Staffing of Time-Varying Queues
To Achieve Time-Stable Performance

Project by: Zohar Feldman szoharf@t2.technion.ac.il

Supervised by: Professor Avishai Mandelbaum avim@ie.technion.ac.il

Industrial Engineering and Management
Technion, Haifa 32000
Israel

June 17th, 2004

Abstract:

Following the ideas and infinite-server heuristics of Jennings, Mandelbaum, Massey and Whitt (2001), a simulation-based algorithm is developed for staffing **time-varying** queues, including abandonment. The algorithm is designed to achieve a given **constant** probability of delay, say α: α near 0 would lead to a Quality-Driven (QD) operation that emphasized service-quality; α near 1 would lead to an Efficiency-Driven (ED) operation, where the focus is on servers' efficiency; and α strictly between 0 and 1 results in a Quality & Efficiency Driven (**QED**) operation, with a careful balance between high levels of service-quality and server-efficiency.

The algorithm results in a time-varying staffing function of the form $S(t) = R(t) + \beta\sqrt{R(t)}$:

S(t) is the recommended number of servers at time t and R(t) is the given offered load on the system; the constant β reflects service grade that depends on the desired α – the smaller the α (higher service level) the larger the β (more servers) . Under this staffing rule, other performance measures (servers' utilization, average waits, queue-lengths, tail-probabilities) turn out close to being constant as well. Moreover, the functional relation between α and β coincides with those known for stationary models. **Thus, we achieve predictably constant (time-stable) performance in a time-varying environment, for operations that are ED, QD or QED.** This is clearly desirable but, apriori, not so clearly feasible. Finally, we analyze convergence of our algorithm, which can be provided a theoretical backing within the framework of Markovian Service Networks (Mandelbaum, Massey and Reiman (1998)).

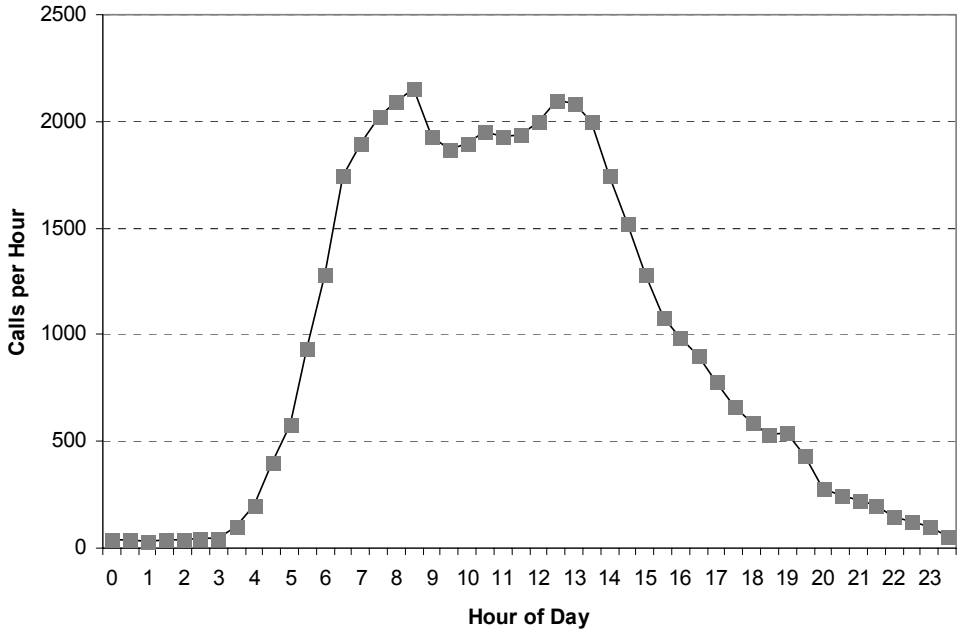
1. Introduction

1.1 Background on Services and Call Centers:

Service systems (banks, insurance, hospitals, government, etc.) are central in our life, and are vital for our economic viability. Services employ about 60-80% of the workforce in Western Economies, and their importance is sharply on the rise, both within service- and manufacturing-companies. In our service-driven economy, it is estimated that over 70% of the business transactions are carried out over the phone. Most of these transactions are processed by telephone *Call Centers*, which have become the preferred and prevalent means for companies to communicate with their customers. Indeed, it is estimated that more than 3% of the U.S. workforce is employed in call centers – more that in agriculture! For an overview of the world of call centers and its state-of-art research, readers are referred to the recent review by Gans, Koole and Mandelbaum (2003).

The modern call center is a highly complex operation that fuses advanced technology and human beings. But the economic and managerial significance of the latter clearly out-weights the former. More specifically, labor costs (agents' salaries, training, etc.) typically run as high as 70% of the total operating costs of a call center, and attrition rates in call centers reach anywhere from 30% per year (considered low) to over 200% at times. In such circumstances, perhaps the most important operational decision to be made is *staffing*: what is the appropriate number of telephone agents that are to be accessible for serving calls. Here, over-staffing would have a negative financial effect, while under-staffing would lead to low service-levels and over-worked agents.

Figure 1: Hourly call volumes to a medium-size call center



1.2 The Staffing Problem:

The staffing problem typically takes the following form: under an existing operational reality, and given a desired service grade (performance measure), one seeks the least number of agents that is required to meet a given service-level constraint. This problem, which has been given a great deal of attention over the years (see Chapter 4 in Gans et. al.), is challenging both

theoretically and practically, which is of no surprise. To wit, the natural *theoretical* settings for the staffing problem are time-varying queues, and these are notoriously difficult to analyze. The *practical* implication of staffing was already discussed above, and it can be further amplified by considering, for example, a bank employing 10,000 telephone agents and catering to millions of customers per day: clearly, even small gains in operational efficiency or service quality are bound to result in major payoffs.

Figure 1 depicts a typical arrival-rate function to a telephone call center. Call volumes are low around midnight, starting to increase in the early hours of the morning, peaking at late morning, then dropping somewhat around 12:00 (lunch break), rising again after noon time, and dropping thereafter to midnight levels. The arrival-rate function is an average of several similar days; the actual number of arrivals, in a given hour on a given day, fluctuates randomly around this average. (As mentioned, the functional shape in Figure 1 is typical; the particular values for the arrival rates were adapted from Green, Kolesar and Soares (2001), in order to benchmark our algorithem; see Section 6.)

Staffing planners are thus faced with two sources of variability: *predictable* variability, namely average time-variations over the day, and *random* (stochastic) variability around the average. Prevalent staffing algorithms are designed to cope only with stochastic variability, which is enabled by assuming out predictable variability in various ways. Then, however, significant predictable variability would result in excessively fluctuating performance (see Figure 16), which has its clear negative consequences. For example, it would indicate that some parts of the day are over- and others are under-staffed. A reformulation of the staffing problem hence naturally arises, which is the subject of the present study: **given a <u>daily</u> performance goal, and faced with predictable and stochastic variability, find the minimal staffing levels that meet this performance goal <u>stably</u> over the day.**

1.3 Our Solution to the Staffing Problem

In this paper, we present a simulation-based *Iterative Staffing Algorithm* for *time-varying* queues. Our algorithm is based on ideas in Whitt (1992) and Jennings, Mandelbaum, Massey and Whitt (1996). (Indeed, the Infinite-Server Approximation in these papers serves as the initial phase of our algorithm, which we then refine iteratively.)

Following Jennings et. al. (1996), we assume that, in principle, any number of servers can be assigned at any time; in our implementation, however, time is divided into short intervals (say 5 minutes), and we keep the number of servers fixed over an interval. The service discipline is FCFS, and servers follow an exhaustive service discipline: a server that finishes a shift in the middle of a service will complete the service and sign out only when finished. (Our results prevail also for pre-emptive service disciplines under which servers leave at end-of-shifts and their customers, if any, are moved to the front of the queue.)

The contribution of our algorithm is its ability to accommodate very general queueing models, such as $G_t/G/s_t+G$ queues (queues with abandonment) and more. Specifically, given a target probability of delay, we identify time-varying staffing levels under which the actual probability of delay essentially equals the given target, **at all times**. Other performance measures, such as the average waiting time, tail delay-probabilities and the probability of abandon, turn out **constant** over time as well. In fact, global performance measures were found to coincide with the performance measures of a naturally-corresponding stationary system.

1.4 An Illustrative Example: Moderate (Im)Patience

To facilitate reading, we start with an example that is representative of what is to follow. Consider a queueing system that is faced with an arrival-rate function $\lambda(t) = 100 + 20 \cdot \sin(t)$; service times are exponential having mean 1, and customers' (im)patience is also exponential with mean 1. Our algorithm generates staffing functions, for any given target delay probability α, as in the following three figures. (Blue is the arrival and Red the staffing function.)

Figure 2: *Quality-Driven* (QD) Staffing Function (α=0.1)

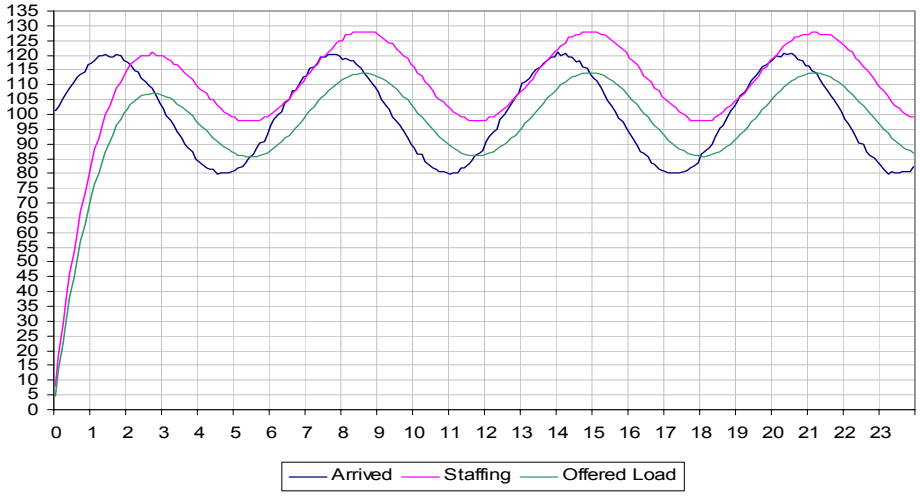


Figure 3: *Efficiency-Driven* (ED) Staffing Function (α=0.9)

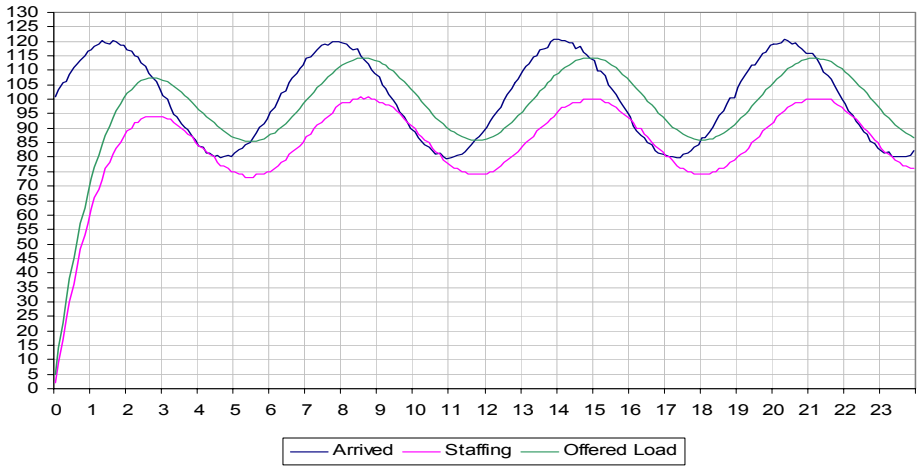
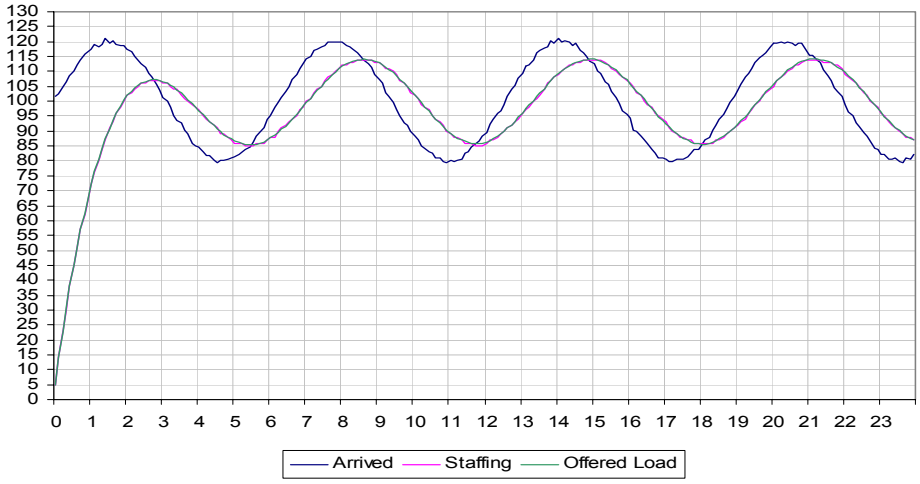
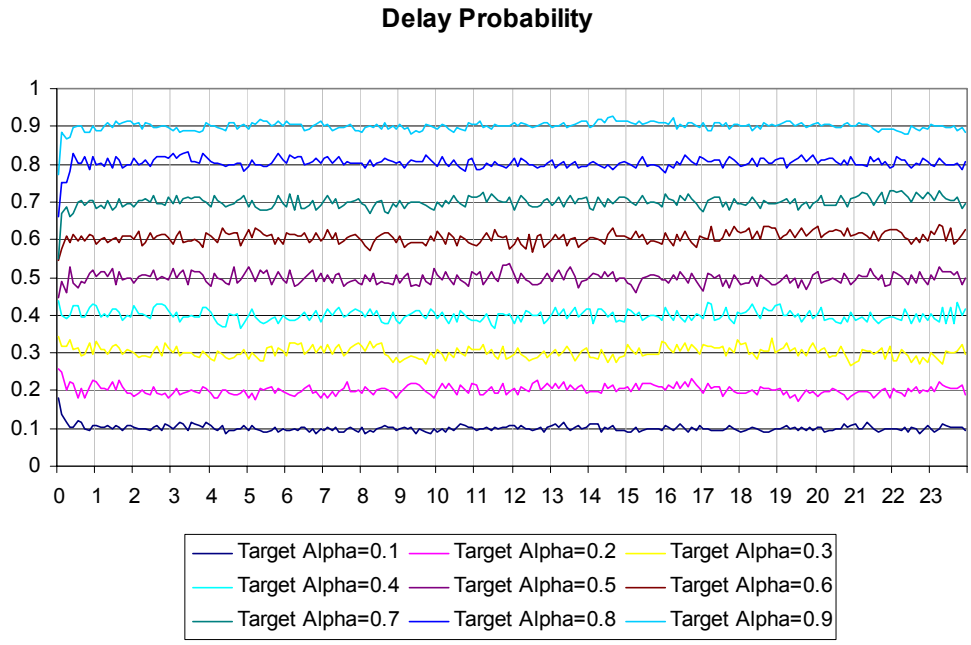


Figure 4: *Quality- and Efficiency-Driven* (QED) Staffing Function (α=0.5)



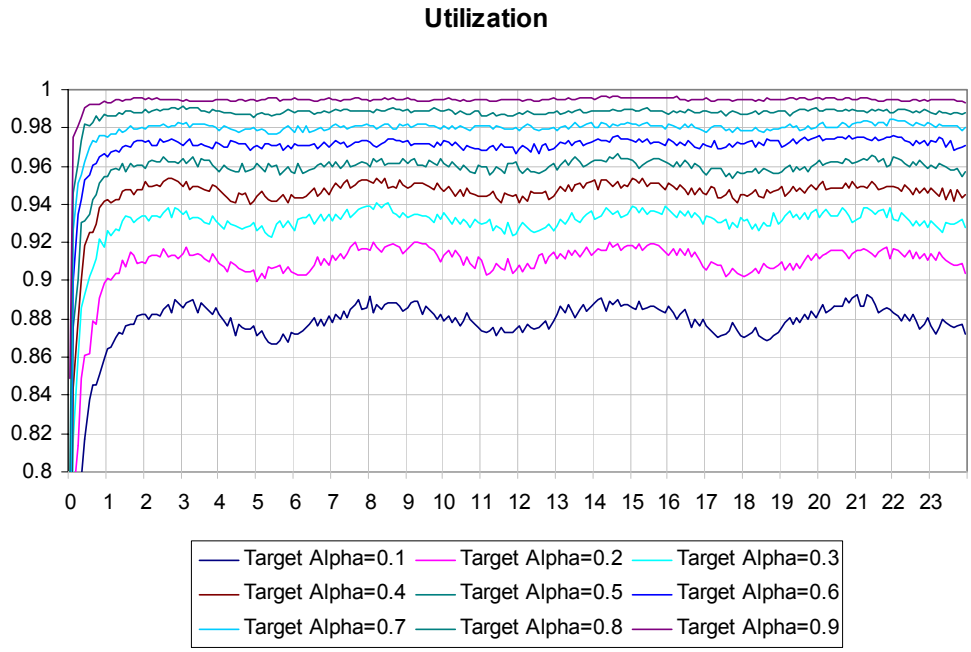
Under the staffing levels arrived at by our algorithm, the target delay probabilities are remarkably accurately (stable) at all times**:**

Figure 5: Delay probability summary for various α's.



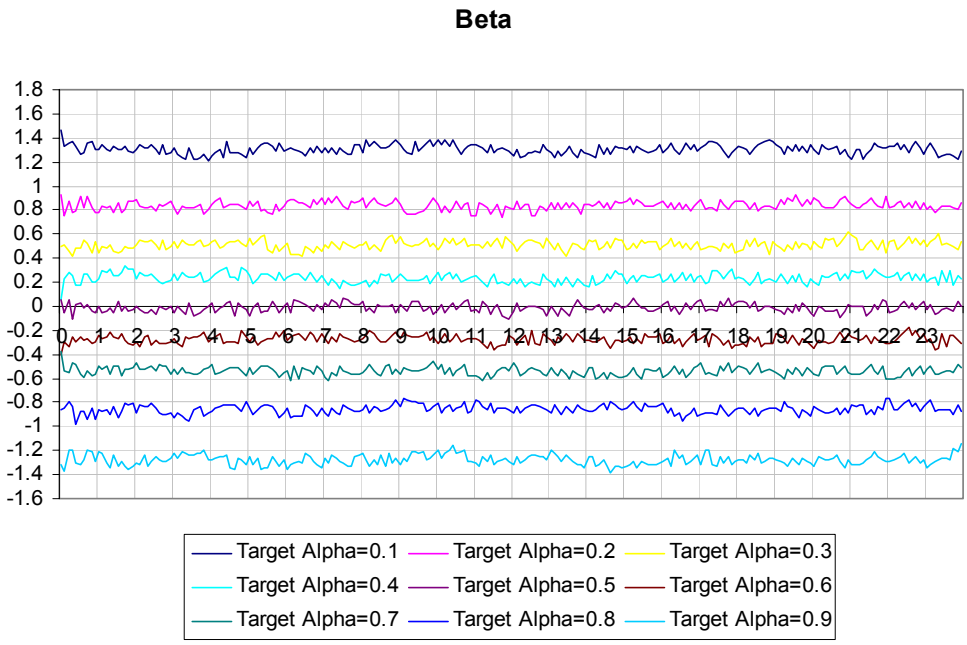
Moreover, with stabilizing the delay probability, other performance measures (e.g. utilization, abandon probability etc.) are found to be quite stable as well. For example:

Figure 6: Utilization Summary



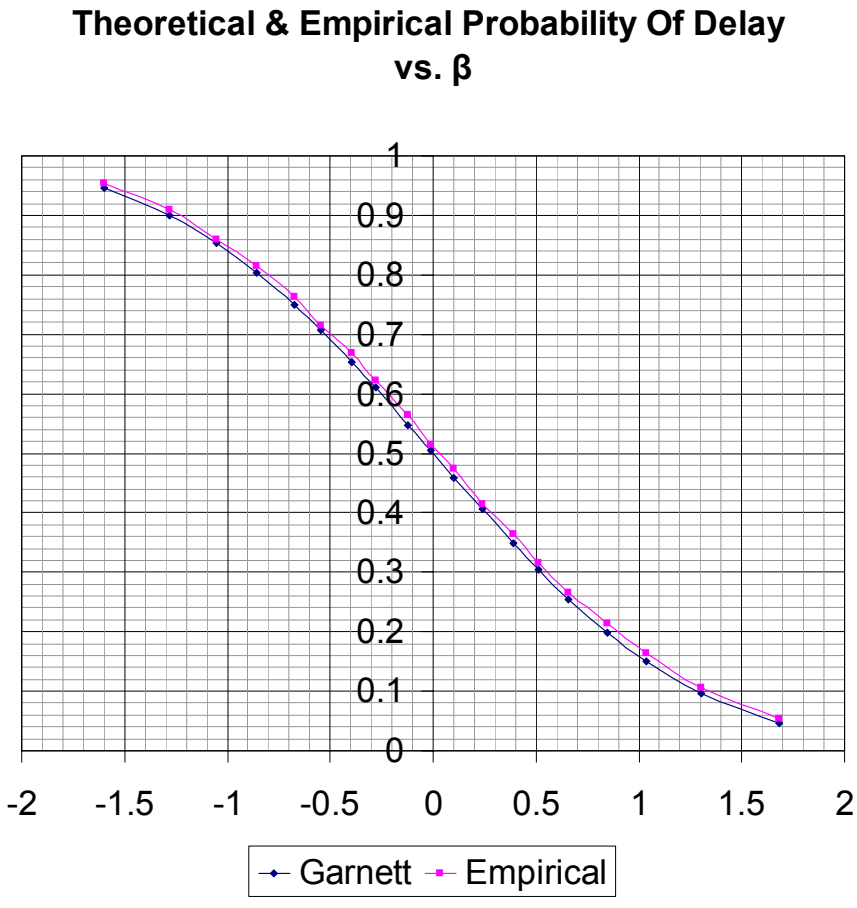
In addition, the implied service-grade β (defined in Subsection 2.2), also turns out constant:

Figure 7: Summary of Implied Service-grade β



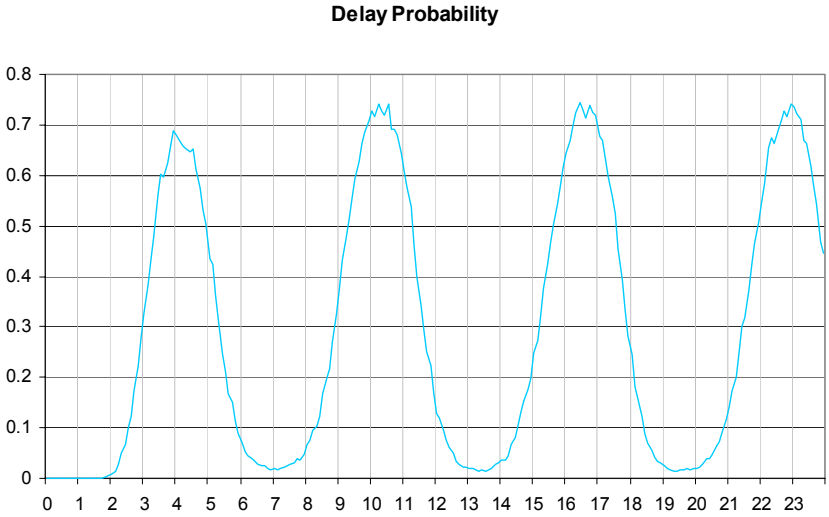
Finally, known functional relations between α and β (by Halfin-Whitt and Garnett), that were derived for stationary queues, prevail very accurately also for our time-varying queues:

Figure 8: **Algorithm-Generated Performance vs. Garnett's Function**



We show that other common approximations fail miserably. For example, using the PSA approximation (see section 5.2) with the aforementioned case, delay probability and other performance measures are quite poor.

Figure 9: Delay probability obtained by the PSA with target delay probability 0.2



1.5 Contents Summary

In §2 we describe our algorithm, followed by definitions of the performance measures that we display throughout this paper. Then, in §3, we start by revisiting and elaborating on the "challenging example" in Jennings, Mandelbaum, Massey & Whitt (2001). Additional examples are presented in §4-§5, emphasizing the stationary (time-stable) performance of our staffing algorithm. In §6, we analyze a realistic example (the one presented in Figure 1). In contrast to Green et. al., we also incorporate abandonment, which significantly and positively impacts staffing results. In §7, the dynamics of the iterative algorithm is explored and theoretically supported. We conclude, in §8, with commentary and ideas for future research.

2. The Algorithm

We determine dynamic staffing levels, so as to achieve a given constant probability of delay at all times. It is then demonstrated that, when the latter prevails other performance measured, such as servers' utilization, tail probabilities, average waits and abandonment probabilities, remain stable as well.

One fixes an arrival rate function, a service time distribution, a patience distribution (when relevant) and a time-horizon [0,T].

Note: During this paper we use the notation $X_n(t)$ to denote the value of X in the n-th iteration at time $t \in [0,T]$ (the given time horizon). Although our algorithm is time-continuous, its implementation is in discrete time, namely the time-horizon is divided into small intervals of length Δ (in this paper we use $\Delta = 0.1$ in all cases), and we then actually treat $X_i(t)$ as constant over the interval $[(k-1)\Delta, k\Delta)$, $k = 1,2,\ldots.$

Denote by $s_n(t)$, $0 \le t \le T$, the staffing level at time t, as determined in iteration n=1,2,…

Under this staffing function, let $L_n(t)$ stand for the total number in the system at time t.

Our algorithm is initialized with a large enough number of servers ($s_0(\cdot) \equiv \infty$, or ample servers). The probability of delay (i.e. all servers are busy upon arrival) is then negligible.

The algorithm performs iteratively the following steps, until convergence is obtained. Here, convergence means that the staffing levels do not change after an iteration. (Practically, they are allowed to change by some threshold)

2.1. The Algorithm:

- Evaluate the distribution of $L_i(t)$, for all t ≥0, using simulation;
- Determine $s_{i+1}(t)$ as follows:

$$s_{i+1}(t) = \arg\min\left\{c \in \mathbb{N} : P\left\{L_i(t) \geq c\right\} < \alpha\right\}, \quad 0 \leq t \leq T \ ; \qquad (1.1)$$

- Check stopping condition:
If $\left\|s_{i+1}(\cdot) - s_i(\cdot)\right\|_\infty \leq \tau$, then $s_{i+1}(\cdot)$ is the proposed staffing level;
Otherwise, advance to the next iteration.

Denote by ∞ the last iteration of the algorithm (iteration of stopping). Then, the algorithm converges to a staffing function $s_\infty(\cdot)$ for which $P\left\{L_\infty(t) \geq s_\infty(t)\right\} \approx \alpha$, $0 \leq t \leq T$.

Note: In this paper we use τ=1.

2.2. Performance Measures – throughout this paper we present several performance measures. Their definition and method of calculations will now be described. Most measures are time-varying. We define them for each time-interval t, and graph their values as function over $t \in [0,T]$. Other measures are global. They are calculated either as total counts (e.g. fraction abandoning during [0,T]), or via time-averages.

Note: We use T=24 in our simulations.

Delay probability *in interval t*, calculated by the fraction of customers who are not served immediately upon arrival, out of all arriving customers during the t time-interval. Namely, for a single replication, it is given by:

$$\alpha(t) = \frac{\sum_i 1\{customer\_i\_enterd\_queue\_at\_interval\_t\}}{\sum_i 1\{customer\_i\_enterd\_system\_at\_interval\_t\}} \qquad (1.2)$$

We display $\alpha(t)$ that is averaged over all replications.

Average waiting time *in interval t*, calculated by the average waiting time of all customers arriving during the t time-interval.

$$W(t) = \frac{\sum_i w_i 1\{customer\_i\_entered\_system\_at\_interval\_t\}}{\sum_i 1\{customer\_i\_entered\_system\_at\_interval\_t\}}, \qquad (1.3)$$

where w_i is the total waiting time of customer i. Again we display $W(t)$ as averaged over all replications.

Average queue length *in interval t,* taken constant over the time-interval (see page 3). The queue length is averaged over all replications.

Tail probability *in interval t*, calculated as the probability that queue size equals or exceeds 5. Specifically, the indicators $1\{L_\infty(t) - s_\infty(t) \geq 5\}$ are averaged over all replications.

Servers' Utilization *in interval t*, calculated as the fraction of busy-servers during a time-interval (accounting for servers who are busy only a fraction of the interval):

$$\rho(t) = \frac{\sum_{i=1}^{s_\infty(t)} b_i}{s_\infty(t) \cdot \Delta} \qquad (1.4)$$

Where b_i denotes the busy time of server i in interval t. The values are again averaged over all replications.

Service grade *in interval t.* In addition to the above measures, we present another parameter called the service grade β, which arises from the following "Square-Root Safety Staffing" representation:

$$N = R + \beta \cdot \sqrt{R} \qquad (1.5)$$

In this formula, R is the *offered load*, measured in units of service-time that arrive to the system per unit-of-time. The "square-root" term is safety staffing against stochastic variability.

In a time-varying context, the offered load clearly varies with time, and we shall denote it by R(t), $0 \le t \le T$. But, less obviously, it is defined in terms of an auxiliary infinite-server system, in which the arrival process and service times coincide with our original system (see Jennings et. al.). More precisely, R(t) it is defined to be the average number of customers (equivalently, average number of busy servers) at time t, for that corresponding infinite-server system. Then, the implied service grade β in all our examples was retrieved by

$$\beta(t) = \frac{s_\infty(t) - R(t)}{\sqrt{R(t)}} \quad , \; 0 \le t \le T, \qquad (1.6)$$

where $s_\infty(t)$ and R(t) are, respectively, the staffing level arrived at by the algorithm and the offered load, both evaluated at time t (interval t).

3. The "Challenging Example"

In this section, we explore the "challenging example" presented in Jennings, Mandelbaum, Massey & Whitt (2001). This is an $M_t/M/s_t$ system, with exponential service times having mean 1, and a non-homogenous Poisson arrival process with the arrival rate function $\lambda(t) = 30 + 20 \cdot \sin(5 \cdot t)$.

As already mentioned, our algorithm is given as input a desired (target) *delay probability*. In order to examine the performance of the algorithm, we applied it with target delay probabilities $\alpha = 0.1, 0.2\ldots, 0.9$.
For each target α from above, we calculated the corresponding performance measures.
We now show summary results graphs for all target delay probabilities

Figure 9: Delay probability summary for the Challenging example

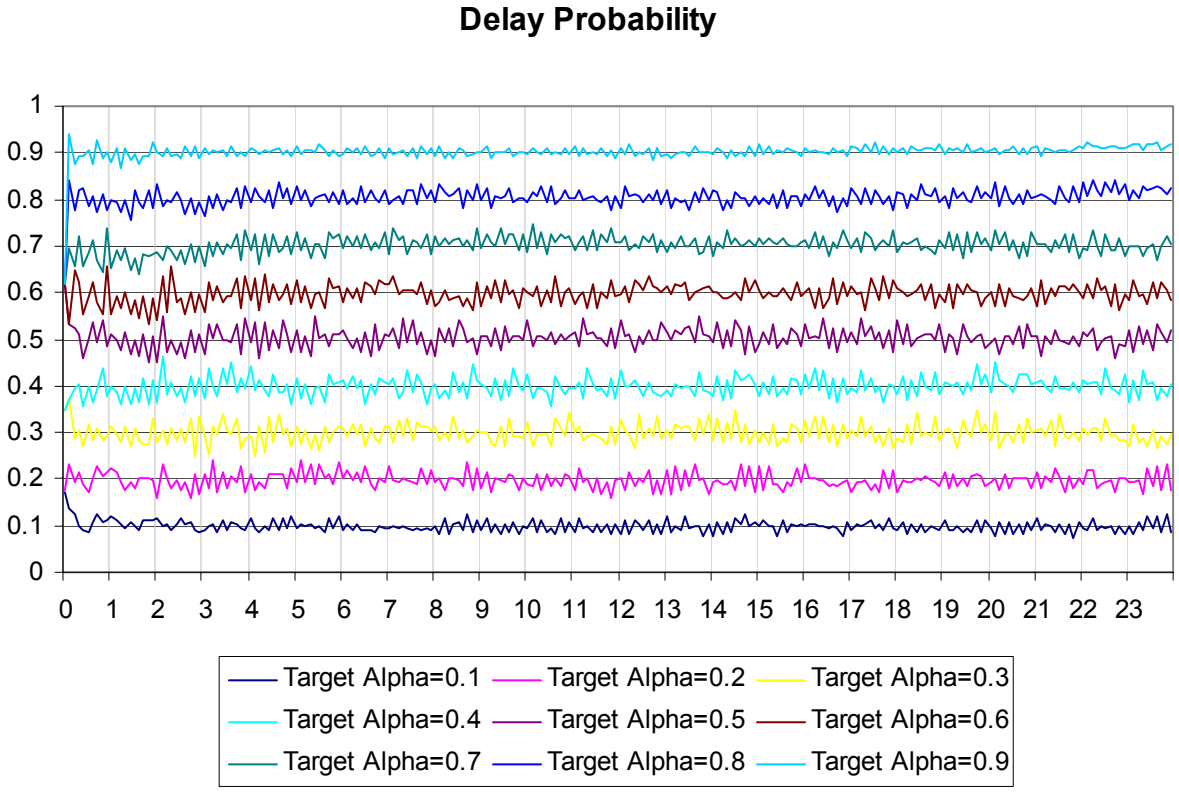


Figure 10: Implied service grade β summary for the Challenging example

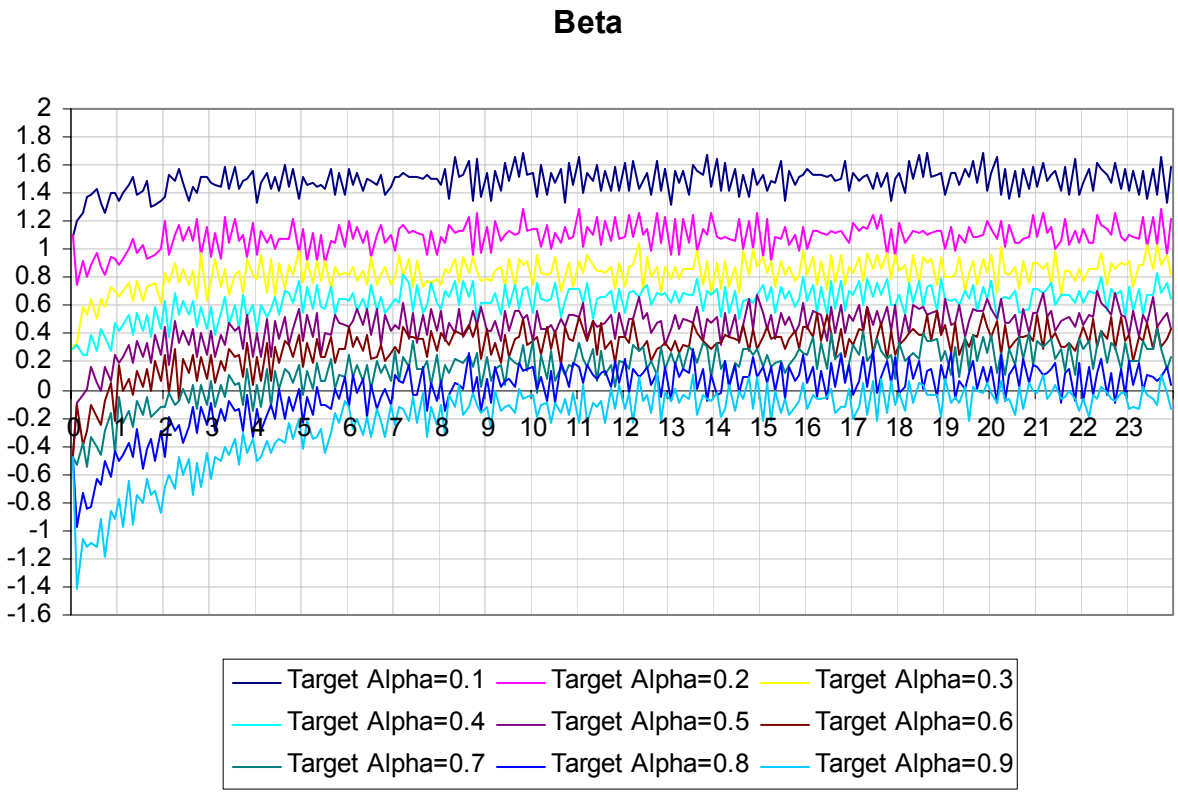


Figure 11: Utilization summary for the Challenging example

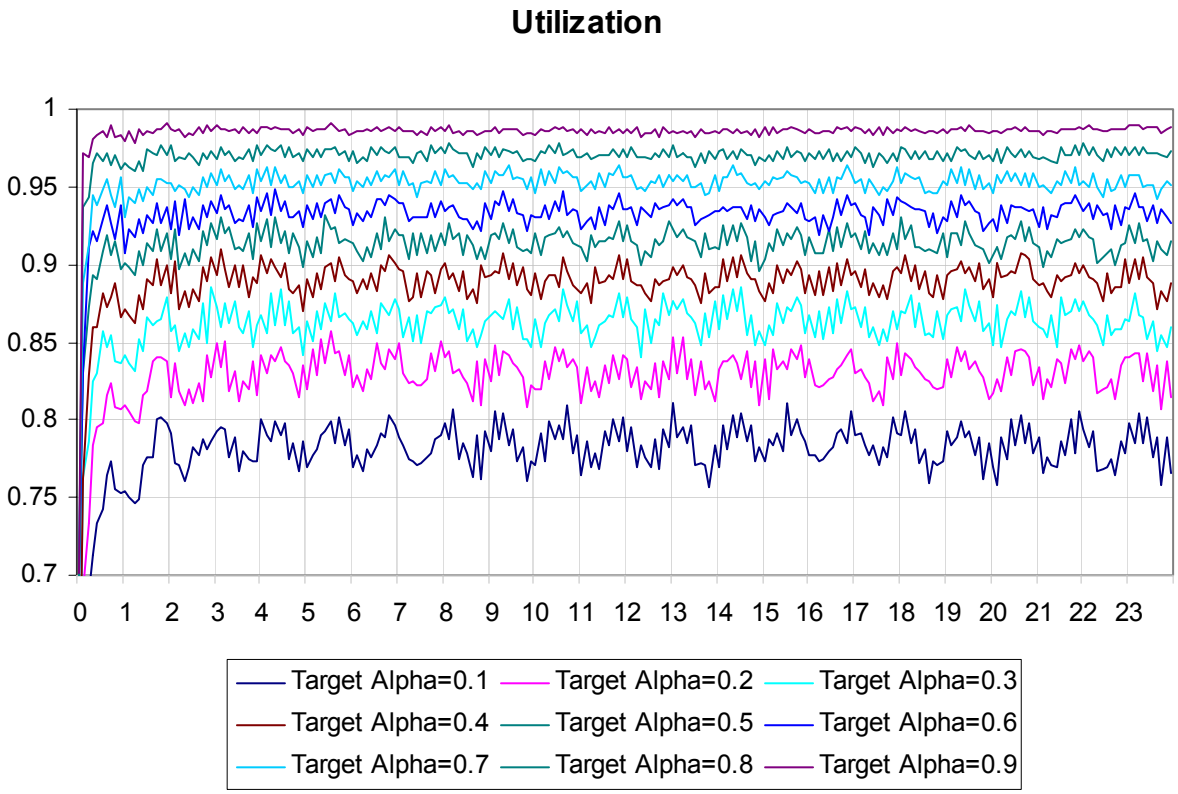
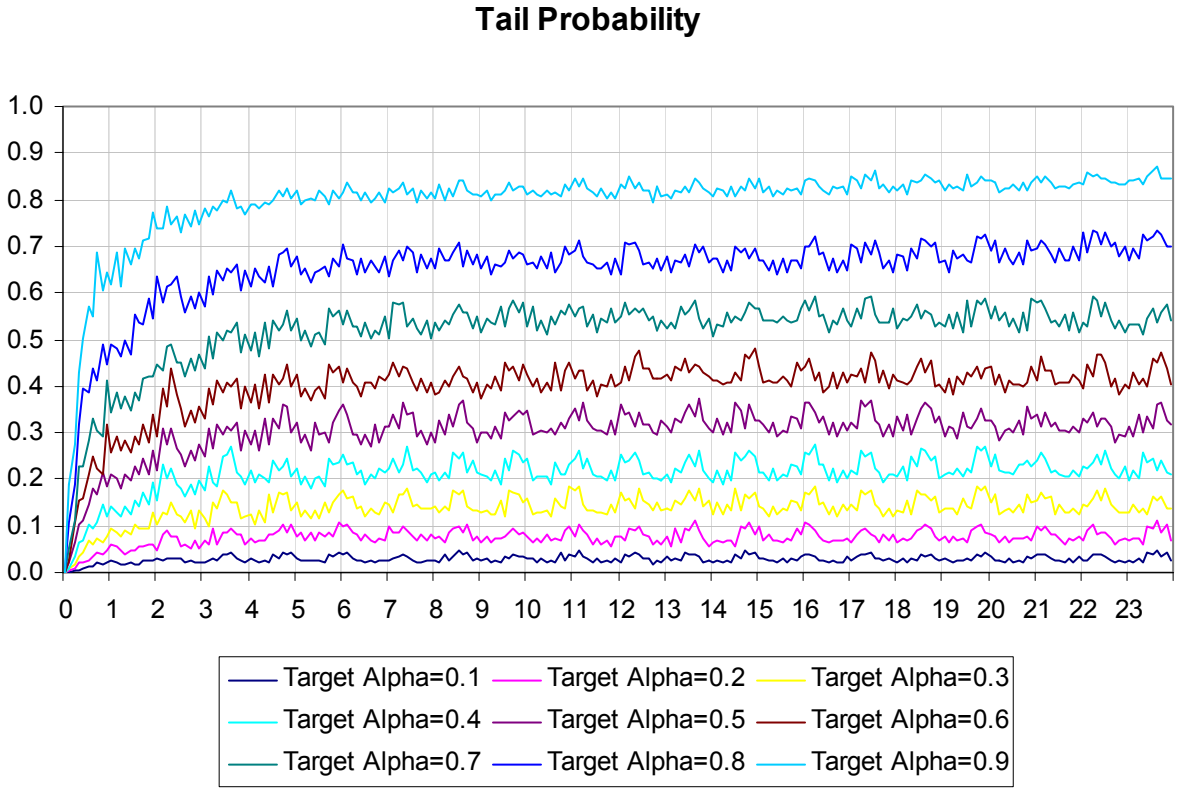


Figure 12: Tail probability summary for the Challenging example



We chose a set of 3 representing target α 's – 0.1, 0.5, 0.9 to demonstrate further performance behavior. (In some contexts, the set above can be associated with regimes of operation)

Figure 13: Target α=0.1: (1) Staffing level, offered load and arrival function, (2) Average queue and waiting time, (3) Waiting time given wait histogram

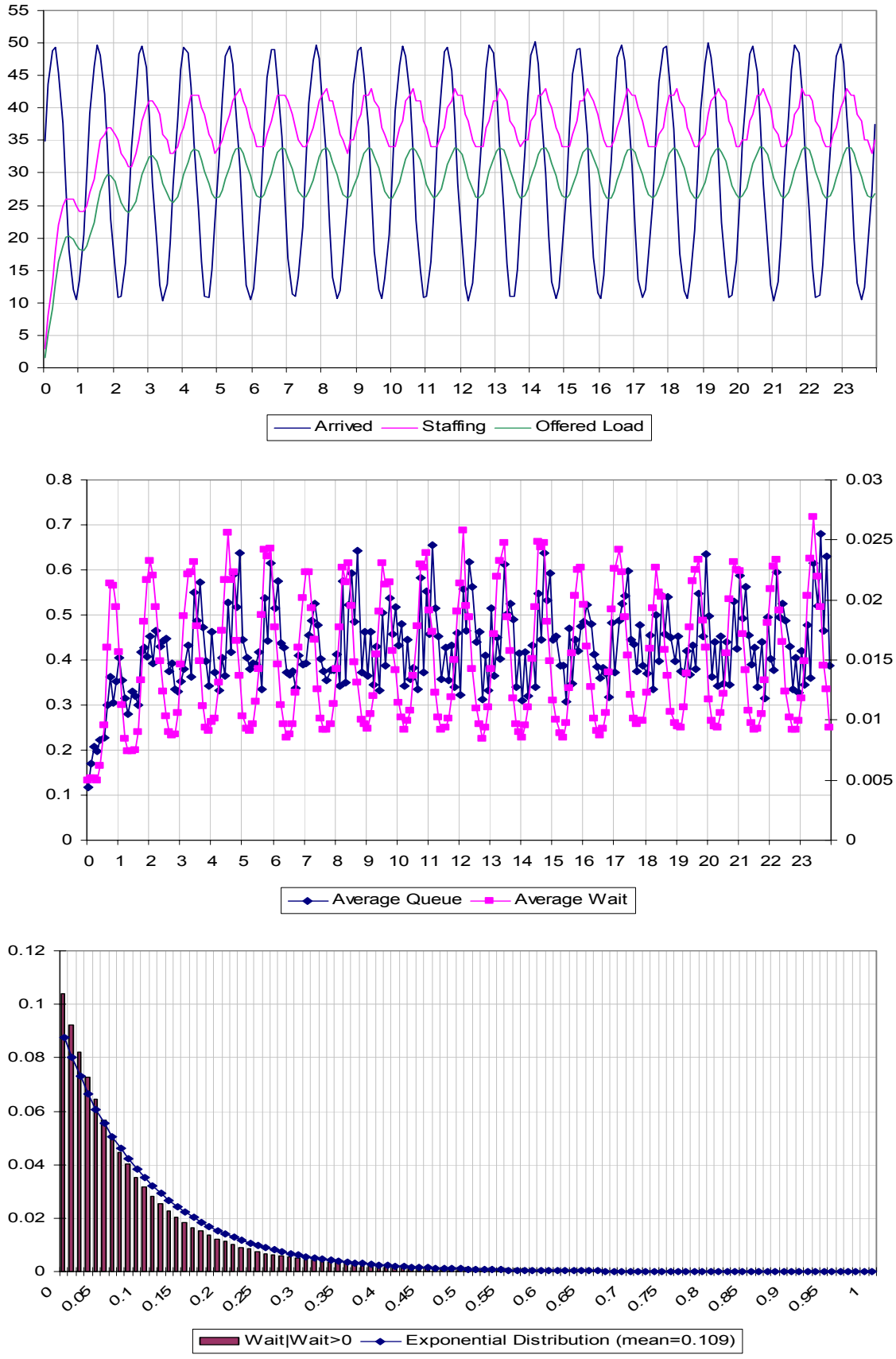
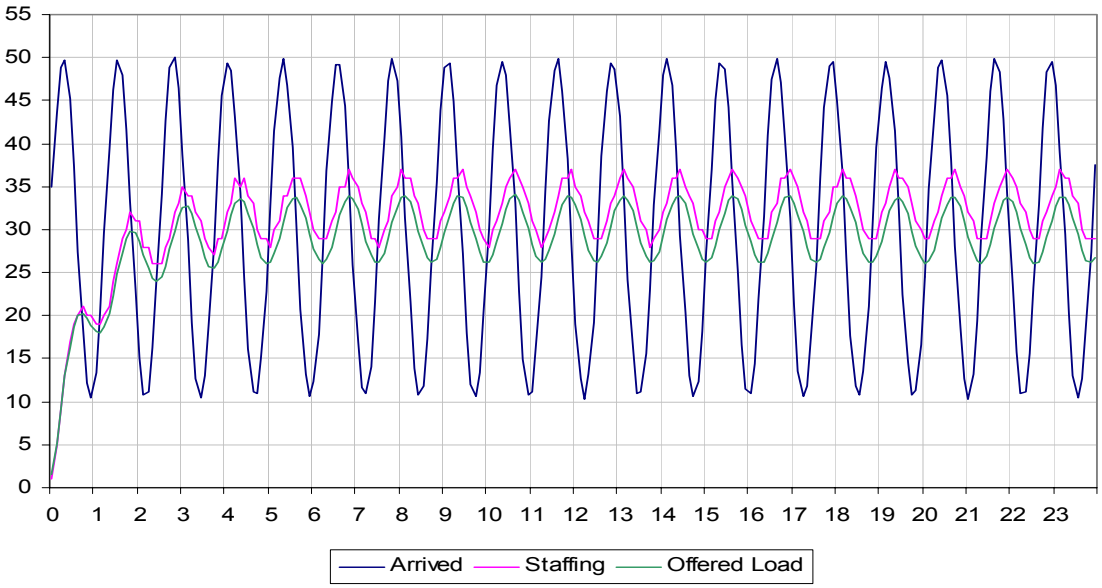
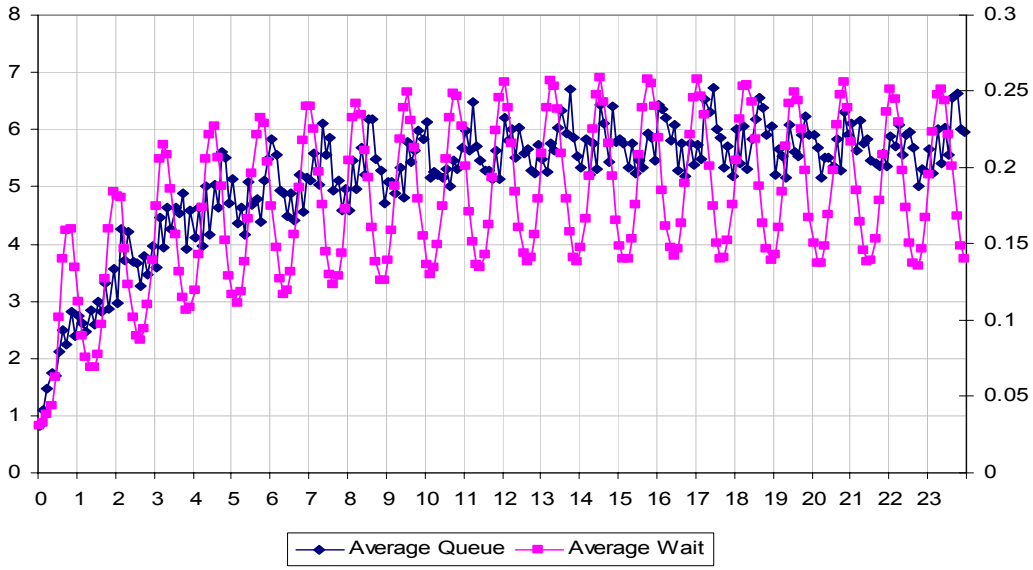


Figure 14: Target α=0.5: (1) Staffing level, offered load and arrival function, (2) average queue and waiting time, (3) Waiting time given wait histogram





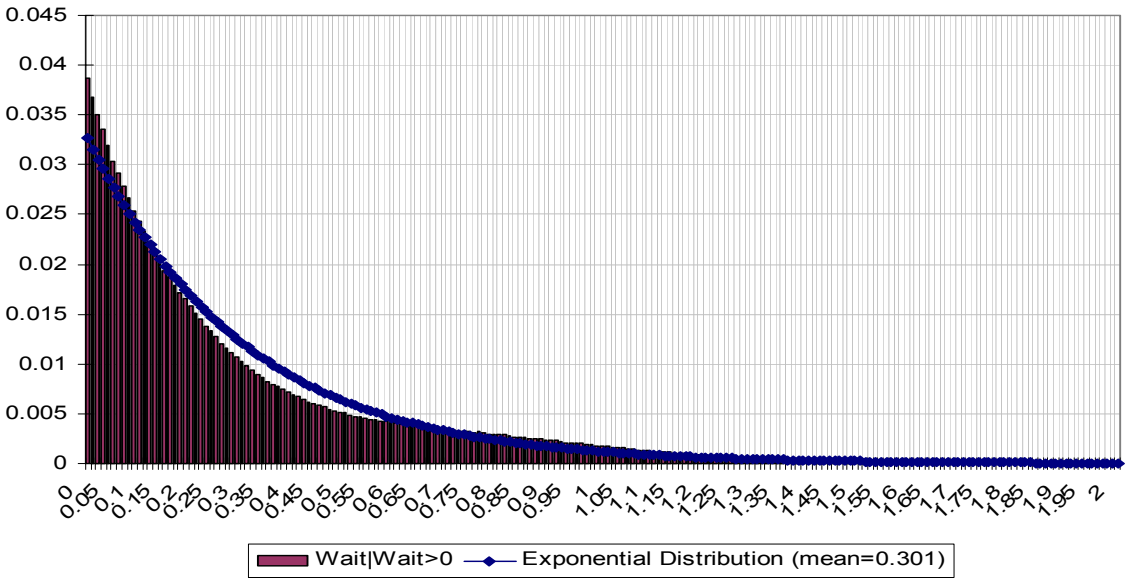
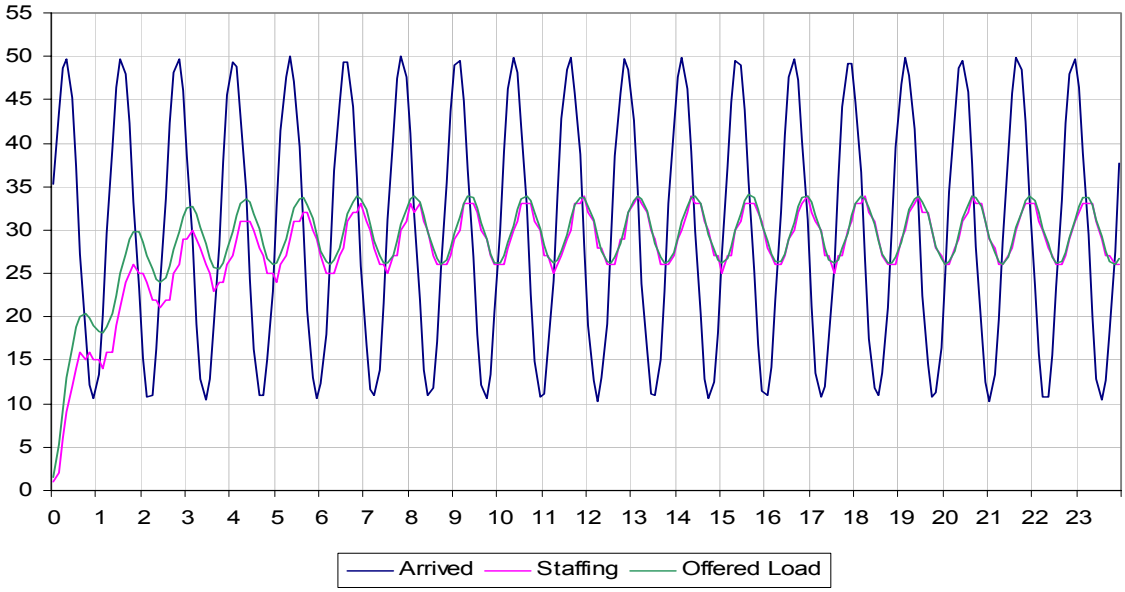
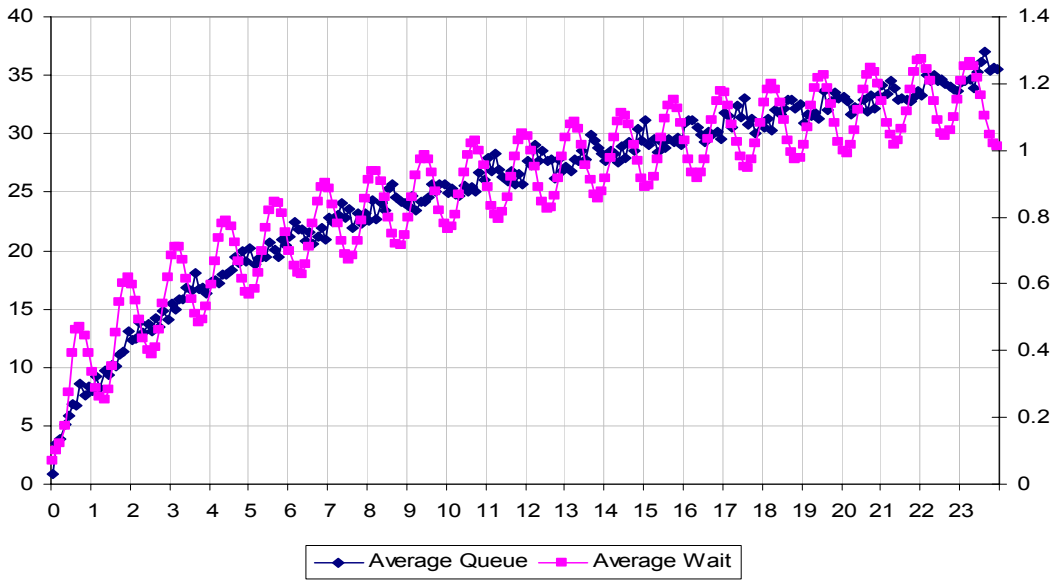
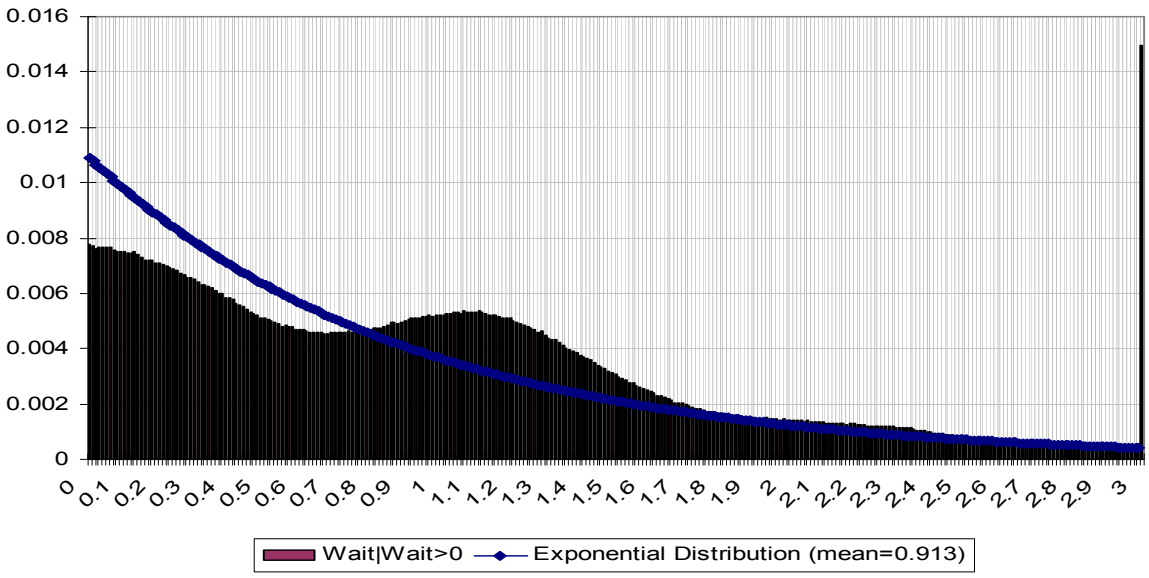


Figure 15: Target α=0.9: (1) Staffing level, offered load and arrival function, (2) average queue and waiting time, (3) Waiting time given wait histogram



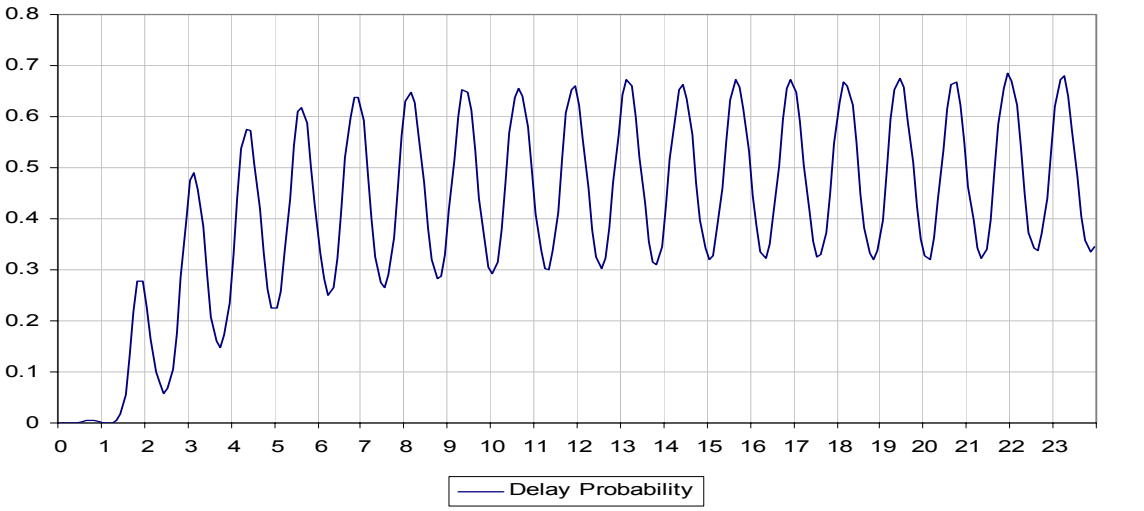


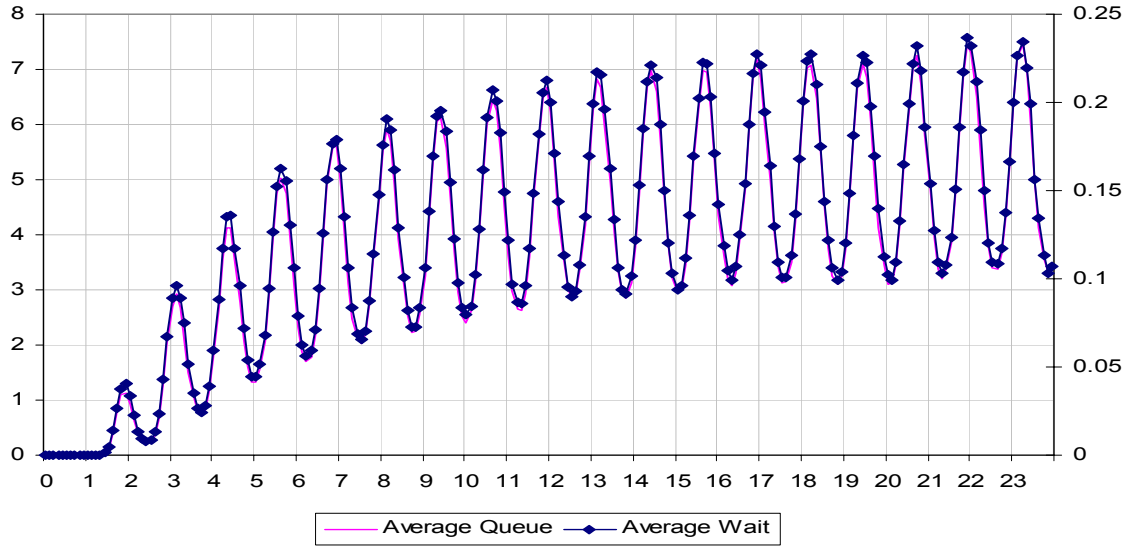


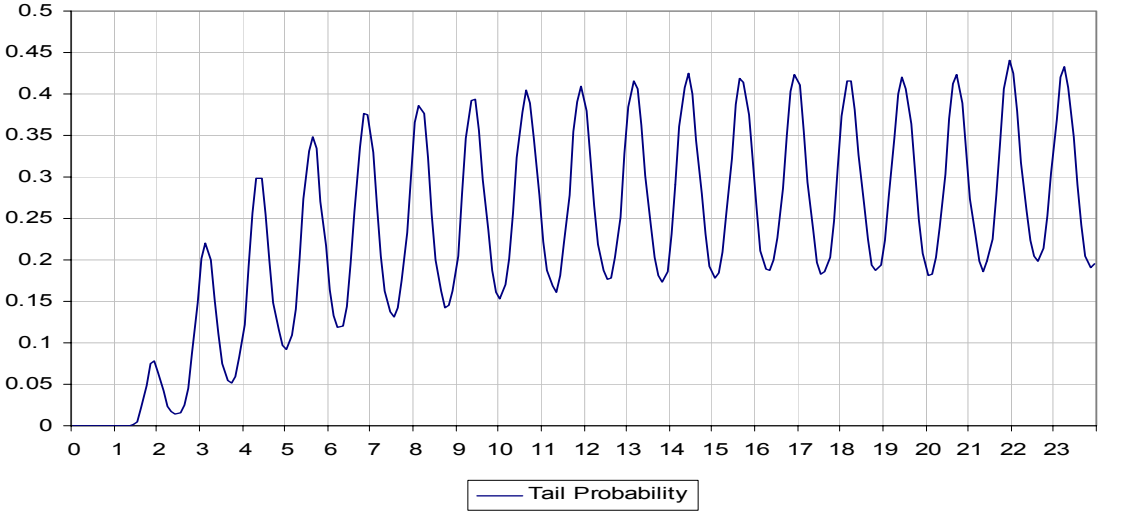
In (3) of figures 5, 6, 7 we plot the empirical distribution of waiting time given wait (i.e. the waiting time of those who were in fact delayed). Recall that for the *stationary* M/M/s queue $W_q \mid W_q > 0$ is exponentially distributed. As seen above, the empirical distribution of wait given wait, in our *time-varying* queue and over *all* customers, also fits the exponential distribution rather well (the lack of similarity for the case of target α=0.9 occurs mainly due to the fact that measures were taken when system had not reached steady state) .
The mean of the exponential distribution was taken to be the overall average waiting time of those who were actually delayed during [0,T].

For comparison, we now present performance measures under the SSA approximation. By this we mean a *constant* staffing level that arises in a naturally corresponding stationary system (i.e. average arrival rate). It suffices to take a representative case in order to notice the very poor and unstable performance. We show the case of target $\alpha = 0.5$.

Figure 16: The SSA approx. (target $\alpha = 0.5$): (1) Delay probability , (2) average queue and average waiting time, (3) tail probability







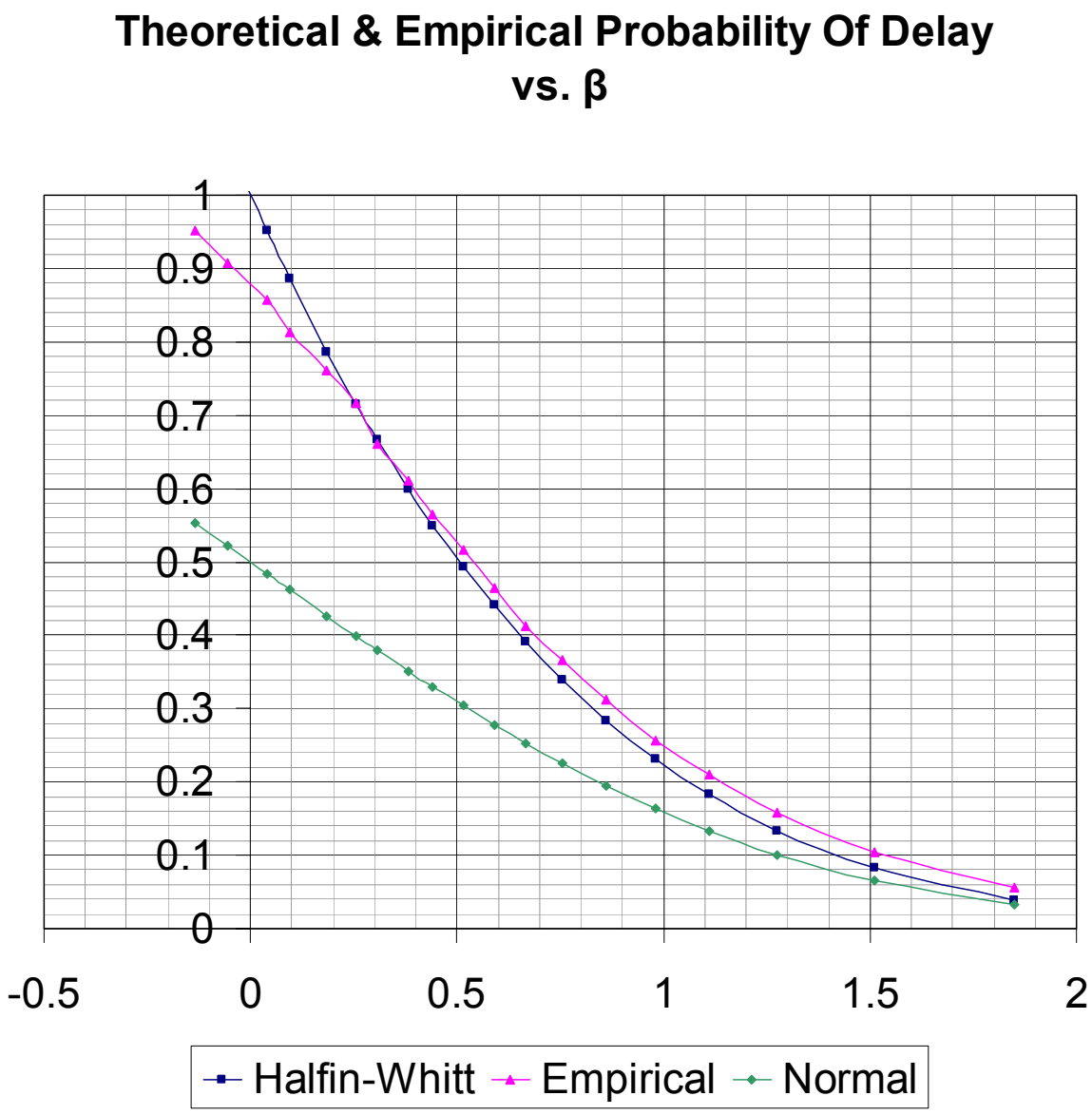
When forcing the probability of delay to be constant in time, all other performance measures tend to be stable over all times as well, including the service grade β defined previously. Thus, it seems reasonable to treat α - the probability of delay, and β - the service grade, as constants over all time. One is then lead naturally to a comparison with the *Halfin-Whitt* function, given by

$$P\{delay\} = \alpha = \left[1 + \beta \cdot \frac{\phi(\beta)}{\varphi(\beta)}\right]^{-1}. \qquad (2.1)$$

where $\varphi(\cdot)$ and $\phi(\cdot)$ are the density function and the distribution function of the standard Normal distribution.

We plotted the pairs of (α_i, β_i) along-side the Halfin-Whitt function. We also added $\bar{\phi}(\cdot)$, the function defined by $1-\phi(\cdot)$, which comes out of the pure infinite-server approximations. Here, α_i was calculated as the fraction of all the delayed customers, out of all arriving customers in $[\tau, T]$ (starting from time τ - the time at which stability was reached); β_i was calculated as the average β, over all times starting in $[\tau, T]$.

Figure 17: Comparison of empirical results with the Halfin-Whitt and Normal approx.



The excellent match between the empirical and theoretical results implies that our staffing procedure is in fact **redundant** for the time-varying Erlang-C model!
Indeed, one can staff this system, challenging as it may be, simply by calculating the offered load (infinite-server based) at all times, and then using the square-root safety staffing rule (1.5). Here, one chooses $\beta = f^{-1}(\alpha)$ obtained via the Halfin-Whitt function, for any target delay probability α.

Remark: The procedure exercised in Jennings et al. uses the Normal curve in the Figure above. Hence, the target delay was not achieved there precisely, and corrections were required. The correction suggested was the use of beta, exactly as above, and the results of our experiments perfectly justify it.
The results in Jennings et al are useful in our context as well: they provide simple means for calculating or approximating the offered load. This is in contrast to simulation, which we use in the present study, and which will in fact be necessary in more complicated modes (such as models with abandonment).

4. QED Example

In this section we analyze an example similar to that in the previous section: service times are exponential having mean 1, arrivals are non-homogenous Poisson with the arrival rate function $\lambda(t) = 100 + 20 \cdot \sin(t)$.

The goal of this example is to examine a larger system (around 100 servers) which, under some parameter values, operates in the QED regime.

Figure 18: Delay probability summary for the QED example

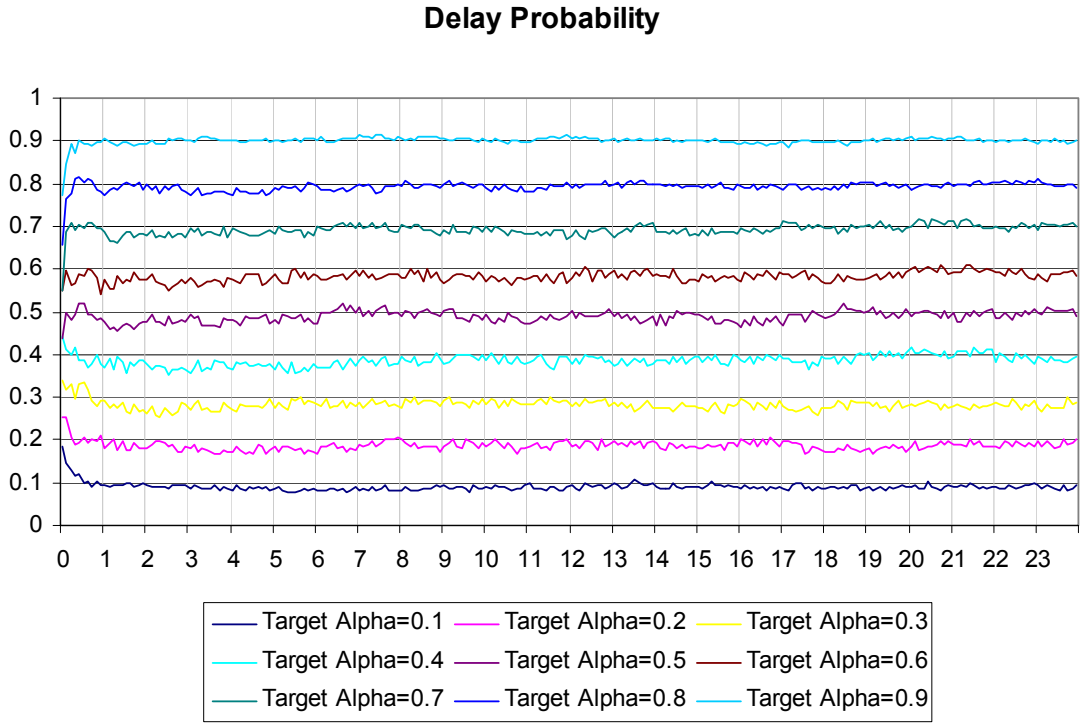


Figure 19: Implied service grade β summary for the QED example

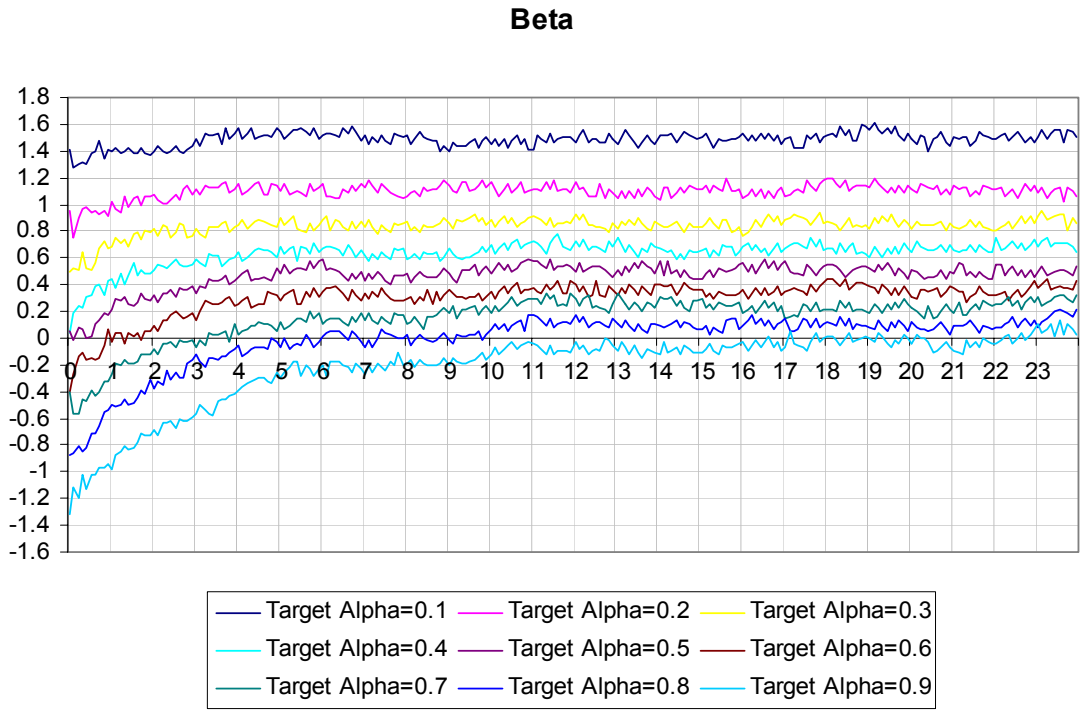
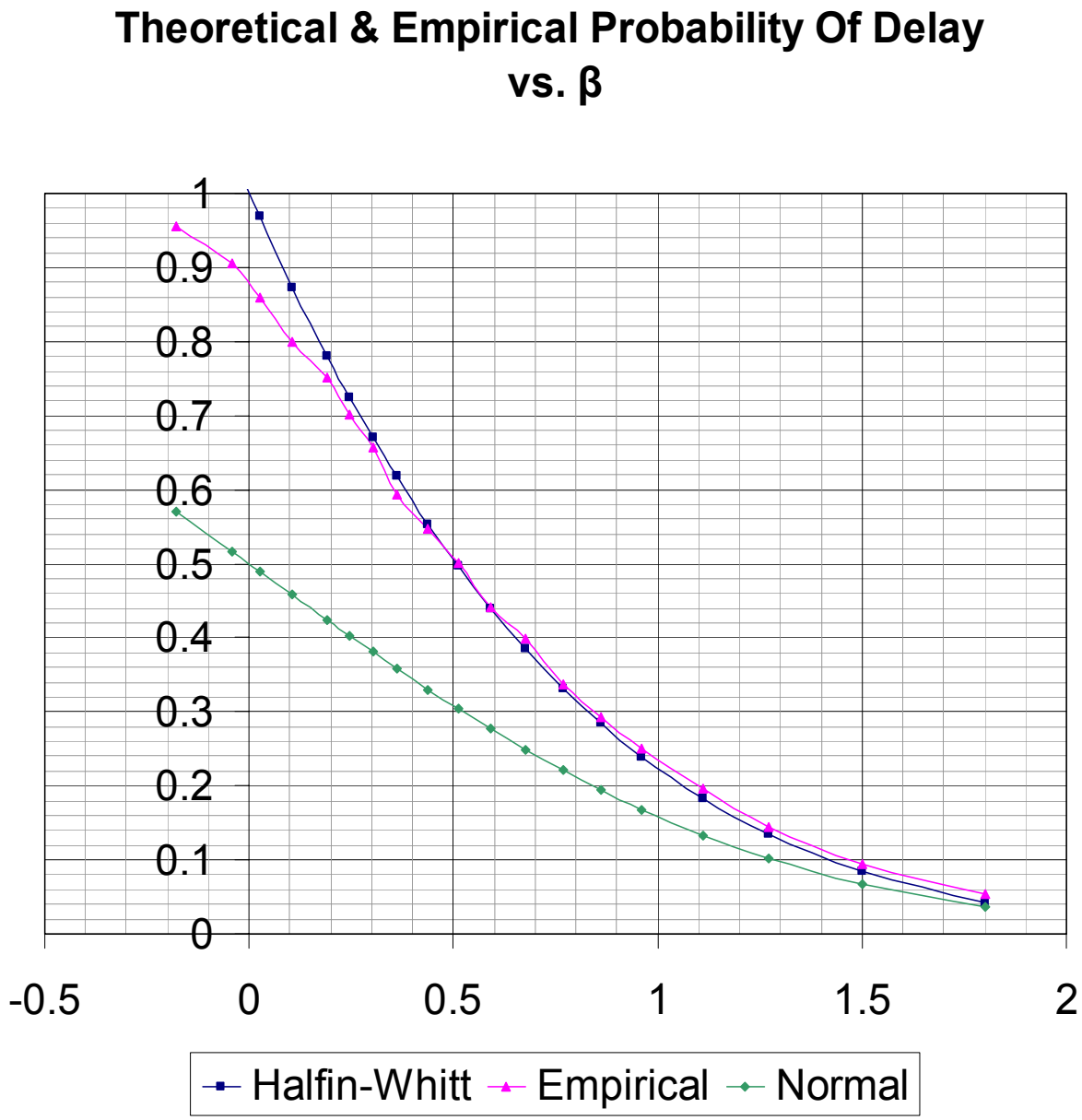


Figure 20: Comparison of empirical results with the Halfin-Whitt and Normal approx.



Note: Negative values of beta arise due to staffing levels that are below the offered load. This occurs, possibly, since steady state has not been yet reached. (See pages 5, 8 for a justification of the latter observation.)

Figure 21: Utilization summary for the QED example

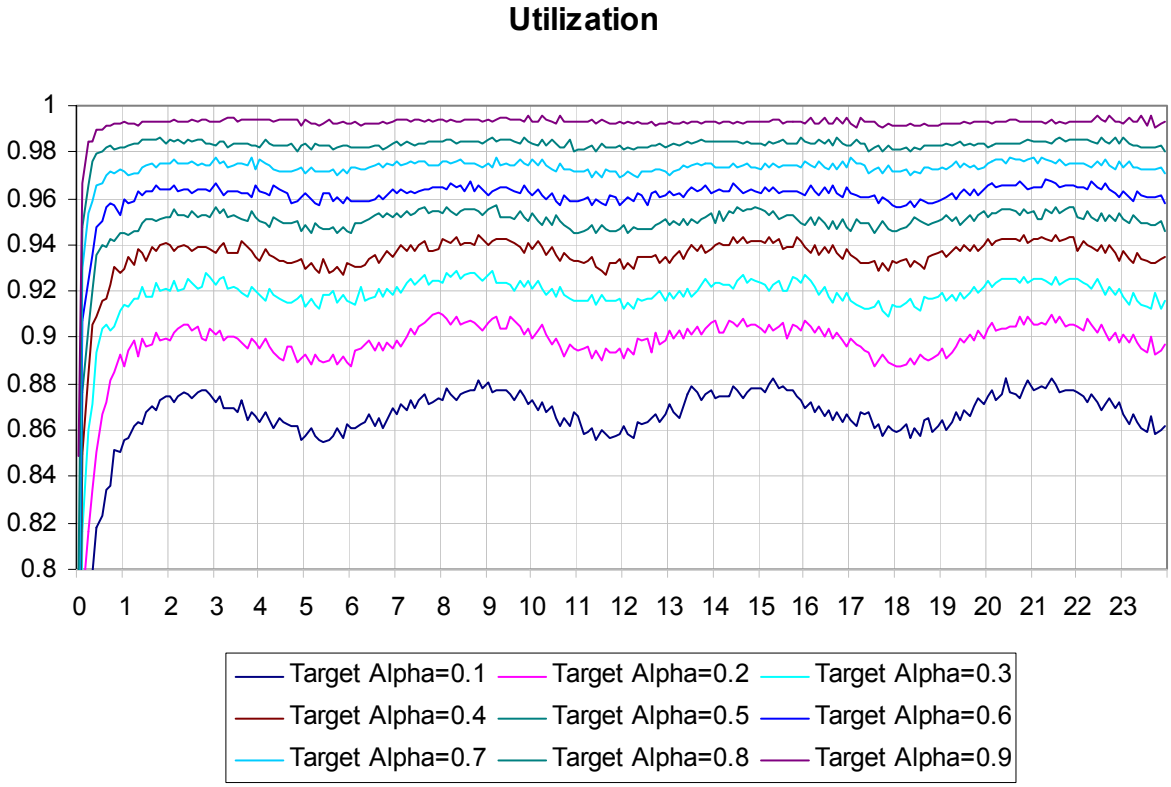


Figure 22: Tail probability summary for the QED example

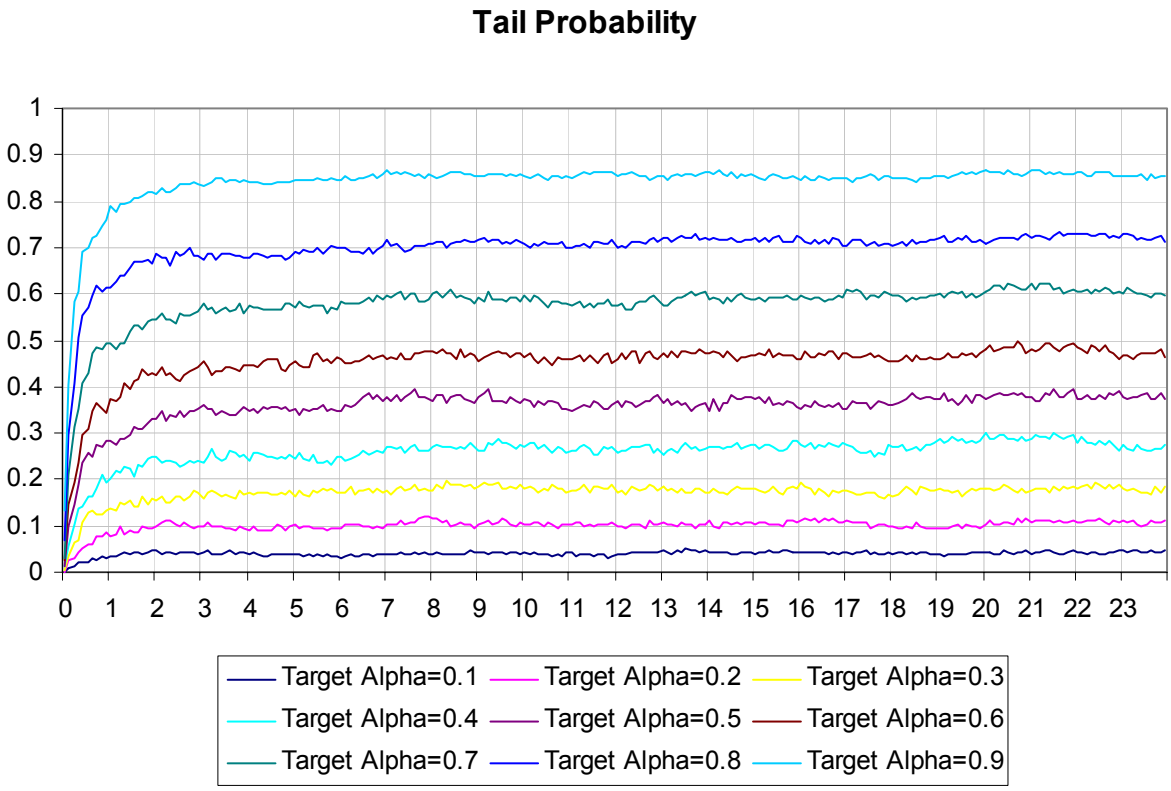
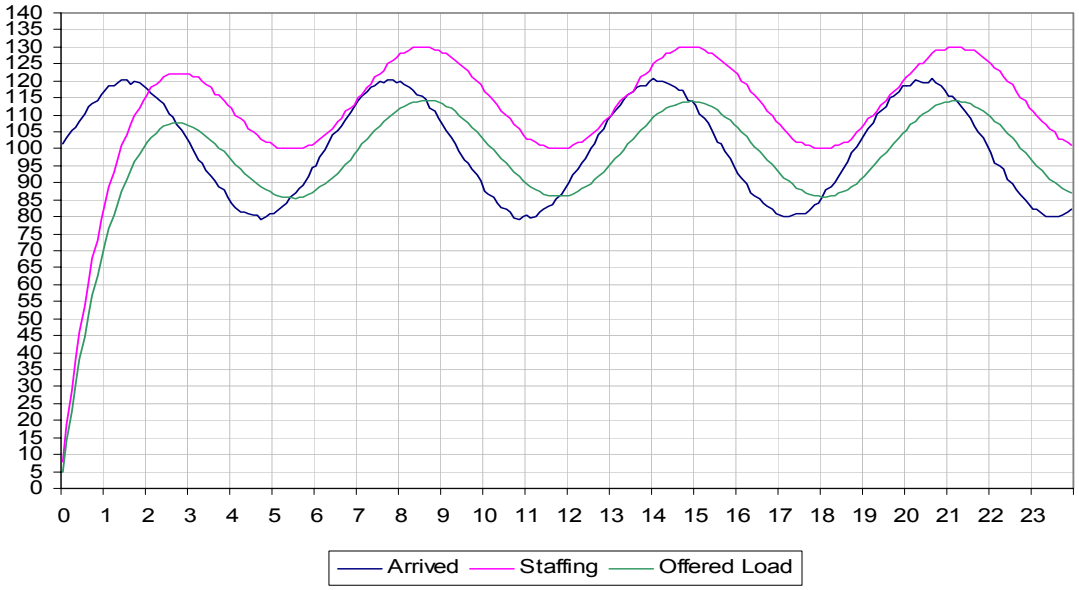
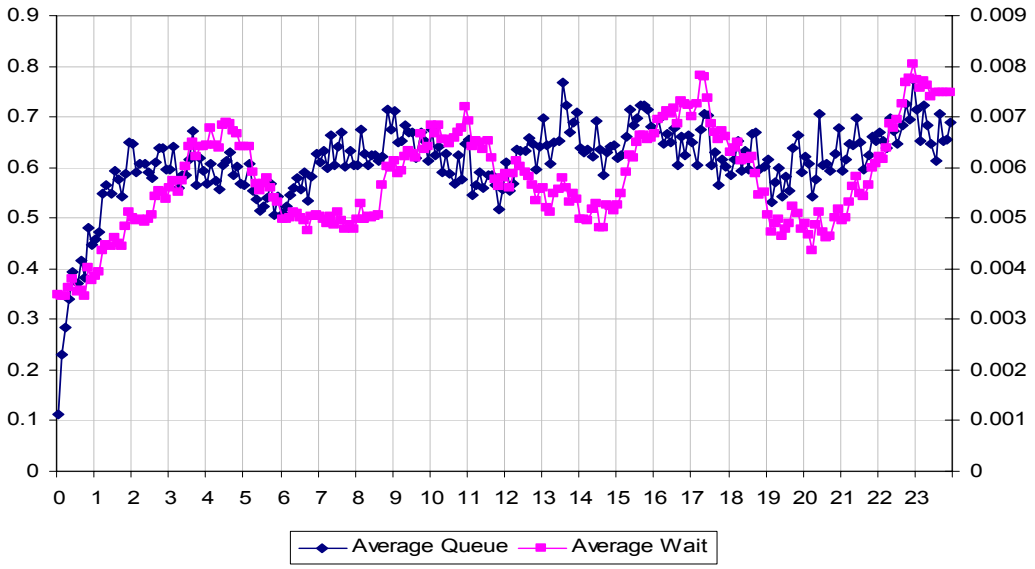


Figure 23: Target α=0.1: (1) Staffing level, offered load and arrival function, (2) average queue and waiting time, (3) Waiting time histogram of those who waited





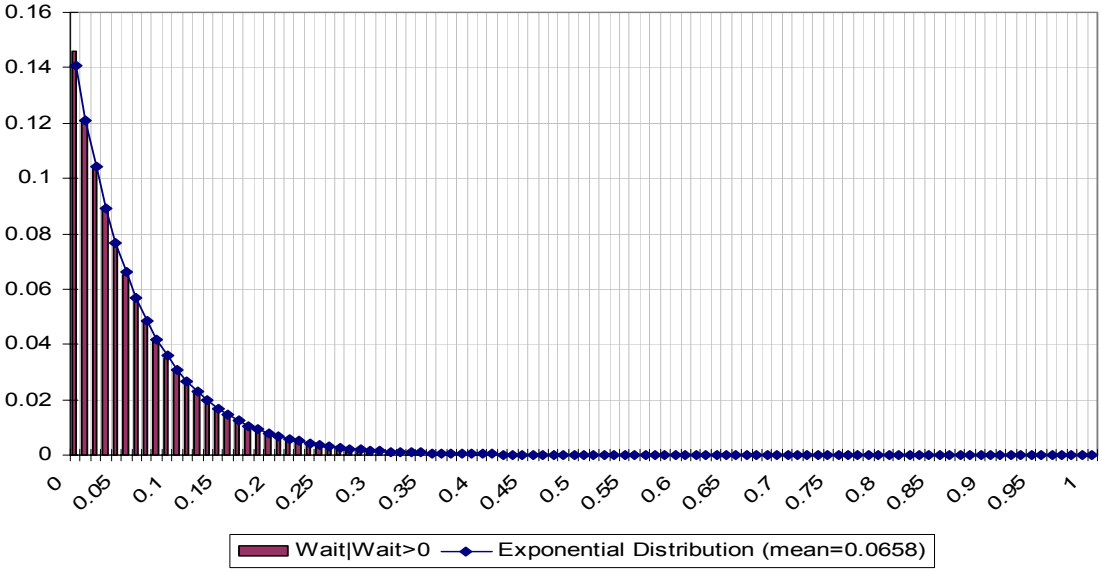
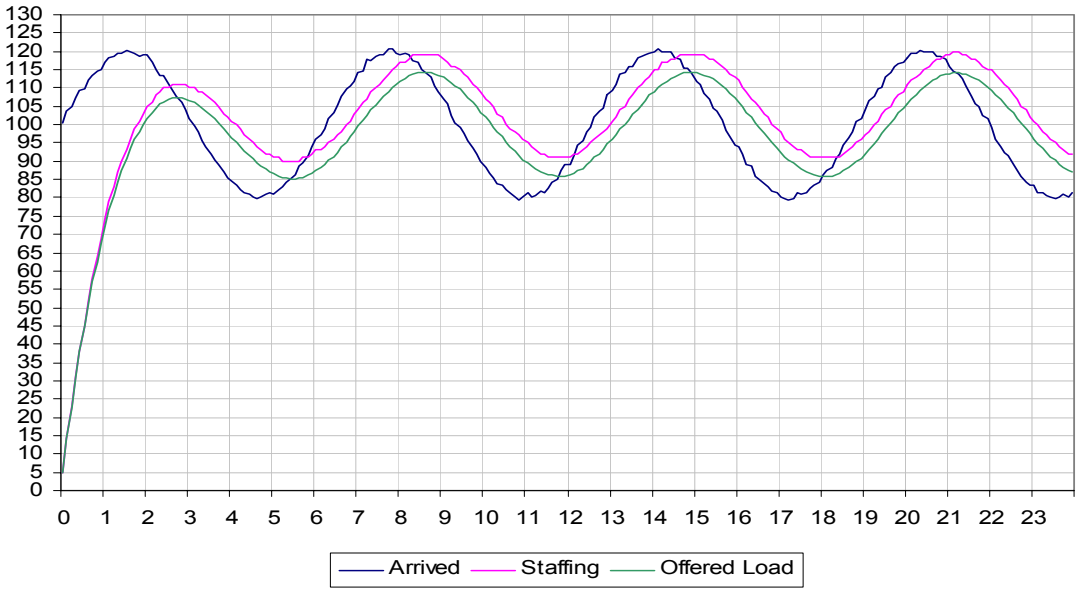
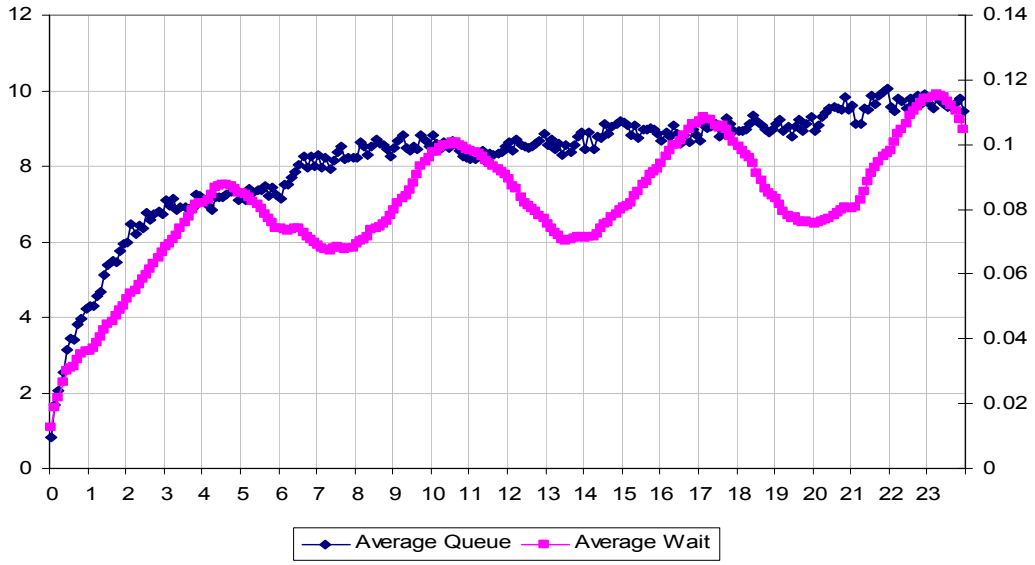


Figure 24: Target α=0.5: (1) Staffing level, offered load and arrival function, (2) average queue and waiting time, (3) Waiting time histogram of those who waited





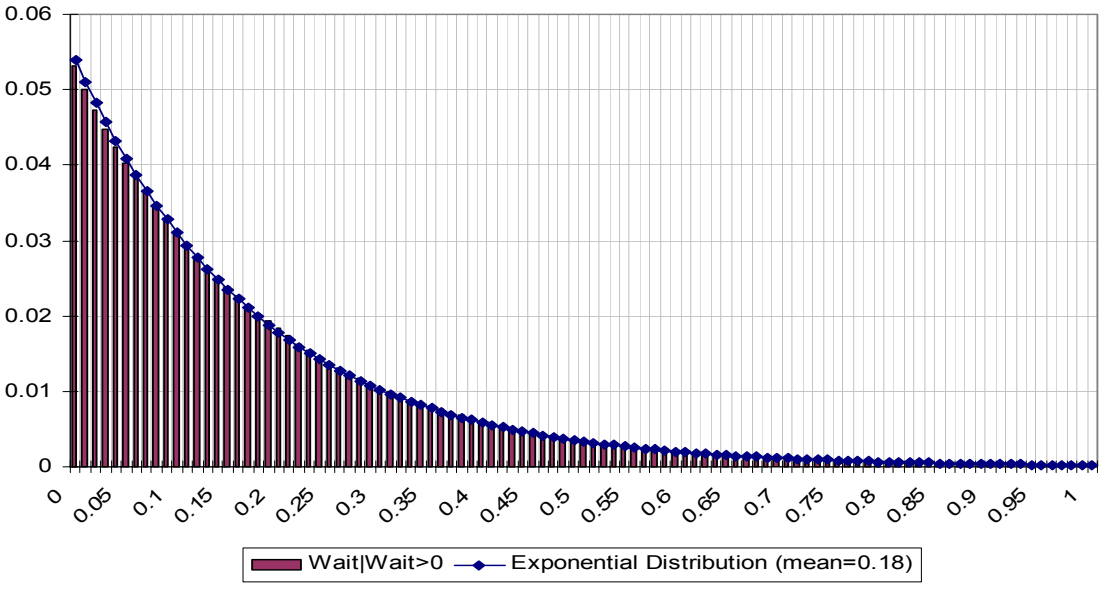
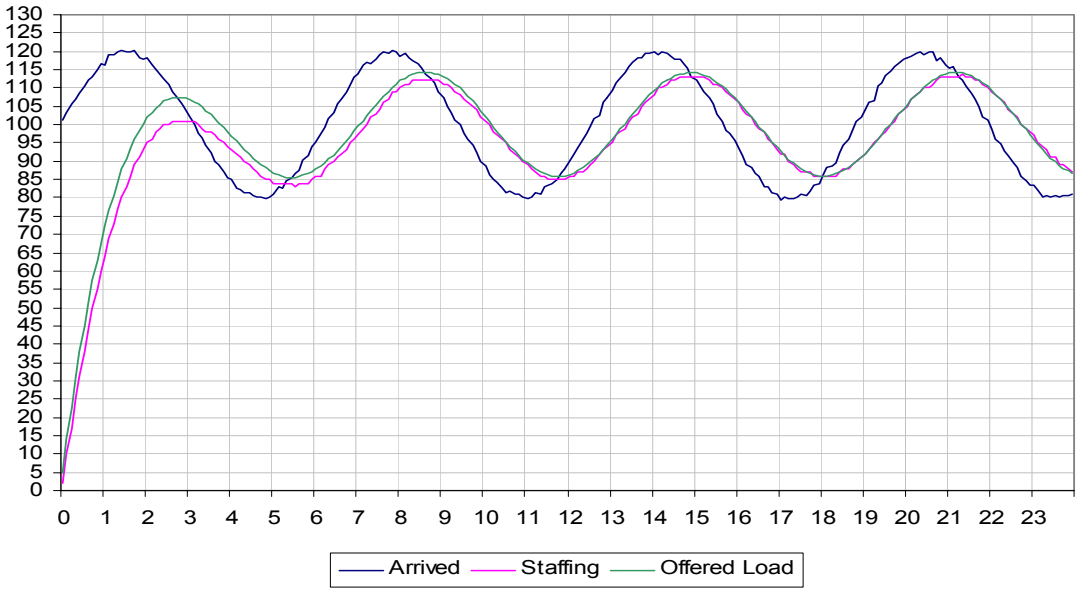
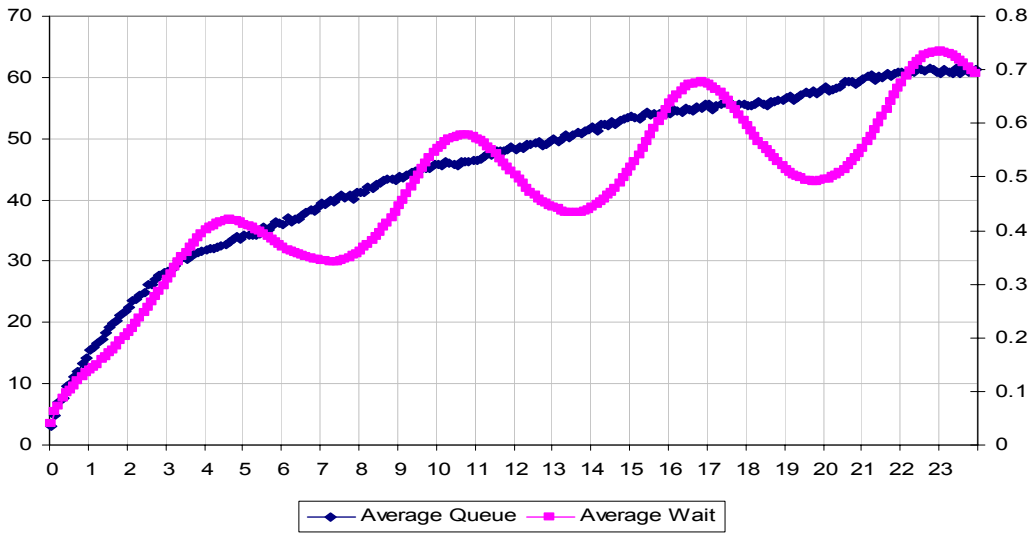
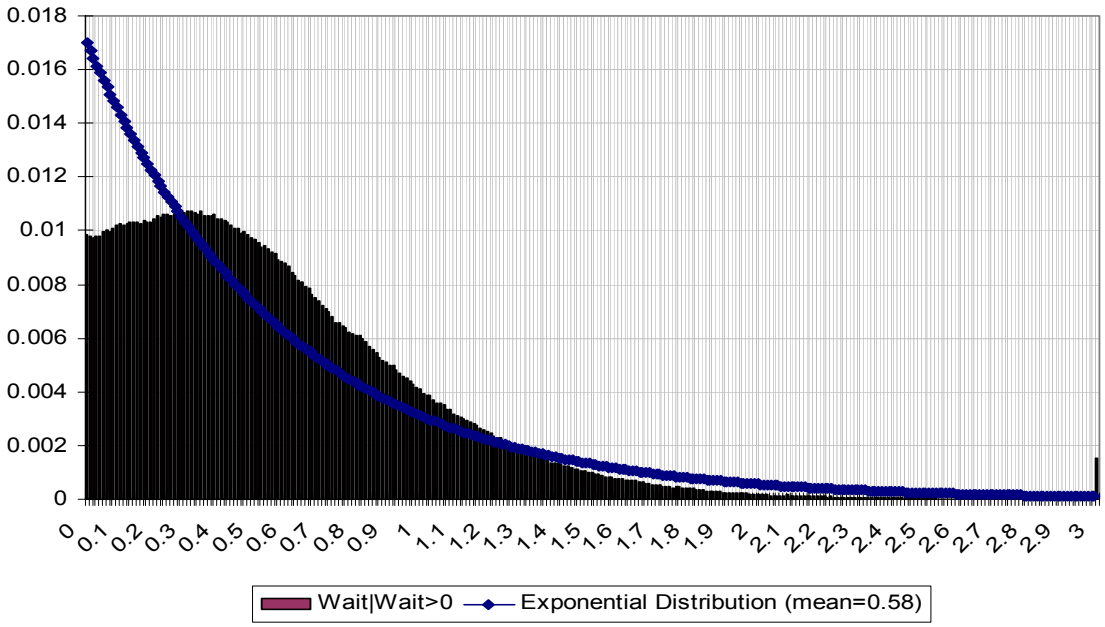


Figure 25: Target α=0.9: (1) Staffing level, offered load and arrival function, (2) average queue and waiting time, (3) Waiting time histogram of those who waited







5. An $M_t/M/s_t+M$ Queue

5.1. In this section we add abandonment (due to exponential (im)patience. Thus, we are analyzing the time-varying Palm/Erlang-A queue. The arrival function is $\lambda(t) = 100 + 20 \cdot \sin(t)$ as before, service times are exponential having mean 1, and, as in the first example, (im)patience is exponential distribution with **parameter 1 (mean 1)**.

Figure 26: Delay probability summary for the Erlang-A example

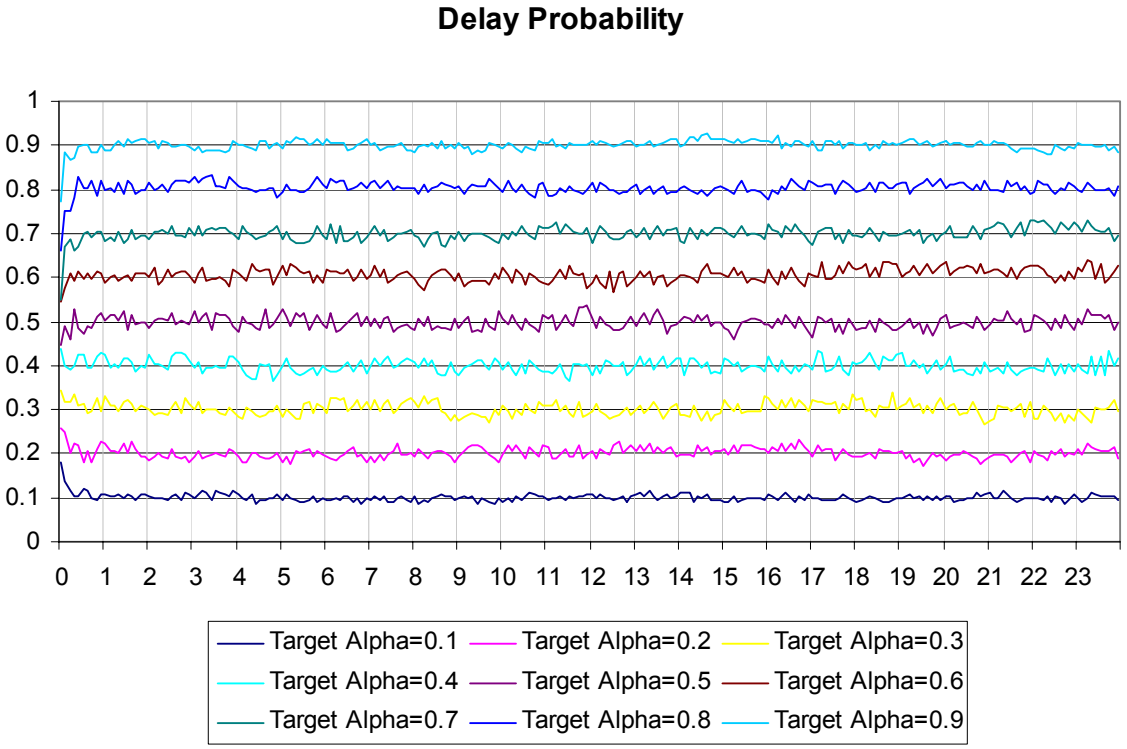


Figure 27: Implied service grade β summary for the Erlang-A example

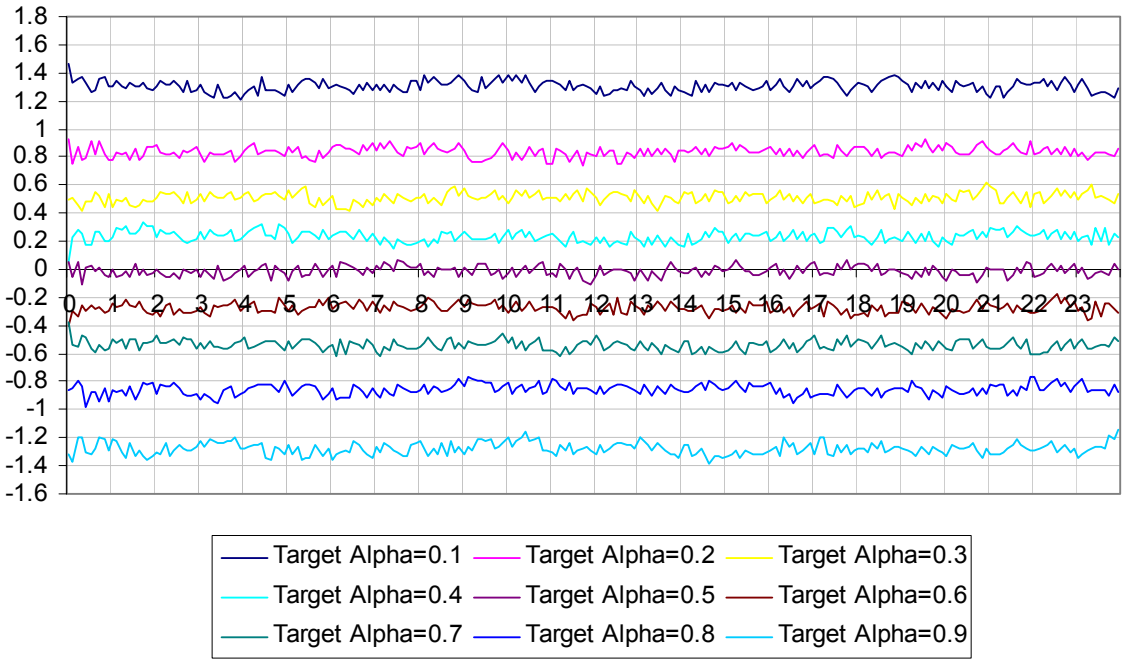


Figure 28: Abandon probability summary for the Erlang-A example

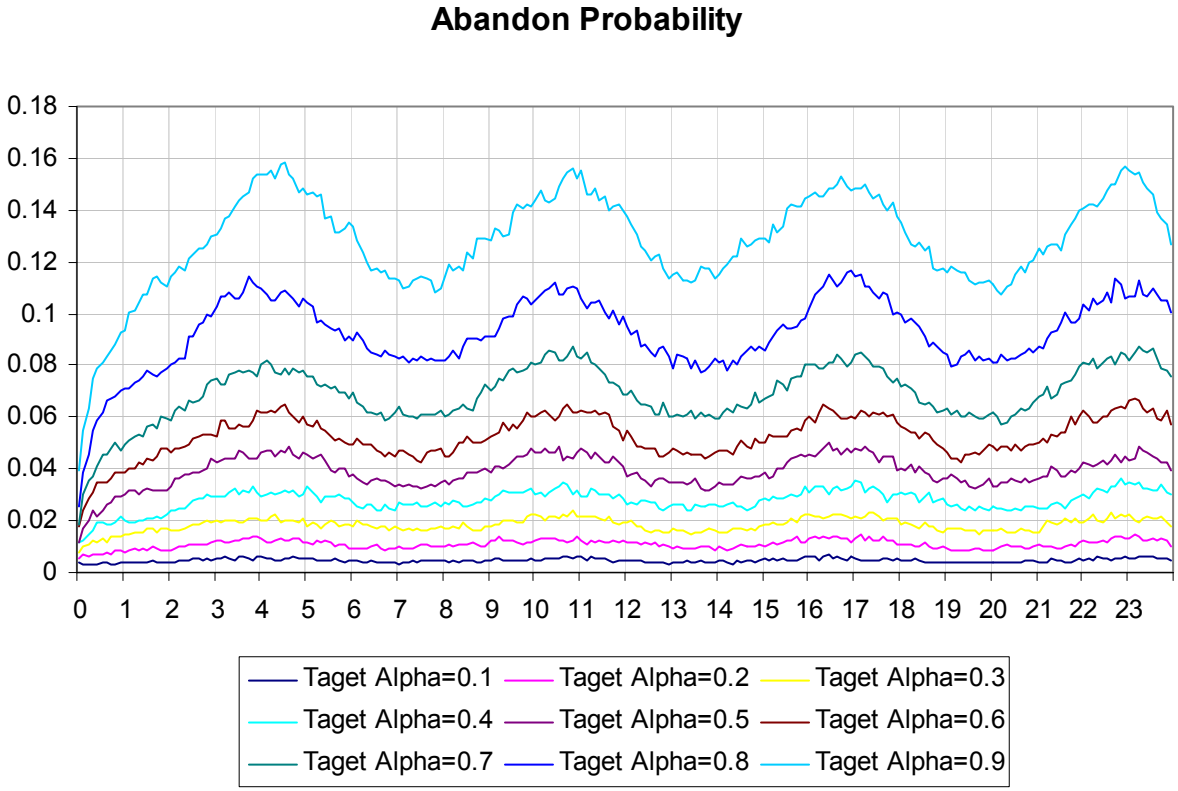


Figure 29: Utilization summary for the Erlang-A example

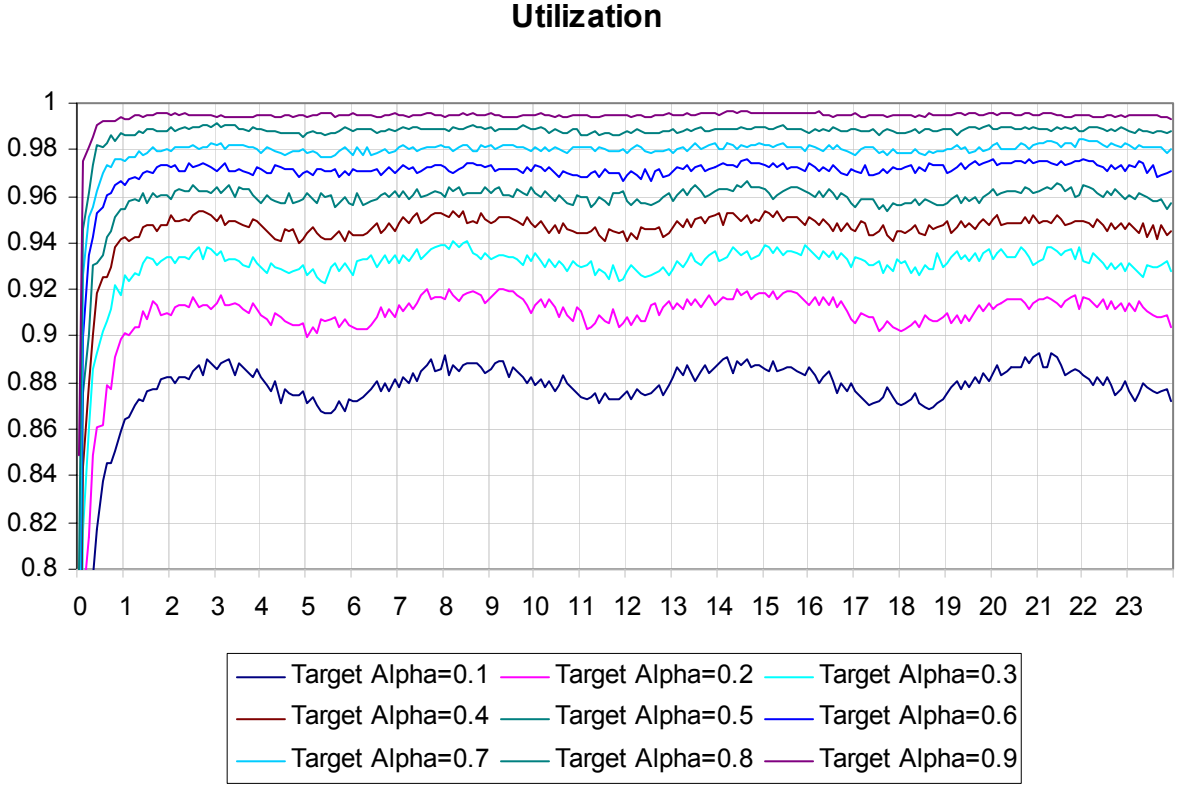


Figure 30: Tail probability summary for the Erlang_A example

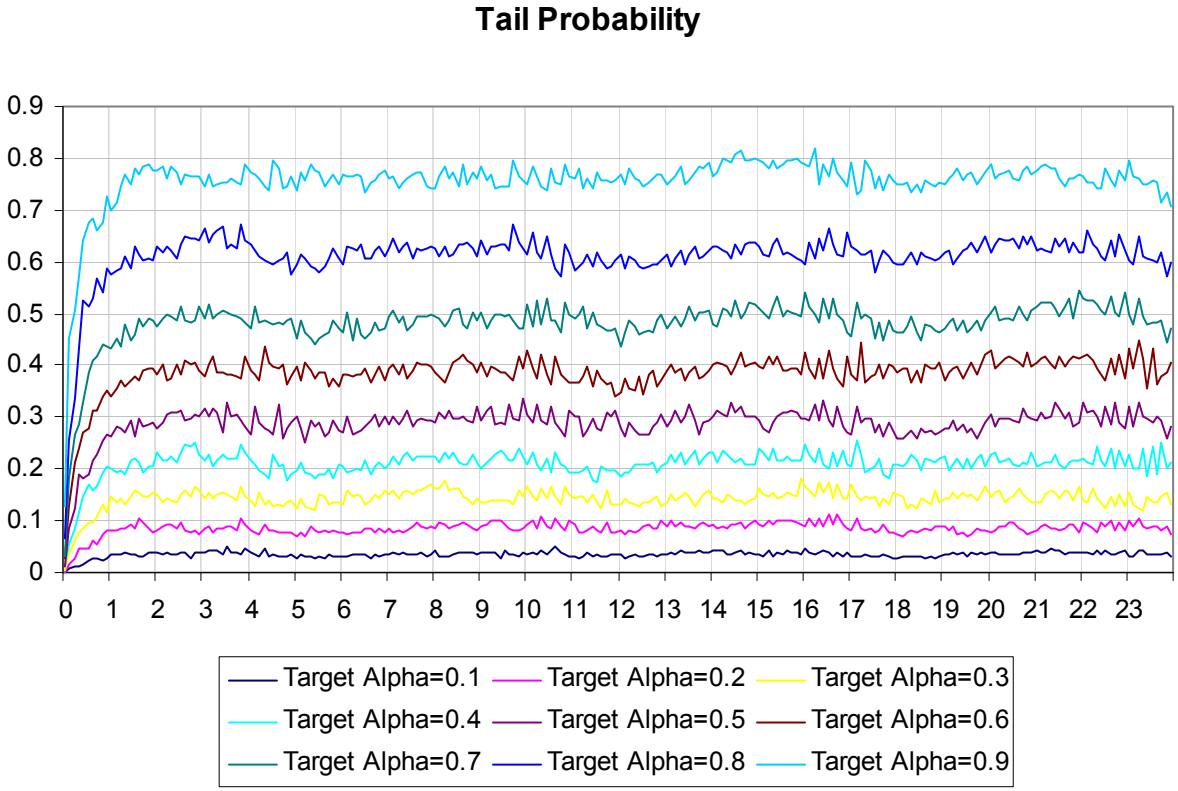
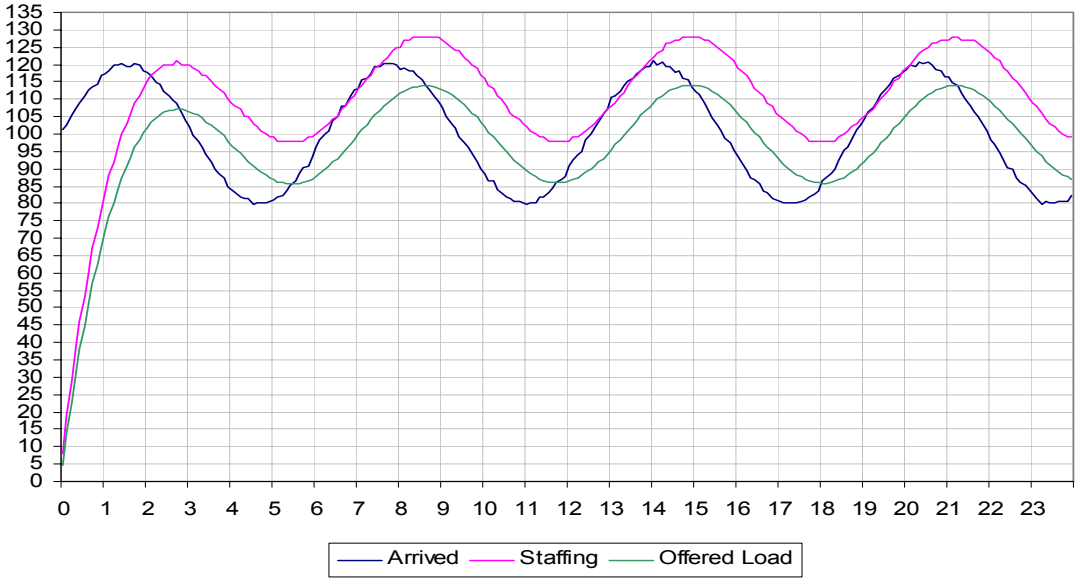
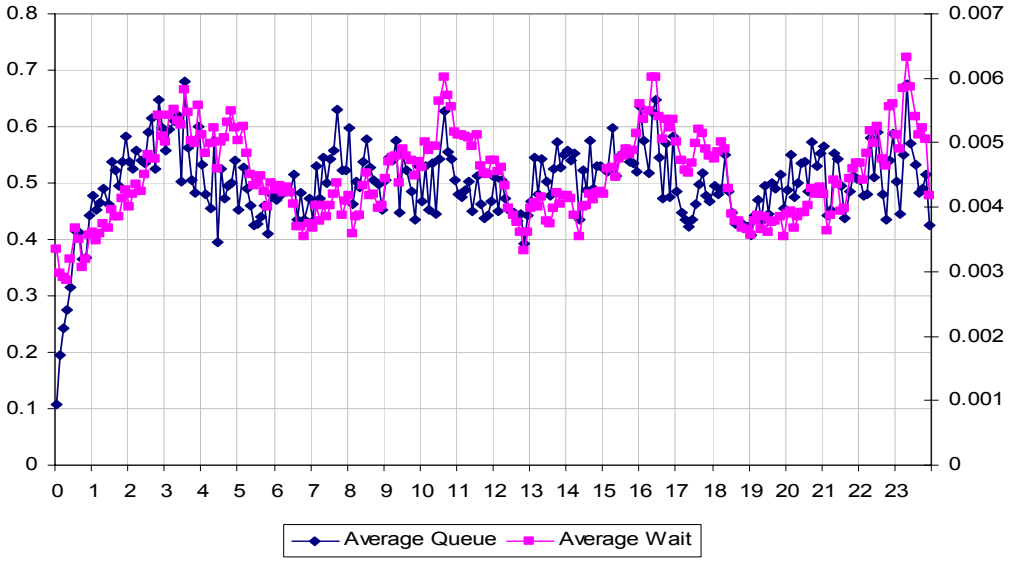


Figure 31: Target α=0.1: (1) Staffing level, offered load and arrival function, (2) average queue and waiting time, (3) Waiting time histogram of those who waited





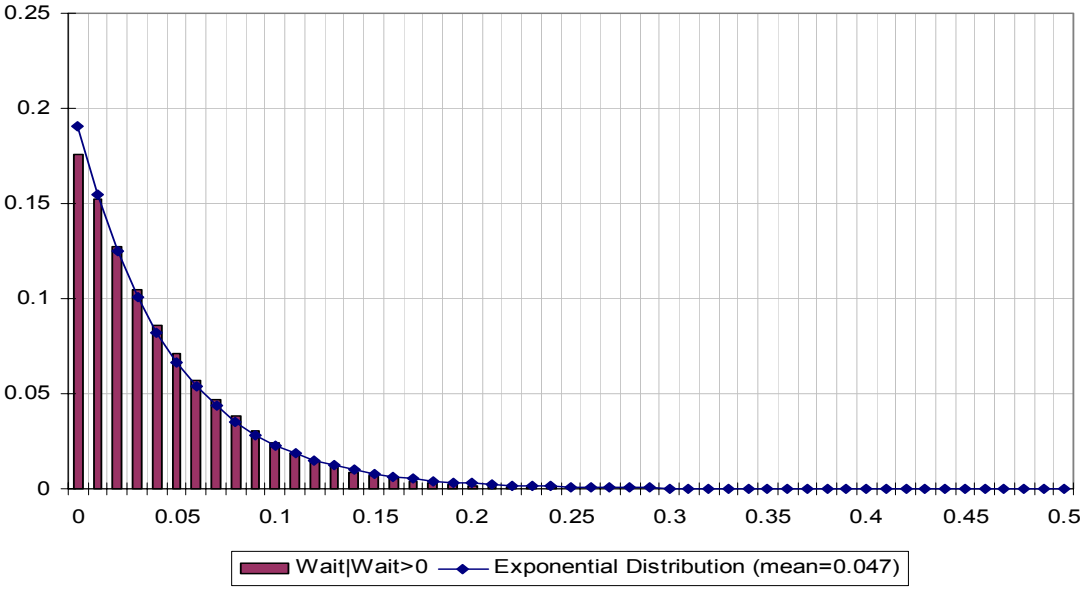
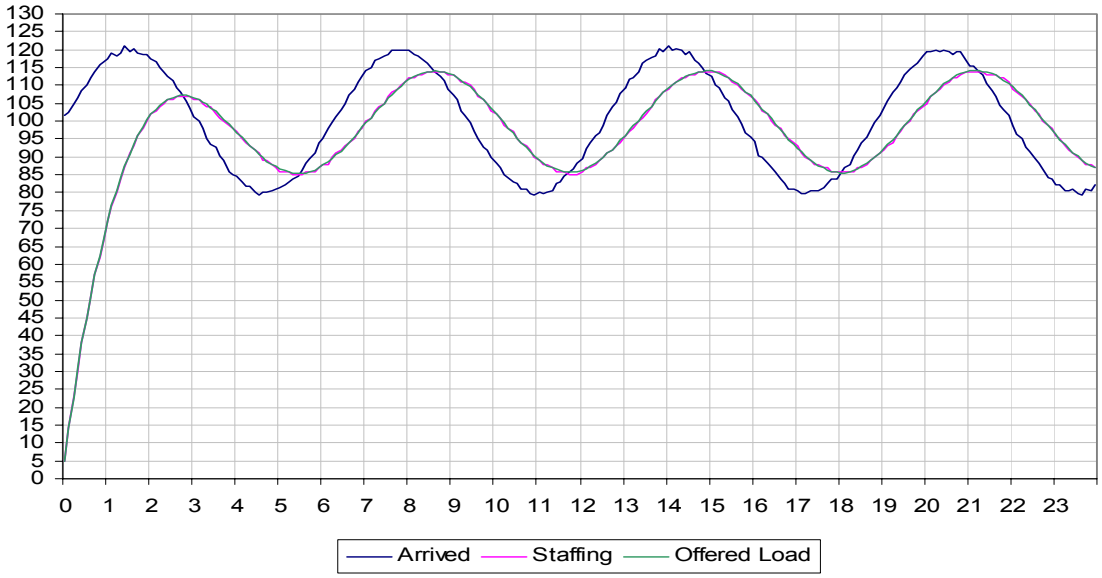
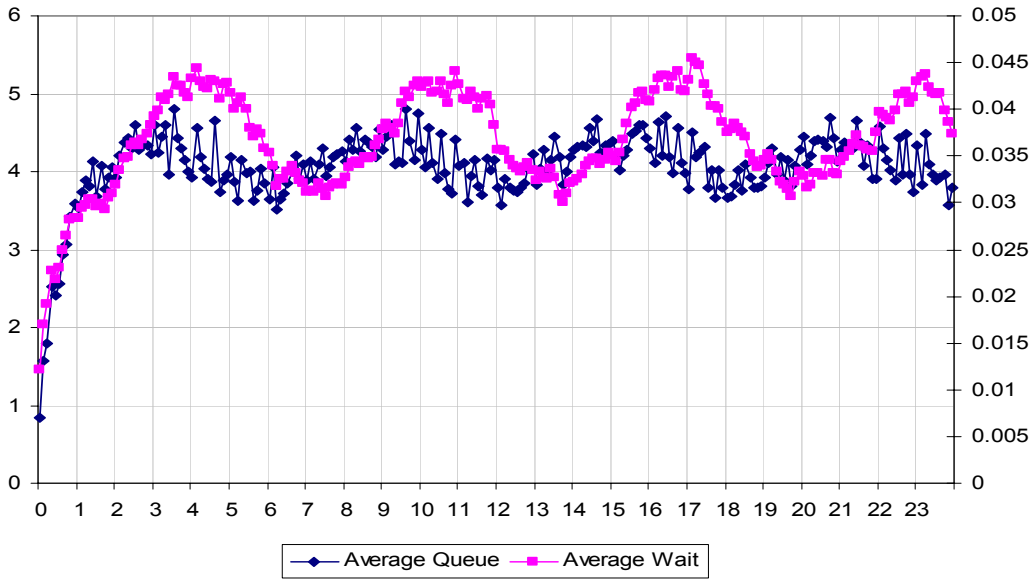


Figure 32: Target α=0.5: (1) Staffing level, offered load and arrival function, (2) average queue and waiting time, (3) Waiting time histogram of those who waited





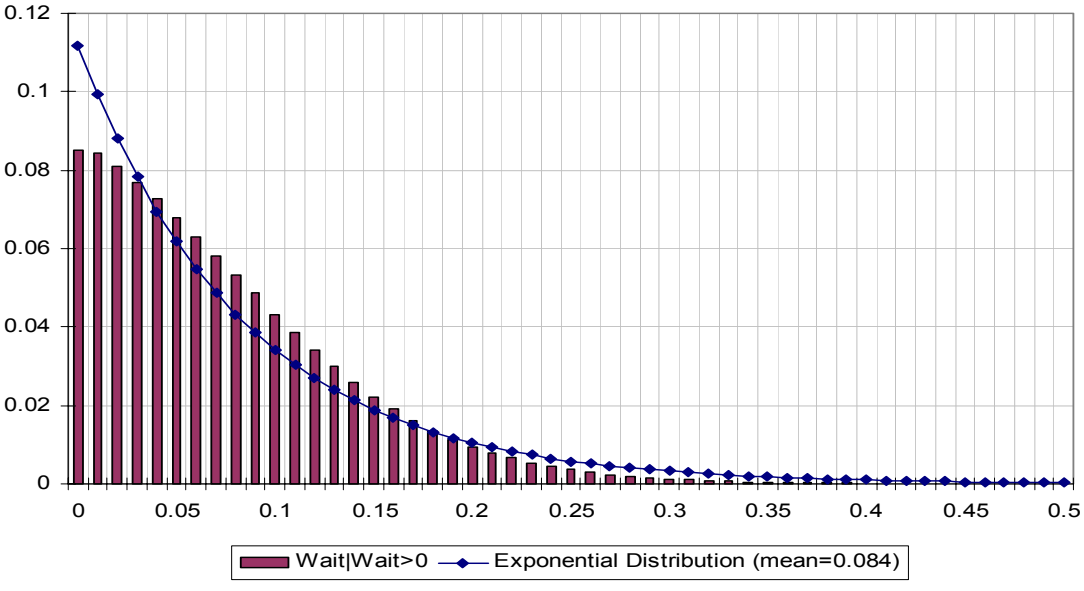
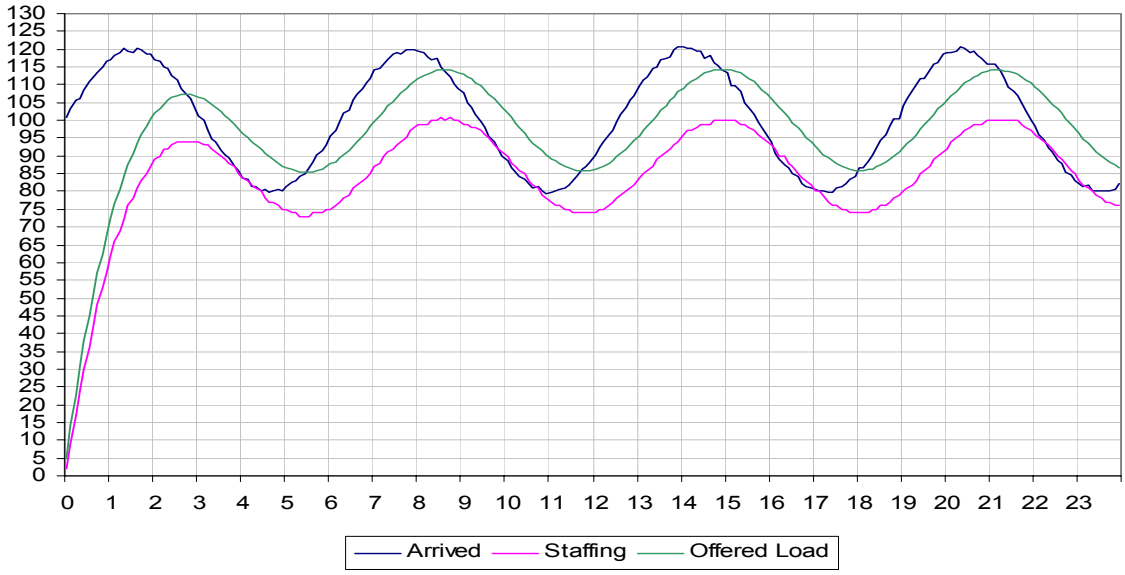
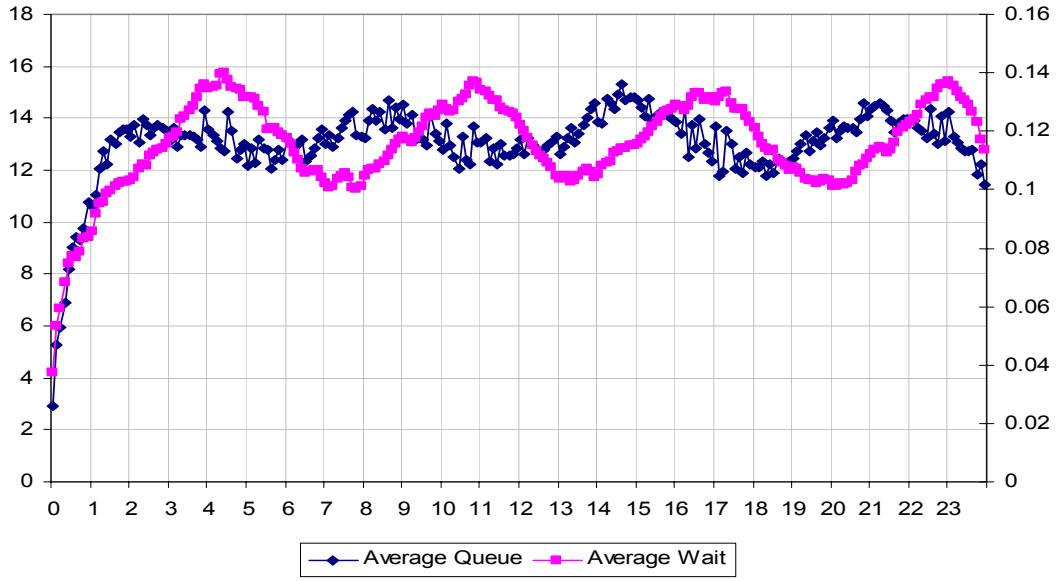


Figure 33: Target α=0.9: (1) Staffing level, offered load and arrival function, (2) average queue and waiting time, (3) Waiting time histogram of those who waited





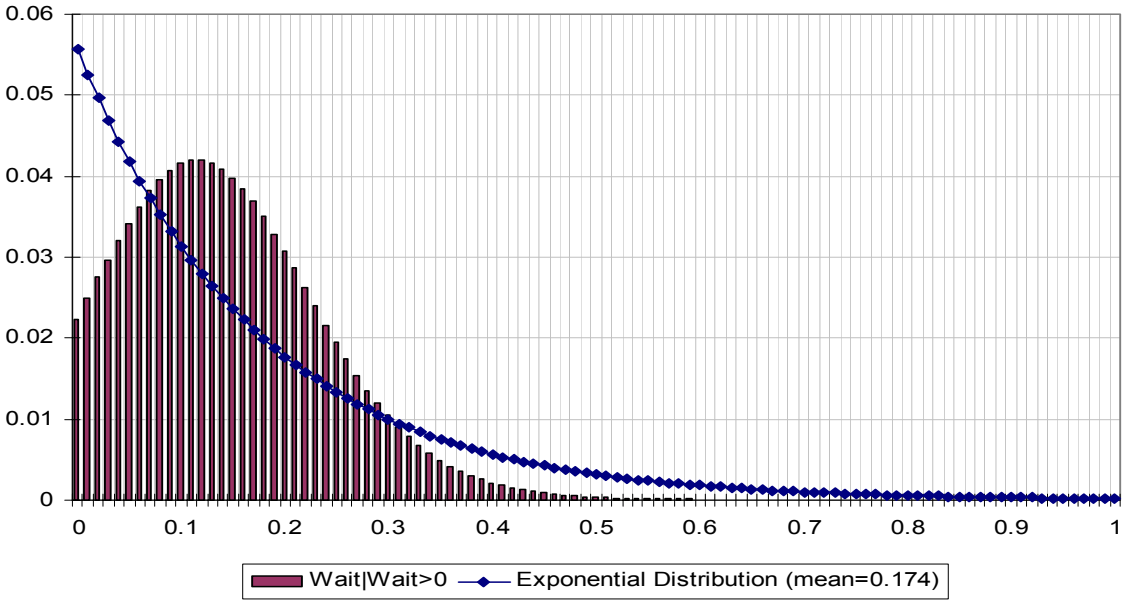
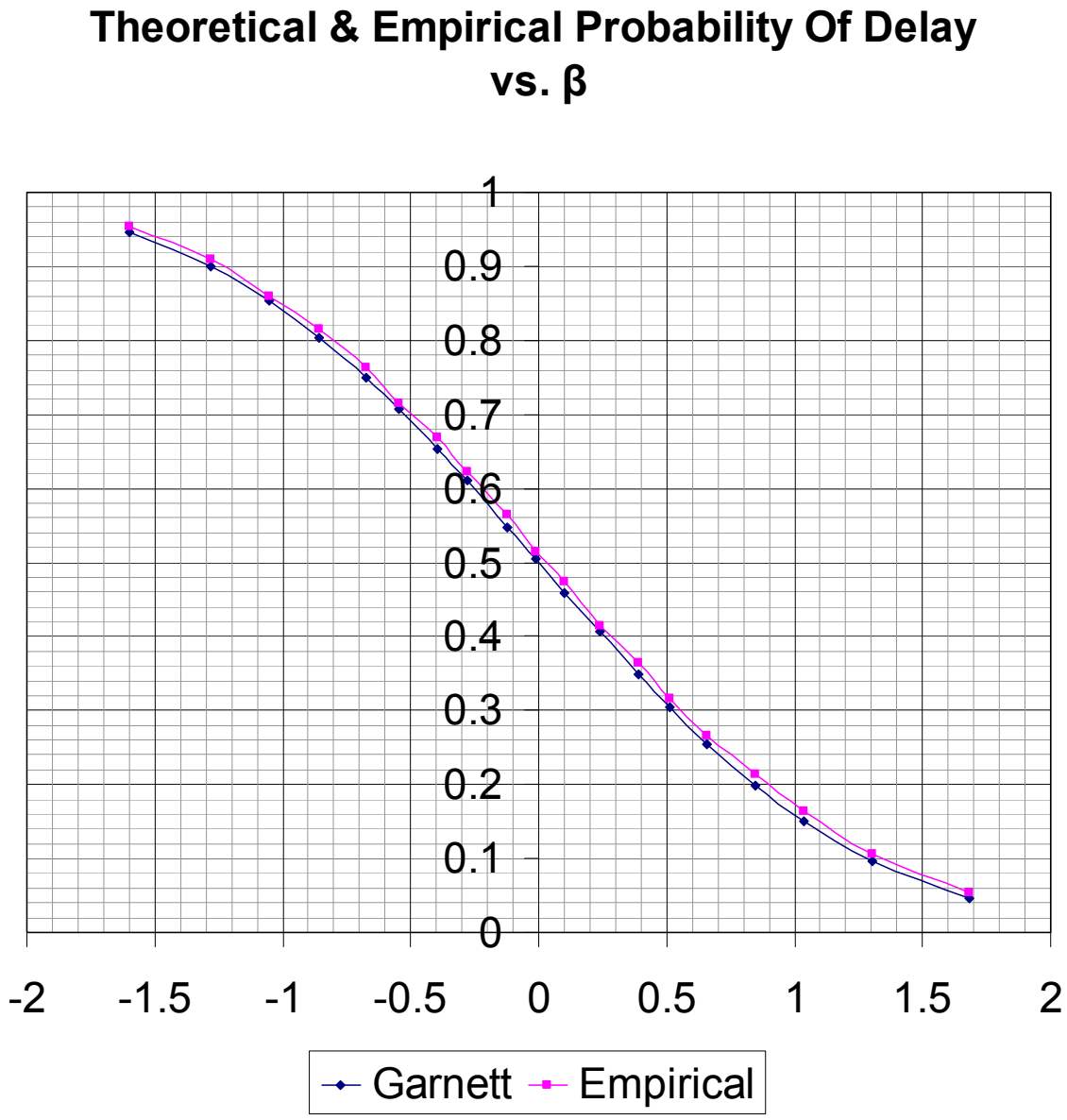


Figure 34: Comparison of empirical results with the Garnett approximation

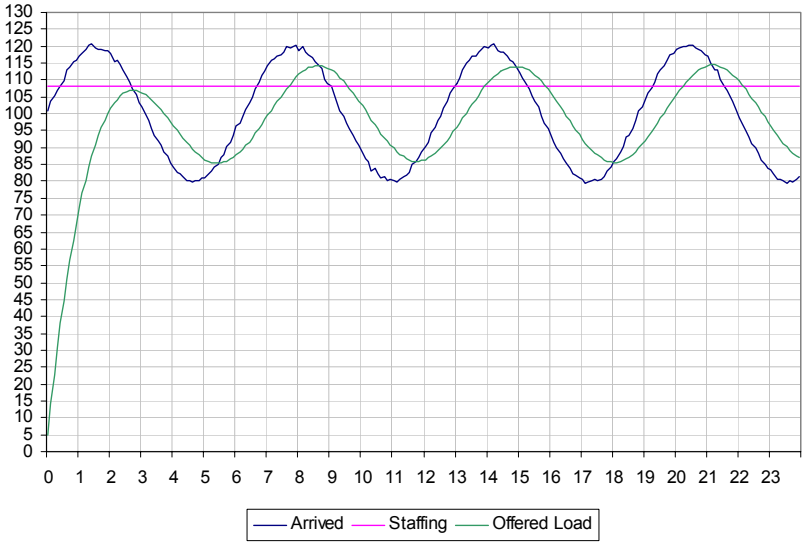


Note: The above graph compares the empirical probability of delay with the corresponding delay-function that arises in Garnett et al (2003). The two functions are indeed remarkably close.

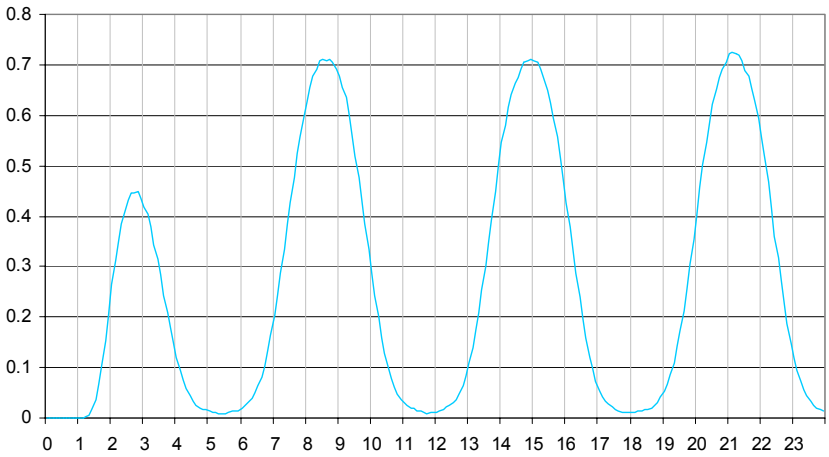
In the above, we are using exponential (im)patience with mean 1.

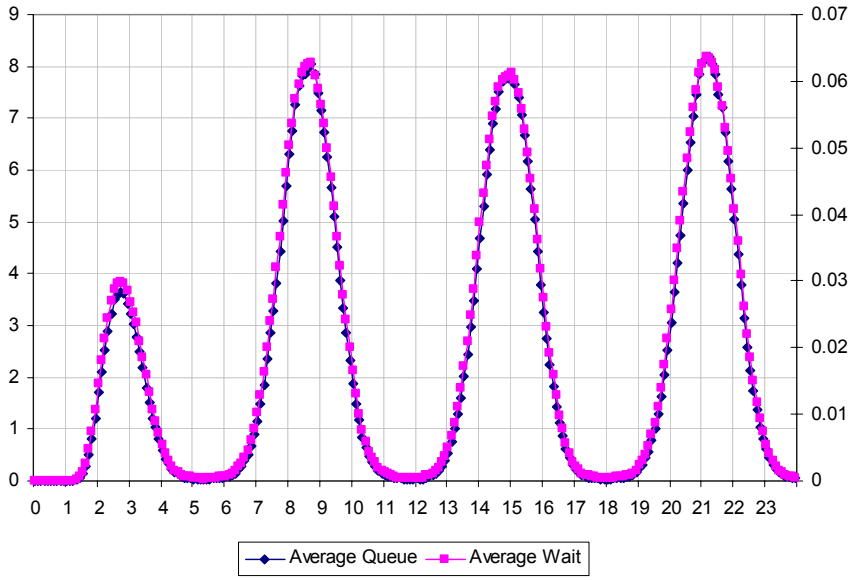
5.2. While the Iterative Algorithm provides excellent results, other approximations fail miserably. We'll look at the performance of 2 common approximations: the **SSA** (as presented in section 2), and the **PSA.**

Figure 35: SSA with target α=0.2: (1) Staffing function, offered load and arrival function, (2) Delay probability (3) Average queue and average wait



Delay Probability

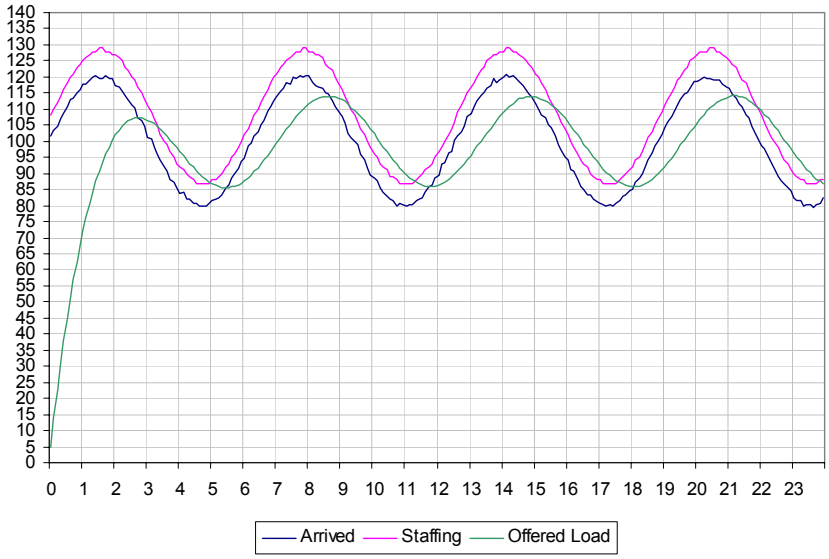


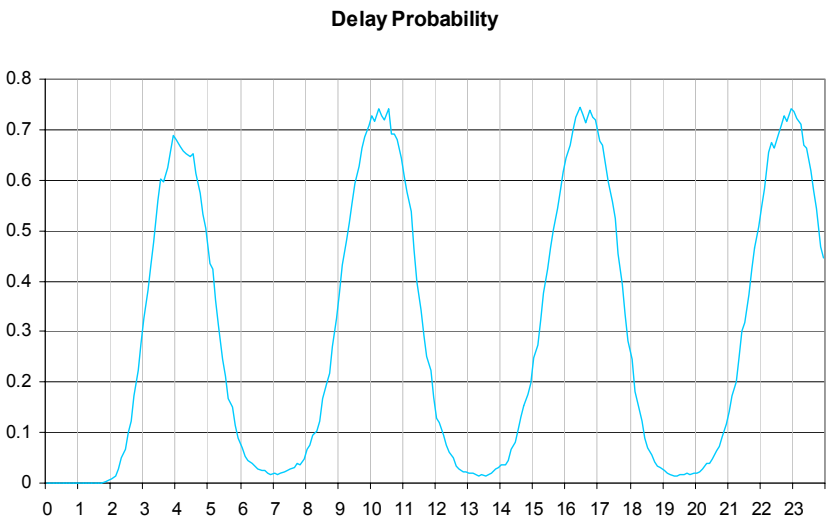


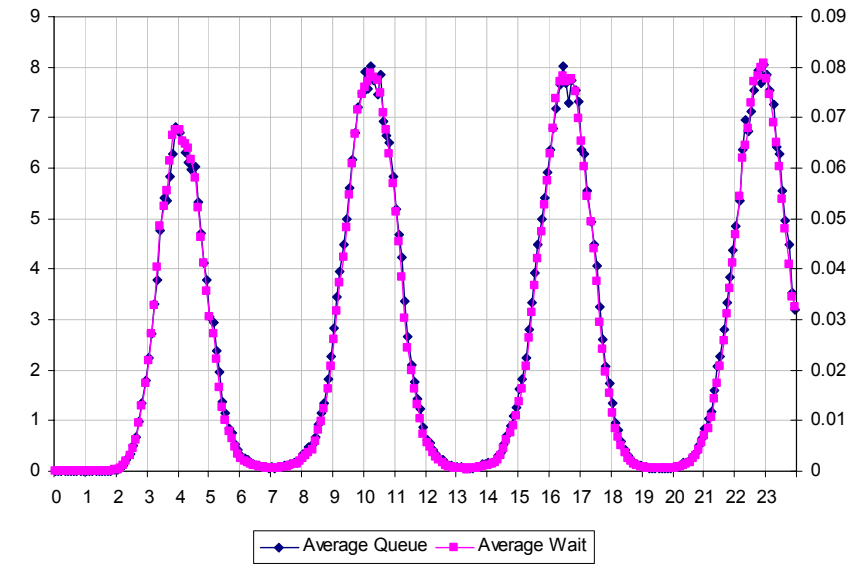
1.

The PSA – Point-wise Stationary Approximation. Time-varying staffing levels, based on steady-state M/M/s, with λ= λ(t), at each time t

Figure 36: PSA target α=0.2: (1) Staffing function, offered load and arrival function, (2) Delay probability (3) Average queue and average wait







5.3. Here, we give an example of a case with very impatient customers. the arrival rate $\lambda(t) = 100 + 20 \cdot \sin(t)$, service times are exponential having mean 1, and (im)patience is exponential with **parameter 5 (mean 0.2).**

Figure 37: Delay probability summary for the impatient Erlang-A example

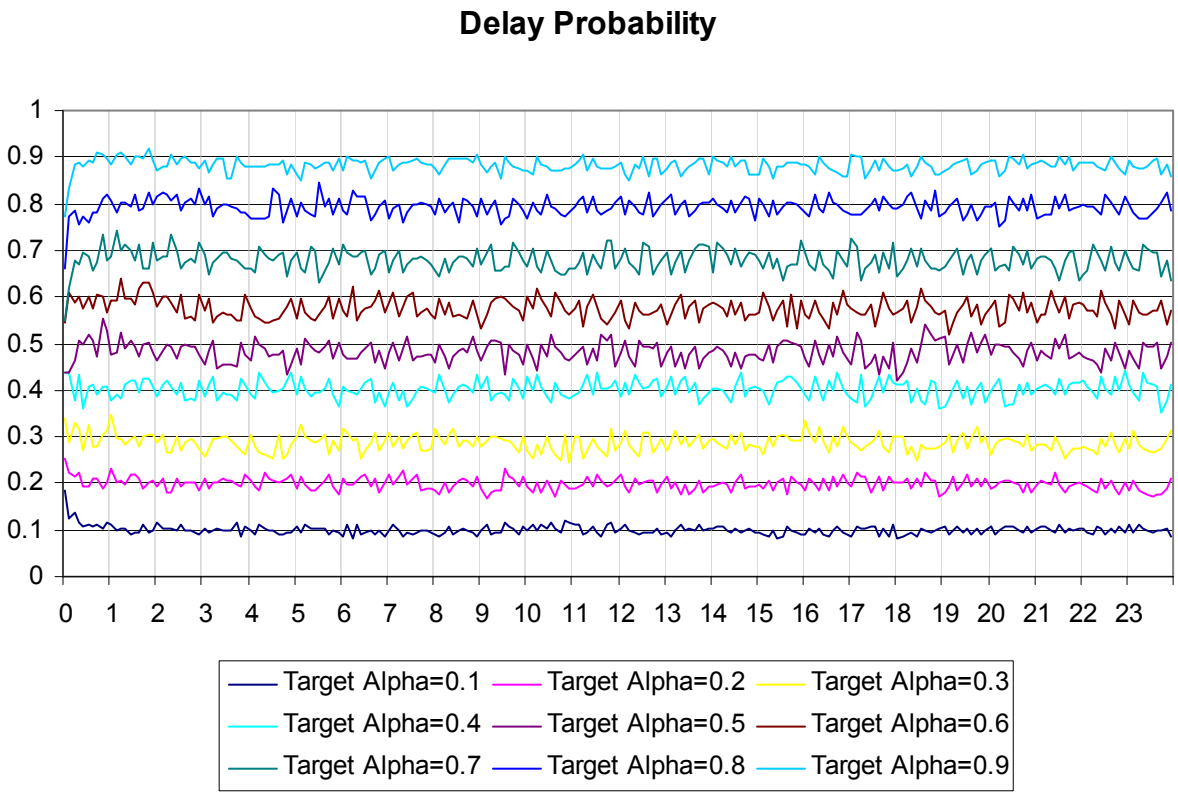


Figure 38: Implied service grade β summary for the impatient Erlang-A example

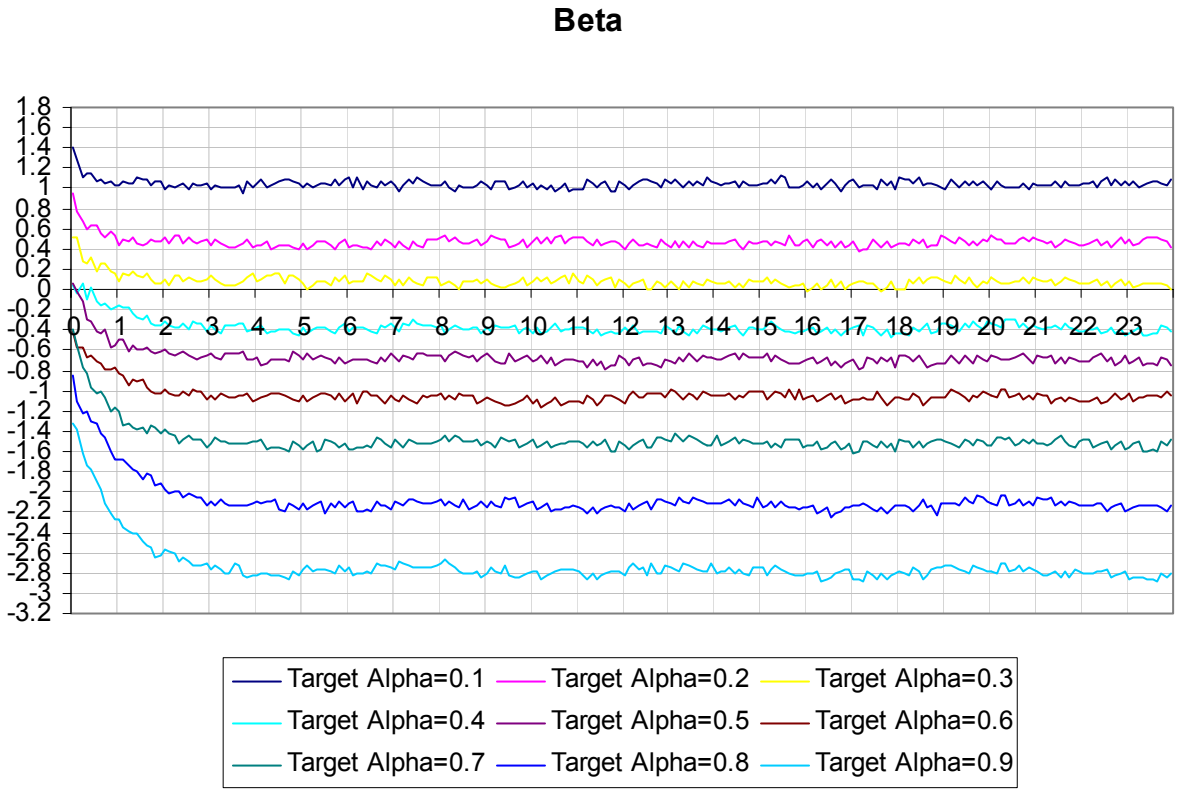


Figure 39: Abandon probability summary for the impatient Erlang-A example

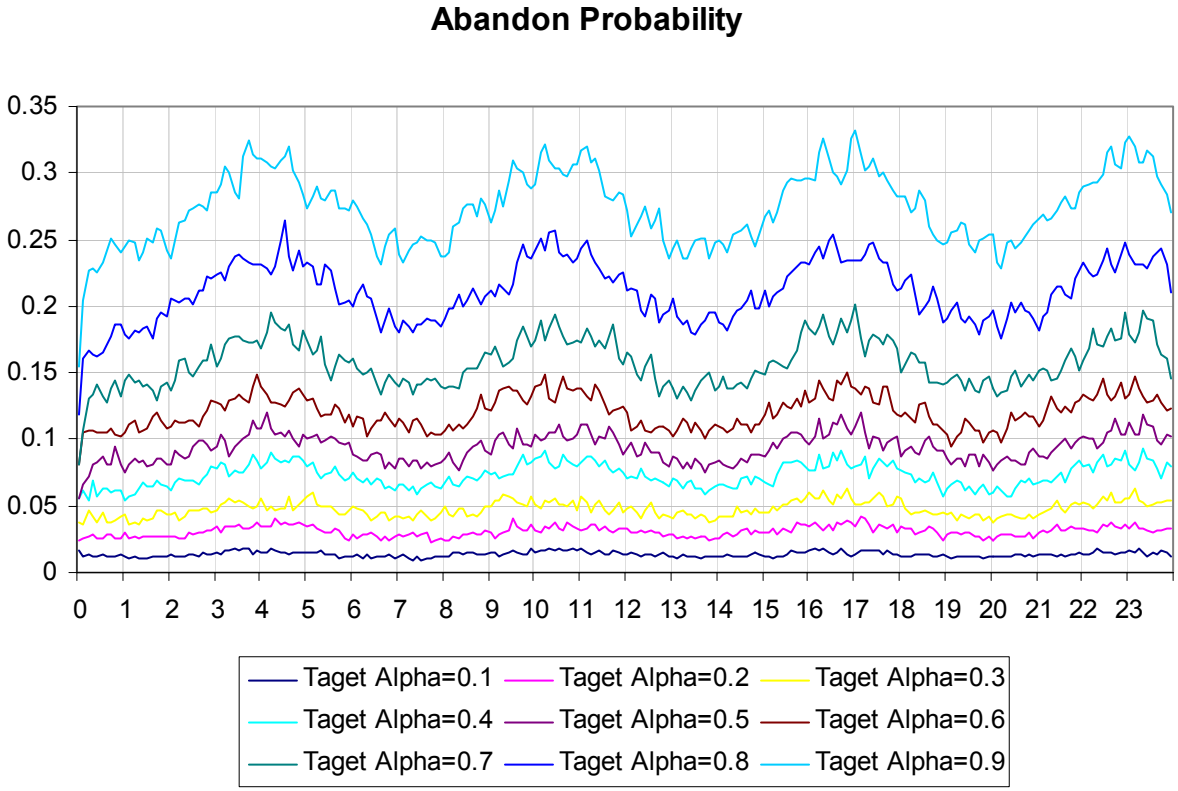


Figure 40: Utilization summary for the impatient Erlang-A example

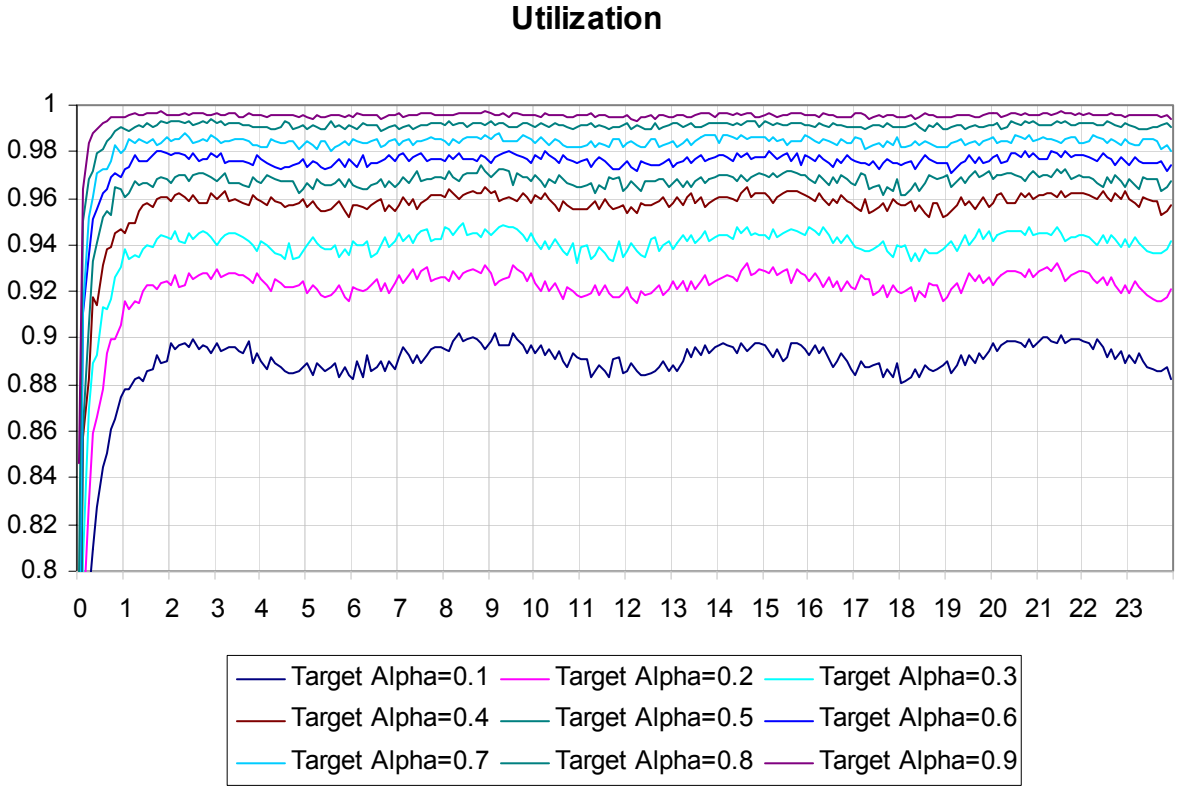


Figure 41: Tail probability summary for the impatient Erlang_A example

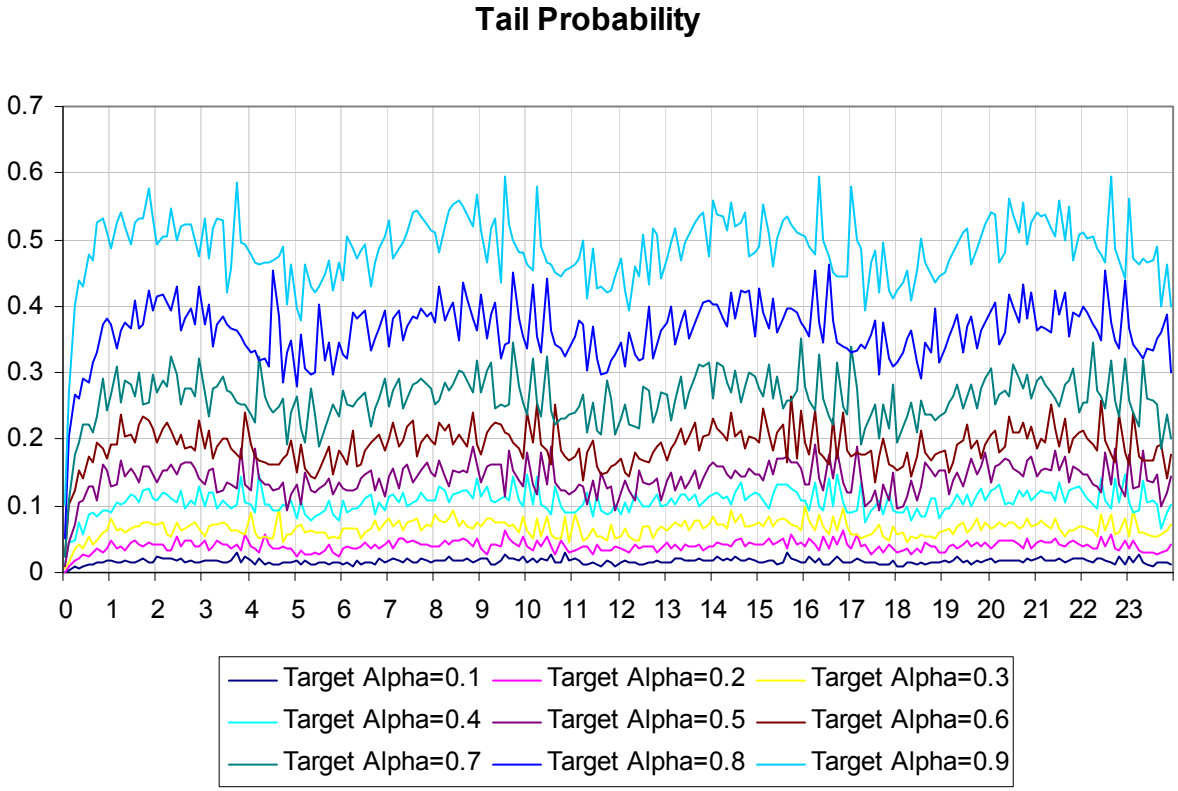
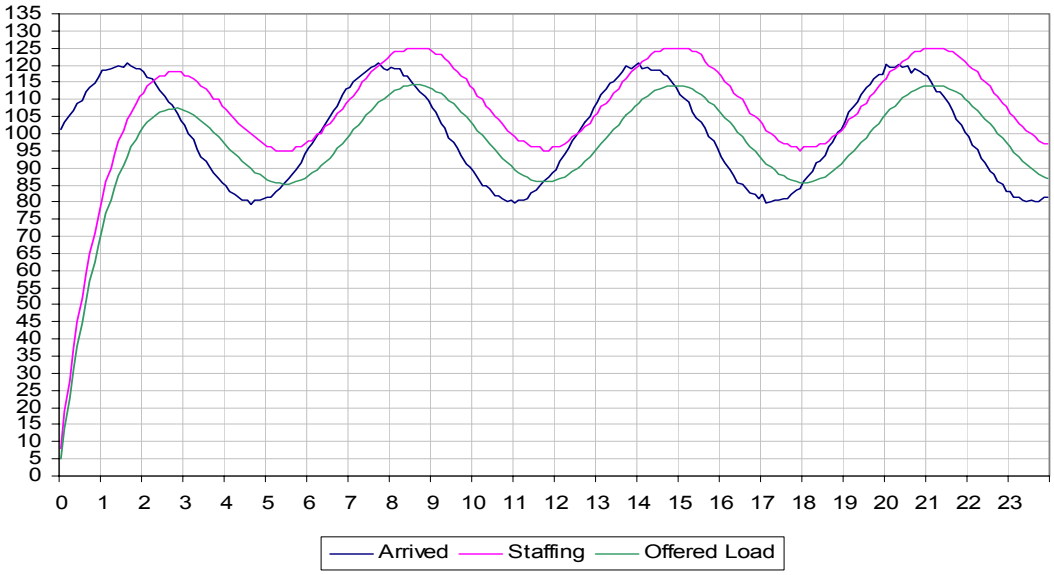
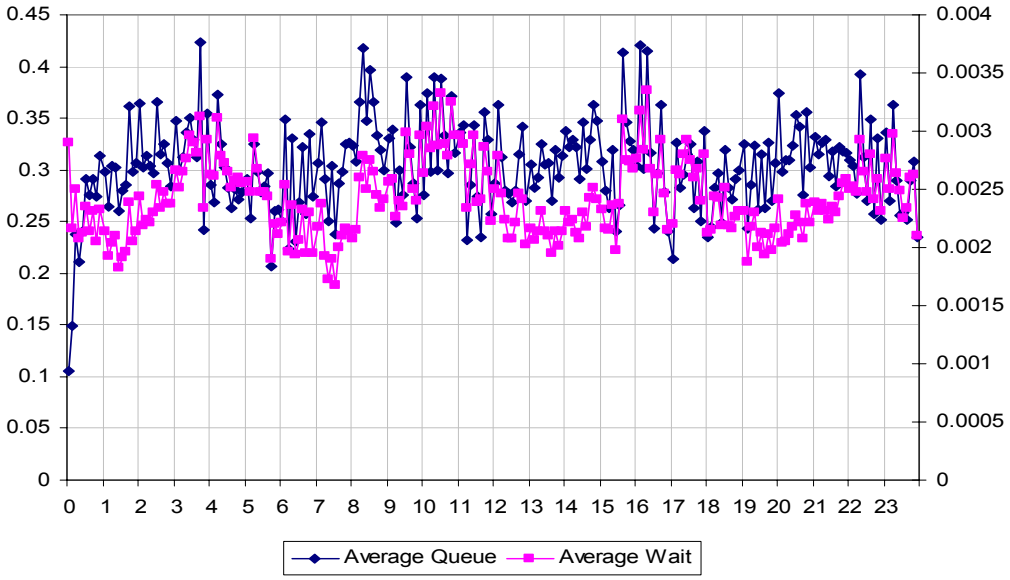


Figure 42: Target α=0.1: (1) Staffing level, offered load and arrival function, (2) average queue and waiting time, (3) Waiting time histogram of those who waited





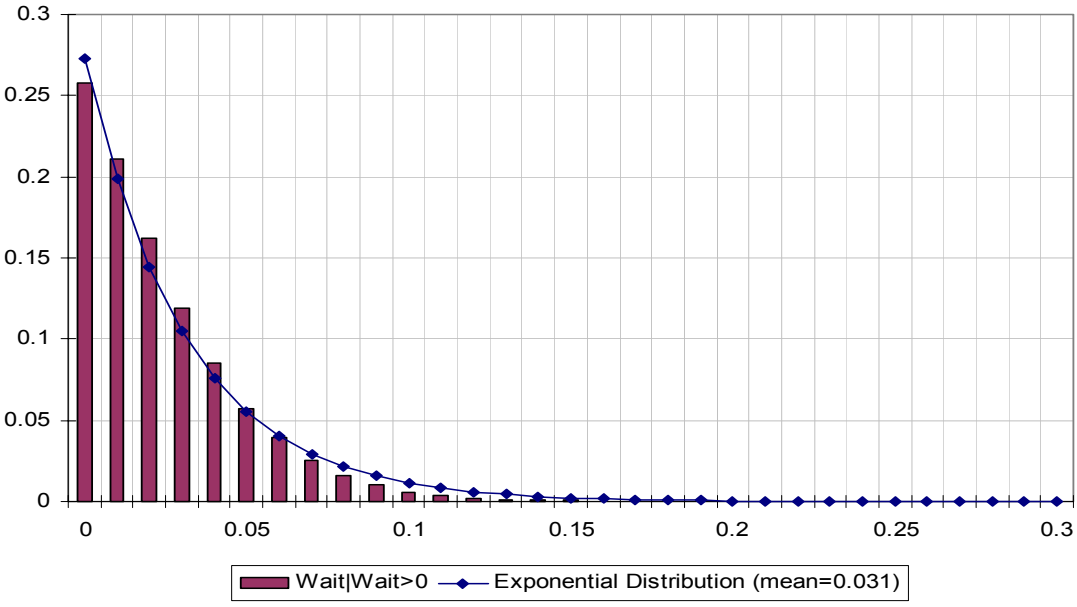
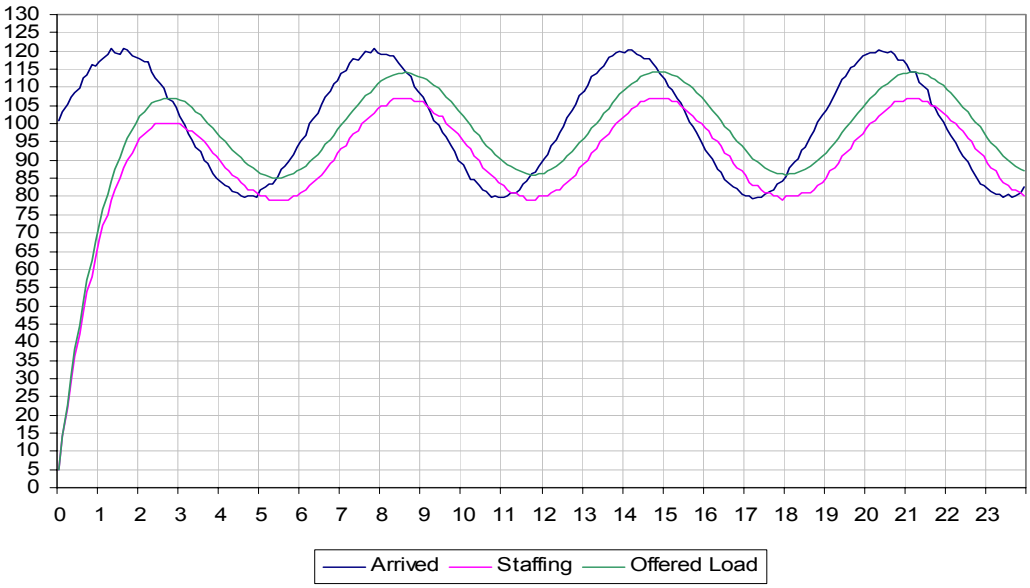
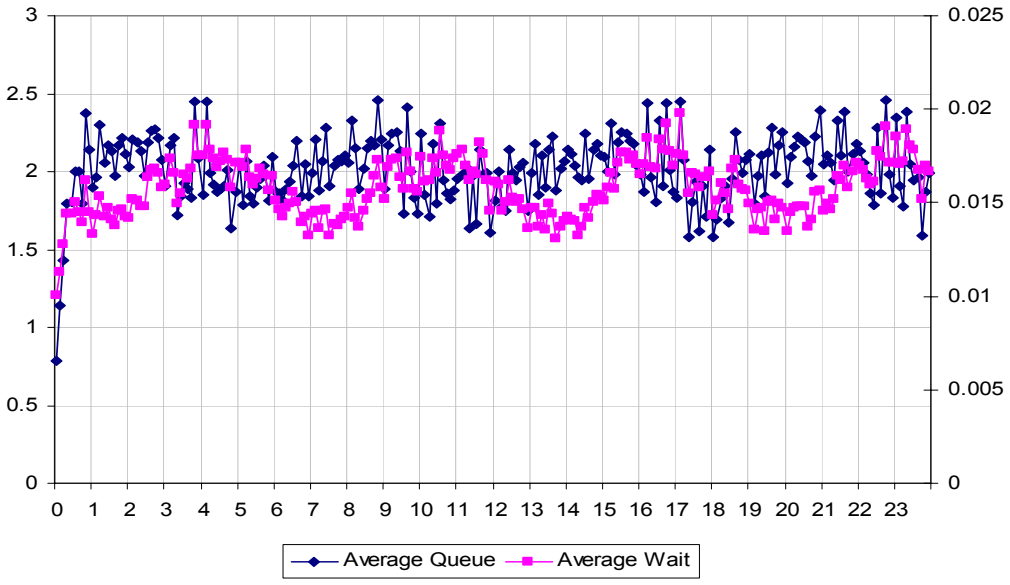


Figure 43: Target α=0.5: (1) Staffing level, offered load and arrival function, (2) average queue and waiting time, (3) Waiting time histogram of those who waited





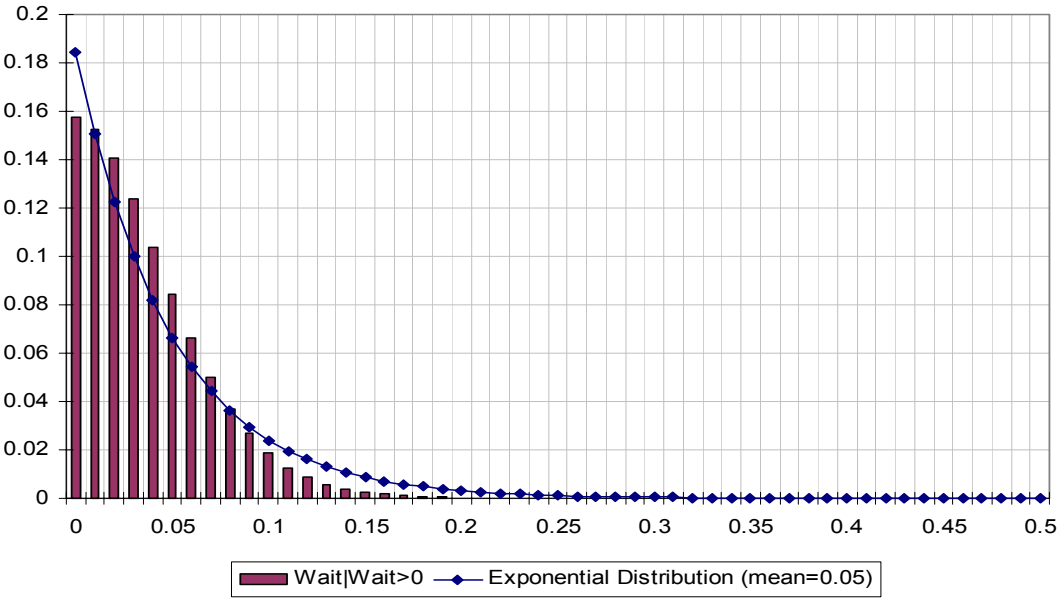
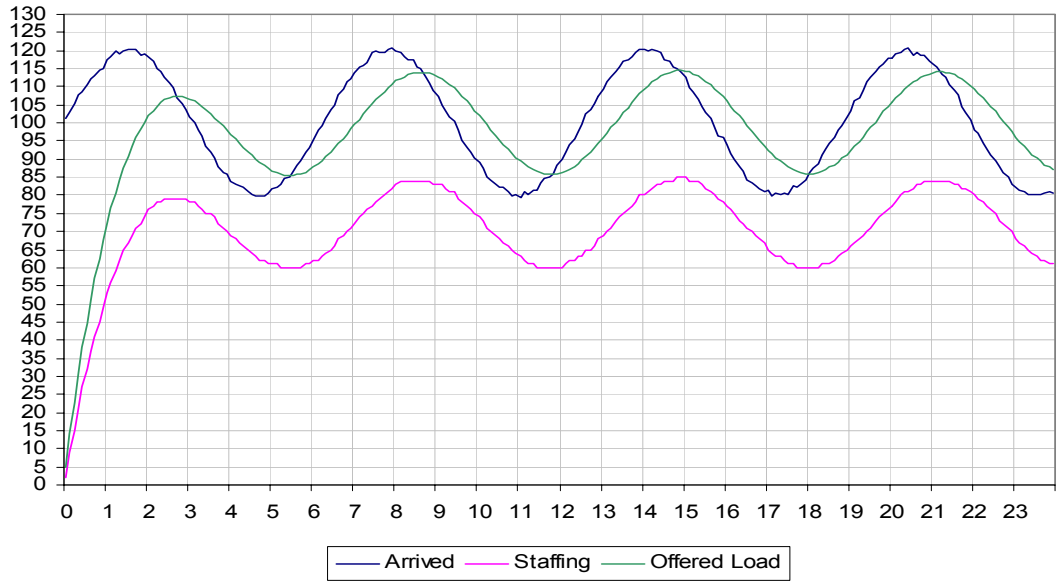
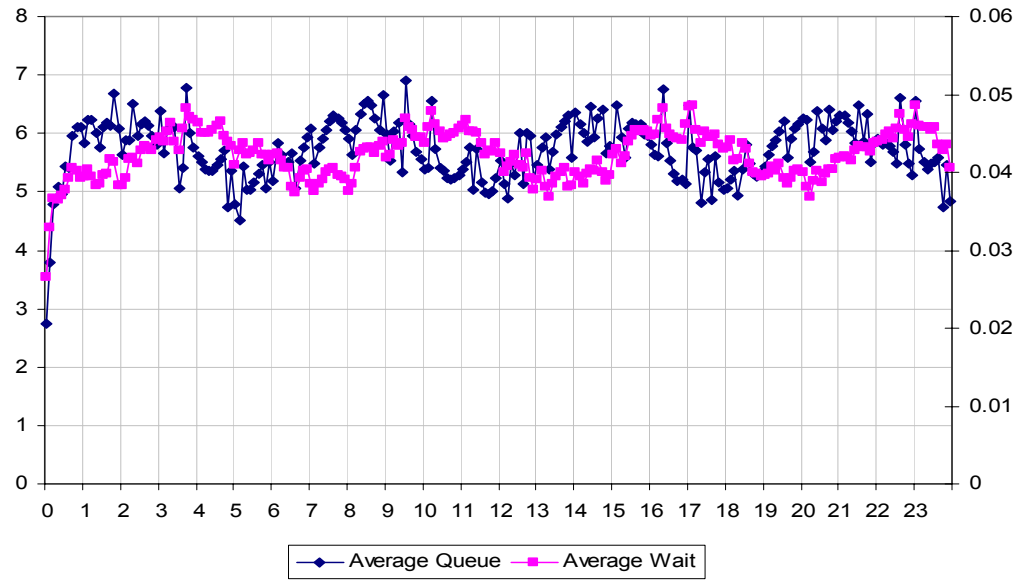


Figure 44: Target α=0.9: (1) Staffing level, offered load and arrival function, (2) average queue and waiting time, (3) Waiting time histogram of those who waited





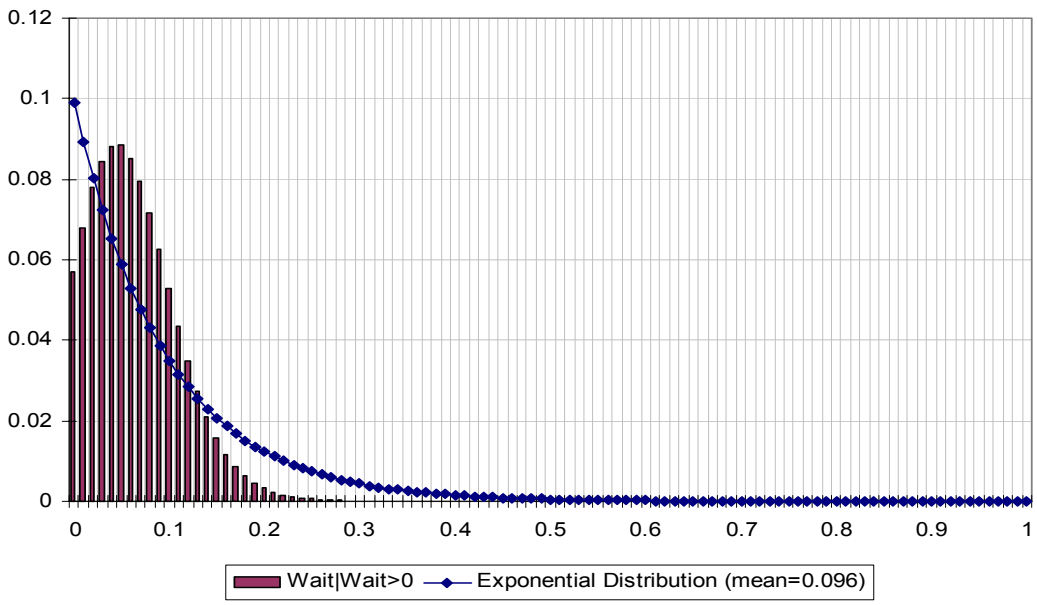
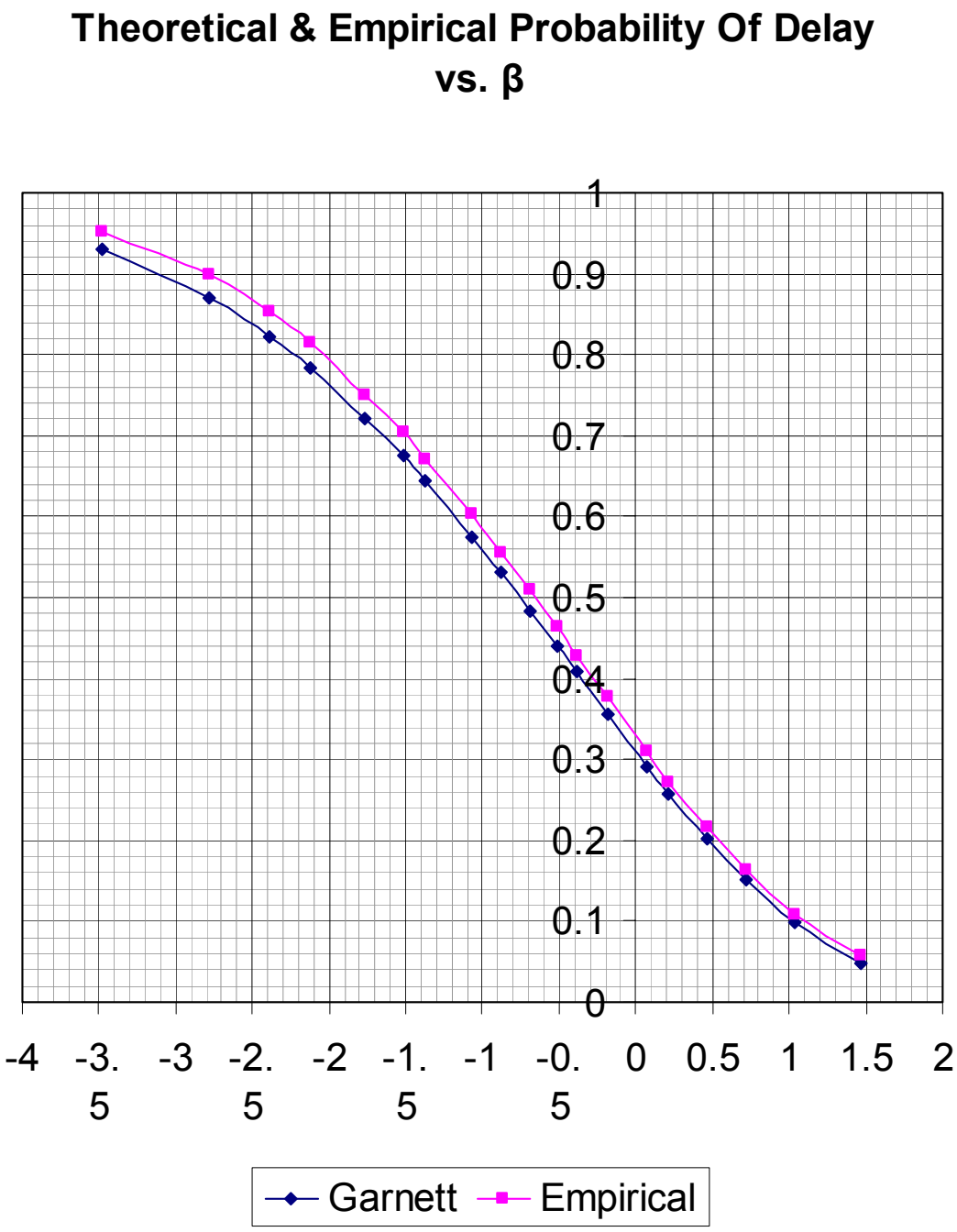


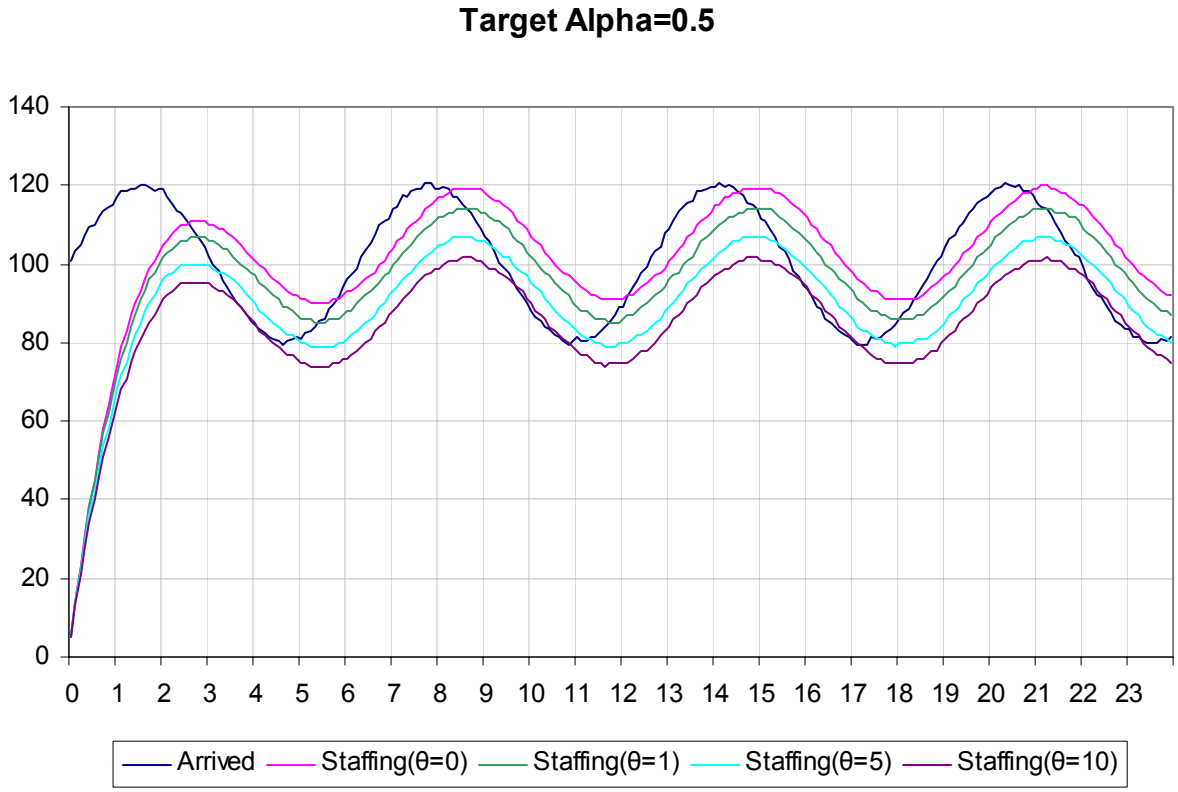
Figure 45: Comparison of empirical results with the Garnett approximation



In the above, we are using exponential (im)patience with mean 5.

The following graph demonstrate the benefit of staffing a system with consideration to abandonment in contrast to using a simple model with no abandonment. We show the difference between staffing levels for (im)patience distribution with parameters θ=0, 1, 5, 10.

Figure 46: Staffing under various (im)patience parameters

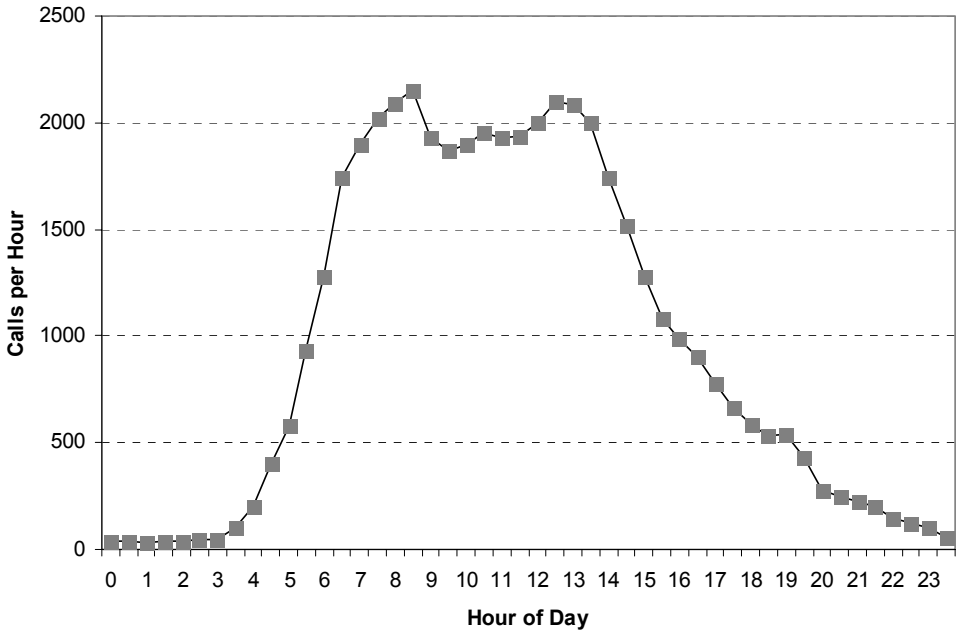


Note the area between curves quantifies the savings of labor in time-units.
In this case the saving in labor, had one used θ=1, is 113.3 time units, 270 time units for θ=5, and 386 time units when θ=10.
The savings of labor increase as α increases.

6. Practical Example

In this section we consider the practical case introduced in Figure 1.
Arrival-rate function is as follows:

Figure 47: Arrival function of the practical example



Service Times are exponential having mean 10, and (im)patience is also exponential with parameter 10.

Figure 48: Delay probability summary for the practical example

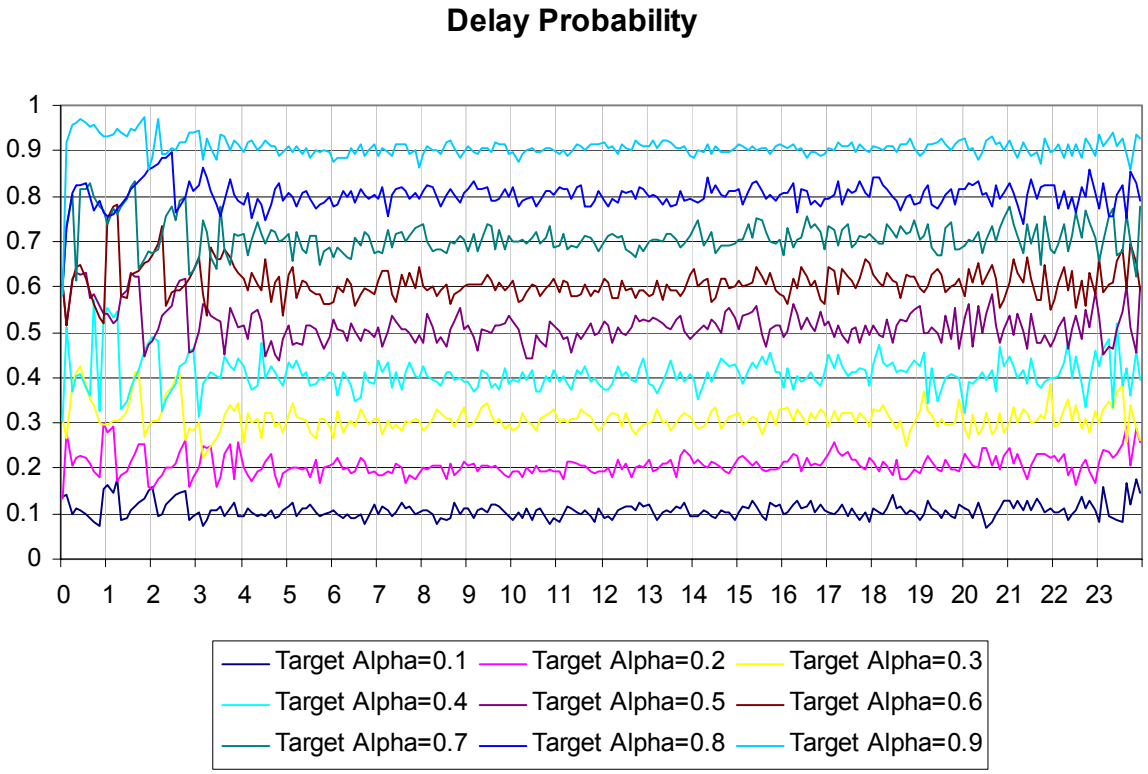


Figure 49: Implied service grade β summary for the practical example

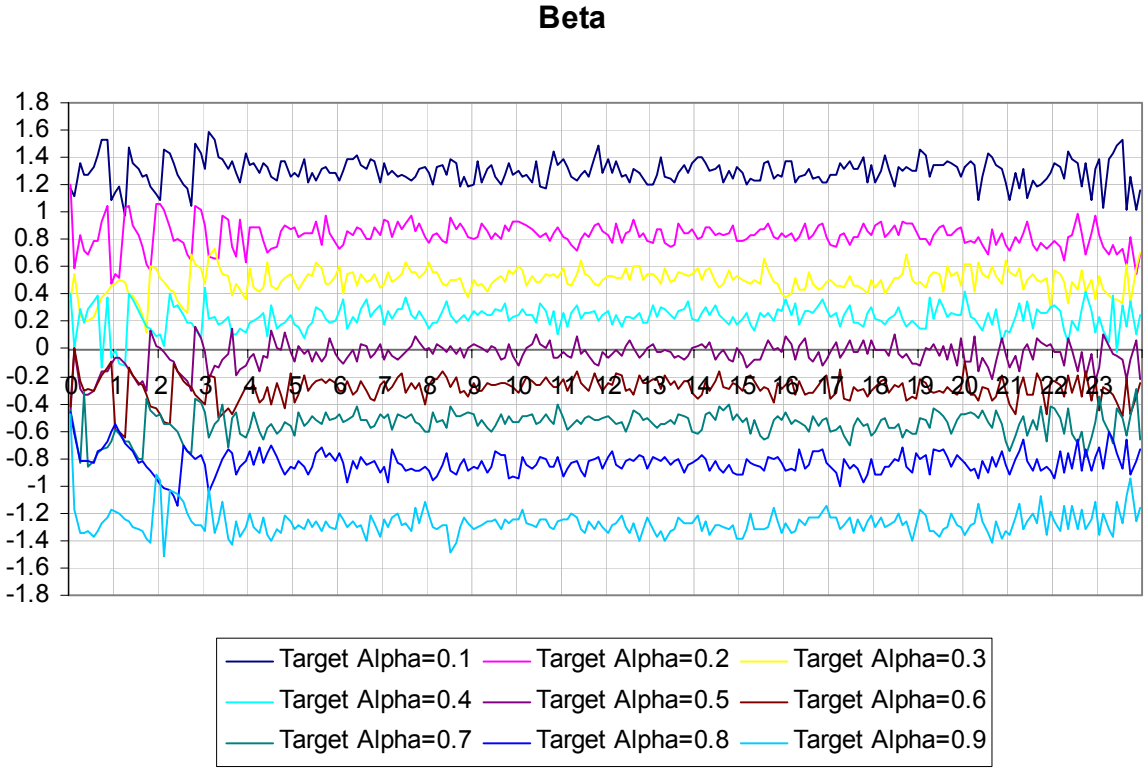


Figure 50: Abandon probability summary for the practical example

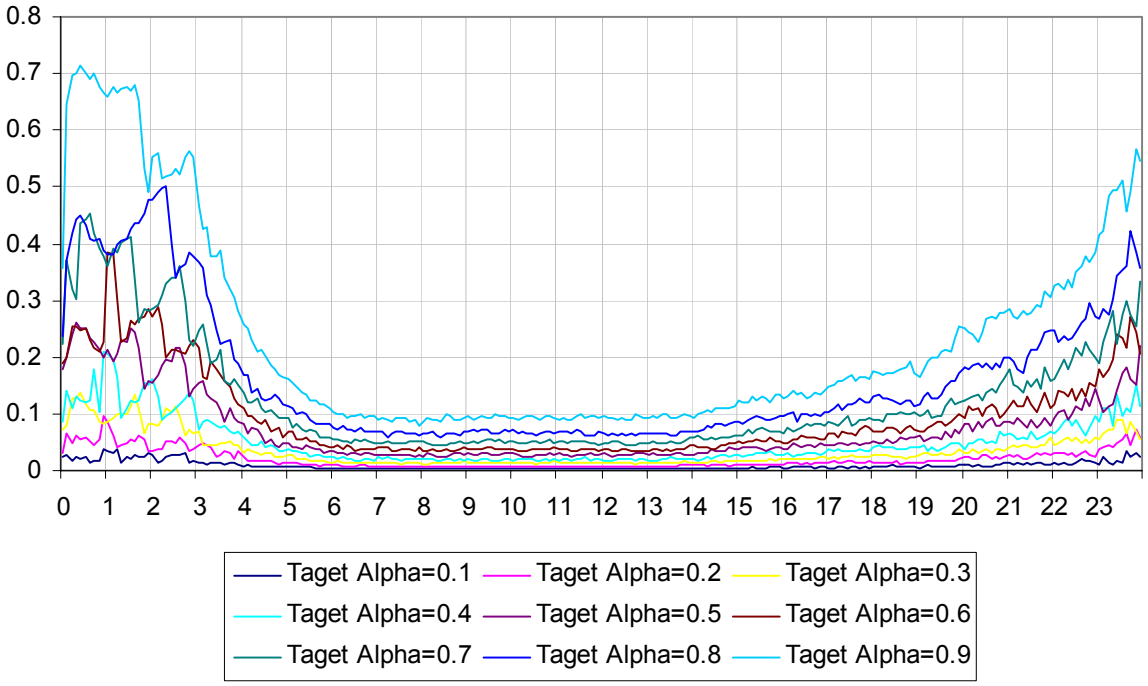


Figure 51: Utilization summary for the practical example

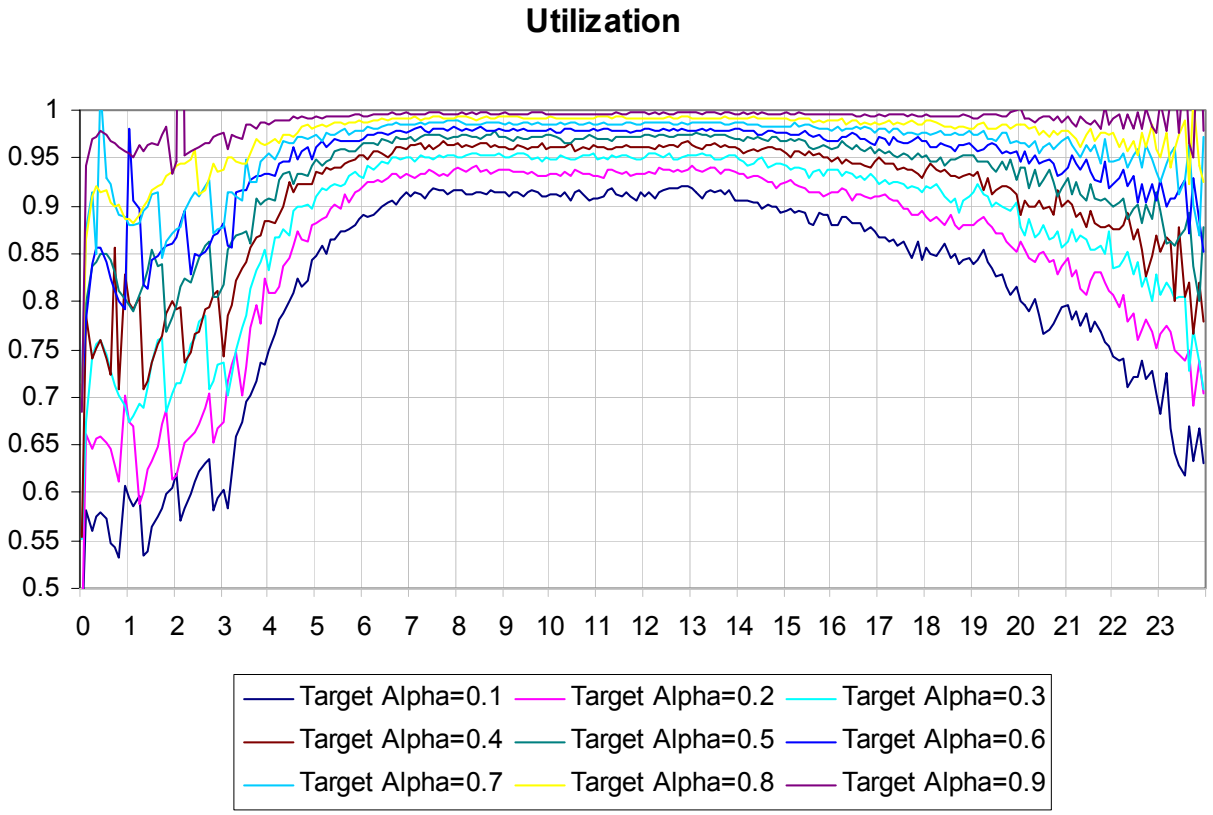
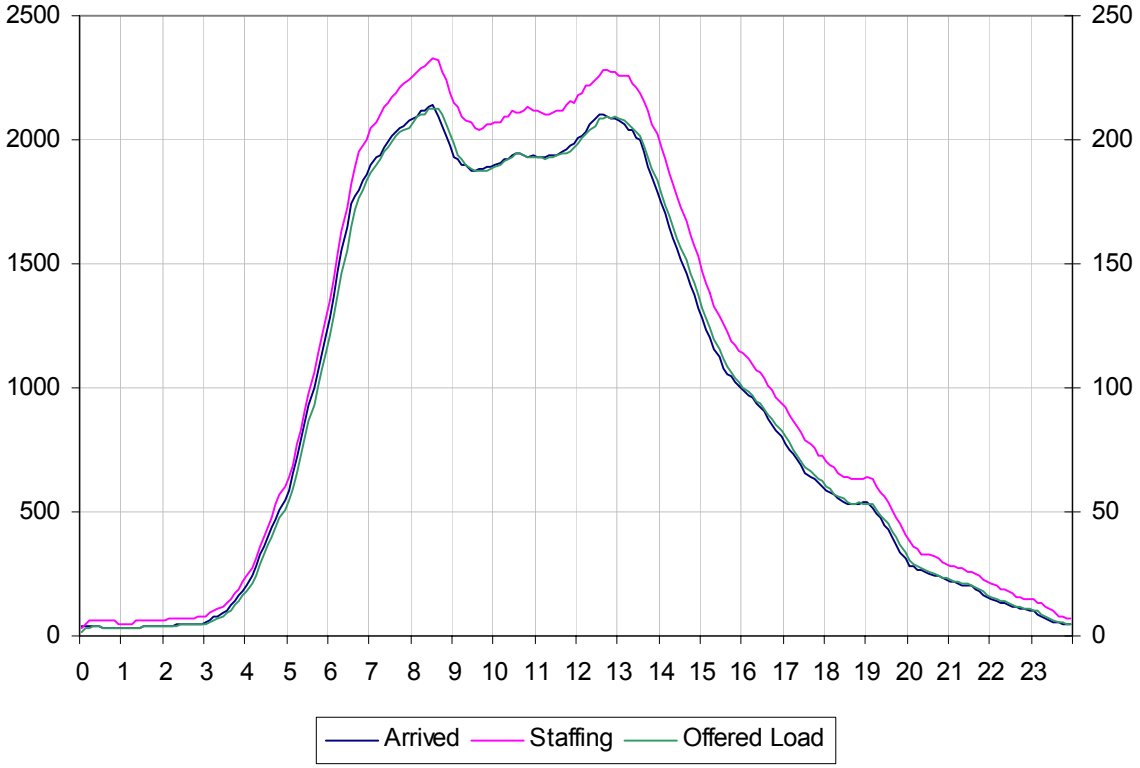


Figure 52: Target α=0.1: (1) Staffing level, offered load and arrival function, (2) average queue and waiting time



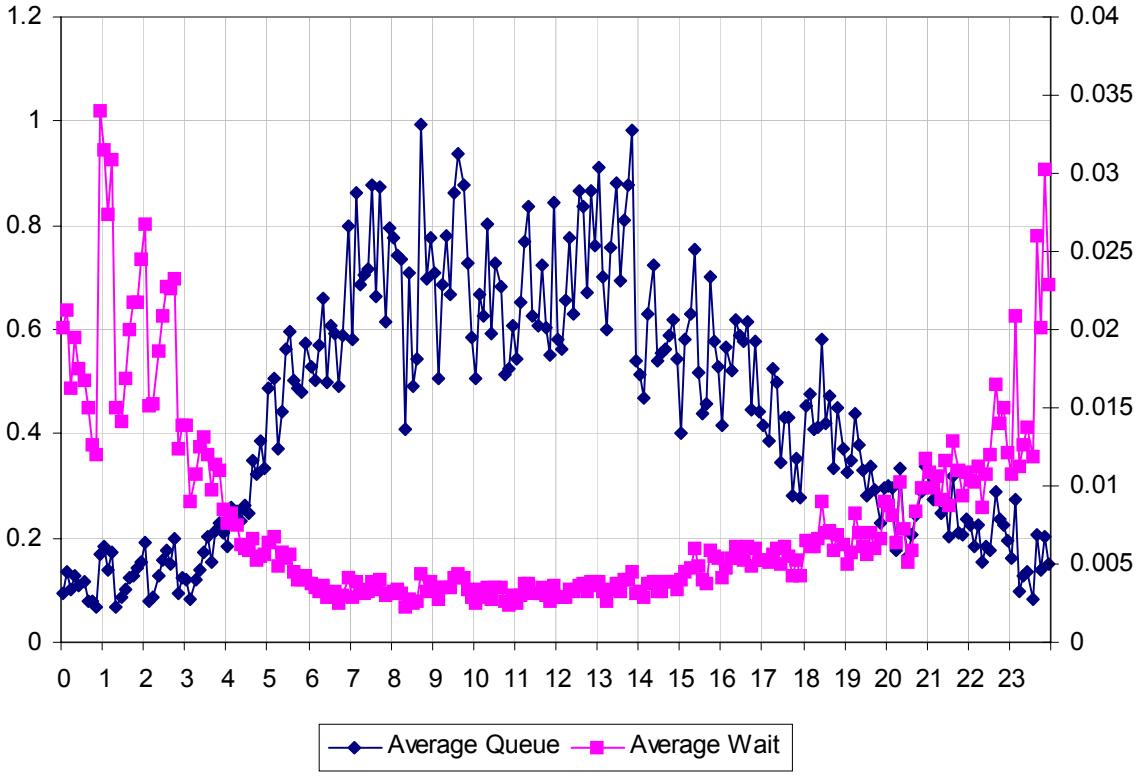


Figure 53: Target α=0.5: (1) Staffing level, offered load and arrival function, (2) average queue and waiting time

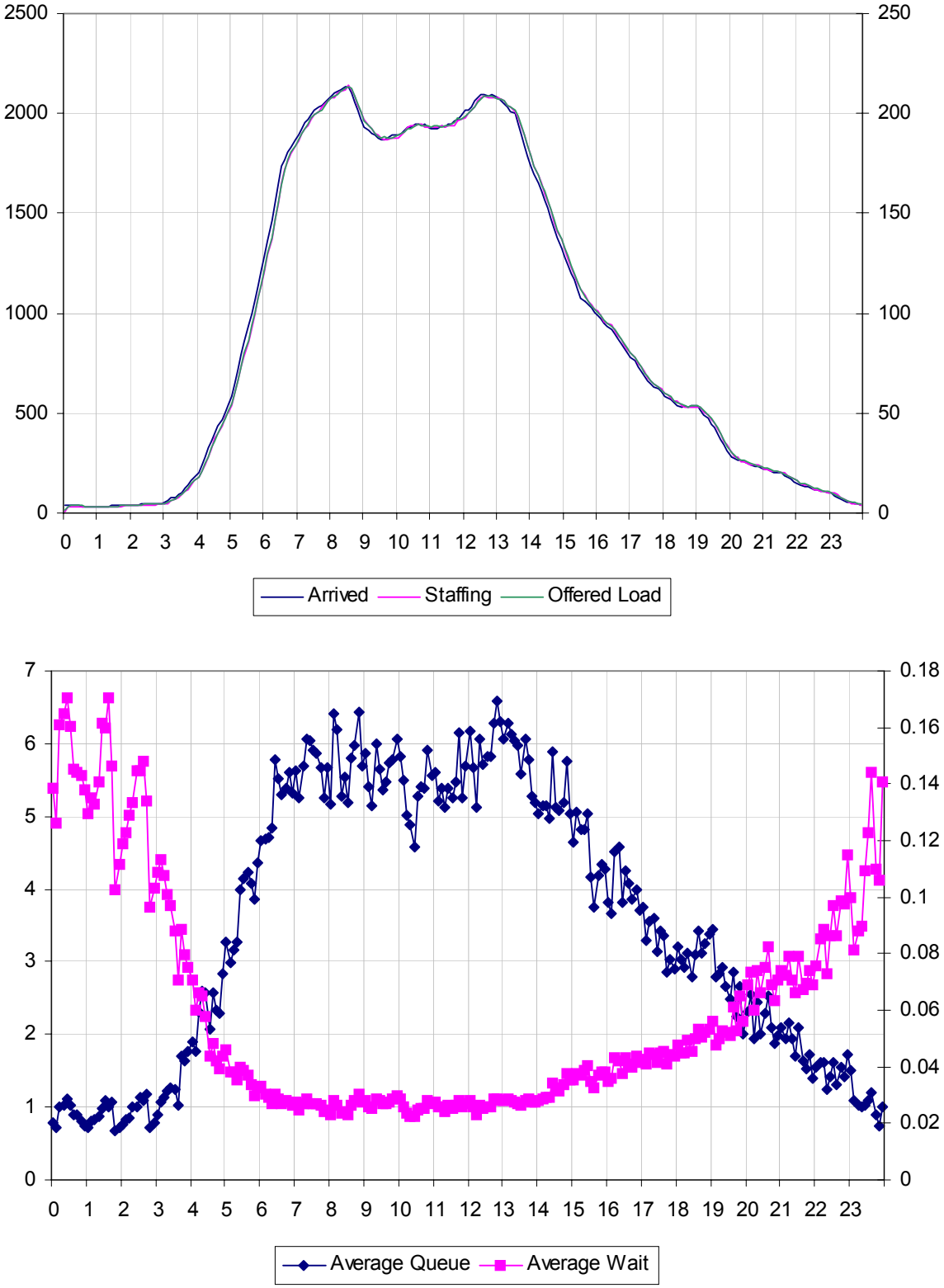
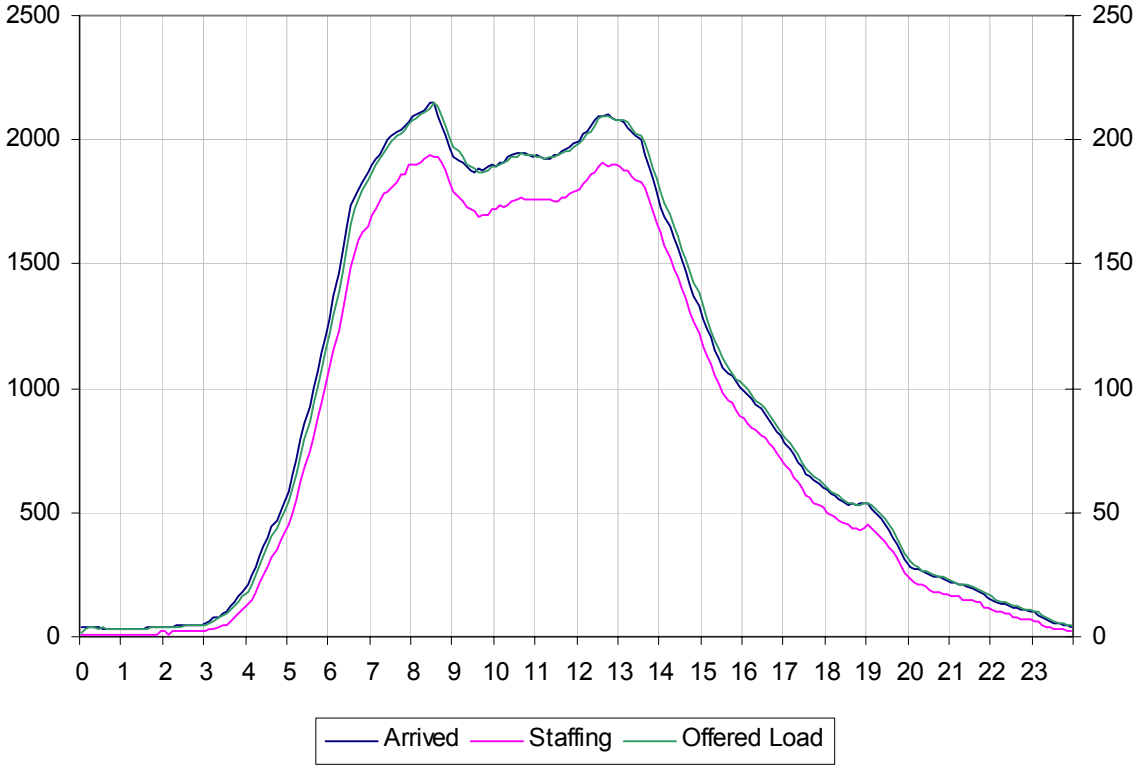


Figure 54: Target α=0.9: (1) Staffing level, offered load and arrival function, (2) average queue and waiting time



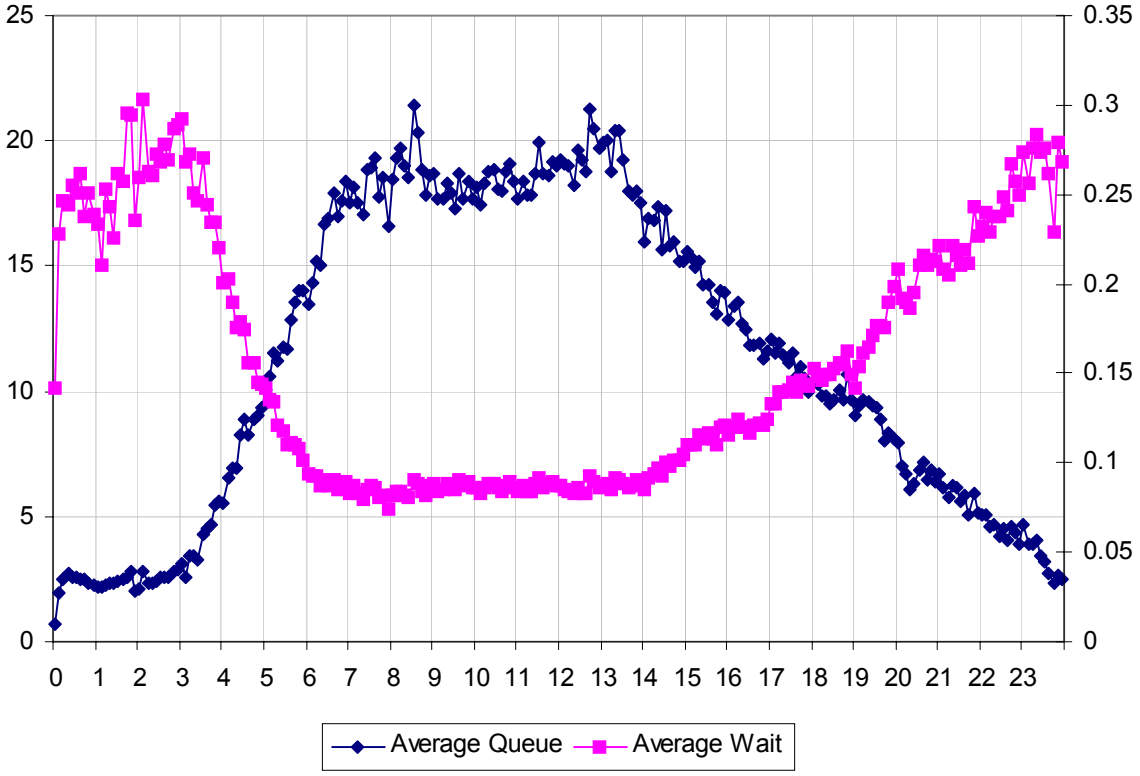
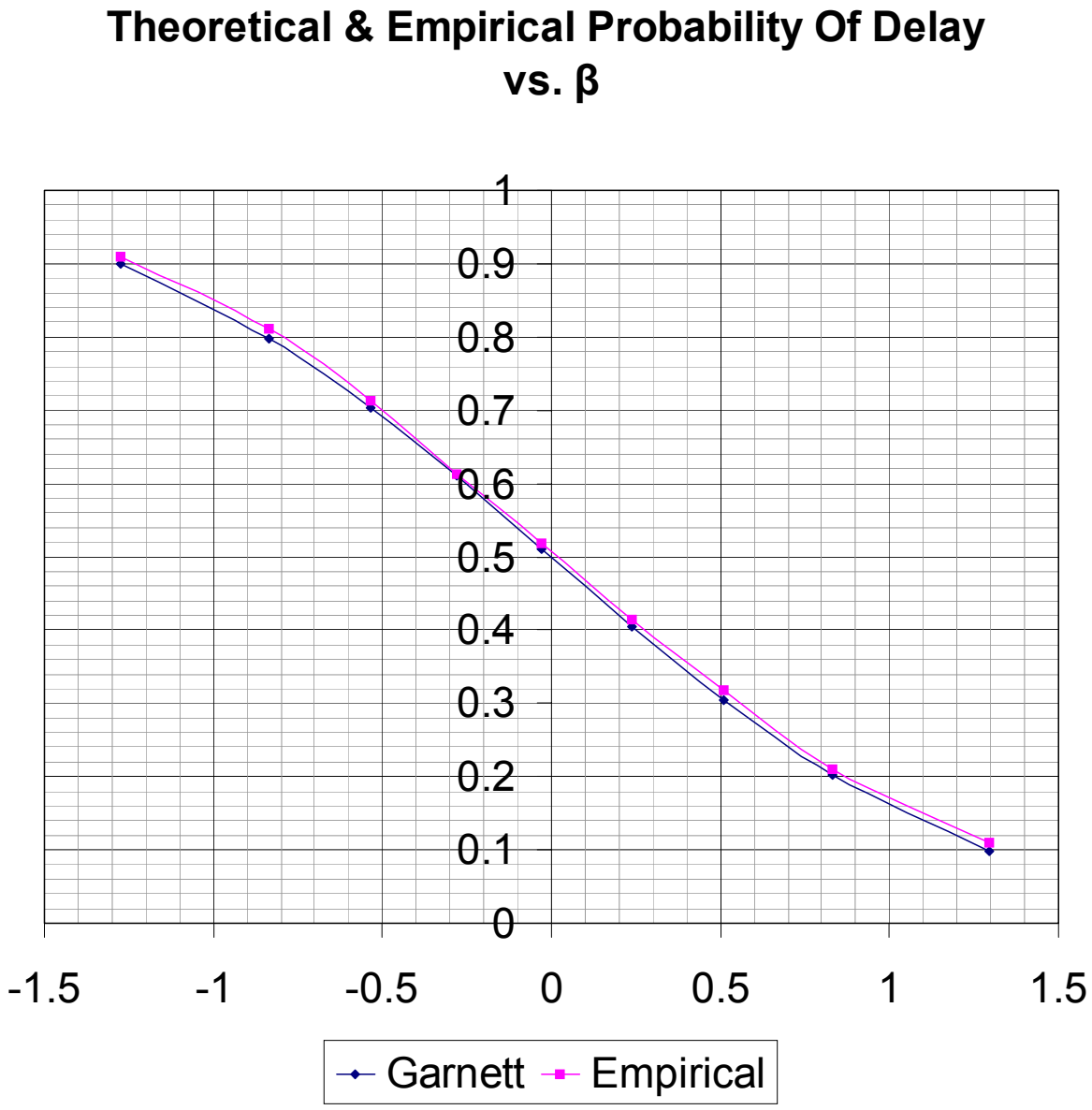


Figure 55: Comparison of empirical results with the Garnett approximation



7. Algorithm Dynamics

In this section we discuss the dynamics of the Iterative Algorithm. For this discussion we consider some observations regarding the dynamics of staffing levels during the algorithm, and its convergence to the desired $s_\infty(\cdot)$.

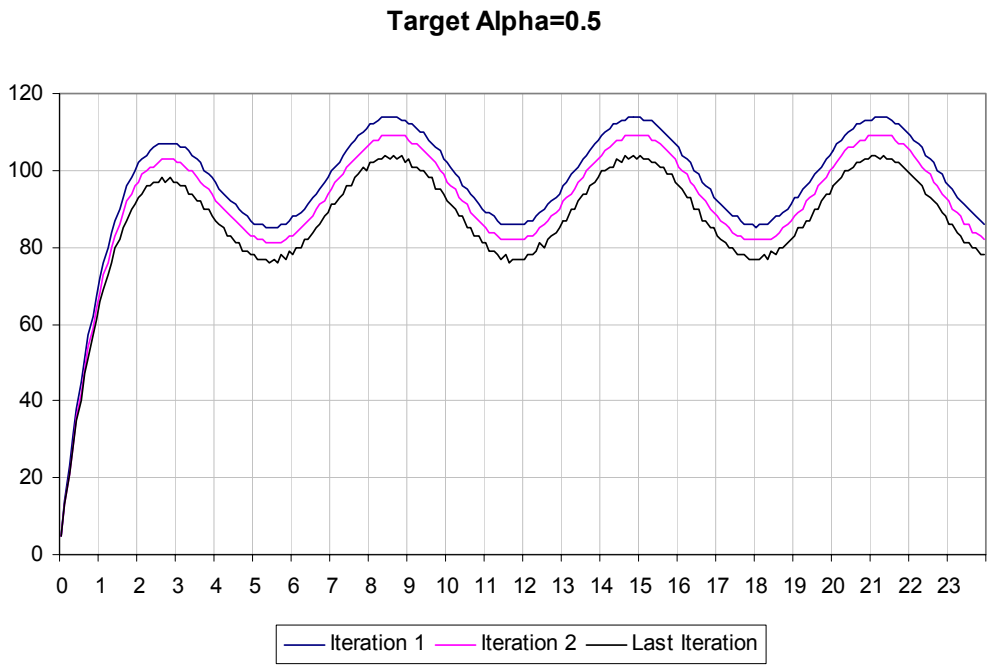
- **The relation between service and (im)patience rate** – whenever service rate exceeds (im)patience rate, we encounter an *oscillating dynamics* of staffing level during the algorithm; else we encounter *monotone dynamics*. This observation can be supported mathematically.

The monotone dynamics – denote $\{s_n(t)\}_{n=1}^{\infty}$ by the sequence of staffing level at time t along the algorithm's iterations. Then, in the monotone dynamics (when starting with $s_0(t) \equiv \infty$) $s_n(t)$ is monotone decreasing in n for all t, i.e. the following prevails:

$$s_n(t) \le s_m(t) \qquad \forall m > n \qquad (6.1)$$

Example for this dynamics can be viewed in Figure 56, where staffing level as obtained by various iterations along the algorithm is plotted for the case of arrival function is $\lambda(t) = 100 + 20 \cdot \sin(t)$, service times exponential having mean 1, and impatience times exponential having mean 0.1.

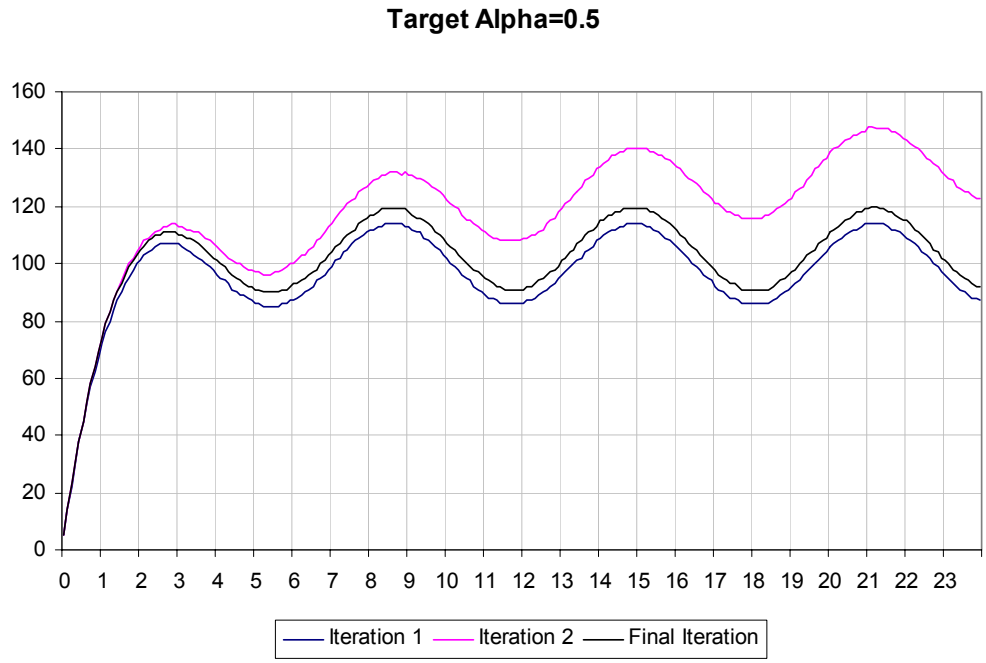
Figure 56: Staffing levels in the 1st, 2nd and last iterations. μ=1, θ=10.



The oscillating dynamics - $\{s_n(t)\}_{n=1}^{\infty}$ *as before. In the oscillating dynamics,* $s_n(t)$ *is oscillating for all t, i.e. there exist 2 sub-sequences* $\{s_k(t)\}_{k=2n}^{\infty}$ *and* $\{s_l(t)\}_{l=2n+1}^{\infty}$*, such that* $s_{2n}(t) \downarrow s_\infty(t)$ *and* $s_{2n+1}(t) \uparrow s_\infty(t)$

Example for this dynamics can be viewed in Figure 57, where staffing level as obtained by various iterations along the algorithm is plotted for the case of arrival function $\lambda(t) = 100 + 20 \cdot \sin(t)$, service times are exponential having mean 1, and no abandonment

Figure 57: Staffing levels in the 1st, 2nd and last iterations'. μ =1, θ=0.



- Target delay probability - the oscillating behavior and monotone behavior both tend to the extreme as target delay probability increases, i.e. the range of alternation of staff level increases as target delay probability increases, and thus convergence to $s_\infty(\cdot)$ requires many more iterations for large delay probabilities. For example, staffing level in the 1st and 2nd iterations, which form the range of the oscillating dynamics, are plotted for both target α=0.1 (figure) and α=-0.5 (figure) for the case of arrival function $\lambda(t)=100+20\cdot\sin(t)$, service times are exponential having mean 1, and no abandonment.

Figure 58: Range of staffing level for target α=0.1

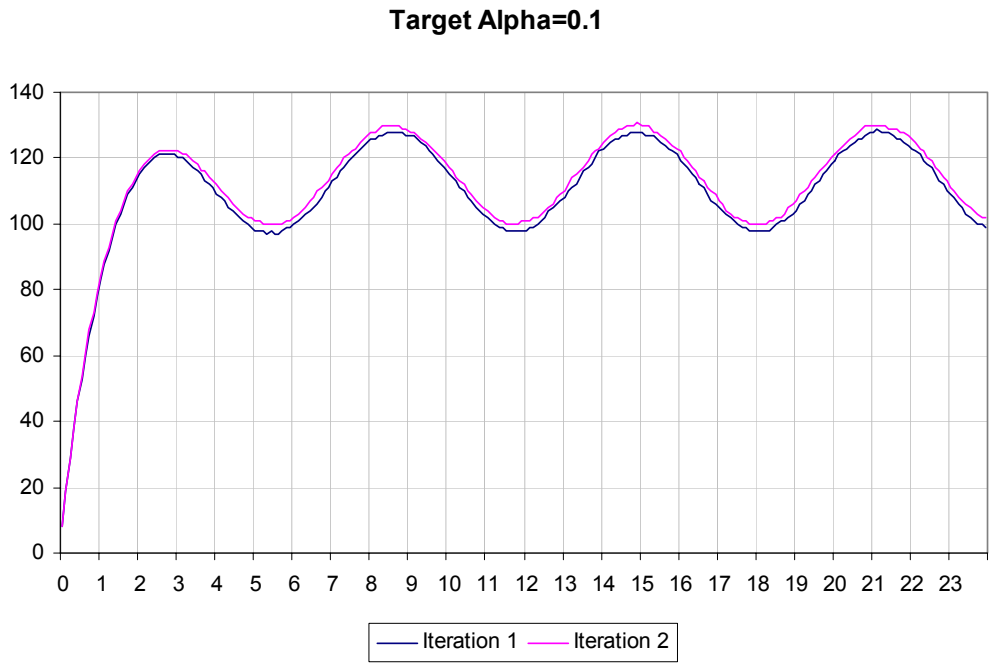
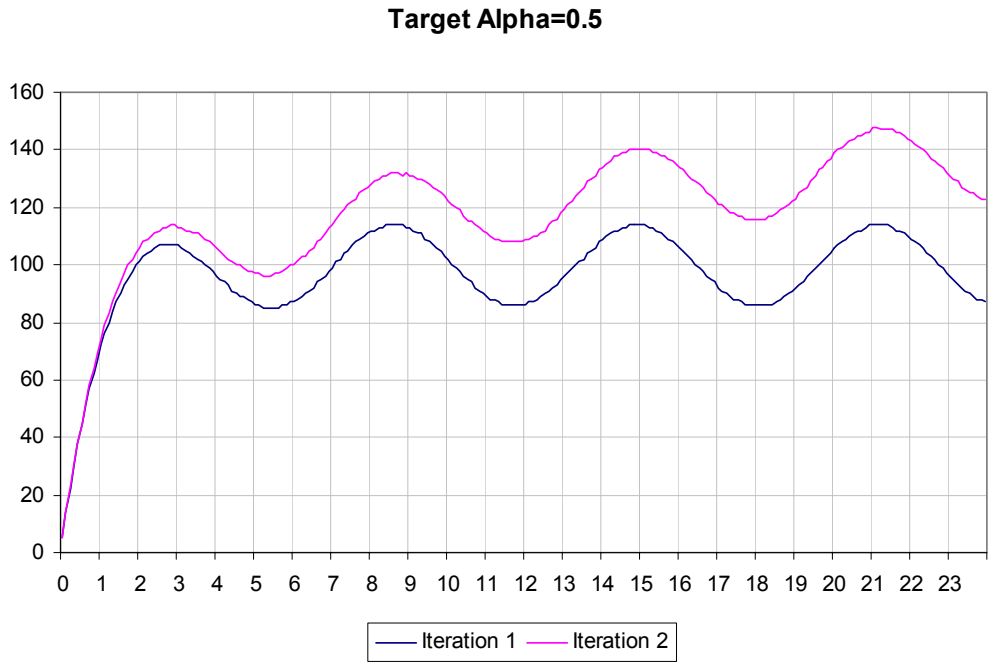
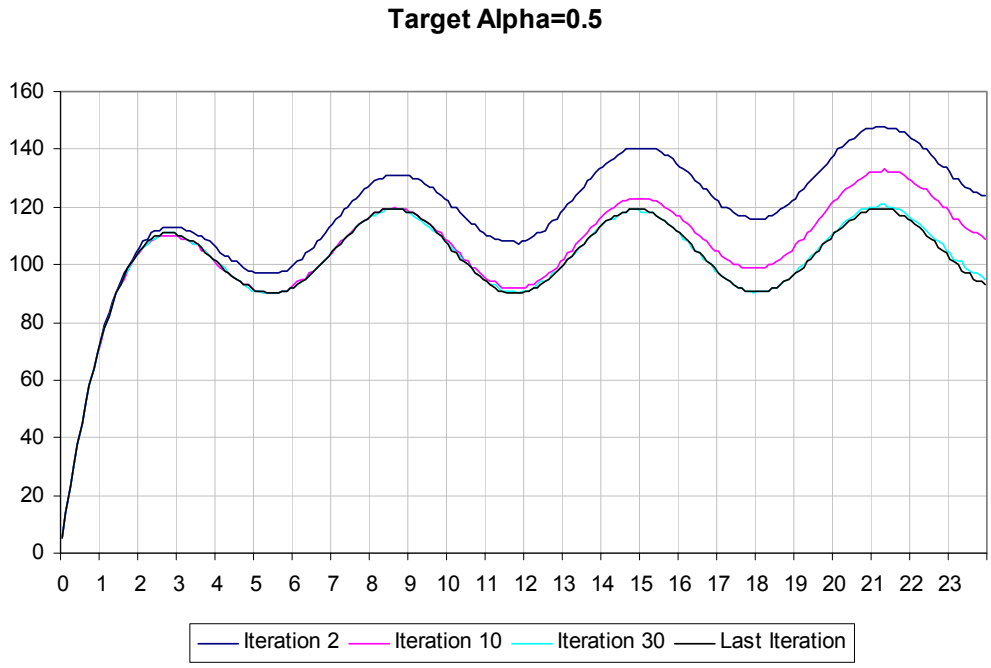


Figure 59: Range of staffing level for target α=0.5



- **Time-dependent convergence** – Let I_t be $\inf\left\{j : s_i(t) = s_j(t) \quad , \forall i \geq j\right\}$, then $I_{t_2} \geq I_{t_1} \quad \forall t_2 > t_1$. An illustration can be viewed in Figure 60.

Figure 60: Evolution of convergence during algorithm run-time



8. Summary and Future Research

An algorithm has been developed that generates staffing functions for which performance is stable in the face of time-varying loads. The results have been found to be remarkably robust, covering the ED, QD and QED operational regimes. Here are some natural "next-steps":

- Explore additional queueing systems. We already have partial (successful) results for deterministic service times (for which some theory has been developed and our algorithm can be compared against) and log-normal services (for which no theory is available). More systems to analyze appear in Mandelbaum et. al. (1998), for example queues with abandonment and retrials and priority queues.

- Analyze convergence of the algorithm rigorously. The monotone and oscillating convergence, displayed in Section 7, can be explained via stochastic-ordering. For example, oscillating convergence arises when the total-number-in system is stochastically decreasing in the number of servers, and there is monotone convergence if it is stochastically increasing. (Interestingly, the latter does arise naturally: consider, for example, a queue with abandonment in which average (im)patience is shorter than service time; then, more servers would lead to more customers in the system.)

- Within the mathematical framework of Service Networks (in Mandelbaum et. el.), we would like to support the algorithm theoretically. Namely, one can hopefully prove that, under proper rescaling, the target probability of delay indeed converges to a time-constant, under some square-root staffing, This has been done already for the moderate-impatience case (equal average patience and service), but further understanding is still required for the general case.

References:

Gans, N., Koole, G., Mandelbaum, A**. Telephone Call Centers: Tutorial, Review and Research Prospects**. Invited review paper by *Manufacturing and Service Operations Management* (M&SOM), 5 (2), 2003

Garnett O., Mandelbaum A. and Reiman M. **Designing a Call Center with Impatient Customers**. *Manufacturing and Service Operations Management*, **4(3)**, 208-227, 2002

Jennings O., Mandelbaum A., Massey W. and Whitt W. **Server Staffing to Meet Time-Varying Demand.** *Management Science, 42:10 (October 1996), pp. 1383-1394*

Mandelbaum A., Massey W.A. and Reiman M. **Strong Approximations for Markovian Service Networks**. *Queueing Systems: Theory and Applications (QUESTA)*, **30**, 149-201, November 1998

Green L., Kolesar P., Soares J. **Improving the SIPP Approach For Staffing Service Systems That Have Cyclic Demand.** *Management Science.* July-August 2001