On the Incomplete Results for the Multi-Server Slow Server Problem*

Francis de Véricourt

Fuqua School of Business
Duke University
Durham, NC 27708
U.S.A.
(p) 1-919-660-7818
(f) 1-919-923-4924
fdv1@duke.edu

Yong-Pin Zhou

Business School
University of Washington
Seattle, WA 98195-3200
U.S.A.
(p) 1-206-221-5324
(f) 1-206-543-3968
yongpin@u.washington.edu

August 2005

Abstract

In this note, we show that existing results for optimal routing policies in the slow server problem with more than two heterogeneous servers are incomplete.

KEY WORDS: Multiple Heterogeneous Servers, Slow Server Problem, Call Centers

*Short Title: Multi-Server Slow Server Problem

1 Introduction

The multi-server slow server problem refers to a queueing control problem in which the job arrival follows a stationary Poisson process with rate λ, and there are K heterogeneous exponential servers with rates $\mu_1 \geq \mu_2 \geq \ldots \geq \mu_K$. The objective is to find a non-preemptive routing policy that minimizes the long-run average time in the system. A particular interesting application is to the call routing problems that occur in contact centers where the service speed varies significantly among the customer representatives. For recent work see for instance Armony [1] or de Véricourt and Zhou [2].

The optimal routing policy for the special case of $K = 2$, the threshold policy, has been extensively studied (e.g. Larsen [5], Larsen and Agrawala [6], Hajek [3], Lin and Kumar [7], Walrand [10], and Koole [4]), where the faster server is always preferred, and the slow server will be used only when the faster server is busy and the number of jobs in the queue exceeds a certain threshold policy. The condition of two servers limits the application of such models, however. In many instances, it is natural to use this model with more than two servers.

Results concerning the optimal policy for $K \geq 3$ are much more challenging to obtain. The growing dimensionality of the underlying state space accounts for the difficulty. Weber [11] uses a coupling argument to show that whenever a job is routed, it should always be routed to the fastest available server; but he only provides a conjecture that the optimal routing follow a state-dependent threshold policy. Recently, two papers claim to have proved the optimality of the state-dependent threshold policy for $K \geq 3$. The first one, Rykov [9], uses value iteration to show that the optimal value function satisfies monotonicity properties. The second one, Luh and Viniotis [8] uses a Linear Programming formulation and sample path analysis. We show in this note, however, that the results in both papers are incomplete. We hope that our observations should motivate future research in the field to tackle this fundamental queuing control problem.

2 Value Iteration Approach

Rykov [9] assumes a finite system with K servers and N spaces for jobs including those in service. He denotes by $x = (q, d_1, \ldots, d_K)$, the state of the system where q is the number

of jobs waiting in the queue and d_j is the state of Server j ($d_j = 1$ if Server j is busy, and $d_j = 0$ otherwise). He also introduces the sets of indices $J_0(x) = \{j : d_j(x) = 0\}$, $J_1(x) = \{j : d_j(x) = 1\}$ and $A_0(x)$ such that,

$$A_0(x) = \begin{cases} J_0(x) \cup \{0\} & \text{for } x \text{ with } \quad q(x) < N - K \\ J_0(x) & \text{for } x \text{ with } \quad q(x) = N - K \end{cases},$$

He then consider the shift operators S_0 and S_j,

$$\begin{aligned} S_0 x &= x + e_0 \mathbf{1}_{\{q(x)<N\}} \\ S_j x &= x + e_j \mathbf{1}_{\{j \in J_0(x)\}} \end{aligned}$$

where e_j is the $(K+1)$-dimensional vector whose jth coordinate (beginning from 0th) is one and all others are zeros. Their respective inverse operators are denote by S_0^{-1} and S_j^{-1} (with $S_0^{-1}x = x$ if $q(x) = 0$ and $S_j^{-1}x = x$ if $j \in J_0(x)$).

$$\begin{aligned} S_0^{-1} x &= x + e_0 \mathbf{1}_{\{q(x)<N\}} \\ S_0 x &= x + e_j \mathbf{1}_{\{j \in J_0(x)\}} \end{aligned}$$

Based on these definitions, the following operators are introduced:

$$T_0 w(x) = \max[w(S_k x) : k \in A_0(x)] \tag{1}$$

$$T_j w(x) = \begin{cases} T_0 w(S_0^{-1} S_j^{-1} x) & \text{for } \quad j \in J_1(x), \quad q(x) > 0, \\ w(S_j^{-1} x) & \text{for } \quad j \in J_1(x), \quad q(x) = 0, \\ w(x) & \text{for } \quad j \in J_0(x). \end{cases} \tag{2}$$

He then formulates the problem as a Markov Decision Process and derives the following optimality equations (equation 9),

$$w(x) = r(x) + \lambda T_0 w(x) + \sum_{1 \le j \le K} \mu_j T_j w(x) = B w(x),$$

where $w(x)$ is the revenue function, and T_0 and T_j are defined in (1) and (2).

Rykov's proof depends on showing that the value iteration preserves the following two properties of the value function:

(i) the function is non-increasing, and

(ii) the function has monotone increments.

That is, he must show that T_0, T_j, and B preserve (i) and (ii). Rykov [9] shows that the operator B preserves properties (i) (Theorem 2) and (ii) (Theorem 3), if T_0 and T_j preserve (i) and (ii).

Therefore, he also needs to show that operators T_0 and T_j preserve properties (i) and (ii). He provides detailed proofs for T_0 (Lemmas 1 for (i) and Lemma 2 for (ii)), and deduces directly that T_j also preserves these properties. We show in the following that such deductions are not straightforward and his proof is incomplete.

3 T_j and Lemma 1

For Theorem 2 to be true, operators $T_j, \forall j$, needs to preserve property (i) (Lemma 1), as stated in the *Remark* on Page 397. In particular, if $h(\cdot)$ is a non-increasing function such that $h(S_j x) \leq h(x)$ for all $j \in J_0(x)$, then we need to show that $T_i h(S_j x) \leq T_i h(x)$ for all i and $j \in J_0$. However, for $i = j$,

$$\begin{aligned} T_j h(S_j x) &= T_0 h(S_0^{-1} x) \\ &= \max(h(S_0^{-1} S_l x) : l \in A_0(x)) \\ &\geq h(S_0^{-1} S_0 x) \\ &= T_j h(x). \end{aligned}$$

Unless $T_j h(S_j x) = T_j h(x)$ for all x and j such that $j \in J_0(x)$ and $N - K > q(x) > 0$, which is unlikely to be true with this formulation (there are strictly more jobs in the system in state $s_j x$ than in State x), T_j does not satisfy Lemma 1.

One approach to fixing this problem is to re-formulate the problem by allowing the decision maker to route multiple calls at the same time (see the definition of $\hat{T}_0$ below). Note that T_0 makes the best single routing in state $S_0 x$, and $\hat{T}_0$ makes the best multiple routings in state x. Since in steady state it is optimal to route at most one job at a time (see de Véricourt and Zhou [2]), we have $T_0 w(x) = \hat{T}_0 w(S_0 x)$ in the recurrent region of the state space (elsewhere, $T_0 w(x) \leq \hat{T}_0 w(S_0 x)$). We then have the following optimality equations

$$\hat{B} w(x) = r(x) + \lambda \hat{T}_0 w(S_0 x) + \sum_j \mu_j \hat{T}_j w(x) = w(x),$$

where $\hat{B}$ is the optimal operator and the operators $\hat{T}_0$ and $\hat{T}_j, \forall j$, are recursively defined as:

$$\hat{T}_0 w(x) = \begin{cases} w(x) & \text{if } q(x) = 0 \text{ or } J_0(x) = \emptyset, \\ \min\{\hat{T}_0 w(S_k S_0^{-1} x), w(x) \mid k \in A_0(x)\} & \text{if } q(x) > 0 \text{ and } J_0(x) \neq \emptyset, \end{cases}$$

and

$$\hat{T}_j w(x) = \begin{cases} \hat{T}_0 w(S_j^{-1} x) & \text{for } j \in J_1(x) \\ w(x) & \text{for } j \in J_0(x) \end{cases}.$$

It is then possible to show that the optimal operator $\hat{B}$ propagates non-increasing functions (see Theorem 1 in de Véricourt and Zhou [2], which shows this property for their corresponding cost minimization problem).

4 T_j and Lemma 2

Similarly, for Theorem 3 to be true operators T_j needs to preserve Property (ii), monotone increments (Lemma 2). That is, for $k = \arg\max\{\mu_l : l \in J_0(x)\}$, $\Delta_{0k} T_0 w(x) = T_0 w(S_0 x) - T_0 w(S_k x)$ is non-increasing in x if $\Delta_{0k} w(x)$ is non-increasing in x. (The fact that k is equal to $\arg\max\{\mu_l : l \in J_0(x)\}$ is used in cases 2 and 4 of the proof of Lemma 2 in [9].)

Consider then j such that $\mu_j > \mu_k$. From the definition of k, j belongs to $J_1(x)$. For x such that $N - K > q(x) > 0$, $\Delta_{0k} T_j w(x) = \Delta_{0k} T_0 w(S_0^{-1} S_j^{-1} x)$. If we could apply Lemma 2, then the desired result would be immediate. However server j is now free in state $S_0^{-1} S_j^{-1} x$ (i.e. $j \in J_0(S_0^{-1} S_j^{-1} x)$) and we have $k \neq \arg\max\{\mu_l : l \in J_0(S_0^{-1} S_j^{-1} x)\} = j$. Therefore, Lemma 2 cannot be applied to $\Delta_{0k} T_0 w(S_0^{-1} S_j^{-1} x)$, and there is no guarantee that T_j satisfies Lemma 2.

Unfortunately, allowing multiple routing at the same time, as we proposed for Lemma 1, does not fix this problem, and the proof still needs to be completed.

5 Sample Path Approach

In their paper, [8], Luh and Viniotis formulate the finite-horizon routing problem as a Linear Program. Then they use a sample path argument (similar to a coupling argument) to show that the optimal policy is a state-dependent threshold policy: for any policy that is not

the state-dependent threshold policy, one can construct a corresponding state-dependent threshold policy that is both feasible (for the Linear Program) and has a better objective function value. Specifically, they show in Lemma 4 that it is optimal to always route a job to the fastest server whenever possible. Then Lemma 4 is used in later proofs to show the optimality of threshold policy.

Since the statement of Lemma 4 (esp. the phrase "whenever possible") is vague, we show in the following that either the proof of Lemma 4 is incomplete, or it does not cover all the possible cases so that it cannot be used in the later proofs. In either case, as it stands, the results in Luh and Viniotis [8] are also incomplete.

Luh and Viniotis use the following Linear Program (call it LP) in the proofs of both Lemmas 4 and 5:

$$\begin{aligned} &\max_{\{v_j^i(\omega^j)\}} c \cdot v && <20> \\ &\mathbf{A} \cdot v \leq b(x^0) && <21> \\ &0 \leq \textstyle\sum_{k=l^*(\omega^{j-1})}^{j} v_k^i(\omega^k) \leq 1 && <21a> \\ &\text{where } 1 \leq i \leq N, \omega^j = \omega^{j-1} \not{\emptyset}_i. \end{aligned}$$

Here, ω^j is a sample path of j uniformized events and all the v_j^is are the decision variables: $v_j^i = 1$ represents allocating a call to server i at the j^{th} uniformized event, and v_j^i=0 represents not allocating a call to server i at the j^{th} uniformized event (for more details see [8]).

The equations are numbered as in [8] except for <21a> which is not numbered in the original paper (note that constraints <21a> plays a crucial role in the proof). Constraints <21> correspond to the constraint that the queue cannot be negative. Constraints <21a> represent the fact that the state of a server cannot be negative before an action and cannot be overfull after an action.

Based on LP, Luh and Viniotis then show Lemma 4 which states that "there exists an optimal policy that activates faster than slower servers, whenever possible". To prove this result, the authors establish that, if server i is faster than server j (i.e., $\mu_i > \mu_j$), then the corresponding costs in the objective function are such that $c_k^i(\omega^k) > c_k^j(\omega^k)$. Moreover, they consider a vector of decision variables $\bar{s}$ that is feasible for the LP, and a corresponding vector s that differs from $\bar{s}$ only in one component k for which $\bar{s}$ allocates a call to server j

while s allocates a call to server i:

$$\bar{s}_k^i(\omega^k) = 0 \qquad \bar{s}_k^j(\omega^k) = 1, \tag{3}$$

$$s_k^i(\omega^k) = 1 \qquad s_k^j(\omega^k) = 0. \tag{4}$$

Decisions for all other events are the same ($\bar{s}_l^i(\omega^l) = s_l^i(\omega^l), l \neq k$). The authors then argue that, since $\bar{s}_k^i(\omega^k)$ and $s_k^j(\omega^k)$ appear together in every constraint of LP, s is also feasible for LP (it is indeed clear that they appear together only in <21>). Because s gives a larger objective function value than $\bar{s}$, the authors deduce then that s is a better solution.

The proof of Lemma 4 does not check that vector s, as constructed in (3) and (4), also satisfies constraints <21*a*> of LP. As a result s is not necessarily feasible for LP. Actually, there are many vector $\bar{s}$ that satisfy <21>, <21*a*>, and (3), but for which the associated s given by (4) is not feasible for <21*a*>. For example, let $\bar{s}$ satisfies <21>, <21*a*>, and

$$\bar{s}_k^i(\omega^k) = 0, \qquad \bar{s}_k^j(\omega^k) = 1, \qquad \text{and} \qquad \bar{s}_{k+1}^i(\omega^k A) = 1. \tag{5}$$

Then the s constructed according to (4) should satisfy

$$s_k^i(\omega^k) = 1, \qquad s_k^j(\omega^k) = 0, \qquad \text{and} \qquad s_{k+1}^i(\omega^k A) = 1. \tag{6}$$

The s in (6) satisfies <21> but it clearly does not satisfy <21*a*>. Intuitively, Luh and Viniotis's proof states that if a policy corresponding to $\bar{s}$ assigns a job at time k to server j instead of server i, then one can do better by assigning this job to server i instead of server j. This intuition is correct, but the proof fails to consider all the cases that may occur after time k. It only considers the cases in which the new assignment policy (s) can be completely coupled with the original assignment policy ($\bar{s}$) after time k. The example we give above is one in which $\bar{s}$ assigns a job to server j at time k, and assigns another job to server i at time $k+1$ when the event at time $k+1$ corresponds to an arrival. The policy corresponding to s should assign a job to server i (instead of j) at time k, but obviously it cannot assign another job to server i at time $k+1$ as $\bar{s}$ does.

Therefore, Lemma 4 does not cover all the possible sample paths. So it is flawed. Of course, the phrase "whenever possible" in the statement of Lemma 4 can be interpreted as "whenever feasible" so that the example we construct above does not apply. In this case, Lemma 4 is correct but since it does not cover all the cases, it cannot be used in the later proofs.

6 Conclusion

The slow server problem with more than 2 heterogeneous servers is still an open problem with important applications. Rykov [9] and Luh and Viniotis [8] have proposed different but incomplete approaches to tackle this issue. Despite these flaws, their approaches are quite insightful, and they have sparked some renewed interest in the slow server problem.

Acknowledgment

The authors are grateful to Ger Koole, Paul Luh, and Vladimir Rykov for their valuable comments.

References

[1] M. Armony, Dynamic routing in large-scale service systems with heterogenous servers, Working Paper, NYU (2004).

[2] F. de Véricourt and Y.-P. Zhou, Managing response time and service quality in a call allocation problem, Forthcoming Operations Research (2004).

[3] B. Hajek, Optimal control of two interacting service stations, IEEE Transactions on Automatic Control 29 (1984) 491-499.

[4] G. Koole, A simple proof of the optimality of a threshold policy in a two-server queueing system, Systems & Control Letters 26 (1995) 301-303.

[5] R. Larsen, Control of multiple exponential servers with application to computer systems, Ph.D. Dissertation, Department of Computer Science, University of Maryland, College Park, 1981.

[6] R. Larsen and A. K. Agrawala, Control of a heterogeneous two-server exponential queueing system, IEEE Transactions on Software Engineering 9 (1983) 522-526.

[7] W. Lin and P.R. Kumar, Optimal control of a queueing system with two heterogeneous servers, IEEE Transactions on Automatic Control AC-29 (1984) 696-703.

[8] H. Luh and I. Viniotis, Threshold control policies for heterogeneous server systems, Mathematical Methods of Operations Research 55 (2002) 121-142.

[9] V.V. Rykov, Monotone control of queueing systems with heterogeneous servers, Queueing Systems 37 (2001) 391-403.

[10] J. Walrand, A note on "Optimal control of a queueing system with two heterogeneous servers", Systems & Control Letters 4 (1984) 131-134.

[11] R. Weber, On a conjecture about assigning jobs to processors of differing speeds, IEEE Transactions on Automatic Control 38 (1993) 166-170.