MATHEMATICS OF OPERATIONS RESEARCH
Vol. 17, No. 1, February 1992
Printed in U.S.A.

MONOTONICITY IN GENERALIZED SEMI-MARKOV PROCESSES*

PAUL GLASSERMAN AND DAVID D. YAO

We establish stochastic monotonicity of the event epoch sequences of generalized semi-Markov processes through the structure of the generalized semi-Markov *schemes* on which they are based. Our main condition states, roughly, that the occurrence of *more* events in the short run never leads to the activation of *less* events in the long run. We consider monotonicity with respect to two types of inputs: clock times (which translate to, e.g., service and interarrival times in queueing systems), and structural parameters (which translate to, e.g., buffer size, number of servers, and job population). For the second type of comparison, we replace a structural change with an equivalent change in clock times to reduce the comparison to one of the first type. When applied to queueing systems, our results yield new comparisons and also unify several existing results previously established using special properties of individual systems.

1. Introduction. Optimal design of a stochastic system is often a difficult problem. Stochastic ordering results make it possible to compare systems and determine which performs "better" without evaluating their performance quantitatively. Hence, they can provide a qualitative basis for design improvement. This paper contributes to a growing literature on stochastic ordering for queues and related systems by identifying structural properties, for a broad class of models, that lead immediately to the monotonicity of performance as a function of key inputs.

We take the point of view that a system is designed to perform certain useful tasks. The completion of such a task constitutes an "event". One system performs better than another, in a very strong sense, if it completes all tasks sooner when both systems are subjected to the same input—i.e., if the event sequences are stochastically ordered. These are the kinds of comparisons we make. In particular, we do not make comparisons between the *states* of different systems; indeed, we do not even assume that our state spaces are in any sense ordered. When applied to queueing systems, our results translate to comparisons between service completion sequences, throughputs, arrival processes, etc.; but not, however, to comparisons between queue lengths or waiting times. Stochastic orderings for these quantities hold less generally. On the other hand, results in Whitt [24] show how orderings between event epoch sequences sometimes imply orderings between queue lengths and waiting times, and our results are likely to have similar applications in specific cases.

Casting our results in the framework of *generalized semi-Markov processes* (GSMP) provides significant generality, and—more importantly—allows us to identify key *structural* properties from which monotonicity follows. When applied to queueing systems, our approach yields new comparisons and also unifies and extends many existing results previously established for specific examples. In particular, several of

*Received November 6, 1989.
AMS 1980 subject classification. Primary: 60K15, 90B22, 60K20.
IAOR 1973 subject classification. Main: Stochastic processes.
OR/MS Index 1978 subject classification. Primary: 571 Probability/Regenerative Processes. Secondary: 697 Queues/Networks.
Key words. Stochastic monotonicity, generalized semi-Markov processes, networks of queues, discrete event systems.

0364-765X/92/1701/0001/$01.25

the results in Sonderman [18, 19], Shanthikumar and Yao [16, 17], and Tsoucas and Walrand [22] fall within our framework. Our conditions point to common features among these examples.

Since we do not make comparisons based on states, many known results for systems which could be modeled as GSMPs do not follow from our results. For example, we do not generalize Sonderman's [20] comparisons for semi-Markov processes. Our results have only trivial implications for an ordinary semi-Markov process, which is far too special a case to be an interesting GSMP. Nor do we consider monotonicity in *time* as in Daley [2], Massey [11], or Whitt [25], though finding conditions—and appropriate orderings, as in [11], [24] and [25]—for spatial and temporal monotonicity of GSMPs would certainly be worthwhile.

We adopt the now common approach of establishing (strong) stochastic ordering by making sample path comparisons—that is, we compare two processes by constructing them on a common probability space and showing that one always dominates the other, in a suitable sense. (See Kamae, Krengel, and O'Brien [10], and Stoyan [21].) §2 gives a description of GSMPs, and outlines the natural construction on which our comparisons are based. Glasserman [3, 4] provides more detailed recursions for the evolution of a GSMP, and those would apply here directly. The construction via point processes of Helm and Schassberger [9] could also be tailored to our formulation. (See Burman [1], Glynn [7], Haas and Shedler [8], Schassberger [13, 14, 15], and Whitt [23] for other approaches in slightly different settings.)

We first present our main conditions and results for a restricted class of GSMPs—those with what we call *deterministic routing* and *unit speeds*. Starting out with less than full generality helps clarify the arguments. For these GSMPs, §3 establishes monotonicity with respect to *clock times* (e.g., service times, interarrival times), and §4 monotonicity with respect to structural parameters, such as buffer size, number of servers, and network population. A novel feature of our approach is that it translates structural changes into equivalent changes in clock times, thus reducing the second type of comparison to the first. This mechanism is based on the notion of *subscheme*. Under appropriate conditions, a reduction in buffer size, number of servers, or network population, for example, can be "simulated" through an increase in service and interarrival times to facilitate comparisons. §5 and §6 establish analogous conditions and results for more general processes—GSMPs with *probabilistic routing* and with nontrivial speeds. §7 shows how our conditions can sometimes be weakened if we ask for weaker conclusions, using a relation called *relevance*. §8 generalizes and recasts our main results from a different point of view, replacing subschemes with *extractions*. §9 and §10 consider simple variants of the GSMP notion of "event". We conclude in §11 with some remarks on further extensions and applications.

2. Generalized semi-Markov processes and schemes.

A generalized semi-Markov process (GSMP) evolves by moving from state to state through the occurrence of *events* at random time instants. Associated with each state is a set of active events, any of which potentially triggers the transition out of that state. To each active event there corresponds a (random) *clock* reading, which determines when that event is scheduled to occur next. Clocks for different events may run at different *speeds* in different states; when a clock runs out, the associated event occurs. The process moves to the next state according to transition probabilities that depend on the current state and on the event that triggers the transition. After the transition, the clocks are adjusted to reflect the set of active events in the new state. In particular, new clocks are set for any newly active events, and clocks from the old state continue to run if the associated events are still active in the new state.

We characterize a GSMP by $(\mathbf{S}, \mathbf{A}, \mathscr{E}, p, \mathscr{P}, r)$ where $\mathbf{S}$ is a (finite or) countable state space representing the possible "physical" configurations of a system; $\mathbf{A}$ is the set of events, which we take to be finite. We denote generic events by α and β, and also let $m = |\mathbf{A}|$ and write $\mathbf{A} = \{\alpha_1, \dots, \alpha_m\}$. The mapping $\mathscr{E}$: $\mathbf{S} \to 2^{\mathbf{A}}$ yields the set of active events in a state—for $s \in \mathbf{S}$, $\mathscr{E}(s)$ is called the *event list*, and is never empty. To avoid trivialities, we require that $\mathbf{A} = \bigcup_s \mathscr{E}(s)$. For each $s \in \mathbf{S}$ and $\alpha \in \mathscr{E}(s)$, $p(\cdot\,; s, \alpha)$ is a probability mass function on $\mathbf{S}$; with probability $p(s'; s, \alpha)$, the next state is s' when a transition out of s is triggered by α. $\mathscr{P}$ is a probability law that governs the sequence of new clocks for each event. We take as given a stochastic process $X = \{(X_{\alpha_1}(n), \dots, X_{\alpha_m}(n)),\ n = 1, 2, \dots\}$ with law $\mathscr{P}$; the *nth* time a clock is set for α, it is set to $X_\alpha(n)$. Under $\mathscr{P}$, every $X_\alpha(n)$ has support in $(0, \infty]$. We allow essentially arbitrary dependence among the elements of X, but do not allow them otherwise to depend on the evolution of the GSMP. An important special case is where the X_α's are independent processes and each $\{X_\alpha(n)\}$ is an i.i.d. sequence. Finally, $r = \{r_{s\alpha},\ s \in \mathbf{S},\ \alpha \in \mathbf{A}\}$ is the set of *clock speeds*: the clock for α runs at the (finite) rate $r_{s\alpha}$ in state s, if $\alpha \in \mathscr{E}(s)$. Without $\mathscr{P}$, $\mathscr{S} = (\mathbf{S}, \mathbf{A}, \mathscr{E}, p, r)$ is called a generalized semi-Markov *scheme* (GSMS).

A simple algorithm maps a realization of X to a sample path of a GSMP. Let τ_n denote the epoch of the *nth* transition; s_n the state entered at the *nth* transition; and $c_n = (c_n(\alpha_1), \dots, c_n(\alpha_m))$ the vector of clock readings just after the *nth* transition. Initially, s_0 is fixed, $\tau_0 = 0$, and the clock readings are initialized by letting $c_0(\alpha)$ be $X_\alpha(1)$ if $\alpha \in \mathscr{E}(s_0)$ and zero otherwise. In words, clocks are *set* for events in $\mathscr{E}(s_0)$. Suppose (τ_n, s_n, c_n) have been determined. Then

$$\tau_{n+1} = \tau_n + \min\{c_n(\alpha)/r_{s_n\alpha} \colon \alpha \in \mathscr{E}(s_n)\},$$

taking division by zero to yield infinity. The event α^*_{n+1} that achieves the minimum on the right triggers the $n + 1st$ transition and is said to *occur* at τ_{n+1}. (If more than one α achieves the minimum, break ties by, say, taking the one with the smallest index in $\{\alpha_1, \dots, \alpha_m\}$.) The next state s_{n+1} is sampled from $p(\cdot\,; s_n, \alpha^*_{n+1})$. After the transition, clocks are updated as follows: Let n_α be the number of times a clock has previously been set for α. If $\alpha \in \mathscr{E}(s_{n+1}) \setminus (\mathscr{E}(s_n) - \{\alpha^*_{n+1}\})$, then $c_{n+1}(\alpha) = X_\alpha(n_\alpha + 1)$, and we say that the activation of α is *triggered* by α^*_{n+1}. If $\alpha \in \mathscr{E}(s_{n+1}) \cap (\mathscr{E}(s_n) - \{\alpha^*_{n+1}\})$ then $c_{n+1}(\alpha) = c_n(\alpha) - r_{s_n\alpha}(\tau_{n+1} - \tau_n)$; and if $\alpha \notin \mathscr{E}(s_{n+1})$ then $c_{n+1}(\alpha) = 0$. The state of the GSMP is now defined to be s_n at time t if $\tau_n \leqslant t < \tau_{n+1}$.

Our results focus on the sequence of *event epochs* $T = \{T_{\alpha_1}, \dots, T_{\alpha_m}\}$, where, for $i = 1, \dots, m$, $T_{\alpha_i} = \{T_{\alpha_i}(n),\ n = 1, 2, \dots\}$, and $T_{\alpha_i}(n)$ is the epoch of the *nth* occurrence of α_i. If α_i fails to occur n times, $T_{\alpha_i}(n) = \infty$. We restrict attention to $\mathscr{P}$'s under which the GSMP is nonexplosive—i.e., for all initial conditions,

$$(1) \qquad P\Bigl(\sup_{n>0} T_{\alpha_i}(n) = \infty\Bigr) = 1, \qquad i = 1, \dots, m.$$

This holds, for example, if under $\mathscr{P}$, almost surely

$$\sum_{n=1}^{\infty} X_{\alpha_i}(n) = \infty, \qquad i = 1, \dots, m.$$

For each event $\alpha \in \mathbf{A}$, the *throughput* of α is $\liminf_{n\to\infty} n/T_\alpha(n)$. Comparisons of T-sequences for different GSMPs translate immediately into comparisons of throughputs.

Our main conditions are most easily expressed in terms of *strings* of events, which are just finite sequences of elements of **A**. A string $\sigma = \beta_0 \cdots \beta_k$ is called *feasible* in s_0 if $\beta_0 \in \mathscr{E}(s_0)$, and if there is a sequence $s_1, \ldots, s_k$ of states for which $\beta_i \in \mathscr{E}(s_i)$, $i = 1, \ldots, k$ and

$$p(s_1; s_0, \beta_0) p(s_2; s_1, \beta_1) \cdots p(s_k; s_{k-1}, \beta_{k-1}) > 0.$$

For any string σ, let $N_\alpha(\sigma)$ be the number of occurrences of α in σ, and let $N(\sigma) = (N_{\alpha_1}(\sigma), \ldots, N_{\alpha_m}(\sigma))$.

The conditions we propose restrict us to GSMPs based on *noninterruptive* schemes (Schassberger [13]), meaning that they satisfy

$$\alpha \in \mathscr{E}(s), \quad p(s'; s, \alpha) > 0 \Rightarrow \mathscr{E}(s) - \{\alpha\} \subseteq \mathscr{E}(s'). \tag{2}$$

In a noninterruptive GSMP, at every transition all events active in the old state continue to be active in the new state, except possibly the event that triggers the transition.

Throughout this paper, all words of comparison are used in their weak sense; thus, "increasing" means "nondecreasing", "faster" means "no slower", "earlier" means "no later", etc.

3. Deterministic schemes. By a deterministic generalized semi-Markov scheme we mean one in which for all s and all $\alpha \in \mathscr{E}(s)$ there is just one s' for which $p(s'; s, \alpha) > 0$—necessarily, with $p(s'; s, \alpha) = 1$. We refer to this as *deterministic routing*. The alternative phrase "deterministic transitions" might incorrectly suggest that the sequence of state transitions is fixed; but the clock times introduce randomness in the order of events, hence also in the sequence of transitions. (There is often a close connection between the probabilities, p, and the routing probabilities in a queueing network; and a network with deterministic routing would typically be modeled by a deterministic scheme. But routing probabilities refer to transitions between queues, and p to transitions between states, so the two should not be confused.)

For a deterministic GSMS, if $\alpha \in \mathscr{E}(s)$ define $\phi(s, \alpha)$ by $p(\phi(s, \alpha); s, \alpha) = 1$. Extend $\phi(s, \cdot\,)$ to a function of strings by the following recursion: if the string $\sigma\alpha$ is feasible in s, let $\phi(s, \sigma\alpha) = \phi(\phi(s, \sigma), \alpha)$, and let $\phi(s, \cdot\,)$ applied to the empty string be s. This definition makes sense because $\sigma\alpha$ feasible implies σ feasible and $\alpha \in \mathscr{E}(\phi(s, \sigma))$.

Say that a scheme has *unit speeds* if, for all s and α, $r_{s\alpha} = \mathbf{1}\{\alpha \in \mathscr{E}(s)\}$. We now state several variants of our main condition for monotonicity in GSMPs with deterministic routing and unit speeds.

(M). *Monotonicity Condition*. If σ_1 and σ_2 are feasible in s, and $N(\sigma_1) \leqslant N(\sigma_2)$ (componentwise), then

$$\mathscr{E}(\phi(s, \sigma_1)) \setminus A_{\sigma_1 \sigma_2} \subseteq \mathscr{E}(\phi(s, \sigma_2)),$$

where $A_{\sigma_1 \sigma_2} = \{\alpha \colon N_\alpha(\sigma_1) < N_\alpha(\sigma_2)\}$.

Remark. Think of σ_1 and σ_2 as representing possible event sequences followed by a GSMP starting in state s. If $N(\sigma_1) \leqslant N(\sigma_2)$, then (M) implies that for any α such that $\sigma_1\alpha$ is feasible in s—i.e., $\alpha \in \mathscr{E}(\phi(s, \sigma_1))$—either $\alpha \in \mathscr{E}(\phi(s, \sigma_2))$ and $N(\sigma_1\alpha) \leqslant N(\sigma_2\alpha)$, or $N(\sigma_1\alpha) \leqslant N(\sigma_2)$. In this sense, under (M), if the ordering $N(\sigma_1) \leqslant N(\sigma_2)$ holds initially, it is preserved under the future evolution of the strings.

The following equivalent condition is sometimes easier to establish directly (e_i denotes the *ith* m-dimensional unit vector):

(M′). *Alternative Statement of* (M). If σ_1 and σ_2 are feasible in s, then

$$N(\sigma_1) = N(\sigma_2) \Rightarrow \mathscr{E}(\phi(s,\sigma_1)) = \mathscr{E}(\phi(s,\sigma_2)) \quad \text{and}$$

$$N(\sigma_1) + e_i = N(\sigma_2) \Rightarrow \mathscr{E}(\phi(s,\sigma_1)) \setminus \{\alpha_i\} \subseteq \mathscr{E}(\phi(s,\sigma_2)).$$

The first part of (M′) is a *permutability* condition: changing the order of events does not change the event list of the state reached. Given this condition, the second part of (M′) reduces to noninterruption.

An important special case of (M) is the following condition, proposed for a different purpose in Glasserman [3, 4]:

(C). *Commuting Condition. If* $\{\alpha, \beta\} \subseteq \mathscr{E}(s)$ *then* $\phi(s, \alpha\beta) = \phi(s, \beta\alpha)$.

Part of the requirement in (C) is that both sides be defined—i.e., that $\beta \in \mathscr{E}(\phi(s,\alpha))$ and $\alpha \in \mathscr{E}(\phi(s,\beta))$, which is the noninterruptive condition (2). An equivalent statement of (C) is given by

(C′). *Alternative Statement of* (C). The scheme is noninterruptive, and if σ is feasible in s, then $\phi(s,\sigma)$ depends on σ only through $N(\sigma)$; i.e., for feasible σ and σ', $N(\sigma) = N(\sigma') \Rightarrow \phi(s,\sigma) = \phi(s,\sigma')$.

Conditions (C) and (C′) require that the state reached through a sequence of events be independent of their order. The first part of condition (M′) weakens this to require only that the event list reached be independent of the order of events. Conditions (M) and (M′) ensure, roughly, that the occurrence of more events in the short run is never penalized in the long run. We summarize connections among these conditions in

PROPOSITION 3.1. (C) ⇔ (C′) ⇒ (M) ⇔ (M′) ⇒ (2).

To prove this we use the following lemma on permissible manipulations of strings, which will also be useful later:

LEMMA 3.2. (i) *In a scheme that satisfies* (M), *if the string* $\sigma\alpha\beta\tilde{\sigma}$ *is feasible in* s *and* $\beta \in \mathscr{E}(\phi(s,\sigma))$, *then* $\sigma\beta\alpha\tilde{\sigma}$ *is feasible in* s. (ii) *In a scheme that satisfies* (C), *let* σ *and* σ' *be feasible in* s. *Then if* σ' *is a permutation of* σ, *it can be obtained from* σ *through a sequence of transpositions of consecutive events, always preserving feasibility.*

PROOF OF LEMMA. Part (i) is an immediate consequence of (M). For part (ii), let $\sigma = \beta_1 \cdots \beta_k$, $\sigma' = \beta_{i_1} \cdots \beta_{i_k}$. Since σ' is feasible in s, $\beta_{i_1} \in \mathscr{E}(s)$. Find the first occurrence of the event β_{i_1} in σ; by part (i), we may repeatedly transpose β_{i_1} and the event that precedes it while maintaining feasibility. This permutes σ to something of the form $\beta_{i_1}\tilde{\sigma}_1$. Since $\beta_{i_2} \in \mathscr{E}(\phi(s,\beta_{i_1}))$ we may repeat this procedure with β_{i_2} and $\tilde{\sigma}_1$ to get something of the form $\beta_{i_1}\beta_{i_2}\tilde{\sigma}_2$. Repeating this for $\beta_{i_3}, \ldots, \beta_{i_{k-1}}$ we end up with $\beta_{i_1} \cdots \beta_{i_k}$ without ever violating feasibility. □

PROOF OF PROPOSITION. We show only (C) ⇒ (C′) and (C) ⇒ (M′); the other implications are obvious. Suppose (C) holds and let σ and σ' be, as in (C′), feasible permutations of each other. By Lemma 3.2(ii), there is a sequence of (feasible) strings $\sigma = \sigma_0, \sigma_1, \ldots, \sigma_n = \sigma'$, such that σ_i is obtained from σ_{i-1}, $i = 1, \ldots, n$, by transposing a pair of consecutive events. Under (C), a feasible transposition of a pair of consecutive events does not change the state reached: $\phi(s,\sigma) = \phi(s,\sigma_1) = \cdots = \phi(s,\sigma')$, so (C′) holds.

For (M′), if $N(\sigma_1) = N(\sigma_2)$ then, under (C), $\phi(s,\sigma_1) = \phi(s,\sigma_2)$, so, in particular, $\mathscr{E}(\phi(s,\sigma_1)) = \mathscr{E}(\phi(s,\sigma_2))$. If $N(\sigma_1) + e_i = N(\sigma_2)$ and (C) holds, then σ_2 can be permuted to $\sigma_1\alpha_i$ while maintaining feasibility. Thus, $\phi(s,\sigma_2) = \phi(s_1,\alpha_i)$, where

$s_1 = \phi(s, \sigma_1)$. Since (C) implies noninterruption, $\mathscr{E}(s_1) \setminus \{\alpha_i\} \subseteq \mathscr{E}(\phi(s_1, \alpha_i))$; i.e., $\mathscr{E}(\phi(s, \sigma_1)) \setminus \{\alpha_i\} \subseteq \mathscr{E}(\phi(s, \sigma_2))$, which is the other half of (M'). □

Given an initial state s_0, we view a GSMS with deterministic routing as a completely deterministic mechanism driven by the input $\omega = (\omega_{\alpha_1}, \dots, \omega_{\alpha_m})$, where $\omega_{\alpha_i} = \{\omega_{\alpha_i}(n),\ n = 1, 2, \dots\}$ represents a realization of $\{X_{\alpha_i}(n)\}$. Denote by Ω the space of all ω's, defined as follows: Let

$$I = \left\{ x \in (0, \infty]^{\infty} \colon \sum_{n=1}^{\infty} x(n) = \infty \right\},$$

and let Ω be the product of m copies of I. Restricting attention to I ensures (1). We obtain the GSMP from the GSMS by endowing Ω with a suitable σ-algebra and the measure $\mathscr{P}$.

For any subset A of $\mathbf{A}$, let $\omega_A = (\omega_\alpha)_{\alpha \in A}$ and $T_A = (T_\alpha)_{\alpha \in A}$. Let $\leqslant$ denote the componentwise ordering on Ω: $\omega \leqslant \omega'$ if and only if for all i and n the corresponding elements $\omega_{\alpha_i}(n)$ and $\omega'_{\alpha_i}(n)$ satisfy $\omega_{\alpha_i}(n) \leqslant \omega'_{\alpha_i}(n)$. Let $\leqslant$ also denote the analogous ordering for event epoch sequences $T = (T_{\alpha_1}, \dots, T_{\alpha_m})$. Let $\leqslant_{\text{st}}$ denote the stochastic ordering induced by $\leqslant$, applied either to measures or to associated random elements. In general, if μ_1 and μ_2 are measures on a partially ordered space, then $\mu_2 \leqslant_{\text{st}} \mu_1$ means that $\int f\, d\mu_2 \leqslant \int f\, d\mu_1$ for all increasing, real-valued functions f for which the integrals exist. If Y_i is a random element associated with μ_i, $i = 1, 2$, then $Y_2 \leqslant_{\text{st}} Y_1$ if, equivalently, $\mathbf{E}[f(Y_2)] \leqslant \mathbf{E}[f(Y_1)]$ for every increasing, real-valued f for which the expectations exist.

THEOREM 3.3. *In a GSMS with deterministic routing, with unit speeds, and satisfying* (M), *T is stochastically monotone; that is, $\mathscr{P}^2 \leqslant_{\text{st}} \mathscr{P}^1 \Rightarrow T^2 \leqslant_{\text{st}} T^1$ for all initial states, where T^i is the event epoch sequence under $\mathscr{P}^i$, $i = 1, 2$.*

Theorem 3.3 follows immediately from the following sample path comparison (see, e.g., Kamae, Krengel, and O'Brien [10]). In Lemma 3.4 and throughout this paper, $T(\omega)$ is the event epoch sequence obtained from the input $\omega \in \Omega$ described above, via the construction outlined in §2.

LEMMA 3.4. *Under* (M), *T is increasing in ω; that is, for any initial state s_0, $\omega \leqslant \omega' \Rightarrow T(\omega) \leqslant T(\omega')$.*

PROOF. Let $V = \{V_{\alpha_1}, \dots, V_{\alpha_m}\}$, where $V_{\alpha_i}(n)$ is the epoch of the *nth* setting of a clock for α_i, with $V_{\alpha_i}(n) = \infty$ if a clock for α_i is not set n times. For noninterruptive GSMPs, $T_\alpha(n) = V_\alpha(n) + X_\alpha(n)$, for all α and n; thus, monotonicity of T follows from that of V. To establish the monotonicity of V, we compare the realizations $v = V(\omega)$ and $v' = V(\omega')$ when $\omega \leqslant \omega'$. We proceed by induction, showing that at every transition on the ω'-path, any clock that is set has already been set on the ω-path. Let $\tau_k(\omega')$ be the epoch of the *kth* transition on the ω'-path, and $\tau_0(\omega') = 0$. We show that for all k, and all α and n,

(3) if, on the ω'-path, a clock for α is set for the *nth* time at the *kth* transition, then $v_\alpha(n) \leqslant \tau_k(\omega')$.

Since the first part of (3) implies that $v'_\alpha(n) = \tau_k(\omega')$, the conclusion is that $v_\alpha(n) \leqslant v'_\alpha(n)$.

At the $0th$ transition, (3) holds: both paths set clocks for those events in $\mathscr{E}(s_0)$, so $v_\alpha(1) = 0 = v'_\alpha(1)$ for all $\alpha \in \mathscr{E}(0)$, and $v'_\alpha(1) > 0$ if $\alpha \notin \mathscr{E}(s_0)$. Take as induction hypothesis that (3) holds up to k. Let σ' be the string of the first $k + 1$ events to occur on the ω'-path; let σ be the string of all events to occur on the ω-path in the interval $[0, \tau_{k+1}(\omega')]$. (If $k + 1$ events do not occur in $(0, \infty)$ on the ω'-path, (3) is vacuously satisfied.) The clock for any event in σ' was necessarily set before the $k + 1$*st* transition on the ω'-path. By the induction hypothesis, its clock was set earlier (and therefore ran out earlier) on the ω-path. Thus, $N(\sigma') \leqslant N(\sigma)$.

Now let $s' = \phi(s_0, \sigma')$ and $s = \phi(s_0, \sigma)$. Condition (M) implies that $\mathscr{E}(s') \setminus A_{\sigma'\sigma} \subseteq \mathscr{E}(s)$, with $A_{\sigma'\sigma}$ as defined there. Consider an event, α, for which a clock is set at the $k + 1$*st* transition on the ω'-path—i.e., for which a clock is set upon entry to s'. This is the n'*th* time a clock is set for α (on the ω'-path), where $n' = N_\alpha(\sigma') + 1$. There are two cases. If $N_\alpha(\sigma) > N_\alpha(\sigma')$, then the n'*th* occurrence of α on the ω-path occurred in $[0, \tau_{k+1}(\omega')]$; hence, $v_{n'}(\alpha) < \tau_{k+1}(\omega') = v'_{n'}(\alpha)$. Otherwise, $N_\alpha(\sigma) = N_\alpha(\sigma') = n' - 1$, and (M) implies that $\alpha \in \mathscr{E}(s)$. But if $N_\alpha(\sigma) = n' - 1$ and $\alpha \in \mathscr{E}(s)$, then a clock for α must have been set for the n'*th* time at or before entry to s—that is, at or before $\tau_{k+1}(\omega') = v'_\alpha(n')$. In either case, $v_\alpha(n') \leqslant v'_\alpha(n')$, which is what we needed to show. □

Condition (M) is necessary for monotonicity of all events with respect to all clocks, in the following sense:

THEOREM 3.5. *For a GSMS violating* (M), *it is possible to choose* $\mathscr{P}^2 \leqslant_{\text{st}} \mathscr{P}^1$ *and an initial state* s_0 *for which* $T^2 \nleqslant_{\text{st}} T^1$ *on the resulting GSMPs.*

PROOF. Suppose (M) is not satisfied. Let σ_1 and σ_2 be strings feasible in some s_0, $N(\sigma_1) \leqslant N(\sigma_2)$, violating (M). We may choose $\mathscr{P}^i$, $i = 1, 2$, to concentrate all probability on a deterministic sequence of clocks under which the GSMS follows σ_i, $i = 1, 2$. For such $\mathscr{P}^i$, the construction of T^i on a probability space is essentially unique, so we can show that $T^2 \nleqslant_{\text{st}} T^1$ by treating only the particular probability space described above. Thus, we choose $\mathscr{P}^i$, $i = 1, 2$, to concentrate all probability on an ω^i that generates the event sequence σ_i. Since following σ_i constrains only the *relative* magnitudes of (finitely many) elements of ω^i, we may require that $\omega^2 \leqslant \omega^1$, which makes $\mathscr{P}^2 \leqslant_{\text{st}} \mathscr{P}^1$. We may further require, for any $0 < \epsilon < 1$, that all events in σ_1 occur in $(1, 1 + \epsilon]$, all events in σ_2 occur in $[1 - \epsilon, 1)$, and for all α

$$(4) \qquad n > N_\alpha(\sigma_2) \Rightarrow \omega_\alpha^2(n) > 1 + \epsilon.$$

Since σ_1 and σ_2 violate (M), there is an event α in $\mathscr{E}(\phi(s_0, \sigma_1))$ which is in neither $\mathscr{E}(\phi(s_0, \sigma_2))$ nor $A_{\sigma_1\sigma_2}$. Let $n_\alpha = N_\alpha(\sigma_1) + 1 = N_\alpha(\sigma_2) + 1$, and suppose, without loss of generality, that $\omega_\alpha^1(n_\alpha) = \omega_\alpha^2(n_\alpha)$. Since $\alpha \notin \mathscr{E}(\phi(s_0, \sigma_2))$, some event other than α must follow σ_2 and trigger the setting of the n_α*th* clock for α; hence, (4) implies that $T_\alpha^2(n_\alpha) > 1 + \epsilon + \omega_\alpha^2(n_\alpha)$. On the other hand, $\alpha \in \mathscr{E}(\phi(s_0, \sigma_1))$ implies that the n_α*th* clock for α must be set at or before the completion of σ_1, so $T_\alpha^1(n_\alpha) < 1 + \epsilon + \omega_\alpha^1(n_\alpha)$. But then $T_\alpha^1(n_\alpha) < T_\alpha^2(n_\alpha)$, and $T^2 \nleqslant T^1$. □

REMARK. A longer argument shows that, in Theorem 3.5, $\mathscr{P}^i$, $i = 1, 2$, can be chosen so that the clock samples for each event are i.i.d.

We now illustrate the role of conditions (M) and (C) through examples.

EXAMPLE 3.6. Consider M single-server queues in tandem; the *ith* queue has room for k_i jobs, and $k_1 = \infty$. Let β_0 denote arrival to the first queue, and let β_i denote service completion at i, $i = 1, \dots, M$. If, upon completing service at i, a job finds queue $i + 1$ full, it waits at i, preventing initiation of the next service time, until space becomes available at $i + 1$. Denote a typical state by $(\mathbf{n}, \mathbf{b}) = ((n_1, \dots, n_M), (b_1, \dots, b_M))$, where n_i is the number of jobs at i, and $b_i = 1$ (0) indicates that i is blocked (not blocked). The event β_0 is in every $\mathscr{E}((\mathbf{n}, \mathbf{b}))$, and, for $i > 0$, $\beta_i \in \mathscr{E}((\mathbf{n}, \mathbf{b}))$ if and only if $n_i > 0$ and $b_i = 0$. For any $i = 0, \dots, M$, if $\beta_i \in \mathscr{E}((\mathbf{n}, \mathbf{b}))$, then $\phi((\mathbf{n}, \mathbf{b}), \beta_i) = (\mathbf{n}', \mathbf{b}')$ defined as follows: Let $k_{M+1} \equiv \infty$, $n_{M+1} \equiv$

0 and $b_0 \equiv 0$; let e_i be the *ith* M-dimensional unit vector, $i = 1, \ldots, M$, and $e_{M+1} = e_0$ the vector of all zeros; let empty products be unity, empty sums zero; then

$$\mathbf{n}' = \mathbf{n} + \mathbf{1}\{n_{i+1} < k_{i+1}\} \sum_{j=0}^{i} (e_{j+1} - e_j) \prod_{l=j}^{i-1} b_l,$$

$$\mathbf{b}' = \mathbf{b} - \mathbf{1}\{n_{i+1} < k_{i+1}\} \sum_{j=1}^{i-1} e_j \prod_{l=j}^{i-1} b_l + \mathbf{1}\{n_{i+1} = k_{i+1}\} e_i.$$

By considering separately the four cases $\mathbf{1}\{n_{i_1+1} < k_{i_1+1}\} = 0, 1$, $\mathbf{1}\{n_{i_2+1} < k_{i_2+1}\} = 0, 1$, one readily checks that $\phi((\mathbf{n}, \mathbf{b}), \beta_{i_1}\beta_{i_2}) = \phi((\mathbf{n}, \mathbf{b}), \beta_{i_2}\beta_{i_1})$ whenever $\{\beta_{i_1}, \beta_{i_2}\} \subseteq \mathscr{E}((\mathbf{n}, \mathbf{b}))$. Thus, (C) is satisfied, and Theorem 3.3 implies that speeding up the interarrival times and the service times accelerates the service completions at each node, hence also the throughput of the line. In §8 we drop the condition that $k_1 = \infty$ for a weaker conclusion. In §9 we extend this example to multiple-server queues.

EXAMPLE 3.7. The variant of Example 3.6 in which $k_1 < \infty$ satisfies (C) if the arrival mechanism is "shut off" when $n_1 = k_1$—i.e., if $n_1 = k_1 \Rightarrow \beta_0 \notin \mathscr{E}((\mathbf{n}, \mathbf{b}))$. But if arrivals may be blocked and lost—i.e., $n_1 = k_1 \Rightarrow \phi((\mathbf{n}, \mathbf{b}), \beta_0) = (\mathbf{n}, \mathbf{b})$—then even (M) is violated. We demonstrate this in the case $M = 1$, a single-server, finite capacity queue. Take the state to be the number in the system, β_0 to be arrival, β_1 to be service completion. Let σ be the string consisting of β_0 followed by k β_1's, where k is the total queue capacity. Then $\phi(k, \sigma) = 0$ (the arrival is lost) and $\mathscr{E}(\phi(k, \sigma)) = \{\beta_0\}$. But if σ' is any permutation of σ, then σ' is feasible in k, $\phi(k, \sigma') = 1$ (the arrival is now admitted) and $\mathscr{E}(1) = \{\beta_0, \beta_1\} \neq \mathscr{E}(\phi(k, \sigma))$. Thus, speeding up interarrival times and service times will not *necessarily* cause every event to occur earlier. Sonderman [18] reaches the same conclusion for this example.

EXAMPLE 3.8. This example satisfies (M) but not (C). Consider a cyclic network of M single-server, infinite capacity queues. From node M jobs go back to node 1. So far, (C) is satisfied; but suppose we "tag" a job and give it nonpreemptive priority over all others, without changing its service requirements. Thus, β_i denotes service completion at node i for either the tagged job or any other job. Represent a typical state by $(\mathbf{n}, i, k)$, where $\mathbf{n}$ is the vector of queue lengths, i is the location (node) of the tagged job, and k indicates that the tagged job is in service ($k = 0$) or at the head of the queue ($k = 1$). Since the event list in a state depends only on $\mathbf{n}$, it is easy to see that (M) is satisfied. On the other hand, if $n_{i+1} > 1$, then $\phi((\mathbf{n}, i, 0), \beta_i\beta_{i+1}) = (\mathbf{n} - e_i + e_{i+2}, i + 1, 0) \neq (\mathbf{n} - e_i + e_{i+2}, i + 1, 1) = \phi((\mathbf{n}, i, 0), \beta_{i+1}\beta_i)$, so (C) is violated.

4. Subschemes.

Theorems 3.3 and 3.5 consider comparisons between the event epochs of two GSMPs based on the same scheme but driven by different clock processes. As the examples illustrate, this type of result is useful in comparing queueing systems with the same structure but different service and interarrival processes. To compare, instead, systems with different buffer sizes, number of servers or number of jobs, we need to compare GSMPs based on different schemes. Condition (M) holds the key to this kind of comparison as well.

Consider, for the moment, GSMS in full generality—nondeterministic routing, nontrivial speeds. The following definition captures the idea of one scheme being contained in another. (The superscripts "S" and "B" suggest "small" and "big".)

DEFINITION 4.1. Call $\mathscr{S}^S$ a *subscheme* of $\mathscr{S}^B$ (denoted $\mathscr{S}^S \subseteq \mathscr{S}^B$) if
(i) $\mathbf{S}^S \subseteq \mathbf{S}^B$;
(ii) $\mathscr{E}^S(s) \subseteq \mathscr{E}^B(s)$ for all $s \in \mathbf{S}^S$;
(iii) $p^S(s'; s, \alpha) = p^B(s'; s, \alpha)$ for all $s, s' \in \mathbf{S}^S$ and all $\alpha \in \mathscr{E}^S(s)$;
(iv) $r^S_{s\alpha} = r^B_{s\alpha}$ for all $s \in \mathbf{S}^S$ and all $\alpha \in \mathscr{E}^S(s)$.

Under our (sensible) requirement that for any scheme $\mathbf{A} = \bigcup_s \mathscr{E}(s)$, (i) and (ii) imply that $\mathbf{A}^S \subseteq \mathbf{A}^B$. Think of a GSMS as a labeled, directed graph: the nodes are states; for every s, s' and α such that $p(s'; s, \alpha) > 0$ there is an arc from s to s'; and the label on that arc is $(p(s'; s, \alpha), r_{s\alpha})$. Then one GSMS is a subscheme of another if it is a subgraph. (Embedding one scheme in another may require reinterpretation of states and events, so it may be more natural to think of a mapping f: $\mathscr{S}^S \to \mathscr{S}^B$ for which $f(\mathbf{S}^S) \subseteq \mathbf{S}^B$, $f(\mathbf{A}^S) \subseteq \mathbf{A}^B$, and (ii)–(iv) hold analogously. This view complicates the notation so we do not adopt it.)

Define ϕ^S and ϕ^B for the two schemes as before; then $\phi^S(s,\sigma) = \phi^B(s,\sigma)$ whenever σ is feasible for $\mathscr{S}^S$ starting in s. Clearly, any such string is also feasible for $\mathscr{S}^B$ starting in s. Let T^S and T^B denote the corresponding event epoch sequences. The following result ties a change in scheme to a change in clock times. To lighten the notation, we carry out only the case $\mathbf{A}^S = \mathbf{A}^B$; this allows us to construct T^S and T^B on the same Ω.

LEMMA 4.2. *Suppose $\mathscr{S}^S$ and $\mathscr{S}^B$ have deterministic routing and unit speeds, and $\mathscr{S}^S \subseteq \mathscr{S}^B$. Consider T^B and T^S starting from a common, fixed state $s_0 \in \mathbf{S}^S$. If $\mathscr{S}^S$ is noninterruptive, then for all $\omega \in \Omega$ there exists $\omega' \in \Omega$, such that $\omega \leqslant \omega'$ and $T^B(\omega') = T^S(\omega)$.*

PROOF. We construct the required ω'. The idea is to choose ω' so that when $\mathscr{S}^B$ is driven by ω' it "mimics" the evolution of $\mathscr{S}^S$. Since clocks are potentially set earlier on $\mathscr{S}^B$ (more events are active in corresponding states), this requires prolonging some clock times.

Let σ_k be the string consisting of the first k events to occur when $\mathscr{S}^S$ is driven by ω, provided at least k events actually occur in $(0, \infty)$. Let $\tau_k(\omega)$ be the corresponding epoch of the kth event. For all $\alpha \in \mathscr{E}^S(s_0)$, set $\omega'_\alpha(1) = T^S_\alpha(1)$. Suppose $\sigma_k = \sigma_{k-1}\alpha^*_k$. Suppose, for some $k \geqslant 1$,

$$(5) \qquad \alpha \in \mathscr{E}^B(\phi^B(s_0, \sigma_k)) \setminus (\mathscr{E}^B(\phi^B(s_0, \sigma_{k-1})) - \{\alpha^*_k\});$$

i.e., a clock would be set for α in $\mathscr{S}^B$ at the kth transition if $\mathscr{S}^B$ followed σ_k. If (5) holds, set

$$(6) \qquad \omega'_\alpha(N_\alpha(\sigma_k) + 1) = T^S_\alpha(N_\alpha(\sigma_k) + 1)(\omega) - \tau_k(\omega).$$

(This definition is consistent; if (5) also holds for $k' > k$ then $N_\alpha(\sigma_{k'}) > N_\alpha(\sigma_k)$.) For any remaining α and n (not covered by (5) and (6)), $T^S_\alpha(n)(\omega) = \infty$ so set $\omega'_\alpha(n) = \infty$. This construction makes $\omega' \geqslant \omega$. Moreover, the sequence and timing of events (indeed, even the sequence of states) when $\mathscr{S}^B$ is driven by ω' are the same as when $\mathscr{S}^S$ is driven by ω. By construction, every $\omega'_\alpha(n)$ is the difference between $T^S_\alpha(n)$ and the epoch of the nth setting of a clock for α in $\mathscr{S}^B$. □

Now consider GSMPs based on $\mathscr{S}^S$ and $\mathscr{S}^B$ via $\mathscr{P}^S$ and $\mathscr{P}^B$. Let T^B and T^S be the corresponding event epoch sequences, starting from a common state $s_0 \in \mathbf{S}^S$.

THEOREM 4.3. *Suppose $\mathscr{S}^S$ and $\mathscr{S}^B$ have deterministic routing and unit speeds, $\mathscr{S}^S \subseteq \mathscr{S}^B$, and $\mathscr{P}^B = \mathscr{P}^S$. If $\mathscr{S}^S$ is noninterruptive and $\mathscr{S}^B$ satisfies* (M), *then $T^B \leqslant_{\mathrm{st}} T^S$.*

PROOF. It is enough to show that for all ω, $T^B(\omega) \leqslant T^S(\omega)$. From Lemma 4.2 we know there is an ω' such that (i) $\omega \leqslant \omega'$ and (ii) $T^B(\omega') = T^S(\omega)$. Then

$$\omega \leqslant \omega' \Rightarrow T^B(\omega) \leqslant T^B(\omega') \quad \text{(by Lemma 3.4)}$$

$$\Rightarrow T^B(\omega) \leqslant T^S(\omega) \quad \text{(by (ii))}. \quad \square$$

REMARK. Lemma 4.2 and Theorem 4.3 easily extend to the case $\mathbf{A}^S \subseteq \mathbf{A}^B$. In the proof of Lemma 4.2, set $\omega'_\alpha(n) = \infty$ whenever $\alpha \in \mathbf{A}^B \setminus \mathbf{A}^S$. In Theorem 4.3 we then get $\mathscr{P}^B_{\mathbf{A}^S} = \mathscr{P}^S \Rightarrow T^B_{\mathbf{A}^S} \leqslant_{\mathrm{st}} T_S$.

EXAMPLE 4.4. Consider, again, queues in tandem as in Example 3.6, still requiring that the first queue have infinite capacity. We may embed such a system into one in which some k_i, $i > 1$, is increased by unity by identifying the state $(\mathbf{n}, \mathbf{b})$ in the original system with state $(\mathbf{n} + e_i, \mathbf{b})$ in the enlarged system. This makes the original GSMS a subscheme of the one with more capacity. Since (M) is satisfied by both, we conclude immediately that increasing buffer size accelerates the occurrence of all events. If we apply the construction of Lemma 4.2 to this example, we find that the enlarged system can "mimic" the original by prolonging some β_{i-1}-clocks: To produce the same T-sequence, the enlarged system must delay service at $i - 1$ whenever, in the smaller system, $i - 1$ would be blocked. Using the device of §9, we could, instead, increase the number of servers at node i. In this case, we embed the original system by mapping states to states that differ only in that the added servers at i are busy. Similarly, in a closed cyclic network we could add jobs. Either change accelerates events. Similar conclusions are drawn in Tsoucas and Walrand [22]; our example is more general in that we make no independence assumptions, but less general in that we do not, as yet, allow $k_1 < \infty$; see §8.

5. Probabilistic schemes.

We drop the assumption that every $p(s'; s, \alpha)$ is zero or one. Without loss of generality, we may suppose that the state transitions are governed by *transition mappings* as follows: For each $\alpha \in \mathbf{A}$ there are mappings f_i^α: $\mathbf{S} \to \mathbf{S}$, $i = 1, \ldots, m_\alpha$ (where m_α may be infinite) such that if $\alpha \in \mathscr{E}(s)$,

$$\sum_{i=1}^{m_\alpha} p(f_i^\alpha(s); s, \alpha) = 1.$$

DEFINITION 5.1. A GSMS has *state-independent routing* if for each $\alpha \in \mathbf{A}$ there are strictly positive constants $\{p_i^\alpha, i = 1, \ldots, m_\alpha\}$ summing to unity, and mappings $\{f_i^\alpha, i = 1, \ldots, m_\alpha\}$, such that for all s with $\alpha \in \mathscr{E}(s)$, $p(f_i^\alpha(s); s, \alpha) = p_i^\alpha$.

In a GSMS with state-independent routing, if α is in both $\mathscr{E}(s)$ and $\mathscr{E}(s')$, the possible transitions out of s and s' due to α are in one-to-one correspondence; and corresponding transitions $s \to f_i^\alpha(s)$ and $s' \to f_i^\alpha(s')$ have the same probability, p_i^α. This property is typical of queueing networks with state-independent routing, where we can make a correspondence between transitions in different states that reflect the movement of a job between a particular pair of nodes. Schemes with deterministic routing automatically have state-independent routing: take $m_\alpha = 1$, $f_1^\alpha(s) = \phi(s, \alpha)$.

To accommodate probabilistic routing, we enlarge our probability space. Let

$$\Pi = \{1, \ldots, m_{\alpha_1}\}^\infty \times \cdots \times \{1, \ldots, m_{\alpha_m}\}^\infty,$$

and let $\tilde{\Omega} = \Omega \times \Pi$. Denote a typical element of Π by $\pi = (\pi_{\alpha_1}, \ldots, \pi_{\alpha_m})$, where $\pi_{\alpha_i} = \{\pi_{\alpha_i}(n), n = 1, 2, \ldots\}$. Denote a typical element of $\tilde{\Omega}$ by $\tilde{\omega} = (\omega, \pi)$. For each

α and n, if the nth occurrence of α occurs in state s, it triggers a transition to $f^{\alpha}_{\pi_\alpha(n)}(s)$. Thus, to induce state-independent routing, we endow Π with the product measure that assigns the probability mass vector $(p^\alpha_1, \dots, p^\alpha_{m_\alpha})$ to each copy of $\{1, \dots, m_\alpha\}$ in Π, for each α.

Through this augmentation of Ω, a GSMP with state-independent routing becomes a deterministic function of $\tilde{\omega} = (\omega, \pi)$. For $\pi \in \Pi$ and $\alpha \in \mathscr{E}(s)$, define $\phi_\pi(s, \alpha) = f^{\alpha}_{\pi_\alpha(1)}(s)$. Extend $\phi_\pi(s, \cdot\,)$ to (feasible) strings recursively via

$$\phi_\pi(s, \sigma\alpha) = f^{\alpha}_{\pi_\alpha(N_\alpha(\sigma)+1)}(\phi_\pi(s, \sigma)).$$

Then $\phi_\pi(s, \sigma)$ is, indeed, the state reached through the string of events σ following the state transitions as determined by π. With π held fixed, the evolution of the GSMP is completely determined by ω. The transitions are "deterministic", but have "memory" in the sense that the transition triggered by α depends on how many times α has already occurred.

The following is the generalization of (M) to schemes with probabilistic routing:

(PM). *Probabilistic Monotonicity Condition*. The GSMS has state-independent routing, and for all $\pi \in \Pi$, (M) holds with ϕ replaced by ϕ_π.

The analog for (C) is sometimes easier to work with:

(PC). *Probabilistic Commuting Condition*. The GSMS has state-independent routing; and for all s, all $\{\alpha, \beta\} \subseteq \mathscr{E}(s)$, and all $i = 1, \dots, m_\alpha$, $j = 1, \dots, m_\beta$, $f^\beta_j(f^\alpha_i(s)) = f^\alpha_i(f^\beta_j(s))$.

PROPOSITION 5.2. (PC) ⇒ (PM).

PROOF. We show that under (PC), for each $\pi \in \Pi$, and every σ feasible in s, $\phi_\pi(s, \sigma)$ is a function of $N(\sigma)$ only. This means that (C′)—hence, also (M)—is satisfied by every ϕ_π. Let $\sigma = \beta_1 \cdots \beta_k$, and let

$$\pi_i = \pi_{\beta_i}(N_{\beta_i}(\beta_1 \cdots \beta_i));$$

then

$$\phi_\pi(s, \sigma) = f^{\beta_k}_{\pi_k} \circ \cdots \circ f^{\beta_1}_{\pi_1}(s).$$

As in Lemma 3.2, any feasible permutation of σ can be obtained through a sequence of transpositions of consecutive events that maintain feasibility. Transposing β_i and β_{i+1} maintains feasibility if and only if $\beta_{i+1} \in \mathscr{E}(\phi_\pi(s, \beta_1, \dots, \beta_{i-1}))$. In this case, (PC) implies that reversing the order of β_i and β_{i+1} does not change the state reached:

$$f^{\beta_{i+1}}_{\pi_{i+1}} \circ f^{\beta_i}_{\pi_i}(\phi_\pi(s, \beta_1 \cdots \beta_{i-1})) = f^{\beta_i}_{\pi_i} \circ f^{\beta_{i+1}}_{\pi_{i+1}}(\phi_\pi(s, \beta_1 \cdots \beta_{i-1})).$$

Applying this to each transition, we conclude that $\phi_\pi(s, \sigma)$ is unchanged if σ is replaced by any feasible permutation. □

THEOREM 5.3. *Theorems* 3.3 *and* 4.3 *hold with* (M) *replaced by* (PM).

This is established by replacing ϕ with ϕ_π in the proofs of the earlier results.

REMARK. For schemes with state-independent routing, (PM) is necessary for T to be increasing in ω for all π, *under our construction* on $\tilde{\Omega} = \Omega \times \Pi$. But we cannot from this generalize Theorem 3.5 and conclude that (PM) is necessary for T to be stochastically increasing in $\mathscr{P}$. Even for $\mathscr{P}$'s that make the clock times deterministic, the presence of nontrivial p's makes the choice of probability space fundamentally nonunique; and we cannot rule out the possibility of an entirely different construction. The necessity of (PM) under one construction does not imply its necessity for all constructions.

EXAMPLE 5.4. Consider a generalized Jackson network of single-server queues. Node i has capacity k_i. Routing of class c jobs is governed by a Markovian, state-independent routing matrix (P_{ij}^c). A fictitious node 0 is the source of every arrival to the network, and the destination of every departure from the network; thus, we consider open, closed and mixed networks simultaneously. Let the state of the network be the vector of queue lengths, supplemented with information about the order in queue of jobs of different classes, and about which nodes are blocked by which other nodes. For each i and c, let β_i^c be completion of service by a class c job at node i. Let α^c be external arrival of a class c job. Service at every node is first come, first served; jobs of different classes may have different service requirements and different routing, but there are no priorities. Jobs blocked internally wait where they completed service.

We impose the following restrictions:

$$P_{0i}^c > 0 \quad \text{for some} \quad c \Rightarrow k_i = \infty; \tag{7}$$

$$P_{ji}^c > 0 \quad \text{and} \quad P_{j'i}^c > 0, \quad j' \neq j \Rightarrow k_i = \infty; \tag{8}$$

$$P_{ji}^c > 0, P_{j'i}^{c'} > 0 \quad \text{and} \quad c \neq c' \Rightarrow j = j'. \tag{9}$$

In words, (7) prevents blocking of external arrivals, (8) requires that a finite capacity queue be fed by a single source, and (9) requires that a queue visited by more than one class of jobs be fed by a single source. (Implicit in (7) is the assumption that blocked external arrivals would be lost. If, instead, each queue had its own external arrival stream which would "shut off" when the queue was full, then (7) would be unnecessary; see Example 3.7.)

Under conditions (7)–(9), (PC) is satisfied; no weakening of these conditions ensures even (PM). (To verify (PC), take $f_i^\alpha(s)$ to be the state reached from s when a job attempts to join queue i upon the occurrence of α, where α is any service completion or external arrival.) To see why we need (8), for example, suppose that j and j', $j \neq j'$, both feed i, and $k_i < \infty$. Then speeding up service at j might cause j' to be blocked by i more often, and may therefore delay service at j'.

From Theorem 5.3 we get

PROPOSITION 5.5. *For the class of networks described above, T is stochastically increasing in $\mathscr{P}$, and decreasing in the buffer sizes and in the number of jobs of each class.*

6. Speeds.

Suppose, now, that the clock for α runs down at speed $r_{s\alpha}$ in state s. We adopt the convention that $r_{s\alpha} = 0$ if $\alpha \notin \mathscr{E}(s)$. We always require that for all α and s, $r_{s\alpha} < \infty$.

(SM). *Speeds Monotonicity Condition*. The GSMS has state-independent routing; and for all π, all s, and all σ_1, σ_2 feasible in s, if $s_1 = \phi_\pi(s, \sigma_1)$, $s_2 = \phi_\pi(s, \sigma_2)$, and

$N(\sigma_1) \leqslant N(\sigma_2)$ then

$$\alpha \in \mathscr{E}(s_1) \setminus A_{\sigma_1\sigma_2} \Rightarrow r_{s_1\alpha} \leqslant r_{s_2\alpha}.$$

This reduces to (PM) when $r_{s\alpha} = \mathbf{1}\{\alpha \in \mathscr{E}(s)\}$. If, in (SM), $N(\sigma_1) = N(\sigma_2)$, then all events have the same speed in $\phi_\pi(s, \sigma_1)$ and $\phi_\pi(s, \sigma_2)$. An important special case of (SM) is

(SC). *Speeds Commuting Condition.* In addition to (PC), if $\{\alpha, \beta\} \subseteq \mathscr{E}(s)$, then $p(s'; s, \beta) > 0 \Rightarrow r_{s\alpha} \leqslant r_{s'\alpha}$.

PROPOSITION 6.1. (SC) $\Rightarrow$ (SM).

PROOF. Similar to Propositions 3.1 and 5.2. □

The following further generalizes Theorem 3.3:

THEOREM 6.2. *Consider two GSMPs with clock laws $\mathscr{P}^1, \mathscr{P}^2$, and based on the same scheme, $\mathscr{S}$. Let T^i, $i = 1, 2$ be the corresponding event epoch sequences. If $\mathscr{S}$ satisfies* (SM), *then $\mathscr{P}^2 \leqslant_{\text{st}} \mathscr{P}^1 \Rightarrow T^2 \leqslant_{\text{st}} T^1$ for all initial states.*

PROOF. Let Π be as before. It is enough to show that for every $\pi \in \Pi$, when transitions are determined by ϕ_π, the event epoch sequence T for $\mathscr{S}$ is increasing in ω. Let $\omega' \geqslant \omega$. As in the proof of Lemma 3.4, let $\tau_k(\omega')$ be the epoch of the *kth* transition on the ω'-path; let σ_k' be the string of the first k events to occur on the ω'-path; let σ_k be the string of all events to occur in $[0, \tau_k(\omega')]$ on the ω-path; let $s_k' = \phi_\pi(s_0, \sigma_k')$ and $s_k = \phi_\pi(s_0, \sigma_k)$, where s_0 is the initial state. We first show by induction that, for all k, $N(\sigma_k') \leqslant N(\sigma_k)$. (It is enough to consider those k for which k events actually occur on the ω'-path—i.e., for which $\tau_k(\omega') < \infty$.) This holds for $k = 0$; suppose it holds up to k. If $\sigma_{k+1}' = \sigma_k\alpha$, we need only show that $n' \equiv N_\alpha(\sigma_{k+1}') \leqslant N_\alpha(\sigma_{k+1})$. On the ω'-path, the *n'th* clock for α was set upon entry to s_l, $l \leqslant k$. By hypothesis, $N(\sigma_l') \leqslant N(\sigma_l)$, and (SM) implies that either $\alpha \in A_{\sigma_l'\sigma_l}$ or $N_\alpha(\sigma_l') = N_\alpha(\sigma_l)$ and $\alpha \in \mathscr{E}(s_l)$. In the first case, $T_\alpha(n')(\omega) \leqslant \tau_l(\omega')$ so $N_\alpha(\sigma_{k+1}) \geqslant n'$ and we are done. In the second case, for every $j = l, \dots, k$, $N(\sigma_j') \leqslant N(\sigma_j)$ so (SM) implies that $r_{s_j'\alpha} \leqslant r_{s_j\alpha}$ for every $j = l, \dots, k$. Thus, the *n'th* clock for α is set no later on the ω-path than on the ω'-path, and is always run faster; hence, it can run out no later. In other words, $N_\alpha(\sigma_{k+1}) \geqslant n' = N_\alpha(\sigma_{k+1}')$.

We may now conclude that $T(\omega) \leqslant T(\omega')$: if, for some α and n, $T_\alpha(n)(\omega)$ were greater than $T_\alpha(n)(\omega')$, then there would be a k for which $N_\alpha(\sigma_k') > N_\alpha(\sigma_k)$. In particular, we could choose k so that $\tau_k(\omega') = T_\alpha(n)(\omega')$. □

We can also compare GSMPs with different speeds:

THEOREM 6.3. *Consider two GSMPs based on schemes $\mathscr{S}^1$ and $\mathscr{S}^2$ differing only in their speeds. Suppose the two GSMPs have the same clock law $\mathscr{P}$, and $\mathscr{S}^1$ and $\mathscr{S}^2$ both satisfy* (SM). *Then $r^2 \geqslant r^1 \Rightarrow T^2 \leqslant_{\text{st}} T^1$ for all initial states. Conversely, if for all $\mathscr{P}$ and all initial states $T^2 \leqslant_{\text{st}} T^1$, then $r^2 \geqslant r^1$, even if* (SM) *is not satisfied.*

PROOF. Arguing just as in the proof of Theorem 6.2, we find that when both $\mathscr{S}^1$ and $\mathscr{S}^2$ are driven by the same input ω, every clock is set earlier and runs faster under $\mathscr{S}^1$ than $\mathscr{S}^2$, so $T^1(\omega) \leqslant T^2(\omega)$. For the converse, suppose there are s and α such that $r_{s\alpha}^1 > r_{s\alpha}^2$, and take this s as initial state. Let $\mathscr{P}$ make all clocks for α identically x. To simplify the argument, let $\mathscr{P}$ make all clocks for all other events infinite—any sufficiently large value would do. Clearly, $T_\alpha^i(1) = x/r_{s\alpha}^i$, $i = 1, 2$, where division by zero yields infinity. Now x can be chosen so that $T_\alpha^2(1) > T_\alpha^1(1)$. □

THEOREM 6.4. *Suppose $\mathscr{S}^S$ and $\mathscr{S}^B$ satisfy $\mathscr{S}^S \subseteq \mathscr{S}^B$, with $\mathbf{A}^S = \mathbf{A}^B$ and* (iv) *of Definition* 4.1 *relaxed to $r_{s\alpha}^S \leqslant r_{s\alpha}^B$. Suppose $\mathscr{P}^B = \mathscr{P}^S$. If $\mathscr{S}^S$ is noninterruptive and $\mathscr{S}^B$ satisfies* (SM), *then $T^B \leqslant_{\text{st}} T^S$ for every common initial state in $\mathbf{S}^S$.*

PROOF. We need to check that Lemma 4.2 still holds—the rest of the proof is the same as in Theorem 4.3. Fix an initial state $s_0 \in \mathbf{S}^S$. Since $\mathbf{A}^S = \mathbf{A}^B$, we may let $\Omega^S = \Omega^B = \Omega$ and $\Pi^S = \Pi^B = \Pi$. Fix a $\pi \in \Pi$ and let transitions be determined by π for both $\mathscr{G}^B$ and $\mathscr{G}^S$, via ϕ^B_π and ϕ^S_π. We want to find, for every $\omega \in \Omega$, an $\omega' \in \Omega$ such that $\omega \leqslant \omega'$ and $T^B(\omega') = T^S(\omega)$. Let σ_k be the string consisting of the first k events when $\mathscr{G}^S$ is driven by ω, provided k events actually occur. Let $\tau_k(\omega)$ be the epoch of the kth event. Let $s_k = \phi^S_\pi(s_0, \sigma_k)$, and $\sigma_k = \sigma_{k-1}\alpha^*_k$. Suppose, for some $k \geqslant 1$,

$$\alpha \in \mathscr{E}^B(\phi^B_\pi(s_0, \sigma_k)) \setminus (\mathscr{E}^B(\phi^B(s_0, \sigma_{k-1})) - \{\alpha^*_k\}),$$

or simply $\alpha \in \mathscr{E}^B(s_0)$ for the case $k = 0$. If $T^S_\alpha(N_\alpha(\sigma_k) + 1) = \infty$, set $\omega'_\alpha(N_\alpha(\sigma_k) + 1) = \infty$; otherwise, choose the smallest j for which $\tau_j(\omega) = T^S_\alpha(N_\alpha(\sigma_k) + 1)$ and set

$$\omega'_\alpha(N_\alpha(\sigma_k) + 1) = r^B_{s_k\alpha}[\tau_{k+1}(\omega) - \tau_k(\omega)] + \cdots + r^B_{s_{j-1}\alpha}[\tau_j(\omega) - \tau_{j-1}(\omega)].$$

Set any remaining $\omega'_\alpha(n)$ to infinity. Since, in the situation described,

$$\omega_\alpha(N_\alpha(\sigma_k) + 1) = r^S_{s_k\alpha}[\tau_{k+1}(\omega) - \tau_k(\omega)] + \cdots + r^S_{s_{j-1}\alpha}[\tau_j(\omega) - \tau_{j-1}(\omega)],$$

this definition makes $\omega' \geqslant \omega$. Moreover, under this definition, a clock set to $\omega'_\alpha(N_\alpha(\sigma_k) + 1)$ at $\tau_k(\omega)$ and run down at speed $r_{s_i\alpha}$ during the interval $[\tau_i(\omega), \tau_{i+1}(\omega))$, $i = k, \dots, j-1$, runs out at $\tau_j(\omega) = T^S_\alpha(N_\alpha(\sigma_k) + 1)$, which is what we needed. □

REMARK. Theorem 6.4 easily extends to the case $\mathbf{A}^S \subseteq \mathbf{A}^B$; see the remark following Theorem 4.3.

EXAMPLE 6.5. Consider a simplification of Example 5.4. The network is closed; there is only one class of jobs; every $k_i = \infty$; but we allow the servers to work at load-dependent rates. Represent the state of the network by a vector $\mathbf{n}$ of queue lengths. Let β_i be service completion at node i, and let there be $\boldsymbol{\mu} = (\mu_i)$ such that $r_{\mathbf{n}\beta_i} = \mu_i(n_i)$. In other words, the speed of service at node i is a function of the queue length at i. Condition (SC) is satisfied if every μ_i is an increasing function. Thus, if $\boldsymbol{\mu}$ is increasing, adding a job to the network increases throughput. This generalizes a similar result in Shanthikumar and Yao [16]. If $\boldsymbol{\mu}^1$ and $\boldsymbol{\mu}^2$ are both increasing, and $\boldsymbol{\mu}^1 \leqslant \boldsymbol{\mu}^2$, then all service completions occur earlier when the service speeds are $\boldsymbol{\mu}^2$ than when they are $\boldsymbol{\mu}^1$.

7. Relevance.

This section and the three that follow treat modifications of our previous results, in some cases obtaining weaker conclusions under weaker assumptions. To simplify the exposition, in what remains *we take all schemes to have unit speeds*. We begin, in this section, by giving a condition for comparing GSMPs based on the same scheme and having different clock processes for only a *subset* of the events. We then make precise the idea that, in some cases, there are subsets A and B of $\mathbf{A}$ such that $T_B = (T_\beta)_{\beta \in B}$, the event epochs of B, depend on $\mathscr{P}$ only through $\mathscr{P}_A = (\mathscr{P}_\alpha)_{\alpha \in A}$, the clocks of A. To facilitate comparisons based on marginals of $\mathscr{P}$, we will assume that $\mathscr{P} = \mathscr{P}_{\alpha_1} \times \cdots \times \mathscr{P}_{\alpha_m}$; in other words, the clock processes for different events are independent. When this is the case, we say that $\mathscr{P}$ *factors* over $\mathbf{A}$. If $\mathscr{P}^i$, $i = 1, 2$, factor, then $\mathscr{P}^2 \leqslant_{\text{st}} \mathscr{P}^1$ if and only if $\mathscr{P}^2_\alpha \leqslant_{\text{st}} \mathscr{P}^1_\alpha$ for all $\alpha \in \mathbf{A}$.

Our results depend on the idea of *relevance* introduced in Glasserman [4]. It identifies when the epochs of one event depend on those of another event:

DEFINITION 7.1. For any $\alpha \in \mathbf{A}$, the set $R(\alpha)$ of *α-relevant* events is as follows: (i) $\alpha \in R(\alpha)$; and (ii) if $\alpha' \in R(\alpha)$, $p(s'; s, \alpha) > 0$ and $\alpha'' \in \mathscr{E}(s') \setminus \mathscr{E}(s)$ then $\alpha'' \in R(\alpha)$. For $A \subseteq \mathbf{A}$, the set of A-relevant events is $R(A) = \bigcup_{\alpha \in A} R(\alpha)$.

Thus, $\alpha' \in R(A)$ if $\alpha' \in A$ or if a clock for α' is ever set by the occurrence of an event in $R(A)$. Informally, $R(A)$ is the closure of A under triggering. Let $R^{-1}(\alpha) = \{\beta\colon \alpha \in R(\beta)\}$, and let $R^{-1}(A) = \bigcup_{\alpha \in A} R^{-1}(\alpha)$. Write $\bar{A}$ for $\mathbf{A} \setminus A$. Simple properties of R and R^{-1} are summarized in

LEMMA 7.2. (i) *R and R^{-1} are idempotent and increasing*: $R(R(A)) = R(A)$, $A \subseteq B \Rightarrow R(A) \subseteq R(B)$, $R^{-1}(R^{-1}(A)) = R^{-1}(A)$, $A \subseteq B \Rightarrow R^{-1}(A) \subseteq R^{-1}(B)$;

(ii) $A \subseteq R(A) \subseteq R^{-1}(R(A))$, $A \subseteq R^{-1}(A) \subseteq R(R^{-1}(A))$;

(iii) $R^{-1}(R(A)) \setminus (R(\bar{A})) \subseteq A$, $R(R^{-1}(A)) \setminus (R^{-1}(\bar{A})) \subseteq A$.

For any $A \subseteq \mathbf{A}$ we have the following relaxed version of (PC):

(R_A). *Relevance Condition.* Condition (PC) holds whenever α or β is in $R(A)$.

THEOREM 7.3. *Fix $A \subseteq \mathbf{A}$ and suppose $\mathscr{S}$ satisfies* (R_A). *Let T^i, $i = 1, 2$ be induced by $\mathscr{P}^i$, $i = 1, 2$, each $\mathscr{P}^i$ factoring over* $\mathbf{A}$. *If $\mathscr{P}^2_A \leqslant_{\mathrm{st}} \mathscr{P}^1_A$ and $\mathscr{P}^2_{\bar{A}} = \mathscr{P}^1_{\bar{A}}$, then $T^2 \leqslant_{\mathrm{st}} T^1$.*

PROOF. Let $T(\omega)$ be the event epoch sequence when $\mathscr{S}$ is driven by ω for any fixed $\pi \in \Pi$, as in §5. It is enough to show that if $\omega_A \leqslant \omega'_A$ and $\omega_{\bar{A}} = \omega'_{\bar{A}}$ then $T(\omega) \leqslant T(\omega')$. An event in $R \equiv R(A)$ never triggers the setting of a clock for an event in $\bar{R}$ (since $R(R(A)) = R(A)$), so a simple inductive argument shows that if $\omega_{\bar{R}} = \omega'_{\bar{R}}$ and (R_A) holds, then $T_{\bar{R}}(\omega) = T_{\bar{R}}(\omega')$. Also, $\omega_{\bar{A}} = \omega'_{\bar{A}} \Rightarrow \omega_{\bar{R}} = \omega'_{\bar{R}}$ because $A \subseteq R(A) \Rightarrow \overline{R(A)} \subseteq \bar{A}$. Thus, changing ω to ω' does not change the order or timing of events in $\bar{R}$. The rest of the proof is the same as that of Lemma 3.4. If, as in Lemma 3.4, σ' is the string of the first k events on the ω'-path and σ is the string of events on the ω-path that occur in the same time interval, then the order of events in $\bar{R}$ is the same in σ and σ'. Thus, in comparing σ and σ', condition (R_A) is no weaker than (PC). □

Since $B \subseteq A \Rightarrow R(B) \subseteq R(A)$, $(\mathrm{R}_A) \Rightarrow (\mathrm{R}_B)$, and in Theorem 7.3 we may, of course, take $\mathscr{P}^1_B \leqslant_{\mathrm{st}} \mathscr{P}^2_B$ and equality on $\bar{B}$. Since, also, $R(A) = R(R(A))$, we may instead take $\mathscr{P}^1_{R(A)} \leqslant_{\mathrm{st}} \mathscr{P}^2_{R(A)}$ and equality on $\overline{R(A)}$.

EXAMPLE 7.4. Consider a single-server queue fed by n arrival streams $\alpha^1, \ldots, \alpha^n$ of jobs of different classes. Let $\beta^1, \ldots, \beta^n$ be the corresponding service completion events. Let the state be the order of jobs in the system. Service is first come, first served. This system violates (PM). (For example, $\mathscr{E}(\phi(0, \alpha^i\alpha^j)) = \{\beta^i\} \neq \{\beta^j\} = \mathscr{E}(\phi(0, \alpha^j\alpha^i))$.) Condition (R_A) is violated if A contains any arrivals. But if A is any nonempty subset of $\{\beta^1, \ldots, \beta^n\}$, then $R(A) = \{\beta^1, \ldots, \beta^n\}$ and (R_A) holds: a change in the order of a service completion and an arrival will not change the resulting state. Thus, speeding up some or all of the service times while holding the interarrival times fixed never delays any events.

The notion of relevance is useful even when (PM) is satisfied, in which case it indicates which clock times need to be ordered to ensure that the epochs of a subset of events are similarly ordered. In the following sense, T_A depends on $\mathscr{P}$ only through $\mathscr{P}_{R^{-1}(A)}$:

THEOREM 7.5. *Suppose $\mathscr{S}$ satisfies* (PM), *and $\mathscr{P}^i$, $i = 1, 2$, factor over* $\mathbf{A}$. *If $\mathscr{P}^2_{R^{-1}(A)} \leqslant_{\mathrm{st}} \mathscr{P}^1_{R^{-1}(A)}$ then $T^2_A \leqslant_{\mathrm{st}} T^1_A$.*

PROOF. Let $B = R^{-1}(A)$. We show that if $\omega_B \leqslant \omega'_B$ then $T_A(\omega) \leqslant T_A(\omega')$. Let $N_B(\sigma) = (N_\alpha(\sigma))_{\alpha \in B}$. We first show that if σ_1 and σ_2 are feasible in s and

$N_B(\sigma_1) \leqslant N_B(\sigma_2)$ then

$$(10) \qquad \left[\mathscr{E}(\phi_\pi(s,\sigma_1)) \setminus A_{\sigma_1\sigma_2}\right] \cap B \subseteq \mathscr{E}(\phi_\pi(s,\sigma_2)) \cap B,$$

for all π. Since $R^{-1}(B) = B$, $\bar{B}$ and $R^{-1}(B)$ are disjoint—no event in $\bar{B}$ ever triggers the setting of a clock for an event in B. Hence, there is a feasible permutation of σ_i into $\sigma_i = \sigma_i^B\sigma_i^{\bar{B}}$, $i = 1, 2$, where all events in σ_i^B ($\sigma_i^{\bar{B}}$) are in B ($\bar{B}$) (use Lemma 3.2(i) repeatedly). Permuting σ_i (feasibly) leaves $\mathscr{E}(\phi_\pi(s,\sigma_i))$ unchanged. Moreover, if events in $\bar{B}$ never activate events in B,

$$(11) \qquad \mathscr{E}(\phi_\pi(s,\sigma_i)) \cap B = \mathscr{E}\left(\phi_\pi\left(s,\sigma_i^B\sigma_i^{\bar{B}}\right)\right) \cap B = \mathscr{E}\left(\phi_\pi\left(s,\sigma_i^B\right)\right) \cap B,$$
$$i = 1, 2.$$

By hypothesis, $N(\sigma_1^B) \leqslant N(\sigma_2^B)$, so (PM) and $A_{\sigma_1^B\sigma_2^B} = A_{\sigma_1\sigma_2} \cap B$ imply that

$$(12) \qquad \mathscr{E}\left(\phi_\pi\left(s,\sigma_1^B\right)\right) \setminus A_{\sigma_1\sigma_2} \subseteq \mathscr{E}\left(\phi_\pi\left(s,\sigma_2^B\right)\right).$$

Together, (11) and (12) yield (10).

Given (10), the result follows exactly as in the proof of Lemma 3.4: In the initial state, clocks for events in B are set no later under ω than ω'. If $\omega_B \leqslant \omega'_B$ and (10) holds, then this is preserved at every transition. Since $A \subseteq B$, this means that $\omega_B \leqslant \omega'_B \Rightarrow T_A(\omega) \leqslant T_A(\omega')$. □

Now let A and B be generic subsets of **A** for a GSMS satisfying (PM). Consider the validity of the implication $\mathscr{P}_B^2 \leqslant_{\text{st}} \mathscr{P}_B^1 \Rightarrow T_A^2 \leqslant_{\text{st}} T_A^1$, with either A or B specified. From Theorem 7.5 and its proof, we have

COROLLARY 7.6. *Suppose $\mathscr{G}$ satisfies* (PM), *and $\mathscr{P}^i$, $i = 1, 2$, factor over* **A**. *If B contains $R^{-1}(A)$ or, equivalently, $A \subseteq R(B) \setminus R(\bar{B})$, then $\mathscr{P}_B^2 \leqslant_{\text{st}} \mathscr{P}_B^1 \Rightarrow T_A^2 \leqslant_{\text{st}} T_A^1$.*

The equivalence of the two conditions follows from Lemma 7.2(ii) and (iii).

8. Extractions.

Theorems 4.3 and 6.4 provide a means of comparing GSMPs based on different schemes when one is a subscheme of another. While the subscheme relation is convenient and quite broadly applicable, it demands more than is really needed. We now introduce a more general relation which leads to further comparisons, and also provides another way of looking at our previous results. For convenience, we continue to restrict attention to unit speeds. Also, we only spell out the case of deterministic routing; the extension to probabilistic (state-independent) routing is immediate.

For motivation, let $\mathscr{G}_k$ be the GSMS of the single-server, k-capacity queue in which blocked arrivals are lost. If $k < k' < \infty$, then $\mathscr{G}_k \subseteq \mathscr{G}_{k'}$ (map state n to state $n + k' - k$); but Theorem 4.3 does not apply, because $\mathscr{G}_{k'}$ violates (M). On the other hand, $\mathscr{G}_\infty$ satisfies (M), but $\mathscr{G}_k \not\subseteq \mathscr{G}_\infty$ (state k cannot be identified with any state of the infinite-capacity queue). Thus, we do not yet have any basis for comparing the finite- and infinite-capacity systems. But even though $\mathscr{G}_k \not\subseteq \mathscr{G}_\infty$, any event epoch sequence T generated by $\mathscr{G}_k$ can also be generated by $\mathscr{G}_\infty$ (much as in Lemma 4.2), and this is most of what we need to make a comparison (as in Theorem 4.3). To carry this out, we "extract" $\mathscr{G}_k$ from $\mathscr{G}_\infty$:

DEFINITION 8.1. Let $\mathscr{G}$ and $\mathscr{G}^*$ be schemes sharing an event set **A**. Let T and T^* be corresponding event epoch sequences defined on a common Ω. Call $g \colon \Omega \to \Omega$ an *extraction* of $\mathscr{G}$ from $\mathscr{G}^*$ if $T^*(g(\omega)) = T(\omega)$ for all $\omega \in \Omega$.

Suppose that g extracts $\mathscr{S}$ from $\mathscr{S}^*$ and that T^* is increasing in ω. From the diagram

$$\begin{tikzcd}
\omega \arrow[r] \arrow[d] & g(\omega) \arrow[d] \\
T(\omega) & = T^*(g(\omega))
\end{tikzcd}$$

it is clear that if g and T^* are increasing then so is T. For the monotonicity of g, we consider more general orderings.

Let $\leqslant^a$, $\leqslant^b$ be any partial orderings of ω's, and $\leqslant^a_{\text{st}}$, $\leqslant^b_{\text{st}}$ the corresponding stochastic orderings. Let $\leqslant$ continue to denote the componentwise ordering. In general, say that a function f is $(\leqslant^i, \leqslant^j)$*-increasing* if $x \leqslant^i y \Rightarrow f(x) \leqslant^j f(y)$. The following simple result generalizes much of §3 and §4:

THEOREM 8.2. *Let $\mathscr{S}^*$ be a scheme for which T^* can be constructed so that $\omega \leqslant^b \omega' \Rightarrow T^*(\omega) \leqslant T^*(\omega')$.* (i) *Let g be an extraction of $\mathscr{S}$ from $\mathscr{S}^*$. If g is $(\leqslant^a, \leqslant^b)$-increasing, then $\mathscr{P} \leqslant^a_{\text{st}} \mathscr{P}' \Rightarrow T \leqslant_{\text{st}} T'$.* (ii) *Let g^i extract $\mathscr{S}^i$, $i = 1, 2$, from $\mathscr{S}^*$, and let T, T' be induced from $\mathscr{S}$ by $\mathscr{P}, \mathscr{P}'$. If the extractions can be chosen so that $g^1 \leqslant^b g^2$, then $T^1 \leqslant_{\text{st}} T^2$ whenever $\mathscr{P}^1 = \mathscr{P}^2$.*

REMARK. If $\mathscr{S}^*$ satisfies (M), T^* is $(\leqslant, \leqslant)$-increasing. If $\mathscr{S}^2 \subseteq \mathscr{S}^1 = \mathscr{S}^*$, then $\mathscr{S}^i$, $i = 1, 2$, can be extracted from $\mathscr{S}^*$ and $g^1(\omega) = \omega$. In the proof of Lemma 4.2 we essentially construct the required g^2 (take $g^2(\omega) = \omega'$ in the notation used there) and show that $g^2(\omega) \geqslant \omega$. Hence $g^1 \leqslant g^2$, and part (ii) includes Theorem 4.3 as a special case.

What makes Theorem 8.2 significant is that *any* noninterruptive $\mathscr{S}$ can be extracted from some $\mathscr{S}^*$ for which T^* is $(\leqslant, \leqslant)$-increasing and which satisfies (PM). For fixed $\mathbf{A}$, define the *shuffle scheme* $\mathscr{S}^*_{\mathbf{A}}$ by $\mathbf{S}^* = \{0\}$, $\mathbf{A}^* = \mathbf{A}$, $\mathscr{E}^*(0) = \mathbf{A}$, and $p^*(0; 0, \cdot\,) \equiv 1$. (Recall that we are considering only unit speeds.) $\mathscr{S}$ is extracted from $\mathscr{S}^*_{\mathbf{A}}$ by defining $[g(\omega)]_\alpha(1) = T_\alpha(1)(\omega)$ and $[g(\omega)]_\alpha(n) = T_\alpha(n)(\omega) - T_\alpha(n-1)(\omega)$, for all $\alpha \in \mathbf{A}$ and all $n = 1, 2, \ldots$. A shuffle scheme trivially satisfies (PM). (The motivation for this terminology is the following: Any sequence of events in $\mathbf{A}$ is feasible for $\mathscr{S}^*_{\mathbf{A}}$, so in the sense of Ramadge and Wonham [12], the "language generated by $\mathscr{S}^*_{\mathbf{A}}$" (the set of feasible strings) is just the "shuffle" of the languages generated by the individual elements of $\mathbf{A}$.)

While any $\mathscr{S}$ can be extracted from a $\mathscr{S}^*$ with a monotonic T^*, it is not always the case that the extraction, g, is monotonic, particularly in the componentwise ordering. The critical ordering for comparing event sequences via extractions is the *cumulative* ordering on Ω:

$$\omega \preccurlyeq \omega' \quad \text{if and only if} \quad \sum_{i=1}^{n} \omega_\alpha(i) \leqslant \sum_{i=1}^{n} \omega'_\alpha(i) \quad \forall \alpha \in \mathbf{A}\, \forall n = 1, 2, \ldots.$$

Clearly, $\omega \leqslant \omega' \Rightarrow \omega \preccurlyeq \omega'$, but not vice-versa. If $T = T^* \circ g$ where T^* is a shuffle event sequence, then in order that T be $(\leqslant, \leqslant)$-increasing it is necessary and sufficient that g be $(\leqslant, \preccurlyeq)$-increasing. This is a trivial consequence of the fact that $T^*_\alpha(n)(\omega) = \Sigma^n_{i=1}\omega_\alpha(i)$. If $\mathscr{S}$ satisfies (M), it can be extracted from its shuffle by a g which is $(\leqslant, \preccurlyeq)$-increasing. In this sense, part (i) of Theorem 8.2 generalizes Theorem 3.3.

Consider, again, the k-capacity queue, $\mathscr{S}_k$, with which we began this section. While it is not a subscheme of $\mathscr{S}_\infty$, it can be extracted from $\mathscr{S}_\infty$. Imagine starting both systems in the same state, and prolonging service times in $\mathscr{G}_\infty$ whenever the k-capacity queue is idle but the infinite-capacity queue is not. $\mathscr{G}_k$ could be extracted from the shuffle scheme $\mathscr{G}^*_{\{\alpha,\beta\}}$ (α = arrival, β = departure) in essentially the same way. We know that in either case the extraction g_k cannot be $(\leqslant, \leqslant)$-increasing; see Example 3.7. Write $\omega \leqslant^\beta \omega'$ if $\omega_\beta \leqslant \omega'_\beta$ and $\omega_\alpha = \omega'_\alpha$. Then g_k is not $(\leqslant^\beta, \leqslant)$-increasing but it is $(\leqslant^\beta, \preccurlyeq)$-increasing, which allows us to conclude that speeding up service times ($\mathscr{P}^2_\beta \leqslant_{\text{st}} \mathscr{P}^1_\beta$) while fixing arrivals ($\mathscr{P}^2_\alpha = \mathscr{P}^1_\alpha$) never delays any events ($T^2 \leqslant_{\text{st}} T^1$). Furthermore, if $k' > k$ then $g_{k'} \preccurlyeq g_k$ (though in general $g_{k'} \nleqslant g_k$) so just adding capacity never delays events either. (Whitt [24, Theorem 12(c)] is similar, but relaxes $\mathscr{P}^2_\alpha = \mathscr{P}^1_\alpha$ through an ordering not considered here.) In much the same fashion, we may extract a network of queues in tandem in which the first queue is finite from one in which the first queue is infinite to extend the comparison of Example 3.6. But detailed verification that the g_k's (and the analogs for queues in tandem) have the stated monotonicity properties is quite involved and may be just as complicated as establishing monotonicity of event sequences directly, as in [18, 19, 22]. For this reason, Theorem 8.2 is not as immediately applicable as condition (M) and its generalizations.

9. Clock multiplicity.

The GSMP framework has the shortcoming that its notion of event does not always coincide with the physically interesting "events." In some cases, the GSMP events are too narrow, differentiating between classes of transitions that have similar meaning; in other cases, they can be too broad, failing to distinguish between different ways in which the same event can occur. An example of the first type of problem is the departure of jobs from a multiple-server queue. In the GSMP framework, departures from different servers are ordinarily associated with different events; but this distinction is artificial if the servers are identical. We are likely to be more interested in the total departure process than in departures from specific servers. An example of the second type of problem is the arrival of jobs to a finite-capacity queue with loss blocking. Typically, we would like to distinguish between admitted and blocked arrivals, though these represent the same GSMP event. In this section, we address a special case of the first of these issues; the second will be touched on in §10.

The usual GSMP framework can be modified to allow several clocks to run simultaneously for the same event. When any one of these clocks runs out, the event occurs and triggers a transition. After the transition, the other clocks associated with the event may continue to run (eventually leading to another occurrence of the event), and yet more clocks may be set for that same event.

Let $\underline{\mathscr{E}}(s)$ be an m-dimensional vector ($m = |\mathbf{A}|$), the *ith* component of which is the number of clocks for α_i in state s. Also let $\underline{\mathscr{E}}_\alpha(s)$ be the number of α-clocks, and suppose that this is always finite. The *nth* time a clock is set for event α, it is set to $X_\alpha(n)$, as before, except that now clocks set to $X_\alpha(n)$ and $X_\alpha(n')$, $n' \neq n$, may run simultaneously. Clocks, events and transition probabilities determine the state transitions just as before. The generalization of noninterruption is $p(s'; s, \alpha_i) > 0 \Rightarrow \underline{\mathscr{E}}(s) - e_i \leqslant \underline{\mathscr{E}}(s')$. The generalization of (M) is

($\mathrm{M}^\times$). *Monotonicity Condition with Clock Multiplicity.* If σ_1 and σ_2 are feasible in s and $N(\sigma_1) \leqslant N(\sigma_2)$, then

$$\underline{\mathscr{E}}(\phi(s,\sigma_1)) - [N(\sigma_2) - N(\sigma_1)] \leqslant \underline{\mathscr{E}}(\phi(s,\sigma_2)).$$

The analogous extension ($\mathrm{PM}^\times$) of (PM) is obtained by requiring that the routing be state-independent and that ($\mathrm{M}^\times$) hold for every ϕ_π. For GSMS with clock multiplicity, change (ii) of Definition 4.1 (the definition of subscheme) to require $\underline{\mathscr{E}}^S(s) \leqslant \underline{\mathscr{E}}^B(s)$.

THEOREM 9.1. (i) *For a GSMS with clock multiplicity satisfying* (PM$^\times$), $\mathscr{P}^2 \leqslant_{\mathrm{st}} \mathscr{P}^1 \Rightarrow T^2 \leqslant_{\mathrm{st}} T^1$. (ii) *Suppose* $\mathscr{S}^S \subseteq \mathscr{S}^B$ *and* $\mathbf{A}^S = \mathbf{A}^B$, *where* $\mathscr{S}^B$ *satisfies* (PM$^\times$) *and* $\mathscr{S}^S$ *is noninterruptive. If* $\mathscr{P}^B = \mathscr{P}^S$, *then* $T^B \leqslant_{\mathrm{st}} T^S$.

The proofs of the two parts of Theorem 9.1 are essentially the same as those of Theorems 3.3 and 4.3, and are therefore omitted. (In the proof of Lemma 3.4, it is no longer true that $T_\alpha(n) = V_\alpha(n) + X_\alpha(n)$, where $V_\alpha(n)$ is the epoch of the *nth* setting of a clock for α, but T is still an increasing function of V and X.)

Consider, again, multiple servers sharing a single infinite capacity queue. Suppose there is a single stream of arrivals, α, and that service completions at the various servers all constitute the same event, β. Thus, all servers draw service times from the same stream $\{X_\beta(n),\ n = 1, 2, \ldots\}$. Setting up the system this way appropriately forces the servers to be indistinguishable while making no assumption that the service times are independent or identically distributed. In this approach, $X_\beta(n)$ is more accurately called the service requirement of the *nth* job than the *nth* service time.

With this set-up, the multiple-server queue satisfies (M$^\times$); without clock multiplicity, it would violate (M): Let α be arrival and β^i service completion at the *ith* server. Denote a typical state by $(n, \{i_1, \ldots, i_r\})$ where n is the number of jobs present and the i_j's are the indices of the $r \leqslant n$ busy servers. Suppose that jobs arriving to find more than one available server go to the one with the smallest index (other policies run into similar difficulties). Then $\mathscr{E}(\phi((1, \{1\}), \alpha\beta^1)) = \{\alpha, \beta^2\} \neq \{\alpha, \beta^1\} = \mathscr{E}(\phi((1, \{1\}), \beta^1\alpha))$. But if, instead, we use r clocks for a single event β when r servers are busy, we may take the state to be just the number of jobs present. Then, for example, $\underline{\mathscr{E}}_\beta(\phi(n, \alpha\beta)) = \underline{\mathscr{E}}_\beta(\phi(n, \beta\alpha))$ and (M$^\times$) is, indeed, satisfied. We may therefore conclude that decreasing service requirements or interarrival times, or adding servers speeds up departures. (Whitt [24, Theorem 12(b)] is similar; it permits a slightly weaker ordering of interarrival times—our $\preccurlyeq$, essentially—but assumes a fixed number of servers.) If we used different events for different servers, we would not be able to draw these conclusions. In fact, it is generally *not* true that the departures from individual servers are monotonic in the service times or in the number of servers.

Using clock multiplicity, Examples 3.6 and 5.4 generalize to networks of multiple-server queues. If l_i is the number of servers at i, then under conditions (7)–(9) and the additional condition

$$P^c_{ji} > 0 \quad \text{and} \quad P^{c'}_{j'i} > 0 \quad \text{for some} \quad c' \neq c \quad \text{and any} \quad j, j' \Rightarrow l_i = 1,$$

adding servers and decreasing service requirements and interarrival times speeds up all service completions. This additional condition says that any queue visited by more than one class of jobs has only one server. For the special case of queues in tandem, to have multiple-server queues we can only allow a single class of jobs. If, to satisfy (7), the first queue has infinite capacity, then adding servers increases the throughput of the line. Tsoucas and Walrand [22] show this directly, with no restrictions on the first queue, but under the assumption of independent service requirements. (Their proof uses the independence assumption. It does not require $k_1 = \infty$ because they do not allow changes in interarrival times.)

10. Induced events. We now briefly address the second shortcoming of GSMP events described at the beginning of the last section. Consider, first, the epochs of admitted arrivals to a single-server, k-capacity queue. If α is the arrival event, then

$T_\alpha(n)$ is the epoch of the *nth* arrival, regardless of whether that arrival is blocked (and lost) or actually admitted. But in comparing the performance of two systems, we are likely to be more interested in ensuring that the epochs of admitted arrivals occur earlier, not that all arrivals occur earlier.

Denote service completion by β, and let $\tilde{T}_\alpha(n)$ be the epoch of the *nth* admitted arrival. Suppose that when an arrival and a departure occur simultaneously the arrival is admitted, and that the server is included in the capacity k. Then (taking $T_\beta(n) = 0$ for $n \leqslant 0$)

$$\tilde{T}_\alpha(n) = \inf\{T_\alpha(i), i = 1, 2, \ldots, : T_\alpha(i) \geqslant T_\beta(n-k)\}.$$

Thus, for fixed T_α, $\tilde{T}_\alpha$ is increasing in T_β and decreasing in k. Since we know from §8 that, with fixed ω_α, T_β is increasing in ω_β and decreasing in k, we may conclude that the same is true of $\tilde{T}_\alpha$. In general, whenever the epochs of such "induced events" are increasing functions of the epochs of ordinary GSMP events, they inherit monotonicity from T.

Another example is the network of M queues in tandem in which the *ith* queue has l_i servers, buffer size k_i, $i = 1, \ldots, M$, and $k_1 = \infty$. (The buffer size k_i does not include the l_i service positions.) Let β_0 be arrival to the system, and let β_i be service completion at any of the servers of queue i. As in [22], consider the epochs of arrivals to the various queues. The GSMP event β_{i-1} does not distinguish between jobs that are blocked after completing service at $i - 1$, and jobs that actually move on to i. An arrival to $i = 2, \ldots, M$ may be triggered by β_{i-1}, or by β_{i+j} if queues $i - 1, \ldots, i + j - 1$ are blocked. Let $\tilde{T}_{\beta_i}(n)$, $i = 2, \ldots, M$, be the epochs of actual departures from i, and for convenience let $\tilde{T}_{\beta_i}(n) = 0$ for $n \leqslant 0$. Then, for all $n = 1, 2, \ldots$,

$$\tilde{T}_{\beta_M}(n) = T_{\beta_M}(n),$$

$$\tilde{T}_{\beta_i}(n) = \max\left(T_{\beta_i}(n), \tilde{T}_{\beta_{i+1}}(n - k_{i+1} - l_{i+1})\right).$$

Hence, $\tilde{T}_{\{\beta_1, \ldots, \beta_M\}}$ is increasing in $T_{\{\beta_1, \ldots, \beta_M\}}$ and inherits monotonicity in $\mathscr{P}$, $(k_2, \ldots, k_M)$ and $(l_1, \ldots, l_M)$.

11. Concluding remarks. We have developed a general approach to studying monotonicity in a GSMP through properties of its scheme. When applied to queueing systems, this approach unifies many existing results previously obtained by *ad hoc* methods, and in some cases leads to new results.

To extend this approach to other areas—e.g., *second-order* properties, such as convexity and submodularity—it is necessary to look more deeply into the algebraic structure of schemes. Proceeding in this direction, it is helpful to view the set of feasible strings as a formal *language* that characterizes the "legal behavior" of a system. With each string in the language one can associate a vector, each component counting the number of occurrences of an event (the *score*). Through this association, the language gives rise to a *score space*, the set of vectors of feasible scores. A condition such as (M) can then be phrased as a statement about the structural properties of the language and the associated score space.

In subsequent work [5, 6], we investigate the algebraic structure of the language and the geometry of the score space, which lead to second-order properties of a GSMP. In [6], we also pursue a variety of applications in such areas as simulation variance reduction, derivative estimation, optimal control, and stochastic Petri nets.

Acknowledgements. This research was undertaken while Paul Glasserman was with, and David Yao was visiting, the Operations Research Department of AT & T Bell Laboratories. David Yao's work was also partially supported by NSF grants ECS-88-03183 and ECS-89-96201.

References

[1] Burman, D. Y. (1981). Insensitivity in Queueing Systems. *Adv. Appl. Probab.* **13** 846–859.

[2] Daley, D. J. (1968). Stochastically Monotone Markov Chains. *Z. Wahrsch. Verw. Gebiete.* **10** 305–317.

[3] Glasserman, P. (1988). Equivalence Methods in the Perturbation Analysis of Queueing Networks. Ph.D. Thesis, Division of Applied Sciences, Harvard University.

[→ _____ (1991). Structural Conditions for Perturbation Analysis Derivative Estimation: Finite-Time Performance Indices. *Oper. Res.* **39** 724–738.

[5] _____ and Yao, D. D. (1992). Generalized Semi-Markov Processes: Antimatroid Structure and Second-Order Properties. *Math. Oper. Res.* (to appear).

[6] _____ and _____ (1991). Algebraic Structure of Some Stochastic Discrete-Event Systems, with Applications. *J. Discrete Event Dynamic Systems: Theory and Appl.* **1** 7–36.

[7] Glynn, P. W. (1989). A GSMP Formalism for Discrete Event Systems. *Proc. IEEE* **77** 14–23.

[8] Haas, P. J. and Shedler, G. S. (1987). Regenerative Generalized Semi-Markov Processes. *Stochastic Models* **3** 409–438.

[9] Helm, W. E. and Schassberger, R. (1982). Insensitive Generalized Semi-Markov Schemes with Point Process Input. *Math. Oper. Res.* **7** 129–138.

[10] Kamae, T., Krengel, U. and O'Brien, G. L. (1977). Stochastic Inequalities on Partially Ordered Spaces. *Ann. Probab.* **5** 899–912.

[11] Massey, W. A. (1987). Stochastic Orderings for Markov Processes. *Math. Oper. Res.* **12** 350–367.

[12] Ramadge, P. J. and Wonham, W. M. (1987). Supervisory Control of a Class of Discrete-Event Processes. *SIAM J. Control Optim.* **25** 206–230.

[13] Schassberger, R. (1976). On the Equilibrium Distribution of a Class of Finite-State Generalized Semi-Markov Processes. *Math. Oper. Res.* **1** 395–406.

[14] _____ (1978). Insensitivity of Steady-State Distributions of Generalized Semi-Markov Processes. Part I. *Ann. Probab.* **5** 87–99.

[15] _____ (1978). Insensitivity of Steady-State Distributions of Generalized Semi-Markov Processes with Speeds. *Adv. Appl. Probab.* **10** 836–851.

[1→ Shanthikumar, J. G. and Yao, D. D. (1987). Stochastic Monotonicity of the Queue Lengths in Closed Queueing Networks. *Oper. Res.* **35** 583–588.

[17] _____ and _____ (1989). Stochastic Monotonicity in General Queueing Networks. *J. Appl. Probab.* **26** 413–417.

[18] Sonderman, D. (1979). Comparing Multi-Server Queues with Finite Waiting Rooms. I. Same Number of Servers. *Adv. Appl. Probab.* **11** 439–447.

[19] _____ (1979). Comparing Multi-Server Queues with Finite Waiting Rooms. II. Different Number of Servers. *Adv. Appl. Probab.* **11** 448–455.

[20] _____ (1980). Comparing Semi-Markov Processes. *Math. Oper. Res.* **5** 110–119.

[21] Stoyan, D. (1983). *Comparison Methods for Queues and Other Stochastic Models.* D. J. Daley (Ed.), Wiley, New York.

[22] Tsoucas, P. and Walrand, J. (1989). Monotonicity of Throughput in Non-Markovian Networks. *J. Appl. Probab.* **26** 134–141.

[23] Whitt, W. (1980). Continuity of Generalized Semi-Markov Processes. *Math. Oper. Res.* **5** 494–501.

[24] _____ (1981). Comparing Point Processes and Queues. *Adv. Appl. Probab.* **13** 207–220.

[25] _____ (1986). Stochastic Comparisons for Non-Markov Processes. *Math. Oper. Res.* **11** 608–618.

GLASSERMAN: GRADUATE SCHOOL OF BUSINESS, COLUMBIA UNIVERSITY, 403 URIS HALL, NEW YORK, NEW YORK 10027

YAO: IE / OR DEPARTMENT, COLUMBIA UNIVERSITY, NEW YORK, NEW YORK 10027