A Numerical Method for Solving Singular Brownian Control Problems

Sunil Kumar
Graduate School of Business
Stanford University

Muthukumar Muthuraman
Scientific Computing & Computational Mathematics
Department of Computer Science
Stanford University

Abstract

The Brownian approximation approach to developing dynamic control policies for multiclass queueing networks is useful when the limiting, usually singular, Brownian control problem can be solved. However, this problem can be rarely solved analytically. In this paper we present a method for numerically solving singular Brownian control problems. We adapt finite element methods to iteratively solve the Hamilton-Jacobi-Bellman equation associated with the Brownian control problem. A key feature of our method is that the presence of singular controls simplifies the procedure. The solution to the Hamilton-Jacobi-Bellman equation is then used to construct an optimal control for the Brownian system. We illustrate the method on two examples of singular Brownian control problems.

1 Introduction

Multiclass queueing networks form natural models of a wide variety of systems in manufacturing and communication. The problem of finding optimal scheduling policies for such networks is, therefore, of considerable interest. In general, analytically deriving such optimal policies is intractable. It is prudent to relax one's expectation and to find "good" policies that are optimal in some ideal asymptotic regime. A good candidate for such a regime is "heavy traffic" where the system is barely capable of handling the load impressed upon it. A systematic procedure for finding such good policies was proposed by Harrison [3] is based on approximating the control problem in heavy traffic by a limiting Brownian control problem. A shortcoming of the approach proposed by Harrison is that it is most applicable when the limiting control problem, which is usually singular, is easily solved and the solution to the limiting control problem is easily interpreted. Unfortunately, it is only for a small class of networks that analytical solutions of the limiting Brownian control problem can be derived. A numerical procedure for solving Brownian control problems, based on approximating Markov chains, was proposed by Kushner and Dupuis [8]. In this paper, we propose an alternate method for solving a large class of singular Brownian control problems, which has the following advantages. First, the procedure exploits the singular nature of the control problem to simplify the procedure. In particular, it reduces the key computational step to solving a linear, elliptic Neumann problem, which can be done efficiently using finite element methods. Second, it provides the numerical solution in a form that contains all the information necessary to interpret the solution as a policy in the original multiclass queueing network. To be more specific, it provides the solution in a form that allows one to apply the mechanistic policy design procedure called BIGSTEP proposed by Harrison [4]. Thus, the combination of the numerical method for solution of the Brownian control problem and the BIGSTEP procedure provides one way to design a purely mechanistic procedure for control of multiclass queueing networks.

The proposed method for solving the Brownian control problem may be described as follows. We begin by articulating the control problem that needs to be solved in its lowest dimensional form. This involves reformulating the problem in terms of an equivalent workload formulation [6]. Then, we formally derive the Hamilton-Jacobi-Bellman equation associated with this problem. This partial differential equation has an embedded optimization problem within it. We solve this equation using a two-step iterative procedure. In the first step, we assume a solution to the embedded optimization problem and solve a partial differential equation with known parameters using the finite element method. In the second step, the solution obtained to the partial differential equation is then used to re-solve the optimization problem. This two-step procedure is then iterated until convergence. In moving from step to step, the algorithm uses *the principle of smoothness of fit* [1] as its guiding principle. Solving singular Brownian control problems using the Hamilton-Jacobi-Bellman equation is not new, see for example [1]. Our contribution is to provide a method that is not problem-specific. The procedure raises two important theoretical issues. First, we need to show that the iterative procedure actually converges, and that the limit is indeed a solution to the Hamilton-Jacobi-Bellman equation. Second, we need to verify that a solution to the Hamilton-Jacobi-Bellman equation is indeed optimal for the Brownian control problem. This conference paper in meant to disseminate the basic method and to illustrate its application. Hence, neither of these theoretical issues will be considered in this paper.

The rest of the paper is organized as follows. In the next section, application of the method outlined above will be illustrated on an example of a storage system with singular controls. In the following section, the method will be illustrated on the now-famous criss-cross network considered by Harrison and Wein [7], which was further analyzed by Martins, Shreve, and Soner [9].

2 Illustrating the method: A Storage System

Rather than articulating the general method, we begin by illustrating the method on a singular Brownian control problem that is a finite state space variant of the system considered by Harrison and Taksar [5]. To be completely honest, the articulation of the general method is non-trivial. Consider the following model of a storage system with finite storage capacity of size 1, whose content is denoted $W(t)$, $t \geq 0$[1]. Let $X(t)$, $t \geq 0$ be a (μ, σ^2) Brownian motion starting at $w \in [0, 1]$. A control policy R is specified by a non-decreasing process right continuous process with left limits (RCLL) $R(t)$, $t \geq 0$ that is adapted to $X(.)$. The content process under the policy R is given by

$$W(t) = X(t) + R(t) - L(t) \; t \geq 0, \tag{1}$$

where $L(.)$ is the (unique) minimal, non-decreasing, RCLL process that keeps $W(t) \leq 1$ for all $t \geq 0$, the so-called one sided-regulator [2] applied to the process $X + R$. We interpret $R(t)$ as the cumulative amount of content increase effected by the controller (by idling the server in a queueing context) up to time t. We are interested in choosing the policy R so as to (a) keep the contents of the system, $W(t)$ in the interval $[0, 1]$ for all $t \geq 0$ and (b) to minimize the cost function

$$v(w) = \mathbf{E}_w \left[\int_0^\infty e^{-\gamma t} h(W(t))\, dt \right], \tag{2}$$

where $h(.)$ is the instantaneous cost function and, for concreteness, is given by $h(w) = k(w - \bar{w})^2$, where k, b are positive constants and $\bar{w} \in (0, 1)$. The non-monotonicity of $h(.)$ may be surprising at first, but in light of the equivalent workload formulation, it is quite plausible. We shall see a derivation of such an $h(.)$ in the next section. Since $h(.)$ is not monotone, and since there are no explicit costs of control, it may be in the controller's interest to instantaneously increase the process $W(t)$ by increasing $R(t)$. Thus the optimal control could involve a barrier or free boundary.

We now go about constructing the optimal policy R. Define Γ to be the differential operator

$$\Gamma \equiv \frac{1}{2}\sigma^2 \frac{\partial^2}{\partial w^2} + \mu \frac{\partial}{\partial w},$$

and let

$$v'(w) \equiv \frac{\partial v(w)}{\partial w}.$$

The following result can be proved exactly along the lines of Harrison and Taksar [5].

Proposition 1 *Suppose there exist a twice continuously differentiable function $v(.) \in C^2$ and a constant $a \in [0, 1]$ that satisfy the following equations.*

$$\begin{aligned}
\Gamma v(w) - \gamma v(w) + h(w) &\geq 0, \forall w \in [0, 1] && (3)\\
v'(w) &\geq 0, \forall w \in [0, 1] && (4)\\
\Gamma v(w) - \gamma v(w) + h(w) &= 0, \forall w \geq a && (5)\\
v'(w) &= 0, \forall w \leq a \text{ and} && (6)\\
v'(1) &= 0. && (7)
\end{aligned}$$

Then the optimal control policy R that achieves the minimum in (2) and the process L are given by the following equations.

$$\begin{aligned}
R(0) &= (a - w)^+, && (8)\\
R(t) - R(0) &= \sup_{0 \leq s \leq t} (X(s) - L(s) - a)^- \text{ and} && (9)\\
L(t) &= \sup_{0 \leq s \leq t} (1 - X(s) - L(s))^-. && (10)
\end{aligned}$$

The proposition states that the optimal control policy is a barrier policy that translates all initial contents w that are below the level a to the level a instantaneously, and maintains the contents in the interval $[a, 1]$ from then on using the minimal amount of translation.

Thus, we see that the optimal problem reduces to finding a and v that satisfy (3-7), which can be considered a special version of the Hamilton-Jacobi-Bellman equation. We now turn our attention to a procedure for finding such an a and v analytically. This procedure will form the basis of our numerical method.

Step 1. Initially we assume that $a_1 = 0$. We then solve the resulting Ordinary Differential Equation (ODE), $\Gamma v - \gamma v + h = 0$ with the boundary conditions $v'(0) = v'(1) = 0$. Although the resulting expressions are messy, this linear ODE can be solved analytically. The top set of plots of Figure 1 show the resulting v and its first and second derivatives, v' and v'' for the case when $\sigma^2 = 2$, $\mu = 0$, and $h(w) = 0.1(w - 0.64)^2$.

[1] Although it is common to denote buffer contents by Z, we use W to consistent with the equivalent workload formulation that is described in the sequel.

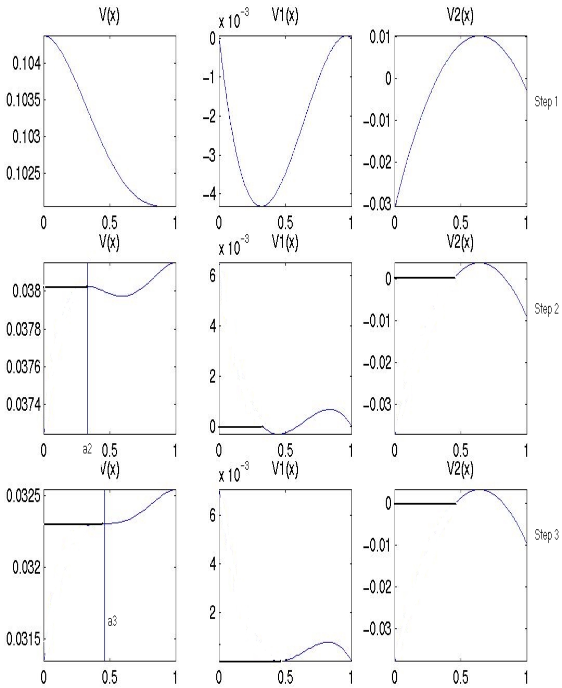

Figure 1: The Storage Example

Step 2. As can be seen from the plots, $v'(w) < 0$ for all $w < 0.9$. Thus, (4) is violated and we need to move the barrier from $a = 0$. In order to resolve the next position of the barrier, we use the *principle of smoothness of fit* [1]. That is, since we expect v to be in C^2, we know that $v'(a) = v''(a) = 0$ at the optimal barrier a. So we choose to update the barrier to that point a_2 where, under the current barrier $v''(a_2) = 0$ and $v'(w) \leq 0$ for all $w \leq a_2$. Loosely speaking, $v'(a) < 0$ indicates that increasing the content decreases the optimal cost, and hence it is beneficial to instantaneously increase the content and hence the barrier should be to the right of a. Thus, we now (analytically) solve $\Gamma v - \gamma v + h = 0$ in $[a_2, 1]$, with $v'(a_2) = v'(1) = 0$ and $v(w) = v(a_2)$ for all $w \leq a_2$. The second row of plots in Figure 1 show the resulting v, v', and v''.

Step 3. As can be seen from the second row of plots in Figure 1, there is still a region where $v'(w) < 0$. So we repeat the procedure in step 2 again, arriving at the plots show in the third row of plots in Figure 1 and a new barrier a_3. In this case, we note that $v'(w) \geq 0$ for all $w \in [0, 1]$ and that $v'(a_3) = v''(a_3) = 0$ upto a non-zero tolerance (which is essential for the termination of the algorithm in a finite number of steps). It is fairly straight forward to verify that the resulting v satisfies (3-7). Thus we are done and the method stops.

Some comments about the method described above are in order. First, the plots of v in Figure 1 show that the procedure is indeed a *policy improvement* procedure, where each iteration results in a barrier policy that is strictly superior to the previous one. Second, the procedure requires that the ODE $\Gamma v - \gamma v + h = 0$ in $[a_2, 1]$, with $v'(a_2) = v'(1) = 0$ be solved. This is the so-called Neumann problem for a linear (elliptic) differential operator. In this simple one dimensional case, we are able to solve this problem analytically. In the general method, this problem will have to be solved numerically. Yet, this is a fairly easy problem for the finite element method and this is one of the most desirable features of the method: that it reduces a singular control problem to a computationally tractable problem by exploiting the nature of the singular control and the resulting barrier or free boundary problem. Third, even if the ODE is solved analytically, we are not guaranteed convergence in a finite number of steps. But as can be seen from the figure, most of the possible improvement is obtained in the first few steps. Finally, the algorithm does exploit some structure of the optimal control problem that is particular to the problem at hand. For example, each iteration assumes that $v'(w_1) < 0$, then $v'(w) < 0$ for all $w < w_1$. This is to be expected in a problem with a single lower barrier, but proving and generalizing such properties in higher dimensions is non-trivial. Yet, in any application, if the algorithm finds a solution to a less specific analog of (3-7), we are done.

3 The Criss-Cross Network

We now illustrate the method on a two-dimensional example where the Neumann problem is not solved analytically. Our treatment in this section will not be completely rigorous in the interest of brevity. We consider a finite variant of the so-called Criss-Cross Network [9]. This network has become an *E. coli* of sorts of singular network control problems. The network is show in Figure 2. It consists of two servers and is populated by two types of customers: A and B. Both the types arrive at Server 1 according to independent renewal processes with unit rates and finite interarrival time variance. Type A customers require service only at server 1 while Type B customers require service first from server 1 and then from server 2. Classes 1, 2 and 3 denote customers of Type A, customers of Type B awaiting first service, and customers of Type B awaiting second service, respectively. In order to articulate the limiting control problem, we consider a sequence of systems indexed by r such that in the r-th system, service times in each class i are independent and identically distributed with mean μ_i^r and finite variance.

Service time are independent across classes and of the interarrival times. We assume that the system is in heavy traffic,

$$\lim_{r\to\infty}\left(\frac{1}{\mu_1^r},\frac{1}{\mu_2^r},\frac{1}{\mu_3^r}\right)=(0.5,0.5,1),$$

and that

$$\lim_{r\to\infty} r\left(1-\frac{1}{\mu_1^r}-\frac{1}{\mu_2^r},1-\frac{1}{\mu_3^r}\right)=(\alpha_1,\alpha_2)=:\alpha',$$

for some constants α_1 and α_2. We also require that the service time processes satisfy a functional central limit theorem, but we will omit details in the interest of brevity. The classes have capacity constraints that will be articulated in terms of the workload below.

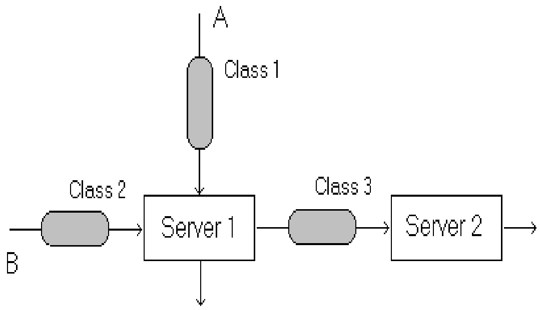

Figure 2: The Criss-Cross Network

The limiting control problem in its workload formulation [7] can be described as follows. Let $d_1=(1,0)'$ and $d_2=(0,1)'$ denote the two Euclidean basis vectors, Let $W(t)=(W_1(t),W_2(t)),t\geq 0$ denote the workload process. We assume that

$$0\leq W_i(t)\leq 1 \text{ for all } t\geq 0 \text{ and } i=1,2.$$

This is our capacity constraint which, although less than natural in the original system, is most convenient in the workload formulation.[2] Let $w=W(0)\in[0,1]\times[0,1]$. The dynamics of the workload process are given by

$$W(t)=w+X(t)+R_1(t)d_1+R_2(t)d_2-L_1(t)d_1-L_2(t)d_2, \tag{11}$$

where $X(.)$ is a (α,Σ) Brownian motion (for some 2×2 covariance matrix Σ), $L_1(.),L_2(.)$ are non-decreasing processes that correspond to 1-sided regulators at the boundaries $W_1=1$ and $W_2=1$ respectively, and $R_1(.),R_2(.)$ denote the non-decreasing RCLL control processes adapated to X that correspond to pushing in the d_1 and d_2 directions respectively. $R_1(.),R_2(.)$ are under the system managers control and the objective is

$$v(w)=\min_{\{R_1,R_2\}}\mathbf{E}_w\left[\int_0^\infty e^{-\gamma t}h(W(t))dt\right], \tag{12}$$

where $v(w)$ is the optimal infinite horizon expected discounted cost function, γ is the interest rate under continuous compounding and $W(.)$ is the workload process. Assuming that jobs in Class 3 incur holding costs at three times the rate for jobs in Classes 1 and 2, $h(.)$, the instantaneous holding cost function, is given by the solution to the linear program [6]

$$h((w_1,w_2))=\min_{\{z_1,z_2,z_3\}}[z_1+z_2+3*z_3] \tag{13}$$

$$\begin{aligned}\text{such that } 0.5z_1+0.5z_2 &= w_1\\ z_2+z_3 &= w_2,\\ z_1,z_2,z_3 &\geq 0.\end{aligned}$$

As the observant reader would have realized, $h(.)$ is not monotone in the sense that, for $w_2>2w_1>2w_1'$, $h((w_1',w_2))>h((w_1,w_2))$. Thus we can expect to see a barrier or free boundary in the optimal prescription, as discussed in the previous section.

The parameter choices correspond to Case 2(b) of Martins, Shreve, and Soner [9]. Motivated by the discussion there, we will try and construct the optimal cost function as follows. For a function $v(.):\mathbf{R}^2\to\mathbf{R}^2$ with $v\in C^2$ define

$$\Gamma v(w)\equiv\frac{1}{2}\sum_{i=1}^{2}\sum_{j=1}^{2}\Sigma_{ij}\frac{\partial^2 v(w)}{\partial w_i\partial w_k}+\sum_{i=1}^{2}\alpha_i\frac{\partial v(w)}{\partial w_i},$$

and let

$$\nabla v(w)\equiv(\frac{\partial v(w)}{\partial w_1},\frac{\partial v(w)}{\partial w_2})'.$$

Further suppose that there exists a continuous function $\phi(.):\mathbf{R}\to\mathbf{R}$ such that

$$\begin{aligned}\Gamma v(w)-\gamma v(w)+h(w) &\geq 0,\forall w\in[0,1] && (14)\\ \nabla v(w) &\geq (0,0)',\forall w\in[0,1] && (15)\\ \Gamma v(w)-\gamma v(w)+h(w) &= 0,\\ &\quad \text{if } w_1\geq\phi(w_2) && (16)\\ \nabla v(w)\cdot d_1 &= 0,\\ &\quad \text{if } w_1\leq\phi(w_2) && (17)\\ \nabla v((1,w_2))\cdot d_1 &= 0 \text{ and} && (18)\\ \nabla v((w_1,1))\cdot d_2 &= 0. && (19)\end{aligned}$$

Then $v(w)$ is the optimal cost function under a singular control that applies the minimum amount of control in the d_2 direction to maintain the process within

[2] Of course, finiteness of the state-space is an unavoidable requirement of many numerical methods, including ours.

$[0,1]\times[0,1]$, and applies a singular control that instantaneously translates any initial state $w=(w_1,w_2)$ with $w_1 \le \phi(w_2)$ to the state $(\phi(w_2),w_2)$ and then applies the minimal amount of control necessary to maintain the workload process in the region $\{\forall(w_1,w_2) : 0 \le w_2 \le 1, \phi(w_2) \le w_1 \le 1\}$.

In order to find the functions ϕ and v, we adapt the procedure described in the previous. The key difference between the two settings is that the sequence of Neumann problems cannot be solved analytically and must be solved numerically. We begin by setting $\phi^0 \equiv 0$. For a given function ϕ^n we solve

$$\begin{aligned}\Gamma v(w) - \gamma v(w) + h(w) &= 0, \\ &\quad \text{if } w_1 \ge \phi^n(w_2) \quad (20)\\ \nabla v((\phi^n(w_2),w_2))\cdot d_1 &= 0, \quad (21)\\ \nabla v((1,w_2))\cdot d_1 &= 0, \text{ and} \quad (22)\\ \nabla v((w_1,1))\cdot d_2 &= 0. \quad (23)\end{aligned}$$

We then use the solution $v(.)$ above update the function ϕ^n to ϕ^{n+1} (and thus update the barrier) once again using the principle of smoothness of fit. To be more specific, for each w_2, we look for a point $w_1^* \equiv \phi^{n+1}(w_2)$ such that $\frac{\partial^2 v(w)}{\partial w_1^2} = 0$ and $\frac{\partial v(w)}{\partial w_1} < 0$, and we repeat the procedure along the lines described in the previous section until we get a solution to (14-19).

We now briefly describe the Finite Element numerical method used to solve (20-23). For simplicity, we will describe the case when $\phi^0 = 0$ is employed. We begin by dividing $[0,1]\times[0,1]$ divided into a partition of $N\times N$ identical square elements, each containing a grid point at the intersection of its diagonals. Then, we express any function $\omega(.) : [0,1]\times[0,1] \to \mathbf{R}$ as a linear combination $\omega(w) = \sum_{b=1}^{N^2} c_b N_b$, of "hat" basis functions $N_b(.)$, defined to be equal to 1 at b (one of the N^2 grid points) equal to 0 at the perimeter of and outside the square element containing b, and linear within the square. We can also approximate the value function $v(w)$ in the same fashion by choosing suitable $\{d_b\}$ to set

$$v(w) = \sum_a d_b N_b. \quad (24)$$

Denoting the inner product of two functions in L_2 as $(f,g) \equiv \int\int fg\, dw_1\, dw_2$, and taking the inner product of equation (14) with an arbitrary function $\omega(.)$ we get

$$(\omega,h) - \gamma(\omega,v) + \frac{1}{2}\sum_{ij}\Sigma_{ij}(\omega,v^{ij}) + \sum_i \alpha_i(\omega,v^i) = 0, \quad (25)$$

where $v^{ij} \equiv \frac{\partial^2 v}{\partial w_i \partial w_j}$ and $v^i \equiv \frac{\partial v}{\partial w_i}$. Using the representation as a linear combination of hat functions, the fact that ω is arbitrary, and simplifying (25) yields an expression that can be written in matrix form as $Gd = H$ where G is a $N^2 \times N^2$ matrix and d and H are N^2 vectors. The vector d contains the elements d_b that we need to calculate in order to find an approximation for the value function as in (24), and G and h contain elements that can be expressed in terms of the hat functions and the primitives α and Σ. Thus we obtain an approximation for the value function at each grid point b as in (24). This is the essence of the Finite Element numerical method.

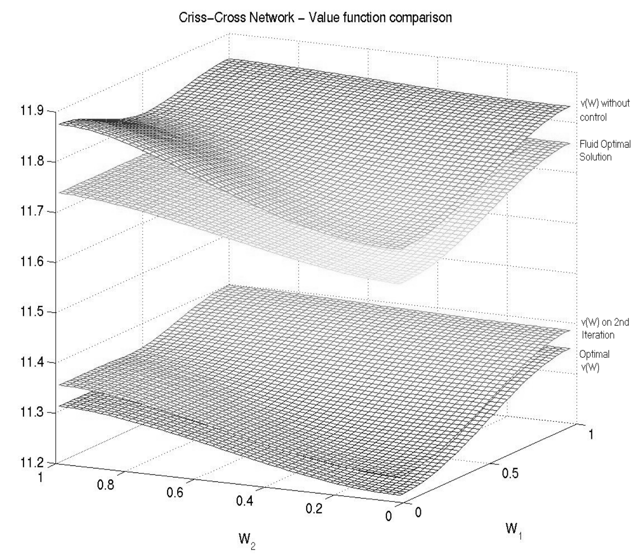

Figure 3: Value functions

We now turn our attention to the results obtained in the Criss-cross example using this method. For the example, we use a 50×50 grid. Figure 3 contains four surfaces as follows, going from top to bottom respectively. (1) No Control: The value function under the *laissez-faire* policy that applies the minimum amount of control to maintain the process $W(t)$ within $[0,1]\times[0,1]$. (2) Fluid Optimal Control: In the fluid (or non-stochastic) version of the problem, the optimal control is to translate any state with $w_1 < 0.5w_2$ to the line $w_1 = 0.5w_2$. The second surface shows the value function under this policy. (3) Second Iteration: This shows the value function after updating ϕ^0 to ϕ^1 in our method. (4) Optimal: This shows the value function after the procedure has converged to within a tolerance of 10^{-3}. The following key observations can be made. (a)The fluid optimal policy can be quite far from optimal in the Brownian network, achieving only a small fraction of the possible improvement in costs over the *laissez-faire* policy. (b) One iteration of our proposed procedure achieves a considerable portion of the possible benefits of using this method.

In light of (b) above, and the fact the brute-force procedure requires the inversion of a $50^2 \times 50^2$ matrix, we look for ways of obtaining policies that are "good" in the sense that they achieve a considerable fraction of the possible improvement in costs over the *laissez-faire* policy while being less computationally intensive. To

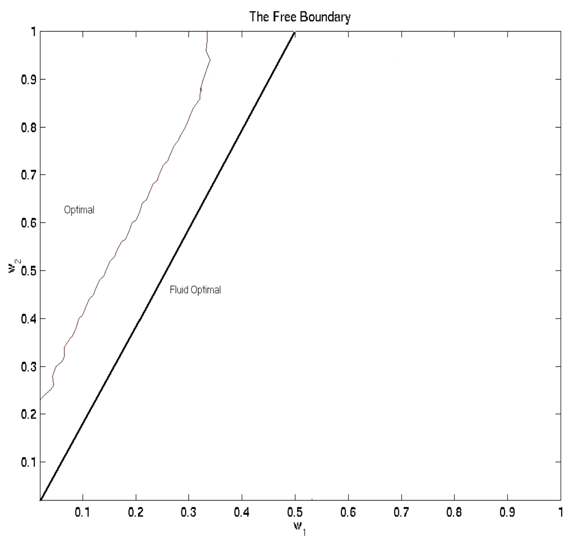

Figure 4: The Optimal Free Boundary

do this, we begin by plotting the optimal $\phi(.)$ or free boundary computed using our procedure and comparing it to the free boundary suggested by the fluid optimal solution. Figure 4 shows the computed optimal free boundary. As can be seen, modulo non-smoothness due to the discrete grid and end effects, a considerable part of the boundary can be considered a shifted version of the fluid optimal boundary, the line $w_1 = 0.5w_2$. So to find "good" policies easily, we look for policies that use boundaries that are just offsets of the line $w_1 = 0.5w_2$. Thus, the problem becomes essentially a single parameter search for the optimal offset.

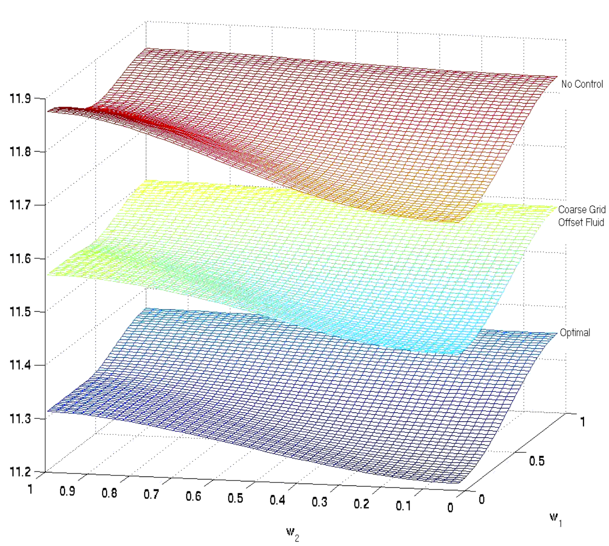

Figure 5: The Coarse-Grid Offset-Fluid Policy

Figure 5 illustrates the improvement over the *laissez-faire* policy that can be obtained by using applying one iteration of our method to compute the offset using a very coarse grid of just 5×5 points. As can be seen from the figure, nearly half the possible reduction in cost over the *laissez-faire* policy can be obtained in just one iteration, where the only significant computation is the inversion of a 25×25 matrix. Thus, we see that our method can yield good control policies with relatively small effort.

In conclusion, we note that, in this conference paper, we have not articulated the general method and that we have only illustrated its application to two interesting problems. A journal version, which is currently under preparation will contain the details of the general method.

References

[1] Benes, V.E., Shepp, L.A. and Witsenhausen, H.S., "Some Solvable Stochastic Control Problems," *Stochastics*, vol. 4, pp. 39-83, 1980.

[2] Harrison, J.M., *Brownian Motion and Stochastic Flow Systems*, Wiley, 1985.

[3] Harrison, J.M., "Brownian models of queueing networks with heterogeneous customer populations," in *Stochastic Differential Systems: Stochastic Control Theory and Their Applications*, IMA vol. 10, W. Fleming and P.L. Lions (eds.), Springer Verlag, New York, pp. 147-186, 1988.

[4] Harrison, J.M., "The BIGSTEP approach to flow management in stochastic processing networks," in *Stochastic Networks: Theory and Applications*, F.P. Kelly and S. Zachary, and I. Ziedins (eds.), Oxford University Press, pp. 57-90, 1996.

[5] Harrison, J.M., and Taksar, M.I., "Instantaneous Control of Brownian Motion," *Mathematics of Operations Research*, vol. 8, no. 3, pp. 439-453, 1983.

[6] Harrison, J.M., and van Mieghem, J.A., "Dynamic control of Brownian networks: state space collapse and equivalent workload formulations," *Annals of Applied Probability*, vol. 7, pp. 747-771, 1997.

[7] Harrison, J.M., and Wein, L.M., "Scheduling network of queues: Heavy traffic analysis of a simple open network," *Queueing Systems,* vol. 5, pp. 265-280, 1989.

[8] Kushner, H.J., and Dupuis, P.G., *Numerical Methods for Stochastic Control Problems in Continuous Time,* Springer, New York, 1992.

[9] Martins, L.F., Shreve, S.E., and Soner, H.M., "Heavy traffic convergence of a controlled multiclass queueing network," *SIAM J. Control and Optimization,* vol. 34, no. 6 , pp. 2133-2171, 1996.