CHARACTERIZATION AND OPTIMIZATION OF ACHIEVABLE PERFORMANCE IN GENERAL QUEUEING SYSTEMS

A. FEDERGRUEN
Columbia University, New York, New York

H. GROENEVELT
University of Rochester, Rochester, New York

(Received April 1986; revision received February 1987; accepted July 1987)

This paper considers general (single facility) queueing systems with exponential service times, dealing with a finite number J of distinct customer classes. Performance of the system, as measured by the vector of steady state expected sojourn times of the customer classes (the *performance vector*) may be controlled by adopting an appropriate *preemptive* priority discipline. We show that the performance space, the set of performance vectors which are achievable under some preemptive work conserving rule, is a polyhedron described by $2^J - 1$ (in)equalities. The special structure of this polyhedron nevertheless allows for efficient procedures to minimize any separable convex function of the performance vector. Linear objectives are shown to be minimized by absolute priority rules, thus generalizing a well known result for M/M/1 systems. We also show that each point in the performance space may be achieved by a specific randomization of at most $J + 1$ absolute priority rules.

Queueing models are increasingly used for the analysis and design of complex production and service systems in which different classes of users (or "customers") compete for a limited number of shared resources (or "servers"). It is often possible to classify the customers in a finite number of distinct classes and to apply a specific type of preferential treatment to one class at the expense of others. Such schemes are referred to as priority queueing systems.

Examples include production facilities which manufacture batch orders for a number of distinct products with the same equipment and/or operators. Often, different service level requirements and/or holding cost rates apply to different items, so that significantly different economic consequences result from the delays or sojourn times experienced by the various items. In modern telecommunication systems, heterogeneous data types (e.g., interactive messages, computer outputs, file transfers, facsimile, etc.) compete with voice for the limited availability of shared transmission equipment, e.g., buses in a local area network or frequency bands in a satellite channel. Appropriate priority systems need to be designed to achieve an optimal tradeoff between (the economic consequences of) the delays encountered by the different traffic types. In other systems, the objective is to achieve an *equitable* scheduling procedure of the different customer types for access to the shared resource(s).

When designing such priority systems, it is natural to think in terms of minimizing some cost function with respect to the vector of (average) delays experienced by the different customer classes. Most of the literature on priority queueing systems is concerned with the performance analysis of a specific priority rule in a given queueing model. Surprisingly, little attention has been given to the *design* of queueing disciplines which minimize well stated and realistic cost functions.

Performance of the system, as measured by the vector of steady state expected waiting times of the customer classes (the *performance vector*), may be controlled by adopting an appropriate priority discipline.

(Since in systems with service preemptions a customer may experience several waiting periods, we define this waiting time as the cumulative amount of time spent in the system while not being served.) We consider the class of all preemptive and strongly *work conserving* rules; see Section 1 for a precise definition.

Our main results are the following: we first characterize the *performance space*, the set of achievable performance vectors. The latter is shown to be

Subject classification: Multichannel queues: preemptive priority rules. Queues: optimization.

0030-364X/88/3605-0733 $01.25

described by a simple polyhedron provided the following two conditions are satisfied:

(a) a work conservation law applies;
(b) let $A^*(S)$ denote the long-run expected amount of work in the system when only the collection $S \subset E$ is served; $A^*(\cdot)$ viewed as a set of functions in S, is *supermodular*, i.e., the marginal increase resulting from the addition of a new class of customers to an existing system is at least as large as when the same class is added to a system dealing with only a subset of its current customer base.

We show that the first condition holds in systems with general arrival processes and exponential service times; supermodularity of $A^*(\cdot)$ holds in very general single server systems as well as in many important multiserver models.

Under the above conditions, the performance space is, in fact, a polyhedron. We show that even though this polyhedron needs to be described by $2^J - 1$ constraints, its very special structure allows for efficient algorithms to minimize system wide performance measures expressed as separable convex functions of the performance vector over the performance space. We also show that each vector in the performance space may be achieved by an appropriately constructed randomization of absolute priority rules. In addition, the special structure explains the optimality of absolute priority rules for linear objectives, a result well known for a number of simple queueing models (see Fife 1965, Smith 1956, Kleinrock 1976 and Gelenbe and Mitrani 1980).

Coffman and Mitrani (1980) characterized the performance space of multiclass M/M/1 systems, with preemptions allowed. Gelenbe and Mitrani achieved the same for nonpreemptive M/G/1 systems. Mitrani (1982) characterized the performance space for M/G/1 systems in which the service time of each customer is known upon arrival. (A partial characterization can be found in Kleinrock, Muntz and Hsu 1971.) Federgruen and Groenevelt (1986b) discuss the characterization and control of achievable performance in *nonpreemptive* systems. Results similar to ours are obtained for M/G/c systems.

It is worth pointing out that the above mentioned special polyhedral structure of the performance space consists of it being the base (of the independence polytope) of a so-called *polymatroid* (cf. Edmonds 1970, Welsh 1976), a generalization of the more familiar *matroids*.

Section 1 defines the class of work conserving rules and derives a conservation law. The performance space is characterized in Section 2. Section 3 describes general classes of queueing systems in which the crucial supermodularity property of A^* is satisfied. Efficient optimization methods for system wide performance measures are described in Section 4.

1. Work Conserving Rules and a Conservation Law

We consider general queueing models with one or several identical servers. The service times of the customers in a given class $j \in \{1, \ldots, J\}$ are assumed to be independent and exponentially distributed with parameter μ_j. A customer with service time V is viewed as consisting of V work units. When a customer arrives, only his class is known but not the actual service time. Throughout this paper, we restrict ourselves to the class $\mathscr{R}$ of work conserving priority rules defined as follows.

Definition. A priority discipline is *work conserving* if

(a) no server is free when a customer is in the queue;
(b) the discipline does not affect the amount of service time given to a customer or the arrival time of any customer;
(c) priorities are assigned on the basis of the history of the process, and the time elapsed since the last epoch at which the system became empty.

Conditions (a) and (b) are standard, see e.g., Heyman and Sobel (1982, p. 418). Condition (c) is similar to one stated in Gelenbe and Mitrani and appears to be the most general, easily describable restriction under which the existence of long-run averages of waiting times may be verified, i.e., under which the performance vector is properly defined.

A work conservation law describes an identity satisfied by any (achievable) performance vector associated with a work conserving rule. As pointed out in the Introduction, such a law provides a key tool in the characterization of the performance space. For a given priority rule, let

W_{nj} = waiting time of the nth customer of class j ($j = 1, \ldots, J$, $n \geq 1$);
$A(t)$ = work in the system at time t ($t \geq 0$);
K_n^j = the number of times the nth customer in class j is preempted from service ($n \geq 1$; $j = 1, \ldots, J$);
l_{nk}^j = the length of the kth service interruption of the nth customer in class j (k, $n \geq 1$; $j = 1, \ldots, J$);

v_{nk}^j = the remaining work required by the nth customer in class j at the beginning of the kth service interruption ($k, n \geq 1; j = 1, \ldots, J$);

S_{nj} = the total amount of time spent in the waiting room by the work units of the nth customer in class j after the service process is initiated = $\sum_{k=1}^{K_n^j} v_{nk}^j l_{nk}^j$ ($n \geq 1, j = 1, \ldots, J$).

D_{nj} = the initial delay experienced by the nth customer in class j.

V_{nj} = the service time of the nth customer in class j.

The following conservation law and its proof are similar to those of Heyman and Sobel (Theorem 11–14). The first proof of this type was given by Schrage (1970) for G/G/1 queues; see Heyman and Sobel for a review of the literature on conservation laws.

Lemma 1. (Conservation Law for Preemptive Systems). *Consider a c-server system and a given work conserving rule. For each class j, assume the long-run average arrival rate λ_j exists and let $\rho_j = \lambda_j/\mu_j$. When $c > 1$, assume $\mu_j = \mu$ for all $j = 1, \ldots, J$. Suppose, in addition, that*

$$\mathrm{E}\left\{\lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} D_{nj}\right\} = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} ED_{nj} \stackrel{\text{def}}{=} D_j^* \qquad (j = 1, \ldots, J),$$

and

$$\mathrm{E}\left\{\lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} S_{nj}\right\} = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} ES_{nj} \stackrel{\text{def}}{=} S_j^* \qquad (j = 1, \ldots, J),$$

Then

(a) *A^*, the long-run average work in the system, exists and is independent of the priority rule*

$$\mathrm{E}\left\{\lim_{T\to\infty} \frac{1}{T} \int_0^T A(t)\, dt\right\} = \mathrm{E}\left\{\lim_{T\to\infty} \frac{1}{T} \int_0^T A_{\mathrm{FIFO}}(t)\, dt\right\} = A^*;$$

and

(b) $\lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} EW_{nj} \stackrel{\text{def}}{=} W_j^*$

exists for all $j \in E$ and satisfies

$$\sum_{j=1}^{J} \rho_j W_j^* = A^*. \qquad (1)$$

Proof. Note that

$$\begin{aligned} ES_{nj} &= \sum_{k=1}^{\infty} \mathrm{Prob}\{v_{nk}^j > 0\} \\ &\quad \cdot \mathrm{E}[v_{nk}^j \mid {}^j_{nk} > 0] \cdot \mathrm{E}[l_{nk}^j \mid l_{nk}^j > 0] \\ &= \mu_j^{-1} \cdot \sum_{k=1}^{\infty} \mathrm{Prob}\{v_{nk}^j > 0\} \cdot \mathrm{E}[l_{nk}^j \mid v_{nk}^j > 0] \\ &= \mu_j^{-1} \cdot \sum_{k=1}^{\infty} \mathrm{E}[l_{nk}^j \cdot 1_{\{k \leq K_n^j\}}] \\ &= \mu_j^{-1} \cdot \mathrm{E}\left[\sum_{k=1}^{\infty} l_{nk}^j \cdot 1_{\{k \leq K_n^j\}}\right] = \mu_j^{-1} \mathrm{E}\left[\sum_{k=1}^{K_n^j} l_{nk}^j\right]. \end{aligned} \qquad (2)$$

(The first equality follows from the properties of work conserving priority rules and the exponentiality of the service time distributions; the next to the last equality follows from the monotone convergence theorem.) Thus,

$$S_j^* = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} ES_{nj} = \mu_j^{-1} \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \mathrm{E}\left[\sum_{k=1}^{K_n^j} l_{nk}^j\right]. \qquad (3)$$

An application of the well known $H = \lambda G$ identity (see e.g., (11-51) in Heyman and Sobel or Heyman and Stidham 1980) as in the proof of Heyman and Sobel's conservation laws (Theorems 11–13 and 11–14) establishes part (a) as well as the identity $A^* = \sum_{j=1}^{J} [\rho_j D_j^* + \lambda_j S_j^* + \lambda_j/\mu_j^2]$. Substitution of (2), the definitions of D_j^* and S_j^*, and the identity $W_{nj} = D_{nj} + \sum_{k=1}^{K_n^j} l_{nk}^j + V_{nj}$ establish $\rho_j W_j^* = \rho_j D_j^* + \lambda_j S_j^* + \lambda_j/\mu_j^2$ and hence part (b).

The assumed existence of long-run averages for the expectation of the quantities $\{D_{nj}\}$ and $\{S_{nj}\}$ is easily verified in models in which all customers arrive according to independent (stationary) renewal processes, cf. Whitt (1982) and Wolff (1984). (Verification may be more tedious in other models.)

2. Characterization of the Performance Space

The conservation law permits us to derive necessary conditions for achievability of a performance vector under the conditions of Lemma 1. Let

$A^*(S)$ = the long-run average work in the system for customers in the collection of classes S when given absolute preemptive priority above customers in other classes (= the long-run average work in system when admitting only customers in the collection of classes S), $S \subset E$.

$W_j^*(S)$ = the long-run average sojourn time for customers in class $j \in S$ in a FIFO system admitting only customers in the collection of classes S, $S \subset E$.

Theorem 1 (Necessary Conditions for Achievability). *If under the conditions of Lemma* 1, *a vector* W *represents an achievable performance vector corresponding with a (preemptive) work conserving rule* R, *then*

$$\sum_{j\in S} \rho_j W_j \geq \sum_{j\in S} \rho_j W_j^*(S) = A^*(S), \quad S \subset E, \tag{4}$$

and

$$\sum_{j=1}^{J} \rho_j W_j = \sum_{j=1}^{J} \rho_j W_j^*(E) = A^*(E). \tag{5}$$

In addition, each of the lower bounds in (4) *is tight.*

Proof. The proof of Theorem 11–14 in Heyman and Sobel shows that

$$\sum_{j\in S} \rho_j W_j = \tilde{A}_R(S), \quad S \subset E,$$

where $\tilde{A}_R(S)$ is the long-run average work in the system corresponding to all customers in any of the classes of S under rule R. Clearly $\tilde{A}_R(S) \geq A^*(S)$ with equality, e.g., for any rule assigning absolute (preemptive) priority to customers in S above customers in other classes while breaking ties in accordance with FIFO. This proves (4) and the fact that the lower bounds in (4) are tight. Equation 5 follows from Lemma 1.

Absolute priority rules rank the classes in a given sequence and determine priorities on the basis of class ranks only (breaking ties according to FIFO).

Corollary 1 (Characterization of Absolute Priority Rules). *Under the conditions of Lemma* 1, *consider an absolute preemptive priority rule* R. *Assume (without loss of generality) that the classes are numbered in descending sequence of their priorities (i.e., class* 1 *has top priority and class* J *has lowest priority). The corresponding performance vector is the unique solution to the triangular system of linear equations*

$$\sum_{j=1}^{l} \rho_j W_j = A^*(\{1, \ldots, l\}), \quad l = 1, \ldots, J. \tag{6}$$

Proof. Fix l, $1 \leq l \leq J$. Let $\tilde{A}_R$ be the long-run average work-in-system under rule R for customers in the collection $\{1, \ldots, l\}$. Note that under rule R customers in $\{1, \ldots, l\}$ receive absolute (preemptive) priority above all other customers. Thus $\tilde{A}_R = A^*(\{1, \ldots, l\})$. Finally, it follows from the proof of Theorem 11–14 in Heyman and Sobel that $\sum_{j=1}^{l} \rho_j W_j = \tilde{A}_R$.

Let $\mathscr{W}^* = \{W \in \mathbb{R}^J$: W satisfies (4) and (5)$\}$. Subtracting the inequalities (4) from (5) we obtain the following alternative representation of $\mathscr{W}^*$: Let $b^*(S) = A^*(E) - A(^*(E \setminus S)$. Then

$$\mathscr{W}^* = \left\{ W \in \mathbb{R}^J \text{: } W \text{ satisfies (5) and } \sum_{j\in S} \rho_j W_j \leq b^*(S),\ S \subset E \right\}. \tag{7}$$

Theorem 2 specifies a general assumption under which (4) and (5) represent sufficient (as well as necessary) conditions for the achievability of a performance vector under a preemptive rule. In other words, a general condition is given under which $\mathscr{W}^*$ represents the performance space.

A set function h: $2^E \to \mathbb{R}$ is called *nondecreasing* if $h(T) \leq h(S)$ whenever $T \subset S$, and *supermodular* (*submodular*) if $h(S \cup \{j\}) - h(S) \geq (\leq)\ h(T \cup \{j\}) - h(T)$ for all $T \subset S$ and $j \notin S$. (In other words, a set function is supermodular if the marginal increase resulting from the addition of a new class of customers to an existing system is at least as large as when the same class is added to a system dealing with only a subset of its current customer base.) For a given set function h: $2^E \to \mathbb{R}$, a polyhedron $X = \{x \in \mathbb{R}_+^J$: $\sum_{j\in S} x_j \leq h(S),\ S \subset E\}$ is called (the independence polytope of) a *polymatroid* provided $h(\varnothing) = 0$ and $h(\cdot)$ is nondecreasing and submodular, and $X^* = X \cap \{x \mid \sum_{j=1}^{J} x_j = h(E)\}$ is called the *base* of X.

Let $X \subset \mathbb{R}_+^J$ be the polyhedron described by the inequalities

$$\sum_{j\in S} x_j \leq b^*(S), \quad S \subseteq E, \tag{8}$$

and let

$$X^* = X \cap \left\{ x \in \mathbb{R}_+^J \;\middle|\; \sum_{j\in E} x_j = b^*(E) \right\}. \tag{9}$$

Theorem 2. *Under the conditions of Lemma* 1, *assume* $A^*(S)$ *is supermodular in* $S \subset E$. *Then*

(a) X^* *is the base of a polymatroid.*
(b) $\mathscr{W}^*$ *is the performance space.*

Proof. (a) Clearly, $b^*(\phi) = A^*(E) - A^*(E) = 0$. Monotonicity of $b^*(\cdot)$ is straightforward from its definition. Since $A^*(\cdot)$ is supermodular, we have for $j \notin S$ and $T \subset S$, $b^*(S \cup \{j\}) -$

$b^*(S) = A^*(E\backslash S) - A^*(E\backslash S\backslash\{j\}) \leqslant A^*(E\backslash T) - A^*(E\backslash T\backslash\{j\}) = b^*(T \cup \{j\}) - b^*(T)$. This verifies the submodularity of $b^*(\cdot)$.

Thus, the polyhedron X^* is the base of a polymatroid and the vector $x = (\rho_1 W_1, \ldots, \rho_J W_J)$ corresponding to any achievable performance vector is contained in this base.

(b) It remains to be shown that for all $x \in X^*$, the vector $(x_1/\rho_1, x_2/\rho_2, \ldots, x_J/\rho_J)$ represents an achievable performance vector. It follows from Corollary 1 that the performance vector W of each of the $J!$ absolute priority rules is an extreme point of $\mathscr{W}^*$, and hence, $(\rho_1 W_1, \ldots, \rho_J W_J)$ is an extreme point of X^*.

Conversely, let x^* be an extreme point of X^*. There exists a linear objective $\sum_{j=1}^{J} c_j x_j$ which attains its maximum over X^* in x^*. Moreover, since X^* is the base of a polymatroid it follows from Edmonds that x^* may be constructed with the following greedy procedure: assume (without loss of generality, after possible renumbering) that $c_1 \geqslant c_2 \geqslant \ldots \geqslant c_J$;

Step 0. Set x_1 to its maximum feasible value, i.e., $x_1 := b^*(\{1\})$; $l := 2$;

Step 1. Given fixed values for $x_1, \ldots, x_{l-1}$, set x_l at its maximum feasible value, i.e., $x_l := b^*(\{1, \ldots, l\}) - \sum_{i=1}^{l-1} x_i$.

Step 2. If $l = J$, terminate, otherwise, $l := l + 1$ and return to Step 1.

The resulting vector $W = (x_1^*/\rho_1, \ldots, x_J^*/\rho_J)$ clearly satisfies (6). Since the solution of (6) is unique, it follows that W is the performance vector of an absolute priority rule. Thus, all extreme points of $\mathscr{W}^*$ are performance vectors of absolute priority rules. Since each point in $\mathscr{W}^*$ may be written as a convex combination of extreme points, it is the performance vector of an appropriate randomization of absolute priority rules.

We conclude that under the conditions of Theorem 2 the performance space is a polyhedron described by $2^J - 1$ constraints. It would appear that this large number of constraints precludes the existence of efficient algorithms to optimize linear—let alone nonlinear—system wide performance measures of $(W_1, \ldots, W_J)$. However, since X^* is the base of a polymatroid, efficient algorithms exist, nevertheless, to minimize any separable convex system-performance measure. (Some nonseparable cases can be handled as well, see Federgruen and Groenevelt 1986a.) Section 4 describes two general algorithms as well as efficient implementations for M/M/c systems.

As pointed out in the previous proof, each achievable performance vector in $\mathscr{W}^*$ corresponds to an appropriate randomization of absolute priority rules. In view of Caratheodory's theorem (see e.g., Theorem 2.1.6 in Bazaraa and Shetty 1979), at most $J + 1$ distinct absolute priority rules need to be involved in the randomization; these absolute priority rules and the required randomization probabilities are obtained by solving a linear program; see Bazaraa and Shetty.

Section 3 describes general classes of queueing systems in which the crucial *supermodularity* condition for A^* is satisfied.

3. Supermodularity of the Long-run Average Work in System A^*

Federgruen and Groenevelt (1987) show that the long-run average work in system $A^*(\cdot)$ is supermodular in *general single server* systems. We thus conclude the following.

Corollary 2. *Consider a single server system under the conditions of Lemma* 1. *Assume that the arrival processes of the customer classes are independent of the state of the system, have countably many arrivals on each sample path, but are otherwise arbitrary. X^* is the base of a polymatroid and $\mathscr{W}^*$ is the performance space.*

To date, the only model in which A^* may be evaluated in closed form is the model with Poisson arrivals. Observe that in a single server system where different classes may have different (though exponential) service time distributions, the expected waiting time component in $W_j^*(S)$ is given by the long-run expected waiting time in a (single class) M/G/1 queue with hyperexponential service times. Thus, from the well known Pollaczek-Khintchine formula

$$W_j^*(S) = \left(\sum_{j\in S} \lambda_j/\mu_j^2\right) \cdot \left(1 - \sum_{j\in S} \lambda_j/\mu_j\right)^{-1} + \mu_j^{-1}.$$

Hence, in view of (4),

$$A^*(S) = \sum_{j\in S} \lambda_j/\mu_j^2 + \left(\sum_{j\in S} \lambda_j/\mu_j\right) \cdot \left(\sum_{j\in S} \lambda_j/\mu_j^2\right) \cdot \left(1 - \sum_{j\in S} \lambda_j/\mu_j\right)^{-1}$$

$$= \left(\sum_{j\in S} \lambda_j/\mu_j^2\right) \cdot \left(1 - \sum_{j\in S} \lambda_j/\mu_j\right)^{-1}.$$

In multiserver systems, counterexamples may be constructed where A^* fails to be supermodular, see Federgruen and Groenevelt (1985). (It remains,

however, an open question whether such counterexamples exist when all service times have the *same exponential* distribution.) As a general rule, supermodularity has only been established for systems where all service times are deterministically identical (cf. ibid.). It, therefore, appears that for systems with *exponential* service times, a separate analysis is required for each specific model. To date, the only such model in which A^* may be evaluated exactly (or in which closed form approximation formulas have been derived) is the model with Poisson arrivals. Below we show that in this model A^*, in addition to being supermodular, is a so-called generalized symmetric function. The special (generalized symmetric) structure is exploited in Section 4 to obtain efficient implementations for algorithms optimizing system wide performance measures. We first need the following definition.

Definition 2. *A set function $h(\cdot)$ defined on 2^E is* generalized symmetric *if $h(S) = f(\sum_{i\in S} \alpha_i)$, $S \subset E$, where $\alpha = (\alpha_1, \ldots, \alpha_J)$ is a positive vector.*

Generalized symmetric functions were first introduced in Groenevelt (1985) and Federgruen and Groenevelt (1986a). They generalize *symmetric* set functions where $\alpha \equiv 1$, i.e., $h(S) = f(|S|)$, see e.g., Lawler and Martel (1982) and Topkis (1982). One easily verifies the following lemma.

Lemma 2. *A generalized symmetric set function $h(\cdot)$ with $h(S) = f(\sum_{i\in S} \alpha_i)$ $(S \subset E)$, $f(0) = 0$, and f nondecreasing and concave satisfies the properties*:

(i) $h(\varnothing) = 0$;
(ii) *h is nondecreasing*;
(iii) *h is supermodular.*

In multiserver systems, a work conservation law only applies when *all* customers have *identical* exponential service time distributions; see Lemma 1. Assuming, in addition, that all customer classes arrive according to a Poisson process, $W_j^*(S)(j \in S)$ is given by the expected sojourn time in a standard M/M/c model. Hence, from Gross and Harris (1974, eq. (3.17), p. 99),

$$W_j^*(S) = W^*(S) \overset{\text{def}}{=} \left(1 + \sum_{i=0}^{c-1} \frac{c!\,(c - \rho(S))}{i!\; c\rho(S)^{c-i}}\right)^{-1} \cdot (\mu(c - \rho(S)))^{-1} + \mu^{-1} \quad (j \in S),$$

where $\rho(S) = \sum_{j\in S} \rho_j$. Moreover, from (4)

$$A^*(S) = \rho(S)W^*(S).$$

Theorem 3. *Consider an M/M/c model with J classes of customers arriving according to independent Poisson processes and with all service times exponentially distributed with parameter μ. Then $A^*(S)$ is generalized symmetric and supermodular.*

Proof. Clearly, $A^*(S) = \rho(S)W^*(S)$ is generalized symmetric since it depends on S only through $\rho(S)$. Let $L^*(S)$ be the expected long-run average number of customers in queue. It follows from Little's theorem that $\rho(S)W^*(S) = (1/\mu) \cdot L^*(S)$ which is convex and nondecreasing in $\rho(S)$, as shown by Grassmann (1983); see also Lee and Cohen (1983). Supermodularity of $A^*(\cdot)$ follows from Lemma 2.

4. Efficient Algorithms to Solve the Performance Maximization Problem

Assume that we wish to minimize a system wide performance measure $\sum_j f_j(W_j)$, stated as a separable convex function of the performance vector W. Apply the transformation of variables $x_j = W_j/\rho_j$, and define $g_j(x_j) = -f_j(\rho_j x_j)$ $(j \in E)$. It follows from Theorem 2 that the performance maximization problem is equivalent to:

$$\text{maximize} \quad \sum_{j\in E} g_j(x_j)$$

$$\text{subject to} \quad x \in X^* \tag{10}$$

where, for general single server systems and for the standard M/M/c model, X^* is the base of a polymatroid; see Theorems 2 and 3 and Corollary 2.

Several algorithms to solve problem (10) for *general* polymatroids are available in the literature, and in this section we show how two such algorithms, the Greedy Algorithm (Girlich and Kowaljow 1981, Federgruen and Groenevelt 1986a), and the Decomposition Algorithm (Groenevelt 1985a) may be implemented efficiently when b^* is generalized symmetric (as is the case in models with Poisson arrivals, see Section 3).

The Greedy Algorithm is extremely easy to state and program. Its number of iterations, however, grows linearly with the value of $b^*(E)$ (for fixed J) and may be very large. The algorithm is therefore only pseudopolynomial. The Decomposition Algorithm, on the other hand, is fully polynomial, i.e., its complexity is (largely) independent of the values of the parameters, see below; it is, however, more elaborate and for a fixed value of $b^*(E)$, its complexity increases faster (by an order of magnitude) with the number of customer classes than the complexity of the Greedy

Algorithm. Thus, neither one of the two procedures dominates under all circumstances.

The Greedy Algorithm can be formulated as follows: (Let u^i be the ith unit vector in $\mathbb{R}^J$.)

Greedy Algorithm

{initialization}
  choose a stepsize $\epsilon > 0$;
  set $x \equiv 0$;
  initialize the permutation $(i_1, \ldots, i_J)$ of $\{1, \ldots, J\}$
    so that

$$g_{i_l}(x_{i_l} + \epsilon) - g_{i_l}(x_{i_l}) \geqslant g_{i_k}(x_{i_k} + \epsilon) - g_{i_k}(x_{i_k}), \qquad 1 \leqslant l < k \leqslant J;$$

{main loop}
  $n \leftarrow 0$;
  repeat
    $n \leftarrow n + 1$;
    while $x + \epsilon \cdot u^{i_n} \in X$ **do**
    begin
      $x \leftarrow x + \epsilon \cdot u^{i_n}$;
      update the permutation $(i_1, \ldots, i_J)$
    end;
  until $n = J$;

The Greedy Algorithm is guaranteed to produce an optimal solution to problem (10) if we assume that the values $b^*(S)$ $(S \subset E)$ are integer multiples of ϵ; see Federgruen and Groenevelt (1986a). For models with Poisson arrivals, assuming that all arrival and service rates are rational, this will be the case for a small enough rational ϵ.

Let κ denote the effort (measured in terms of elementary operations) required for a single evaluation of the while-condition in the Greedy Algorithm (membership test). Under the standard assumption that an evaluation of g_j can be performed in constant time, the complexity of the Greedy Algorithm is easily seen to be $O((J + b^*(E)/\epsilon)(\kappa + \log J))$. The next lemma provides the foundation for an efficient implementation of the membership tests for *generalized symmetric* polymatroids, i.e., when $b^*(\cdot)$ is generalized symmetric.

Note that $A^*(\cdot)$ is generalized symmetric if and only if $b^*(\cdot)$ is generalized symmetric. Using Theorem 3, one easily verifies that $b^*(\cdot)$ satisfies the conditions in Lemma 3 for the multiserver model with Poisson arrivals.

Lemma 3 (Membership Test for Generalized Symmetric Polymatroids). *Let $b^*(\cdot)$ be generalized symmetric, i.e., $b^*(S) = f(\sum_{i \in S} \alpha_i)$ for some positive vector $\alpha \in R^J$ and some concave, nondecreasing function $f(\cdot)$ with $f(0) = 0$. Then $x \in R_+^J$ satisfies* (8) *if and only if*

$$\sum_{l=1}^{k} x_{j_l} \leqslant b^*(\{j_1, \ldots, j_k\}) \quad k = 1, \ldots, J \tag{11}$$

holds for some permutation $(j_1, \ldots, j_J)$ of $\{1, \ldots, J\}$ that satisfies

$$x_{j_l}/\alpha_{j_l} \geqslant x_{j_k}/\alpha_{j_k} \quad 1 \leqslant l < k \leqslant J. \tag{12}$$

Proof. See Federgruen and Groenevelt (1986a, Lemma 9).

Hence, for generalized symmetric polymatroids the membership test can be performed in $O(J)$ time once the components of x have been arranged in the proper sequence. Maintaining a second permutation $(j_1, \ldots, j_J)$ satisfying (12) throughout execution of the main loop requires only $O(\log J)$ time per iteration, so the complexity of the entire algorithm is $O((J + b^*(E)/\epsilon)\log J)$.

For single server models with nonidentical mean service times, membership may be tested directly by verifying whether all of the $(2^J - 1)$ constraints $\sum_{i \in S} x_i \leqslant b^*(S)$ $(S \subset E)$ are satisfied. This approach is tractable as long as the number of customer classes is not too large (say $J \leqslant 10$). Alternatively, the *polynomial* membership test in Grötschel, Lovasz and Schrijver (1981) (which applies to general submodular functions) may be employed. (This procedure employs the ellipsoid method and its implementation is therefore somewhat cumbersome. It remains an open question whether a simple combinatorial test, as in Lemma 3, could be applied to the Poisson arrivals model.)

An alternative to the Greedy Algorithm is the Decomposition Algorithm described in Groenevelt (1985a).

Decomposition Algorithm

1. Let y be a solution to the single constraint problem
   maximize $\sum_{j \in E} g_j(x_j)$
   subject to $\sum_{j \in E} x_j = b^*(E)$;
2. Find a maximal element z of
   $X_y = \{x: x \text{ satisfies (8) and } x_j \leqslant y_j, j \in E\}$;
3. Determine $E_1 = \cup \{S \subset E$: (8) is tight w.r.t. z and $S\}$; $E_2 \leftarrow E \backslash E_1$;
4. If $E_1 = E$ then go to Step 8, otherwise continue with Step 5;

5. Use the Decomposition Algorithm recursively to find a solution y^1 of the problem
   maximize $\sum_{j\in E_1} g_j(x_j)$
   subject to $\sum_{j\in S} x_j \leq b^*(S), \quad S \subset E_1$
   $\sum_{j\in E_1} x_j = b^*(E_1)$;
6. Use the Decomposition Algorithm recursively to find a solution y^2 of the problem
   maximize $\sum_{j\in E_2} g_j(x_j)$
   subject to
   $\sum_{j\in S} x_j \leq b^*(S \cup E_1) - b^*(E_1), \quad S \subset E_2$
   $\sum_{j\in E_2} x_j \leq b^*(E) - b^*(E_1)$;
7. Set $y_j \leftarrow y_j^1$ for $j \in E_1$, $y_j \leftarrow y_j^2$ for $j \in E_2$;
8. Stop: y is an optimal solution.

Since E_1 and E_2 are disjoint, it follows that the total number of times that Steps 1 to 4 are executed is at most $2J - 1$. Several alternatives may be available for the solution of the single constraint problem in Step 1: if the functions g_j can be written as

$$g_j(t) = c_j \cdot G\left(\frac{t - d_j}{c_j}\right), \quad j = 1, \ldots, J$$

then $O(J \log J)$ algorithms exist; see Groenevelt, 1985b, Section 4.5. If the functions g_j are differentiable, algorithms in Zipkin (1980) can be used. Finally, the discrete algorithm of Frederickson and Johnson (1982) can be used to find an optimal solution in $O(J \cdot \log(b^*(E)/\epsilon))$, with ϵ chosen small enough as before.

Groenevelt (1985a) shows that Steps 2 and 3 may be implemented with the following procedure.

General Implementation Procedure for Steps 2 and 3 of the Decomposition Algorithm:

2a. $T \leftarrow \varnothing$;
2b. **for** $j \leftarrow 1$ **to** J **do**
begin
find

$$\bar{z} = \min\{b^*(S \cup \{j\}) - \textstyle\sum_{l\in S} z_l : T \subset S \subset \{1, \ldots, j-1\}\} \qquad (13)$$

and let S' be the largest subset for which this minimum is assumed;
if $\bar{z} > y_j$ **then** $z_j \leftarrow y_j$
else begin $z_j \leftarrow \bar{z}$; $T \leftarrow S' \cup \{j\}$ **end**;
end;
3. $E_1 \leftarrow T$;

When $b^*(\cdot)$ is generalized symmetric as is the case in the multi-server model with Poisson arrivals, see Theorem 3, Steps 2 and 3 of the Decomposition Algorithm may in fact be implemented by the following $O(J \cdot \log J)$ procedure, see Groenevelt (1985b, Procedure 1):

Steps 2 and 3 of the Decomposition Algorithm for Generalized Symmetric Polymatroids

2a. Determine a permutation $(j_1, \ldots, j_J)$ of $\{1, \ldots, J\}$ for which
$y_{j_l}/\alpha_{j_l} \geq y_{j_k}/\alpha_{j_k}, \quad 1 \leq l < k \leq J$.
2b. **for** $l \leftarrow 1$ **to** J **do**
if $y_{j_l} < b^*(\{j_1, \ldots, j_l\}) - \sum_{k=1}^{l-1} z_{j_k}$
then $z_{j_l} \leftarrow y_{j_l}$
else begin $k' \leftarrow l$;
$z_{j_l} \leftarrow b^*(\{j_1, \ldots, j_l\}) - \sum_{k=1}^{l-1} z_{j_k}$ **end**;
3. $E_1 \leftarrow \{j_1, \ldots, j_{k'}\}$; $E_2 \leftarrow \{j_{k'+1}, \ldots, j_J\}$.

Since Steps 1 to 4 are executed at most $2J - 1$ times (as explained previously), the total time spent in (the implementation of) Steps 2 and 3 of the Decomposition Algorithm is $O(J^2 \log J)$.

For *single server* models with *nonidentical* mean service times, the minima in the "General Implementation Procedure for Steps 2 and 3" may be computed directly, by evaluating the expression within brackets in (13) for all relevant sets S. This requires at most $2^J - 1$ evaluations of the set function $b^*(\cdot)$ in every execution of Steps 2 and 3, which is tractable as long as the number of customer classes is not too large. Alternatively, for large values of J, a polynomial procedure in Grötschel, Lovasz and Schrijver may be employed. (This procedure is a variant of their above discussed membership test.)

References

BAZARAA, M., AND C. SHETTY. 1979. *Nonlinear Programming Theory and Algorithms*, John Wiley & Sons, New York.

COFFMAN, E., AND I. MITRANI. 1980. A Characterization of Waiting Time Performance by Single Server Queues. *Opns. Res.* **28,** 810–821.

EDMONDS, J. 1970. Submodular Functions, Matroids and Certain Polyhedra. In *Combinatorial Structures and Their Applications*, pp. 69–87. R. Guy et al. (eds.), Gordon and Breach, New York.

FEDERGRUEN, A., AND H. GROENEVELT. 1987. The Impact of the Composition of the Customer Base in General Queueing Models. *J. Appl. Prob.* **24,** 709–724.

FEDERGRUEN, A., AND H. GROENEVELT. 1986a. The Greedy Procedure for Resource Allocation Problems: Necessary and Sufficient Conditions for Optimality. *Opns. Res.* **34,** 909–918.

FEDERGRUEN, A., AND H. GROENEVELT. 1986b. *M/G/c* Queueing Systems with Multiple Customer Classes:

Characterization and Control of Achievable Performance under Nonpreemptive Priority Rules. Columbia University, Graduate School of Business Working Paper, New York. To appear in *Mgt. Sci.*

FIFE, D. 1965. Scheduling with Random Arrivals and Linear Loss Functions. *Mgmt. Sci.* **11,** 429–437.

FREDERICKSON, G. N., AND D. B. JOHNSON. 1982. The Complexity of Selection and Ranking in $X + Y$ Matrices with Sorted Columns. *J. Comput. Syst. Sci.* pp. 197–208.

GELENBE, E., AND I. MITRANI. 1980. *Analysis and Synthesis of Computer Systems*, Academic Press, New York.

GIRLICH, E., AND M. KOWALJOW. 1981. *Nichtlineare Diskrete Optimierung*, Vol. 6. Akademie-Verlag, Berlin.

GRASSMANN, W. 1983. The Convexity of the Mean Queue Size of the $M/M/c$ Queue with Respect to the Traffic Intensity. *J. Appl. Prob.* **20,** 916–919.

GROENEVELT, H. 1985a. Two Algorithms for Maximizing a Separable Concave Function over a Polymatroid Feasible Region. Graduate School of Management Working Paper, University of Rochester, Rochester, N.Y.

GROENEVELT, H. 1985b. Resource Allocation Problems with Decreasing Marginal Returns to Scale. Ph.D. Dissertation, Columbia University, New York.

GROSS, D., AND C. HARRIS. 1974. *Fundamentals of Queueing Theory*, John Wiley & Sons, New York.

GRÖTSCHEL, M., L. LOVASZ AND A. SCHRIJVER. 1981. The Ellipsoid Method and Its Consequences in Combinatorial Optimization. *Combinatorica* **1,** 169–197.

HEYMAN, D., AND M. SOBEL. 1982. *Stochastic Models in Operations Research*, Vol. I. McGraw-Hill, New York.

HEYMAN, D., AND S. STIDHAM. 1980. A Note on the Relation between Customer and Time Averages in Queues. *Opns. Res.* **28,** 943–944.

KLEINROCK, L. 1976. *Queueing Systems*, Vol. 2. John Wiley & Sons, New York.

KLEINROCK, L., R. MUNTZ AND J. HSU. 1971. Tight Bounds on Average Response Time for Processor Having Models of Time-Shared Computer Systems. *Inf. Process.* **71,** TA-2, 50–58.

LAWLER, E. L., AND C. U. MARTEL. 1982. Computing Maximal Polymatroidal Network Flow. *Math. Opns. Res.* **7,** 334–347.

LEE, H., AND M. COHEN. 1983. A Note on the Convexity of Performance Measures of $M/M/c$ Queueing Systems. *J. Appl. Prob.* **20,** 920–923.

MITRANI, I. 1982. On the Delay Functions Achievable by Non-Preemptive Scheduling Strategies in $M/G/1$ Queues. In *Deterministic and Stochastic Scheduling*, M. Dempster et al. (eds.). D. Reidel, Dordrecht, Netherlands.

SCHRAGE, L. 1970. An Alternative Proof of a Conservation Law for the Queue $G/G/1$. *Opns. Res.* **18,** 185–187.

SMITH, W. 1956. Various Optimizers for Single-Stage Production. *Naval Res. Logist. Quart.* **3,** 59–66.

TOPKIS, D. 1982. Adjacency on Polymatroids. *Math. Prog.* **30,** 229–237.

WELSH, D. 1976. *Matroid Theory*, Academic Press, London.

WHITT, W. 1982. Existence of Limiting Distributions in the $GI/G/s$ queue. *Math. Opns. Res.* **7,** 88–94.

WOLFF, R. 1984. Conditions for Finite Ladder Height and Delay Moments. *Opns. Res.* **32,** 909–916.

ZIPKIN, P. H. 1980. Simple Ranking Methods for Allocation of One Resource. *Mgmt. Sci.* **26,** 34–43.