Worst-case and Nash Routing Policies in Parallel Queues with Uncertain Service Allocations

Eitan Altman* and Nahum Shimkin†

March 12, 1993

Abstract

We consider two dynamic routing problems into several parallel queues. First, we consider the case where the total service rate in all queues is constant, but its distribution between the queues may vary. Arriving customers are routed by a controller, with the purpose of minimizing the total discounted holding cost under worst-case service conditions. A zero-sum game analysis yields that the optimal strategy for the router is 'join the shortest queue', while the worst service strategy is 'serve the shortest queue'. The second problem considered is individually optimal routing in a similar queueing system, where a single server exists which always serves the longest queue, and each customer seeks to minimize its own sojourn time. Nash equilibrium routing strategies are characterized for this case.

Subject classification: Queues, optimization: dynamic routing in parallel queues. Games/group decisions, stochastic games: controlled queueing systems.

Introduction

We consider two dynamic routing problems into $N \geq 2$ parallel queues. We first consider the socially optimal control problem where a single router wishes to minimize a discounted sum of the holding costs in the different queues. Arrival to the system are Poisson, with exponentially distributed i.i.d. service requirements. The queues are attended by several servers which are free to move between the queues. Fixed servers which are identical in all queues may also be present. Finite buffers are assumed, possibly of different lengths. Allocation of the free servers (or service

*INRIA, Centre Sophia Antipolis, 06565 Valbonne Cedex, France.

†Institute for Mathematics and its Applications, University of Minnesota, Minneapolis, MN 55455, USA.

rate) may change in time in a way unpredictable to the controller. A plausible objective of the router is then to design a control strategy that guarantees the best performance (lowest cost) under the worst service allocation. The problem is thus viewed as a zero-sum stochastic game between the routing controller and a service controller, which controls the service allocation. We prove that the optimal (min-max) strategy for the router is "join the shortest queue". The worst service conditions are shown to be "serve the shortest queue" (i.e., assign all free service to a shortest queue).

This extends well-known results on the optimality of "join the shortest queue" in symmetric queues with fixed servers (Winston (1977), Weber (1978)).

The second problem concerns the case where a single server is shared by the queues. As opposed to the previous case, we take here the optimistic viewpoint in assuming that the server always attends the **longest** queue. We then consider the problem of individual optimal routing, i.e., each arriving customer may decide independently which queue to join in order to minimize an individual performance measure, taken as his own sojourn time. However, because of the service discipline this measure depends on the decisions of future arrivals, not available to the current decision maker. This leads us to model the problem as a dynamic game, in which there are an infinite number of players which are the different arriving customers. The goal is to identify Nash equilibria for the game. We show that under some conditions "join the shortest queue" is an equilibrium, whereas under slightly different service conditions, "join the longest queue" is a Nash equilibrium.

Other research on applications of stochastic games in queueing systems can be found in Hsiao and Lazar (1986), Altman (1992), Altman (1993), Altman and Kool (1993), Altman and Shimkin (1992), Shimkin and Shwartz (1993).

1 Socially optimal routing

We assume a Poisson arrival process with rate λ. There are N parallel queues, each queue $1 \le i \le N$ having a finite buffer of length $q_i < \infty$. Denote $q = (q_1, \ldots, q_N)$. At each arrival the router takes a decision $a \in \{1, ..., N\}$ to which queue the arrival should be routed to. If all queues are full the customer is lost.

The server in each queue is assumed to be exponential. The service rate of each server is known to be at least μ (for some $\mu \geq 0$). Moreover, it is known that the sum of the service rates is $\sum_{i=1}^{N} \mu_i = N\mu + \nu_o$, with ν_o a positive constant. Subject to these constraints, the μ_i's may change in time in a way unpredictable by the router.

We shall consider the worst-case control problem, where the objective is to design a routing strategy that guarantees the best performance under the worst service conditions. This leads to formulate the problem as a zero-sum game, where service rates are controlled (by a fictitious service controller) so as to maximize the router's costs. One may assume that each queue has a basic service rate μ, and an extra service rate of b_i is allocated to queue i such that $\sum_{i=1}^{N} b_i = \nu_o$. Let $\mathbf{B}$ denote the compact convex set of possible service allocations, which is the set of actions available to the service controller.

Let the state of the system be the number of customers in the different queues (including in service) just prior to a transition. Denote the state space by $\mathbf{X} \stackrel{\triangle}{=} \prod_{i=1}^{N}\{0, 1, \ldots, q_i\}$. Let $\mathbf{A}_x$ be the set of feasible actions for the routing controller at state $x \in \mathbf{X}$, subject to the buffer-size constraints. Thus $\mathbf{A}_x = \{1 \leq i \leq N : x_i < q_i\}$ if $x \neq q$. For $x = q$ we formally define $\mathbf{A}_q = \{0\}$, which corresponds to rejecting a customer.

Let e_i, $i = 1, 2, ..., N$ be the vector whose entries are all zero except for the ith entry, which is one.

It will be useful to assume that the service rate allocation may change only when the system state (potentially) changes. Since the system is memoryless this should not affect the optimal strategies. It is then possible to formulate the problem as an equivalent discrete-time problem, by observing the system only at certain "transition" times. We employ a standard uniformization procedure (e.g. Lippman (1975)), whereby the times between transitions are exponential with a fixed parameter. Thus, the system is observed at time instants when either arrival or a (possibly fictitious, if the queue is empty) service completion occurs, which form a Poisson process with rate $\alpha = \lambda + N\mu + \nu_o$. The probability that a transition corresponds to an arrival, an uncontrolled end of service in queue j, or an end of a controlled service in queue j are λ/α, μ/α and b_j/α respectively. The transition probabilities are thus

$$P_{y|xab} = \begin{cases} \lambda/\alpha & \text{if } \ y = x + e_a \leq q \ \text{ or } \ x = y = q\,, \\ (\mu + b_i)/\alpha & \text{if } \ y = (x - e_i)^+\,. \end{cases}$$

A strategy $u \in U$ for the router is a sequence $u = \{u_0, u_1, ...\}$ where u_n is a probability over $\mathbf{A}_x$, conditioned on the history of all previous states and actions of both players, as well as the current state $X_n = x$. Similarly, a strategy $v \in V$ for the server is a sequence $v = \{v_0, v_1, ...\}$ where v_n is a probability over $\mathbf{B}$, conditioned on the history of all previous actions of both players and states, as well as the current state.

Each pair u, v and initial state x induce a probability measure $P_x^{u,v}$ on the sequence of random variables X_n, A_n and B_n (the state of the system and the actions of both players at the nth transition).

We introduce a holding cost $c : \mathbf{X} \to \mathbb{R}_+$. Given a discount factor $\xi \in (0,1)$, we introduce the discounted cost

$$V(x, u, v) = \sum_{n=0}^{\infty} \xi^n E_x^{u,v} c(X_n).$$

We note that this cost equals the continuous-time discounted cost $V = \alpha \int_0^\infty e^{-\eta t} Ec(X_t)dt$, with $\xi = \alpha/(\alpha+\eta)$. The objective of the router is

P1: find a strategy u that achieves $V^*(x) = \inf_{u \in U} \sup_{v \in V} V(x, u, v)$

Let S be the operator defined by

$$S(x, a, b, f) = c(x) + \xi \sum_y P_{y|xab} f(y) \tag{1}$$

By standard results on stochastic games with compact action spaces (e.g. Maitra (1971), Kumar and Shia (1981)) we know that $V^*(x)$ is the unique solution (fixed point) of the equation

$$V(x) = \min_{p \in M(\mathbf{A}_x)} \max_{\beta \in M(\mathbf{B})} \sum_{j \in \mathbf{A}_x} p_j \int_{b \in \mathbf{B}} d\beta S(x, j, b, V), \quad x \in \mathbf{X}, \tag{2}$$

where $M(\mathbf{A}_x)$ and $M(\mathbf{B})$ are the sets of probability measures over $\mathbf{A}_x$ and $\mathbf{B}$ respectively. Moreover, let u^* be the stationary strategy such that the randomized action $u^*(x)$ at each state x is chosen as a minimizer in (2) (for $V = V^*$). Let v^* denote the stationary strategy such that $v^*(x)$ is a maximizer in the counterpart of (2) with min and max interchanged. Then (u^*, v^*) are optimal (saddle-point) strategies, namely

$$V(x, u, v^*) \geq V(x, u^*, v^*) \geq V(x, u^*, v), \qquad \forall u \in U, v \in V, x \in \mathbf{X} \tag{3}$$

(Theorems 2 and 3 in Kumar and Shia (1981)), hence u^* solves problem **P1**. v^* is identified here as the "worst case service conditions".

We "guess" below a pair of stationary strategies u^o and v^o and show that $V(x, u^o, v^o)$ solves (2) and hence equals $V^*(x)$. Moreover, we show that u^o is minimizer in (2), hence it solves problem **P1**.

For any $y \in \mathbf{X}$ and $j = 1, 2, ..., N$, introduce the following operators:

$$A_j y = y + e_j,$$

(restricted to $j \in \mathbf{A}_y$), which corresponds to joining the jth queue;

$$s_{\mathrm{nf}}(y) = \{k \in \mathbf{A}_y : y_k = \min_{i \in \mathbf{A}_y} y_i\}$$

which is the set of shortest non-full queues; and

$$A^s y = A_j y, \ j = \min\{k \in s_{\mathrm{nf}}(y)\}$$

which corresponds to joining the shortest non-full queue (if there are several shortest queues, then the one with the smallest index is chosen). If $y = q$ we set $A_0 q = q$ (which corresponds to rejecting a customer) and $A^s q = q$. Also define

$$D_j y = \max(y - e_j, 0) \,,$$

(where maximization is componentwise), which corresponds to end of service in the jth queue;

$$s(y) = \{1 \le k \le N : y_k = \min_i y_i\}$$

which is the set of shortest queues (full or not); and

$$D^s y = D_j y, \ j = \min\{k \in s(y)\}$$

which corresponds to end of a service in the shortest queue.

We introduce the following properties of a function $f : \mathbf{X} \to \mathbb{R}$.

C1 (monotonicity) $f(z) \le f(y)$ for $z \le y$,

C2 (symmetry) $f(y) = f(\pi y)$ for any permutation πy of y such that $\pi y \in \mathbf{X}$.

C3 $f(A_k y) \le f(A_l y)$ for y such that $y_k \le y_l$, $y + e_k + e_l \le q$.

Theorem 1.1 *Assume that the cost function c satisfies* **C1,C2,C3**. *Consider a routing strategy u^o that always joins a (non-full) shortest queue. Consider a server's strategy v^o that always serves a shortest queue. Then (u^o, v^o) satisfy (3) and thus u^o solves* **P1**.

Remark 1.1 In stochastic games the optimal strategies generally require randomization. The reason that no randomization is required here is that the game has a special structure, namely it is a game with perfect information. In a stochastic game this means that the players do not move simultaneously.

In order to prove the Theorem, we need the following lemmas.

Lemma 1.1 *Assume that c satisfies* **C1,C2,C3**. *Then $V(\cdot,u^o,v^o)$ satisfies* **C1,C2,C3**.

Proof. Introduce the operator T:

$$Tf(x) = c(x) + \xi\alpha^{-1}\left[\lambda f(A^s x) + \mu \sum_{j=1}^{N} f(D_j x) + \nu_o f(D^s x)\right] . \tag{4}$$

Then $V(\cdot,u^o,v^o)$ is the unique fixed point of $f = Tf$, and

$$\lim_{n\to\infty} T^n c = V(\cdot,u^o,v^o) \tag{5}$$

(e.g. Ross (1970), Lemma 6.2). We show by induction that $T^n c$ satisfies **C1,C2,C3**. Suppose that $f : \mathbf{X} \to \mathbb{R}$ satisfies them. We proceed to show that Tf satisfies them. Considering **C1**, let $y \le z$. Then a term-wise comparison in (4) will establish that $Tf(z) \le Tf(y)$. Note first that f is monotone by **C1**. It is readily verified that $y \le z$ implies $A^s y \le A^s z$ and $D_j y \le D_j z$. Also, there exists a permutation π such that $D^s y \le \pi D^s z$, so that **C1,C2** imply $f(D^s y) \le f(D^s z)$. Using these facts and assumption **C1** in (4) establishes that Tf satisfies **C1**.

To establish **C2** we note that for each permutation π and state y such that $y, \pi y \in X$, the states $A^s \pi y$, $D^s \pi y$ and $D_j \pi y$ are permutations of $A^s y$, $D^s y$ and $D_j y$ (respectively). By assumption **C2** on f and (4) it follows that Tf satisfies **C2**.

Next we establish **C3**. Consider y with $y_k < q_k$, $y_l < q_l$. If $y_k = y_l$ then $A_k y$ is a permutation of $A_l y$, so that $Tf(A_k y) = Tf(A_l y)$ follows form **C2**. Suppose then that $y_k < y_l$.

It is first shown that

$$f(A^s A_k y) \le f(A^s A_l y) . \tag{6}$$

Let i, j be the (shortest non-full) queue indexes which satisfy $A^s A_k y = A_i A_k y$ and $A^s A_l y = A_j A_l y$. If $i = j$, then noting $(A_i y)_k \le (A_i y)_l$ we obtain by **C2,C3**:

$$f(A^s A_k y) = f(A_k(A_i y)) \le f(A_l(A_i y)) = f(A^s A_l y) .$$

If $i \ne j$, it follows that $j = k$ (since arrival to queue k changed the shortest non-full queue; obviously an arrival to the longer queue l cannot have this effect). Thus $A^s A_k y = A_i(A_k y)$, and $A^s A_l y = A_k A_l y = A_l(A_k y)$. Noting that $(A_k y)_i \le (A_k y)_l$ (since $i \in s_{\text{nf}}(A_k y)$ while $(A_k y)_l = y_l < q_l$), then (6) again follows by **C3**.

Similarly,

$$f(D_jA_ky) \le f(A_kD_jy) \le f(A_lD_jy) = f(D_jA_ly)\,. \tag{7}$$

Here the first inequality follows form **C1** since $D_jA_ky \le A_kD_jy$ (in fact equality holds unless $j = k$ and $y_k = 0$). The second follows from **C3** since $(D_jy)_k \le (D_jy)_l$, and the last since $A_lD_jy = D_jA_ly$ (which holds even if $j = l$ since $y_l > 0$ by assumption).

Finally, we have

$$f(D^sA_ky) \le f(D^sA_ly)\,. \tag{8}$$

To understand this inequality, note that $l \notin s(y)$ (the set of shortest queues), hence $A_lD^sy = D^sA_ly$. Now, if $k \notin s(y)$ then $A_kD^sy = D^sA_ky$. Otherwise, if k is the only shortest queue in state y, then $D^sA_ky \le \pi(A_kD^sy)$ for some permutation π. Noting that $(D^sy)_k \le (D^sy)_l$, we obtain in these two cases

$$f(D^sA_ky) \le f(A_kD^sy) \le f(A_lD^sy) = f(D^sA_ly)\,.$$

The remaining case is when k is not the only member of $s(y)$. Then for some $i \in s(y)$ we have $D^sA_ly = D_iA_ly = A_lD_iy$, and for some $j \in s(y)$, $j \ne k$: $D^sA_ky = D_jA_ky = A_kD_jy$. Since A_lD_iy is a permutation of A_lD_jy, we obtain

$$f(D^sA_ky) = f(A_kD_jy) \le f(A_lD_jy) = f(A_kD^sy) = f(A_lD_iy) = f(D^sA_ly)\,,$$

where the inequality is a consequence of **C3**. Thus (8) is established, and combined with (6),(7) it implies that Tf satisfies **C3**.

We conclude that if f satisfies **C1,C2,C3** then so does Tf. Thus T^nc satisfies **C1,C2,C3** for $n = 1, 2, \ldots$, and by (5) so does $V(\cdot, u^o, v^o)$. ∎

Lemma 1.2 **C1,C2,C3** *imply the following counterpart of* **C3***:*
C4 $\quad f(D_ky) \ge f(D_ly)$ *for* y *such that* $y_k \le y_l$*.*

Proof. Consider $y \in \mathbf{X}$ with $y_k \le y_l$. If $y_k = 0$ then $D_ky = y \ge D_ly$, and **C4** follows from **C1**. If $y_k > 0$, define $y' = D_kD_ly$. Then $y'_k \le y'_l$, and by **C3**:

$$f(D_ky) = f(A_ly') \ge f(A_ky') = f(D_ly)\,.$$

∎

Proof of Theorem 1.1: Equation (2) is equivalent to

$$\begin{aligned} V(x) &= \min_{p\in M(\mathbf{A}_x)} \max_{\beta\in M(\mathbf{B})} \sum_{j\in\mathbf{A}_x} p_j \int_{b\in\mathbf{B}} d\beta \left\{ c(x) + \xi\alpha^{-1} \left[\lambda V(A_j x) + \sum_{j=1}^{N} (\mu + b_j) V(D_j x) \right] \right\} \\ &= c(x) + \xi\alpha^{-1} \left[\lambda \min_{p\in M(\mathbf{A}_x)} \sum_{j\in\mathbf{A}_x} p_j V(A_j x) + \mu \sum_{j=1}^{N} V(D_j x) \right. \\ &\quad \left. + \max_{\beta\in M(\mathbf{B})} \int_{b\in\mathbf{B}} d\beta \sum_{j=1}^{N} b_j V(D_j x) \right] . \qquad (9) \end{aligned}$$

We show that $V_o(x) \stackrel{\triangle}{=} V(x, u^o, v^o)$ satisfies this equation. By Lemma 1.1, $V_o(x)$ satisfies **C3**, hence $\min_j V_o(A_j x) = V_o(A^s x)$. By Lemma 1.2 $V_o(x)$ also satisfies **C4**, hence $\max_j V_o(D_j x) = V_o(D^s x)$. Recalling the definition of T in (4), for $V = V_o$ equation (9) thus reduces to $V_o = TV_o$, which is satisfied. It thus follows $V_o = V^*$ (the value of the game). Since the strategies u^o and v^o correspond to minimizers and maximizers (respectively) in (9), they are saddle-point strategies.

∎

We conclude this section by mentioning some straightforward but important consequences of the above results:

Average cost: An alternative cost functional which is often considered to account for long-term performance is the average cost:

$$V_{av}(x, u, v) = \limsup_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} E_x^{u,v} c(X_k) .$$

Noting that the strategies (u^o, v^o) are optimal in the discounted-cost game for *every* discount factor, it is easy to show that they are also optimal in the average-cost game (e.g. Ligget and Lippman (1969)). Thus, Theorem 1.1 is valid for the average cost well.

Throughput maximization: Since the buffers are finite, customers are rejected when all buffers are full. It is reasonable to include in the cost functional a term which penalizes for such rejections (equivalently, encourages high throughput). In the present case this can be done simply by adding to the (holding) cost function $c(x)$ a term proportional to $\mathbf{1}\{x = q\}$. This corresponds to a penalty of α/λ units for each rejected customer, since rejection occurs with probability λ/α if (and only if) $X_n = q$. It is easily verified that this cost satisfies the required properties **C1,C2,C3**.

Discrete service allocations: It was assumed for simplicity of presentation that all service-rate allocations in the set $B = \{(b_1, \ldots, b_N) : \sum b_i = \nu_o\}$ are possible. If the free service is carried by servers in fixed quanta, the set of possible allocations may turn to be a (finite) subset of B. It is easily verified that the above results remain valid for any subset which includes the vertices of B (so that the 'serve the shortest queue' strategy remains feasible).

2 Individually optimal routing

We consider N queues with infinite buffers, attended by a single server. Let $\{T_i,\ i = 0, 1, 2, \ldots\}$ be an arbitrary increasing sequence such that T_i denotes the arrival time of the ith customer (the distribution of the arrival times is not important here). The service times τ_i of the different customers are i.i.d. and exponentially distributed. Upon arrival, a customer has to decide which queue to join, based only on the current state (the queue lengths). The goal is to minimize his own expected waiting time, i.e. the time till the beginning of service. As opposed to the previous case, we assume here that the server is socially conscious, and always serves a longest queue (SLQ policy). Within each queue customers are served in order of arrival.

Clearly, if the service rates at each queue were fixed, the obvious choice for each arriving customer is to join the shortest queue (in terms of expected service time of existing customers). However, under the present service discipline he might consider joining a longer queue, to which the server may assign more attention. Still he faces an unknown factor in terms of the decisions of future arrivals, which will affect the queue lengths hence the division of service. This leads to consider the problem as a non-cooperative game between the customers, with each customer considered a distinct player.

Let the state of the system be the number of customers in the different queues (including in service). We assume that the state changes just *after* a transition (namely, it is left-continuous). We also assume that arrivals and service completions do not occur simultaneously (w.p. 1). The actions available to each customer are $\mathbf{A} = \{1, ..., N\}$, i.e. which queue to join. We define a multi-strategy u to be a sequence of decision rules $u = \{u_0, u_1, ...\}$; u_i corresponds to the decision rule for the ith customer, and is a probability measure over $\mathbf{A}$ conditioned on the state at T_i. Our goal is to identify Nash equilibria, i.e. a multi-strategy u^* such that no player can benefit by unilaterally deviating from his decision rule. We define the multi-strategies JSQ and JLQ, of joining the shortest (resp. longest) queue; if there are several such queues then one of them is chosen at random.

Finally, we need to specify the choice of the server in case several queues are longest. We first consider the following rule:

The random SLQ policy: The queue to be served is chosen at random (form the set of longest queues) with equal probability.

The following Theorem identifies a Nash equilibrium.

Theorem 2.1 *Assume random SLQ service policy. Then the multi-strategy JSQ is a Nash equilibrium.*

Proof. Consider customer C_o which arrives at time $t = 0$, in initial state y. All subsequent customers are using JSQ. We shall perform sample-path comparison of the waiting time of C_o in the following two copies of our system, which will be operated side by side under appropriate coupling:

X–system: C_o joins a shortest queue (say Q_1). We denote the corresponding state by $X(t)$.

$\bar{X}$–system: C_o joins a queue (say Q_2) which is not a shortest queue. We denote the corresponding state by $\bar{X}(t)$.

Thus $X(0) = \bar{X}(0) = y$, with $y_1 = \min_i y_i$ and $y_2 > y_1$; and $X(0+) = X(0) + e_1$, $\bar{X}(0+) = \bar{X}(0)+e_2$. It will be shown that the waiting time of C_o in the X–system is greater than or equals his waiting time in the coupled $\bar{X}$–system for *each* sample path. We use the notation $m(x) = \min_i x_i$ and $M(x) = \max_i x_i$ for the minimal and maximal queue lengths, $s(x) = \{k : x_k = m(x)\}$ for the set of shortest queues, $\ell(x) = \{k : x_k = M(x)\}$ for the set of longest queues, and $|A|$ for the cardinality of a finite set A.

The following coupling will be used. The arrival times and service completion times are assumed the same in both systems (note that under the SLQ service policy this implies that the total number of customers is always the same in the two systems). We shall also require that the random choices of the longest queue at each service instant are coupled to satisfy the following. If Q_1 in the X–system has a higher (or equal) probability of being served than Q_2 in the $\bar{X}$–system, then the first will get served whenever the latter is. That is, if $1 \in \ell(X(t))$, $2 \in \ell(\bar{X}(t))$, and $|\ell(X(t))| \leq |\ell(\bar{X}(t))|$, then $\bar{X}(t+) = \bar{X}(t) - e_2$ implies $X(t+) = X(t) - e_1$. No further restrictions are required. We note that this coupling of random decisions will only be used in claim (c) below. (It is however crucial for the proof; under the persistent SLQ to be defined below, such coupling is impossible, and consequently, as will be shown, the result of the present Theorem does not hold.)

The basic observation which facilitates the proof is the following:

(a) For each $t \geq 0$, $m(X(t)) \geq m(\bar{X}(t))$ and $M(X(t)) \leq M(\bar{X}(t))$. Moreover, if $m(X(t)) = m(\bar{X}(t))$ then $|s(X(t))| \leq |s(\bar{X}(t))|$, and if $M(X(t)) = M(\bar{X}(t))$ then $|\ell(X(t))| \leq |\ell(\bar{X}(t))|$.

To see that, consider starting the system at (any) initial state y, and assume we are given any finite sequence of arrival and departure times. Let X_f be the final state. It may be seen that the minimal (over all pairs of joining and service policies, causal or not) $M(X_f)$ and maximal $m(X_f)$ is achieved by the pair JSQ/SLQ. Since the X–system follows JSQ/SLQ, (a) is a simple consequence. We now claim:

(b) Whenever $X_1(t) < \bar{X}_2(t)$, no arrivals occur to Q_2 in the $\bar{X}$–system at time t (i.e. $\bar{X}_2(t+) = \bar{X}_2(t)$).

Indeed, by (a), $X_1(t) < \bar{X}_2(t)$ implies that Q_2 is not shortest in $\bar{X}(t)$, and therefore will not be joined under the JSQ policy. Also,

(c) Whenever $X_1(t) \geq \bar{X}_2(t)$, the number of service completions in Q_1 in the X–system at time t is at least as the number of service completions in Q_2 in the $\bar{X}$–system at time t (i.e. $\bar{X}(t+) = \bar{X}(t) - e_2$ implies $X(t+) = X(t) - e_1$).

This is again a consequence of (a). If $X_1(t) > \bar{X}_2(t)$ then Q_2 is not longest in $\bar{X}(t)$ and is therefore not served. If $X_1(t) = \bar{X}_2(t)$ and Q_2 is longest, it follows that Q_1 is longest in $X(t)$ and $|\ell(X(t))| \leq |\ell(\bar{X}(t))|$. The conclusion then follows by the above coupling scheme.

Using (b) and (c) it follows that as long as C_o is in the X–system, his position there is at least as good as in the $\bar{X}$–system (in the sense that the number of customers standing before him in his queue is smaller). Indeed, this is the case at $t = 0+$ since he joined a shortest queue; and (b),(c) guarantee that neither arrivals nor departures can change that. This clearly implies that he leaves the X–system no later than he leaves the $\bar{X}$–system, which completes the proof. ∎

Remarks:

1. The proof implies that, given that all subsequent customers use JSQ, joining the shortest queue minimizes the waiting time of a customer not only in the sense of expected value, but in the stronger sense of stochastic dominance.

2. It may be seen that there exist sample paths (of positive probability) such that the waiting time of the customer if he joins a shortest queue (i.e., in the X–system) is strictly shorter than if he decides otherwise. Thus, if he deviates unilaterally from JSQ he strictly looses.

Despite this sharp result, we cannot yet consider JSQ as *the* solution of our problem. First, we did not establish uniqueness. However, it may be argued that JSQ (along possibly with JLQ) are the most "plausible" candidates for possible behavior; since the JLQ in not necessarily a Nash equilibrium under the above assumptions (as we now show) the JSQ is singled out.

Proposition 2.1 *Assume random SLQ service policy. Then the multi-strategy JLQ is not always a Nash equilibrium (depending on the arrival process).*

Proof. Suppose that some arriving customer C_i finds no customers in queues $1, ..., N-1$ and one in queue N (which is being served). Suppose that the arrival rate is much lower than the service rate; thus, with very high probability the system will empty before the next arrival. In that case, if C_i follows JLQ (by joining queue N), then he will get served after the present customer. If he deviates by joining an empty queue, then he has a probability of 0.5 to be served first and probability of 0.5 of being served second. Since we assumed that the probability of arrivals before his service completion is negligible, he is clearly better off in the second case. ∎

A second important point to observe is that the results above are are highly sensitive to the rule used by the server to choose between longest queues. Consider the following alternative scheme:

The persistent SLQ policy: The queue that has just been served is the one to be served again if it is still the longest queue, otherwise the queue to be served is chosen at random from the set of longest queues with equal probability.

We then have the following results, which in a sense are the converse of the previous ones:

Theorem 2.2 *Assume that the persistent SLQ is used by the server.*

(i) The multi-strategy JSQ is not always a Nash equilibrium (depending on the arrival process).

(ii) Assume that the system starts initially empty or with one non-empty queue. Then the multi-strategy JLQ is a Nash equilibrium.

Proof. (i) Consider an arriving customer C_i, and assume that JSQ is used by all subsequent customers. Suppose that C_i finds no customers in queues $1, ..., N-1$ and one in queue N (which is being served). Suppose further that the arrival rate is much lower than the service rate, so that the probability of more than one arrival before the system empties is negligible. If no arrivals occur before two service completions, then C_i will be served second (i.e. leave at the second service completion), no matter which queue he joins. Suppose next that one customer (C_{i+1}) does arrive before two service completions, and joins a shortest (empty) queue. If he arrives after the first service completion then C_i will be served second, no matter which queue he joined. However, if C_{i+1} arrives before the first service completion, then if C_i had joined a shortest (empty) queue he will be served second or third with equal probabilities, while if he had joined queue N he will be served second. Thus C_i benefits by deviating from the JSQ policy. Since the latter scenario has a strictly positive probability to happen, the JSQ is not a Nash equilibrium under the persistent SLQ.

(ii) Assume that all customers follow JLQ, and the system starts initially with at most one non-empty queue. Since all customers use the JLQ multi-strategy, they will line up in the same queue, and the service is simply given according to a global FCFS order (First Come First Serve). Suppose that some arbitrary customer, say C_i, deviates from JLQ and joins a shorter (i.e. an empty) queue. Then it can be seen that he has to wait at least till all those that arrived before him are served (this follows from the fact that service is persistent). Moreover, if during that time other arrivals occurred then they will be served before C_i. Hence JLQ is a Nash equilibrium under the persistent JLQ. ∎

Thus, under the condition stated in (ii), a stable behavior for the customers is to line up in the same queue while leaving all others empty. We note however that JLQ is not necessarily a Nash equilibrium if more queues are initially (or somehow become) non-empty; this is demonstrated by the following example.

Assume that the persistent SLQ is used by the server. Suppose some arriving customer C_i finds 4 non-empty queues, with one customer in queue 1 and two in each of the others. Suppose that arrivals are very rare, so that none occur before the system empties. If C_i joins one of the longest queues, it is easily verified that he will be the 6th, 7th or 8th to get served, with probability 1/3 each. But if he joins queue 1 he will be the 5th w.p. 1/3, and the 6th 7th or 8th w.p. 2/9 each. Thus C_i benefits by deviating from JLQ, and therefore this multi-strategy is not Nash. (We note that this example, as well as the previous results, are valid both under preemptive and non-preemptive service. For preemptive service it is enough to consider a similar example with only three queues.)

In conclusion, under slightly different assumptions on the service policy two opposite arrival multi-strategies (namely JSQ and JLQ) were seen to form equilibria. If the service policy is not precisely defined or not known by each customer, then the resulting behavior becomes harder to predict, and in fact may depend on other factors not captured by the present model.

References

[1] E. Altman. 1992, "Flow control using the theory of zero-sum Markov games," INRIA Report No. 1597; to appear in *IEEE Trans. Automat. Contr.*, February 1994.

[2] E. Altman. 1993. "Monotonicity of optimal policies in a zero sum game: a flow control model", to appear in *Annals of dynamic games*, Vol. 1.

[3] E. Altman and G. Koole. 1993. "Stochastic Scheduling Games with Markov Decision Arrival Processes", *Journal Computers and Mathematics with Appl.*, third special issue on "Differential Games".

[4] E. Altman and N. Shimkin. 1992. "Individually optimal dynamic routing in a processor sharing system: stochastic game analysis," Technical Report, No. EE-849, Technion – Israel Institute of Technology, August 1992; submitted to *Opns. Res.*

[5] M. T. Hsiao and A. A. Lazar. 1986. "Optimal Decentralized Flow Control of Markovian queueing Networks with Multiple Controllers", *CTR Technical Report*, CUCTR-TR-19, Columbia University.

[6] P. R. Kumar and T. H. Shiau. 1981. "Existence of value and randomized strategies in zero-sum discrete-time stochastic dynamic games", *SIAM J. Contr. and Opt.* **19** No. 5, pp. 617-634.

[7] T. M. Ligget and S. A. Lippman. 1969. "Stochastic games with perfect information and time average payoff," *SIAM Review* **11**, pp. 604–607.

[8] S. A. Lippman. 1975. "Applying a new device in the optimization of exponential queueing systems," *Opns. Res.* **23**, pp. 687–710.

[9] A. Maitra and T. Parthasarathy. 1971. "On stochastic games," *Journal of Optimization Theory and Applications,* pp. 154–160.

[10] S. M. Ross. 1970. *Applied Probability Models with Optimization Applications*, Holden-Day, San Francisco.

[11] N. Shimkin and A. Shwartz. 1993. "Guaranteed performance regions in Markovian systems with competing decision makers," *IEEE Trans. Automat. Contr.* **90**, pp. 84–95.

[12] R. Weber. 1978. "On the optimal assignment of customers to parallel servers," *J. Appl. Prob.* **15**, pp. 406–413.

[13] W. Winston. 1977. "Optimality of the shortest-processing-time discipline," *J. Appl. Prob.* **14**, pp. 181–189.

Appendix

This appendix is intended for the reviewers' attention only. It explains some statements in the proof of Lemma 1.1 which are quite straightforward, but, mainly due to non-equal buffer sizes, may not be completely obvious at first sight.

a. "$y \leq z$ implies $A^s y \leq A^s z$":
Let $A^s y = A_i y$. If $y_i < z_i$ then $A^s y \leq z \leq A^s z$. If $y_i = z_i$, then $z \geq y$ implies $A^s z = A_i z$, and the required inequality again follows.

b."[$y \leq z$ implies] there exist π such that $D^s y \leq \pi D^s z$":
Let $A^s y = A_i y$, $A^s z = A_j y$. If $i = j$ then obviously $D^s y \leq D^s z$. If $i \neq j$, let π be the permutation which exchanges i and j. Thus
$(D^s y)_i = (D_i y)_i \leq (D_j z)_j = (D^s z)_j = (\pi D^s z)_i$,
$(D^s y)_j = y_j \leq z_j \leq z_i \leq (D_j z)_i = (\pi D^s z)_j$.
Since all other components are unaffected, these imply $D^s y \leq \pi D^s z$.

c. "For each π and y such that y, $\pi y \in \mathbf{X}$, $A^s \pi y$ is a permutation of $A^s y$":
If $y = q$ then $\pi y = q$ (since otherwise $\pi y \notin \mathbf{X}$). Assume then $y \neq q$, which implies $\pi y \neq q$. Then $A^s y = A_i y$ and $A^s \pi y = A_j \pi y$ for appropriate i and j. It is sufficient to show that $y_i = (\pi y)_j$. Suppose first, in contradiction, that $y_i > (\pi y)_j$. This can happen onli in the following case:
(*) $\exists l$ s.t. $q_j > (\pi y)_j = y_l = q_l$ (i.e. queue l was shorter than i but full, and its content Y_l was transferred by π to a queue with a larger buffer).
However this cannot happen if $\pi y \in \mathbf{X}$. Indeed, by its definition $i \in s_{\mathrm{nf}}(y)$, which implies that *all* queues shorter than y_i are full (i.e. $y_l = q_l$ for $l \in L(y_i) := \{l : y_l < y_i\}$). Thus, since πy must satisfy the buffer constraints, π cannot change the contents of these queues Thus, since πy must satisfy the buffer constraints, π cannot change the contents of these queues (i.e. $(\pi y)_l = y_l = q_l$ for $l \in L(y_i)$). The second possibility $y_i < (\pi y)_j$ may be refuted symmetrically (by considering πy as a permutation of y).