Bhulai, Sandjai; Farenhorst-Yuan, Taoying; Heidergott, Bernd; van der Laan, Dinard

Working Paper

Optimal Balanced Control for Call Centers

Tinbergen Institute Discussion Paper, No. 10-013/4

Provided in Cooperation with:
Tinbergen Institute, Amsterdam and Rotterdam

Suggested Citation: Bhulai, Sandjai; Farenhorst-Yuan, Taoying; Heidergott, Bernd; van der Laan, Dinard (2010) : Optimal Balanced Control for Call Centers, Tinbergen Institute Discussion Paper, No. 10-013/4, Tinbergen Institute, Amsterdam and Rotterdam

This Version is available at:
https://hdl.handle.net/10419/86918

TI 2010-013/4
Tinbergen Institute Discussion Paper

Optimal Balanced Control for Call Centers

Sandjai Bhulai[1]
Taoying Farenhorst-Yuan[1]
Bernd Heidergott[1,2]
Dinard van der Laan[1,2]

[1] *VU University Amsterdam;*
[2] *Tinbergen Institute.*

Tinbergen Institute

The Tinbergen Institute is the institute for economic research of the Erasmus Universiteit Rotterdam, Universiteit van Amsterdam, and Vrije Universiteit Amsterdam.

Tinbergen Institute Amsterdam
Roetersstraat 31
1018 WB Amsterdam
The Netherlands
Tel.: +31(0)20 551 3500
Fax: +31(0)20 551 3555

Tinbergen Institute Rotterdam
Burg. Oudlaan 50
3062 PA Rotterdam
The Netherlands
Tel.: +31(0)10 408 8900
Fax: +31(0)10 408 9031

Optimal Balanced Control for Call Centers

Sandjai Bhulai * Taoying Farenhorst-Yuan †
Bernd Heidergott ‡ Dinard van der Laan §

January 18, 2010

Abstract

In this paper we study a challenging call center operation problem. The goal of our analysis is to identify an optimal policy for allocating tasks to agents. As a first step, we discuss promising randomized policies and use stochastic approximation for finding the optimal randomized policy when implemented via a Bernoulli scheme. As we will show in this paper, the performance of the call center can be improved if the randomized policy is implemented by a deterministic sequence satisfying some regularity conditions. Such sequences are called balanced and we will show that implementing randomized policies by balanced sequences provide an additional step in optimization and control. This motivates our new approach where we avoid the intermediate step of first finding an optimal randomized control and directly find the optimal balanced sequence for control of the call center via stochastic approximation.

Keywords: Call Center; Measure–Valued Differentiation; Balanced Sequence, Optimization.

*Faculty of Sciences, Vrije Universiteit Amsterdam,
Email: sbhulai@few.vu.nl

†Department of Econometrics and Operations Research, Vrije Universiteit Amsterdam, Email: tyuan@feweb.vu.nl

‡Tinbergen Institute, and Department of Econometrics and Operations Research,Vrije Universiteit Amsterdam, Email: bheidergott@feweb.vu.nl

§Tinbergen Institute, and Department of Econometrics and Operations Research, Vrije Universiteit Amsterdam, Email: dalaan@feweb.vu.nl

1 Introduction

The call center industry is a large and rapidly growing sector that provides a variety of services to organizations and customers, e.g., handling of orders, complaints, and questions, providing technical support, etc. This is an integral part of the customer relationship management of many organizations, which is an easy concept but a hard reality. As more and more organizations diversify and their products and services become more complex, the efficient operational control of call centers has grown in complexity as well. In particular, efficient workforce management has been the center of attention, since this forms a substantial part of the operational costs. Generally speaking, the objective of the call center is to constrain the expected waiting time in the queue of an arbitrary customer. On the other hand, modern call centers are also faced with other tasks with a less strict requirement, such as emails, web messages, and outbound calls. This work typically has a less tight constraint, and therefore the objective of the call center is to serve as many jobs as possible per time unit (called throughput).

In this paper, we study a call center where we can distinguish the two types of work by type 1 (incoming calls) and type 2 (emails, outbound calls, etc.) jobs. Both job types have different service requirements, and there is a common pool of agents to serve both of them in a non-preemptive regime. The system is depicted in Figure 1. More specifically, the two types of jobs have independent exponentially distributed service requirements with rates μ_1 and μ_2, and we let $\mu_1 \neq \mu_2$. Type 1 jobs arrive according to a Poisson process with rate λ, and there is an infinite waiting capacity for jobs that cannot be served yet. There is an infinite supply of type 2 jobs. There are a total of C identical agents (servers). The question is how to efficiently use the workforce to maximize the throughput of type 2 jobs while guaranteeing that the long-term average waiting time of type 1 jobs is below a predefined constant α.

While this type of system has been intensively studied in the literature, no exact optimal policy has been identified yet in the case of unequal service rates (i.e., $\mu_1 \neq \mu_2$), which is due to the complexity of the solution and the model. See Section 2 for a detailed discussion of the available literature.

In this paper, we develop a new approach to finding an optimal control for the described call center by means of deterministic sequences enjoying the property of "balancedness". Balanced sequences have been introduced for implementing randomized policies. We set off with the classical approach

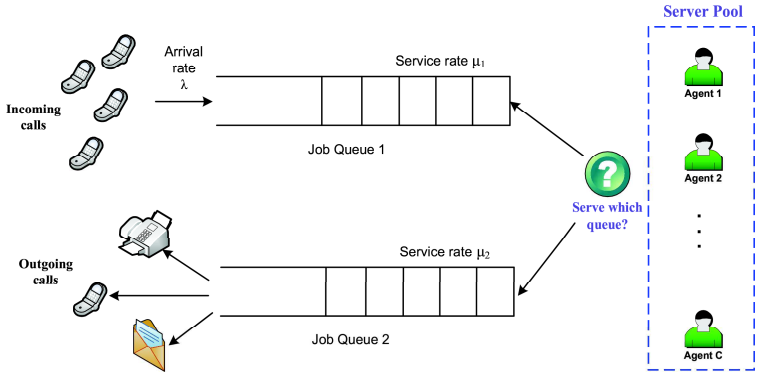


Figure 1: A call center system

by first introducing randomized policies for the call center control problem via the Bernoulli scheme. Secondly, we apply stochastic approximation techniques to find the optimal randomized policy. Finally, we model the optimal randomized policy by a balanced sequence to show that this leads to an even better system performance.

A key observation of our analysis is that the overall optimal randomized sequence can be improved if the optimization is *directly* carried out over the set of balanced sequences. To this end we will introduce an estimator for the sensitivity of the call center performance with respect to the density of the balanced sequence. This leads to the following **optimal balanced control** (OBC) approach:

1. **Identifying Policies:** Define competing deterministic policies for the control problem at hand.

2. **Stochastic Optimization:** Apply stochastic approximation techniques in order to find the optimal balanced sequence.

3. **Implementation:** Give the deterministic balanced sequence resulting from the previous step to the call center manager for implementation.

The OBC method transcends to other call center problems, which is an important methodological contribution in call center management. OBC is focused on mixing well-understood policies to achieve multiple objectives, and finds mixing parameters to randomize insightful policies in a deterministic fashion. This has a major impact on control policies in hardware (such as automatic call distributors [ACDs] in call centers).

This paper is organized as follows. A detailed review literature is provided in Section 2. In Section 3 we present a challenging call center operation problem, which we will use throughout the paper for numerical experiments. In Section 4 we address optimization of the call center for Bernoulli scheme, and we will provide a phantom gradient estimator. In Section 5, we introduce balanced sequences, and show that applying a balanced sequence to the control of the call center yields better performance than when a Bernoulli control policy is used. Moreover, our approach to finding the optimal balanced sequences is presented. Extended numerical results are provided in Section 6.

2 Literature Review on MDP and Balanced Sequences

2.1 The Call Center Model in the Literature

The call center model under discussion has been initially studied for the case of equal service rates (i.e., $\mu_1 = \mu_2$) by Bhulai and Koole (2003), Gans and Zhou (2003), Perry and Nilsson (1992), and by Gurvich et al. (2008) who study multiple classes of calls. Shumsky (2004), Stanford and Grassmann (2000), and Wallace and Whitt (2005) consider fixed, static priority policies. A similar approach is adopted by Koole and Talim (2000) and Franx et al. (2006), who provide an approximate analysis of the overflow behavior from one pool of agents to another. For a literature survey on asymptotic heavy-traffic regimes we refer to Koole and Mandelbaum (2002) and Gans et al. (2003).

Armony and Maglaras (2004) study a problem which is related to our model but in many ways different. The authors assume that the service rates of both job types are equal and do not address the more difficult case of unequal service rates. Moreover, they perform an asymptotic analysis based on 'many-server' limits. A similar remark holds for Dai and Tezcan (2008),

and Gurvich, Armony and Mandelbaum (2008). However, our method does not require an asymptotic analysis and provides a policy that works well in all regimes, not only the QED-regime or the Halfin-Whitt regime. This is a much stronger result especially when practically applied to smaller call centers with a mix of heterogeneous call types.

2.2 Randomized Policies and MDP

A standard approach for deriving effective scheduling and routing policies for workforce management in call centers is via dynamic programming, respectively, Markov decision processes. This technique results in dynamic policies that depend on the current state of the call center, i.e., the number of calls currently in service and the number of calls in the queues distinguished by their type if there are multiple customer classes. Unfortunately, the identification of effective scheduling policies via dynamic programming is often impractical, both analytically and numerically, due to the dimensionality of the state space. Hence, standard algorithms such as value iteration or policy iteration for computing optimal policies break down. Moreover, in a constrained problem the optimal policy is usually found within the class of *randomized policies* as opposed to the class of deterministic policies (see Altman (1999)). This provides another reason why standard techniques fail, since they are not tailored to find the optimal randomized policy.

A randomized policy introduces a probabilistic law over the set of possible actions and tries to find the optimal distribution so that the (steady-state) performance is maximized. In the simplest case, there is in each state the choice between two actions, say a_1 and a_2. Both actions define a particular Markov kernel, say a_1 yields P and a_2 yields Q. The randomized policy is then modeled through

$$Q_\theta = \theta P + (1-\theta)Q, \tag{1}$$

with $\theta \in [0,1]$. Denoting the stationary distribution of Q_θ by π_θ (existence assumed), the dynamic programming problem can be expressed as follows

$$\begin{array}{l} \max_{\theta\in[0,1]} \pi_\theta f \\ \text{s.t.} \quad \pi_\theta g \leq c\,, \end{array} \tag{2}$$

where f, g are performance indicators and c is a constant.

Given the optimal solution θ^* of the above optimization problem, the actual policy that has to be implemented is that at each transition moment

a coin is tossed with probability of success θ^*, and action a_1 is taken if the experiment yields "success" and a_2 is taken otherwise. This is called the Bernoulli scheme. This interpretation of θ^* has two major disadvantages: (i) The coin tossing mechanism introduces additional randomness to the system; and (ii) the implementation of the policy is somewhat awkward as, for example, a manager of a call center has to agree to let the control actions be governed by a coin-tossing experiment.

2.3 Balanced Sequences

This problem of implementing randomized policies has been acknowledged in the literature, see, for example, Altman and Shwartz (1993). To overcome the above drawbacks of randomized policies, we propose to use so-called *balanced sequences* for the modeling and interpretation of randomized MDPs. The basic idea of balanced sequences is a rather simple one. Suppose θ^* equals $2/5$, and let 1 refer to decision a_1 and 0 to decision a_2. A property of a (recurrent) balanced sequence for $2/5$ is that it is a deterministic sequence of zeros and ones for which in each subsequence of length 5 there are exactly 2 occurrences of a "one". More specifically, for $2/5$ a possible balanced sequence is $0\,0\,1\,0\,1\,0\,0\,1\,0\,1\,0\,0\,1\ldots$. Details on balanced sequences will be provided latter on in the paper.

Balanced sequences obviously overcome the drawback of a randomized policy that for the later one control decisions are drawn according to a random mechanism, see (ii) above. An even more interesting fact is that the variance reduction induced by balanced sequences, see (i) above, can lead to better performance than a straightforward implementation of the corresponding randomized policy. For some special cases this could be proved with mathematical rigor, see Altman et al. (2003) and the references therein.

Balanced sequences have been studied for a long time (see, e.g., Morse and Hedlund (1940)) and many properties were derived but this was not in the context of optimal control. However, in Hajek (1985) it was proved for some specific admission control problem that the optimal control sequence $U = (u_1, u_2, \ldots)$ is within this subset of sequences with "good balance". After that control by such sequences have been applied to more scheduling, admission and routing problems in the area of queueing and discrete-event systems, see, for example, Altman et al. (2002), Altman et al. (1998), Altman et al. (2000a), Altman et al. (2000b), Shinya et al. (2004).

Using the concept of multimodularity, Altman et al. (2003) gives an

overview of control problems for which optimality of such sequences follows. However, quite some assumptions and specific type of control policies were needed to apply the concept of multimodularity. First it was used for admission control and then extended to some specific routing and/or polling problems. In this paper we apply these sequences with "good balance" to problems which do not fall within the framework for which multimodularity was established in Altman et al. (2003). Therefore we do not know a priori that the optimal control sequence is within the subset of sequences with "good balance". However, our objective is to show that restricting the optimization to such sequences is still very useful since the obtained policy outperforms other well-known and applied policies such as the Bernoulli policies.

3 The Call Center Operation Optimization Problem

A detailed description of the model will be given in Section 3.1. In Section 3.2 and Section 3.3 we will introduce two randomized policies for the call center operation problem. The optimization problem will be presented in Section 3.4.

3.1 The System Process

We model the system through the queue length process embedded at event epochs, where each arrival of a job, the finishing of a service by an agent and the assignment of a job to an agent is called an event. Specifically, we introduce the following variables:

- $L_q(n)$, the queue length just before the nth event;
- $S_1(n)$, the number of type 1 jobs being in service at the nth event;
- $S_2(n)$, the number of type 2 jobs being in service at the nth event;
- $\tau(n)$, the time at which the nth event occurs, with $\tau(0) = 0$;
- $T(n)$, the time elapsed between the $(n-1)$st and the nth event; more specifically, $T(n) = \tau(n) - \tau(n-1)$, for $n \geq 1$.

We describe the system by a Markov chain:

$$X \triangleq \Big\{ X(n) \triangleq \Big(L_q(n), S_1(n), S_2(n), \tau(n), T(n) \Big) : n \in \mathbb{N} \Big\},$$

and denote its state space by $S \triangleq \mathbb{N} \times \{0, \ldots, C\}^2 \times [0, \infty)^2$.

The objective for type 2 jobs is to maximize its throughput, i.e., to serve on average per unit of time as many type 2 jobs as possible, of course at the same time obeying the constraint on type 1 waiting time. Due to the fact that we are considering long-term average performance it is only optimal to schedule jobs at completion or arrival instants. Indeed, if it is optimal to keep a server idle at a certain instant, then this remains optimal until the next event in the system. Therefore it suffices to consider the system only at completion or arrival instants. Note that the policy for assigning jobs to vacant servers is not specified at the moment and we will discuss in the subsequent section two standard policies.

3.2 Trunk-θ Reservation Policy

In Bhulai and Koole (2003), the optimal policy for the case of equal service requirements (i.e., $\mu_1 = \mu_2$) is trunk reservation. In this study we will explore the performance of a trunk reservation for the case of unequal service requirements. For trunk reservation, there will be always K agents reserved for type 1 jobs. Only when there are more than K agents idle, those extra idle agents will serve type 2 jobs. Figure 2 displays the behavior of the waiting time of type 1 jobs and the throughput of type 2 jobs when K varies. In this example with time unit of one minute the used values of parameters are $\lambda = 1/2, \mu_1 = 4/10, \mu_2 = 3/10$, and $C = 5$. We see that the average waiting time decreases slowly, while the throughput decreases almost linearly.

The figure could be interpreted as follows, for a given waiting time constraint α (the guaranteed maximum average waiting time of type 1 jobs), one can read the optimal threshold value K. Suppose $\alpha = 1/6$ minutes, then the optimal K value is between 1 and 2. So we would like to have a policy where the threshold of trunk reservation is randomized between $K_1 = 1$ and $K_2 = 2$. This means that at an arrival or service completion instant, with probability θ the trunk reservation policy with threshold K_1 will be used, and with probability $(1 - \theta)$ the one with threshold K_2 will be used. We call such policy a *trunk-θ reservation policy.*

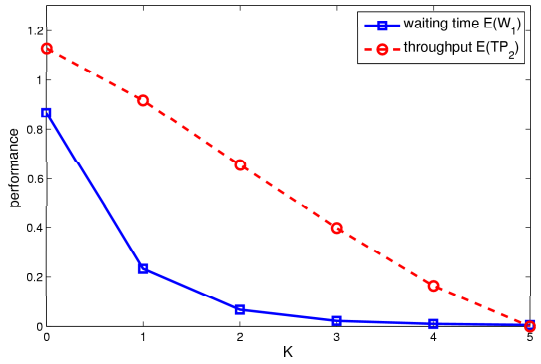
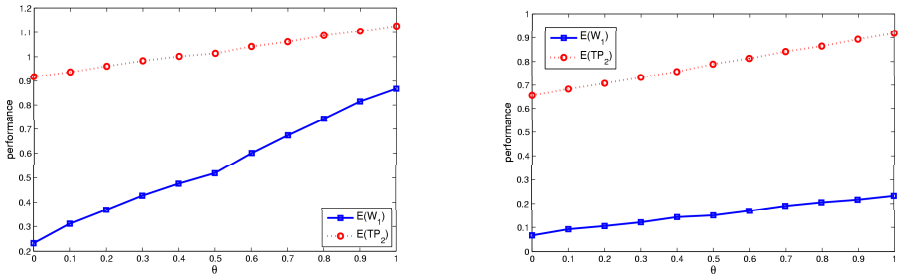


Figure 2: The performance of a trunk reservation policy as a function of K

The performance of the trunk-θ reservation policy with $K_1 = 0, K_2 = 1$ are displayed in Figure 3(a), and with $K_1 = 1, K_2 = 2$ are shown in Figure 3(b). As we can see from the figures, both the average waiting time and the throughput increase as θ increases. Thus the throughput of type 2 job reaches its maximum at the point where the average waiting of the type 1 job is equal to the constraint α.



(a) $K_1 = 0, K_2 = 1$ (b) $K_1 = 1, K_2 = 2$

Figure 3: The performance of a randomized trunk-θ reservation policy

For the trunk-θ reservation policy, P denotes the transition kernel of a Markov chain when the trunk reservation policy with threshold K_1 is used and Q denotes the transition kernel of a Markov chain when the trunk reservation policy with threshold K_2 is used. Thus the Markov kernel corresponding

to the randomized trunk reservation policy is given by

$$Q_\theta = \theta P + (1-\theta)Q.$$

Solving the dynamic programming problem stated in (2) will lead to an optimal randomized policy, i.e., it will find the optimal value for θ that maximizes the throughput while satisfying the waiting time constraint.

3.3 Flow Rate Policy

Bhulai and Koole (2003) noted that if a agent becomes idle while there are type 1 jobs waiting in the queue, then the action that schedules a type 1 job is among the set of optimal actions. Furthermore, since we are interested in the long-term throughput, delaying the processing of a type 2 job does not change the performance for this class. Based on these reasons, we propose the following flow rate policy: if there is a type 1 customer in the queue, idle agents will always serve them; only when the queue is empty, idle agents have probability θ to serve type 2 customers. More specifically: at each arrival and departure instant, assume that there are N idle agents, and there are M type 1 jobs in the queue. If $N \leq M$, all idle agents will serve type 1 jobs. If $N > M$, M idle agents will serve type 1, and at the same time, each of the extra $N - M$ idle agents will simultaneously and independently flip a coin, where with probability θ this idle agent will serve a type 2 job and with probability $1-\theta$ this agent will remain idle in order to wait for the next possible arriving type 1 job. We call this the *θ-flow rate policy.*

Figure 4 displays the average waiting time of type 1 jobs and the throughput of type 2 jobs as a function of θ while the flow rate policy is used. The system parameters are $\lambda = 1/2, \mu_1 = 4/10, \mu_2 = 3/10$, and $C = 5$. As we can see from Figure 4, both the waiting time and the throughput increase when θ increases. Intuitively θ presents the service capacity assigned to type 2 jobs. We tend to increase θ to obtain a larger throughput for type 2 jobs. On the other hand, we cannot choose θ too large since the constraint on the waiting time of type 1 job has to be satisfied.

Denote P as the transition kernel of a Markov chain where idle agents first check whether there are type 1 jobs, if there are, they will serve a type 1 job. If there are no type 1 jobs, they will serve a type 2 job. Denote Q as the transition kernel of a Markov chain where idle agents check also whether there are type 1 jobs, if there are, they will serve a type 1 job. If there are

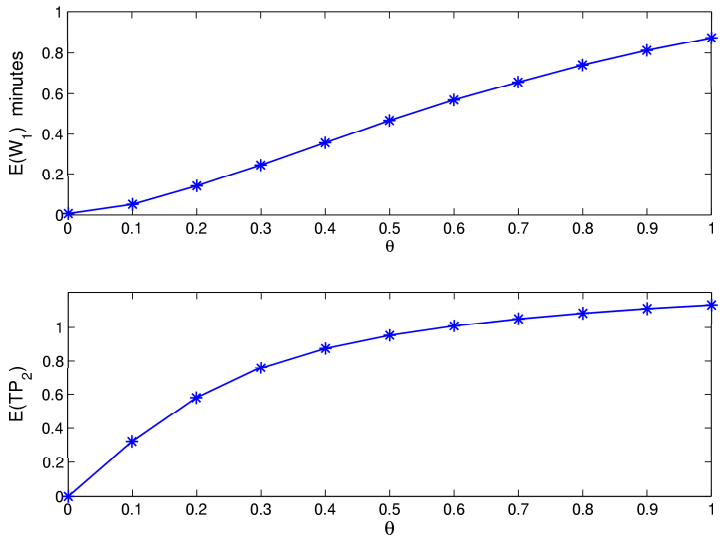


Figure 4: The waiting time and the throughput of the system with the flow rate policy

no type 1 jobs, they will always stay idle. Then the Markov kernel of the system under the θ-flow rate policy is given by

$$Q_\theta = \theta P + (1-\theta)Q \tag{3}$$

and solving the dynamic programming problem stated in (2) will lead to an optimal randomized policy, i.e., it will find the optimal value for θ.

Remark 1. *Note that the model put forward in (3) is simplified in the sense that transitions of the Markov chains are triggered by arrivals and service completions only; and, for the sake of simplicity, we have omitted specifying the detailed transition dynamic modeling the assignment of tasks to the agents.*

3.4 The Optimization Problem

Observe that both $\mathbb{E}[W_1(\theta)]$ and $\mathbb{E}[TP_2(\theta)]$ increase as θ increases for the flow rate policy as well as for the trunk-θ reservation policy. The optimization problem in (2) is therefore equivalent to solving the implicit equation

$\mathbb{E}[W_1(\theta)] = \alpha$, which has the same solution as $\mathbb{E}[(W_1(\theta) - \alpha)^2] = 0$. Since $\mathbb{E}[(W_1(\theta) - \alpha)^2] \geq 0$, the constrained optimization problem in (2) is equivalent to the following unconstrained optimization problem

$$\min_{\theta \in [0,1]} \mathbb{E}[A(W_1(\theta) - \alpha)^2],$$

where A is a scaling factor to compensate for the fact that $(W_1(\theta) - \alpha)^2$ is usually small.

For $s = (l, s_1, s_2, \tau, t) \in S$, let

$$g(s) = l \cdot t \tag{4}$$

denote the queue length mapping.

The long-run average queue length $L_q(\theta)$ is then given as:

$$L_q(\theta) = \lim_{n \to \infty} \frac{\sum_{i=1}^{n} g(X_\theta(i))}{\tau(n)}, \tag{5}$$

where we introduce the θ-subscript to the system process $X(n)$ to indicate the θ-dependence of the process. By Little's law:

$$W_1(\theta) = \frac{L_q(\theta)}{\lambda} = \lim_{n \to \infty} \frac{\sum_{i=1}^{n} g(X_\theta(i))}{\lambda \tau(n)}.$$

Denote the long-run cost function by $C(\theta)$:

$$C(\theta) = \mathbb{E}\left[A\left(W_1(\theta) - \alpha\right)^2\right] = \lim_{n \to \infty} A\left(\frac{\sum_{i=1}^{n} g(X_\theta(i))}{\lambda \tau(n)} - \alpha\right)^2. \tag{6}$$

To keep the presentation simple, we will approximate the long-run average cost by a transient simulation experiment over a fixed time horizon T. Let $M(T, \theta)$ denote the (random) number of the events until time T and choose T sufficiently large so that

$$C(\theta) \approx C(T, \theta) = \mathbb{E}\left[A\left(\frac{1}{\lambda T} \sum_{i=1}^{M(T,\theta)} g(X_\theta(i)) - \alpha\right)^2\right]. \tag{7}$$

This way we avoid a discussion on simulation techniques of steady-state characteristics which would distract from the main analysis of the paper. Note that the model in (7) allows for letting T denote the operation period of the call center, for example, the opening time during a week-day.

4 Optimizing the Bernoulli Control Policy

As we have explained in the previous section, the constrained optimization problem (2) is equivalent to the following unconstrained version of the optimization problem:

$$C(\theta^\star) = \min_{0\le\theta\le 1} C(\theta)\,. \tag{8}$$

In words, the optimal scheduling policy is given by the value of θ that yields the minimal long-run average cost $C(\theta) = \mathbb{E}_\theta\,[A(W_1(\theta) - \alpha)^2]$. In general, call centers violate the rather restrictive conditions for solving (8) analytically. Thus one has to resort to simulation for obtaining the optimal value of the control parameter. A standard method for finding the optimal value in an iterative procedure is stochastic approximation (SA). The general form of SA is as follows

$$\theta_{k+1} \;=\; \Pi_{(0,1)}\big(\theta_k - a_k \nabla C_k\big)\,, \tag{9}$$

where θ_k is the parameter vector at the beginning of iteration k, ∇C_k is an estimate of $\nabla C(\theta_k)$ (the gradient of $C(\theta_k)$), a_k is a (positive) sequence of step sizes, and $\Pi_{(0,1)}$ is the projection onto $(0, 1)$. It can be shown that under suitable conditions $\theta_k \to \theta^*$ for k towards ∞ with probability one.

The randomized policy between P and Q is given by the convex combination

$$Q_\theta = \theta P + (1-\theta)Q,$$

and differentiating the above expression yields

$$Q'_\theta = P - Q.$$

The above representation of the derivative of Q_θ with respect to θ as difference between two Markov kernels is called a *measure-valued derivative* in the literature. Measure-valued differentiation (MVD) is an extension of the concept of weak differentiation introduced by Pflug, see Pflug (1996). The theory of MVD for Markov chains as worked out in Heidergott and Vázquez-Abad (2008, 2006), Heidergott and Hordijk (2003), Heidergott et al. (2006) is an operator approach to calculating derivatives of Markov kernels such as Q_θ.

An MVD based estimator evaluates the gradient by the difference between the performance evaluated for variants of the processes, called *phantom processes*. Evaluating the difference between sample paths of the same

Markov chain with different initial values are also called *perturbation realization factors* in the literature, see Cao (2007). There is a close relationship between gradient estimation via realization factors and that via measure-valued differentiation, see Heidergott and Cao (2002).

A phantom estimator for the Markov chain with kernel Q_θ can be obtained as follows. We simulate the system process $X_\theta(n)$ according to Q_θ. At a particular transition of the system process we split the sample path. We do this by performing this particular transition for one sub–path according to the positive part of Q'_θ, that is, P, and for the other sub–path according to the negative part of Q'_θ, that is, Q. Then, we again generate the transitions of the two processes according to Q_θ. We estimate the gradient by the difference between the performance evaluated for both variants of the processes. More specifically, we introduce "plus" and "minus" processes $\{X_\theta^\pm(s,k) : k \geq 1\}$, called *phantoms*, as follows. At a particular state s, the nominal process "splits" in three different trajectories. The transition from s to $X_\theta^+(s,1)$ is governed by P and that from s to $X_\theta^-(s,1)$ by Q, respectively. For $k > 1$, the transition from $X_\theta^\pm(s,k-1)$ to $X_\theta^\pm(s,k)$ is governed by Q_θ. That is, from then on, the remaining transitions are governed by Q_θ, and the phantoms and the nominal process show statistical identical transition behavior.

Let $M^\pm(s,T,\theta)$ denote the number of events until time T of the "plus" and "minus" phantoms where the phantoms are generated from the state s in the nominal path, and let

$$W_1^\pm(s,j) = \frac{1}{\lambda T} \sum_{k=1}^{M^\pm(s,T,\theta)-j+1} g(X_\theta^\pm(s,k))$$

denote the waiting time estimate for the phantoms originated from perturbing the transition out of state $s = X_\theta(j)$. With this notation, the performance measure $A(W_1(\theta)-\alpha)^2$ obtained by the "plus" and "minus" phantoms where the phantom generated from perturbing from the jth transition is given by

$$A(W_1^\pm(s,j)-\alpha)^2 = A\left(\frac{1}{\lambda T} \sum_{k=1}^{M^\pm(s,T,\theta)-j+1} g(X_\theta^\pm(s,k)) - \alpha\right)^2. \qquad (10)$$

The difference between the cumulative costs over the "plus" and "minus" phantoms for the transient simulation is given by

$$D(s,j) \triangleq A(W_1^+(X_\theta(s,j)) - \alpha)^2 - A(W_1^-(X_\theta(s,j)) - \alpha)^2, \qquad (11)$$

with $s \in S$ and $j \geq 1$. The *Phantom Estimator* (PhE) is obtained by:

$$\mathrm{PhE}(T) \triangleq \sum_{k=1}^{M(T,\theta)} D(X_\theta(k), k). \tag{12}$$

The above expression for $\mathrm{PhE}(T)$ can be phrased as follows: at each state of the nominal process a "plus" and a "minus" phantom is generated and the derivative is estimated by the scaled difference between the cumulative costs over the "plus" and "minus" phantoms. By standard theory of MVD, see Heidergott and Vázquez-Abad (2006), it then holds that

$$\frac{d}{d\theta}\mathbb{E}[A(W_1(\theta) - \alpha)^2] = \mathbb{E}[\mathrm{PhE}(T)].$$

In words, the phantom estimator is unbiased for the derivative of the weighted average waiting costs.

4.1 Coupling schemes

A straightforward application of the phantom estimator requires to simulate the phantoms until they reach the fixed time horizon T. Fortunately, due to the strong Markov property and the fact that service and interarrival times are exponential (and thus memoryless), as soon as the phantoms reach the same physical state, i.e., the number of jobs waiting and the number of type 1 and type 2 jobs in service, then their future is, given that state, independent from the past. This allows to identify their sample paths from that state on. More precisely, let

$$X_\theta^{\pm}(n) = \Big(L_q^{\pm}(n), S_1^{\pm}(n), S_2^{\pm}(n), \tau^{\pm}(n), T^{\pm}(n)\Big),$$

for $n \in \mathbb{N}$. Then, as soon as for some n it holds that information on the physical state is identical, that is,

$$(L_q^+(n), S_1^+(n), S_2^+(n)) = (L_q^-(n), S_1^-(n), S_2^-(n)),$$

the future of the phantoms can be identified and we may set $X_\theta^+(m) = X_\theta^-(m)$, for $m > n$. This coupling is called *discrete time coupling* (DTC) and the phantom estimator with this coupling scheme is called the *DTC-phantom estimator*.

Alternatively, a different coupling can be constructed elaborating on the fact that $X_\theta^+(n)$ and $X_\theta^-(n)$ are embedded chains of a continuous-time queue-length process. To see this, introduce the continuous-time physical process as follows. For $t \geq 0$ let

$$(L_q^\pm(t), S_1^\pm(t), S_2^\pm(t)) = (L_q^\pm(n), S_1^\pm(n), S_2^\pm(n)), \quad \tau^\pm(n-1) \leq t < \tau^\pm(n).$$

At a transition of the, say, "plus" phantom we compare the new state of the "plus" phantom with the state of the "minus" phantom at the same instance in time. Specifically, if either

$$\Big(L_q^+(\tau^+(n)), S_1^+(\tau^+(n)), S_2^+(\tau^+(n))\Big) = \Big(L_q^-(\tau^+(n)), S_1^-(\tau^+(n)), S_2^-(\tau^+(n))\Big)$$

or

$$\Big(L_q^+(\tau^-(n)), S_1^+(\tau^-(n)), S_2^+(\tau^-(n))\Big) = \Big(L_q^-(\tau^-(n)), S_1^-(\tau^-(n)), S_2^-(\tau^-(n))\Big),$$

then the physical states of both phantoms coincide at time $\tau^+(n)$ (respectively, $\tau^-(n)$) and we can couple the phantoms from that moment on. We call this the *continuous time coupling* (CTC) and the phantom estimator with this coupling scheme is called the *CTC-phantom estimator*.

Before we apply the above gradient estimators to finding the optimal θ for the Bernoulli sequence via the optimization procedure put forward in (9), we will first introduce the balanced control scheme in the subsequent section.

5 Balanced Control

5.1 Introduction

Recall that P and Q denote the two Markov kernels that are mixed to control the call center. A mixture of P and Q is represented by an infinite sequence $U := (u_1, u_2, \ldots)$ of zeros and ones where u_j corresponds to the policy that is applied at the j-th decision event for $j = 1, 2, \ldots$. Moreover, we let $u_j = 1$ correspond to applying P while $u_j = 0$ corresponds to applying Q at the j-th decision event. More formally, we denote a transition kernel obtained from mixing P and Q according to $u \in \{0, 1\}$ by $R(u)$ and define it as follows:

$$R(u) = \begin{cases} P, & \text{if } u = 1, \\ Q, & \text{if } u = 0. \end{cases} \tag{13}$$

For any $\theta \in [0,1]$ and $\phi \in \mathbb{R}$ a balanced sequence $U_{\theta,\phi}$ of density θ is obtained by putting

$$u_{\theta,\phi}(j) = \lfloor j\theta + \phi \rfloor - \lfloor (j-1)\theta + \phi \rfloor \text{ for } j = 1, 2, \ldots, \tag{14}$$

where $\lfloor x \rfloor$ is the largest integer smaller than or equal to x. Sequences constructed in this way are also called lower bracket sequences which are known to be regular, see Appendix 9 for details. The density θ uniquely determines the sequence modulo a shift and in fact by varying the so-called *initial phase* ϕ the sequence is only shifted. Thus varying ϕ in this construction does not change the performance of the corresponding balanced control policy and in practice any ϕ can be chosen, in particular, $\phi = 0$. For example, letting $\theta = 1/2$ and $\phi = 0$, yields the sequence $0, 1, 0, 1, \ldots$ whereas, for example, $\phi = 3/4$ yields $1, 0, 1, 0, \ldots$.

A key observation for balanced sequences is that when ϕ is taken to be a uniformly $[0,1]$ distributed random variable, then the marginal distribution of $u_{\theta,\phi}(j)$ is again a Bernoulli distribution. In this paper we will use this fact to simulate balanced control of rate θ as follows. Let Φ be uniformly distributed on $[0,1]$, and independent of everything else. Then for given $\theta \in [0,1]$ we have an infinite sequence of Markov kernels $\{R(U_{\theta,\Phi}(j))\}_{j=1}^{\infty}$ defined by

$$R(U_{\theta,\Phi}(j)) = U_{\theta,\Phi}(j)P + (1 - U_{\theta,\Phi}(j))Q, \tag{15}$$

where $U_{\theta,\Phi}(j)$ is defined in (14).

We call the random sequence $\{R(U_{\theta,\Phi}(j))\}_{j=1}^{\infty}$ of Markov kernels a *randomized balanced (P,Q)-policy of rate θ*. For a given realization $\phi \in [0,1)$ of random variable Φ, we get a corresponding deterministic sequence $\{R(U_{\theta,\phi}(j))\}_{j=1}^{\infty}$ of Markov kernels, where $R(U_{\theta,\phi}(j)) = u_{\theta,\phi}(j)P + (1 - u_{\theta,\phi}(j))Q$ for $j = 1, 2, \ldots$ as shown in Equation (15). This is called the *lower bracket (P,Q)-policy of rate θ and shift ϕ*. As shown in the following lemma, when the expected value of the Markov kernel defined in (15) is taken with respect to Φ it yields the convex combination $\theta P + (1-\theta)Q$. The proof of the lemma is provided in Appendix 10.

Lemma 2. *Let $\{R(U_{\theta,\Phi}(j))\}_{j=1}^{\infty}$ be a randomized balanced (P,Q)-policy of rate θ. Then for $j = 1, 2, \ldots$ we have that*

$$\mathbb{E}_{\Phi}[R(U_{\theta,\Phi}(j))] = \theta P + (1-\theta)Q.$$

Sampling over Φ, yields a sequence of Markov kernels $R(U_{\theta,\phi}(1))$, $R(U_{\theta,\phi}(2))$, ... that has marginal distribution $Q_\theta = \theta P + (1-\theta)Q$, which is identical to the Bernoulli mixture of P and Q. However, due to the construction in (14), the elements of $R(U_{\theta,\phi}(n))$ are not mutually independent, as opposed to the Bernoulli scheme, where an i.i.d. sequence $U_\theta(n)$ of Bernoulli random variables with probability $\mathbb{P}(U_\theta(n) = 1) = \theta$ is used and the resulting sequence of kernels $\{R(U_\theta(n))\}$ is an i.i.d. sequence.

5.2 Balanced Sequences are Better

For an infinite sequence $U = (u_1, u_2, \ldots)$ of zeros and ones let

$$s(k,n) = \sum_{j=k}^{k+n-1} u_j, \quad k \leq n,$$

denote the numbers of ones in the subsequence of length n beginning at the k-th element of U. With this notation $s(n) := s(1,n)$ is the number of ones in the prefix of length n of U. We say that $U = (u_1, u_2, \ldots)$ has *density* $\theta \in [0,1]$ if $\lim_{n\to\infty} s(n)/n$ exists and is equal to θ.

Let $\{U_\theta(n)\}$ be an i.i.d. Bernoulli sequence with probability of success θ. From the strong law of large numbers it follows that $\{U_\theta(n)\}$ has density θ almost surely. For the sequence constructed in (14) it can be shown that that $\mid s(k,n) - n\theta \mid < 1$ for every $k, n \in \mathbb{N}$. On the other hand for the Bernoulli sequence $s(k,n)$ is a random variable that is binomially distributed with n trials and success probability θ. Hence the expectation of $s(k,n)$ equals $n\theta$, but it has a variance equal to $n\theta(1-\theta)$. Moreover, the deviation $\mid s(k,n) - n\theta \mid$ is no longer bounded as n goes to infinity. Because of this variance in $s(k,n)$ one could expect that a Bernoulli policy for θ has worse performance than a regular policy with corresponding density θ. Numerical results shown in the following example confirm this for the policies we apply in this paper. The following example shows that this effect on the variance is present in the call center.

Example 3. *We apply the flow rate policy as introduced in Section 3.3. A policy with parameter θ is implemented in two ways:*

(1) Bernoulli sequence: at each decision event a new random number u is generated uniformly distributed on $[0,1]$. If $u < \theta$, a type 2 job will be scheduled.

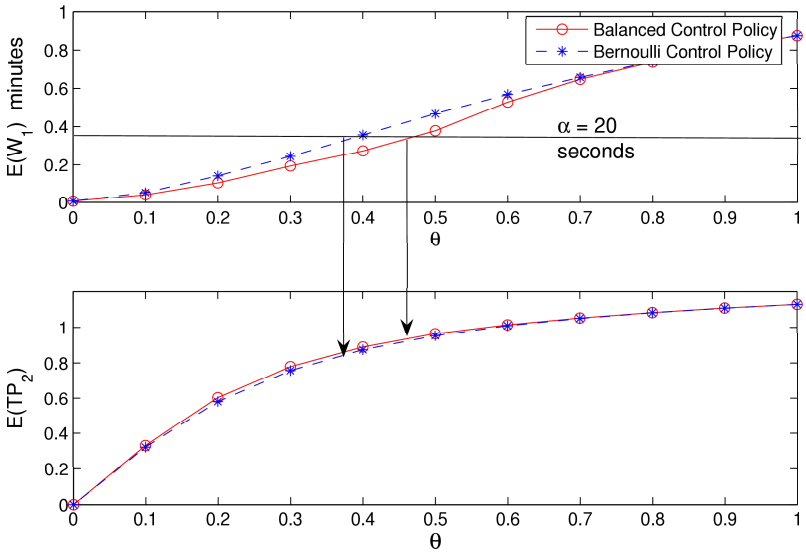
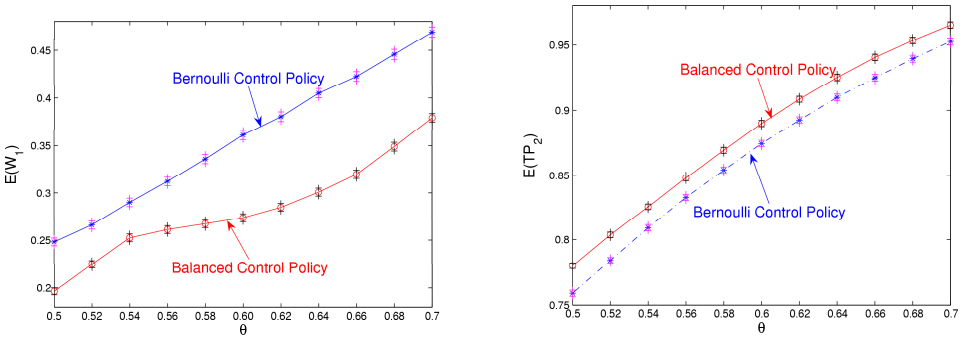


Figure 5: The performance comparison between the balanced control policy and the Bernoulli control policy

(2) *Balanced sequence: at the n-th decision event the next symbol u from the regular sequence of density θ is picked. If $u = 1$, a type 2 job will be scheduled.*

We have simulated the call center using both methods for generating a control policy. The average waiting time of type 1 jobs and the throughput of type 2 jobs are plotted in Figure 5. As can be seen from Figure 5, using balanced control leads to better performance than using a simple Bernoulli random control. The average waiting time of a type 1 job by using balanced control is much smaller than the one by using the Bernoulli policy in the area of $\theta \in [0.3, 0.7]$. Especially for values of θ around 0.5 the differences in performance between the Bernoulli policy and the balanced policy are quite significant, but the difference becomes much smaller if θ is small (close to 0) or large (close to 1). This can be explained by the fact that in that case the variance in $s(k, n)$ for the Bernoulli policy, $n\theta(1-\theta)$, is relatively small.

To illustrate this effect in more detail, Figure 6(a) shows the average waiting time of type 1 customers, denoted by $\mathbb{E}[W_1]$, and its 95 percent confidence



(a) The average waiting time

(b) The throughput

Figure 6: The performance comparison with confidence interval between the balanced control policy and the Bernoulli control policy

interval, and Figure 6(b) shows the average throughput of type 2 customers, denoted by $\mathbb{E}[TP_2]$*, and its* 95 *percent confidence interval. As shown in Figure 6, the balanced control policy yields significantly shorter average waiting times and a higher average throughput than the Bernoulli control policy.*

Furthermore, Figure 5 can be also interpreted as follows. To a given mean waiting time α *(the guaranteed maximum average waiting time of type 1 jobs), one can read the optimal value of* θ*. Next, one can read the throughput of type 2 jobs associated with* α *under the policy using the flow rate of* θ*. Suppose the service level of type 1 jobs is* 20 *seconds. As we can see from Figure 5, for the balanced control policy, by setting* θ *around* 0.46*, we can obtain the maximum throughput of type 2 jobs around* 0.95 *while satisfying the waiting constraint on type 1 jobs; whereas for the Bernoulli control policy, by setting* θ *around* 0.37*, the obtained maximum throughput of type 2 jobs is around* 0.8*. Thus it is clear that the throughput of type 2 jobs by using the balanced control policy is higher than the one by using the Bernoulli policy, when they both reach the constraint on the waiting time of type 1 jobs (i.e.,* $\alpha = 20$ *seconds).*

As shown in the above example, the balanced control policy yields better performance than the Bernoulli policy, that is, lower expected waiting times and a higher throughput for a given θ. Most importantly, the optimal balanced control yields better performance than the balanced control version of the optimal Bernoulli control.

5.3 Optimizing the Balanced Control Policy

Recall that, by Lemma 2, it holds that $\mathbb{E}_\Phi[R(U_{\theta,\Phi}(j))] = \theta P + (1-\theta)Q$, and therefore

$$\frac{d}{d\theta}\mathbb{E}_\Phi[R(U_{\theta,\Phi}(j))] = \frac{d}{d\theta}\Big(\theta P + (1-\theta)Q\Big) = P - Q.$$

In words, the derivative of the marginal Markov kernel of the Bernoulli sequence and that of the balanced sequence are equal. This makes it possible to apply the phantom estimator presented in Section 4 to the balanced sequence. Indeed, for the Bernoulli sequence as for the balanced sequence the input of the gradient estimator is identical: a sequence of $0, 1$ decisions, and the derivative is given by forcing a particular decision to either equal to 1 or to 0; for details on the adaptation of the phantom estimator to balanced sequences we refer to Appendix 8. Unfortunately, the resulting estimator is, in general, biased. However, as we will illustrate by numerical experiments the gradient estimator yields the correct results for the call center problem. A deeper investigation on the unbiasedness of the phantom estimator experienced in our numerical experiments will be a topic of further research.

6 Optimal Balanced Control: Numerical Examples

6.1 Derivative Estimation

In this section, we consider a call center system with trunk-θ reservation policy. The system parameters are $\alpha = 0.167, \lambda = 1/2, \mu_1 = 4/10, \mu_2 = 3/10, C = 5$, T $= 840$ (measured in minutes representing 14h of operating time of the call center) and $K_1 = 1, K_2 = 2$. In order to reduce variance in the simulation, we use three independent random number streams for generating interarrival times and service times of type 1 and type 2 jobs.

We first perform an intensive simulation of $\mathbb{E}_\theta[W_1]$ for various values of θ and fit a polynomial to the observed values. The derivative of the polynomial serves as numerical approximation for the unknown true derivative.

We first address the question which phantom coupling scheme introduced in Section 4 is preferable. Specifically, we compare

- the naive phantom estimator, i.e., without coupling;

- the DTC phantom estimator; and
- the CTC phantom estimator.

A comparison of the three variants of the phantom estimator is shown in Figure 7. The phantom estimator without coupling shows considerable numerical inaccuracy, while the other estimators yield good results compared to the numerical approximation.

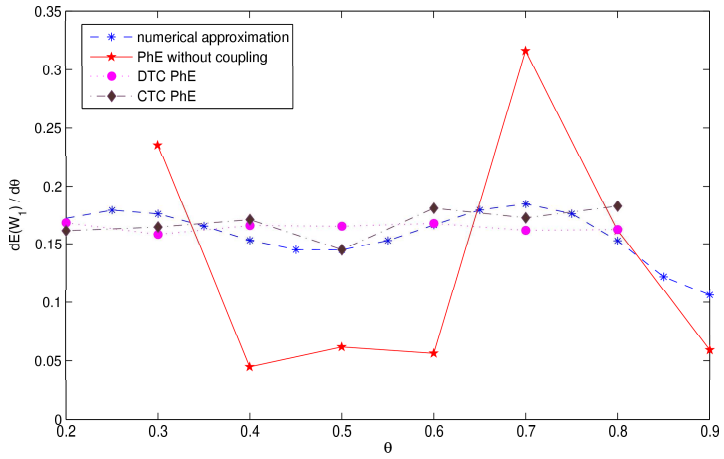


Figure 7: The derivative estimation comparison between different implementations of the phantom estimator

The direct implementation of the naive phantom estimator is not only computationally inefficient it also tends to have high variance. The DTC estimator avoids computing the period of the regular sequence corresponding to a rational θ, however it has more simulation burden than the CTC phantom estimator.

To compare the efficiency of the estimators, we use the concept of *work-normalized variance*, which balances the computational effort and variance of the estimator, and is given by the product of the variance and the expected work per run balancing computational effort and estimator variance, see Glynn and Whitt (1992). We compare the work-normalized variance of three variants of the phantom estimator in Table 1. The naive estimator without coupling has poor performance. Its work-normalized variance is 53371 times

Table 1: The work-normalized variance of three implementations of phantom estimator

	the work-normalized variance				
θ	0.4	0.5	0.6	0.7	0.8
Without coupling	3.5654	3.3037	2.6927	13.4690	21.4063
DTC	0.0070	0.0055	0.0111	0.0181	0.0257
CTC	$0.0977e^{-003}$	$0.0330e^{-003}$	$0.1000e^{-003}$	$0.3094e^{-003}$	$0.1231e^{-003}$

as high as that of the CTC estimator. Furthermore, the work-normalized variance of the DTC method is on average 93 times higher than the one of the CTC method.

Based on the above experiment results, the CTC estimator turns out to have the best performance. We now compare the performance of this phantom estimator with the gradient-free method of Finite Difference (FD). FD estimates the derivative through

$$\frac{dC(\theta)}{d\theta} = \frac{C(\theta+\Delta) - C(\theta)}{\Delta}.$$

The quality of the FD estimate heavily depends on the value of Δ, and for our experiments, we set $\Delta = 0.01$. Figure 8 shows the WNV of the CTC phantom estimator and the FD estimate. On average the WNV of the FD method is equal to 5 times the one of the CTC phantom estimator. Note that apart from a significantly smaller WNV the CTC estimator does not require the choice of Δ which is of key importance for the FD method.

6.2 Optimization

In this section we illustrate the application of the phantom estimator for finding the optimal rate for the trunk-θ reservation policy, where we use the stochastic approximation algorithm (9) for optimization.

For our experiment we set the waiting time constraint to $\alpha = 10s$. Consequently we conduct the randomized policy with $K_1 = 1, K_2 = 2$. The initial guess is $\theta_0 = 0.3$. The gain sequences are: $a_k = a/(k+1)$. The value of parameters are chosen as $a = 0.7$. The iteration algorithm is terminated if

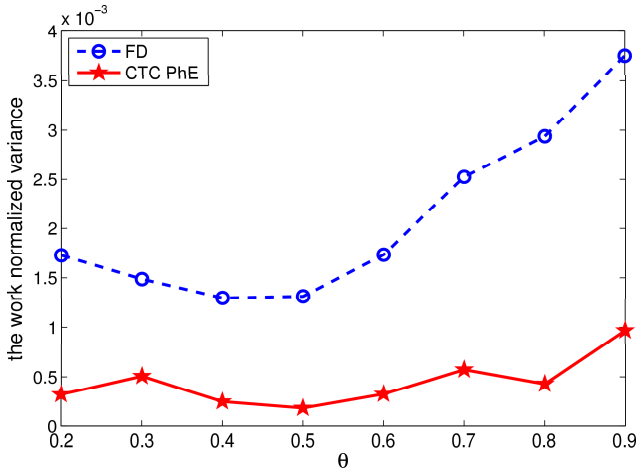


Figure 8: The performance comparison of CTC phantom estimator and the FD method

$|C(\theta_{k+1}) - C(\theta_k)| < 0.01$ in three successive iterates. Since the numerical values for $\mathbb{E}[(W_1 - \tau)^2]$ are rather small, we scale the performance function by a factor $A = 100$ and the resulting problem is to minimize $\mathbb{E}[100(W_1 - \tau)^2]$.

For the CTC phantom estimator, the simulation time per iteration depends on the coupling rate of two phantoms. In order to fairly compare the performance between the phantom estimator and the FD method, we give both methods for each iteration of the optimization algorithm around 30 seconds of CPU time. The optimization trace using the CTC phantom estimator and the one using FD are shown in Figure 9. As can be seen in this figure, the phantom estimator has a better performance than FD. Specifically, the FD method requires 11 iterations to find the optimal setting of $\theta = 0.576424$ and the optimal performance measure is 0.130642; whereas the phantom estimator finds the optimal setting after 5 iterations, with optimal setting of $\theta = 0.535893$ the performance measure is 0.110930 which is better than that obtained by FD.

6.3 Which Policy to Choose?

So far we have dealt with the problem of finding the optimal parameter for a balanced mixing policy. We conclude this section with a comparison between the two policies under discussion. For illustration purposes, we

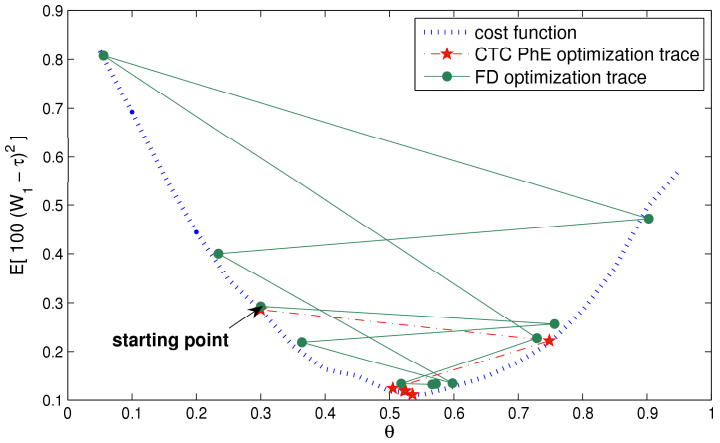
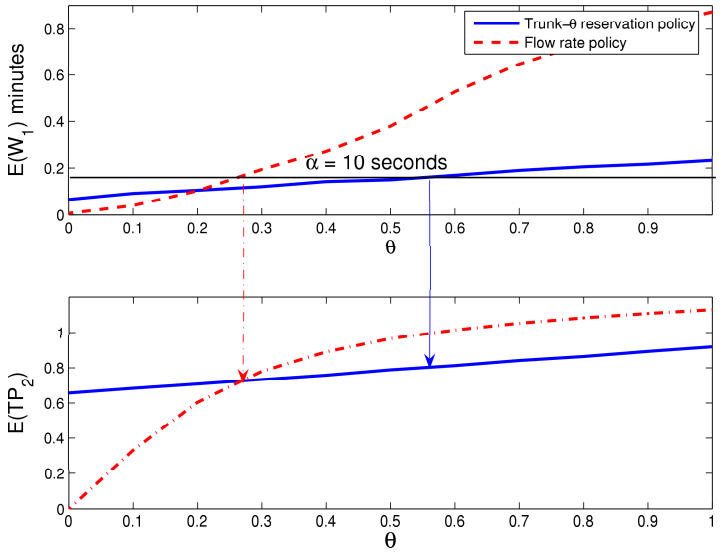


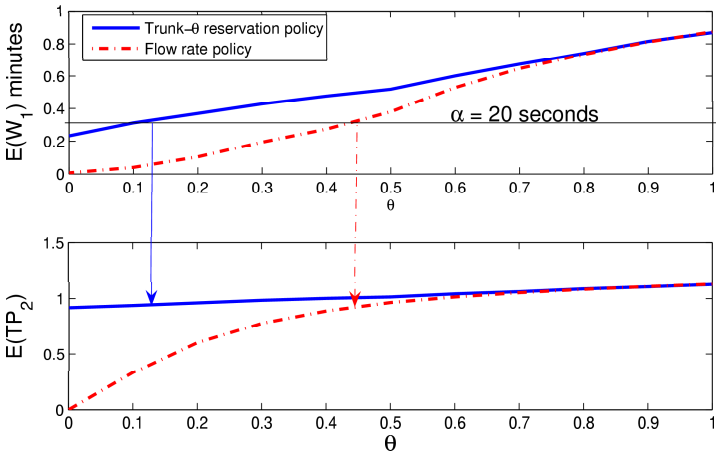
Figure 9: The optimization trace FD and phantom estimator of $K_1 = 1, K_2 = 2$, and $\alpha = 0.167$

compare the performance of these two policies as in Figure 10. As we have pointed out before, in order to solve the constrained optimization problem, the figure can be interpreted as follows: to a given guaranteed waiting time of type 1 jobs α, one can read the optimal value of θ. Next, one reads the throughput of type 2 jobs associated with α for this value of θ. Suppose that the service level of type 1 jobs is 10 seconds and choose a randomization between $K_1 = 1, K_2 = 2$. As shown in Figure 10(a), the optimal throughput of the flow rate policy is smaller than the one using the trunk reservation policy. Figure 10(b) shows the performance for the waiting time constraint $\alpha = 20$ seconds, and $K_1 = 0, K_2 = 1$. In this setting, both policies reach a similar throughput for type 2 jobs.

The numerical examples put forward in this paper indicate that the flow rate policy proposed in this paper yields a performance that is comparable to the well-known trunk reservation policy, which has been proven to be optimal in the case of equal service requirement in Bhulai and Koole (2003). It is worth noting that the trunk reservation policy requires to identify the values K_1 and K_2, see Section 3.2. This makes the flow rate policy easier to implement in practice.



(a) $K_1 = 1, K_2 = 2$ and $\alpha = 10s$



(b) $K_1 = 0, K_2 = 1$ and $\alpha = 20s$

Figure 10: The performance of flow rate policy compared with the one of trunk-θ reservation policy

7 Conclusion

We presented a new approach to the control of call centers via randomized policies. Our approach relies on balanced sequences as implementation of randomized policies and we have presented a gradient estimator that allows to find the optimal balanced sequence for the control problem discussed in this paper. This is a first result on the application of balanced sequences in optimal control for complex systems. Topics of further research are to find sufficient conditions for unbiasedness of our estimator. Another line of further research will be studying control problems where the constraint is given by a quantile.

References

Altman, E. (1999). *Constrained Markov Decision Processes.* London: Chapman and Hall.

Altman, E., B. Gaujal, and A. Hordijk (2000a). Balanced sequences and optimal routing. *Journal of the ACM 47*, 4.

Altman, E., B. Gaujal, and A. Hordijk (2000b). Multimodularity, convexity and optimization properties. *Mathematics of Operations Research 25*(2), 324–347.

Altman, E., B. Gaujal, and A. Hordijk (2002). Regular ordering and applications in control policies. *Discrete Event Dynamic Systems 12*(2), 187 – 210.

Altman, E., B. Gaujal, and A. Hordijk (2003). *Discrete-Event Control of Stochastic Networks: Multimodularity and Regularity.* ASecaucus, NJ, USA: Springer-Verlag New York, Inc.

Altman, E., B. Gaujal, A. Hordijk, and G. Koole (1998). Optimal admission, routing and service assignment control: the case of single buffer queues. In *37th IEEE Conference on Decision and Control*, Volume 2, Tampa, FL, USA, pp. 2119–2124.

Altman, E. and A. Shwartz (1993). Time-sharing policies for controlled Markov chains. *Operations Research 41*(6), 1116–1124.

Bhulai, S. and G. Koole (2003). A queueing model for call blending in call centers. *IEEE Transactions on Automatic Control 48*, 1434 –1438.

Cao, X. (2007). *Stochastic Learning and Optimization: A Sensitivity-Based Approach.* New York: Springer.

Franx, G., G. Koole, and S. Pot (2006). Approximating multi-skill blocking sys-

tems by hyperexponential decomposition. *Performance Evaluation 63*(8), 799–824.

Gans, N., G. Koole, and A. Mandelbaum (2003). Telephone call centers: tutorial, review, and research prospects. *Manufacturing and Service Operations Management 5*, 79–141.

Gans, N. and Y. Zhou (2003). A call-routing problem with service-level constraints. *Operations Research 51*, 255–271.

Glynn, P. W. and W. Whitt (1992). The asymptotic efficiency of simulation estimators. *Operations Research 40*(3), 505–520.

Gurvich, I., M. Armony, and A. Mandelbaum (2008). Service level differentation in call centers with fully flexible servers. *Management Science 54*(2), 279–294.

Hajek, B. (1985). Extremal splittings of point processes. *Mathematics of Operations Research 10*(4).

Hardy, G. H. and E. M. Wright (1960). *An Introduction to the Theory of Numbers* (Fourth ed.). Oxford: The Clarendon Press.

Heidergott, B. and X. R. Cao (2002). A note on the relation between weak derivatives and perturbation realization. *IEEE Transactions on Automatic Control 47*(7), 1112–1115.

Heidergott, B. and A. Hordijk (2003). Taylor series expansions for stationary Markov chains. *Advances in Applied Probability 23*, 1046–1070.

Heidergott, B., A. Hordijk, and H. Weisshaupt (2006). Measure-valued differentiation for stationary Markov chains. *Mathematics of Operations Research 31*(1), 154–172.

Heidergott, B. and F. Vázquez-Abad (2006). Measure-valued differentiation for random horizon problems. *Markov Processes and Related Fields 12*, 509–536.

Heidergott, B. and F. Vázquez-Abad (2008). Measure-valued differentiation for Markov chains. *Journal of Optimization and Applications 136*(2), 187–209.

Koole, G. and A. Mandelbaum (2002). Queueing models of call centers: an introduction. *Annals of Operations Research 113*, 41–59.

Koole, G. and J. Talim (2000). Exponential approximation of multi-skill call centers architecture. In *QNETs 2000: Fourth International Workshop on Queueing Networks with Finite Capacity*, Craiglands Hotel, Ilkley, West Yorkshire, UK, pp. 23/1–10.

Lothaire, M. (2002). *Algebraic Combinatorics on Words.* Cambridge University Press.

Morse, M. and G. Hedlund (1940). Symbolic dynamics II: Sturmian trajectories. *American Journal of Mathematics 62*, 1–42.

Perry, M. and A. Nilsson (1992). Performance modeling of automatic call distributors: Assignable grade of service staffing. In *14th International Switching Symposium*, Yokohama, pp. 294–298.

Pflug, G. (1996). *Optimization of Stochastic Models.* Boston: Kluwer Academic Publishers.

Shinya, S., M. Naoto, and K. Ryohei (2004). m-Balanced words: A generalization of balanced words. *Theoretical computer science 314*(1), 97–120.

Shumsky, R. (2004). Approximation and analysis of a queueing system with flexible and specialized servers. *OR Spektrum 26*, 307–330.

Stanford, D. and W. Grassmann (2000). Bilingual server call centers. In D. McDonald and S. Turner (Eds.), *Analysis of Communication Networks: Call Centers, Traffic and Performance*, Volume 208, pp. 31–47. Fields Institute Communications.

Wallace, R. and W. Whitt (2005). A staffing algorithm for call centers with skill-based routing. *Manufacturing and Service Operations Management 7*(4), 276–294.

8 Appendix A: Implementation of the Phantom Estimator

In the following we discuss the implementation of the phantom estimator. Denote the waiting time difference between two phantoms by $\Delta_w(s)$:

$$\begin{aligned}&\Delta_w(s,j) = W_1^+(s,j) - W_1^-(s,j)\\ =&\frac{1}{\lambda T}\left(\sum_{k=1}^{M^+(s,T,\theta)-j+1} g(X_\theta^+(s,k)) - \sum_{k=1}^{M^-(s,T,\theta)-j+1} g(X_\theta^-(s,k))\right). \qquad (16)\end{aligned}$$

Note that there are only two different choices for the Markov kernel (P or Q) at the splitting point. Therefore, one of the two phantoms is equal to the nominal path. Specifically, if the decision variable (either zero or one) for the j-th choice is $u(j) = 1$, then the "plus" phantom is equal to the nominal path, which yields $W_1^+(s,j) = W_1(\theta)$. In this case, using the difference variable $\Delta_w(s)$, then average waiting time of the "minus" phantom is obtained by

$$W_1^-(s,j) = W_1^+(s,j) - \Delta_w(s,j) = W_1(\theta) - \Delta_w(s,j). \qquad (17)$$

On the other hand, if $u(j) = 0$, then average waiting time of the "minus" phantom is equal to the nominal path, which yields $W_1^-(X_\theta(s,j) = W_1(\theta)$ and

$$W_1^+(s,j) = W_1^-(s,j) + \Delta_w(s,j) = W_1(\theta) + \Delta_w(s,j). \tag{18}$$

Let $\mathcal{I}(j)$ be the indicator whether the transition kernel of the nominal path is equal to P or Q:

$$\mathcal{I}(j) = \begin{cases} 1, & \text{if } u(j) = 1, \\ -1, & \text{if } u(j) = 0. \end{cases}$$

The nominal path coincides with either of the phantoms and the average waiting time of the phantom that has to be additionally simulated is obtained from combining (17) with (18):

$$W_1(\theta) - \mathcal{I}(j)\Delta_w(s,j).$$

Following this train of thoughts, the difference between the cumulative costs in Equation (11) can be written as

$$D(s,j) = A\mathcal{I}(j)\Big((W_1(\theta) - \alpha)^2 - (W_1(\theta) - \mathcal{I}(j) \cdot \Delta_w(s,j) - \alpha)^2\Big) \tag{19}$$

and the resulting gradient estimator is

$$\frac{d}{d\theta}\mathbb{E}_\theta[A(W_1(\theta) - \alpha)^2] =$$

$$\mathbb{E}\left[\sum_{j=1}^{M(T,\theta)} A\mathcal{I}(j)\left((W_1(\theta) - \alpha)^2 - (W_1(\theta) - \mathcal{I}(j) \cdot \Delta_w(s,j) - \alpha)^2\right)\right]. \tag{20}$$

In the following we discuss implementation issues for the phantom estimator when the balanced control policy is used. Let θ be a rational number. In this case θ can be written as a fraction of two division-free integers, that is, $\theta = p/q$, for $p, q \in \mathbb{N}$. In the balanced sequence for θ, there will be p 1s in any subsequence of length q of $u_{\theta,\phi}(n)$, and $\{u_{\theta,\phi}(k)\}$ will be periodic with the period length q. For example, the period of the balanced sequence generated for $\theta = 0.3 = 3/10$ and $\phi = 0$ is: $(0,0,0,1,0,0,1,0,0,1)$ with period length $q = 10$. Let $n\%q$ denote the position in the period of the nth element of $\{u_{\theta,\phi}(k)\}$. It is easy to see that if $m\%q = n\%q$, then for $k \geq 1$, it holds that

$$u_{\theta,\phi}(m+k) = u_{\theta,\phi}(n+k).$$

We extend the state-space of the phantom process $X_\theta^\pm(n)$ with the auxiliary variable $u_{\theta,\phi}^\pm(n)$ denoting at which position in the period of the θ-balanced sequence the process is. For irrational θ, using the fact that, on a computer, θ is given in finite decimal notation, we may obtain a rational number equal or arbitrarily close to θ by applying the continued fraction algorithm on θ (see, for example, Chapter 10 in Hardy and Wright (1960)). Then in the coupling schemes we extend the physical state by $u_{\theta,\phi}^\pm(n)$. In words, the "plus" and "minus" version couple as soon as the following variables are identical: the number of jobs waiting, the number of type 1 jobs in service, the number of type 2 jobs in service, and the balanced sequences driving the policy have the same position in the period.

9 Appendix B: Balanced Sequences

For an infinite sequence $U = (u_1, u_2, \ldots)$ of zeros and ones,

$$s(k,n) = \sum_{j=k}^{k+n-1} u_j, \quad k \leq n,$$

denotes the numbers of ones in the subsequence of length n beginning at the k-th element of U. With this notation $s(n) := s(1,n)$ is the number of ones in the prefix of length n of U. We say that $U = (u_1, u_2, \ldots)$ has *density* $\theta \in [0,1]$ if $\lim_{n\to\infty} s(n)/n$ exists and is equal to θ. Consider, for example, the case $\theta = 1/2$. Then the sequence $(0,1,0,1,0,1,\ldots)$ has rate $1/2$ and could be used for modeling a mixing policy at $\theta = 1/2$. However, $(0,1,0,0,1,1,0,0,0,1,1,1,\ldots)$ has also rate $1/2$. The latter sequence is probably not a good choice as the sequences of consecutive ones and zeros lead to a rater unbalanced system behavior.

In the following we introduce the concept of uniformly recurrent sequences. An infinite sequence $U = (u_1, u_2, \ldots)$ is *uniformly recurrent* if for every finite subsequence W of U, an integer k exists such that W is a subsequence of every subsequence of U of length k. Note that any periodic sequence is uniformly recurrent. For example, the sequence $(0,1,0,1,\ldots)$ is uniformly recurrent as any subsequence of length l is contained in any other finite subsequence of length $l+1$.

Note that if U has some density θ, it is the asymptotic frequency of the ones in U. If such U is applied as control sequence then policy P is

applied with fraction θ of all decision events while Q is applied with remaining fraction $1-\theta$. Moreover, it follows for $k = 0, 1, \ldots$ that

$$\lim_{n\to\infty} \frac{s(k,n) - n\theta}{n} = 0.$$

Intuitively a small maximal absolute deviation between $s(k,n)$ and $n\theta$ means that the ones and zeros are regularly distributed with density θ. Thus U should be such that the maximal absolute deviation is small. It can be shown that for every density $\theta \in [0,1]$ there exists U for which the maximal deviation of $|s(k,n) - n\theta|$ is smaller than one. Such sequences are called regular. Following, for example, (Altman et al. (2000a)), a sequence of zeros and ones is called *balanced* if the difference in number of ones (or zeros which is equivalent) for subsequences of the same length is at most one.

Obviously any regular sequences is a balanced sequence. Also it can be proved (see, for example, Lothaire (2002) Chapter 1 and Chapter 2) that regular sequences are uniformly recurrent.

Example 4. *An example of a regular sequences with density* $\theta = 2/5$ *is the periodic sequence* $(1,0,1,0,0,1,0,1,0,0,1,0,1,0,0,\ldots)$ *. If* θ *is rational then corresponding regular sequences are always periodic with period equal to the denominator of* θ*. Thus 5 is the period of the sequence in this example. Another example of a regular sequence is the well-known (see Lothaire (2002)) Fibonacci sequence* $(1,0,1,1,0,1,0,1,1,0,1,1,0,\ldots)$ *with density* $\theta = (\sqrt{5}-1)/2$ *which can not be periodic since its density* θ *is irrational. However, also the Fibonacci sequence is uniformly recurrent as any regular sequence is.*

By the classification of balanced sequences (see Morse and Hedlund (1940)) it follows that any sequence which is both uniformly recurrent and balanced is in fact a regular sequence. Thus within the subset of uniformly recurrent control sequences to which we restricted earlier regular is equivalent to balanced. Therefore in this paper we use the term "balanced control policy" in the following sense:

A balanced control policy of rate $\theta \in [0,1]$ is a deterministic policy mixing P and Q according to an infinite control sequence $U = (u_1, u_2, \ldots)$ of zeros and ones where U is regular of density θ. This implies that U is both uniformly recurrent and balanced. Moreover, on average over all decision events P is applied with fraction θ and Q with fraction $1-\theta$.

Another useful fact is that regular sequences of given density θ have the same finite subsequences (see Lothaire (2002)) which implies that regularity of a given density θ uniquely determines the sequence (and thus also the policy) modulo a shift ϕ. Therefore balanced control policies of the same density θ have the same Césaro average performance and thus we may speak of **the** balanced policy of rate θ instead of a balanced policy of rate θ.

In this paper we compare while varying over $\theta \in [0,1]$ the performance of the balanced control policy of rate θ with the Bernoulli policy of rate θ while varying over $\theta \in [0,1]$. For simulation a practical implementation of the balanced policy of rate θ is necessary, several ways to construct regular sequences are known. In this paper we use the one given by (14).

10 Appendix C: Proof of Lemma 2

Lemma 2: *Let $\{R(U_{\theta,\Phi}(j))\}_{j=1}^{\infty}$ be a randomized balanced (P,Q)-policy of rate θ. Then for $j = 1, 2, \ldots$ we have that*

$$\mathbb{E}_{\Phi}[R(U_{\theta,\Phi}(j))] = \theta P + (1-\theta)Q.$$

Proof. From $0 \leq \theta \leq 1$ it is easily seen that $U_{\theta,\Phi}(j) \in= \{0,1\}$ for $j = 1, 2, \ldots$. Thus the random variable $U_{\theta,\Phi}(j)$ has a Bernoulli distribution. Also for any $x \in \mathbb{R}$ we trivially have that

$$\int_0^1 \lfloor x + \phi \rfloor d\phi = x. \tag{21}$$

By (21) it follows that

$$\begin{aligned}
\mathbb{E}_{\Phi}[U_{\theta,\Phi}(j)] &= \int_0^1 (\lfloor j\theta + \phi \rfloor - \lfloor (j-1)\theta + \phi \rfloor) d\phi \\
&= \int_0^1 (\lfloor j\theta + \phi \rfloor) d\phi - \int_0^1 (\lfloor (j-1)\theta + \phi \rfloor) d\phi \\
&= j\theta - (j-1)\theta = \theta.
\end{aligned}$$

Hence

$$U_{\theta,\Phi}(j) = \begin{cases} 1, & \text{with probability } \theta, \\ 0, & \text{with probability } 1-\theta. \end{cases}$$

Thus

$$\mathbb{E}_\Phi[R(U_{\theta,\Phi}(j))] = \mathbb{E}_\Phi[U_{\theta,\Phi}(j)]P + (1 - \mathbb{E}_\Phi[U_{\theta,\Phi}(j)])Q = \theta P + (1-\theta)Q.$$

□