Centrum voor Wiskunde en Informatica

REPORTRAPPORT

Optimization of static traffic allocation policies

M.B. Combé, O.J. Boxma

Department of Operations Research, Statistics, and System Theory

Report BS-R9302 January 1993

P.O. Box 4079, 1009 AB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

Optimization of Static Traffic Allocation Policies

M.B. Combé, O.J. Boxma
CWI
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Abstract

We consider the traffic allocation problem: customers in an arrival process have to be assigned to one of a group of servers. Such problems arise, e.g., in distributed computer systems and flexible manufacturing networks. The aim is to optimize system performance measures, such as mean waiting time of a customer or total number of customers in the system, under a given static allocation policy. Two static policies are considered: probabilistic assignment and allocation according to a fixed pattern. For these two policies general properties as well as optimization aspects are discussed.

1991 Mathematics Subject Classification: 60K25, 68M20.
Keywords & Phrases: static load balancing, probabilistic allocation, pattern allocation, waiting times minimization, MAP/G/1 queue.
Note: The first author was supported by NFI. Part of the second author's research has been supported by the European Grant BRA-QMIPS of CEC DG XIII.

1 INTRODUCTION

In a distributed computer system, a group of users generates tasks for a number of processors. This in contrast to systems in which a single processor provides (global) computer capacity for all users, or systems in which each user is provided with its own local processor, usually with very limited capacity. An analogous processing structure occurs in manufacturing systems. Machines are able to perform, and rapidly switch between, various operations that different jobs require.

In both these situations workload is offered to a number of servers with limited capacity. An operational aspect of such systems is the availability of a protocol which optimally balances the workload over the servers: a *load balancing protocol.* Such a protocol has to optimize performance measures for the systems, such as mean amount of workload, throughput, or mean waiting times of jobs.

An important element of such a protocol is the information it requires to operate. This information can range from total knowledge about the system at any point in time, to only information about some basic characteristics, like arrival rate and service times. In general, the term *dynamic* is used for policies which operate under time dependent information, whereas protocols operating under time independent characteristics of the system are called *static.*

It is clear that the more information is available for making decisions, the better the allocation of workload can be. Dynamic policies in general perform better than static policies. However,

Report BS-R9302
ISSN 0924-0659
CWI
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

static load balancing protocols are of considerable interest. First of all, the situation of total knowledge at all times is unrealistic. From a viewpoint of costs, overhead grows as the amount of information to be exchanged, stored and processed increases. Moreover, dynamic policies are not always that effective: there will always be some kind of time delay between updates of the system's current state, and this time delay can have a considerable effect on the quality of the protocol.

A second reason for studying static allocation policies is that they can be useful tools in the design phase of a computer- or manufacturing network. Static policies can provide performance bounds for dynamically controlled systems; the performance measures under static policies are in general evaluated reasonably fast, whereas dynamic policies are harder to analyse and performance evaluations can only be obtained with time consuming methods.

In this paper we consider the load balancing problem for two *static* allocation protocols for the model of a single Poisson stream of jobs offered to a fixed number of server stations. The problem of allocation of customers to queues is also referred to as the *traffic allocation problem.* The allocation protocols we study are static in the sense described above; only the traffic intensity and the server characteristics are used. We present an overview of the results for these policies and also extend optimization procedures for some models.

In the remainder of this section a brief survey of related literature and an outline of the paper are presented.

Related literature

Several papers have addressed the load balancing problem. Below we refer to two overview papers for the general load balancing problem, then present a more extensive overview of the traffic allocation problem. Wang & Morris[24] present a taxonomy for the current load balancing protocols. They formulate the load balancing problem in its most general form, also discriminating between server initiative protocols, i.e. the servers determine from which input sources they draw their customers, and source initiative protocols, i.e. at the moment of arrival in the system jobs are (irrevocably) routed to one of the servers. Wang & Morris[24] provide numerical comparisons, based on analysis and simulation, of various allocation protocols. An overview of load balancing policies and their performances is also given by Boel & Van Schuppen[3]. They consider the problem from a control point of view, and discuss the question which amount of information is required at the routing points to achieve good system performance. Their paper concentrates on analytically and numerically tractable models.

Two static allocation policies have been proposed for the traffic allocation problem: viz. probabilistic assignment and pattern allocation. With the probabilistic policy each arriving customer is routed to one of the servers with fixed probabilities. Under pattern allocation each arriving customer is routed to a server according to an allocation table.

For probabilistic allocation, Buzen & Chen[6] present an algorithm for determining the allocation which minimizes the mean sojourn time of a customer. Their mathematical programming formulation can easily be extended for various other performance measures and fits into the framework of Ibaraki & Katoh[11] for resource allocation problems (RAP). Optimal probabilistic load balancing has been studied by Jean-Marie[9] for the case of two parallel exponential servers and resequencing.

Numerical comparisons (cf. Wang & Morris[24]) reveal that probabilistic allocation performs considerably less than more dynamic allocation policies. Yum[26] proposed the pattern allocation policy (semi-dynamic deterministic routing), which performs notably better than the probabilistic allocation policy (cf. Agrawala & Tripathi[1] and Yum[26]). The reason for this is that the arrival processes at the servers under the pattern allocation policy are less irregular than under probabilistic allocation. However, constructing *the* optimal allocation pattern is yet an unsolved problem. For the case of two identical exponential servers, Ephremides et al.[8] proved that alternately assigning customers to each queue is optimal, a result which was extended by Ramakrishnan[19] for the model with more than two identical exponential servers. The problem of determining the optimal allocation for the pattern allocation policy has also been addressed by Ramakrishnan[19], who proposed a useful approximation procedure for the case of non-identical exponential servers.

The present paper extends the approximation procedure proposed by Ramakrishnan[19] in several directions, in particular allowing *general* service time distributions. We also give an overview of the results for the two above-mentioned static allocation policies. By comparing both policies from a more theoretical viewpoint than in most previous studies, we develop insights for general allocation problems and clarify some reported, but hitherto unexplained, properties.

The present study is concluded with several remarks concerning two extensions of the traffic allocation problem; the first extension deals with the case in which all server stations receive a ('dedicated') Poisson arrival stream, on top of which an extra arrival stream has to be allocated. The second model considers allocation to multiple server stations.

Outline of the paper

In section 2 a mathematical description of the allocation problem is presented, and the probabilistic allocation policy is discussed. In section 3 we argue that allocation policies which result in more regular arrival processes than the Poisson arrival process are to be preferred to probabilistic allocation. There also the pattern allocation policy is introduced. In section 4 an optimization procedure for pattern allocation is presented. In section 5 the performance measures under both allocation policies are numerically compared for various models. We also compare both policies with a dynamic policy that is expected to outperform most policies for the objective functions we consider. Section 6 discusses the two above-mentioned extensions of dedicated arrival streams and multiple server stations.

2 Probabilistic Allocation

Before studying the probabilistic allocation policy we first present a mathematical description of the traffic allocation problem.

Model description

Customers arrive at a routing point according to a Poisson process with rate Λ. At the instance of arrival, a customer has to be assigned to one of N single servers in parallel. This assignment is irrevocable.

The service time $\mathbf{B}_i$ of a customer that is assigned to server i has general distribution $B_i(\cdot)$, with first and second moment β_i and $\beta_i^{(2)}$ respectively. All service times are independent.

Let P denote an allocation policy and p_i, $i = 1, \ldots, N$, be the fraction of the customers that is routed to server i under policy P.

In our traffic allocation problem, the aim is to minimize

$$\sum_{i=1}^{N} f_i(P) C_i E\mathrm{W}_i(P). \tag{2.1}$$

In (2.1) $E\mathrm{W}_i(P)$ denotes the mean waiting time of a customer assigned to server i when the system is controlled by allocation policy P. For each queue C_i is a cost factor which is independent of the load assigned to server i. The factors $f_i(P)$ are additional, load dependent, weight factors. The objective function can have various interpretations by varying $f_i(\cdot)$ and C_i. For example, if $f_i(P) = p_i$ and $C_i = 1, i = 1, \ldots, N$, then the objective function represents the mean waiting time of an arbitrary customer. Or, with $f_i(P) = \Lambda p_i$ and $C_i = \beta_i$, Little's law shows that the objective is to minimize the mean total amount of work in the queues. Instead of $E\mathrm{W}_i(P)$, also $E\mathrm{R}_i(P)$, the mean sojourn time of a customer assigned to queue i, could have been used in (2.1).

Probabilistic allocation

As described in the introduction, the assignment of an arriving customer to a queue can depend on all kinds of information contained in the history and the present state of the system. In this section we discuss the probabilistic allocation policy, also known as random splitting. Under this policy, a fraction p_i of the arrivals is routed to queue i by assigning a customer, arriving at the routing point, to server station i with probability p_i, $i = 1, \ldots, N$ ($\sum_i p_i = 1$). Moreover, these probabilities p_i are the same for all customers, and do not change in time. Let P_{pr} denote the class of probabilistic allocation policies. This class can be completely described by $P_{pr} = \{p \mid p \in [0,1]^N, \sum_{i=1}^N p_i = 1\}$.

The probabilistic allocation policy is static in the sense that when a customer has to be routed to one of the queues, no information about the history and the present state of the system is used. Under $P \in P_{pr}$, the arrival process at queue i is Poisson with intensity $\lambda_i = p_i \Lambda$, $i = 1, \ldots, N$, and the objective function becomes

$$\sum_{i=1}^{N} f_i(P) C_i E\mathrm{W}_i(P) \quad = \quad \sum_{i=1}^{N} f_i(P) C_i \frac{p_i \Lambda \beta_i^{(2)}}{2(1 - p_i \Lambda \beta_i)}. \tag{2.2}$$

Among the first to study the probabilistic allocation policy were Buzen & Chen[6]. Their aim was to minimize the mean sojourn time of a customer for a model with generally distributed service times at the server stations. They solved the problem using standard mathematical programming techniques.

As an example, we take $f_i(P) = \frac{\lambda_i}{\Lambda}$ in (2.2) and solve the allocation problem. In this case, the objective is to minimize the mean weighted waiting time of a customer or, using Little's law, to minimize a weighted sum of the mean number of waiting customers in the system. To obtain the assignment probabilities $p_i^* = \frac{\lambda_i^*}{\Lambda}$ which minimize this function, the following Mathematical Programming Problem has to be solved

PA1:

$$\min \quad \sum_{i=1}^{N} \lambda_i C_i \frac{\lambda_i \beta_i^{(2)}}{2(1 - \lambda_i \beta_i)} \tag{2.3}$$

$$s.t. \quad \sum_{i=1}^{N} \lambda_i = \Lambda, \tag{2.4}$$

$$0 \le \lambda_i < \frac{1}{\beta_i}, \quad i = 1, \dots, N. \tag{2.5}$$

Note that the objective function in PA1 can be separated in terms $T_i = \lambda_i C_i \frac{\lambda_i \beta_i^{(2)}}{2(1-\lambda_i\beta_i)}$, which are strictly convex functions in λ_i. It can also be verified that PA1 has a feasible solution provided that $\sum_i \frac{1}{\beta_i} > \Lambda$. Here and in the remainder of the present paper it is assumed that such is the case.

Problem PA1 has a special characteristic; the control variables only interact through the linear restriction (2.4). This characteristic is typical for the class of Resource Allocation Problems (RAP), as studied in Ibaraki & Katoh[11]. In their book they also consider a RAP which has almost the same form as PA1, the only difference being that the control variables are allowed to equal the upper bounds.

This special case, to which our traffic allocation problems can be translated, can easily be solved numerically. In appendix A we present an algorithm to solve the traffic allocation problem that has a separable objective function, consisting of strictly convex terms, and that has strict upper bounds on the control variables. The algorithm presented in appendix A strongly depends on the strict convexity of the terms. Due to this property, there is only one local minimum, which consequently has to be the optimal solution for the allocation problem. If the objective function is not separable into strictly convex terms then in general there may exist several local minima for the allocation problem. Moreover, in most situations only approximately optimal allocation probabilities can be obtained. One of the cases in which the property of strictly convex terms may not hold is the traffic allocation problem with a general arrival process, as studied in Tang & van Vliet[23]. Their method involves an algorithm for quadratic programming and provides one of the local minima. They also argue that this local minimum should be close to the global minimum.

Problem PA1 can numerically be solved with this algorithm, but it also allows an analytical solution. To find this solution, we first relax PA1 by dropping the last constraint. Using standard Lagrange-multiplier techniques we obtain, with δ the Lagrange-multiplier, the following first order Kuhn-Tucker constraints:

$$\frac{d}{d\lambda_i}\left\{\lambda_i C_i \frac{\lambda_i \beta_i^{(2)}}{2(1-\lambda_i\beta_i)}\right\} = \delta, \qquad i = 1, \dots, N, \tag{2.6}$$

$$\sum_{i=1}^{N} \lambda_i - \Lambda = 0. \tag{2.7}$$

From (2.6) and remarking that (2.3) concerns a sum of terms T_i which are convex functions of λ_i we find the unique optimal values λ_i^*

$$\lambda_i^* = \frac{1}{\beta_i} - \frac{1}{\beta_i}\left(\sqrt{1 + \frac{2\beta_i\delta}{C_i\beta_i^{(2)}}}\right)^{-1}, \qquad i = 1, \dots, N, \tag{2.8}$$

in which the value of the Lagrange-multiplier δ is determined by the constraint (2.7). The optimal splitting probabilities are given by $p_i^* = \frac{\lambda_i^*}{\Lambda}$, $i = 1, \dots, N$. In (2.8) we see that

$0 \leq \lambda_i^* < \frac{1}{\beta_i}$, $i = 1, \ldots, N$; so the vector λ^* is also the optimal solution of PA1.

This example shows us the structure of our traffic allocation problem. Due to the separability in strictly convex terms, the optimal solution is determined by the derivatives of the terms rather than their values. In (2.6) we notice that in the optimal solution the derivatives of the terms are all equal to δ. The property of strict convexity is induced by a basic characteristic of all GI/G/1 queueing systems: the mean waiting time of a customer behaves like $\frac{1}{1-\rho}$, where ρ is the load of the system.
A second observation follows from (2.7) and (2.8): if $\Lambda > 0$, then $\lambda_i^* > 0$ for all i. This is a direct consequence of the above mentioned properties; in the example $\frac{d}{d\lambda_i}T_i \mid_{\lambda_i=0} = 0$ and $\frac{d}{d\lambda_i}T_i$ is an increasing function in λ_i , hence $\lambda_i^* > 0$, provided that $\Lambda > 0$.
However, for other naturally arising objective functions, such as $\sum_i C_i E\mathbf{W}_i$ or $\sum_i f_i C_i E\mathbf{R}_i$, with $\mathbf{R}_i$ denoting the sojourn time of a customer routed to queue i, the optimal values of some λ_i's (and p_i's) may be equal to zero. For these objective functions $\frac{d}{d\lambda_i}T_i \mid_{\lambda_i=0}$ can be so large, relative to the other queues, that it is advantageous not to assign customers to queue i, but to allocate all arrivals amongst the other queues.
For the latter objective functions the Mathematical Programming Problems have the same structure as PA1. Hence we can use the algorithm presented in appendix A to find the optimal probabilistic allocation. This algorithm first determines the set of queues for which $\lambda_i^* > 0$, and then for this set solves a set of equations of the form of (2.6) and (2.7).

3 Less variable arrival processes

In the previous section we saw that under probabilistic allocation, it can be optimal to assign no customers to some of the queues. This waste of server capacity indicates that probabilistic allocation is somewhat inefficient.
Intuitively one expects that when traffic allocation leads to a more regular arrival process, then the mean waiting times are reduced and consequently also the value of the objective function. However, it is very difficult to prove such statements, except for special cases. A detailed investigation of these issues would not fit into the framework of this paper, and hence we restrict ourselves to presenting some basic results on comparison between queueing systems, along with a special case to support the above mentioned intuition.
In this section we discuss a single server queue with an arrival process that is more regular than the Poisson process, and we argue that for an important class of such arrival processes, the behaviour of the mean waiting time as a function of the load is better than for Poisson arrivals.

General comparisons of GI/G/1 systems are presented in Stoyan[22]. Particularly useful for our purposes is his theorem 5.2.1, which states the following monotonicity property for the waiting times:

Lemma 3.1:
Consider two GI/G/1 queueing systems with identically distributed service times. If for the interarrival times $\mathbf{A}_1$ and $\mathbf{A}_2$, $\mathbf{A}_1 \leq_c \mathbf{A}_2$, then also for the steady state waiting times $\mathbf{W}_1 \leq_c \mathbf{W}_2$.
Here $\leq_c$ denotes the convex stochastic ordering for random variables, and indices 1,2 refer to the two queuing systems. Since $\mathbf{W}_1$ and $\mathbf{W}_2$ are positive random variables, $\mathbf{W}_1 \leq_c \mathbf{W}_2$

implies $E\,\mathbf{W}_1^r \leq E\,\mathbf{W}_2^r$, $r = 1, 2, \dots$.
In particular, if $E\,\mathbf{A}_1 = E\,\mathbf{A}_2$, $\mathbf{A}_1 \leq_c \mathbf{A}_2$ holds in the following two cases:
(i) $\mathbf{A}_1$ is constant (cf. Stoyan[22], example 1.9(a)).
(ii) $\mathbf{A}_1$ is NBUE and $\mathbf{A}_2$ has an exponential distribution.
A stochastic variable $\mathbf{X}$ with distribution function F is New Better than Used in Expectation (NBUE) if $\int_t^\infty (1 - F(x))dx/(1 - F(t)) \leq E\,\mathbf{X}$ for all $t \geq 0$. Note that if $\mathbf{X}$ has an increasing failure rate, $\mathbf{X}$ is NBUE. As examples, Gamma(Λ, y) with $y \geq 1$, Weibull(Λ, y) with $y \geq 1$ and Uniformly distributed random variables are NBUE (cf. Stoyan[22] chapter 1). The Gamma(Λ, y) case is now discussed in more detail, as it plays and important role in the remainder of the paper.

Case 3.1:
Consider a Gamma(λ, y)/M/1 queueing model with $y > 1$, so that the arrival process has a coefficient of variation which is smaller than that of a Poisson process with arrival rate $\frac{\lambda}{y}$. Let μ be the service rate and $\frac{\lambda}{\mu y} < 1$, i.e. the queue is stable. In this queue the mean waiting time of a customer $E\mathbf{W}^G$ is given by $\frac{\omega}{\mu(1-\omega)}$, with $\omega = Pr\{\mathbf{W}^G > 0\}$ the smallest positive real solution of $x = \alpha(\mu(1 - x))$, $\alpha(\cdot)$ being the Laplace-Stieltjes Transform of the arrival process (cf. Cohen[7]). For Gamma(λ, y) we have $\alpha(x) = \left(\frac{\lambda}{\lambda + x}\right)^y$. Firstly, for this queueing model $E\mathbf{W}^G < E\mathbf{W}^M$, which follows from lemma 3.1. Here $E\mathbf{W}^M$ denotes the mean waiting time in the M/M/1 queue with arrival rate $\frac{\lambda}{y}$ and service rate μ.
Secondly, it is readily verified that $\frac{d}{d\lambda}E\mathbf{W}^G \downarrow 0$ as $\lambda \downarrow 0$.
The consequence of these two properties for the model with N Gamma(λ_i, y_i)/M/1 queues, $y_i > 1$, $i = 1, \dots, N$ is: if one has to assign intensities λ_i such that the overall arrival rate $\sum_i \frac{\lambda_i}{y_i} = \Lambda$ while the objective is to minimize $\sum_i E\mathbf{W}_i^G$, then $\lambda_i > 0$ for all i. Moreover, the value of the objective function will be lower than if a Poisson Λ arrival stream had been allocated probabilistically to the N server stations.
For the Gamma(Λ, y)/M/1 queueing model we also find that $\frac{d}{dy}E\mathbf{W}^G \downarrow 0$ as $y \to \infty$. As a consequence, suppose that for the model with N Gamma(Λ, y_i)/M/1 queues one has to assign y_i's such that $\sum_i \frac{1}{y_i} = 1$, i.e., the sum of the arrival rates at the queues is Λ. Then $\frac{1}{y_i} > 0$ for all i. This special queueing model is used in the next section as an approximation for a queue with a special non-renewal arrival process.

To summarize, we conclude that the value of the objective function will be lowered if in traffic allocation the allocation policy results in a more regular arrival process than the Poisson process. A side benefit of such an allocation policy could be less permanently idle servers than under probabilistic allocation.

Pattern allocation - the MAP/G/1 queue
Next we introduce a traffic allocation policy which allocates the Poisson arrival stream such that the arrival processes at the queues are less variable than under probabilistic assignment, but which still is static in the sense that no state information of the queues is used and that the allocations are time independent. This policy is pattern allocation. Pattern allocation uses an infinite string of integers $\{a_1, a_2, \dots, a_{n-1}, a_n, a_{n+1}, \dots\}$, where a_n denotes the number of the queue to which the n-th customer in the arrival process is routed. For practical reasons it

is assumed that this string contains a sub-pattern S of finite length M which is repeated over and over. Thus $a_i = a_{i+kM}$ for all $i = 1, \ldots, M$ and $k = 1, 2, \ldots$. Like for the probabilistic allocation policy we can completely describe P_{pa}, the class of pattern allocations. This is done by $P_{pa} = \{a \mid a \in [1..N]^k, k = 1, 2, \ldots\}$.
Let A_{i_n} be the time between the n-th and $n+1$-st arrival at queue i. Under pattern allocation, the distributions of A_{i_n} form a repeated sequence of Erlang distributions. For example, if $S = \{1, 2, 1, 3, 4, 1, 2\}$, then the sequence of the interarrival distributions at queue 1 is a repetition of {Erlang($\Lambda, 2$),Erlang($\Lambda, 3$),Erlang($\Lambda, 2$)}.
The pattern allocation policy was first introduced by Yum[26] as semi-dynamic deterministic routing. For the cases of two and infinitely many identical exponential server stations, Yum[26] shows a considerable reduction in mean waiting time if the pattern allocation policy is used instead of probabilistic allocation.

The arrival processes which result from pattern allocation are a special variant of the *Markovian Arrival Process* (MAP). A MAP is characterized by a continuous time Markov process with finite state space $\{1, \ldots, M\}$, where arrivals can occur only at transition epochs in the Markov process. The transitions at which an arrival takes place are defined by a 0-1 $M \times M$ matrix D, with $D_{ij} = 1$ if and only if the transition from i to j in the Markov process is associated with an arrival. In the MAP arising under pattern allocation, the Markov chain has the special property that only transitions from state i to state (i mod M) $+1$ can occur. In Appendix B we present some results from Lucantoni[15] for the MAP/G/1 queue, whose analysis is based on the matrix geometric techniques as developed by Neuts[17] and Ramaswami[20]. Using more classical techniques, Agrawala & Tripathi[2] analyse the waiting times in the MAP/M/1 queue for the typical MAP that we consider.
Observe that the Markovian Arrival Process in general is not a renewal process. The earlier mentioned comparisons from Stoyan[22] are for GI/G/1 queues and do not apply to MAP/G/1 queues. Besides, a useful characterization of the irregularity of a MAP is much more complicated than for GI arrival processes. However, in order to compare the MAP/G/1 queue with an M/G/1 queue with the same service time distribution we state the following conjecture, which is based on the observations made earlier in this section and supported by numerical experience.

Conjecture 3.1:
Consider a stable M/G/1 queue in which the arrival rate is $p\Lambda$, with $p < 1$, and the service time has distribution $B(\cdot)$. Then there exists a MAP with phase intensity Λ and overall arrival rate arbitrarily close to $p\Lambda$ such that in the MAP/G/1 queue with the same service time distribution $B(\cdot)$, $E\mathbf{W}^{MAP} < E\mathbf{W}^{M}$, where $\mathbf{W}^{MAP}$ and $\mathbf{W}^{M}$ denote the steady state waiting times of customers in the MAP/G/1 queue and M/G/1 queue respectively.

Conjecture 3.1 is clarified by viewing the Poisson($p\Lambda$) arrival process as the result of a probabilistic allocation and the MAP as the result of a pattern allocation. Let M be the number of phases in the MAP. Then in the pattern allocation out of every M arriving customers an exact fraction p is routed to the queue, whereas under probabilistic allocation this fraction is only in expectation equal to p. Moreover, in the MAP the arrivals can be better regulated, e.g., for $p = \frac{2}{5}$ every second and fifth customer can be routed to the queue.
Note that not for *all* MAP with phase intensity Λ and overall arrival rate $p\Lambda$, $E\mathbf{W}^{MAP} < E\mathbf{W}^{M}$; for example, again viewing the MAP as the result of pattern allocation, when of every

$2M$ customers the first M are routed to the queue, then for $\frac{1}{2} < \Lambda\beta < 1$ the mean waiting time of a customer at the queue tends to infinity as $M \to \infty$, while $p = \frac{1}{2}$ and $E\mathbf{W}^M < \infty$.

Also note that the refinement "arbitrarily close to $p\Lambda$" has to be made, because for p irrational there does not exist a MAP with finite state space of the underlying Markov process such that the overall arrival rate is exactly equal to $p\Lambda$. If we skip this assumption of finiteness then Hajek[10] gives the optimal arrival pattern with respect to minimizing mean queue length of a queue with an exponential server. And due to the optimality of this arrival pattern, for any other arrival sequence the mean queue length is bounded from below by the mean queue length using this optimal arrival pattern (this holds in particular for a sequence resulting from probabilistic allocation). Using Little's Law we also obtain bounds for mean waiting times of customers.

A second benefit of the pattern allocation policy, besides lowering the value of the objective function, is that it is more robust than probabilistic allocation. For example, from the explicit expression for the mean waiting times in an exponential server queue (cf. case 3.1), it follows that slightly altering the arrival intensity in an Erlang($\Lambda, 2$)/M/1 queue has less influence than changing the arrival intensity in an M/M/1 queue.

In this section we have argued that in the traffic allocation problem the pattern allocation policy is to be preferred to the probabilistic allocation policy, because of the reduction of variability in the arrival processes. In the next section we turn our attention to an optimization procedure for the pattern allocation policy.

4 Optimal Pattern Allocation

The mean waiting time of a customer in the MAP/G/1 queue is given by formula (B.5) of appendix B; it is a closed expression which can be evaluated. However, (B.5) is not very suitable for a direct optimization procedure; the matrix structure of (B.5) makes an exact analytical optimization actually impossible. This in contrast to probabilistic allocation where only the N optimal assignment probabilities p_i^* have to be determined and the simple structure of the objective function (2.3) allows an analytical solution of the Mathematical Programming Problem PA1.

Moreover, for pattern allocation it is impossible to determine the optimal allocation pattern by comparing patterns; there are too many patterns with length smaller than some practical bound, say $M_{\max}$, and the matrix operations involved in the evaluation of expression (B.5) are too time consuming.

We therefore have to resort to an approximate optimization procedure. Our procedure consists of two steps:

(1) Approximate p_i^*, $i = 1, \ldots, N$, the assignment frequencies of customers to the queues in the optimal allocation pattern.

(2) Use these frequencies for the construction of the allocation pattern.

In this section, our attention is mainly devoted to step 1. The problems related to step 2 are more of a combinatorial nature, and in fact a quite difficult cyclic scheduling problem has to be solved. In remark 4.3 we mention some of the difficulties occurring here, in appendix C we present a heuristic for building an allocation pattern from a set of allocation frequencies.

An obvious option for step 1 is using the optimal probabilistic allocation for the approximation of p_i^*, $i = 1, \ldots, N$. However, in general this does not lead to the optimal allocation pattern, as illustrated in Agrawala & Tripathi[1] for the traffic allocation problem with the mean sojourn time of a customer as objective function.
An exception is the special case of identical servers with a service time distribution that has an increasing failure rate. Liu & Towsley[14] prove that for this case $\{1, 2, \ldots, N\}$, the allocation pattern based on the optimal probabilistic allocation $p_i = \frac{1}{N}$, $i = 1, \ldots, N$, is the optimal allocation pattern.

Due to the matrix operations involved, comparing assignment frequencies directly using (B.5) is too time consuming, so further approximations have to be made.
To avoid the matrix operations we approximate the MAP with a GI arrival process in which the interarrival times are Gamma distributed. We then determine the optimal allocation fractions for the model in which we are to assign customers from an infinite reservoir of customers over N parallel Gamma/G/1 queues maintaining an overall arrival rate Λ. This we call the Gamma approximation procedure.
The idea of approximating the arrival process with a Gamma arrival process was first used by Ramakrishnan[19], who studied various allocation policies for the case of exponentially distributed service times. Using the exact implicit expressions for the mean waiting times in the Gamma/M/1 queue (cf. case 3.1 in section 3), Ramakrishnan numerically solved the Gamma/M/1 allocation problem for the case of two queues.
The Gamma(Λ, y) arrival process looks to be a reasonable approximation for the MAP with overall arrival intensity $\frac{\Lambda}{y}$. It possesses the same phase character as the MAP, and if y is an integer and the MAP is as regular as possible, both arrival processes are the same Erlang arrival process.
The Gamma arrival process can be viewed as the ideal MAP; if from an infinite reservoir of customers a_i customers out of every M have to be routed to queue i such that the interarrival times of the customers are i.i.d. and the sum of a_i interarrival times has an Erlang(Λ, M) distribution (the length of the arrival pattern), then the interarrival time of a customer has a Gamma$(\Lambda, \frac{M}{a_i})$ distribution. This implies that a Gamma$(\Lambda, \frac{M}{a_i})$ arrival process is more regular than the MAP with the same arrival intensity. Hence we expect the mean waiting times in the MAP/G/1 queue to be bounded from below by the mean waiting times in the corresponding Gamma/G/1 queue. Again, such a statement is hard to prove, except for the case of exponential servers, for which the proof readily follows from the results in Hajek[10].

Unfortunately, the expression for the mean waiting times of customers in a Gamma/G/1 queue (cf. Cohen[7]) is too complicated to be useful in an optimization procedure, and hence we have to resort to more simple approximate expressions for this mean waiting time.
However, this is not a point of too great concern; important is that the Gamma arrival process possesses the phase character of the MAP. Our experience suggests that a reasonably good approximation of the mean waiting time, which also fairly accurately captures its global behaviour, leads to good estimates of the optimal assignment frequencies.
In general, using the fractions obtained with the Gamma approximation procedure for the al-

location pattern gives a lower value of the objective function than using the fractions obtained with probabilistic allocation. This will be numerically illustrated in section 5.

The next part of this section is devoted to the actual determination of the allocation fractions. For the mean waiting times in a Gamma/G/1 queue we apply the two-moment approximation proposed by Krämer and Langenbach-Belz (KLB; cf. Krämer & Langenbach-Belz[13]) for GI/G/1 queues:

$$EW \;=\; \frac{\rho\beta}{2(1-\rho)}\left[c_a^2+c_s^2\right]e^{-\frac{2(1-\rho)}{3\rho}\frac{(1-c_a^2)^2}{c_a^2+c_s^2}} \qquad (4.1)$$

in which β is the mean service time, ρ is the load of the queue, and c_a^2 and c_s^2 denote the squared coefficient of variation (variance divided by squared mean) of the arrival time and service time distributions respectively. Remark that (4.1) is exact if the arrival process is Poisson.
For a Gamma(Λ,y) process the arrival rate λ is given by $\frac{\Lambda}{y}$ and $c_a^2 = \frac{1}{y}$, and for the Gamma/G/1 queue (4.1) thus becomes

$$EW \;=\; \frac{\Lambda\beta^2}{2(y-\Lambda\beta)}\left[\frac{1}{y}+\frac{\beta^{(2)}-\beta^2}{\beta^2}\right]e^{-\frac{2(y-\Lambda\beta)}{3\Lambda\beta}\frac{(1-\frac{1}{y})^2}{\frac{1}{y}-1+\frac{\beta^{(2)}}{\beta^2}}}. \qquad (4.2)$$

With (4.2) we can formulate the Mathematical Programming Problem for the Gamma approximation procedure. For objective function (2.2), substituting $\alpha_i = f_i = \frac{\lambda_i}{\Lambda} = \frac{1}{y_i}$ we find
GA1:

$$\begin{array}{ll} \min & \displaystyle\sum_{i=1}^{N}\frac{\Lambda\alpha_i^2\beta_i^2C_i}{2(1-\alpha_i\Lambda\beta_i)}\left[\alpha_i+\frac{\beta_i^{(2)}-\beta_i^2}{\beta_i^2}\right]e^{-\frac{2(1-\alpha_i\Lambda\beta_i)}{3\alpha_i\Lambda\beta_i}\frac{(1-\alpha_i)^2}{\alpha_i-1+(\beta_i^{(2)}/\beta_i^2)}} \qquad (4.3)\\ s.t. & \displaystyle\sum_{i=1}^{N}\alpha_i = 1,\\ & 0\le\alpha_i<\dfrac{1}{\Lambda\beta_i}, \quad i=1,\ldots,N. \end{array}$$

Problem GA1 has the same structure as PA1 in section 2, and hence it can easily be solved numerically with the algorithm presented in appendix A. Note that $\lim_{\{\epsilon\downarrow 0\}}\frac{dEW_i}{d\alpha_i}\,|_{\alpha_i=\epsilon}= 0$. Hence not only the optimal assignment frequencies resulting from GA1 are all greater than 0, but this would also be the case for objective function $\sum_i C_iEW_i$. The latter was not always the case for the optimal probabilistic allocation.
Earlier in this section we stated that the mean waiting times in the MAP/G/1 queue are bounded from below by the mean waiting times in the corresponding Gamma/G/1 queue. According to that, the solution of GA1 provides an approximate lower bound for the mean waiting costs under the optimal allocation pattern.

Remark 4.1: An important observation is that for the optimal pattern allocation more load than under probabilistic allocation is assigned to the queues with, relative to the other queues, high first moment of the service time distribution. This property was first reported by Agrawala & Tripathi[1].

The explanation of this property is that the effect of regularizing is stronger for the queues with relatively small assignment probabilities. For example: consider a traffic allocation problem with two queues for which the optimal probabilistic assignment fractions are $p_1^* = \frac{8}{9}$ and $p_2^* = \frac{1}{9}$. Then the MAP for the first queue would approximately be equal to a Poisson arrival process with arrival intensity $\frac{8}{9}\Lambda$, hence the switch from probabilistic to pattern allocation would not cause great changes in the arrival process at queue 1. However, for queue 2, switching from probabilistic to pattern allocation also changes the arrival process at queue 2 from a Poisson($\frac{1}{9}\Lambda$) into an Erlang($\Lambda, 9$) arrival process. The switch from probabilistic to pattern allocation has a more regularizing effect on queue 2 than on queue 1, and hence the decrement of the mean waiting times is larger for queue 2 than for queue 1. In terms of derivatives: for queue 1 the derivative of the mean waiting time as a function of the arrival rate in a neighbourhood of $\frac{8}{9}\Lambda$, is roughly equal to that for a queue with Poisson input. On the other hand, for queue 2 the behaviour of the waiting time much more resembles the behaviour of the waiting time in a queue with Erlang($\Lambda, 9$) input. Hence it will be smaller than in a queueing model with Poisson input. Recalling that in the optimal allocation, the derivatives of the terms in the objective function are all equal (cf. section 2), and that these terms largely depend on the mean waiting time, we find that more load is assigned to queue 2 under pattern allocation than under probabilistic allocation.

This example also shows why the Gamma approximation procedure has a better performance than the approximations obtained from probabilistic allocation: *the Gamma arrival process better captures the influence of assignment fractions on the degree of regularization.*

Remark 4.2: Elaborating on remark 4.1, we expect that the effect of a transition from probabilistic allocation to pattern allocation will be stronger when the assignment fractions are closer to each other. In that situation all servers will profit from regularization. An interesting conclusion is that for the case of non-identical service rates, comparing the Gamma approximation procedure with probabilistic patterns, the difference in patterns is in particular pronounced for low system loads. When the load increases both methods will lead to allocation fractions close to the capacities of the queues, but for low load probabilistic allocation tends to assign many more customers to the faster queue than the Gamma approximation. Another interesting conclusion is that when the number of servers increases, the effect of regularizing becomes stronger. For example, consider the case of k identical servers with service rate 1 and $\Lambda = \rho k$, $\rho < 1$ and all servers receiving the same fraction $\frac{1}{k}$ of the arrivals. The allocation pattern based on these fractions leads to k Erlang($\rho k, k$) arrival processes. Stoyan[22] example 1.5.1(e) shows that Erlang($\rho(k+1), k+1$) $\leq_c$ Erlang($\rho k, k$). Hence the value of the objective function decreases when k increases. Note that for $k \to \infty$ the arrival processes at the queues become deterministic.

We conclude this section with a remark concerning the validity of our optimization procedure. In this remark we also reveal some problems which occur in the second step of the procedure, where allocation frequencies are to be translated into patterns.

Remark 4.3: The optimization procedure *assumes* a 1-1 correspondence between assignment fractions and an allocation pattern. However, such a 1-1 correspondence does not hold.

From an allocation pattern one can obtain a_i, the number of occurrences of queue i in the pattern, and also the assignment frequencies p_i by using $p_i = \frac{a_i}{\sum_j a_j}$, $i = 1, \ldots, N$.

Reversely, assignment fractions p_i do not determine a unique allocation pattern. Firstly, as explained in the previous section, the p_i's can be irrational, so in general a finite pattern with corresponding assignment fractions p_i for $i = 1, \ldots, N$ does not exist. And secondly, even if there exist integer numbers a_i such that $p_i = \frac{a_i}{\sum_j a_j}$ for $i = 1, \ldots, N$, the orders in which the queue numbers can be placed in a pattern are numerous.

However, the natural requirement that the arrival processes should be as regular as possible causes a set of allocation fractions to lead to a more or less uniquely determined allocation pattern. Let us now consider the translation of assignment fractions into patterns.

First of all $(a_1, \ldots, a_N)$ are defined in the following way. For all $\epsilon > 0$, there exists an integer $\bar{m}$ such that $\bar{m} = \min\{m > N| \parallel (p_i m - [p_i m])/[p_i m] \parallel < \epsilon, \; \frac{[p_i m]}{m}\Lambda < \beta_i, \; i = 1, \ldots, N\}$. Let $a_i = [p_i \bar{m}]$, $i = 1, \ldots, N$. Hence, the a_i's are uniquely defined by a chosen $\epsilon > 0$. Note that the value of ϵ has a strong influence on the length of the pattern.

Secondly, in section 3 we saw that the mean waiting time decreases with increasing regularity of the arrival process; so given numbers a_i, $i = 1, \ldots, N$, we try to construct an allocation pattern in which the occurrences of the queue numbers are as uniformly distributed as possible. In this way, given $\epsilon > 0$, assignment fractions p_i correspond to a more or less uniquely determined allocation pattern.

This does not imply monotonicity of the waiting times as a function of the assignment fractions. For certain values p_i, $i = 1, \ldots, N$, placing the queue numbers into a pattern in a uniformly distributed way can be rather difficult, whereas after slightly altering the frequencies, a much more regular pattern would arise. This property also has consequences for the value of the objective function. This in contrast with probabilistic allocation where, given the assignment fractions, the model is equivalent to N independent M/G/1 queues and the objective function is strictly convex.

The actual construction of an allocation pattern is an interesting combinatorial problem, for which we present a heuristic in appendix C. Here the main problem is that the interests of the queues interfere, i.e. we try to make the arrival process as regular as possible for all queues simultaneously. An example of such interference is with $N = 3$, $a_1 = 1$, $a_2 = 2$, $a_3 = 3$. The reader can easily check that there exists no pattern of length 6 in which the arrival process at all three queues is a renewal process.

In general *the* optimal allocation pattern can not be determined, hence it is not to be expected that the optimal assignment frequencies are determined by applying the Gamma approximation procedure. However, our numerical experience indicates that this procedure results in a pattern under which the objective function is close to the approximate lower bound for the optimal arrival pattern, this lower bound being the value of the solution of GA1.

5 Numerical results

In this section we present some numerical results. We compare various allocation policies and also discuss the quality of the Gamma approximation procedure. We show five instances for the case of two servers and three instances for the case of three servers in parallel. For each instance the objective was to minimize the mean waiting time of a customer. As a function of the load of the total system, we present absolute and relative values of the objective function for various optimized allocation policies and for the solutions of the mathematical programs.

Description of numerical instances and presented results

In appendix D in figures 5.1-5.8 we show numerical results for 8 instances. We have considered the problem of minimizing the mean waiting time of an arbitrary customer (taking $f_i(P) = p_i$, $C_i = 1$, in (2.2), $i = 1, \ldots, N$). For each instance we have optimized various allocation policies for system loads ρ that we increased from 0.05 to 0.95 with steps of 0.05. The system load we defined as $\rho = \Lambda(\sum_i \frac{1}{\beta_i})^{-1}$, i.e., the offered traffic to the system divided by the total service capacity of the system.

Figures 5.1-5.5 concern the case of two servers, 5.6-5.8 the case of three servers. We have considered three types of service time distributions: Exponential, Erlang 2 and Hyper-Exponential. For the Hyper-Exponential distribution the coefficient of variation is 2. In cases 5.1-5.3 and 5.6-5.8 the servers are of the same type, but differ in service rates. For cases 5.4 and 5.5 the servers are not of the same type; in case 5.4 both servers have identical service rate, in case 5.5 the rates are different.

For each instance two figures are presented, one displaying absolute value of the objective function, the other showing this value relative to the value for optimal probabilistic allocation. The abbreviations in the figures stand for:

prob: mean waiting times, under the optimal probabilistic allocation.

prop: pattern that is based on the optimal probabilistic pattern.

klbp: pattern obtained via the gamma approximation procedure, using the KLB approximation for the Gamma/G/1 queue.

klbb: approximate lower bound for the mean waiting times under the optimal allocation pattern (see section 4).

lb: hard lower bound for the mean waiting times under the optimal allocation pattern. This is for the case of exponential servers (see section 4).

jlw: mean waiting times under the dynamic policy that allocates a customer to the queue with the least waiting time. These are simulation results.

Comparing probabilistic and pattern allocation

In section 3 we argued that regularizing arrival processes leads to lower mean waiting times. Also, in conjecture 3.1 we stated that for each M/G/1 queue there exists a MAP/G/1 queue with lower mean waiting times, where the Poisson arrival process has intensity $p\Lambda$, $p < 1$, and the MAP has the same arrival rate and phase intensity Λ. We concluded that pattern allocation leads to lower mean waiting times than probabilistic allocation. This conclusion is supported by our numerical results. In all cases considered, the allocation pattern based on the optimal probabilistic allocation fractions (curves **prop** in fig. 5.1-5.8) performs better than the probabilistic allocation itself (**prob**). The relative differences vary from 7 to 40% for low loads up to about 40% for high load, except for the case of non-identical servers with identical rate, where the difference for low loads is even 50%. Also, the effect of a transition from probabilistic to pattern allocation is stronger for the case of three servers. All observations are illuminated by remarks 4.1 and 4.2.

Figures 5.1-5.3 and 5.6-5.8 suggest that for identical servers the effect of a transition from probabilistic to pattern allocation is stronger when the service time distribution has a smaller coefficient of variation.

The non-smoothness of the curves (**prop**) in figures 5.6-5.8 is caused by the way the patterns were constructed in our numerical experiments. Due to pattern length limitations, imposed by computer capacity, some inaccuracies occur. The assigned fractions in the pattern are for some values of ρ closer to the optimal probabilistic allocation than for others. It is interesting to see that when the deviation results in - relative speaking - more (less) load at a slower server, this decreases (increases) the value of the objective function. In remark 4.1 this observation is explained. The effect is most pronounced for ρ=0.1, wher the constructed allocation pattern actually assigns no customers to the slowest server.

Comparing the Gamma approximation procedure with pattern allocation based on optimal probabilistic allocation

In section 4 we concluded that the Gamma approximation procedure would lead to better allocation patterns than probabilistic allocation because the Gamma arrival process better captures the behaviour of the MAP than the Poisson arrival process. This conclusion is supported by the numerical results. The difference between objective functions for Gamma approximation based patterns(**klbp**) and probabilistically based patterns(**prop**) is larger for lower loads than for higher loads. The difference ranges from 0% to 45%.

Optimal pattern allocation

Finally we turn to the questions (i) how good is pattern allocation compared to the best policy, and (ii) how close is the pattern obtained with the Gamma approximation procedure to the optimal allocation pattern?

Concerning (i), it is very hard to determine the optimal - probably a dynamic - allocation policy. Hence we have considered a dynamic policy which is expected to perform better than most policies, and considerably better than the static policies. This dynamic policy has total knowledge of the system at the moment of arrival and sends each customer to the queue with smallest waiting time. Our claim that this is a nearly optimal allocation policy, is based on the fact that this policy uses all information available at moments of arrival and seems to use this information in a very sensible way. In the figures one can see that this dynamic policy(**jlw**) performs from 40% up to about 95% better than probabilistic allocation. The difference with the optimal Gamma approximated pattern ranges from 20% up to 45%.

Concerning question (ii), we know that it is very hard to determine *the* optimal allocation pattern. However, in an indirect way we are able to make a statement about the quality of the Gamma approximation procedure. In section 4 we stated that waiting times in a MAP/G/1 queue are bounded from below by the waiting times in the corresponding Gamma/G/1 queue. Numerical experience shows that the approximation of the mean waiting times in the MAP/G/1 queue, using the KLB formula for waiting times in the corresponding Gamma/G/1 queue, is fairly accurate. Hence the value of the objective function for the solution of mathematical program GA1(**klbb**) is an approximate lower bound for the optimal allocation pattern.

We also notice that, in particular for high loads, this value reasonably accurately approximates the mean waiting times under the allocation pattern that is constructed from the solution of GA1.

So GA1 provides an approximate lower bound for the mean waiting times under the optimal allocation pattern, as well as an accurate approximation of the mean waiting times under the allocation pattern that is based on the solution of GA1. We conclude that the Gamma approximation procedure provides us with a nearly optimal allocation pattern.

6 Extensions of the Traffic Allocation Problem

In this section we briefly discuss the traffic allocation problem for two extensions of the model that was discussed in the previous sections. For both models, the traffic allocation problem can be approached in a similar way as the original problem.
First we look at the situation in which one has to allocate a Poisson arrival stream to N queues, where each queue already receives a Poisson arrival stream. This problem is known as the traffic allocation problem with dedicated arrival streams. The second model under consideration is the allocation problem with multiple server stations.
In section 2 we argued that regularizing arrival streams has a lowering effect on mean waiting times. Based on similar intuitive arguments, we now make the same conjecture for the extended allocation problems, realizing that proving the same statements for the extended models can only be harder than for the original allocation problem.
According to this assumption, the customers are allocated using an allocation pattern rather than assigning them to the queues probabilistically.

The allocation problem with dedicated arrival streams

A Poisson arrival stream with intensity Λ has to be allocated to N queues Q_i, each queue already receiving a Poisson arrival stream with intensity $\lambda_i^d \geq 0$, $i = 1, \ldots, N$. Note that the original problem returns if $\lambda_i^d = 0$ for $i = 1, \ldots, N$. For the allocation problem with dedicated arrival streams, for the case of exponential servers, Ni & Hwang[18] optimize the probabilistic allocation policy with the mean sojourn time of a customer as objective function.

As indicated in the introduction of this section, we again try to approximate the optimal allocation pattern, although the benefit of using pattern allocation is less substantial than in the original allocation problem without dedicated arrivals, because allocated arrivals from the additional Poisson arrival stream (forming a MAP) join in with the arrivals from the dedicated Poisson (λ_i^d) arrival stream. So the arrival processes at the queues are not as regular as the MAP in the original problem, they are the sum of such a MAP and a Poisson arrival process. Although the sum of two MAP's is also a MAP, and the Poisson process is just a special MAP, the arrival processes at the queues are hard to approximate by any GI arrival process, in particular the Gamma arrival process.
As a result, we try to approximate the optimal assignment fractions for the allocation pattern with probabilistic allocation or with the Gamma optimization procedure, depending on the ratio between the sum of the dedicated arrival rates and the extra arrival rate. For example: if the sum of dedicated arrival rates is large compared to the rate of the extra arrival stream, then the resulting arrival processes will resemble a Poisson process more than a MAP, hence it makes more sense to use the assignment fractions from probabilistic allocation for the allocation pattern.
Below the Mathematical Programming Problem is formulated for the probabilistic allocation policy; for the Gamma optimization procedure the formulation is quite similar.

DA1:

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} f_i C_i \frac{(\lambda_i + \lambda_i^d)\beta_i^{(2)}}{2(1-(\lambda_i+\lambda_i^d)\beta_i)} & (6.1) \\ s.t. \quad & \sum_{i=1}^{N} \lambda_i = \Lambda, \\ & \lambda_i + \lambda_i^d < \frac{1}{\beta_i}, \ i = 1,\ldots,N, \\ & \lambda_i \geq 0, \ i = 1,\ldots,N. \end{aligned}$$

The structure of DA1 is similar to the earlier presented Mathematical Programming Problems PA1 and GA1. Hence, the solution of DA1 can easily be determined using the allocation algorithm in appendix A.

Note that under probabilistic allocation, the original probabilistic allocation problem reappears with the additional constraints that the arrival rate at queue i should be at least λ_i^d, $i = 1,\ldots,N$.

Remark 5.1: If f_i in (6.1) represents the fraction of the customers sent to queue i then we have the choice between two interpretations:

1. $f_i = \frac{\lambda_i}{\Lambda}$, f_i is the fraction of the customers of the additional arrival stream routed to queue i.
2. $f_i = \frac{\lambda_i+\lambda_i^d}{\Lambda+\sum_j \lambda_j^d}$, f_i is the fraction of the total arrival stream which is allocated to queue i.

We finish this discussion of load balancing with dedicated arrival streams by mentioning two references on this topic.

Bonomi & Kumar[4] discuss an adaptive probabilistic allocation policy for the case of exponential and the case of identical servers with objective function the mean sojourn time of a customer. They consider the situation where not all system parameters are known, or where some of the parameters may change from time to time. In their model the customers are assigned probabilistically, but the allocation probabilities are not fixed; they may be changed at periodical information updates about idling times of the servers. Their main concern is the speed of convergence of the allocation policy towards the optimal assignment probabilities.

Ross & Yao[21] consider the following N server model. At server i a set S_i of customer types arrives according to Poisson processes with arrival intensities λ_{ij}, $j \in S_i$, $i = 1,\ldots,N$. The $j-th$ arrival stream at server i has service time distribution $B_{ij}(\cdot)$, $j \in S_i$, $i = 1,\ldots,N$. Furthermore, each server generates additional customers, according to a Poisson process, which may be routed to one of the other servers. The service time of such a customer at server i has distribution $B_i(\cdot)$. The aim in Ross & Yao[21] is to find the probabilistic allocation policy that minimizes the sum of the mean sojourn time and some rerouting delay of a customer from the additional arrival process, under the constraints that the mean sojourn time of the $j-th$ dedicated customer stream at server i is less than or equal to α_{ij}, $j \in S_i$, $i = 1,\ldots,N$. Ross & Yao[21] allow local priority scheduling of customer types, which also involves the additional customers. The essential problem is to derive an expression for $E\mathbf{R}_i$,

the mean sojourn time at server i of an additional customer when local priority scheduling of customers is allowed. Using matroid theory Ross & Yao[21] prove that $x_i E\mathbf{R}_i$ is a convex function in x_i, where x_i denotes the additional load assigned to server i. The remaining problem, determining the optimal assignment vector $x^* = (x_1^*, \ldots, x_N^*)$, proceeds in a way that is similar to solving a common RAP.
It might be interesting to use the resulting assignment vector for determining a good pattern allocation.

Allocation for the case of Multiple Server Queues
In this model, a Poisson arrival stream with intensity Λ has to be allocated over N multiple server queues $Q_1, \ldots, Q_N$, where the number of servers at Q_i is s_i, $i = 1, \ldots, N$.
Except for a few special cases, no explicit expressions for the mean waiting times in GI/G/s or MAP/G/s queueing systems are available. Hence, in this model, an optimal allocation can not be determined analytically, for both probabilistic assignment and pattern allocation.
We again assume that regularizing the arrival streams decreases mean waiting times, and again we expect to obtain better allocation fractions using the Gamma/G/s_i approximation for the MAP/G/s_i queue than when using the M/G/s_i approximation.
Using this assumption, the path towards an allocation pattern is reasonably straightforward; by choosing a suitable strictly convex approximation for the mean waiting times in a Gamma/G/s queue one can formulate a Mathematical Programming Problem which possesses all the required properties for applying the algorithm in appendix A. Whitt[25] discusses several mean waiting time approximations for the GI/G/s queue.

Acknowledgement
The authors are indebted to professor J.W. Cohen for carefully reading the manuscript, to S.C. Borst for valuable suggestions concerning the algorithm for constructing an allocation pattern, and to G.M. Koole for interesting discussions.

References

[1] Agrawala, A.K., Tripathi, S.K. (1981). On the optimality of semidynamic routing schemes. *Inf. Proc. Letters* **13**, 20-22.

[2] Agrawala, A.K., Tripathi, S.K. (1982). On an exponential server with general cyclic arrivals. *Acta Inf.* **18**, 319-334.

[3] Boel, R.K., van Schuppen, J.H. (1989). Distributed routing for load balancing. *Proc. IEEE* **77**, 210-221.

[4] Bonomi, F., Kumar, S. (1990). Adaptive optimal load balancing in a nonhomogeneous multiserver system with a central job scheduler. *IEEE Trans. Computers* **39**, 1232-1250.

[5] Boxma, O.J., Levy, H., Weststrate, J.A. (1991). Efficient visit frequencies for polling tables: minimization of waiting costs. *Queueing Systems* **9**, 133-162.

[6] Buzen, J.P., Chen, P.P.-S. (1974). Optimal load balancing in memory hierarchies. In:*Proceedings of IFIP* 1974, 271-275. (North-Holland, Amsterdam, J.L. Rosenfeld, (ed.)).

[7] Cohen, J.W. (1982). *The Single Server Queue.* (North-Holland, Amsterdam, 2nd ed.).

[8] Ephremides, A., Varaiya, P., Walrand, J. (1980). A simple dynamic routing problem. *IEEE Trans. Automatic Control.* **AC-25**, 690-693.

[9] Jean-Marie, A. (1987). Load balancing in a system of two queues with resequencing. In:*Proceedings of Performance* '87, 75-88. (North-Holland, Amsterdam, P.J. Courtois, G. Latouche (eds.)).

[10] Hajek, B. (1985). Extremal splittings of point processes. *Math. of Oper. Res.* **10**, 543-556.

[11] Ibaraki, T.I., Katoh, N. (1988). *Resource Allocation Problems.* (MIT Press, Cambridge).

[12] Itai, A., Rosberg, Z. (1984). A Golden Ratio control policy for a multiple-access channel. *IEEE Trans. Automatic Control* **AC-29**, 399-419.

[13] Krämer, W., Langenbach-Belz, M. (1976). Approximate formulae for the delay in the queueing system GI/G/1. In:*Proceedings of the 8th ITC Congress*, Melbourne, 1976, 235.1-235.8 .

[14] Liu, Z., Towsley, D. (1992). Optimality of the Round Robin routing policy. COINS Technical Report, TR 92-55.

[15] Lucantoni, D.M. (1991). New results on the single server queue with a batch Markovian arrival process. *Commun. Statist.-Stochastic Models* **7**, 1-46.

[16] Neuts, M.F. (1979). A versatile Markovian arrival process. *J. Appl. Prob.* **16**, 764-779.

[17] Neuts, M.F. (1989). *Structured Stochastic Matrices of M/G/1 Type and their Applications.* (Dekker, New York).

[18] Ni, L.M., Hwang, K. (1985). Optimal load balancing in a multiple processor system with many job classes. *IEEE Trans. Software Engineering* **SE-11**, 492-496.

[19] Ramakrishnan, K.K. (1983). *The Design and Analysis of Resource Allocation Policies in Distributed Systems.* (Ph. D. Thesis, University of Maryland, Dept. of Computer Science).

[20] Ramaswami, V. (1980). The N/G/1 queue and its detailed analysis. *Adv. Appl. Prob.* **12**, 222-261.

[21] Ross, K.W., Yao, D.D. (1991). Optimal load balancing and scheduling in a distributed computer system. *Journal of the ACM* **38**, 676-690.

[22] Stoyan, D. (1983). *Comparison Methods for Queues and other Stochastic Models.* (Wiley, New York, Translated and revised version of German original (1977)).

[23] Tang, C.S., van Vliet, M. (1991). Traffic allocation for manufacturing systems. *Technical Report 9116/A, Econometric Institute, Erasmus University Rotterdam, 1991.* To appear in *Eur. J. Oper. Res.*

[24] Wang, Y-T., Morris, R.J.T. (1985). Load sharing in distributed systems. *IEEE Trans. Computers* **C-34**, 204-217.

[25] Whitt, W. (1985). Approximations for the GI/G/m queue. Technical unpublished paper, AT & T Bell Labs, Murray Hill, N.J..

[26] Yum, T.P. (1981). The design and analysis of a semidynamic deterministic routing rule. *IEEE Trans. Comm.* **29**, 498-504.

Appendices

A Traffic Allocation Algorithm

In this appendix we present an algorithm to solve the Mathematical Programming Problems that where defined in sections 2 and 4.

The Mathematical Programming Problems for the traffic allocation problem fit into the class of Resource Allocation Problems (RAP) with a separable, strictly convex continuous objective function. In chapter 2 of Ibaraki & Katoh[11] this class is studied. They also present several algorithms from which we have derived the algorithm for the traffic allocation problem.
The basic formulation of a RAP as studied in chapter 2 of Ibaraki & Katoh[11] is:
RAP:

$$\min \quad \sum_{i=1}^{N} T_i(x_i) \tag{A.1}$$

$$s.t. \quad \sum_{i=1}^{N} x_i = C, \tag{A.2}$$

$$0 \leq x_i, \quad i = 1, \ldots, N. \tag{A.3}$$

Here $T_i(\cdot)$ is a strictly convex function, $i = 1, \ldots, N$. For the traffic allocation problem, the x_i's have upper bounds u_i, which follow from the stability conditions of the queues. Hence, we obtain the following Mathematical Programming Problem:
MP:

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} T_i(x_i) \\ s.t. \quad & \sum_{i=1}^{N} x_i = C, \\ & 0 \leq x_i < u_i, \quad i = 1, \ldots, N. \end{aligned}$$

For the traffic allocation problem with dedicated arrival streams (cf. Section 4), the x_i's have lower bounds $l_i \geq 0$. However, using a translation of the variables, we can always formulate a Mathematical Programming Problem which has the form of MP.

Using the separability and convexity of the objective function, the first order Kuhn-Tucker constraints lead to the following lemma:

Lemma A1:
Provided that a feasible solution exists, the unique optimum $x^* = (x_1^*, \ldots, x_N^*)$ of MP fulfills the following set of equations:

$$\sum_{i=1}^{N} x_i = C, \tag{A.4}$$

$$\text{if } x_i > 0 \text{ then } T_i'(x_i) = \delta, \tag{A.5}$$

$$\text{if } x_i = 0 \text{ then } T_i'(x_i) \geq \delta, \tag{A.6}$$

where δ is the Lagrange multiplier belonging to constraint (A.2).

Proof: Knowing that $T_i(\cdot)$ is strictly convex and that for the optimum $x_i^* < u_i$, $i = 1, \ldots, N$, these equations follow directly from the first order Kuhn-Tucker constraints for the Lagrange relaxation of MP. □

The following lemma is an extension of lemma 2.2.1 in Ibaraki & Katoh[11].

Lemma A2:
Let T_i be strictly convex and continuously differentiable over the interval $[0, u_i)$ and $T_i'(x_i) \rightarrow \infty$ as $x_i \rightarrow u_i$, $i = 1, \ldots, N$, and let the indices of T_i be arranged such that

$$T_1'(0) \leq T_2'(0) \leq \ldots \leq T_N'(0).$$

If $x^* = (x_1^*, \ldots, x_N^*)$ is an optimal solution of MP, then there exists an index $i^* \in \{1, \ldots, N\}$ such that

$$\begin{aligned} x_i^* &> 0, \quad i = 1, \ldots, i^*, \\ x_i^* &= 0, \quad i = i^* + 1, \ldots, N. \end{aligned}$$

Proof: Since x^* is optimal, there exists a δ for which (A.5) and (A.6) hold. Let i^* be the smallest index such that $x_{i^*+1}^* = 0$. Then, from lemma A1, $T_{i^*+1}'(0) \geq \delta$. Suppose $x_i^* > 0$ for a certain $i > i^*$. Then, from the strict convexity of the T_i's and (A.6), it follows that $T_i'(x_i^*) > T_i'(0) \geq T_{i^*}'(0) \geq \delta$, which is in contradiction with (A.5). Thus $x_i^* = 0$ for $i = i^* + 1, \ldots, N$. □

Below we present an algorithm for solving MP, based on lemma A1 and A2. The algorithm consists of two phases; first i^* is computed, subsequently MP is solved with a reduced set of variables $x_1, \ldots, x_{i^*}$, putting $x_{i^*+1}, \ldots, x_N$ equal to zero.

Algorithm 1:

Phase 1:
0. Order the terms T_i such that $T_1'(0) \leq T_2'(0) \leq \ldots \leq T_N'(0)$.

1. Compute $i_0 = \min\{i \mid \sum_{j=1}^{i} u_j > C\}$. (from lemma A2 and (A.2) it follows that i_0 is the minimal number of positive x_i's that is required for a feasible solution.)
2. $k := i_0$.
3. Compute x_i for $i = 1, \ldots, k$, such that $T_i'(x_i) = T_{k+1}'(0)$ (binary search, or Newton's method).
4. if $\sum_{i=1}^{k} x_i > C$ then GOTO 6.
5. $k := k + 1$, if k =N then GOTO 6 else GOTO 3.
6. $i^* = k$.

Phase 2:

7. Compute δ and $x_i(\delta)$ such that $T_i'(x_i(\delta)) = \delta$, $i = 1, \ldots, i^*$, and $\sum_{i=1}^{i^*} x_i = C$. This can be done by a binary search for δ in $[T_{i^*}'(0), T_{i^*+1}'(0)]$. (if $i^* = N$ then set $T_{i^*+1}'(0)$ equal to $\max_i T_i(y_i)$ for an arbitrary feasible allocation y.)
8. $x_i^* = x_i(\delta)$, $i = 1, \ldots, i^*$, $\quad x_i^* = 0$, $i = i^* + 1, \ldots, N$.
9. STOP.

B Waiting times in the MAP/G/1 queue

In this appendix we present some results on the waiting times of customers in the MAP/G/1 queue, and apply them to the typical MAP/G/1 queue which arises under pattern allocation. For general results on the MAP/G/1 queue we refer to papers by Lucantoni[15], Ramaswami[20] and to the book of Neuts[17] which in great detail discuss all aspects of the MAP and the MAP/G/1 queue as parts of a more general framework. In Lucantoni[15] analytical results are obtained for queue lengths, busy period lengths, and waiting times in the MAP/G/1 queue. Lucantoni[15] also provides algorithms to obtain explicit results for distributions and moments of distributions.

The MAP is defined by a continuous time Markov process $\{J(t), t \geq 0\}$ with a finite state space $E = \{1, \ldots, M\}$, where at transition epochs arrivals can occur. E represents the set of phases of the arrival process.

The Markov process has generator D, which can be decomposed in two $M \times M$ matrices, D_0 and D_1. Let $P = (p_{ij})$, $i, j \in E$, be the one-step transition probability matrix, then $D = D_0 + D_1$ where

$$(D_0)_{ij} = \begin{cases} -\Lambda, & i,j \in E, i = j, \\ \Lambda p_{ij}, & i,j \in E, i \neq j \text{ and no arrival} \\ & \text{at a transition from } i \text{ to } j \\ 0 & \text{otherwise.} \end{cases}$$

$$(D_1)_{ij} = \begin{cases} \Lambda p_{ij}, & i,j \in E, i \neq j \text{ and an arrival} \\ & \text{at a transition from } i \text{ to } j \\ 0 & \text{otherwise.} \end{cases}$$

D_0 (D_1) represents transitions in the Markov process $\{J(t), t \geq 0\}$ without (with) arrivals of customers.

The fundamental arrival rate for the MAP is defined as

$$\lambda' = \pi D_1 e,$$

in which π is the stationary probability row vector of the Markov process with generator D and e the M-dimensional unit column vector.

For the MAP/G/1 queue, let the service time B have an arbitrary distribution $B(\cdot)$ with first and second moment β and $\beta^{(2)}$ respectively, and let $\beta(\cdot)$ be the Laplace-Stieltjes Transform of B. Neuts[17] shows that the MAP/G/1 queue is stable if $\lambda'\beta < 1$.
Let $W_v(\cdot) = \{W_1(\cdot), \ldots, W_M(\cdot)\}$, where $W_j(x)$ is the joint probability that at an arbitrary time the arrival process is in phase j and the amount of work at the server is at most x. $\mathbf{W}_v$ is the row vector of the virtual waiting times.
A basic result for the MAP/G/1 queue is the Laplace-Stieltjes Transform of $\mathbf{W}_v$ (which is the matrix equivalent of the Pollaczek-Khintchine formula for the ordinary M/G/1 queue):

$$\tilde{W}_v(s) = s\,(1 - \lambda'\beta)g\,[sI + D_0 + \beta(s)D_1]^{-1}, \qquad s \geq 0. \tag{B.1}$$

Here g is the invariant probability vector of a matrix $G = (G_{ij})$, G_{ij} is the probability that at the end of a busy period the arrival process is in phase j, given that it was in state i at the beginning of that busy period. G can be computed using the following matrix functional equation:

$$G = \int_{x=0}^{\infty} e^{(D_0 + D_1 G)x} dB(x). \tag{B.2}$$

The functional equation (B.2) is obtained by an extension of the branching argument for the M/G/1 busy period (cf. Cohen[7], p. 249).

From (B.1), for the specific MAP/G/1 queue which arises under pattern allocation, one obtains $E\mathbf{W}_v$, the vector of mean virtual waiting times:

$$E\mathbf{W}_v \;=\; (E\mathbf{W}_v)e\pi + \pi - ((1 - \lambda'\beta)g + \beta\pi D_1)(e\pi + D)^{-1}, \tag{B.3}$$

in which $(E\mathbf{W}_v)e$ is given by

$$(E\mathbf{W}_v)e \;=\; \frac{1}{2(1 - \lambda'\beta)}\left[2(\lambda'\beta - ((1 - \lambda'\beta)g + \pi\beta D_1)(e\pi + D)^{-1}\beta D_1 e) + \lambda'\beta^{(2)}\right]. \tag{B.4}$$

Using (B.3), the PASTA property, and remarking that for the typical MAP that we study $p_{ij} = 1$ if $j = (i \bmod M) + 1$, we obtain $E\mathbf{W}_{MAP/G/1}$, the mean waiting time of a customer in the MAP/G/1 queue:

$$E\mathbf{W}_{MAP/G/1} \;=\; E\mathbf{W}_v \frac{D_1 e}{\Lambda}. \tag{B.5}$$

C Constructing the allocation pattern

In this appendix we discuss the problem of constructing an allocation pattern from a given set $(p_1, \ldots, p_N)$ of allocation fractions. First of all, these frequencies are translated into the vector $(a_1, \ldots, a_N)$, in which a_i is the number of occurrence of index i in the pattern. The integers a_i are computed by defining $\bar{m} = \min\{m \geq N \mid (p_i m - \lfloor p_i m \rceil)/\lfloor p_i m \rceil < \epsilon, p_i > 0, i = 1, \ldots, N\}$, and taking $a_i = \lfloor p_i \bar{m} \rceil$, $i = 1, \ldots, N$. Note that the choice of ϵ has a strong influence on the length of the pattern. After this translation there remains the problem of determining an allocation pattern, such that the number of arriving customers at the routing point between two consecutive allocations to queue i is as constant as possible. Moreover, one has to achieve this for all queues simultaneously. In various optimization problems this combinatorial cyclic scheduling problem has been encountered. Itai & Rosberg[12] suggest the so-called Golden Ratio method for a cyclic scheduling problem that arises in the access control for a multi-access channel. Boxma et al. [5] study a polling model in which a server visits the queues according to a polling table. For this more or less dual problem of the traffic allocation problem they follow an optimization procedure which is similar to our approach for the traffic allocation problem. First good visit frequencies are computed for a polling model in which the server chooses his next queue probabilistically, subsequently a polling table based on these frequencies is constructed, using the Golden Ratio method.

The combinatorial complexity of the cyclic scheduling problem is yet undetermined. However, it seems to be a hard problem; it can be translated to known NP-hard problems, although with special structures, but those special structures do not seem to reduce the problem to a polynomially solvable one.

In this appendix a heuristic based on the paper by Hajek[10] on extremal splittings of point processes is presented. This heuristic is an alternative for the Golden Ratio policy as described in Itai & Rosberg[12].

First some notation and a mathematical criterion for optimality are introduced. A pattern S is defined by $S = \{s_1, \ldots, s_k\}$ in which $k = | S |$ is the length of S. Let S_0 be the class of patterns of length $M = \sum_i [f_i \bar{m}]$ in which index i occurs exactly a_i times. In the rest of this appendix we assume that $a_1 \geq a_2 \geq \ldots \geq a_{\nabla'}$ and we set N equal to the number of queues with $a_i > 0$. Under the allocation pattern $S \in S_0$ the interarrival times at queue i form a repeated sequence of a_i Erlang$(\Lambda, d_{i_j}(S))$ distributed variables, $j = 1, \ldots, a_i$. The 'distances' $d_{i_j}(S)$ are the numbers of arriving customers at the routing point between two consecutive allocations to queue i.

Next, the problem is to determine $S^* = \text{argmax}_{S \in S_0} V(S)$, in which $V(S) = \{\sum_i a_i \sum_{j=1}^{a_i} d_{i_j}^2(S)\}$. The objective function $V(\cdot)$ tries to capture the notion of even spreading in the pattern by a kind of second moment function. The weights a_i have been chosen such that in the optimal allocation, i.e. $d_{i_j} = \frac{M}{a_i}$, $j = 1, \ldots, a_i$, $i = 1, \ldots, N$, the contributions of the queues to the objective function are all equal. The optimality criterion is quite arbitrary, for example the weight factors could also have favored the queues with high or those with low frequencies. The same holds for the order in which the indices are included. At the moment it is unclear which objective function is best. However, our numerical experience suggests that slightly altering these factors does not have a substantial influence on the value of the objective function.

The Algorithm

The algorithm consists of two phases. In phase 1 a basic pattern is created with a method derived from Hajek[10]. In phase 2, this pattern is improved with the use of a local search method.

Phase 1:

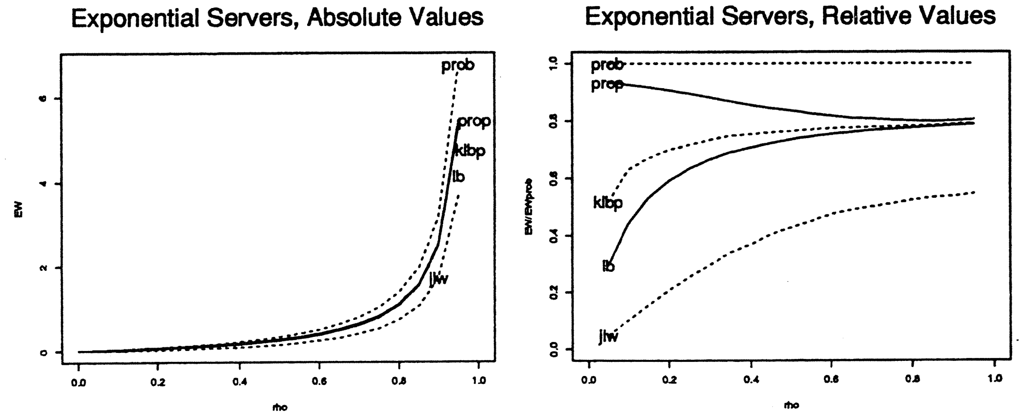
A basic pattern is constructed in an iterative way, starting with an empty pattern and consecutively inserting the indices of the queues into the pattern. After step i, the algorithm has produced a sub-pattern S_i, which contains the indices of queues $1, \ldots, i$. The method operates as follows: If in step i in sub-pattern S_{i-1} of length k, the index of queue i has to be inserted a_i times, then first the distances d_{i_j} for the next subpattern S_i are computed, following Hajek[10], by $d_{i_j} = [j\frac{k+a_i}{a_i}]$, $j = 1, \ldots, a_i$. In this way, the distances for queue i are regularly placed around their mean $\frac{k+a_i}{a_i}$. After computing these distances d_{i_j}, the indices still can be inserted in various ways into S_{i-1}. To illustrate, if $S_2 = \{1, 1, 2\}$ and $a_3 = 1$, then there are three different patterns to choose S_3 from: {3,1,1,2}, {1,3,1,2} and {1,1,3,2}. In this example, the insertion can start from three different points in S_2. In general, there can be k different ways of inserting, creating possible new subpatterns $S_i^1, \ldots, S_i^k$. From these patterns, S_i is chosen such that $V(S_i) = \min_{1 \leq j \leq k} V(S_i^j)$.

We see that in the ith step of phase 1, index i is optimally placed in the subpattern. However, this optimality could be ruffled in subsequent iteration steps. In phase 2 therefore a local search method is applied, trying to restore some of the regularity.

Phase 2:
In the local search S_N is replaced by $S'_N(k,l)$ if $V(S'_N(k,l)) < V(S_N)$, where $S'_N(k,l) = S_N$ except for entries k and l, which in $S'_N(k,l)$ are interchanged compared to S_N. This local search is repeated until no further improvements can be made.

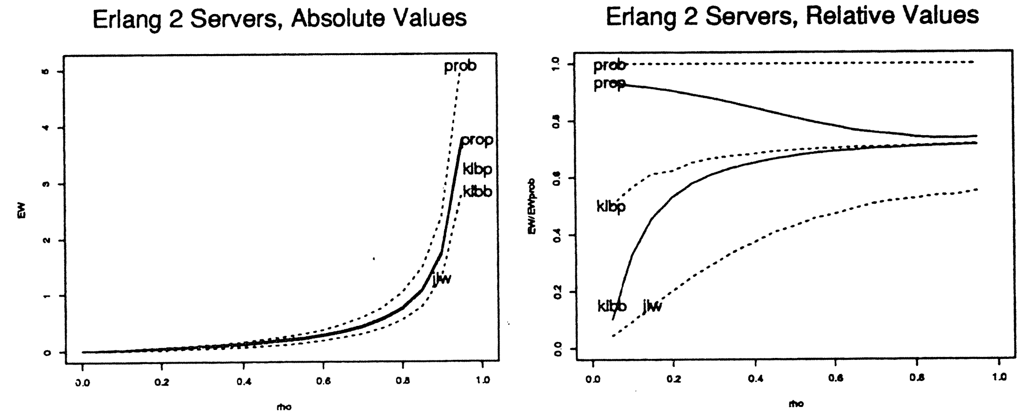
Remark C.1: For the first phase also the Golden Ratio method, as described by Itai & Rosberg[12], could have been applied. In the Golden Ratio method the allocation pattern is built in one step. In short, Golden Ratio operates as follows. Let $\Phi^{-1} = \frac{1}{2}(\sqrt{5} - 1) =$ 0.618034.... (Φ^{-1} is also known as the Golden Ratio). Put the $M = \sum_i a_i$ numbers $\Phi^{-1} \pmod 1), 2\Phi^{-1} \pmod 1), \ldots, M\Phi^{-1} \pmod 1)$ in increasing order. This creates a table L in which $L(i)$ corresponds to the ranking of $i\Phi^{-1} \pmod 1)$. For the allocation pattern, index 1 is assigned to $s_{L(1)}, \ldots, s_{L(a_1)}$, index 2 to $s_{L(a_1+1)}, \ldots, s_{L(a_1+a_2)}$, etc.
The Golden Ratio method (without the local search applied afterwards) possesses a number of interesting properties, some of which can be found in Itai & Rosberg[12]. For instance, in the allocation pattern at most three different values do occur for the distances d_{i_j} for each i. However, these values do not necessarily have to be close to the mean distance. The local search method does improve the Golden Ratio pattern, but in general the above-described heuristic based on Hajek[10] performs better. Our numerical experience shows that the latter method provides a pattern S for which the objective function $V(S)$ lies between 0 and 5 percent of the theoretical minimum, whereas Golden Ratio's relative error is in most cases between 2 and 4 times as high.

D Figures of numerical results


Exponential Servers, Absolute Values

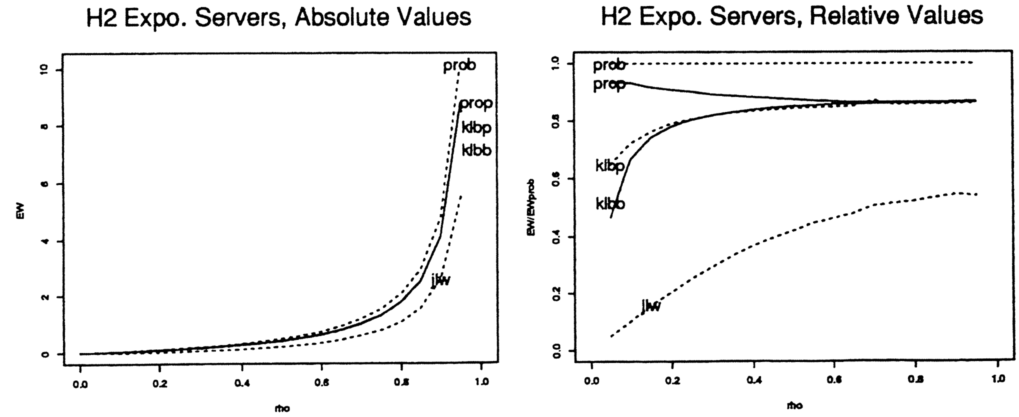
Exponential Servers, Relative Values

Figure 5.1: Two exponential servers, with service rate 1 and 4 respectively.


Erlang 2 Servers, Absolute Values

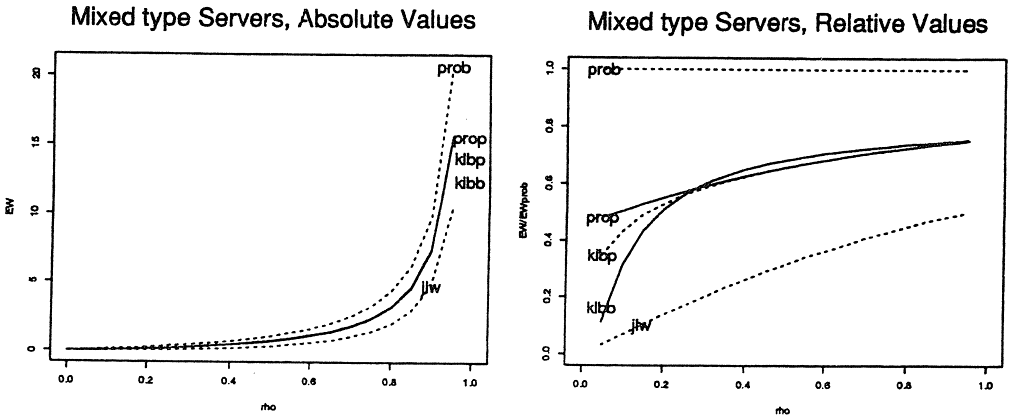
Erlang 2 Servers, Relative Values

Figure 5.2: Two Erlang 2 servers, with service rate 1 and 4 respectively.


H2 Expo. Servers, Absolute Values

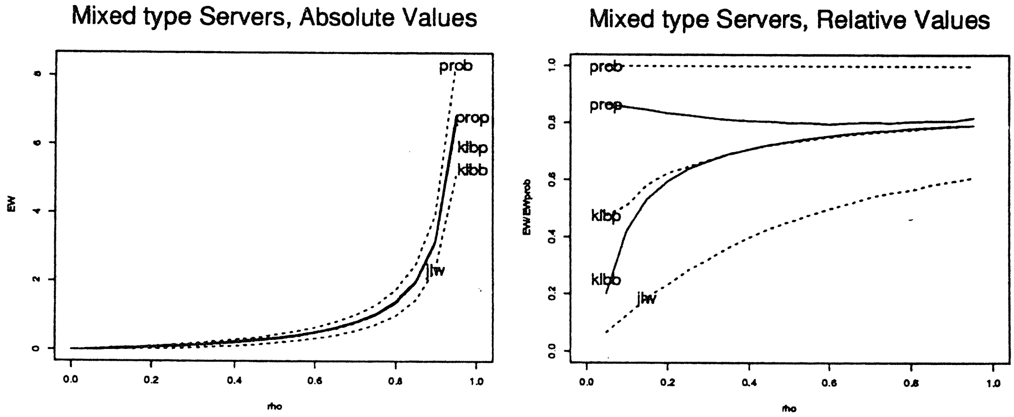
H2 Expo. Servers, Relative Values

Figure 5.3: Two Hyper-Exponential servers, with service rate 1 and 4 respectively. The service time of a customer is with probability $q = \frac{1}{3}$ exponentially distributed with parameter $\frac{1}{2}\mu$, and has with probability $1 - q = \frac{2}{3}$ an exponential distribution with parameter 2μ. In this way the service rate is μ and the coefficient of variation is 2.


Mixed type Servers, Absolute Values

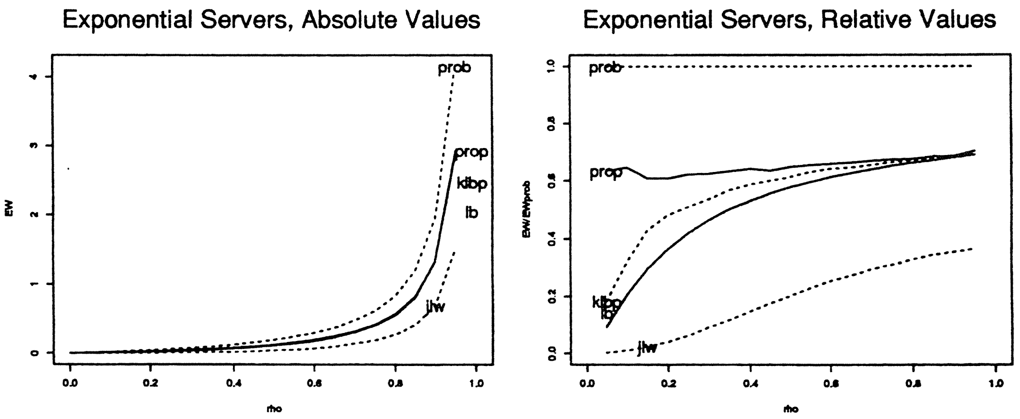
Mixed type Servers, Relative Values

Figure 5.4: Two servers with service rate 1. The first server has Erlang 2 distributed service times, the second server has Hyper-Exponentially distributed service times as described by figure 5.3.


Mixed type Servers, Absolute Values

Mixed type Servers, Relative Values

Figure 5.5: Two servers. The First server has Erlang 2 distributed service times with rate 1. The second server has Hyper-Exponentially distributed service time as described by figure 5.3.


Exponential Servers, Absolute Values

Exponential Servers, Relative Values

Figure 5.6: Three exponential servers, with service rates 1,4 and 7 respectively.

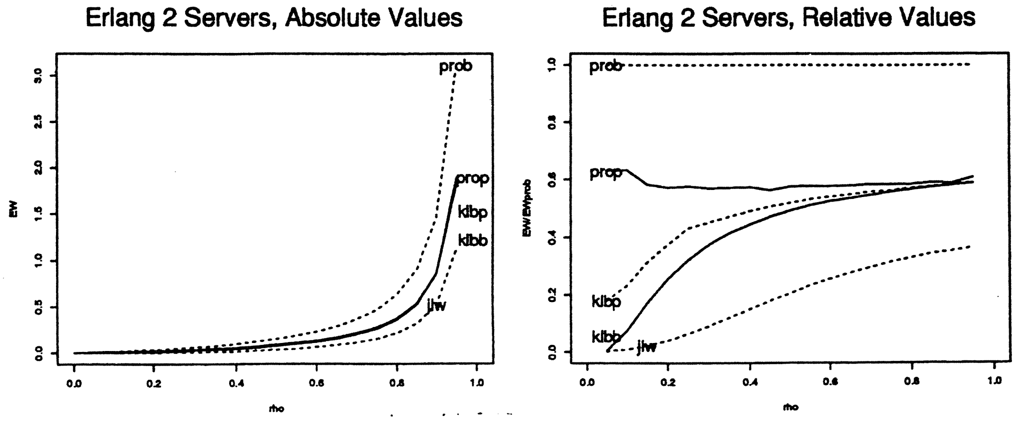


Figure 5.7: Three Erlang 2 servers, with service rates 1,4 and 7 respectively.

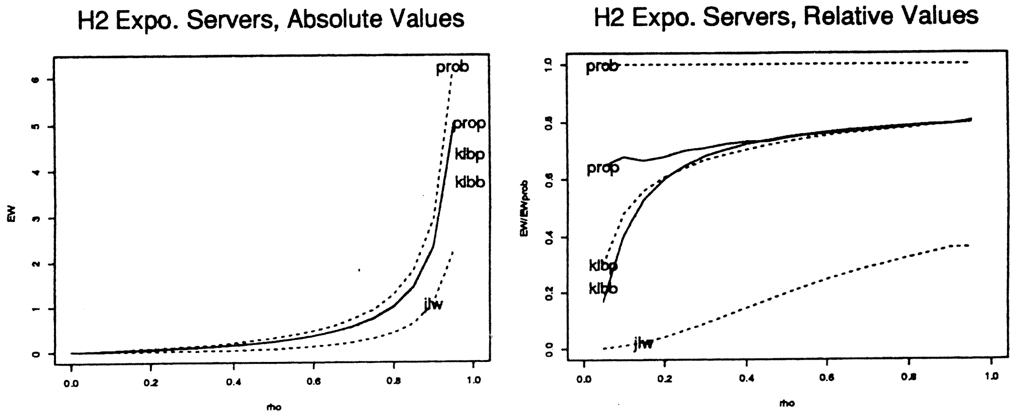


Figure 5.8: Three Hyper-Exponential servers, with service rates 1,4 and 7 respectively. Service time distributions as described by figure 5.3 .