Fluid Analysis of Arrival Routing

Michael H. Veatch,*Gordon College

July 2000

Abstract

In Hajek's arrival routing problem, customers are routed to one of n queues to minimize average holding cost. Interarrival and service times are exponentially distributed. We solve the associated *fluid model*. The optimal fluid policy tells us the asymptotic slopes of the switching surfaces in the original problem when the queues are large. If these slopes are nonzero, then numerical tests indicate that the fluid policy performs well in the original stochastic network. The fluid policy also indicates the approximate path that will be taken to recover from large queues: Routing only switches to queues with larger holding cost and once a large queue empties it will remain approximately empty.

Keywords: Arrival routing, fluid models, stochastic networks

*Department of Mathematics, Gordon College, Wenham, MA 01984, (978) 927-2300, veatch@gordon.edu

1 Introduction

Hajek [3] considers the following arrival routing problem. Upon their arrival, customers are routed to one of two servers, each with its own queue. After being served, they depart. The objective is to minimize long-run average holding cost. Under exponential service and interarrival time assumptions, he shows that the optimal policy has a threshold form: Route to server 1 if the number of customers at server 2 exceeds a threshold. This threshold is a nondecreasing function of the number of customers at server 1.

We analyze a *fluid model* [2] of this system with n servers. The motivation for studying the fluid model is that it is tractable while the original problem is not. Although the fluid model is transient and deterministic, it gives a surprising amount of information about the original problem. The analysis also gives insight into when the optimal policy for the fluid model does not contain useful information. Optimal fluid trajectories for this problem have a simple form. First, they avoid starving servers whenever possible. Routing of the remaining arrivals switches only to higher-cost servers with nonempty buffers. Once a buffer empties it remains empty. Thus, the control has at most $2n-1$ points at which it changes: n times when a buffer empties and up to $n-1$ switch points. Although we do not have a polynomial algorithm for computing these trajectories, a method is given for finding the switching surfaces so that the complete policy can be specified for small n.

Call the optimal policy for the original network the *discrete policy* and the optimal policy for the fluid model the *fluid policy*. We give numerical examples of using the fluid policy in the discrete system. These examples illustrate that the fluid policy can perform very poorly when the switching surface lies on the boundary of the state space. However, it is generally good when the switching surfaces lie in the interior. The numerical results comport nicely with theoretical results on the fluid and discrete switching surfaces. Because the discrete policy has the same asymptotic slopes of switching surfaces as the fluid policy, the fluid solution contains important

information about the nature of the discrete policy and whether or not the fluid policy is useful.

The fluid analysis of this problem is simplified because the fluid policy can be easily translated to the discrete system. Ambiguity arises only in those states where the fluid policy splits the arrivals among two or more servers. Even this control can be implemented as a randomized control in the discrete system. The difficulty that usually arises in translating sequencing controls is that the fluid policy moves fluid through an empty buffer, while in the discrete system a server must idle (see [5] for more discussion). This difficulty does not occur for arrival routing.

The next section introduces the discrete and fluid models. The fluid solution for two servers is given in section 3 with a proof of optimality. Section 4 gives the solution for n servers, with some of the proofs omitted. The relationship between the fluid and discrete policies is addressed in section 5, including the numerical examples.

2 The Discrete and Fluid Models

In Hajek's discrete model, customers arrive according to a Poisson process with rate λ. They are immediately routed to one of n servers, numbered $1, \ldots, n$, each with its own queue. Service time at server i is exponentially distributed with mean $1/\mu_i$. Holding cost is incurred at the rate $c_i > 0$. The state is $\Phi(t) \in Z_+^n$, where $\Phi_i(t)$ denotes the number of customers at server i at time t. The decision, at each arrival epoch, is which server to route to. For consistency with the fluid model, let $\widetilde{v}_i(t)$ be the probability that an arrival at time t is routed to i and $\widetilde{u}_i(t)$ indicate whether or not server i is busy at time t. The tilde denotes the discrete model. Let c, $\widetilde{v}(t)$ and $\widetilde{u}(t)$ be vectors defined in the natural way. A feasible control must satisfy

$$\sum_{i=1}^{n} \widetilde{v}_i(t) = 1 \tag{1}$$

$$\widetilde{u}_i(t) = 0 \ \text{ if } \ x_i = 0, \quad \widetilde{v}(t) \geq 0, \quad 0 \leq \widetilde{u}(t) \leq 1$$

at each t. The objective is to minimize long-run average holding cost. The appropriate control for this problem is a stationary state-feedback control. In a slight abuse of notation, let $\widetilde{u}$ denote the state-feedback control $(\widetilde{u}(x), \widetilde{v}(x))$ in state $\phi(t) = x$.

The fluid model is obtained when all transitions are replaced by their mean rates and a continuous state is used; $x_i(t)$ is the length of the server i queue at time t, with $x(t) \in R_+^n$. Under the stability condition $\lambda < \sum_i \mu_i$, the fluid model will drain from any initial state; i.e., for initial state x we can choose a time horizon T such that $x(t) = 0$ for all $t \geq T$ for a suitable class of policies. Hence, the fluid control problem uses "cost to drain" as its objective:

$$\begin{aligned} J(x) = \min \quad & \int_0^T c'x(t)dt \\ & \dot{x}_i(t) = \lambda v_i(t) - \mu_i u_i(t) \\ & \sum_{i=1}^{n} v_i(t) = 1 \\ & x(0) = x \\ x(t) \;\; \geq \;\; & 0, \quad v(t) \geq 0, \quad 0 \leq u(t) \leq 1. \end{aligned} \tag{2}$$

If $v_i(t) = 1$, arrivals are routed to class i at t. The control u_i has been added to enforce idling. It is optimal to set $u_i(t) = 1$ if $x_i(t) > 0$ and $u_i(t) = 0$ otherwise. The greedy policy will avoid starving servers whenever possible and then route to the server with smallest holding cost. In light of this, we order the classes $c_1 \leq \ldots \leq c_n$.

3 Fluid Solution for Two Servers

For $n = 2$ servers, we consider two cases. If $\lambda \geq \mu_2$, the fluid policy is greedy. It routes to 1 whenever $x_2 > 0$. When $x_2 = 0$, it avoids starving server 2 by setting

$$u_1 = 1 - \mu_2/\lambda \;\text{ and }\; u_2 = \mu_2/\lambda. \tag{3}$$

The second case, $\lambda < \mu_2$, has a switching curve: Route to server 2 if

$$x_2 \leq \gamma x_1 = \frac{c_1}{c_2}\left(\frac{\mu_2 - \lambda}{\mu_1}\right) x_1, \tag{4}$$

otherwise route to server 1. Note that this policy makes a tradeoff between the higher short term cost of routing to server 2 and postponing the starvation of server 2.

To show that this policy is optimal for the first case, observe that server 2 is never starved. Consequently, this policy postpones starvation as long as possible and minimizes the total queue length $x_1(t) + x_2(t)$ at every t. Since $c_1 \leq c_2$, it also minimizes the cost rate $c'x(t)$ at every t.

In the second case, we make use of the scaling property of fluid policies: If a control is optimal in state x, it is also optimal in state αx for $\alpha > 0$. Furthermore, we need only consider extreme point controls, where $v_i(x) = 0$ or 1, except on regions of smaller dimension (such as $x_2 = 0$). Hence, the policy must consist of regions of constant control bounded by linear switching curves through the origin. Clearly, the routing is to server 2 when $x_2 = 0$. Let $\{x : x_2 \leq \gamma x_1\}$ be the maximal region containing the boundary $x_2 = 0$ in which routing is to server 2. When routing to server 2 and $x > 0$,the trajectory has slope $(\mu_2 - \lambda)/\mu_1$. First, assume $0 < \gamma \leq (\mu_2 - \lambda)/\mu_1$. With this upper bound, trajectories only pass from above the switching curve $x_2 = \gamma x_1$ to below it and the routing switches from server 1 to server 2. Consider the following perturbation of a trajectory: Shift the switching curve by decreasing γ slightly. Any trajectory that crosses the curve will be perturbed by some Δx, where $\Delta x_1 = -\Delta x_2$. For γ to be the an optimal switching curve, it must satisfy the first order condition $\partial J/\partial \gamma = 0$, or $\partial J/\partial x_1 = \partial J/\partial x_2$. From an initial state x on the switching curve, routing is to server 2 and the cost is

$$J(x) = \frac{c_1}{2\mu_1} x_1^2 + \frac{c_2}{2(\mu_2 - \lambda)} x_2^2.$$

The first order condition gives the value of γ in (4). Furthermore, the cost of any trajectory that starts above $x_2 = \gamma x_1$, routing to server 1 until it crosses the curve

and to server 2 afterward, is convex in γ. Hence, γ is a local minimum.

Now suppose $\gamma > (\mu_2 - \lambda)/\mu_1$. Let x be an initial state with $x_2 \geq ((\mu_2 - \lambda)/\mu_1)\, x_1$. Then $x_2(t) > 0$ whenever $x_1(t) > 0$. The switching curve $\gamma' = (\mu_2 - \lambda)/\mu_1$ routes more arrivals to server 1 and still does not starve server 2. Hence, γ' has at least as small a total queue length and cost rate as γ at every t, contradicting the assumption. The final possibility, $\gamma = 0$, is eliminated by continuity and convexity in γ.

We have shown that (4) is an optimal switching curve. In principle, there could be additional switching curves above the one we have found, however, they would violate the monotonicity established by Hajek. The absence of additional switching curves can also be established by calculating the cost of the resulting trajectories, as we have done above. We have established optimality by direct cost calculations. The separated continuous linear programming duality approach of [1] could also be used.

4 Fluid Solution for n Servers

This section presents similar results for n servers; however, only first order conditions are given for the switching surfaces and these conditions are not explicit. Instead, an algorithm is informally described for computing the switching surfaces. Numerical results and the results for two servers suggest that the first order conditions do in fact give the optimal policy.

Let M be the integer satisfying

$$\sum_{i=M}^{n} \mu_i > \lambda \geq \sum_{i=M+1}^{n} \mu_i.$$

The fluid policy avoids starving whenever possible and can be chosen so that servers $M+1, \ldots, n$ never starve. Let $E(x) = \{i : x_i = 0\}$. Avoiding starving requires routing $\lambda_0(t) = \sum_{i \in E(x(t))} \mu_i$ to empty buffers. The remaining decision is how to route $\lambda_1(t) = \lambda - \lambda_0(t)$. We will describe policies by how they route $\lambda_1(t)$; essentially, this

is the routing to nonempty buffers. We consider only policies that do not split $\lambda_1(t)$ between servers, since splitting is not needed to achieve optimality. This part of the problem ends at the starvation time $\tau_S = \min\{t : \lambda_1(t) < 0\}$. No servers are starved before τ_S. Let $x(t)$ be an optimal trajectory and $t_i = \liminf \{t > 0 : x_i(t) = 0\}$. Note that if $x_i(0) = 0$ but the initial routing makes $\dot{x}_i(0) > 0$, then t_i is the *next* time buffer i empties.

It is optimal to route to the server with the minimal index $k_i(x) = D_{e_i} J(x)$. This directional derivative will exist for all x and can be thought of as the incremental cost per unit of fluid initially in buffer i. Furthermore, [1] shows that $k_i(x)$ is continuous along a trajectory. Thus, if the optimal routing switches from server i to server j at x, then

$$k_i(x) = k_j(x). \tag{5}$$

Consider a small amount of additional fluid initially in buffer i. Relative to $x(t)$, the additional fluid stays in buffer i until t_i. If $t_i \geq \tau_S$ then the additional fluid leaves the system at t_i. If $t_i < \tau_S$ and routing is to l at t_i, then the additional fluid is in buffer l from t_i until t_l. This shifting continues until τ_S, giving a sequence of cost terms

$$k_i(x) = c_i t_i + c_l(t_l - t_i) + \dots \tag{6}$$

Theorem 1 *The fluid policy has the following properties.*

1. *Never route to servers* $M + 1, \dots, n$.
2. *Never switch to an empty buffer at* $t > 0$.
3. *Only switch to higher-cost servers (from* i *to* j, $j > i$*).*
4. *If* $t_i < \tau_S$*, then never route to servers* i *or higher.*

Proof. Property 1 follows from the fact that these servers never starve. Property 2: Switching from server i to j when $x_j = 0$ before τ_S means that x_j will increase. If it is optimal to route to j for $x_j > 0$, then there is an i to j switching surface in the interior of the state space. Since all switching surfaces are linear and through the origin, the

trajectory must have been in the region where server j is preferred to i before t_j. But then x_j would have increased and the trajectory could not have reached $x_j = 0$. Property 3: Suppose it is optimal to switch from server i to server j at x, so that (5) holds. By Property 2, $x_j > 0$. First, assume $x_i > 0$. In a previous small interval Δt, the indices have changed by $\Delta k_i = k_i(t - \Delta t) - k_i(t) = c_i \Delta t$ and $\Delta k_j = c_j \Delta t$ because only the first term in (6) changes. But optimality requires $\Delta k_i \leq \Delta k_j$, $c_i \leq c_j$, and, if we break ties appropriately, $j > i$. Property 4: The optimal routing cannot be to server i at t_i^- because this would mean that $\lambda_1(t_i^-) < \mu_i$ and $t_i = \tau_S$. By Property 2, it is not optimal to switch to server i after it empties. Now suppose routing is to a higher cost server $j > i$ for some interval of length Δt while server i is empty. Change the policy by routing a small increment Δx to server i at the beginning of the interval. At the end of the interval, shift the same amount back to server j by suspending the portion of λ_0 being routed to server i. This change shifts Δx from buffer j to i for time Δt, reducing cost. Therefore, it is never optimal to route to a higher cost server while buffer i is empty. By Property 3, the routing also must be to lower cost servers at earlier times. □

A consequence of Theorem 1 is that trajectories are acyclic, in the sense that once a buffer empties it remains empty. It also simplifies condition (5).

Corollary 1 *The switching surface from server i to j satisfies*

$$c_i t_i = c_j t_j. \tag{7}$$

Proof. If a trajectory switches from server i to j at x, then, by Property 4, $t_i \geq \tau_S$. Hence, (6) contains only the first term and (5) simplifies to (7). □

The optimal policy never routes to i again, so $t_i = x_i/\mu_i$. Because t_j depends on the policy, we do not have a simple algorithm for computing an optimal trajectory. Let τ be the time of the next switch after switching to j at time 0. If there are no more switches, set $\tau = t_j$. Again using Property 3,

$$t_j = \frac{x_j + \int_0^\tau \lambda_1(s) ds}{\mu_j}. \tag{8}$$

Note that τ and $\lambda_1(\cdot)$ depend on x. These conditions are consistent with those in Section 3 for two servers, as can be seen by setting $i = 1$, $j = 2$, $\tau = t_2$ and $\lambda_1(s) = \lambda$. For $i < j$, (7) has a positive solution if and only if $j \leq M$. The parameter M specifies which servers have switching surfaces in the interior of the state space.

In principle, the following approach can be used to compute the switching surfaces. However, the number of cases to be computed is exponential in n. First, find the i, M switching surfaces. Setting $j = M$, there are no more switches, so $\tau = t_j$. Knowing the future control, find $\lambda_1(s)$ in terms of x and solve (8) and (7). Next, find the $i, M-1$ switching surfaces (set $j = M-1$). To find τ, consider the trajectory that continues routing to $M-1$. If it intersects the $M-1, M$ switching surface, the intersection is at τ. If not, set $\tau = t_j$. Again, the future routing is known and we can find $\lambda_1(s)$. Continue in this fashion, decreasing j to find all of the switching surfaces.

5 Comparing the Fluid and Discrete Policies

We have compared the fluid and discrete policies for three examples. Example 1 has two servers and no interior fluid switching curve, example 2 has two servers and a switching curve, and example 3 has three servers and switching surfaces. The discrete policy was found using dynamic programming value iteration on a truncated state space. The policies were compared by computing their average cost in the discrete model. For convenience, cost of the fluid policy was computed using a "value iteration" algorithm without the minimization operator.

The fluid policy was translated to the discrete model as follows. In example 1, if $x_2 = 0$ and $x_1 > 0$, route to 2 instead of splitting the arrivals as in (3). The control used when exactly on a switching curve, e.g., when equality holds in (4), does not affect fluid trajectories but does affect discrete cost. At $x > 0$ we used (4) for example 2 and the analogous inequality for example 3: Route to the higher cost server when on a switching surface. At the origin, we first tried routing to the server with minimum

index c_i/μ_i. This index gives the optimal control in light traffic and matches the discrete policy in examples 1 and 2. However, the discrete policy in example 3 routes to server 3 when $x = 0$, which does not match this index. To see how well the fluid policy could perform with appropriate corrections on the boundary, we changed the routing to match the discrete policy.

The fluid and discrete policies for example 1 are shown in Figure 1. Routing is to server 1 above the curve and server 2 below it. Average cost is 2.41 for the discrete policy and 7.78 for the fluid policy (223% suboptimal). For example 2, average cost is 1.18 for the discrete policy and 1.36 for the fluid policy (15% suboptimal). Figure 2 shows the similarity between the discrete and fluid switching curves (minor translation effects for the fluid curve, described above, are not shown). For example 3,with parameters $\lambda = 1$, $\mu = (0.5, 0.5, 1.5)$, and $c = (1, 2, 4)$, average cost is 0.86 for the discrete policy and 0.91 for the fluid policy (6% suboptimal). The fluid switching surfaces are given by

$$\begin{aligned} \text{switch } 1 &\rightarrow 2: \quad x_2 = 0.7x_1 - 0.8x_3 \\ \text{switch } 2 &\rightarrow 3: \quad x_3 = 0.5x_2 \\ \text{switch } 1 &\rightarrow 3: \quad x_3 = 0.25x_1. \end{aligned}$$

The discrete switching surfaces, viewed in cross sections of the state space, resemble the fluid. The main features of fluid trajectories can be seen in Figure 3, which plots $x(t)$ for one initial state. It uses the parameters of example 3 except that a fourth server has been added that never starves. Buffer 4 empties first, without being starved. Routing switches from server 1 to 2 to 3, then buffer 3 empties and starvation occurs. Finally, buffers 2 then 1 empty. As required by (7), when switching from server 1 to server 2, the times until these buffers drain are in the ratio $t_1/t_2 = c_2/c_1 = 2$. Similarly, when switching from server 2 to server 3, $t_2/t_3 = c_3/c_2 = 2$.

These numerical tests suggest that the fluid policy is similar to the discrete policy when there are switching surfaces on the interior of the state space. Theoretical results

support this finding. [3] demonstrates that the discrete policy for two servers has a threshold form and that the switching curve is a nondecreasing function. [9] shows that it is *always* unbounded. Fluid scaling arguments establish a link between the discrete policy and the fluid policy: The fluid limit obtained by scaling the discrete policy achieves the optimal fluid cost [6] , [7], [8]. An important implication of this optimality is that the discrete policy has the same asymptotic slopes of switching surfaces as the fluid policy. This relationship was pointed out in [7] for the case of interior switching surfaces. Thus, when the fluid policy switches in the interior it captures the main feature of the discrete policy, but when it switches on the boundary it is infinitely far from the discrete switching curve.

References

[1] F. Avram, D. Bertsimas, and M. Ricard, "Fluid models of sequencing problems in open queueing networks: An optimal control approach," F. Kelly and R. Williams, eds., *Stochastic Networks,* vol. 71 of the Proceedings of the IMA, pp. 199-234. New York: Springer-Verlag, 1995.

[2] H. Chen and A. Mandelbaum, "Discrete flow networks: Bottleneck analysis and fluid approximations," *Math. Oper. Res.* vol. 16, pp. 408-446, 1991.

[3] B. Hajek, "Optimal control of two interacting service stations," *IEEE Trans. Automat. Contr.*, vol. 29, pp. 491-499, 1984.

[4] X. Luo and D. Bertsimas, "A new algorithm for state-constrained separated continuous linear programming," *SIAM J. Control and Optim.*, vol. 37, pp. 177-210, 1998.

[5] C. Maglaras, "Discrete-review policies for scheduling stochastic networks: Fluid-scale asymptotic optimality," submitted to *Adv. Appl. Prob.*

[6] S. Meyn, "The policy improvement algorithm for Markov decision processes with general state space," *IEEE Trans. Automat. Control.*, vol. AC-42, pp. 191-197, 1997.

[7] ———, "Stability and optimization of multiclass queueing networks and their fluid models," vol. 33 of *Lectures in Applied Mathematics*, pp. 175-179. American Mathematical Society, 1998.

[8] ———, "Feedback regulation for sequencing and routing in multiclass queueing networks," working paper http://black.csl.uiuc.edu:80/~meyn, 1998.

[9] S. Xu and H. Chen, "On the asymptote of the optimal routing policy for two service stations," *IEEE Trans. Automat. Control*, vol. 38, pp. 187-189, 1993.