Centrum voor Wiskunde en Informatica

REPORTRAPPORT

Optimal probabilistic allocation of customer types to servers

S.C. Borst

Department of Operations Research, Statistics, and System Theory

Report BS-R9415 May 1994

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

Optimal Probabilistic Allocation of Customer Types to Servers

S.C. Borst
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

Abstract

The model under consideration consists of n customer types attended to by m parallel non-identical servers. Customers are allocated to the servers in a probabilistic manner; upon arrival customers are sent to one of the servers according to an m x n matrix of routing probabilities. We consider the problem of finding an allocation that minimizes a weighted sum of the mean waiting times. We expose the structure of an optimal allocation and describe for some special cases in detail how the structure may be exploited in actually determining an optimal allocation. Further we consider the problem of finding an optimal deterministic allocation, i.e., an optimal allocation that involves a 0-1 matrix of routing probabilities. We show the problem to be NP-hard and indicate how the structure of an optimal non-deterministic allocation may be used as a heuristic guideline in searching for an optimal deterministic allocation.

AMS Subject Classification (1991): 60K25, 68M20, 90B22, 90B35.
Keywords & Phrases: load balancing, load sharing, random splitting, source partitioning, stochastic scheduling, waiting-cost minimization.
Note: the work was supported in part by the European Grant BRA-QMIPS of CEC DG XIII.

1 Introduction

In this paper we consider a system in which there are several parallel servers to process jobs generated at several distinct sources. Such a system arises quite naturally in modelling situations where a pool of resources is available to perform various kinds of activities. Examples may be found in distributed computer systems, flexible manufacturing systems, and telecommunication networks. Another example may be found in a situation where a pool of repair crews is available to perform maintenance activities at various installations.

In such a system, in which there are several servers to process jobs generated at several sources, usually some freedom of decision exists as to which server is to process which job at what time. Thus the need arises for a scheduling strategy, i.e., a collection of decision instructions for scheduling the jobs. At a global level decisions need to be made about which server is to process which job. Subsequently at a local level decisions need to be made about the order of service. In the present paper we mainly focus on the global scheduling problem; we hardly touch on the local scheduling problem. Locally, the order of service is assumed to not discriminate between the sources from which the jobs originated.

The main function of a global scheduling strategy is load sharing; a strategy should make the servers cooperate in sharing the load of the system so as to optimize the system performance.

Report BS-R9415
ISSN 0924-0659
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

Load sharing is also frequently referred to as load balancing. The term load balancing arises from the intuition that to optimize the system performance the load should be balanced among the servers. In the present paper we find however that the load, in the sense of traffic intensity, in general should *not* be completely balanced.

Wang & Morris [16] give a comprehensive survey of the overwhelming variety of approaches to load sharing in the literature. They identify some fundamentally distinguishing features of load sharing strategies. A first distinction refers to the side that takes the initiative in scheduling the jobs. In source-initiative policies a source decides to which server to route a job. In server-initiative policies a server decides from which source to get a job. Consequently in source-initiative policies decisions are typically made at arrival epochs, whereas in server-initiative policies decisions are typically made at service completion epochs (possibly at arrival epochs when servers are idle). Moreover, in source-initiative policies queues tend to form at the servers, whereas in server-initiative policies queues tend to form at the sources. Of course there are also policies conceivable in which both the sources and the servers participate in allocating the jobs.

A second distinction refers to the amount of information that is used in allocating the jobs. In purely static policies only information is used about the basic characteristics of the system, like the traffic intensities. In dynamic policies also information is used about the actual state of the system, like the queue lengths. Evidently, in principle the performance of the system may improve substantially by using such information in allocating the jobs. However, gathering such information and implementing a sophisticated dynamic allocation strategy may involve a considerable communication overhead and complicate the operation of the system significantly. Therefore dynamic policies are not necessarily preferable to static policies.

In the present paper we assume that customers are allocated to the servers in a probabilistic manner; upon arrival customers are sent to one of the servers according to a matrix of routing probabilities. Such a load sharing strategy is commonly referred to as random splitting. In the taxonomy of Wang & Morris [16] random splitting belongs to the class of static source-initiative load sharing strategies. We are interested in the problem of finding a random splitting that minimizes a weighted sum of the mean waiting times.

The novelty of the model lies in the combination of heterogeneous servers (i.e. different service rates), heterogeneous sources (i.e. different service times), and a fairly general cost function. Buzacott & Shanthikumar [3] consider a version of the problem with homogeneous servers and the overall mean waiting time as performance measure, which we will discuss later on in greater technical detail. Buzen & Chen [4] consider a variant of the problem with a single source and the overall mean sojourn time as performance criterion. A natural approach to deal with heterogeneous servers and heterogeneous sources might be to aggregate the different sources into a single source, and then use the results of Buzen & Chen. Each server would thus handle a traffic mix of the same, heterogeneous, composition, but of possibly different intensity, depending on the processing rate of the servers. In the present paper we find however that *each server should handle a traffic mix as homogeneous as possible.* For homogeneous sources Boxma & Combé [2] show that 'pattern' allocation outperforms probabilistic allocation, but that the optimal routing probabilities of Buzen & Chen provide a reasonable indication for the optimal occurrence fractions in 'pattern' allocation. It is likely that similar observations hold for heterogeneous sources.

Tantawi & Towsley [12], [13] and De Souza e Silva & Gerla [7] consider optimal load balancing models of distributed computer systems consisting of a number of heterogeneous host computers connected by a communication network. A job may be either processed at the host to which it arrives or transferred to another host. In the latter case, a transferred job incurs a communication delay in addition to the queueing delay at the host on which it is processed. The assumptions in [12], [13], and [7] on the service requirements of jobs are however somewhat restrictive.

In some situations it may be desirable that job classes are not split among different servers. In flexible manufacturing systems e.g. such a splitting may be undesirable because handling a particular job class usually requires special expensive tools. In case job classes are not split among different servers, a load sharing strategy is commonly referred to as source partitioning. In view of the practical relevance we are specifically interested in finding an optimal source partitioning. For general partitioning problems Anily & Federgruen [1] identify analytical properties of the cost function under which an optimal solution has a simple structure, thus allowing for a simple solution method. They mention the problem of finding an optimal source partitioning as an example for which the mean number of waiting customers as cost function (which by Little's law is nothing but a specific weighted sum of the mean waiting times) does *not* have such analytical properties.

The remainder of the paper is organized as follows. In section 2 we present a detailed model description. We then consider the problem of finding an optimal random splitting. In section 3 we expose the structure of an optimal allocation and in section 4 we describe for some special cases in detail how the structure may be exploited in actually determining an optimal allocation. In section 5 we consider the problem of finding an optimal source partitioning. We show the problem to be NP-hard and indicate how the structure of an optimal non-deterministic allocation may be used as a heuristic guideline in searching for an optimal deterministic allocation. In section 6 we conclude with some remarks and suggestions for further research.

2 Model description

The model under consideration consists of n customer types attended to by m parallel non-identical servers. Customers arrive according to Poisson processes. The arrival rate of type-j customers is λ_j, $j = 1, \ldots, n$. The total arrival rate is $\lambda := \sum_{j=1}^{n} \lambda_j$. Upon arrival customers are routed to one of the servers. Type-j customers are routed to server i with probability x_{ij}, $i = 1, \ldots, m$, $j = 1, \ldots, n$. The matrix $x = (x_{ij})$ of routing probabilities will be referred to as the (probabilistic) allocation of the customer types to the servers. In case the matrix of routing probabilities is 0-1 the (probabilistic) allocation will be referred to as deterministic. When processed at server i, type-j customers require service times having distribution $F_{ij}(t) = F_j(\mu_i t)$, i.e., type-j customers require amounts of service having distribution $F_j(t)$, while server i has processing rate μ_i. In other words, the servers may have different characteristics, the customer types may have different characteristics, but servers cannot 'specialize' in some of the customer types. Denote by β_j and $\beta_j^{(2)}$ the first and second moment of $F_j(t)$, $j = 1, \ldots, n$. We assume $F_j(0) < 1$, so $\beta_j > 0$, $\beta_j^{(2)} > 0$, $j = 1, \ldots, n$. Define the traffic intensity associated with type-j customers as $\rho_j := \lambda_j \beta_j$, $j = 1, \ldots, n$. The total traffic intensity is $\rho := \sum_{j=1}^{n} \rho_j$. The order of service is assumed to not discriminate between the various customer types. Further all arrival, service, and routing processes are assumed to be mutually independent.

The queues that form at the servers are ordinary $M/G/1$ queues. The arrival rate at server i is $\sum_{j=1}^{n} x_{ij}\lambda_j$. Customers that are routed to server i require service times having distribution $\sum_{j=1}^{n} x_{ij}\lambda_j F_j(\mu_i t) / \sum_{j=1}^{n} x_{ij}\lambda_j$ with first moment $\left[\sum_{j=1}^{n} x_{ij}\lambda_j\beta_j\right] / \left[\mu_i \sum_{j=1}^{n} x_{ij}\lambda_j\right]$ and second moment $\left[\sum_{j=1}^{n} x_{ij}\lambda_j\beta_j^{(2)}\right] / \left[\mu_i^2 \sum_{j=1}^{n} x_{ij}\lambda_j\right]$, $i = 1, \ldots, m$. Define the traffic intensity at server i as $\sum_{j=1}^{n} x_{ij}\lambda_j\beta_j$, $i = 1, \ldots, m$. Necessary and sufficient ergodicity conditions are

$$\sum_{j=1}^{n} x_{ij}\lambda_j\beta_j < \mu_i, \qquad i = 1, \ldots, m. \tag{2.1}$$

Denote by $\mu = \sum_{i=1}^{m} \mu_i$ the total processing rate of the servers. Summing (2.1) with respect to $i = 1, \ldots, m$ yields $\rho < \mu$, as $\sum_{i=1}^{m} x_{ij} = 1$, $j = 1, \ldots, n$. Throughout the paper $\rho < \mu$ is assumed to hold.

We are interested in the problem of finding an allocation that minimizes a weighted sum of the mean waiting times. Therefore we first derive a formula that describes the mean waiting times as function of the allocation matrix $x = (x_{ij})$. Denote by $\mathbf{W}_j$ the waiting time of an arbitrary type-j customer, i.e., the time from its arrival to the start of its service. Denote by $\mathbf{V}_i$ the waiting time of an arbitrary customer that is routed to server i.

As the order of service is assumed to not discriminate between the various customer types,

$$\mathrm{E}\mathbf{W}_j = \sum_{i=1}^{m} x_{ij}\mathrm{E}\mathbf{V}_i, \qquad j = 1, \ldots, n. \tag{2.2}$$

As the queues that form at the servers are ordinary $M/G/1$ queues,

$$\mathrm{E}\mathbf{V}_i = \frac{\sum_{j=1}^{n} \lambda_j\beta_j^{(2)} x_{ij}}{2\mu_i\left(\mu_i - \sum_{j=1}^{n} \lambda_j\beta_j x_{ij}\right)}, \qquad i = 1, \ldots, m. \tag{2.3}$$

Let c_j represent the waiting cost per unit of time of a type-j customer, $j = 1, \ldots, n$. We assume $c_j > 0$, $j = 1, \ldots, n$. The mean total waiting cost per unit of time amounts to $\sum_{j=1}^{n} c_j\lambda_j\mathrm{E}\mathbf{W}_j$. Using (2.2) and (2.3),

$$\sum_{j=1}^{n} c_j\lambda_j\mathrm{E}\mathbf{W}_j = \sum_{i=1}^{m} \frac{\left(\sum_{j=1}^{n} \lambda_j c_j x_{ij}\right)\left(\sum_{j=1}^{n} \lambda_j\beta_j^{(2)} x_{ij}\right)}{2\mu_i\left(\mu_i - \sum_{j=1}^{n} \lambda_j\beta_j x_{ij}\right)}. \tag{2.4}$$

3 Finding an optimal random splitting

In the present section we consider the problem of finding an optimal random splitting, i.e., a probabilistic allocation of the customer types to the servers that minimizes the mean total waiting cost per unit of time. Using (2.1) and (2.4), we formulate the problem as follows.

Problem (I).

$$\text{minimize} \qquad f(x) = \sum_{i=1}^{m} \frac{\left(\sum_{j=1}^{n} \lambda_j c_j x_{ij}\right)\left(\sum_{j=1}^{n} \lambda_j \beta_j^{(2)} x_{ij}\right)}{\mu_i \left(\mu_i - \sum_{j=1}^{n} \lambda_j \beta_j x_{ij}\right)} \tag{3.1}$$

$$\begin{aligned}\text{subject to} \qquad & \sum_{j=1}^{n} \lambda_j \beta_j x_{ij} < \mu_i, \qquad i = 1, \ldots, m;\\ & \sum_{i=1}^{m} x_{ij} = 1, \qquad j = 1, \ldots, n;\\ & x_{ij} \geq 0, \qquad i = 1, \ldots, m,\ j = 1, \ldots, n.\end{aligned}$$

Problem (I) is a non-linear programming problem. It is easily verified that the objective function $f(\cdot)$ is not convex, so that it is not guaranteed that there exists a unique Kuhn-Tucker point. Moreover, finding a Kuhn-Tucker point is not quite straightforward.
All in all there is not an obvious way of solving problem (I). Nevertheless, if one is purely interested in computing an optimal allocation for some given parameters, then one might in principle proceed to solving problem (I) by standard non-convex programming techniques. That is however not what we are interested in here. What we are primarily interested in, is obtaining some insight into the structural properties of an optimal allocation. We will show that an optimal solution of problem (I) indeed exhibits a very characteristic structure. As secondary motivation, the structural properties do not only provide some insight, but are also very useful in computing an optimal allocation. Specifically we will describe in section 4 how in cases with identical servers where all the customer types are in a sense ordered, the structure may be exploited in a very simple manner in actually determining an optimal solution of problem (I). In these cases there exists a unique Kuhn-Tucker point exhibiting the structure of an optimal solution. So it is guaranteed that this Kuhn-Tucker point *is* the optimal solution. Moreover, finding this Kuhn-Tucker point is comparatively straightforward in these cases. In cases where not all the customer types are ordered, the structure may still be exploited in actually determining an optimal solution of problem (I), but not in a manner as simple. In section 5 we indicate how the knowledge of the structure of an optimal solution of problem (I) may also be used as a guideline in heuristically solving the NP-hard *integer* version (integer x_{ij}'s) of problem (I).

We now expose the structure of an optimal allocation x^*. We first introduce some notation. For a given allocation x, define $K_i(x) = \{j \mid x_{ij} > 0\}$ to be the index set of the customer types (partially) allocated to server i. Define $A_i(x) = \left\{ \left(\frac{c_j}{\beta_j}, \frac{\beta_j^{(2)}}{\beta_j} \right) \mid j \in K_i(x) \right\}$ to be the set of $\left(\frac{c_j}{\beta_j}, \frac{\beta_j^{(2)}}{\beta_j}\right)$-values corresponding to the customer types allocated to server i. Denote $P_i(x) = \text{int}(\text{conv}(A_i(x)))$, with int(conv(·)) denoting the interior of the convex hull. The set $P_i(x)$ may be interpreted as the global range of $\left(\frac{c_j}{\beta_j}, \frac{\beta_j^{(2)}}{\beta_j}\right)$-values corresponding to the customer types allocated to server i. Denote $B_i(x) = \left[\sum_{j=1}^{n} \lambda_j \beta_j^{(2)} x_{ij}\right] / \left[\mu_i \left(\mu_i - \sum_{j=1}^{n} \lambda_j \beta_j x_{ij}\right)\right]$, $C_i(x) = \left[\sum_{j=1}^{n} \lambda_j c_j x_{ij}\right] / \left[\mu_i \left(\mu_i - \sum_{j=1}^{n} \lambda_j \beta_j x_{ij}\right)\right]$. The numbers $B_i(x)$ and $C_i(x)$ may be interpreted as measures for the '$\beta_j^{(2)}/\beta_j$-weight' and the 'c_j/β_j-weight' associated with the customer types allocated to server i.

We now expose the structure of an optimal allocation x^* in terms of the corresponding sets $P_i(x^*)$. Intuitively, it is to be expected that an optimal allocation will satisfy one of the following two (in general mutually exclusive) 'extremal' properties.

(i). Each server handles a traffic mix of the same composition (e.g. $x_{ij}^* = \mu_i/\mu$, $i = 1, \ldots, m$, $j = 1, \ldots, n$), so that the traffic mix at each server is completely heterogeneous;

(ii). Each server handles a traffic mix as homogeneous as possible, so that different servers deal with traffic mixes of a completely different composition.

The next Lemma says that an optimal allocation in fact satisfies the second property (so the first one not in general).

Lemma 3.1

$P_{i'}(x^*) \cap P_{i''}(x^*) = \emptyset$ for $i' \neq i''$.

In other words, if $\left(\frac{c_j}{\beta_j}, \frac{\beta_j^{(2)}}{\beta_j}\right) \in P_i(x^*)$, then $x_{ij}^* = 1$.

Proof

See appendix A.

□

Lemma 3.1 suggests that the customer types should be clustered according to the corresponding $\left(\frac{c_j}{\beta_j}, \frac{\beta_j^{(2)}}{\beta_j}\right)$-values. As a consequence, different servers will deal with different traffic mixes. The next Lemma says that different traffic mixes may however not involve an arbitrarily different '$\beta_j^{(2)}/\beta_j$-weight' and 'c_j/β_j-weight'.

Lemma 3.2

If $B_{i'}(x^*) \geq B_{i''}(x^*)$, $C_{i'}(x^*) \geq C_{i''}(x^*)$, then $\mu_{i'} B_{i'}(x^*) C_{i'}(x^*) \leq \mu_{i''} B_{i''}(x^*) C_{i''}(x^*)$.

Proof

See appendix B.

□

Lemma 3.2 states that if one server carries both larger $B_i(x^*)$ and $C_i(x^*)$ than another, then it cannot carry larger $\mu_i B_i(x^*) C_i(x^*)$ as well.

In the remainder of this section as well as in the next section we consider the case of identical servers, i.e., $\mu_i = \mu/m$, $i = 1, \ldots, m$. Lemma 3.2 then states that it is no longer possible that one server carries both larger $B_i(x^*)$ and $C_i(x^*)$ than another.

Corollary 3.1
$B_{i'}(x^*) \geq B_{i''}(x^*) \Longleftrightarrow C_{i'}(x^*) \leq C_{i''}(x^*)$.

We now assume that the servers are indexed such that $B_{i'}(x^*) \geq B_{i''}(x^*)$, $C_{i'}(x^*) \leq C_{i''}(x^*)$ for $i' < i''$.

Lemma 3.3
Assume $x^*_{i'j'} > 0$, $x^*_{i''j''} > 0$.
If $\dfrac{c_{j'}}{\beta_{j'}} \leq \dfrac{c_{j''}}{\beta_{j''}}$, $\dfrac{\beta^{(2)}_{j'}}{\beta_{j'}} \geq \dfrac{\beta^{(2)}_{j''}}{\beta_{j''}}$, $\left(\dfrac{c_{j'}}{\beta_{j'}}, \dfrac{\beta^{(2)}_{j'}}{\beta_{j'}}\right) \neq \left(\dfrac{c_{j''}}{\beta_{j''}}, \dfrac{\beta^{(2)}_{j''}}{\beta_{j''}}\right)$, then $i' \leq i''$.

Proof
See appendix C.

□

Lemma 3.3 states that expensive, calm (cheap, wild) customer types with large (small) c_j/β_j and small (large) $\beta^{(2)}_j/\beta_j$ should be sent to servers with small (large) $B_i(x^*)$ and large (small) $C_i(x^*)$, thus experiencing a small (large) waiting time. (Note that $B_i(x)$ is in fact twice the mean waiting time at server i for allocation x.) Lemma 3.3 does however not indicate what should be done with expensive but wild (cheap but calm) customer types with large (small) c_j/β_j and large (small) $\beta^{(2)}_j/\beta_j$. Indeed, it depends not only on their own individual c_j/β_j and $\beta^{(2)}_j/\beta_j$ but also on some other less seizable factors whether they should be sent to servers with small $B_i(x^*)$ and large $C_i(x^*)$ or with large $B_i(x^*)$ and small $C_i(x^*)$.
Lemma 3.3 allows us to strengthen the statements on the clustering of the customer types in Lemma 3.1. We first introduce some additional notation. Define

$$Q_i(x) = \bigcup_{j \in K_i(x)} \left\{ (y,z) : y \leq \frac{c_j}{\beta_j}, z \geq \frac{\beta^{(2)}_j}{\beta_j}, (y,z) \neq \left(\frac{c_j}{\beta_j}, \frac{\beta^{(2)}_j}{\beta_j}\right) \right\},$$

$$R_i(x) = \bigcup_{j \in K_i(x)} \left\{ (y,z) : y \geq \frac{c_j}{\beta_j}, z \leq \frac{\beta^{(2)}_j}{\beta_j}, (y,z) \neq \left(\frac{c_j}{\beta_j}, \frac{\beta^{(2)}_j}{\beta_j}\right) \right\}.$$

Denote $S_i(x) = Q_i(x) \cap R_i(x)$, $T_i(x) = P_i(x) \cup S_i(x)$.

Lemma 3.4
$S_{i'}(x^*) \cap S_{i''}(x^*) = \emptyset$ for $i' \neq i''$.
In other words, if $\left(\frac{c_j}{\beta_j}, \frac{\beta_j^{(2)}}{\beta_j}\right) \in S_i(x^*)$, then $x^*_{ij} = 1$.

Proof
See appendix D.

□

Lemma 3.4 suggests that in the case of identical servers the customer types should be clustered according to the corresponding $\left(\frac{c_j}{\beta_j}, \frac{\beta_j^{(2)}}{\beta_j}\right)$-values in an even stronger sense than stated before in Lemma 3.1.

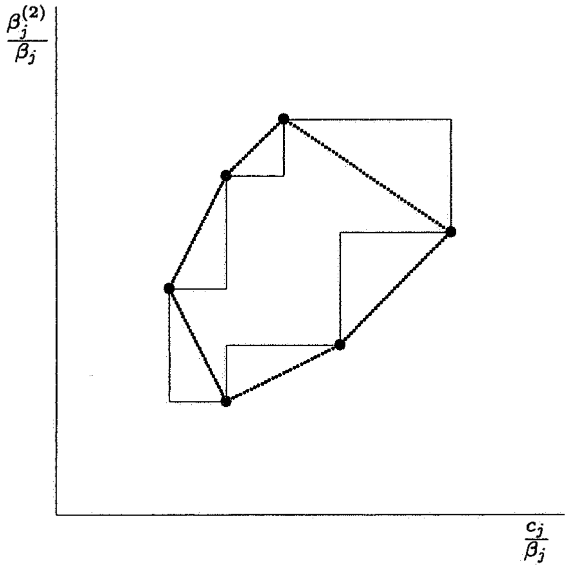

Figure 3.1 The sets $P_i(x)$ and $S_i(x)$.

It is easily verified from the definition of $P_i(x)$ and $S_i(x)$ that if $P_{i'}(x) \cap P_{i''}(x) = \emptyset$ and $S_{i'}(x) \cap S_{i''}(x) = \emptyset$, then also $P_{i'}(x) \cap S_{i''}(x) = \emptyset$, i.e., $T_{i'}(x) \cap T_{i''}(x) = \emptyset$, see Figure 3.1, where the bold dots constitute the set $A_i(x)$. The area inside the dotted lines corresponds to the set $P_i(x)$. The rectangular area represents the set $S_i(x)$. So in the case of identical servers Lemma 3.1 and Lemma 3.4 may be summarized as follows.

Corollary 3.2
$T_{i'}(x^*) \cap T_{i''}(x^*) = \emptyset$ for $i' \neq i''$.
In other words, if $\left(\frac{c_j}{\beta_j}, \frac{\beta_j^{(2)}}{\beta_j}\right) \in T_i(x^*)$, then $x^*_{ij} = 1$.

The optimality of clustering as exposed in the previous Lemma's suggests that the optimal routing probabilities are almost all equal to either 0 or 1. Although the settings are quite different, the latter observation strongly reminds of the vertex-allocation theorem for the optimal routing of single customer chains in closed product-form networks, saying that each customer should consistently select the same server for each request type rather than choose probabilistically, cf. Tripathi & Woodside [14], Woodside & Tripathi [15], Cheng & Muntz [5].

In the next section we show how the structure, as characterized in the previous Lemma's, may be exploited in computing an optimal allocation.

4 The case of ordered customer types

In this section we show how the structure, as characterized in the Lemma's of the previous section, may be exploited in computing an optimal allocation. We make the following assumption.

Assumption 4.1

The customer types are ordered such that $\frac{c_{j'}}{\beta_{j'}} \leq \frac{c_{j''}}{\beta_{j''}}, \frac{\beta^{(2)}_{j'}}{\beta_{j'}} \geq \frac{\beta^{(2)}_{j''}}{\beta_{j''}}, \left(\frac{c_{j'}}{\beta_{j'}}, \frac{\beta^{(2)}_{j'}}{\beta_{j'}}\right) \neq \left(\frac{c_{j''}}{\beta_{j''}}, \frac{\beta^{(2)}_{j''}}{\beta_{j''}}\right)$ for $j' < j''$.

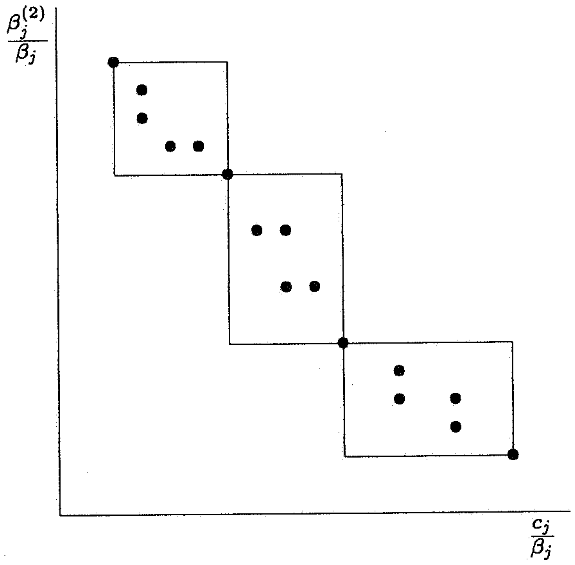

Figure 4.1 The case of ordered customer types.

Below we describe in detail how the structure may be exploited in actually determining an optimal allocation in cases where the customer types are ordered in the sense of Assumption 4.1. If Assumption 4.1 is satisfied, then Corollary 3.2 provides a very strong characterization of an optimal allocation, see Figure 4.1, where the rectangles represent the sets $T_i(x)$ for an instance with $m = 3$ servers and $n = 16$ customer types. The fact that the sets $T_i(x)$ do not intersect, completely determines their 'position', so that only the problem remains to determine their 'size'. In cases where the customer types are not ordered, the structure may still be exploited in actually determining an optimal allocation but not in a manner as simple.

Theoretically speaking, Assumption 4.1 is somewhat restrictive. However, there are several cases of practical interest that satisfy Assumption 4.1.

Case i. $c_j/\beta_j = \gamma$, $j = 1, \ldots, n$.
In other words, the waiting costs per unit of time are proportional to the mean service times. This is the case when the goal is minimizing the mean amount of waiting work, $\sum_{j=1}^{n} \rho_j \mathbf{E}\mathbf{W}_j$. Minimizing the mean amount of waiting work is equivalent to minimizing the mean total amount of work, as the difference, the mean amount of work in service always equals $\frac{1}{2}\sum_{j=1}^{n} \lambda_j \beta_j^{(2)}$, irrespective of the allocation x. When the customer types have the same mean service time, minimizing the mean amount of waiting work is also equivalent to minimizing the overall mean waiting time.
For $c_j/\beta_j = \gamma$, $j = 1, \ldots, n$, Corollary 3.1 reduces to

$$\frac{\sum_{j=1}^{n} \lambda_j \beta_j^{(2)} x^*_{i'j}}{\mu/m - \sum_{j=1}^{n} \lambda_j \beta_j x^*_{i'j}} \geq (\leq) \frac{\sum_{j=1}^{n} \lambda_j \beta_j^{(2)} x^*_{i''j}}{\mu/m - \sum_{j=1}^{n} \lambda_j \beta_j x^*_{i''j}} \iff \sum_{j=1}^{n} \lambda_j \beta_j x^*_{i'j} \leq (\geq) \sum_{j=1}^{n} \lambda_j \beta_j x^*_{i''j}.$$

In particular,

$$\sum_{j=1}^{n} \lambda_j \beta_j x^*_{i'j} = \sum_{j=1}^{n} \lambda_j \beta_j x^*_{i''j} \iff \sum_{j=1}^{n} \lambda_j \beta_j^{(2)} x^*_{i'j} = \sum_{j=1}^{n} \lambda_j \beta_j^{(2)} x^*_{i''j}. \tag{4.1}$$

For $c_j/\beta_j = \gamma$, $j = 1, \ldots, n$, the set $T_i(x)$ reduces to the line-segment

$$\left\{ (\gamma, z) \mid \min_{j \in K_i(x)} \beta_j^{(2)}/\beta_j < z < \max_{j \in K_i(x)} \beta_j^{(2)}/\beta_j \right\}.$$

Thus Corollary 3.2 says that the customer types should be clustered according to the corresponding $\beta_j^{(2)}/\beta_j$-values. This means that

$$\left(\sum_{j=1}^{n} \lambda_j \beta_j x^*_{ij}, \sum_{j=1}^{n} \lambda_j \beta_j^{(2)} x^*_{ij} \right) = \left(\frac{1}{m} \sum_{j=1}^{n} \lambda_j \beta_j, \frac{1}{m} \sum_{j=1}^{n} \lambda_j \beta_j^{(2)} \right), \qquad i = 1, \ldots, m,$$

will only hold when all the customer types do not only have the same $c_j/\beta_j = \gamma$, but also happen to have the same $\beta_j^{(2)}/\beta_j$, which in general is not the case. In view of (4.1) we may thus conclude that in general $\sum_{j=1}^{n} \lambda_j \beta_j x^*_{i'j} \neq \sum_{j=1}^{n} \lambda_j \beta_j x^*_{i''j}$, i.e., the total load should *not* be completely balanced.

Case ii. $\beta_j^{(2)}/\beta_j = \delta,\ j = 1, \ldots, n.$
In other words, the mean residual service times are constant. This is the case when the customer types have the same service time characteristics, but different priorities reflected in different waiting costs per unit of time.
For $\beta_j^{(2)}/\beta_j = \delta$, $j = 1, \ldots, n$, Corollary 3.1 reduces to

$$\sum_{j=1}^{n} \lambda_j \beta_j x_{i'j}^* \geq (\leq) \sum_{j=1}^{n} \lambda_j \beta_j x_{i''j}^* \Longleftrightarrow \frac{\sum_{j=1}^{n} \lambda_j c_j x_{i'j}^*}{\mu/m - \sum_{j=1}^{n} \lambda_j \beta_j x_{i'j}^*} \leq (\geq) \frac{\sum_{j=1}^{n} \lambda_j c_j x_{i''j}^*}{\mu/m - \sum_{j=1}^{n} \lambda_j \beta_j x_{i''j}^*}.$$

In particular,

$$\sum_{j=1}^{n} \lambda_j \beta_j x_{i'j}^* = \sum_{j=1}^{n} \lambda_j \beta_j x_{i''j}^* \Longleftrightarrow \sum_{j=1}^{n} \lambda_j c_j x_{i'j}^* = \sum_{j=1}^{n} \lambda_j c_j x_{i''j}^*. \tag{4.2}$$

For $\beta_j^{(2)}/\beta_j = \delta$, $j = 1, \ldots, n$, the set $T_i(x)$ reduces to the line-segment

$$\left\{ (y, \delta) \mid \min_{j \in K_i(x)} c_j/\beta_j < y < \max_{j \in K_i(x)} c_j/\beta_j \right\}.$$

Thus Corollary 3.2 says that the customer types should be clustered according to the corresponding c_j/β_j-values (which strongly reminds of the $c\mu$-rule, cf. [11], although the $c\mu$-rule in fact refers to the order of service of customer types rather than the clustering of customer types). This implies that

$$\left(\sum_{j=1}^{n} \lambda_j \beta_j x_{ij}^*, \sum_{j=1}^{n} \lambda_j c_j x_{ij}^* \right) = \left(\frac{1}{m} \sum_{j=1}^{n} \lambda_j \beta_j, \frac{1}{m} \sum_{j=1}^{n} \lambda_j c_j \right), \qquad i = 1, \ldots, m,$$

will only hold when all the customer types do not only have the same $\beta_j^{(2)}/\beta_j = \delta$, but also happen to have the same c_j/β_j, which in general is not the case. In view of (4.2) we may thus again conclude that in general $\sum_{j=1}^{n} \lambda_j \beta_j x_{i'j}^* \neq \sum_{j=1}^{n} \lambda_j \beta_j x_{i''j}^*$, i.e., the total load should *not* be completely balanced.

Case iii. $c_j = c,\ j = 1, \ldots, n,\ \beta_{j'} \leq \beta_{j''} \Longleftrightarrow \beta_{j'}^{(2)}/\beta_{j'} \leq \beta_{j''}^{(2)}/\beta_{j''}.$
In other words, the waiting costs per unit of time are constant. This is the case when the goal is minimizing the overall mean waiting time. Moreover, a larger mean service time corresponds to a larger mean residual service time. For the majority of service time distributions this is indeed the case. Corollary 3.2 then says that the customer types should be clustered according to the corresponding β_j-values. Again it may verified that the total load should *not* be completely balanced, unless the values of λ_j, β_j, $\beta_j^{(2)}$ of the customer types happen to satisfy some very specific relationships.

Remark 4.1 Buzacott & Shanthikumar [3] consider the problem of finding an optimal allocation in the case $c_j = c = 1, j = 1, \ldots, n$, i.e., the goal is minimizing the mean overall waiting time. In addition they require that the total load be balanced, i.e., $r_i^* = \sum_{j=1}^{n} \rho_j x_{ij} = \rho/m$, $i = 1, \ldots, m$. They show that if the agreeability condition $\beta_{j'} \leq \beta_{j''} \Longleftrightarrow \beta_{j'}^{(2)}/\beta_{j'} \leq \beta_{j''}^{(2)}/\beta_{j''}$ is satisfied, then the customer types should be clustered according to the corresponding β_j-values, as we also concluded in Case iii. above, *without* requiring that the total load be balanced. In fact we concluded in Case iii. that the total load should *not* be completely balanced.

□

We now describe a method for determining an optimal allocation in cases that satisfy Assumption 4.1. Here we sketch the main idea of the method. In appendix E we describe the method in detail.
From Lemma 3.3 we know that the structure of an optimal allocation is (i) $(x_{i1}^*, \ldots, x_{in}^*) = (0, \ldots, 0, x_{ij_i'}^*, 1, \ldots, 1, x_{ij_i''}^*, 0, \ldots, 0)$, with (ii) $\sum_{k=1}^{i} x_{kj_i'}^* = 1$, $\sum_{k=i}^{m} x_{kj_i''}^* = 1$. So an optimal allocation is then completely characterized by $s_i^* = \sum_{j=1}^{n} x_{ij}^*$, $i = 1, \ldots, m$. The global idea of the method is now to perform a kind of binary search with regard to s_1^*.
Step 1. Determine a lower and an upper bound for s_1^*.
Step 2. Make an estimate s_1 for s_1^*, somewhere in between lower and upper bound.
Step 3. Given s_i, determine s_{i+1}, for $i = 1, 2, \ldots, m-1$, from the knowledge of the structure of an optimal allocation (i), (ii), together with (iii) $\frac{\partial f(x)}{\partial x_{ij_i''}} = \frac{\partial f(x)}{\partial x_{i+1j_{i+1}'}}$. The latter condition necessarily holds for an optimal allocation, as may be formally verified from the Kuhn-Tucker conditions. In appendix E we show that through (i), (ii), (iii), s_i uniquely determines s_{i+1}, $i = 1, 2, \ldots m-1$.
Step 4. Sooner or later one either runs out of servers or out of customer types. If one runs out of servers, then apparently $s_1 < s_1^*$, so then replace the old lower bound by s_1. If one runs out of customer types, then apparently $s_1 > s_1^*$, so then replace the old upper bound by s_1. Repeat the procedure until lower and upper bound are sufficiently close.

We now consider some examples illustrating that in general the load should indeed not be completely balanced. We assume that there are $m = 4$ servers and $n = 4$ customer types. We take $(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = \alpha(8, 8, 1, 1)$, $(\beta_1, \beta_2, \beta_3, \beta_4) = (1, 2, 4, 8)$, so $(\rho_1, \rho_2, \rho_3, \rho_4) = 4\alpha(2, 4, 1, 2)$. We take successively $\alpha = 0.01$, $\alpha = 0.05$, $\alpha = 0.10$, $\alpha = 0.11$, so successively $\rho = 0.36$, $\rho = 1.80$, $\rho = 3.6$, $\rho = 3.96$. We assume that $\beta_j^{(2)} = \kappa \beta_j^2$, $j = 1, \ldots, n$, so that the agreeability condition is trivially satisfied. Note that κ influences the value of the objective function $f(\cdot)$ linearly, so that an optimal allocation is independent of κ. We compare the value of the objective function $f(\cdot)$ with $\kappa = 1$ for each of the following three allocations:
(i). x^S, the completely symmetric allocation, i.e., $x_{ij}^S = 1/m$, $i = 1, \ldots, m$, $j = 1, \ldots, n$;
(ii). x^B, the optimal allocation with $r_i^B = \sum_{j=1}^{n} \rho_j x_{ij}^B = \rho/m$, $i = 1, \ldots, m$;
(iii). x^*, the true optimal allocation computed by the method of appendix E;

as well as the $r_i^* = \sum_{j=1}^{n} \rho_j x_{ij}^*$ for the latter allocation. Table 4.1 contains the results for the case $c_j = c = 1$, $j = 1, \ldots, n$, like in Buzacott & Shanthikumar [3], i.e., $f(x)$ measures (twice) the overall mean waiting time for allocation x.

α	$f(x^S)$	$f(x^B)$	$f(x^*)$	r_1^*	r_2^*	r_3^*	r_4^*
0.01	0.05934	0.03747	0.03728	0.0876	0.0916	0.0986	0.0822
0.05	2.4545	1.5500	1.5468	0.4463	0.4502	0.4673	0.4362
0.10	54.100	34.100	34.086	0.8999	0.8980	0.9030	0.8992
0.11	653.40	412.61	412.46	0.9900	0.9897	0.9903	0.9900

Table 4.1 The value of the objective function for x^S, x^B, and x^* with $c_j = c = 1$.

Table 4.1 supports the conclusion that in general the load should not be completely balanced, but suggests that for $c_j = c$, $j = 1, \ldots, n$, the quality of x^B tends to match that of x^*. Table 4.2 contains the results for the case $c_j = \beta_j$, $j = 1, \ldots, n$, i.e., $f(x)$ measures (twice) the mean amount of waiting work for allocation x.

α	$f(x^S)$	$f(x^B)$	$f(x^*)$	r_1^*	r_2^*	r_3^*	r_4^*
0.01	0.11868	0.11868	0.10477	0.0553	0.0647	0.1140	0.1260
0.05	4.9091	4.9091	4.3542	0.3366	0.4105	0.5000	0.5528
0.10	108.00	108.00	94.941	0.8464	0.9024	0.9162	0.9351
0.11	1306.8	1306.8	1149.7	0.9840	0.9904	0.9917	0.9938

Table 4.2 The value of the objective function for x^S, x^B, and x^* with $c_j = \beta_j$.

Table 4.2 suggests that for $c_j \neq c$, $j = 1, \ldots, n$, the quality of x^B in comparison with x^* tends to deteriorate.

5 Finding an optimal source partitioning

In the present section we consider the problem of finding an optimal *source partitioning*, i.e., a *deterministic* allocation of the customer types to the servers that minimizes the mean total waiting cost per unit of time. Using (2.1) and (2.4), we formulate the problem as follows.

Problem (II).

$$\text{minimize} \qquad f(x) = \sum_{i=1}^{m} \frac{\left(\sum_{j=1}^{n} \lambda_j c_j x_{ij}\right)\left(\sum_{j=1}^{n} \lambda_j \beta_j^{(2)} x_{ij}\right)}{\mu_i \left(\mu_i - \sum_{j=1}^{n} \lambda_j \beta_j x_{ij}\right)} \tag{5.1}$$

$$\text{subject to} \quad \sum_{j=1}^{n} \lambda_j \beta_j x_{ij} < \mu_i, \qquad i = 1, \ldots, m;$$
$$\sum_{i=1}^{m} x_{ij} = 1, \qquad j = 1, \ldots, n;$$
$$x_{ij} \in \{0, 1\}, \qquad i = 1, \ldots, m, \; j = 1, \ldots, n.$$

Problem (II) is a non-linear integer programming problem. Finding a feasible solution of problem (II) is equivalent to packing n elements of size $\rho_1, \ldots, \rho_n$ in m bins of capacity $\mu_1, \ldots, \mu_m$, which is known to be NP-hard for $m \geq 2$, even in the case $\mu_i = \mu/m$, cf. Garey & Johnson [8]. Finding an optimal solution is of course at least as hard as finding a feasible solution of problem (II).

Concluding, even in the case of identical servers there is not likely to be an efficient way of solving problem (II), which motivates a heuristic approach. As even finding a feasible solution of problem (II) is NP-hard, we have to take for granted however that heuristic methods cannot be guaranteed to yield even a feasible solution. Globally speaking, the larger n and the larger $m - \rho$, the more feasible solutions there are and the more likely heuristic methods are to yield a feasible solution of problem (II). From a practical point of view, neither small n nor very small $m - \rho$ are of great concern with respect to solving problem (II). For small n the NP-hardness of problem (II) does not really matter, so that it be wiser to decide solving problem (II) by an enumerative method. For very small $m - \rho$ the system will be critically loaded anyway, so that it may be wiser in the design of the system to decide employing an extra server.

Below we indicate how the structure of an optimal solution of problem (I), the *non-integer* version (x_{ij}'s non-integer) of problem (II), may be used as a guideline in heuristically solving problem (II). To illustrate the specific complexity of problem (II), in comparison with problem (I), we first consider the special case of identical servers, i.e., $\mu_i = \mu/m$, $i = 1, \ldots, m$, and 'almost' identical customer types, i.e., $\left(\frac{c_j}{\beta_j}, \frac{\beta_j^{(2)}}{\beta_j}\right) = (\gamma, \delta)$, $j = 1, \ldots, n$.
Minimizing $f(x)$ then reduces to minimizing $g(x)$ defined as

$$g(x) = \sum_{i=1}^{m} \frac{\left(\sum\limits_{j=1}^{n} \rho_j x_{ij}\right)^2}{\mu/m - \sum\limits_{j=1}^{n} \rho_j x_{ij}}.$$

In a sense minimizing $g(x)$ amounts to making $\sum\limits_{j=1}^{n} \rho_j x_{ij}$ for all $i = 1, \ldots, m$ as equal as possible. In particular, if $\sum\limits_{j=1}^{n} \rho_j x_{ij} = \rho/m$ for all $i = 1, \ldots, m$, then $g(x)$ is certainly minimal, cf. Lemma 3.2. So an optimal solution of problem (I) is simply $x_{ij}^* = 1/m$, $i = 1, \ldots, m$, $j = 1, \ldots, n$. Due to the integrality conditions an optimal solution of problem (II) is however not so simple to obtain.

Minimizing $g(x)$ is strongly related to minimizing $h(x)$ defined as

$$h(x) = \max_{i=1,\ldots,m} \sum_{j=1}^{n} \rho_j x_{ij}.$$

Just like minimizing $g(x)$, minimizing $h(x)$ in a sense amounts to making $\sum_{j=1}^{n} \rho_j x_{ij}$ for all $i = 1, \ldots, m$ as equal as possible. In particular, if $\sum_{j=1}^{n} \rho_j x_{ij} = \rho/m$ for all $i = 1, \ldots, m$, then $h(x)$ is certainly minimal, again just like $g(x)$. For $m \leq 2$ minimizing $h(x)$ is completely equivalent to minimizing $g(x)$. In machine scheduling minimizing $h(x)$ is known as minimizing the makespan of n jobs of length $\rho_1, \ldots, \rho_n$ on m parallel identical machines, which is known to be NP-hard for $m \geq 2$, cf. Garey & Johnson [8]. (Of course this is not very surprising, knowing that packing is already NP-hard; minimizing the makespan is equivalent to packing the jobs in m bins of capacity as small as possible, which is of course at least as hard as packing the jobs in m bins of given capacity.)
A prominent family of heuristics for minimizing the makespan is the class of list scheduling rules. Characteristically of list scheduling rules, jobs are successively selected in order of appearance on some prespecified list, to be assigned to the machine with the least total processing time already assigned. The worst-case ratio for a list scheduling rule is $2-1/m$, cf. [9]. Evidently, the list ordering is the critical factor for the performance of a list scheduling rule. There is e.g. always a list ordering for which the list scheduling rule yields an optimal schedule. The worst-case instances suggest that the performance of a list scheduling rule may be better when jobs are selected in order of non-increasing ρ_j, which features the LPT (Longest Processing Time) rule. Indeed, the worst-case ratio for the LPT rule is $4/3 - 1/(3m)$, cf. [10]. Of course a worst-case ratio renders an inherently sombre picture, which not necessarily reflects the average performance. Probabilistic analysis reveals that the average performance of the LPT rule is indeed considerably better than the worst-case ratio may suggest, cf. [6]. Although minimizing $h(x)$ is closely related to minimizing $g(x)$, measures of the good performance with regard to $h(x)$ do not immediately carry over to measures of a good performance with regard to $g(x)$. The crux is that $g(x)$ is substantially more sensitive to suboptimality than $h(x)$; the larger the total traffic intensity, the more sensitive.

So far we considered the special case of identical servers and 'almost' identical customer types, in which it was easy to conclude that the total traffic stream should be balanced among the servers. Due to the integrality conditions it was however not so easy to accomplish an exactly balanced division. Returning now to the general case we cannot expect the picture to be brighter. On the contrary, in the general case it is not even known how exactly the total traffic stream should be divided among the servers, not to mention the problem of accomplishing a desirable division.
We now indicate how the structure of an optimal solution of problem (I) may be used as a guideline in heuristically solving problem (II). As the cost structure of problem (II) and problem (I) do not differ, there is no reason to believe that the structure of an optimal solution of problem (II) and problem (I) will dramatically differ. Globally speaking, the larger n and the larger $m - \rho$, the more feasible integer allocations there are and the more the structure of an optimal integer allocation is likely to resemble the structure of an optimal non-integer

allocation. As remarked before, from a practical point of view, neither small n nor very small $m - \rho$ are of great concern with respect to solving problem (II). There are several options as to how the knowledge of an optimal solution of problem (I), the non-integer version of problem (II), may be used in heuristically solving problem (II).

Option 1.
A first option is to construct an integer allocation, starting from an optimal non-integer allocation. One may e.g. somehow round an optimal non-integer allocation x^* to an integer allocation x^{**}, taking into account the condition $\sum_{i=1}^{m} x_{ij}^{**} = 1$, $j = 1, \ldots, n$, i.e., $x_{ij}^{**} = 1$ for $j = j_i$, $x_{ij}^{**} = 0$ for $j \neq j_i$, for some j_i with $x_{ij_i}^{*} > 0$, $i = 1, \ldots, m$. As observed before, the optimality of clustering suggests that an optimal non-integer allocation is in fact 'almost' integer. Thus rounding may be expected to yield quite acceptable results. The main drawback is that an optimal non-integer allocation is needed, which is not so simple to obtain in cases with non-identical servers or where the customer types are not ordered.

Option 2.
A second option, which to some extent meets the drawback of the first option, is to construct an integer allocation, not starting from an optimal allocation *itself*, but from the *structure*. One may e.g. select the best from all integer allocations that satisfy Lemma 3.1, i.e., $P_{i'}(x^*) \cap P_{i''}(x^*) = \emptyset$ for $i' \neq i''$. It is easily verified that the best of all integer allocations that satisfy Lemma 3.1 is at least as good as the best integer allocation obtained from rounding an optimal non-integer allocation. The main drawback is that such a procedure, although polynomial in n for fixed m, may prove to be rather time-consuming.

Option 3.
A third option, which to some extent meets the drawback of the second option, is to construct an integer allocation, again starting from the structure of an optimal integer allocation, but not so rigorously. One may e.g. select the best from not all but some proper subclass of integer allocations that satisfy Lemma 3.1, i.e., integer allocations that satisfy $U_{i'}(x) \cap U_{i''}(x) = \emptyset$ for $i' \neq i''$ for some $U_i(x) \supseteq P_i(x)$, $i = 1, \ldots, m$. One may define e.g. $U_i(x) = \left[\min_{j \in K_i(x)} c_j/\beta_j, \max_{j \in K_i(x)} c_j/\beta_j\right] \times \left[\min_{j \in K_i(x)} \beta_j^{(2)}/\beta_j, \max_{j \in K_i(x)} \beta_j^{(2)}/\beta_j\right]$, thus blowing up the sets $P_i(x)$ to rectangles, so that indeed $P_i(x) \subseteq U_i(x)$, $i = 1, \ldots, m$.

6 Concluding remarks and suggestions for further research

We considered the problem of finding an allocation that minimizes the mean total waiting cost per unit of time. We showed that the customer types should be clustered according to the corresponding $\left(\frac{c_j}{\beta_j}, \frac{\beta_j^{(2)}}{\beta_j}\right)$-values and we described for some special cases in detail how that property may be exploited in computing an optimal allocation. In other cases (e.g. when the customer types are not ordered in the sense of Assumption 4.1) that property may still be exploited in calculating an optimal allocation, but not in a manner as simple. An interesting topic for further research might be to develop efficient (heuristic) methods for this.

Further we considered the problem of finding an optimal deterministic allocation. We showed the problem to be NP-hard and indicated how the structure of an optimal non-deterministic allocation may be used as a heuristic guideline in searching for an optimal deterministic allocation. An interesting topic for further research might be to investigate the quality of the proposed heuristic methods, either in a theoretical framework or by numerical experiments. In either way, the quality of the heuristics may be judged by comparison with the optimal non-deterministic allocation which provides a lower bound for the optimal deterministic allocation.

In the present paper we focused on the global scheduling problem; we hardly touched on the local scheduling problem. We assumed the order of service to not discriminate between the various customer types. When the order of service *does* discriminate between the various customer types, the expressions for the mean waiting times are more complicated. However, for $c_j/\beta_j = \gamma$, $j = 1, \ldots, n$, for some γ, the results of the present paper still hold. Minimizing the mean total waiting cost per unit of time $\sum_{j=1}^{n} c_j \lambda_j \mathbf{EW}_j$ then amounts to minimizing the mean amount of waiting work $\sum_{j=1}^{n} \rho_j \mathbf{EW}_j$. Minimizing the mean amount of waiting work is equivalent to minimizing the mean total amount of work, as the difference, the mean amount of work in service, always equals $\frac{1}{2} \sum_{j=1}^{n} \lambda_j \beta_j^{(2)}$, irrespective of the allocation x. A sample path comparison shows however that the total amount of work is not influenced by the local scheduling strategy, which property is frequently referred to as work conservation. Consequently minimizing $\sum_{j=1}^{n} \rho_j \mathbf{EW}_j$ is completely insensitive to the local scheduling strategy.

In the present paper we implicitly assumed the various customer types to be served without any interruptions. In some situations changing over from one customer type to another may however require a non-negligible switch-over time (e.g., tool changing in a flexible manufacturing system, travelling in the case of a repair crew visiting various installations). When the switch-over times are non-negligible, the expressions for the mean waiting times are more complicated. For some local scheduling strategies so-called pseudo-conservation laws provide comparatively simple expressions for $\sum_{j=1}^{n} \rho_j \mathbf{EW}_j$, but these expressions appear to be less amenable to optimization procedures than the objective function (2.4).

In the present paper we assumed that any customer type could, in principle, be allocated to any server. In some situations it may however occur that some customer types cannot be allocated to some servers, or that some customer types cannot be combined. Such restrictions may be translated into additional constraints on the routing probabilities, the x_{ij}'s. It would be interesting to investigate how those restrictions on the x_{ij}'s affect the structure of an optimal allocation.

Acknowledgement The author is grateful to O.J. Boxma for several valuable discussions. Further the author is indebted to J. van den Berg for some useful suggestions concerning the proof of Lemma 3.3.

REFERENCES

[1] Anily, S., Federgruen, A. (1991). Structured partitioning problems. *Oper. Res.* **39**, 130-149.

[2] Boxma, O.J., Combé, M.B. (1994). Optimization of static traffic allocation policies. *Theor. Comp. Sc.* **125**, 17-43.

[3] Buzacott, J.A., Shanthikumar, J.G. (1992). Design of manufacturing systems using queueing models. *Queueing Systems* **12**, *Special Issue on Queueing Models of Manufacturing Systems*, 135-213.

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

[5] Cheng, W.C., Muntz, R.R. (1990). Optimal routing for closed queueing networks. In: *Perf. '90*, eds. P.J.B. King, I. Mitrani, R.J. Pooley (North-Holland, Amsterdam), 3-17.

[6] Coffman, E.G. Jr., Lueker, G.S., Rinnooy Kan, A.H.G. (1988). Asymptotic methods in the probabilistic analysis of sequencing and packing heuristics. *Mgmt. Sc.* **34**, 266-290.

[7] De Souza e Silva, E., Gerla, M. (1984). Load balancing in distributed systems with multiple classes and site constraints. In: *Perf. '84*, ed. E. Gelenbe (North-Holland, Amsterdam), 17-33.

[8] Garey, M.R., Johnson, D.S. (1979). *Computers and Intractability: a Guide to the Theory of NP-Completeness* (Freeman, San Francisco).

[9] Graham, R.L. (1966). Bounds for certain multiprocessing anomalies. *Bell Syst. Techn. J.* **45**, 1563-1581.

[10] Graham, R.L. (1969). Bounds on multiprocessing timing anomalies. *SIAM J. Appl. Math.* **17**, 263-269.

[11] Meilijson, I., Yechiali, U. (1977). On optimal right-of-way policies at a single-server station when insertion of idle times is permitted. *Stoch. Proc. Appl.* **6**, 25-32.

[12] Tantawi, A.N., Towsley, D. (1984). A general model for optimal static load balancing in star network configurations. In: *Perf, '84*, ed. E. Gelenbe (North-Holland, Amsterdam), 277-291.

[13] Tantawi, A.N., Towsley, D. (1985). Optimal static load balancing in distributed computer systems. *J. ACM* **32**, 445-465.

[14] Tripathi, S.K., Woodside, C.M. (1988). A vertex-allocation theorem for resources in queueing networks. *J. ACM* **35**, 221-230.

[15] Woodside, C.M., Tripathi, S.K. (1986). Optimal allocation of file servers in a local network environment. *IEEE Trans. Softw. Eng.* **12**, 844-848.

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

APPENDICES

A PROOF OF LEMMA 3.1

Lemma 3.1
$P_{i'}(x^*) \cap P_{i''}(x^*) = \emptyset$ for $i' \neq i''$.
In other words, if $\left(\frac{c_j}{\beta_j}, \frac{\beta_j^{(2)}}{\beta_j}\right) \in P_i(x^*)$, then $x^*_{ij} = 1$.

Proof
Suppose not, i.e., by definition of $P_i(x^*)$, there exist k_0, i', i'', $i' \neq i''$, such that

$$\left(\frac{c_{k_0}}{\beta_{k_0}}, \frac{\beta_{k_0}^{(2)}}{\beta_{k_0}}\right) = \sum_{k \in K_{i''}(x^*)} \alpha_k \left(\frac{c_k}{\beta_k}, \frac{\beta_k^{(2)}}{\beta_k}\right), \tag{A.1}$$

with $x^*_{i'k_0} > 0$, $x^*_{i''k} > 0$, $\alpha_k \geq 0$, $k \in K_{i''}(x^*)$, $\sum_{k \in K_{i''}(x^*)} \alpha_k = 1$. Moreover, not all the points $\left(\frac{c_k}{\beta_k}, \frac{\beta_k^{(2)}}{\beta_k}\right)$ with $\alpha_k > 0$ lie on a line, i.e., there is no linear equality that is satisfied by all the points $\left(\frac{c_k}{\beta_k}, \frac{\beta_k^{(2)}}{\beta_k}\right)$ with $\alpha_k > 0$.

Now consider $\epsilon \left\{\frac{\partial f(x)}{\partial x_{i''k_0}} - \frac{\partial f(x)}{\partial x_{i'k_0}}\right\}_{|x=x^*}$ and $\epsilon \left\{\frac{\partial f(x)}{\partial x_{i'k}} - \frac{\partial f(x)}{\partial x_{i''k}}\right\}_{|x=x^*}$, which measure the first-order effect on $f(x)$ around x^* of transferring a fraction ϵ of customer type k_0 from server i' to server i'' and transferring a fraction ϵ of customer type k from server i'' to server i', respectively, $k \in K_{i''}(x^*)$. As x^* is an optimal solution of problem (I), the first-order effect on $f(x)$ around x^* of transferring a customer type from one server to another cannot be negative, so $\left\{\frac{\partial f(x)}{\partial x_{i''k_0}} - \frac{\partial f(x)}{\partial x_{i'k_0}}\right\}_{|x=x^*} \geq 0$, $\left\{\frac{\partial f(x)}{\partial x_{i'k}} - \frac{\partial f(x)}{\partial x_{i''k}}\right\}_{|x=x^*} \geq 0$, $k \in K_{i''}(x^*)$, as may also be formally verified from the Kuhn-Tucker conditions.
Differentiating $f(\cdot)$ once,

$$\frac{\partial f(x)}{\partial x_{ij}} = \lambda_j c_j B_i(x) + \lambda_j \beta_j^{(2)} C_i(x) + \lambda_j \beta_j \mu_i B_i(x) C_i(x). \tag{A.2}$$

From (A.1), $\lambda_{k_0}(c_{k_0}, \beta_{k_0}^{(2)}, \beta_{k_0}) = \sum_{k \in K_{i''}(x^*)} \alpha_k \frac{\lambda_{k_0}\beta_{k_0}}{\lambda_k \beta_k} \lambda_k (c_k, \beta_k^{(2)}, \beta_k)$.

So $\left\{\frac{\partial f(x)}{\partial x_{i''k_0}} - \frac{\partial f(x)}{\partial x_{i'k_0}}\right\}_{|x=x^*} = - \sum_{k \in K_{i''}(x^*)} \alpha_k \frac{\lambda_{k_0}\beta_{k_0}}{\lambda_k \beta_k} \left\{\frac{\partial f(x)}{\partial x_{i'k}} - \frac{\partial f(x)}{\partial x_{i''k}}\right\}_{|x=x^*}$.

As $\left\{\frac{\partial f(x)}{\partial x_{i''k_0}} - \frac{\partial f(x)}{\partial x_{i'k_0}}\right\}_{|x=x^*} \geq 0$, $\alpha_k \left\{\frac{\partial f(x)}{\partial x_{i'k}} - \frac{\partial f(x)}{\partial x_{i''k}}\right\}_{|x=x^*} \geq 0$, $k \in K_{i''}(x^*)$, we conclude that $\alpha_k \left\{\frac{\partial f(x)}{\partial x_{i'k}} - \frac{\partial f(x)}{\partial x_{i''k}}\right\}_{|x=x^*} = 0$, $k \in K_{i''}(x^*)$. So

$$\frac{c_k}{\beta_k}(B_{i'}(x^*) - B_{i''}(x^*)) + \frac{\beta_k^{(2)}}{\beta_k}(C_{i'}(x^*) - C_{i''}(x^*)) = \mu_{i''} B_{i''}(x^*) C_{i''}(x^*) - \mu_{i'} B_{i'}(x^*) C_{i'}(x^*)$$

for all the points $\left(\frac{c_k}{\beta_k}, \frac{\beta_k^{(2)}}{\beta_k}\right)$ with $\alpha_k > 0$, i.e., there *is* a linear equality that is satisfied by all the points $\left(\frac{c_k}{\beta_k}, \frac{\beta_k^{(2)}}{\beta_k}\right)$ with $\alpha_k > 0$. Contradiction.

□

B Proof of Lemma 3.2

Lemma 3.2

If $B_{i'}(x^*) \geq B_{i''}(x^*)$, $C_{i'}(x^*) \geq C_{i''}(x^*)$, then $\mu_{i'} B_{i'}(x^*) C_{i'}(x^*) \leq \mu_{i''} B_{i''}(x^*) C_{i''}(x^*)$.

Proof

Suppose not, i.e.,

$$B_{i'}(x^*) \geq B_{i''}(x^*), C_{i'}(x^*) \geq C_{i''}(x^*), \mu_{i'} B_{i'}(x^*) C_{i'}(x^*) > \mu_{i''} B_{i''}(x^*) C_{i''}(x^*). \tag{B.1}$$

Take k_0 such that $x^*_{i'k_0} > 0$.

Now consider $\epsilon \left\{ \frac{\partial f(x)}{\partial x_{i''k_0}} - \frac{\partial f(x)}{\partial x_{i'k_0}} \right\}_{|x=x^*}$, which measures the first-order effect on $f(x)$ around x^* of transferring a fraction ϵ of customer type k_0 from server i' to server i''. As x^* is an optimal solution of problem (I), the first-order effect on $f(x)$ around x^* of transferring a customer type from one server to another cannot be negative, so $\left\{ \frac{\partial f(x)}{\partial x_{i''k_0}} - \frac{\partial f(x)}{\partial x_{i'k_0}} \right\}_{|x=x^*} \geq 0$, as may also be formally verified from the Kuhn-Tucker conditions.

From (A.2), if (B.1) were to hold,

$$\begin{aligned}
\left\{ \frac{\partial f(x)}{\partial x_{i''k_0}} - \frac{\partial f(x)}{\partial x_{i'k_0}} \right\}_{|x=x^*} &= \lambda_{k_0} c_{k_0} \left(B_{i''}(x^*) - B_{i'}(x^*)\right) + \lambda_{k_0} \beta_{k_0}^{(2)} \left(C_{i''}(x^*) - C_{i'}(x^*)\right) \\
&+ \lambda_{k_0} \beta_{k_0} \left(\mu_{i''} B_{i''}(x^*) C_{i''}(x^*) - \mu_{i'} B_{i'}(x^*) C_{i'}(x^*)\right) \\
&< 0.
\end{aligned}$$

Contradiction. □

C Proof of Lemma 3.3

Lemma 3.3

Assume $x^*_{i'j'} > 0$, $x^*_{i''j''} > 0$.

If $\frac{c_{j'}}{\beta_{j'}} \leq \frac{c_{j''}}{\beta_{j''}}$, $\frac{\beta_{j'}^{(2)}}{\beta_{j'}} \geq \frac{\beta_{j''}^{(2)}}{\beta_{j''}}$, $\left(\frac{c_{j'}}{\beta_{j'}}, \frac{\beta_{j'}^{(2)}}{\beta_{j'}}\right) \neq \left(\frac{c_{j''}}{\beta_{j''}}, \frac{\beta_{j''}^{(2)}}{\beta_{j''}}\right)$, then $i' \leq i''$.

Proof

Suppose not, i.e., $x^*_{i'j'} > 0$, $x^*_{i''j''} > 0$, $i' \neq i''$,

$$\frac{c_{j'}}{\beta_{j'}} \leq \frac{c_{j''}}{\beta_{j''}}, \quad \frac{\beta_{j'}^{(2)}}{\beta_{j'}} \geq \frac{\beta_{j''}^{(2)}}{\beta_{j''}}, \quad \left(\frac{c_{j'}}{\beta_{j'}}, \frac{\beta_{j'}^{(2)}}{\beta_{j'}}\right) \neq \left(\frac{c_{j''}}{\beta_{j''}}, \frac{\beta_{j''}^{(2)}}{\beta_{j''}}\right), \tag{C.1}$$

$B_{i'}(x^*) \leq B_{i''}(x^*)$, $C_{i'}(x^*) \geq C_{i''}(x^*)$.

Now consider $\epsilon \left\{ \frac{\partial f(x)}{\partial x_{i''j'}} - \frac{\partial f(x)}{\partial x_{i'j'}} \right\}_{|x=x^*}$ and $\epsilon \left\{ \frac{\partial f(x)}{\partial x_{i'j''}} - \frac{\partial f(x)}{\partial x_{i''j''}} \right\}_{|x=x^*}$, which measure the first-order effect on $f(x)$ around x^* of transferring a fraction ϵ of customer type j' from server i' to server i'' and a fraction ϵ of customer type j'' from server i'' to server i', respectively. As x^* is an optimal solution of problem (I), the first-order effects on $f(x)$ around x^* of transferring customer types from one server to another cannot be negative, so $\left\{ \frac{\partial f(x)}{\partial x_{i''j'}} - \frac{\partial f(x)}{\partial x_{i'j'}} \right\}_{|x=x^*} \geq 0$, $\left\{ \frac{\partial f(x)}{\partial x_{i'j''}} - \frac{\partial f(x)}{\partial x_{i''j''}} \right\}_{|x=x^*} \geq 0$, as may also be formally verified from the Kuhn-Tucker conditions.

From (A.2),

$$\lambda_{j''}\beta_{j''} \left\{ \frac{\partial f(x)}{\partial x_{i''j'}} - \frac{\partial f(x)}{\partial x_{i'j'}} \right\} + \lambda_{j'}\beta_{j'} \left\{ \frac{\partial f(x)}{\partial x_{i'j''}} - \frac{\partial f(x)}{\partial x_{i''j''}} \right\} =$$

$$\lambda_{j'}\lambda_{j''}\beta_{j'}\beta_{j''} \left\{ \left(\frac{c_{j'}}{\beta_{j'}} - \frac{c_{j''}}{\beta_{j''}} \right) (B_{i''}(x) - B_{i'}(x)) + \left(\frac{\beta_{j'}^{(2)}}{\beta_{j'}} - \frac{\beta_{j''}^{(2)}}{\beta_{j''}} \right) (C_{i''}(x) - C_{i'}(x)) \right\}.$$

So, if (C.1) were to hold, then $\lambda_{j''}\beta_{j''} \left\{ \frac{\partial f(x)}{\partial x_{i''j'}} - \frac{\partial f(x)}{\partial x_{i'j'}} \right\}_{|x=x^*} + \lambda_{j'}\beta_{j'} \left\{ \frac{\partial f(x)}{\partial x_{i'j''}} - \frac{\partial f(x)}{\partial x_{i''j''}} \right\}_{|x=x^*}$ ≤ 0. Contradiction, unless $\left\{ \frac{\partial f(x)}{\partial x_{i''j'}} - \frac{\partial f(x)}{\partial x_{i'j'}} \right\}_{|x=x^*} = 0$, $\left\{ \frac{\partial f(x)}{\partial x_{i'j''}} - \frac{\partial f(x)}{\partial x_{i''j''}} \right\}_{|x=x^*} = 0$.

Now consider $\epsilon^2 \sum_{i_1,i_2=i',i''} \sum_{j_1,j_2=j',j''} \left\{ \frac{\partial^2 f(x)}{\partial x_{i_1j_1} \partial x_{i_2j_2}} \alpha_{i_1j_1}\alpha_{i_2j_2} \right\}_{|x=x^*}$ for $\alpha_{i'j'} = -\lambda_{j''}\beta_{j''}$, $\alpha_{i''j'} = \lambda_{j''}\beta_{j''}$, $\alpha_{i'j''} = \lambda_{j'}\beta_{j'} + \alpha$, $\alpha_{i''j''} = -\lambda_{j'}\beta_{j'} - \alpha$, which measures the second-order effect on $f(x)$ around x^* of transferring a fraction $\epsilon\lambda_{j''}\beta_{j''}$ of customer type j' from server i' to server i'' and a fraction $\epsilon(\lambda_{j'}\beta_{j'} + \alpha)$ of customer type j'' from server i'' to server i'. As x^* is an optimal solution of problem (I), while the first-order effects on $f(x)$ around x^* of transferring customer types from one server to another are zero, the second-order effect cannot be negative, so $\sum_{i_1,i_2=i',i''} \sum_{j_1,j_2=j',j''} \left\{ \frac{\partial^2 f(x)}{\partial x_{i_1j_1} \partial x_{i_2j_2}} \alpha_{i_1j_1}\alpha_{i_2j_2} \right\}_{|x=x^*} \geq 0$.

Differentiating $f(\cdot)$ twice,

$$\frac{\partial^2 f(x)}{\partial x_{i_1j_1} \partial x_{i_2j_2}} = \delta_{i_1i_2} \left\{ \frac{1}{\mu/m - \sum_{j=1}^{n} \lambda_j\beta_j x_{i_1j}} \left(\lambda_{j_1}\lambda_{j_2} \left(c_{j_1}\beta_{j_2}^{(2)} + c_{j_2}\beta_{j_1}^{(2)} \right) + 2\lambda_{j_2}\beta_{j_2} \frac{\partial f(x)}{\partial x_{i_1j_1}} \right) + \right.$$

$$\left. \frac{\lambda_{j_1}\lambda_{j_2}}{\left(\mu/m - \sum_{j=1}^{n} \lambda_j\beta_j x_{i_1j} \right)^2} \left((\beta_{j_1}c_{j_2} - \beta_{j_2}c_{j_1}) \sum_{j=1}^{n} \lambda_j\beta_j^{(2)} x_{i_1j} + \left(\beta_{j_1}\beta_{j_2}^{(2)} - \beta_{j_2}\beta_{j_1}^{(2)} \right) \sum_{j=1}^{n} \lambda_j c_j x_{i_1j} \right) \right\}$$

with $\delta_{i_1 i_2}$ the Kronecker delta. So, as $\dfrac{\partial f(x)}{\partial x_{i''j'}} - \dfrac{\partial f(x)}{\partial x_{i'j'}} = 0$, $\dfrac{\partial f(x)}{\partial x_{i'j''}} - \dfrac{\partial f(x)}{\partial x_{i''j''}} = 0$, $i' \neq i''$,

$$\sum_{i_1,i_2=i',i''} \sum_{j_1,j_2=j',j''} \frac{\partial^2 f(x)}{\partial x_{i_1j_1} \partial x_{i_2j_2}} \alpha_{i_1j_1}\alpha_{i_2j_2} =$$

$$2\left\{\frac{1}{\mu/m - \sum\limits_{j=1}^{n} \lambda_j \beta_j x_{i'j}} + \frac{1}{\mu/m - \sum\limits_{j=1}^{n} \lambda_j \beta_j x_{i''j}}\right\}\left\{\lambda_{j'}^2 \lambda_{j''}^2 \beta_{j'}^2 \beta_{j''}^2 \left(\frac{c_{j''}}{\beta_{j''}} - \frac{c_{j'}}{\beta_{j'}}\right)\left(\frac{\beta_{j''}^{(2)}}{\beta_{j''}} - \frac{\beta_{j'}^{(2)}}{\beta_{j'}}\right) + \right.$$

$$\alpha \lambda_{j'} \lambda_{j''}^2 \beta_{j'} \beta_{j''}^2 \left(\left(\frac{c_{j''}}{\beta_{j''}} - \frac{c_{j'}}{\beta_{j'}}\right)\left(\frac{\beta_{j''}^{(2)}}{\beta_{j''}} + B_{i'}(x)\right) + \left(\frac{\beta_{j''}^{(2)}}{\beta_{j''}} - \frac{\beta_{j'}^{(2)}}{\beta_{j'}}\right)\left(\frac{c_{j''}}{\beta_{j''}} + C_{i'}(x)\right)\right) +$$

$$\left.\alpha^2 \left(\lambda_{j''}^2 c_{j''} \beta_{j''}^{(2)} + \lambda_{j''} \beta_{j''} \frac{\partial f(x)}{\partial x_{i'j''}}\right)\right\}.$$

So, if (C.1) were to hold, then $\displaystyle\sum_{i_1,i_2=i',i''} \sum_{j_1,j_2=j',j''} \left\{\frac{\partial^2 f(x)}{\partial x_{i_1j_1} \partial x_{i_2j_2}} \alpha_{i_1j_1}\alpha_{i_2j_2}\right\}_{|x=x^*} < 0$ for some α. Contradiction.

□

D Proof of Lemma 3.4

Lemma 3.4

$T_{i'}(x^*) \cap T_{i''}(x^*) = \emptyset$ for $i' \neq i''$.

In other words, if $\left(\dfrac{c_j}{\beta_j}, \dfrac{\beta_j^{(2)}}{\beta_j}\right) \in T_i(x^*)$, then $x_{ij}^* = 1$.

Proof

Suppose not, i.e., by definition of $T_i(x^*)$ there exist i', i'', $i' \neq i''$, $j'_{i'}, j''_{i'} \in K_{i'}(x^*)$, $j'_{i''}, j''_{i''} \in K_{i''}(x^*)$, y, z, such that

$$\frac{c_{j'_{i'}}}{\beta_{j'_{i'}}} \leq y \leq \frac{c_{j''_{i'}}}{\beta_{j''_{i'}}}, \frac{\beta_{j'_{i'}}^{(2)}}{\beta_{j'_{i'}}} \geq z \geq \frac{\beta_{j''_{i'}}^{(2)}}{\beta_{j''_{i'}}}, \left(\frac{c_{j'_{i'}}}{\beta_{j'_{i'}}}, \frac{\beta_{j'_{i'}}^{(2)}}{\beta_{j'_{i'}}}\right) \neq (y,z) \neq \left(\frac{c_{j''_{i'}}}{\beta_{j''_{i'}}}, \frac{\beta_{j''_{i'}}^{(2)}}{\beta_{j''_{i'}}}\right),$$

$$\frac{c_{j'_{i''}}}{\beta_{j'_{i''}}} \leq y \leq \frac{c_{j''_{i''}}}{\beta_{j''_{i''}}}, \frac{\beta_{j'_{i''}}^{(2)}}{\beta_{j'_{i''}}} \geq z \geq \frac{\beta_{j''_{i''}}^{(2)}}{\beta_{j''_{i''}}}, \left(\frac{c_{j'_{i''}}}{\beta_{j'_{i''}}}, \frac{\beta_{j'_{i''}}^{(2)}}{\beta_{j'_{i''}}}\right) \neq (y,z) \neq \left(\frac{c_{j''_{i''}}}{\beta_{j''_{i''}}}, \frac{\beta_{j''_{i''}}^{(2)}}{\beta_{j''_{i''}}}\right).$$

From Lemma 3.3, on the one hand, $i' \leq i''$, as $x^*_{i'j'_{i'}} > 0$, $x^*_{i''j''_{i''}} > 0$,

$$\frac{c_{j'_{i'}}}{\beta_{j'_{i'}}} \leq \frac{c_{j''_{i''}}}{\beta_{j''_{i''}}}, \frac{\beta^{(2)}_{j'_{i'}}}{\beta_{j'_{i'}}} \geq \frac{\beta^{(2)}_{j''_{i''}}}{\beta_{j''_{i''}}}, \left(\frac{c_{j'_{i'}}}{\beta_{j'_{i'}}}, \frac{\beta^{(2)}_{j'_{i'}}}{\beta_{j'_{i'}}}\right) \neq \left(\frac{c_{j''_{i''}}}{\beta_{j''_{i''}}}, \frac{\beta^{(2)}_{j''_{i''}}}{\beta_{j''_{i''}}}\right);$$

on the other hand, $i' \geq i''$, as $x^*_{i'j''_{i'}} > 0$, $x^*_{i''j'_{i''}} > 0$,

$$\frac{c_{j''_{i'}}}{\beta_{j''_{i'}}} \geq \frac{c_{j'_{i''}}}{\beta_{j'_{i''}}}, \frac{\beta^{(2)}_{j''_{i'}}}{\beta_{j''_{i'}}} \leq \frac{\beta^{(2)}_{j'_{i''}}}{\beta_{j'_{i''}}}, \left(\frac{c_{j''_{i'}}}{\beta_{j''_{i'}}}, \frac{\beta^{(2)}_{j''_{i'}}}{\beta_{j''_{i'}}}\right) \neq \left(\frac{c_{j'_{i''}}}{\beta_{j'_{i''}}}, \frac{\beta^{(2)}_{j'_{i''}}}{\beta_{j'_{i''}}}\right).$$

So $i' = i''$. Contradiction.

□

E A METHOD FOR DETERMINING AN OPTIMAL ALLOCATION

In this appendix we describe a method for determining an optimal allocation in cases that satisfy Assumption 4.1. In section 4 we sketched the main idea of the method. Here we describe the method in detail.

Lemma

a. $\sum_{j=1}^{n} \lambda_j \beta_j^{(2)} x^*_{1j} \geq \frac{1}{m} \sum_{j=1}^{n} \lambda_j \beta_j^{(2)} \geq \sum_{j=1}^{n} \lambda_j \beta_j^{(2)} x^*_{mj}$.

b. $\sum_{j=1}^{n} \lambda_j c_j x^*_{1j} \leq \frac{1}{m} \sum_{j=1}^{n} \lambda_j c_j \leq \sum_{j=1}^{n} \lambda_j c_j x^*_{mj}$.

Proof

Proof of a. The servers are indexed such that

$$\frac{\sum_{j=1}^{n} \lambda_j \beta_j^{(2)} x^*_{ij}}{\mu/m - \sum_{j=1}^{n} \lambda_j \beta_j x^*_{ij}} \geq \frac{\sum_{j=1}^{n} \lambda_j \beta_j^{(2)} x^*_{i+1j}}{\mu/m - \sum_{j=1}^{n} \lambda_j \beta_j x^*_{i+1j}}.$$

So

$$\frac{\mu}{m} \sum_{j=1}^{n} \lambda_j \beta_j^{(2)} x^*_{ij} - \left(\sum_{j=1}^{n} \lambda_j \beta_j x^*_{i+1j}\right) \left(\sum_{j=1}^{n} \lambda_j \beta_j^{(2)} x^*_{ij}\right) \geq \tag{E.1}$$

$$\frac{\mu}{m} \sum_{j=1}^{n} \lambda_j \beta_j^{(2)} x^*_{i+1j} - \left(\sum_{j=1}^{n} \lambda_j \beta_j x^*_{ij}\right) \left(\sum_{j=1}^{n} \lambda_j \beta_j^{(2)} x^*_{i+1j}\right).$$

On the one hand

$$\frac{\sum\limits_{j=1}^{n} \lambda_j \beta_j^{(2)} x^*_{ij}}{\sum\limits_{j=1}^{n} \lambda_j \beta_j x^*_{ij}} \geq \min_{j \in K_i(x^*)} \frac{\beta_j^{(2)}}{\beta_j};$$

on the other hand

$$\frac{\sum\limits_{j=1}^{n} \lambda_j \beta_j^{(2)} x^*_{i+1j}}{\sum\limits_{j=1}^{n} \lambda_j \beta_j x^*_{i+1j}} \leq \max_{j \in K_{i+1}(x^*)} \frac{\beta_j^{(2)}}{\beta_j}.$$

From Lemma 3.3, $\min\limits_{j \in K_i(x^*)} \frac{\beta_j^{(2)}}{\beta_j} \geq \max\limits_{j \in K_{i+1}(x^*)} \frac{\beta_j^{(2)}}{\beta_j}$.
So

$$\left(\sum_{j=1}^{n} \lambda_j \beta_j x^*_{i+1j}\right) \left(\sum_{j=1}^{n} \lambda_j \beta_j^{(2)} x^*_{ij}\right) \geq \left(\sum_{j=1}^{n} \lambda_j \beta_j x^*_{ij}\right) \left(\sum_{j=1}^{n} \lambda_j \beta_j^{(2)} x^*_{i+1j}\right). \tag{E.2}$$

Combining (E.1), (E.2), $\sum\limits_{j=1}^{n} \lambda_j \beta_j^{(2)} x^*_{ij} \geq \sum\limits_{j=1}^{n} \lambda_j \beta_j^{(2)} x^*_{i+1j}$. Further $\sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n} \lambda_j \beta_j^{(2)} x^*_{ij} = \sum\limits_{j=1}^{n} \lambda_j \beta_j^{(2)}$, as $\sum\limits_{i=1}^{m} x^*_{ij} = 1$, $j = 1, \ldots, n$. So $\sum\limits_{j=1}^{n} \lambda_j \beta_j^{(2)} x^*_{1j} \geq \frac{1}{m} \sum\limits_{j=1}^{n} \lambda_j \beta_j^{(2)} \geq \sum\limits_{j=1}^{n} \lambda_j \beta_j^{(2)} x^*_{mj}$.
Proof of b. Follows immediately by symmetry considerations.

□

Step 1. Start with the allocation to server 1, which is to carry the largest $B_i(x^*)$ and the smallest $C_i(x^*)$.
Determine a lower bound $s_1^l = \sum\limits_{j=1}^{n} x^l_{1j}$ for s_1^* from (i) $\sum\limits_{j=1}^{n} \lambda_j \beta_j^{(2)} x^l_{1j} = \frac{1}{m} \sum\limits_{j=1}^{n} \lambda_j \beta_j^{(2)}$, (ii) if $x^l_{1j'} > 0$, then $x^l_{1j''} = 1$ for $j'' < j'$. However, if $\sum\limits_{j=1}^{n} \lambda_j \beta_j^{(2)} x^l_{1j} = \frac{1}{m} \sum\limits_{j=1}^{n} \lambda_j \beta_j^{(2)}$ implies $\sum\limits_{j=1}^{n} \lambda_j \beta_j x^l_{1j} \geq 1$, then replace (i) by (iii) $\sum\limits_{j=1}^{n} \lambda_j \beta_j x^l_{1j} = 1 - \epsilon$, with ϵ sufficiently small.
Determine an upper bound $s_1^u = \sum\limits_{j=1}^{n} x^u_{1j}$ for s_1^* from (i) $\sum\limits_{j=1}^{n} \lambda_j c_j x^u_{1j} = \frac{1}{m} \sum\limits_{j=1}^{n} \lambda_j c_j$, (ii) if $x^u_{1j'} > 0$, then $x^u_{1j''} = 1$ for $j'' < j'$. However, if $\sum\limits_{j=1}^{n} \lambda_j c_j x^u_{1j} = \frac{1}{m} \sum\limits_{j=1}^{n} \lambda_j c_j$ implies $\sum\limits_{j=1}^{n} \lambda_j \beta_j x^u_{1j}$ $\leq \rho - (m-1)$, then replace (i) by (iii) $\sum\limits_{j=1}^{n} \lambda_j \beta_j x^u_{1j} = \rho - (m-1) + \epsilon$, with ϵ sufficiently small.

Step 2. Make an estimate $s_1 = \sum_{j=1}^{n} x_{1j}$ for s_1^*, somewhere in between s_1^l and s_1^u, e.g., $s_1 = (s_1^l + s_1^u)/2$.

Step 3. For $i = 1, 2, \ldots, m-1$, determine $s_{i+1} = \sum_{j=1}^{n} x_{i+1j}$ from (i) $x_{i+1j_i} \leq 1 - \sum_{k=1}^{i} x_{kj_i}$, with $j_i = \max\{j \mid x_{ij} > 0\}$, (ii) if $x_{i+1j'} > 0$ for $j_i < j'$, then $x_{i+1j_i} = 1 - \sum_{k=1}^{i} x_{kj_i}$, $x_{i+1j''} = 1$ for $j_i < j'' < j'$, (iii) $\frac{\partial f(x)}{\partial x_{ij_i}} = \frac{\partial f(x)}{\partial x_{i+1j_i}}$. Provided $j_i = \max\{j \mid x_{ij} > 0\}$, the latter condition necessarily holds for an optimal solution of problem (I), as may be formally verified from the Kuhn-Tucker conditions. Note that $\frac{\partial f(x)}{\partial x_{ij_i}} > 0$ and constant in s_{i+1}, and $\frac{\partial f(x)}{\partial x_{i+1j_i}} = 0$ for $s_{i+1} = 0$ and increasing in s_{i+1}. So through (i), (ii), (iii), s_i uniquely determines s_{i+1}, unless $\frac{\partial f(x)}{\partial x_{ij_i}} > \frac{\partial f(x)}{\partial x_{i+1j_i}}$ even for $x_{i+1n} = 1$, to which we return in the next step.

Step 4. Sooner or later one either runs out of servers, i.e., $\frac{\partial f(x)}{\partial x_{m-1j_{m-1}}} = \frac{\partial f(x)}{\partial x_{mj_{m-1}}}$, but $x_{mn} < 1$, or out of customer types, i.e., $x_{i+1n} = 1$, but $\frac{\partial f(x)}{\partial x_{ij_i}} > \frac{\partial f(x)}{\partial x_{i+1j_i}}$. If one runs out of servers, then apparently $s_1 < s_1^*$, so then replace s_1^l by s_1. If one runs out of customer types, then apparently $s_1 > s_1^*$, so then replace s_1^u by s_1. One may of course make a more sophisticated estimate for s_1^* than the estimate suggested above. One may e.g. use information about how soon one either ran out of servers or out of customer types. Repeat the procedure until lower and upper bound are sufficiently close.