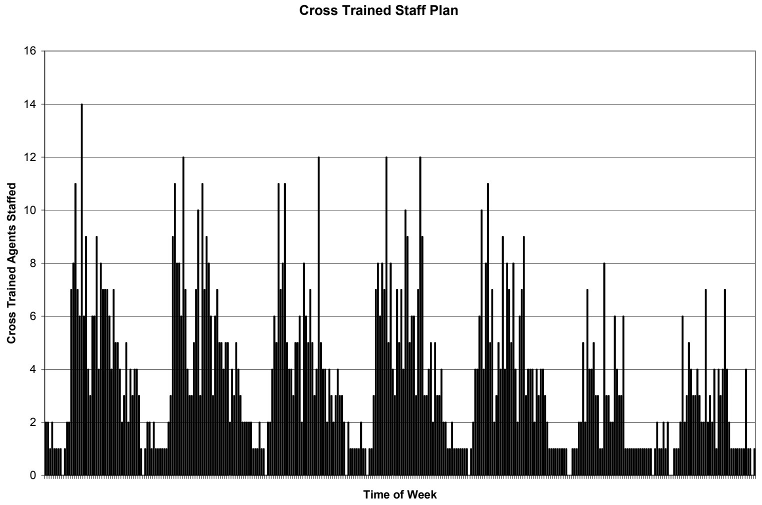


Figure 6-12 Period by Period Cross Training Requirements

The highly variable nature of this staffing plan is problematic, as the fixed staff covering model is likely to introduce a significant amount of slack into the resulting staff plan. To test this I run a basic weighted set covering model, scheduling each pool independently. I repeat this process for each schedule option with the results summarized in the following table.

	No Pooling - SCCS Optimization			SimOpt Approach					
	Direct Labor J	Direct Labor S	Total Labor	Pool 1 Labor	Pool 2 Labor	Pool 3 Labor	Total Labor	DW Loss	% DW Loss
Sched A	11,280	30,960	42,240	10,000	24,000	14,500	48,500	16,320	50.7%
Sched B	10,800	30,320	41,120	9,200	21,600	13,000	43,800	11,620	36.1%
Sched C	10,944	30,384	41,328	8,720	21,120	12,800	42,640	10,460	32.5%
Sched D	10,844	30,092	40,936	8,400	20,120	11,800	40,320	8,140	25.3%
Sched E	10,720	30,096	40,816	8,080	19,820	10,725	38,625	6,445	20.0%

Table 6-11 SimOpt Outcomes Projects J and S

This table lists the direct labor costs for the independent optimization from Chapter 4 along with the labor costs from the SimOpt approach. The simulation based optimization calculated a staffing model that costs $32,180; the difference between this figure and the total labor figure calculated is the result of excess staffing due to shift constraints; the deadweight loss. We can see that staffing constraints add substantial cost to the resulting schedule, as much as 50% in the low flexibility case and 20% in the most flexible case. Based on these results the SimOpt approach does not appear to be very promising, and I turn now to the alternative OptSim approach.

6.4.2.2 OptSim Approach

In the OptSim approach I begin with a baseline plan generated from the optimization model of Chapter 4. Specifically I generated a schedule with a lower service level goal and a single agent minimum staffing requirement then apply the search process outlined in Figure 6-9 and Table 6-6. The idea is to create a staffing plan that is slightly under staffed. In this situation single agent changes are likely to improve the staffing plan and since improving single staff changes are easier to find than improving dual changes, the algorithm should find improvements quickly.

As an initial test I ran this analysis for the pairing of projects J and S, testing each schedule option. The results are summarized below.

	No Pooling - SCCS Optimization			SimOpt Approach				
	Direct	Direct	Total	Pool 1	Pool 2	Pool 3	Total	%
	Labor J	Labor S	Labor	Labor	Labor	Labor	Labor	Savings
Sched A	11,280	30,960	42,240	9,600	25,200	6,500	41,300	2.2%
Sched B	10,800	30,320	41,120	9,200	24,000	7,500	40,700	1.0%
Sched C	10,944	30,384	41,328	9,280	23,280	7,800	40,360	2.3%
Sched D	10,844	30,092	40,936	8,920	23,860	7,375	40,155	1.9%
Sched E	10,720	30,096	40,816	8,880	23,000	8,450	40,330	1.2%

Table 6-12 OptSim Outcomes Projects J and S

This approach appears far more promising than the SimOpt approach outlined above. In each case the algorithm finds a staffing plan with a total lower cost, in spite of the higher cost of cross trained agents. Since the optimization effort here is global, the algorithm does not suffer from the deadweight loss issue faced by the OptSim approach. Based on these preliminary results I went forward with a detailed analysis of the SimOpt approach.

6.4.3 Detailed Evaluation of OptSim

6.4.3.1 Overview

In the previous section I investigated two alternative mechanisms for scheduling a pooled project pair and found that the OptSim approach appear to provide much better results. In this section I examine in more detail the resulting savings. I also examine how project characteristics impact the pooling decision. In the previous section the preliminary screen looked only at the direct labor component, but a more complete analysis obviously requires the evaluation of total cost. Also, in order to make a fair apples-to-apples comparison we must evaluate single and pooled schedules optimized using the same approach. For that reason I compared pooled project results to those found from the simulation based optimization fine tuning developed in Section 4.8.

6.4.3.2 Pooled Optimization – Project J and S

In this section I test the impact of pooling Project's J and S. Recall that Project J is a corporate project with relatively stable arrival patterns. Project S is a retail project with somewhat volatile arrival patterns. Since one project is corporate and one is retail these projects have different seasonality patterns. The busy period for project S extends later into the day, and the project has busier weekends. Project S also has less of a lunchtime lull in call volume than Project J.

The following table summarizes the results of the pooled optimization effort:

	Individual Optimization				Pooled Optimization					Comparison		
Sched Set	Labor Cost	Expected Outcome	TSF 1	TSF2	% Agents Pooled	Labor	Outcome	TSF 1	TSF2	Labor Savings	Total Savings	% Savings
A	41,600	44,504	78.3%	83.5%	13.0%	41,356	42,560	83.2%	83.4%	244	1,944	4.4%
B	40,400	44,504	78.1%	84.7%	15.3%	40,769	41,873	84.4%	83.6%	-369	2,631	5.9%
C	40,320	44,504	78.9%	85.0%	16.1%	40,424	41,171	83.0%	84.0%	-104	3,333	7.5%
D	40,120	44,504	79.4%	84.4%	17.0%	40,732	41,537	83.0%	84.3%	-612	2,968	6.7%
E	40,000	44,504	78.9%	85.3%	18.7%	40,197	41,664	81.4%	83.4%	-197	2,840	6.4%

Table 6-13 Pooled Optimization – Projects J-S

The data shows that even with a 25% premium for pooled agents, pooling reduces the overall cost of operation. Cost savings vary from 4.4% to 7.5% depending on the scheduling set option. In each case the number of labor hours drawn from the cross trained pool is less than 20%. As was the case in the steady state analysis, pooling a relatively small percentage of the agents provides the optimal results. Note that Project J, the smaller project, sees an improvement in service level in each case while the service level for Project S remains constant or declines slightly. Intuitively, in the single pool case Project S must carry safety capacity to hedge against costly spikes, which is evident by the average service level cushion or 3%-5%. In the pooled case spare capacity can be allocated to Project J as necessary and each project has an average service level just above the targeted level. Further insight can be gleaned from the graphical vies of the resulting staff plan. In the following figure I plot the staffing plan for schedule set C.

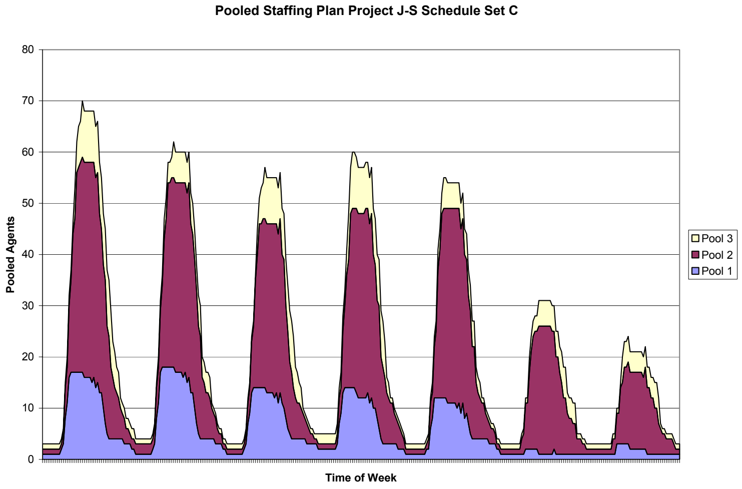


Figure 6-13 Pooled Staffing Plan

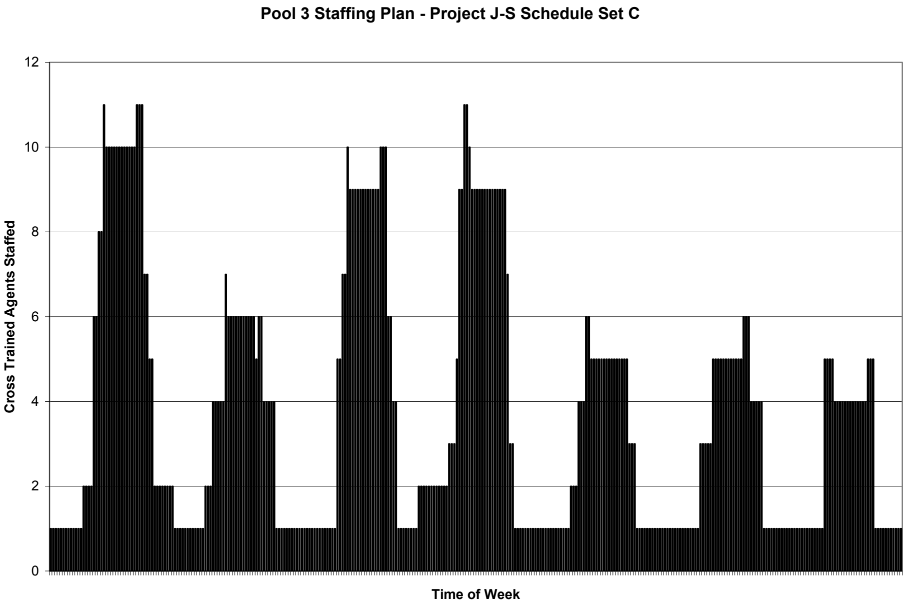


Figure 6-14 Cross Trained Agent Staffing Plan

Cross trained agents are scheduled throughout the week but are most heavily deployed during the busy periods.

6.4.3.3 Pooled Optimization Projects J-O

Similar results are found for the pairing of Project J and Project O as summarized below.

	Individual Optimization				Pooled Optimization					Comparison		
Sched Set	Labor Cost	Expected Outcome	TSF 1	TSF2	% Agents Pooled	Labor	Outcome	TSF 1	TSF2	Labor Savings	Total Savings	% Savings
A	23,200	24,606	78.3%	79.9%	14.3%	23,228	23,938	80.8%	81.2%	-28	668	2.7%
B	22,800	24,606	78.1%	78.5%	14.5%	22,834	23,547	81.7%	81.4%	-34	1,060	4.3%
C	22,800	24,606	78.9%	78.3%	21.2%	23,115	23,504	81.8%	82.3%	-315	1,102	4.5%
D	22,540	24,606	79.4%	79.7%	19.0%	23,143	23,758	80.7%	82.8%	-603	848	3.4%
E	22,460	24,606	78.9%	79.1%	18.8%	22,698	23,550	80.8%	81.5%	-238	1,056	4.3%

Table 6-14 Pooled Optimization – Projects J-O

In this case the savings are slightly less, in the range of 2.7% - 4.3% and the proportion of agents cost trained is slightly higher. In each case labor costs are increased slightly resulting in a higher level of confidence that the service level goal will be achieved. The average service level of each project improves in each case. Recalling that these projects are of approximately the same size the benefits are roughly equally distributed. The average service level for each project moves up from just below the target to just above the target. Intuitively, since the incremental capacity can be allocated to rather project as needed, the cost of incremental labor is offset by the reduction in penalty costs.

6.4.3.4 Pooled Optimization Projects S-O

In this final pairing I examine a pooling of Project S and Project O, both of which have retail oriented seasonality patterns. The results are summarized below:

	Individual Optimization				Pooled Optimization					Comparison		
Sched	Labor	Expected			% Agents					Labor	Total	%
Set	Cost	Outcome	TSF 1	TSF2	Pooled	Labor	Outcome	TSF 1	TSF2	Savings	Savings	Savings
A	41,600	44,387	83.5%	79.9%	10.1%	40,654	42,349	82.4%	80.4%	946	2,038	4.6%
B	40,800	44,387	84.7%	78.5%	13.7%	39,370	41,523	81.2%	80.6%	1,430	2,864	6.5%
C	40,400	44,387	85.0%	78.3%	15.4%	40,034	41,966	82.8%	80.3%	366	2,421	5.5%
D	40,540	44,387	84.4%	79.7%	14.5%	39,768	42,103	82.8%	79.8%	772	2,284	5.1%
E	40,620	44,387	85.3%	79.1%	13.7%	40,273	42,188	82.5%	80.7%	347	2,199	5.0%

Table 6-15 Pooled Optimization – Projects S-O

As in the previous case pooling reduces cost of operation for these projects around 5% by pooling 10%-15% of agents. But unlike the two previous cases, this situation reduces total cost by reducing labor. The intuition is that each of these projects is relatively volatile and must carry significant spare capacity to hedge against uncertainty. By pooling, project spare capacity can be shared and the total amount of spare capacity is reduced.

6.4.3.5 The Impact of Cross Training Wage Differential

The analysis shows that cross training a portion of the workforce can reduce costs even if cross training resources is expensive. In the analysis so far we have assumed that cross training creates a 25% cost premium. In this section I examine the impact of varying the wage differential.

For this experiment I test the same project and schedule pairs tested above, but allow the wage differential to vary. I maintain the base agent wage at $10.00 per hour, but I test super agent wage rates of $11.25, $12.00, and $13.75. Overall I find that cross training is a viable tactic over this range of costs. The expected savings is naturally declining in the wage differential as is the proportion of agents cross trained – although the proportion of agents cross trained is less sensitive to the wage differential than one might expect. The results are summarized in the following table

		No Cross	Cross Training Wage Differential					
		Training	$11.25		$12.50		$13.75	
	Sched	Expected	% Agents	%	% Agents	%	% Agents	%
Pairing	Set	Outcome	Pooled	Savings	Pooled	Savings	Pooled	Savings
J-S	A	44,504	15.3%	7.1%	13.0%	4.4%	14.3%	3.9%
	B	43,529	17.3%	5.7%	15.3%	3.8%	13.3%	3.7%
	C	43,780	15.9%	6.9%	16.1%	6.0%	15.1%	4.0%
	D	43,120	19.0%	5.4%	17.0%	3.7%	16.4%	2.6%
	E	43,240	19.4%	5.5%	18.7%	3.6%	17.4%	0.9%
J-O	A	24,606	14.3%	4.1%	14.3%	2.7%	10.7%	0.9%
	B	24,643	19.6%	5.5%	14.5%	4.4%	16.1%	1.5%
	C	24,597	22.9%	5.8%	21.2%	4.4%	15.4%	2.5%
	D	24,396	28.3%	5.4%	19.0%	2.6%	14.9%	0.9%
	E	24,513	20.1%	6.3%	18.8%	3.9%	18.3%	0.6%
S-O	A	44,387	9.1%	6.3%	10.1%	4.6%	6.1%	5.2%
	B	44,424	18.2%	5.9%	13.7%	6.5%	14.4%	3.3%
	C	44,378	15.9%	7.4%	15.4%	5.4%	13.9%	3.4%
	D	44,177	16.5%	6.1%	14.5%	4.7%	13.0%	3.3%
	E	44,294	17.5%	5.6%	13.7%	4.8%	16.7%	1.9%

Table 6-16 - The Impact of Wage Premiums on Cross Training Results

6.4.3.6 Conclusions

Evaluation of these three project pairings shows that the ability to reduce operating costs by partial pooling is robust across different project combinations. The overall results in terms of savings of around 5% with a pooling of around 15% of agents are consistent across pairings. The mechanism in which the savings are obtained is however different. In some cases the aggregate service level is increased when adding more (pooled) agents allows efficient improvement in service level goal attainment. In other cases pooling allows redundant capacity to be reduced through efficient sharing of spare capacity.

6.5 Summary and Conclusions

6.5.1 Summary

In this model I examine the concept of partial pooling of agents in call centers. The basic premise is that in cases where training is expensive, it is not practical to train all agents to handle multiple call types. I investigate the option of training some agents to handle multiple call types and show that this approach can yield substantial benefits.

I first analyzed steady state performance, independent of costs, and showed that that pooling yields significant, but rapidly declining benefits. I then investigated the optimal level of cross training in a steady state environment when cost training is costly. I develop a straightforward and efficient simulation based optimization method for finding the optimal level of cross training. Finally I extend this method to a project oriented setting, where arrivals are nonstationary with day of week and time of day seasonality.

6.5.2 Contributions

This model makes a contribution by evaluating a pooling approach not previously analyzed. A model very similar in concept to mine is (Wallace and Whitt 2005). (I refer to this paper as W&W) In the W&W model there are 6 call types and every agent is trained to handle a fixed number of those types. The authors use a simulation based optimization model to find the ideal cross training level. The paper's key insight is that a low level of cross training provides "most" of the benefit. Specifically, they find that training every agent in 2 skills provides the bulk of the benefit, while additional training has a relatively low payoff. Although the general finding in our paper is similar, e.g. small levels of cross training give the majority of the benefit, the models are very different. While their best solution has every agent cross trained in 2 skills, our model assumes that only a small proportion of agents are cross trained. In our scenario cross training is very expensive and 100% cross training is not practical. W&W show that adding a second skill gives most of the value, but they don't analyze the cost associated with cross training. In our model we include the cost of cross training and seek an optimal level. Additionally, W&W examine cross training only in steady state, where arrival rates and staff levels are fixed. Our analysis focuses on the case where both arrival rates and staff levels change dramatically during the course of the SLA period. We are very interested in how the variable fit of capacity to load impacts the benefit of partial pooling. At a detailed level the W&W model ignores abandonment - an important consideration in our situation. The model presented here moves beyond the W&W model to examine the case where cross training is expensive and service levels are important. This model also allows for abandonment.

6.5.3 Management Implications

The clear implication for managers from this analysis is that cross training a limited number of agents is a cost effective option under a wide range of assumptions and conditions. The model presented here provides a specific methodology for finding the appropriate level of cross training, but also provides some basic insight. Managers should seek to cross train a moderate level of the agent base to support multiple call streams. In the case of multilingual call centers, managers need a few multilingual agents, but don't need all agents to be multilingual.

6.5.4 Future Research

In this initial analysis I looked at a relatively simple case of two call types and three agent types. An natural extension is to examine a larger number of call type and additional pooling types. Consider the scenario depicted in the following graphic

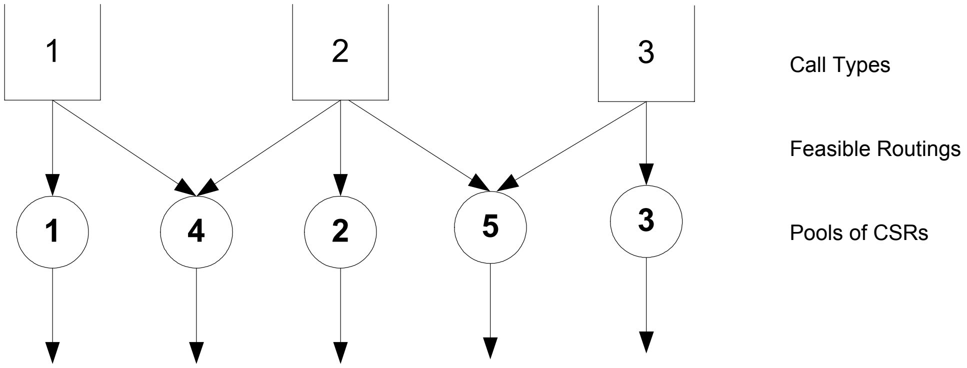


Figure 6-15 Three Project Pooling Approach

In this case three projects are pooled together. The capacity for project 2 is now even more flexible as agents can be pushed out or pulled in from two other projects. This configuration might be especially appropriate if Project 2 is highly volatile. This configuration could be further modified if a 6th type of agent was added cross trained in Project's 1 and 2. This would allow capacity to adjust around the circle in a fashion similar to bucket brigades in manufacturing work cells.

An alternative approach would be to create super-super agents, cross trained in three (or more) projects. This type of configuration already exists in multilingual support. In some of the projects I examined in Europe support must be provided in 10 or more languages and many agents speak three or more languages. All of these opportunities present significant opportunities for the author to continue this research effort.

7 Summary and Conclusions

7.1 Summary

The focus of this dissertation has been on the impact of uncertainty on call center operations, a type of service supply chain. The work is based on a project with an outsourced provider of call center based support services. From a business perspective the specific problems addressed in this dissertation include:

- How can we improve our staff scheduling process to lower labor costs while still making the monthly service level?
- How can we improve the staffing process for new launches to avoid the common problem of very poor service levels during start-up?
- What innovations are possible in project staffing to allow increased efficiency without sacrificing customer service levels?

In the course of my work with this company we analyzed call arrival patterns in an effort to characterize the arrival process and better understand uncertainty. We found that managers often mischaracterized variability and struggled to separate stochastic variability, seasonality, and trends. Overall we found that after controlling for stochastic variability and weekly trends the data tends to be very stable.

We also found that managers either did not consider or mischaracterized other key metrics. Managers intuitively understood that agents learn over time but had never conducted any formal analysis of the learning process and never explicitly factored learning into their models. Managers also believed that attrition rates were driven by burnout and that the longer an individual stayed the more likely they were to quit. Our analysis showed that the probability to quit decreased significantly with tenure and that bad hiring was a bigger issue than burnout.

We also analyzed scheduling and staffing processes and methods. We found that while managers had access to relatively sophisticated scheduling tools, they often relied on manual processes with minimal automated support; for example, Erlang C calculators to identify basic requirements and manual scheduling via Excel.

The work in this dissertation sought to build on this analysis and develop decision support models to address various capacity management decisions. The objective is to develop specific model that could in the future be converted into operational systems, but also to develop insights into the capacity management problem.

7.2 Model Validation

The analysis in this dissertation was validated throughout my work with the subject firm. All of the statistical analysis in Chapter 2 was reviewed with managers at various levels in the organization, throughout the analysis process. The optimization models were run in various early forms and the results were shared with managers. All of the results were considered valid.

Much of the work done to develop this dissertation was taking the project specific analysis done during the study, generalizing it, and studying it under various situations. While none of the specific models developed in this dissertation are going to be adopted as operational software anytime soon, much of the *insight* developed during this analysis is being adopted. For example:

- Senior management has begun restructuring the management structure to better leverage cross project synergies. Agents are being cross trained on several projects.
- Managers have utilized a limited number of part time resources to better match the seasonality of demand.
- The project launch process is being updated to account for attrition and learning by *over-engineering the launch*; i.e. hiring agents above and beyond the steady state staffing projections.
- Hiring practices are being revaluated to address the high failure rate of new agents. Given a new awareness of the cost of hiring and training agents turnover rates are being given more senior management attention.

7.3 Key Insights

Abstracting away from the specific issues at this firm, some general insights that emerge from this analysis include the following:

- **Variability**: call arrival patterns are much more volatile than the models in the literature assume. Explicitly considering this variability has a material impact.
- **Seasonality**: the seasonality of call patterns on many call types is very significant and significantly effects cost of service delivery.
- **Individual Learning**: the rate of learning by individual agents in this application is substantial and has very significant impacts on operations.
- **A Little Flexibility Goes a Long Way**: throughout this analysis I find that limited flexibility can lead to significant improvement. A few part time agents achieve most of the benefit of universal part timers. Similarly, cross training a few agents give most of the benefit of cross training all of them.

7.4 Contributions

This dissertation makes several important contributions to the applied Operations Research literature, including the following:

- **Integration of Server Sizing and Staff Scheduling**: my scheduling model combines these two steps into a single optimization model in contrast with the existing literature that treats these as separate processes. I show that the resulting schedule is lower cost.
- **Stochastic Call Center Scheduling Model**: my scheduling model is, as far as I know, the first application of stochastic programming to the staff scheduling problem. I show that considering variability lowers costs.
- **Partial Pooling Model**: the cross training model I introduce is very practical and unexplored in the literature.
- **Non-stationary Problem**: my analysis is one of the few that considers staffing a call center with non-stationary arrival rates and is perhaps the first to assess a realistic arrival pattern.

7.5 Future Research

The research in this dissertation can of course be extended and expanded. I address specific extensions of each model in the associated chapter. Some of the key areas for potential future research include the following:

- **Agent Heterogeneity**: like most of the models in the literature most of the analysis in this dissertation assumes agents are statistically identical. (I break from this slightly in the hiring model.) The data indicates that agent productivity is highly variable, an area that has received very little attention in the OR literature.
- **More Complex Workflows:** in this dissertation I consider very basic workflows, even in the cross training model. In practice workflows are often very complex dues to skills based routing, escalation, etc. These models could be extended to address more complex call routing.
- **Multilingual Call Centers**: I allude to this several time in this dissertation but the multilingual capacity management problem is quite difficult. The firm studied in this dissertation has several European projects that must support more than 10 languages.
- **Variable Time Horizons**: in this analysis I have dealt exclusively with the situation where SLAs are evaluated over a month long period. An extension of this work would analyze the impact of different time horizons on operating cost.
- **Alternate Service Level Measures**: in this analysis I have focused exclusively on a TSF based SLA. It would be interesting to examine alternative SLAs, in particular multiple measure SLAs which are common in practice.
- **Contracting**: a great deal of literature addresses contracting in supply chain operations. It would be interesting to extend that research to the call center environment and examine the impact of SLA structures on risk and profit sharing.
- **Supply Chain Operations**: this analysis focuses on call center operations, but many of the concepts can be applied to other situations where service level agreements are used. In particular, fill rate SLAs in supply chains.
- **Forecasting Models**: while I developed a general characterization of the call arrival process, forecasting remains an issue. A more detailed analysis, including a closer examination of annual seasonality, could lead to better forecasting models for professional services.

8 Appendix

8.1 Miscellaneous Technical Notes

8.1.1 Simulation Model

The simulation model used in this dissertation is a custom application developed by the author in Microsoft Visual Basic.NET. The code was developed as a port of a model previously developed using Automod. That application is described in (Robbins, Medeiros *et al.* 2006). I ported the code to VB in order to have an increased ability to perform a neighborhood search for the purpose of optimization. Porting the code to VB also allowed a common code base to be used for simulation and scenario generation.

The code was developed completely from scratch implements a basic simulation model in an object oriented framework. For random number generation the code implements a combined multiple recursive generator (CMRG) based on the Mrg32k3a generator described in (L'Ecuyer 1999). The generator was translated from C code posted on L'Ecuyer's web site to VB by the author. This number generator has excellent statistical properties and is considered one of the best generators available. Source code is available from the author.

8.2 Linear and Integer Programming

The Linear and Integer programs described in the dissertation were all formulated using the GAMS modeling language and solved using CPLEX on a Unix host. The models were run on various servers within the Penn State high performance computing environment. Code for all models is available from the author.

9 References

Aarts, E., J. Korst and W. Michiels (2005). Simulated Annealing. Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques. E. K. Burke and G. Kendall. New York, NY, Springer**:** 187-210.

Abelson, M. A. and B. D. Baysinger 1984. Optimal and Dysfunctional Turnover: Toward an Organizational Level Model. *The Academy of Management Review* **9**(2) 331-341.

Accenture 2004. Annual Report 2004.

Aksin, Z., M. Armony and V. Mehrotra 2007. The Modern Call-Center: A Multi-Disciplinary Perspective on Operations Management Research. Working Paper 61p.

Anderson, E. G., Jr. 2001. The Nonstationary Staff-Planning Problem with Business Cycle and Learning Effects. *Management Science* **47**(6) 817-832.

Andrews, B. and H. Parsons 1993. Establishing Telephone-Agent Staffing Levels through Economic Optimization. *Interfaces* **23**(2) 14.

Andrews, B. H. and S. M. Cunningham 1995. L.L. Bean Improves Call-Center Forecasting. *Interfaces* **25**(6) 1.

Andrews, B. H. and H. L. Parsons 1989. L.L. Bean Chooses a Telephone Agent Scheduling System. *Interfaces* **19**(6) 1.

Atlason, J., M. A. Epelman and S. G. Henderson 2004. Call center staffing with simulation and cutting plane methods. *Annals of Operations Research* 333-358.

Avramidis, A. N., W. Chan and P. L'Ecuyer 2007. Staffing multi-skill call centers via search methods and a performance approximation. Working Paper p.

Avramidis, A. N., A. Deslauriers and P. L'Ecuyer 2004. Modeling Daily Arrivals to a Telephone Call Center. *Management Science* **50**(7) 896-908.

Avramidis, A. N., M. Gendreau, P. L'Ecuyer and O. Pisacane 2007. Simulation-Based Optimization of Agent Scheduling in Multiskill Call Centers. 2007 Industrial Simulation Conference.

Avriel, M. and A. C. Williams 1970. The Value of Information and Stochastic Programming. *Operations Research* **18**(5) 947-954.

Aykin, T. 1996. Optimal Shift Scheduling with Multiple Break Windows. *Management Science* **42**(4) 591-602.

Baker, K. R. 1974a. Scheduling a Full-Time Workforce to Meet Cyclic Staffing Requirements. *Management Science* **20**(12, Application Series) 1561-1568.

Baker, K. R. 1974b. Scheduling Full-Time and Part-Time Staff to Meet Cyclic Requirements. *Operational Research Quarterly* **25**(1) 65-76.

Baker, K. R. and M. J. Magazine 1977. Workforce Scheduling with Cyclic Demands and Day-Off Constraints. *Management Science* **24**(2) 161-167.

Banks, J. 2005. *Discrete-event system simulation*, Pearson Prentice Hall. Upper Saddle River, N.J.

Bartholomew, D. J. 1971. The Statistical Approach to Manpower Planning. *The Statistician* **20**(1, Statistics and Manpower Planning in the Firm) 3-26.

Bartholomew, D. J. 1982. *Stochastic models for social processes*, Wiley. Chichester [England]; New York.

Bartholomew, D. J. and A. F. Forbes 1979. *Statistical techniques for manpower planning*, Wiley. Chichester [Eng.]; New York.

Beale, E. M. L. 1955. On Minimizing A Convex Function Subject to Linear Inequalities. *Journal of the Royal Statistical Society. Series B (Methodological)* **17**(2) 173-184.

Bechtold, S. E. and L. W. Jacobs 1990. Implicit Modeling of Flexible Break Assignments in Optimal Shift Scheduling. *Management Science* **36**(11) 1339-1351.

Birge, J. R. 1982. The Value of the Stochastic Solution in Stochastic Linear Programs, with Fixed Recourse. *Mathematical Programming* **24** 314-325.

Birge, J. R. 1985. Decomposition and Partitioning Methods for Multistage Stochastic Linear Programs. *Operations Research* **33**(5) 989-1007.

Birge, J. R. and F. Louveaux 1997. *Introduction to Stochastic Programming*, Springer. New York.

Bordoloi, S. K. and H. Matsuo 2001. Human resource planning in knowledge-intensive operations: A model for learning with stochastic turnover. *European Journal of Operational Research* **130**(1) 169.

Borst, S., A. Mandelbaum and M. I. Reiman 2004. Dimensioning Large Call Centers. *Operations Research* **52**(1) 17-35.

Box, G. E. P. and N. R. Draper 1987. *Empirical model-building and response surfaces*, Wiley. New York.

Box, G. E. P., J. S. Hunter and W. G. Hunter 2005. *Statistics for experimenters: design, innovation, and discovery*, Wiley-Interscience. Hoboken, N.J.

Box, G. E. P., W. G. Hunter and J. S. Hunter 1978. *Statistics for experimenters: an introduction to design, data analysis, and model building*, Wiley. New York.

Box, G. E. P., G. N. Jenkins and G. C. Reinsel 1994. *Time series analysis: forecasting and control*, Prentice Hall. Englewood Cliffs, N.J.

Bres, E. S., D. Burns, A. Charnes and W. W. Cooper 1980. A Goal Programming Model for Planning Officer Accessions. *Management Science* **26**(8) 773-783.

Brown, L., N. Gans, A. Mandelbaum, A. Sakov, S. Haipeng, S. Zeltyn and L. Zhao 2005. Statistical Analysis of a Telephone Call Center: A Queueing-Science Perspective. *Journal of the American Statistical Association* **100**(469) 36-50.

Brown, L., N. Gans, A. Mandelbaum, A. Sakov, S. Zeltyn, L. Zhao and S. Haipeng 2002. Statistical Analysis of a Telephone Call Center: A Queueing-Science Perspective. Accepted to JASA. Working Paper p.

Brusco, M. J. and L. W. Jacobs 1998. Personnel Tour Scheduling When Starting-Time Restrictions Are Present. *Management Science* **44**(4) 534-547.

Brusco, M. J. and L. W. Jacobs 2000. Optimal Models for Meal-Break and Start-Time Flexibility in Continuous Tour Scheduling. *Management Science* **46**(12) 1630-1641.

Brusco, M. J. and T. R. Johns 1996. A sequential integer programming method for discontinuous labor tour scheduling. *European Journal of Operational Research* **95**(3) 537-548.

Burke, E., K., P. De Causmaecker, G. Vanden Berghe and H. Van Landeghem 2004. The State of the Art of Nurse Rostering. *Journal of Scheduling* **7**(6) 441.

Burke, E. K. and G. Kendall, Eds. (2005). *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques* Springer. New York, NY.

Cachon, G. P. and C. Terwiesch 2006. *Matching Supply With Demand*, McGraw-Hill Irwin. New York, NY.

Cezik, M. and P. L'Ecuyer 2007. Staffing Multiskill Call Centers via Linear Programming and Simulation. Working Paper 34p.

Charnes, A., W. W. Cooper and R. J. Niehaus 1978. *Management science approaches to manpower planning and organization design*, North-Holland Publishing Co. Amsterdam; New York.

Cordes, C. L. and T. M. Dougherty 1993. A Review and Integration of Research on Job Burnout. *Academy of Management Review* **18**(4) 621-656.

Cotton, J. L. and J. M. Tuttle 1986. Employee Turnover: A Meta Analysis and Review with Implications for Research. *Academy of Management Review* **11**(1) 55-70.

Dantzig, G. B. 1954. A Comment on Edie's "Traffic Delays at Toll Booths". *Journal of the Operations Research Society of America* **2**(3) 339-341.

Dantzig, G. B. 1955. Linear Programming under Uncertainty. *Management Science* **1**(3/4) 197-206.

Dantzig, G. B. and G. Infanger 1993. Multi-stage stochastic linear programs for portfolio optimization. *Annals of Operations Research* **45**((1993)) 59-76.

Dietrich, B. and T. P. Harrison 2006. The Emerging Science of Service Management and the Complementary Role of OR/MS. *OR/MS Today* **June 2006** 10.

Dupačová, J. 1995. Postoptimality for Multistage stochastic linear programs. *Annals of Operations Research* **56**(1995) 65-78.

Dupačová, J., G. Consigli and S. W. Wallace 2000. Scenarios for multistage stochastic programs. *Annals of Operations Research* **100**.

Dupačová, J., N. Gröwe-Kuska and W. Römisch 2003. Scenario reduction in stochastic programming. *Mathematical Programming* **95**(3) 493-511.

Dupacova, J. and R. Wets 1988. Asymptotic Behavior of Statistical Estimators and of Optimal Solutions of Stochastic Optimization Problems. *The Annals of Statistics* **16**(4) 1517-1549.

Ebert, R. J. 1976. Aggregate Planning with Learning Curve Productivity. *Management Science* **23**(2) 171-182.

Fang, K.-T. and M.-L. Du 1998. Uniform Design v 3.0, Hong Kong Baptist University.

Fang, K.-T. and D. K. J. Lin (2003). Uniform Experiment Design and their Application in Industry. Handbook of Statistics. R. Khattree and C. R. Rao, Elsevier. **Vol 22**.

Fang, K.-T., D. K. J. Lin, P. Winker and Y. Zhang 2000. Uniform Design: Theory and Application. *Technometrics* **42**(3) 237-248.

Feldman, Z., A. Mandelbaum, W. A. Massey and W. Whitt 2005. Staffing of Time-Varying Queues to Achieve Time-Stable Performance. Working Paper 27p.

Forbes, A. F. 1971. Non-Parametric Methods of Estimating the Survivor Function. *The Statistician* **20**(1, Statistics and Manpower Planning in the Firm) 27-52.

Freimer, M. B., D. J. Thomas and J. T. Linderoth 2006. Reducing Bias in Stochastic Linear Programs with Sampling Methods. Working Paper 37p.

Fu, M. C. 2002. Optimization for Simulation: Theory vs. Practice. *INFORMS Journal on Computing* **14**(3) 192-215.

Gaballa, A. and W. Pearce 1979. Telephone Sales Manpower Planning at Qantas. *Interfaces* **9**(3) 9p.

Gaimon, C. 1997. Planning Information Technology-Knowledge Worker Systems. *Management Science* **43**(9) 1308-1328.

Gaimon, C. and G. L. Thompson 1984. A Distributed Parameter Cohort Personnel Planning Model that Uses Cross-Sectional Data. *Management Science* **30**(6) 750-764.

Gans, N., G. Koole and A. Mandelbaum 2003. Telephone call centers: Tutorial, review, and research prospects. *Manufacturing & Service Operations Management* **5**(2) 79-141.

Gans, N. and Y.-P. Zhou 2002. Managing learning and turnover in employee staffing. *Operations Research* **50**(6) 991.

Gans, N. and Y.-P. Zhou 2007. Call-Routing Schemes for Call-Center Outsourcing. *Manufacturing & Service Operations Management* **9**(1) 33-51.

Garnett, O., A. Mandelbaum and M. I. Reiman 2002. Designing a Call Center with impatient customers. *Manufacturing & Service Operations Management* **4**(3) 208-227.

Gass, S. I., R. W. Collins, C. W. Meinhardt, D. M. Lemon and M. D. Gillette 1988. The Army Manpower Long-Range Planning System. *Operations Research* **36**(1) 5-17.

Gassman, H. I. 1990. MSLip: A computer code for the multistage stochastic linear programming problem. *Mathematical Programming* **47** 407-423.

Gendreau, M. and J.-Y. Potvin (2005). Tabu Search. <u>Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques</u>. E. K. Burke and G. Kendall. New York, NY, Springer**:** 165-186.

Geoffrion, A. M. 1970. Elements of Large-Scale Mathematical Programming: Part I: Concepts. *Management Science* **16**(11, Theory Series) 652-675.

Glover, F. and G. A. Kochenberger 2003. *Handbook of metaheuristics*, Kluwer Academic Publishers. Boston.

Green, L. and P. Kolesar 1991. The Pointwise Stationary Approximation for Queues with Nonstationary Arrivals. *Management Science* **37**(1) 84-97.

Green, L. V., P. Kolesar and J. Soares 2003. An Improved Heuristic for Staffing Telephone Call Centers with Limited Operating Hours. *Production and Operations Management* **12**(1) 46-61.

Green, L. V. and P. J. Kolesar 1997. The Lagged PSA for Estimating Peak Congestion in Multiserver Markovian Queues with Periodic Arrival Rates. *Management Science* **43**(1) 80-87.

Green, L. V., P. J. Kolesar and J. Soares 2001. Improving the SIPP Approach for Staffing Service Systems That Have Cyclic Demands. *Operations Research* **49**(4) 549-564.

Green, L. V., P. J. Kolesar and W. Whitt 2005. Coping with Time-Varying Demand when Setting Staffing Requirements for a Service System. Working Paper 58p.

Grinold, R. C. 1976. Manpower Planning with Uncertain Requirements. *Operations Research* **24**(3) 387-399.

Halfin, S. and W. Whitt 1981. Heavy-Traffic Limits for Queues with Many Exponential Servers. *Operations Research* **29**(3) 567-588.

Hall, R. W. 1991. *Queueing methods: for services and manufacturing*, Prentice Hall. Englewood Cliffs, NJ.

Hansen, P. and N. Mladenovic 2001. Variable neighborhood search: Principles and applications. *European Journal of Operational Research* **130**(3) 449-467.

Hansen, P. and N. Mladenovic (2005). Variable Neighborhood Search. <u>Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques</u>. E. K. Burke and G. Kendall. New York, NY, Springer**:** 211-238.

Hanssmann, F. and S. W. Hess 1960. A Linear Programming Approach to Production and Employment Scheduling. *Management Technology* **1**(1) 46-51.

Harrison, T. P. and J. E. Ketz 1989. Modeling Learning Effects via Successive Linear Programming. *European Journal of Operational Research* **40**(1) 78-84.

Henderson, D., S. H. Jacobson and A. W. Johnson (2003). The Theory and Practice of Simulated Annealing. Handbook of metaheuristics. F. Glover and G. A. Kochenberger. Boston, Kluwer Academic Publishers.

Henderson, W. B. and W. L. Berry 1976. Heuristic Methods for Telephone Operator Shift Scheduling: An Experimental Analysis. *Management Science* **22**(12) 1372-1380.

Higle, J. L. 2005. Stochastic Programming: Optimization When Uncertainty Matters. *Tutorials in Operations Research*.

Higle, J. L., B. Rayco and S. Sen 2004. Stochastic Scenario Decomposition for Multi-Stage Stochastic Programs. Working Paper 41p.

Higle, J. L. and S. Sen 1996. *Stochastic decomposition: a statistical method for large scale stochastic linear programming*, Kluwer. Dordrecht; Boston.

Higle, J. L. and S. Sen 2006. Multistage stochastic convex programs: Duality and its implications. *Annals of Operations Research* **142** 129-146.

Holman, D. 2002. Employee wellbeing in call centers. *Human Resource Management Journal* **12**(4) 35-50.

Holt, C. C., F. Modigliani, J. F. Muth and H. A. Simon 1960. *Planning production, inventories, and work force*, Prentice-Hall. Englewood Cliffs, N.Y.

Holz, B. W. and J. M. Wroth 1980. Improving Strength Forecasts: Support For Army Manpower Management. *Interfaces* **10**(6) 37.

Hoyland, K. and S. W. Wallace 2001. Generating Scenario Trees for Multistage Decision Problems. *Management Science* **47**(2) 295-307.

Huang, C. C., I. Vertinsky and W. T. Ziemba 1977. Sharp Bounds on the Value of Perfect Information. *Operations Research* **25**(1) 128-139.

IBM 2004. Annual Report.

Infosys 2005. Annual Report 2004-2005.

Jennings, O. B. and A. Mandelbaum 1996. Server staffing to meet time-varying demand. *Management Science* **42**(10) 1383.

Jennings, O. B., A. Mandelbaum, W. A. Massey and W. Whitt 1996. Server Staffing to Meet Time-Varying Demand. *Management Science* **42**(10) 1383-1394.

Johnson, M. E. 1987. *Multivariate statistical simulation*, Wiley. New York.

Kall, P. 1976. *Stochastic linear programming*, Springer-Verlag. Berlin; New York.

Kall, P. and J. Mayer 2005. *Stochastic linear programming: models, theory, and computation*, Springer Science. New York.

Kall, P. and S. W. Wallace 1994. *Stochastic programming*, Wiley. Chichester; New York.

King, A. J. and R. T. Rockafeller 1993. Asymptotic Theory for Solutions in Statistical Estimation and Stochastic Programming. *Mathematics of Operations Research* **18**(1) 148-162.

Kleywegt, A. J., A. Shapiro and T. Homem-de-Mello 2001. The Sample Average Approximation Method for Stochastic Discrete Optimization. *SIAM Journal of Optimization* **12**(2) 479-502.

Koole, G. and A. Pot 2005. An Overview of Routing and Staffing in Multi-Skill Contact Centers. Working Paper 1-32p.

Koole, G. and E. van der Sluis 2003. Optimal shift scheduling with a global service level constraint. *IIE Transactions* **35** 1049-1055.

Krass, I. A., M. C. Pinar, T. J. Thompson and S. A. Zenios 1994. A Network Model to Maximize Navy Personnel Readiness and Its Solution. *Management Science* **40**(5) 647-661.

Kutner, M. H., C. Nachtsheim, J. Neter and W. Li 2005. *Applied linear statistical models*, McGraw-Hill/Irwin. Boston; New York.

L'Ecuyer, P. 1999. Good Parameters and Implementations for Combined Multiple Recursive Random Number Generators. *Operations Research* **47**(1) 159-164.

Lasdon, L. S. 2002. *Optimization theory for large systems*, Macmillan. [New York].

Law, A. M. 2007. *Simulation modeling and analysis*, McGraw-Hill. Boston.

Law, A. M. and W. D. Kelton 2000. *Simulation modeling and analysis*, McGraw-Hill. Boston.

Lawless, J. F. 2003. *Statistical models and methods for lifetime data*, Wiley-Interscience. Hoboken, N.J.

Linderoth, J. T., A. Shapiro and S. Wright 2006. The empirical behavior of sampling methods for stochastic programming. *Annals of Operations Research* **142**(1) 215-241.

Madansky, A. 1960. Inequalities for Stochastic Linear Programming Problems. *Management Science* **6**(2) 197-204.

Mak, W.-K., D. P. Morton and R. K. Wood 1999. Monte Carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters* **24**(1-2) 47-56.

Mandelbaum, A. and S. Zeltyn 2004. Service Engineering in Action: The Palm/Erlang-A Queue, with Applications to Call Centers Draft, December 2004. Working Paper p.

Mandelbaum A., Sakov A. and Z. S. 2001. Empirical Analysis of a Call Center. Working Paper 73p.

Mason, A. J., D. M. Ryan and D. M. Panton 1998. Integrated Simulation, Heuristic and Optimisation Approaches to Staff Scheduling. *Operations Research* **46**(2) 161-175.

McKay, M. D., R. J. Beckman and W. J. Conover 1979. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics* **21**(2) 239-245.

Miller, A. C., III and T. R. Rice 1983. Discrete Approximations of Probability Distributions. *Management Science* **29**(3) 352-362.

Mo, Y., T. P. Harrison and R. R. Barton 2006. Solving Stochastic Programming Models in Supply-Chain Design using Sampling Heuristics. Working Paper 21p.

Morey, R. C. and J. M. McCann 1980. Evaluating and Improving Resource Allocation for Navy Recruiting. *Management Science* **26**(12, Application Series) 1198-1210.

Nemhauser, G. L. and L. A. Wolsey 1988. *Integer and combinatorial optimization*, Wiley. New York.

Papadimitriou, C. H. and K. Steiglitz 1998. *Combinatorial optimization: algorithms and complexity*, Dover Publications. Mineola, N.Y.

Pflug, G. C. 1996. *Optimization of stochastic models: the interface between simulation and optimization*, Kluwer Academic. Boston, Mass.

Pflug, G. C. 2001. Scenario tree generation for multiperiod financial optimization by optimal discretization. *Mathematical Programming* **89**(2) 251-271.

Pinedo, M. 2005. *Planning and scheduling in manufacturing and services*, Springer. New York, NY.

Prekopa, A. 1995. *Stochastic programming*, Kluwer Academic Publishers. Dordrecht; Boston.

Price, W. L. (1978). Measuring Labor Turnover for Manpower Planning. Management science approaches to manpower planning and organization design. A. Charnes, W. W. Cooper and R. J. Niehaus. Amsterdam; New York, North-Holland Publishing Co.**:** 61-73.

Quinn, P., B. Andrews and H. Parsons 1991. Allocating Telecommunications Resources at L. L. Bean, Inc. *Interfaces* **21**(1) 75.

Reeves, C. (2003). Genetic Algorithms. Handbook of metaheuristics. F. Glover and G. A. Kochenberger. Boston, Kluwer Academic Publishers.

Riordan, J. 1962. *Stochastic service systems*, Wiley. New York.

Robbins, T. R. 2007. Addressing Arrival Rate Uncertainty in Call Center Workforce Management. 2007 IEEE/INFORMS International Conference on Service Operations and Logistics, and Informatics. Philadelphia, PA, Penn State University**:** 6.

Robbins, T. R. and T. P. Harrison 2006. Manpower Planning with Limited Hiring Opportunities-The Value of Stochastic Modeling. Seventeenth Annual Conference of POMS, Boston, MA.

Robbins, T. R., D. J. Medeiros and P. Dum 2006. Evaluating Arrival Rate Uncertainty in Call Centers. Proceedings of the 2006 Winter Simulation Conference, Monterey, CA.

Robbins, T. R., D. J. Medeiros and T. P. Harrison 2007. Partial Cross Training in Call Centers with Uncertain Arrivals and Global Service Level Agreements. Submitted to 2007 Winter Simulation Conference, Washington, DC.

Rockafellar, R. T. and R. J.-B. Wets 1991. Sceanrios and policy aggregation optimization under uncertainty. *Mathematics of Operations Research* **16**(1) 119-147.

Ross, S. M. 2003. *Introduction to probability models*, Academic Press. San Diego, CA.

Saltzman, R. M. and V. Mehrotra 2001. A Call Center Uses Simulation to Drive Strategic Change. *Interfaces* **31**(3) 87.

Santner, T. J., B. J. Williams and W. Notz 2003. *The design and analysis of computer experiments*, Springer. New York.

Sastry, K., D. Goldberg and G. Kendall (2005). Genetic Algorithms. Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques. E. K. Burke and G. Kendall. New York, NY**:** 97-126.

Schindler, S. and T. Semmel 1993. Station Staffing at Pan American World Airways. *Interfaces* **23**(3) 91-98.

Segal, M. 1974. The Operator-Scheduling Problem: A Network-Flow Approach. *Operations Research* **22**(4) 808-823.

Shapiro, A. 1991. Asymptotic analysis of stochastic programs. *Annals of Operations Research* **30** 169-186.

Shapiro, A. and T. Homem-de-Mello 2000. On the Rate of Convergence of Optimal Solutions of Monte Carlo Approximations of Stochastic Programs. *SIAM Journal of Optimization* **11**(1) 70-86.

Shrimpton, D. and A. M. Newman 2005. The US Army Uses a Network Optimization Model to Designate Career Fields for Officers. *Interfaces* **35**(3) 230-237.

Simpson, T. W., D. K. J. Lin and W. Chen 2001. Sampling Strategies for Computer Experiments: Design and Analysis. *International Journal of Reliability and Application* **2**(3) 209-240.

Singh, J. 2000. Performance Productivity and Quality of Frontline Employees in Service Organizations. *Journal of Marketing* **64**(2) 15-34.

Singh, J., J. R. Gollsby and G. K. Rhoads 1994. Behavioral and Psychological Consequences of Boundary Spanning Burnout for Customer Service Representatives. *Journal of Marketing Research* **31**(4) 558-569.

Thompson, G. M. 1995. Improved Implicit Optimal Modeling of the Labor Shift Scheduling Problem. *Management Science* **41**(4) 595-607.

Wallace, R. B. and W. Whitt 2005. A Staffing Algorithm for Call Centers with Skill-Based Routing. *Manufacturing & Service Operations Management* **7**(4) 276-294.

Wang, Y., D. K. J. Lin and K.-T. Fang 1995. Designing Outer Array Points. *Journal of Quality Technology* **27**(3) 226-241.

Warner, D. M. 1976. Scheduling Nursing Personnel according to Nursing Preference: A Mathematical Programming Approach. *Operations Research* **24**(5, Special Issue on Health Care) 842-856.

Warner, D. M. and J. Prawda 1972. A Mathematical Programming Model for Scheduling Nursing Personnel in a Hospital. *Management Science* **19**(4, Application Series, Part 1) 411-422.

Whitt, W. 1989. An Interpolation Approximation for the Mean Workload in a GI/G/1 Queue. *Operations Research* **37**(6) 936-952.

Whitt, W. 2006a. Sensitivity of Performance in the Erlang A Model to Changes in the Model Parameters. *Operations Research* **54**(2) 247-260.

Whitt, W. 2006b. Staffing a Call Center with Uncertain Arrival Rate and Absenteeism. *Production and Operations Management* **15**(1) 88-102.

Wikipedia. (2007). "Importance sampling --- Wikipedia-The Free Encyclopedia." from http://en.wikipedia.org/w/index.php?title=Importance_sampling&oldid=115513404 .

Witt, L. A., M. C. Andrews and D. S. Carlson 2004. When Conscientiousness Isn't Enough: Emotional Exhaustion and Performance Among Call Center Customer Service Representatives. *Journal of Management* **30**(1) 149-160.

Wolff, R. W. 1982. Poisson Arrivals See Time Averages. *Operations Research* **30**(2) 223-231.

Wolff, R. W. 1989. *Stochastic modeling and the theory of queues*, Prentice Hall. Englewood Cliffs, N.J.

Wolsey, L. A. 1998. *Integer programming*, J. Wiley. New York.

Yu, G., J. Pachon, B. Thengvall, D. Chandler and A. Wilson 2004. Optimizing Pilot Planning and Training for Continental Airlines. *Interfaces* **34**(4) 253.

Vita

Thomas R. Robbins

Penn State University
trr147@psu.edu
www.personal.psu.edu/faculty/t/r/trr147/

Office Address

462A Business Building
Penn State University
University Park, PA 16802
Phone: 814 883-0749

Home Address

100 Rainlo Street
State College, PA 16801

Education

- PhD - Penn State Smeal College, Supply Chain and Information Systems 2007
- Master of Business Administration - Weatherhead School of Management, Case Western Reserve University. 1989
- Bachelor of Science in Electrical Engineering - Pennsylvania State University. 1985

Employment History

- PhD candidate and independent consultant (5/03-present)
- Aztec Software (7/02 – 4/03)
- Ernst & Young /Cap Gemini Ernst & Young (7/97-7/02)
- Advanced Graphical Applications (6/95-6/97)
- Boston Chicken (6/94-6/95)
- IBM Consulting Group (4/93-6/94)
- Ernst & Young (1/88-4/93)
- General Electric (5/85-12/87)