University of
Waterloo Research Institute

THE DYNAMIC FILE ALLOCATION PROBLEM IN A THREE-COMPUTER NETWORK WITH A LINEAR TOPOLOGY

FINAL REPORT

AUGUST, 1984

UNIVERSITY OF WATERLOO

WATERLOO RESEARCH INSTITUTE

(2)

The Dynamic File Allocation Problem in a Three-Computer Network with a Linear Topology

Project No. 212-21

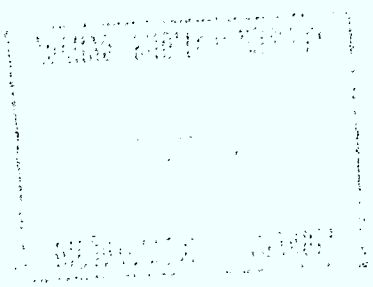
Submitted to:

Department of Communications

by

(1)
Professor M. Vidyasagar
Department of Electrical Engineering

August, 1984



Abstract

This report is devoted to the dynamic file allocation problem in a three-computer network, which is not completely connected.

Under several assumptions the problem is formulated as a discrete-time optimal control problem and an explicit solution is derived using backward dynamic programming.

Several numerical results from computer simulations are also presented.

Contents

List of Tables

List of Illustrations

Chapter 1
Introduction

1.1 Statement and Discussion of the Problem

During the first two decades of their existence, computer systems were higly localized, usually within a single large room to which the users were supposed to bring their work for processing. This model of the "computer center" had two obvious disadvantages : the concept of a single large computer doing all the work and the idea of users bringing work to the computer, instead of bringing the computer to the users. Since about 1970 the centralized computer systems are being replaced by *computer networks* . According to the definition in [Tannenbaum 1981], a computer network consists of a number of separate but interconnected computers capable of exchanging information through communication lines. In the last few years there has been a growing interest in problems of modeling, analysis, and the design of such networks. Dynamic file allocation, dynamic routing, load sharing, flow control, processor allocation, reliability and connectivity are some of the problems associated with computers networks.

In this report the dynamic file allocation problem is studied for a computer network with a special topology. The problem can be briefly described as follows : One of the main purposes of a computer network is

to provide the facility for common use of data bases and information files by all computers in the system. When a file is used by several computers in the network , it can be stored in the memory of (at least) one of them and be accessed by the other computers via the communication channels. The problem is to find the optimal locations for these files and minimize the total operation cost within a certain period of operation of the system.

1.2 Previous Work

The optimal file allocation problem is similar to several other problems that have received considerable attention over the past twenty years. The problem of the optimum location of a switching center in a communication network, the problem of the optimum location of a police station in a highway system and the problem of the optimum location of a hospital in a multi-community system, are typical examples of problems similar to the problem of the optimum location of a file in a computer network.

S. L. Hakimi [1964] formulated and solved the problem of the optimum location of a switching center in a communication network using graph theory and game theory techniques. This problem is very similar to the optimal file allocation problem, if one considers the switching center as a file and the traffic messages as messages requesting the file. One year later, he formulated and solved the same problem, considering the case where more than one switching center exists in the network [Hakimi

1965]. Clearly, this more general problem is similar to the problem of the optimum location of more than one file in a computer network.

An extension of the switching center allocation problem to the case where the network traffic is considered to be random can be found in [Frank 1966]. Very interesting work on the optimal file allocation problem has been done by [Chu 1969,1973], who developed a model describing the problem and proved that it can be formulated as a linear zero-one programming problem.

In all these approaches described so far, the optimal file allocation problem is studied as a static problem. It is assumed that all parameters of the system are known a priori and that the design is based on their average value over the period operation of the system. The goal is to find the best location for the files, under the assumption that this location will remain fixed for the entire operating period. The criterion of optimality is minimal overall operating costs.

A. Segall [1976] was the first to present models describing the problem of *dynamic* file allocation in a computer network. He treated the problem for the case when the (time varying) rates of the file requests are known in advance, as well as when only prior statistics are available for these rates. Segall assumed that only one file exists in the network at any given time and he gave analytic solutions for this case, based on a dynamic programming approach. The extension of Segall's work to problems where multiple copies of the file exist in the system has been studied in [Ros 1976]. Subsequently Segall and Sandell studied the same

problem with a view to deriving a decentralized optimal solution [Segall and Sandell 1979].

1.3 Summary of Report

In [Segall 1976] it is assumed that the computer network under study is completely connected, that is, there is a direct communication path between every pair of computers in the network. In this report we study the dynamic file allocation problem without this assumption. A new model is developed describing the dynamic file allocation problem in a three-computer network, which is not completely connected. Also, an explicit solution for this problem is given using the backward dynamic programming approach.

The rest of the report is organized as follows : In Chapter 2 Segall's work is presented in detail. Emphasis is given to the assumptions under which the problem has been formulated. In Chapter 3 the new model as well as the solution to the problem are given. Simulation results and a detailed study of the effects of the various parameters of the system on the optimal solution are presented in Chapter 4. Finally, a summary of results and suggestions for further studies are given in Chapter 5.

Chapter 2
The Dynamic File Allocation Problem in a Completely Connected Computer Network

2.1 Statement of the Problem and Basic Assumptions

As mentioned earlier, a static analysis of the file allocation problem assumes that the parameters of the system are known a priori and the design is based on their average value over the period of operation of the system. But if the parameters of the system - for instance the demand rates - vary with time, a dynamic allocation might give a substantial improvement in performance. Dynamic file assignment might also be necessary when there is the possibility of node or link failures, in which case the files may have to be reallocated according to the changing topology of the network.

Segall in [Segall 1976] gave a model describing dynamic file assignment under certain assumptions. He considered the situation where there are several computers connected together with a direct path from each computer to every other (i.e. the network is completely connected). The procedure that Segall proposed is as follows : Suppose the file is stored at time t in the memory of computer i. If at time t, it is requested only by computer i, then no transmission cost is incurred and there is no decision to be made. If it is requested by another

computer, say j, the file is transmitted for use to computer j (where it is kept temporarily in a buffer) and now a decision is to be made whether the file is to be left in memory i or erased from memory i and written in memory j. A similar decision is to be made if the file is requested by more than one computer at time t. The restriction of reallocating the file only in conjunction with a regular transmission is reasonable for this model, because if a change of location is decided upon, one might as well wait until the file is requested next time by an appropriate computer. Otherwise it is conceivable that the file might be transferred back and forth, without anybody actually using it. It is important to note at this point that any decision is made by a *central controller* which decides at any time where the file is to be located.

Segall made several simplifying assumptions that are still consistent with the models appearing in real networks. Because of updating and memory limitations it is desirable to have in the system as few copies of any single file as possible. On the other hand, high communication costs might dictate keeping a large number of copies. Segall has assumed that the decisive factor is the updating of the files and therefore he decided that only one copy of each file is allowed to exist in the system at any given time. He also assumed that the files are requested by the computers according to mutually independent processes and also that the files are sufficiently short. Moreover he considered the communication lines to have sufficient capacity and the computers sufficient memory so that the transmission of the file takes a very short time and there is no restriction

on how many files a computer can carry. Under these assumptions, it is clear that in fact the files do not interfere with each other, and one can therefore treat each file separately.

2.2 The Modeling Equations

Segall formulated the problem both in continuous and discrete time. His model can be discribed as follows : Consider a completely connected system of M computers. Let $y_i(t)$ be defined as

$$y_i(t) = \begin{cases} 1 & \text{if the file is held in the memory} \\ & \text{of computer } i \text{ at time } t \\ 0 & \text{otherwise} \end{cases} \qquad (2.1)$$

where $i = 1,2,...,M$ and $t = 1,2,...$ for the discrete time analysis and $t \geq 0$ for continuous time. Clearly, at any given time only one of the variables $\{y_i(t), i = 1,2,...M\}$ can be one and all others will be zero, since it is desirable to keep only one copy of the file in the system at any time. Then a model for the requests of the file of interest by the various computers is given.

Continuous-Time

Let $\{N_i(t), t > 0\}$ be M independent Poisson processes with rates exactly known describing the requests of the file by computers $i = 1,2,...,M$, where

$$N_i(t) = \begin{cases} 1 & \text{if the file is requested by computer } i \\ & \text{at time } t \\ 0 & \text{otherwise} \end{cases} \qquad (2.2)$$

where $i = 1,2,...,M$ and $t > 0$

Discrete-Time

Let $\{n_i(t), t = 1,2,...\}$ be M independent Bernoulli processes with rates exactly known (for more details see Appendix A) describing the requests of the file by computers $i = 1,2,...M$, where

$$n_i(t) = \begin{cases} 1 & \text{if the file is requested by computer } i \\ & \text{at time } t \\ 0 & \text{otherwise} \end{cases} \qquad (2.3)$$

where $i = 1,2,...,M$ and $t = 1,2,...$

Finally, the decision variables $\alpha_{ij}(t)$ with $i \neq j$ are defined as follows

$$\alpha_{ij}(t) = \begin{cases} 1 & \text{if, given that the file is in the memory of} \\ & \text{computer } i \text{ at time } t \text{ and is requested} \\ & \text{by computer } j, \text{ the decision is to transfer} \\ & \text{it to the memory of computer } j \\ 0 & \text{otherwise} \end{cases} \qquad (2.4)$$

where $i,j = 1,2,...,M$ with $i \neq j$ and $t = 1,2,...$ for the discrete time analysis and $t \geq 0$ for continuous time. Then the dynamics of the file are described by the following equations

Continuous-Time

$$dy_i(t) = -y_i(t-) \sum_{j \neq i} \alpha_{ij}(t) dN_j(t) + \sum_{j \neq i} \alpha_{ji}(t) y_j(t-) dN_i(t) \quad (2.5)$$

where $i,j = 1,2,...,M$ and $t \geq 0$

The explanation behind the equations (2.5) is as follows : The first term in the right-hand side of (2.5) reflects the fact that if the file is in memory i at time $t-$, and is transferred to memory j at time t, then $y_i(t)$ changes from $y_i(t-)=1$ to $y_i(t)=0$. The second term reflects an opposite transfer and if there is no transfer from or into memory i, then $dy_i(t)=0$.

Discrete-Time

$$y_i(t+1) = y_i(t) \left[1 - \sum_{j \neq i} \alpha_{ij}(t) n_j(t) \right] + \sum_{j \neq i} y_j(t) \alpha_{ji}(t) n_i(t) \quad (2.6)$$

where $i,j = 1,2,...,M$ and $t = 1,2,...$

The explanation behind the equations (2.4) is as follows: Suppose that at time t the file of interest is in the memory of the i th computer. Then $y_i(t) = 1$ and $y_j(t) = 0$ for $j \neq i$. The first term in the right-hand side of (2.4) reflects the fact that, if another computer j (with $j \neq i$) requests the file at time t ($n_j(t) = 1$) and the central controller decides to transfer the file to the memory of computer j ($\alpha_{ij} = 1$), then at time $t+1$ the file is removed from the memory of computer i ($y_i(t+1) = 0$).

The file remains in the same location (in other words $y_i(t+1) = y_i(t)$) if there is no request by other computer ($n_j(t) = 0$ for all $j \neq i$), or if there is some request but the central controller decides not to transfer the file ($\alpha_{ij}(t) = 0$ for all $j \neq i$).The second term in the right-hand side of (2.4) can be explained as follows: Suppose that at time t there is no file in the memory of the i th computer ($y_i(t) = 0$). Then, at time $t+1$, computer i gets the file if and only if it requests the file at time t ($n_i(t)=1$) and the central controller decides to transfer the file from computer j which has the file at time t to computer i ($\alpha_{ji}(t) = 1$). This results in $y_i(t+1) = 1$.

As control variables, define

$$u_{ij}(t) = \alpha_{ij}(t) n_j(t) \tag{2.7}$$

where $i \neq j$ and $i, j = 1,2,...,M$. Then the dynamics of the file are described by

$$y_i(t+1) = y_i(t)\left[1 - \sum_{j \neq i} u_{ij}(t)\right] + \sum_{j \neq i} y_j(t) u_{ji}(t) \tag{2.8}$$

where $i \neq j$ and $i, j = 1,2,...,M$.

Equation (2.8) may be written in a general form

$$\mathbf{y}(t+1) = \mathbf{f}(\mathbf{y}(t), \mathbf{u}(t)) \tag{2.9}$$

where $\mathbf{y}$ (t) is the M-dimensional state vector given by

$$\mathbf{y}(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_M(t) \end{bmatrix} \tag{2.10}$$

and $\mathbf{u}$ (t) is the (M^2-M)-dimensional control vector given by

$$\mathbf{u}(t) = \begin{bmatrix} u_{12}(t) \\ u_{13}(t) \\ \vdots \\ u_{1M}(t) \\ u_{21}(t) \\ u_{23}(t) \\ \vdots \\ u_{2M}(t) \\ \vdots \\ u_{M,M-1}(t) \end{bmatrix} \tag{2.11}$$

Also, the definition of the function $\mathbf{f}$ is evident.

From Equation (2.9) it is easy to see that there are M possible states of the system. Each of these M states is associated with the file being located in the memory of one of the M computers of the network. It is also easy to see that there are M^2-M+1 different control vectors (including the zero vector, corresponding to the situation where the central controller decides to keep the file in the same location).

Then an expression for the total operating cost is given as follows:

Continuous-Time

The total cost per unit time over any operating period of length N is

$$C = E \int_0^N \left\{ \sum_{i=1}^{M} \left[C_i + \sum_{j \neq i} C_{ij} \lambda_j (t) \right] y_i (t) \right\} dt \tag{2.12}$$

where C_i is the storage cost per unit time in memory of computer i, and C_{ij} is the communication cost per transmission over the line connecting computer i and computer j. Also $\lambda_j (t)$ denotes the rate of the request described by the Poisson process $N_j (t)$. As we assume that the constants $\lambda_j (t)$ are all known, the cost function can be finally expressed in the form

$$C = E \int_0^N L_t (\mathbf{y}(t)) dt \tag{2.13}$$

where $\mathbf{y}(t)$ is given by (2.10).

Equations (2.5) and (2.13) describe entirely the file allocation problem according to Segall's continuous time analysis. The goal is to find the optimal controls $\alpha_{ij} (t)$ that minimize the cost function described by (2.13) subject to the set of differential equations (2.5).

Discrete-Time

The total cost per unit time over any operating period of length N is

$$C = E\left\{\sum_{t=1}^{N}\sum_{i=1}^{M}\left[C_i + \sum_{j\neq i} C_{ij}\, n_j(t)\right] y_i(t)\right\} \tag{2.14}$$

where the definition of C_i and C_{ij} is given in (2.12). Note that the expected value is used since the requests $n_i(t)$ are random variables. Define the request rates as

$$a_i(t) = Pr\left\{n_i(t) = 1\right\} \qquad i = 1,2,...,M \tag{2.15}$$

It is assumed that the request rates $a_i(t)$ are perfectly known. It can also be proved that

$$a_i(t) = E\left\{n_i(t)\right\} \tag{2.16}$$

For details see Appendix A. Thus the cost function (2.14) may be written in the form

$$C = \sum_{t=1}^{N}\sum_{i=1}^{M}\left[C_i + \sum_{j\neq i} C_{ij}\, a_j(t)\right] y_i(t) \tag{2.17}$$

As we assume that the constants $a_i(t)$ are all known, the cost function can be finally expressed in the form

$$C = \sum_{t=1}^{N} L_t\left[\mathbf{y}(t)\right] \tag{2.18}$$

Equations (2.6) and (2.18) describe entirely the file allocation problem according to Segall's approach. The goal is to find the optimal controls $\mathbf{u}(t)$ that minimize the cost function described by (2.18) subject to the set of difference equations (2.6).

2.3 Solution of the Problem

For the case of continuous time analysis, Segall proved that the optimal control variables $\alpha_{ij}(t)$ are given by

$$\alpha_{ij}(t) = \begin{cases} 0, & V(t,j) \geq V(t,i) \\ 1, & V(t,j) < V(t,i) \end{cases} \triangleq I_{(V(t,j)<V(t,i))} \tag{2.19}$$

where $i = 1,...,M; j = 1,...,M; i \neq j$ and $V(t,i)$ is given by the set of backward differential equations

$$-\frac{dV(t,i)}{dt} = \left[C_i + \sum_{j \neq i} C_{ij} \lambda_j(t)\right] + \sum_{j \neq i} \lambda_j(t)[V(t,j) - V(t,i)] I_{(V(t,j)<V(t,i))} \tag{2.20}$$

with terminal condition

$$V(N,i) = 0, \quad \text{for all } i, \quad i = 1,...,M \tag{2.21}$$

For the discrete time analysis of the problem the solution is given by using dynamic programming techniques (see Appendix B). Since there is a direct path between every pair of computers, a computer that needs the file simply broadcasts a request for the file over the network and the computer that actually has the file at that instant will receive the request within one sampling instant. For this reason Segall only studied in detail the case where there are just two computers passing one file back and forth. Finally he studied the same problem in the case that the request rates are not perfectly known. He assumed that the request rates for the file at the two computers are random Markov processes whose transition

probabilities are themselves random. Under these conditions he derived the optimal control policies of dynamic file allocation using dynamic programming techniques [Howard 1960].

Chapter 3
The Dynamic File Allocation Problem in a Three-Computer Network with a Linear Topology

3.1 Statement of the problem

In this chapter we study the dynamic file allocation problem by removing the assumption that the computer network is completely connected. As is the case with Segall's work, we make the following simplifying assumptions:

(1) The files are requested by the computers according to mutually independent processes

(2) The files are short compared to the memory of the computers

(3) Communication lines have sufficient capacity

(4) Computers have sufficient memory

Under these assumptions the problem of allocating any one file can be treated indepedently of that of allocating any other. Thus we also assume that there is only one file being passed back and forth. At any time, if a computer wishes access to the file and does not have it, it simply broadcasts a request for the file to a *central controller*. This central controller knows which computer has the file at that time instant and makes a decision (based on the optimality criteria described later) whether or not the file is to be sent to the computer that requested it.

This presupposes a parallel network for broadcasting requests, which seems to be a reasonable assumption for most existing networks. If a decision is made to transmit the file to the requesting computer, the file has to be transmitted over the network, and need not arrive at the requesting computer until several sampling instants later, since the network is not completely connected. In most existing computer networks, it seems to be the case that, while there need not be a direct path between every pair of computers, there is always a path of length no longer than two. Thus a computer that requested a file will receive it (if at all) no later than two sampling instants from the time that it requested it. Unlike the simple case treated by Segall, this assumption still does not narrow down the collection of network topologies very much. Since this is the first time that such a problem is being studied, we choose a configuration as shown in Figure 3.1.1, which might be referred to as a linear topology.

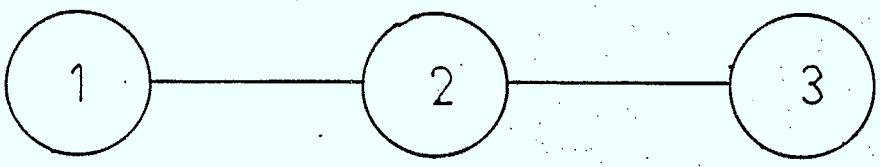


Figure 3.1.1 -A Computer Network with a Linear Topology

3.2 The Modeling Equations

In this section we derive the equations describing the file transfer between the three computers connected as shown in Figure 3.1.1. If computer i has the file at time t and there is a request from computer j for the file, then at time t a central controller decides whether or not the file is to be sent to computer j. If there is no request, or if only computer i wants the file at time t, then the file is left in computer i. On the other hand, if a decision is made to transmit the file to computer j at time t, the file itself need not arrive at computer j until time $t+2$, depending on the values of the indices i and j. Let us define some quantities as in Chapter 2. Let $y_i(t)$ be defined as

$$y_i(t) = \begin{cases} 1 & \text{if the file under consideration is held} \\ & \text{in memory of computer } i \text{ at time } t \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

where $i = 1,2,3$ and $t = 1,2,...$

Clearly, at any given time only one of the three variables $\{y_1(t), y_2(t), y_3(t)\}$ can be one and the other two will be zero.The requests of the file are again modelled as three independent Bernoulli processes $\{n_i(t), t = 1,2,... i = 1,2,3\}$ with rates exactly known (see Appendix A). The variables $n_i(t)$ are defined as

$$n_i(t) = \begin{cases} 1 & \text{if the file is requested by computer } i \text{ at time } t \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

where $i = 1,2,3$ and $t = 1,2,...$

Also, let us define the decision variables $\alpha_{ij}(t)$ as follows

$$\alpha_{ij}(t) = \begin{cases} 1 & \text{if, given that the file is in memory} \\ & \text{of computer } i \text{ at time } t \text{ and is requested} \\ & \text{by computer } j \text{, the decision is to transfer it} \\ & \text{to the memory of computer } j \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

Note that $\alpha_{ij}(t)$ is only defined for $i \neq j$. Then the dynamics of the file are described by the following equations:

$$\begin{aligned} y_1(t+1) = {} & y_1(t)[1 - n_2(t)\alpha_{12}(t) - n_3(t)\alpha_{13}(t)] \\ & + y_3(t-1)n_1(t-1)\alpha_{31}(t-1) \\ & + y_2(t)n_1(t)\alpha_{21}(t)[1 - y_3(t-1)n_1(t-1)\alpha_{31}(t-1) \\ & \qquad - y_1(t-1)n_3(t-1)\alpha_{13}(t-1)] \end{aligned} \tag{3.4}$$

$$
\begin{aligned}
y_2(t+1) = {} & y_1(t)[n_2(t)\alpha_{12}(t)+n_3(t)\alpha_{13}(t)] \\
& + y_3(t)[n_2(t)\alpha_{32}(t)+n_1(t)\alpha_{31}(t)] \\
& + y_2(t)[1-n_1(t)\alpha_{21}(t)-n_3(t)\alpha_{23}(t)] \\
& \quad [1-y_3(t-1)n_1(t-1)\alpha_{31}(t-1) \\
& \quad -y_1(t-1)n_3(t-1)\alpha_{13}(t-1)]
\end{aligned} \tag{3.5}
$$

$$
\begin{aligned}
y_3(t+1) = {} & y_3(t)[1-n_2(t)\alpha_{32}(t)-n_1(t)\alpha_{31}(t)] \\
& + y_1(t-1)n_3(t-1)\alpha_{13}(t-1) \\
& + y_2(t)n_3(t)\alpha_{23}(t)[1-y_3(t-1)n_1(t-1)\alpha_{31}(t-1) \\
& \quad -y_1(t-1)n_3(t-1)\alpha_{13}(t-1)]
\end{aligned} \tag{3.6}
$$

The explanation behind these equations is as follows: Let us begin with computer 1. In what ways can computer 1 have the file at time $t+1$? That is, in what ways can $y_1(t+1)$ equal 1? There are three possible ways, namely:

(i) Computer 1 has the file at time t, and there is either no request from the other two computers, or else there is a request but the central controller decides to let the file continue to reside in computer 1; this explains the first term on the right side of (3.4).

(ii) At time $t-1$, computer 3 has the file, computer 1 wants it, and the central controller decides that the file is to be sent from computer 3 to computer 1. Since the transit time between computers 1 and 3 is two sampling instants, the file requested at time $t-1$ only arrives at time $t+1$. This explains the second term on the right side of (3.4).

(iii) At time t, computer 2 has the file, computer 1 requests it, and the central controller decides to send it to computer 1. This partially explains the third term of the right side of (3.4). Now, in computing this term, we must ignore the situation where the file happens to reside in computer 2 at time t only because it is "en route," because this possibility is already accounted for by the second term. This explains the second part of the third term on the right side of (3.4). In the same way, suppose that the file was in computer 1 at time $t-1$, computer 3 requested it at that time, and that the central controller desided to send it to computer 3 at that time. As a subsequence, the file will be in computer 2 at time t. If now computer 1 requests the file at time t, it is assumed that the central controller has sufficient "memory" to let the file continue on its original course and not to divert it back to computer 1. This explains the last part of the third term on the right side of (3.4).

Now the equation (3.6) can be explained in an entirely analogous way, by interchanging the roles of computers 3 and 1. Finally we turn to computer 2. There are numerous ways in which the file can come to computer 2 at time $t+1$. First, it may be that the file is in computers 1 or

3 at time t , computer 2 requests it, and the central controller "grants" this request. Alternatively, the file may have been in computer 1 at time t , computer 3 requests it at the same time, and the file is in transit as a result of this request having been granted. The same situation may also occur with 1 and 3 interchanged. All of these terms are accounted for by the first and the second term on the right side of (3.5). The third term on the right side of (3.5) consists of two products. If the file is already in computer 2 at time t and there is a request from one of the other computers, then computer 2 can be directed by the central controller to send the file there, provided that the file did not just come from there on the way to the other computer. All in all, it is evident from the complexity of the equations above in comparison to those of Segall that the removal of the complete connectedness assumption substantially complicates the problem.

To put the equations above in a form suitable for application of dynamic programming, let us define the control variables:

$$u_{ij}(t) = n_j \alpha_{ij}(t) \tag{3.7}$$

where $i \neq j$, $i,j = 1,2,3$. Then the system of equations (3.4), (3.5) and (3.6) can be expressed in the form

$$\mathbf{y}(t+1) = \mathbf{f}(\mathbf{y}(t),\mathbf{y}(t-1),\mathbf{u}(t),\mathbf{u}(t-1)) \tag{3.8}$$

where the vector $\mathbf{y}(t)$ is given by

$$\mathbf{y}(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \end{bmatrix} \tag{3.9}$$

and the control vector $\mathbf{u}(t)$ is given by

$$\mathbf{u}(t) = \begin{bmatrix} u_{12}(t) \\ u_{13}(t) \\ u_{21}(t) \\ u_{23}(t) \\ u_{31}(t) \\ u_{32}(t) \end{bmatrix} \tag{3.10}$$

Finally, if we define the new state vector $\mathbf{x}(t)$ as

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{y}(t) \\ \mathbf{y}(t-1) \end{bmatrix} \tag{3.11}$$

then the set of equations (3.8) describing the dynamics of the file can be expressed in the simpler form:

$$\mathbf{x}(t+1) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{u}(t-1)) \tag{3.12}$$

where the definition of the function $\mathbf{g}$ is evident. Now, the vector $\mathbf{x}(t)$ has six components, each of which is either 0 or 1. This suggests that there are $2^6 = 64$ possible states. However, it is routine to verify that most of these combinations do not make sense, and there are in fact only seven possible states of the system, namely:

$$\begin{bmatrix}1\\0\\0\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\\1\\0\end{bmatrix}, \begin{bmatrix}1\\0\\0\\0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\\0\\1\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\\0\\1\end{bmatrix}$$

The above seven possible states of the computer network under study describe the following situations respectively: (i) the file being in memory of computer 1 at time $t-1$, remains in the same location at time t, (ii) the file being in memory of computer 1 at time $t-1$, is transferred to computer 2 at time t, (iii) the file being in memory of computer 2 at time $t-1$, remains in the same location at time t, (iv) the file being in memory of computer 2 at time $t-1$, is transferred to computer 1 at time t, (v) the file being in memory of computer 2 at time $t-1$, is transferred to computer 3 at time t, (vi) the file being in memory of computer 3 at time $t-1$, remains at the same location at time t, and (vii) the file being in memory of computer 3 at time $t-1$, is transferred to computer 2 at time t.

In the same way, the control vector $\mathbf{u}(t)$ is a 6-dimensional vector whose components are all either 0 or 1, but again there are only seven possible control vectors, namely:

$$\begin{bmatrix}0\\0\\0\\0\\0\\0\end{bmatrix},\ \begin{bmatrix}1\\0\\0\\0\\0\\0\end{bmatrix},\ \begin{bmatrix}0\\1\\0\\0\\0\\0\end{bmatrix},\ \begin{bmatrix}0\\0\\1\\0\\0\\0\end{bmatrix},\ \begin{bmatrix}0\\0\\0\\1\\0\\0\end{bmatrix},\ \begin{bmatrix}0\\0\\0\\0\\1\\0\end{bmatrix},\ \begin{bmatrix}0\\0\\0\\0\\0\\1\end{bmatrix}$$

Note that the zero vector corresponds to the situation where there is no request from any computer, or there is some request, but the central controller decides to keep the file at the same location where it was at time $t-1$. Let us use the symbols X, U to denote the set of possible states and the set of possible controls, respectively.

Finally, let us consider the cost function to be minimized. Let C_i denote the storage cost per unit time in computer i, and C_{ij} denote the cost of transmission from computer i to computer j. Then the total cost per unit time over any operation period of length N is:

$$J = \frac{1}{N}\sum_{t=1}^{3}\Big[\sum_{i=1}^{3} C_i\, y_i(t) + y_1(t)\big[C_{12}n_2(t) + (C_{12}+C_{23})n_3(t)\big]$$

$$+ y_2(t)\big[C_{21}n_1(t) + C_{23}n_3(t)\big]$$

$$+ y_3(t)\big[C_{32}n_2(t) + (C_{32}+C_{21})n_1(t)\big]\Big] \qquad (3.13)$$

In Equation (3.13) each term can be explained as follows: The first term within the brackets is the expression for the overall storage cost during the operation period N. The second term within the brackets is the expression for the situation where the file is stored in the memory of computer 1 and it is requested by the other computers. Note that with the

request $n_3(t)$ of the 3rd computer a cost of value $C_{12}+C_{23}$ is associated since the file has to be transmitted through the paths 1-2 and 2-3. An analogous explanation can be given to the third and the forth terms. Clearly, the overall transmission cost is expressed by the last three terms within the brackets of equation (3.13). Since the requests are random variables, the true cost function is the expected value of the quantity J in (3.13). Define the request rate $a_i(t)$ as

$$a_i(t) = Pr\left[n_i(t) = 1 \right] \quad \text{where } i = 1,2,3 \tag{3.14}$$

Then it is easy to see that the expected value of J, which we again denote by J, is given by (3.13) with the $n_i(t)$ replaced by $a_i(t)$ (see Appendix A). As the constants $a_i(t)$ are all known, the cost function can be expressed in the form

$$J = \sum_{t=1}^{N} L_t(\mathbf{x}(t)) \tag{3.15}$$

3.3 Solution Using Backward Dynamic Programming

In this section we show how the problem formulated in the preceding section may be solved using the technique of backward dynamic programming. In Appendix B the backward dynamic programming technique for a class of dynamic systems is presented. The objective here is to show how this technique may be readily adapted to the situation where there are delays in the control variables. Suppose the system under study is described by

$$\mathbf{x}(t+1) = \mathbf{g}(\mathbf{x}(t),\mathbf{u}(t),\mathbf{u}(t-1)) \tag{3.16}$$

and the objective is to minimize the cost function

$$J = \sum_{t=1}^{N} L_t(\mathbf{x}(t),\mathbf{u}(t)) + h(\mathbf{x}(N+1)) \tag{3.17}$$

Note that the function L_t changes when t changes. The problem described by (3.16) and (3.17) is somewhat more general than the one posed in the previous section, in that there is no explicit dependence on the control variable in the cost function (3.15) and there is also no terminal cost. However, it turns out that the theory is no more complicated for this case than for the case described by (3.15), so that we chose this more general cost function. Clearly, the variables to be chosen to achieve the minimum are $\mathbf{u}(1),\mathbf{u}(2),\ldots,\mathbf{u}(N)$. It is important to note that the quantity $\mathbf{x}(2)$ depends not only on the (presumably known) initial state $\mathbf{x}(1)$ and the control variable $\mathbf{u}(1)$, but also on $\mathbf{u}(0)$. Accordingly we assume that both $\mathbf{x}(1)$ and $\mathbf{u}(0)$ are given. To solve this problem by backward dynamic programming, define the "cost to go" $Q_i(\mathbf{x},\mathbf{u})$ as

$$Q_i(\mathbf{x},\mathbf{u}) = \min_{\mathbf{u}(i),\mathbf{u}(i+1),\ldots,\mathbf{u}(N)} \left\{ \sum_{t=i}^{N} L_t(\mathbf{x}(t),\mathbf{u}(t)) + h(\mathbf{x}(N+1)) \right\} \tag{3.18}$$

subject to the conditions $\mathbf{u}(i-1) = \mathbf{u}$, $\mathbf{x}(i) = \mathbf{x}$ (where $\mathbf{x}$ and $\mathbf{u}$ are known). Then

$$Q_N(\mathbf{x},\mathbf{u}) = \min_{\mathbf{u}(N)} \left\{ L_N(\mathbf{x},\mathbf{u}(N)) + h(\mathbf{g}(\mathbf{x},\mathbf{u}(N),\mathbf{u})) \right\} \tag{3.19}$$

since $\mathbf{x}(N+1) = \mathbf{g}(\mathbf{x}(N),\mathbf{u}(N),\mathbf{u}(N-1)) = \mathbf{g}(\mathbf{x},\mathbf{u}(N),\mathbf{u})$.

Finally, for $i = N-1,N-2,...,1$ we have

$$Q_i(\mathbf{x},\mathbf{u}) = \min_{\mathbf{u}(i)} \left\{ L_i(\mathbf{x},\mathbf{u}(i)) + \min_{\mathbf{u}(i+1),...,\mathbf{u}(N)} \left\{ \sum_{t=i+1}^{N} L_t(\mathbf{x}(t),\mathbf{u}(t)) + h(\mathbf{x}(N+1)) \right\} \right\}$$

Now, it is easy to see that, according to the definition (3.19), the second term within the big braces is simply the analytic expression of the term $Q_{i+1}(\mathbf{x}(i+1))$. So Equation (3.19) may be written as

$$Q_i(\mathbf{x},\mathbf{u}) = \min_{\mathbf{u}(i)} \left\{ L_i(\mathbf{x},\mathbf{u}(i)) + Q_{i+1}(\mathbf{g}(\mathbf{x},\mathbf{u}(i),\mathbf{u}),\mathbf{u}(i)) \right\} \qquad (3.20)$$

for $i = N-1,N-2,...,1$. Thus, the optimal controls $\mathbf{u}(i)$ can be found by solving backwards the set of equations (3.20) and considering Equation (3.19) as the initial step. We can see that it is quite easy to solve for $\mathbf{u}(i)$ at each of the N steps of the procedure. Note that we have to check only seven possible controls and we can choose the one that minimizes the term on the right side of Equation (3.20) The optimal locations of the file during the operating period of length N can be found by substituting the optimal controls $\{\mathbf{u}(i), i = 1,2,...,N\}$ into the equations (3.12). Finally, the minimum cost function $J_{\min}$ is given by $Q_1(\mathbf{x},\mathbf{u})$, which is determined during the last step of the backward procedure.

Chapter 4
Simulation Results

In this chapter numerical results on the application of the dynamic file assignment studied in Chapter 3 are presented. In particular, the effects of the parameters of the system (request rates, storage and transmission costs) are studied in detail. The effect of the number of time instants for the discrete analysis of the problem, as well as the effect of the initial location of the file are also studied in sections 4.2 and 4.3 respectively. The system is considered over an operation period of 24 hours. If N denotes the number of the discrete instants of time within this operation period, then clearly the central controller decides about the optimal location of the file every $\frac{24}{N}$ hours. The variables $a_i(t)$ (request rates) can take integer values, denoting how many times per hour the computer i requests the file of interest. Note that all the variables $a_i(t)$ are considered as known a priori. The storage costs C_i are expressed in dollars per second (for the file of interest) and the transmission costs C_{ij} are expressed in dollars per transmission (for the file of interest). As an application of the method developed in Chapter 3, the optimal allocation of the file is presented during the whole operation period of 24 hours. Also, the total minimum expected cost is given in dollars.

4.1 The Effect of the Request Rates on the Optimal Allocation

In this section the effect of the request rates of each computer is studied, by giving several examples with different request rates and keeping the same values for the other parameters of the system. A discussion about these examples follows.

Example 4.1

As a first example, we consider the case where the file is requested by only one computer at any time. The request rates pattern for each of the three computers over 24 hours is given in Figure 4.1.1. The values of the other parameters of the system are shown in Table 4.1.1. The optimal allocation of the file, according to the method studied in Chapter 3 is given in Figure 4.1.2. Finally, the minimum expected operation cost for this example is found to be $ 86.77.

Table 4.1.1 - Summary of data for Example 4.1

$N = 96$	Initial location : Computer 1
$C_1 =$ \$ 0.001 per second	$C_{12} =$ \$ 0.5 per transmission
$C_2 =$ \$ 0.001 per second	$C_{21} =$ \$ 0.5 per transmission
$C_3 =$ \$ 0.001 per second	$C_{23} =$ \$ 0.5 per transmission
	$C_{32} =$ \$ 0.5 per transmission

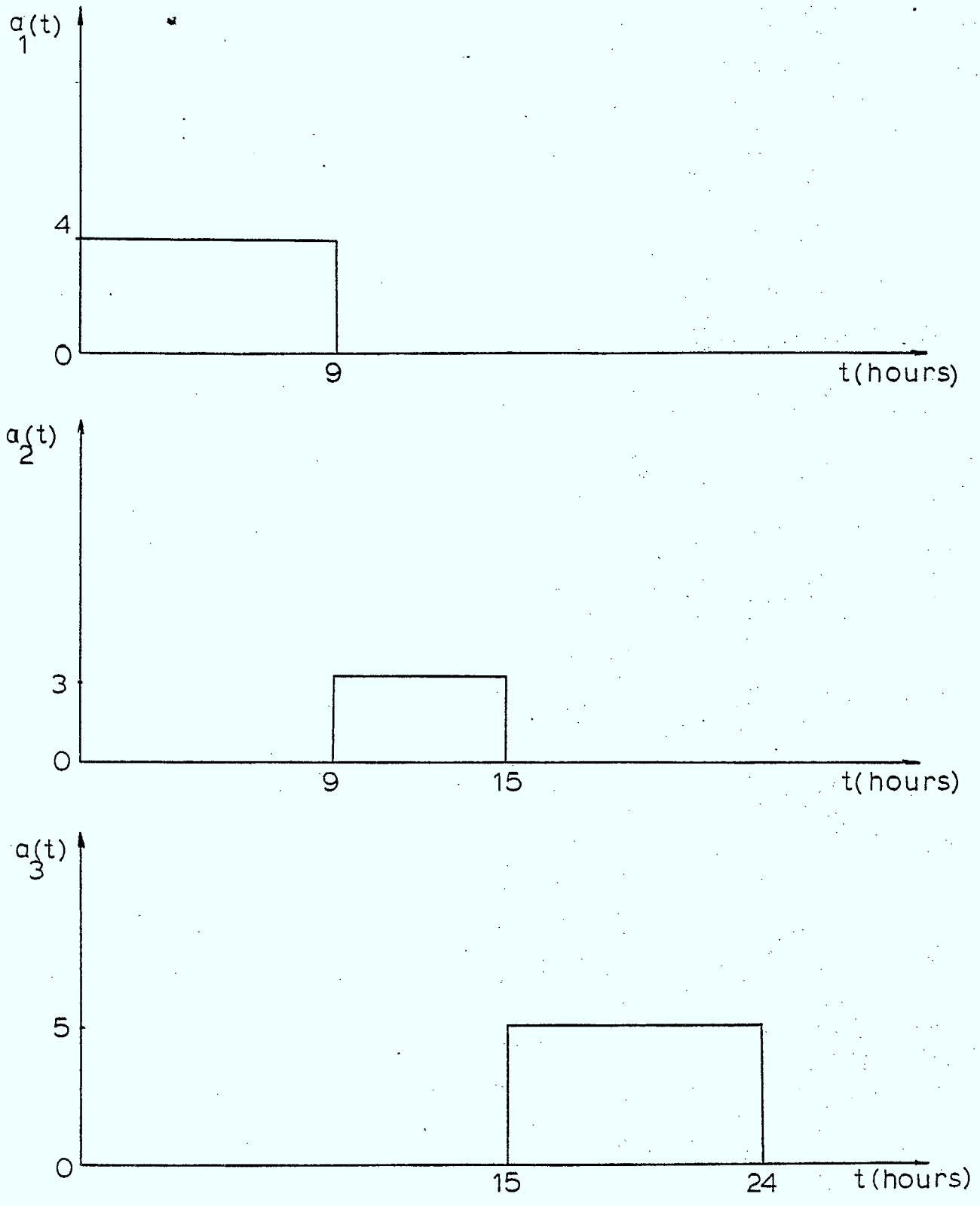


Figure 4.1.1 -Request rates for Example 4.1

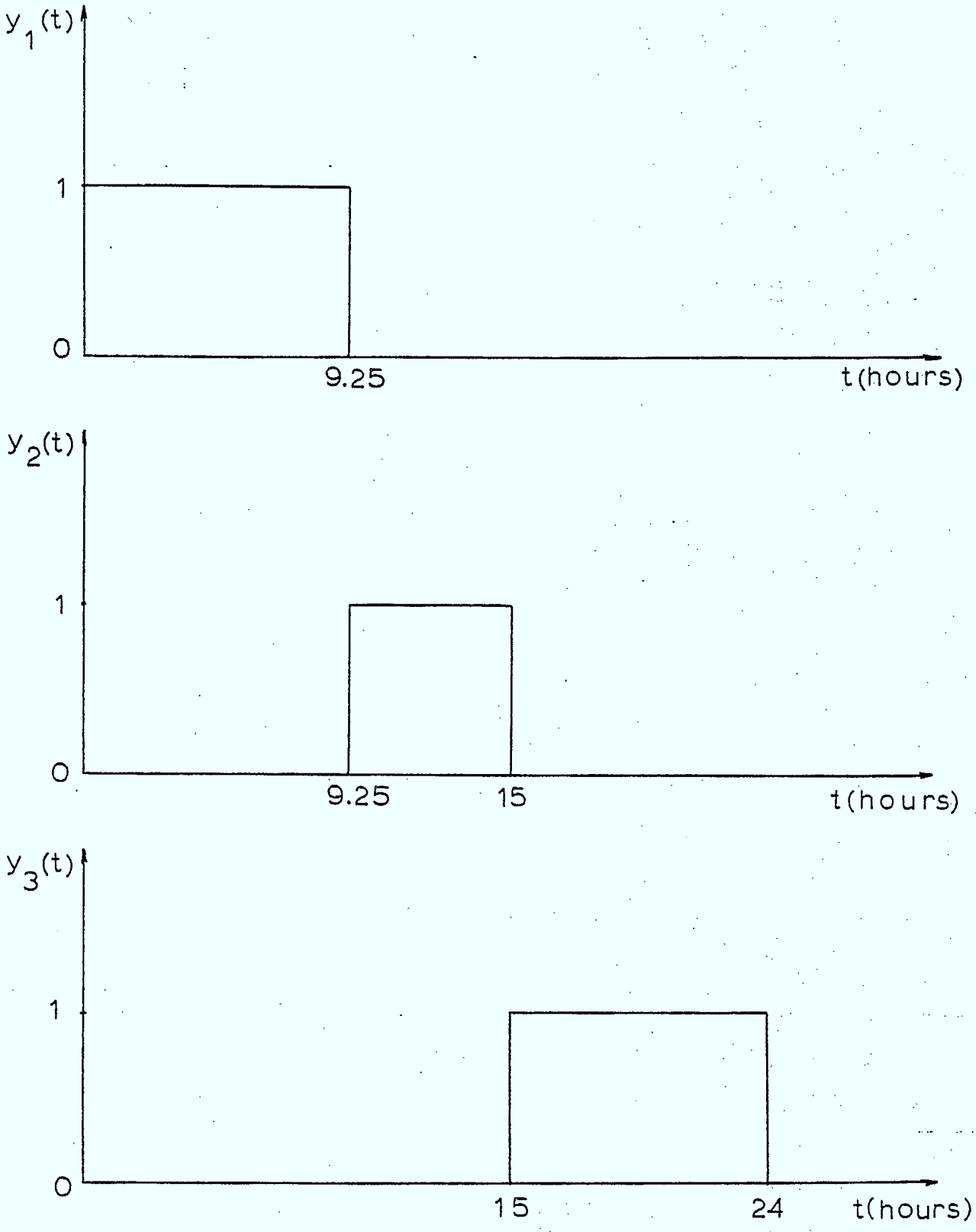


Figure 4.1.2 -Optimal file allocation for Example 4.1

By examining the results for the first example, we can see that the optimal location of the file at any time t is in the memory of the computer that requests it at that time, since there is no other request at the same time. The situation becomes complicated in the case where more than one computers requests the file at the same time. This situation is considered in the following examples.

Example 4.2

In this example, we consider that eventually more than one computers may request the file at the same time. Figure 4.1.3 shows the pattern of the request rates for each of the three computers, over a period of 24 hours. We can see for instance that the file is requested by all the computers during the period 9-15 hours. The values of the other parameters of the system are the same as in Example 1.1 (see Table 4.1.1). The optimal allocation of the file is given in Figure 4.1.4. In this case, the minimum expected total cost is $ 142.27.

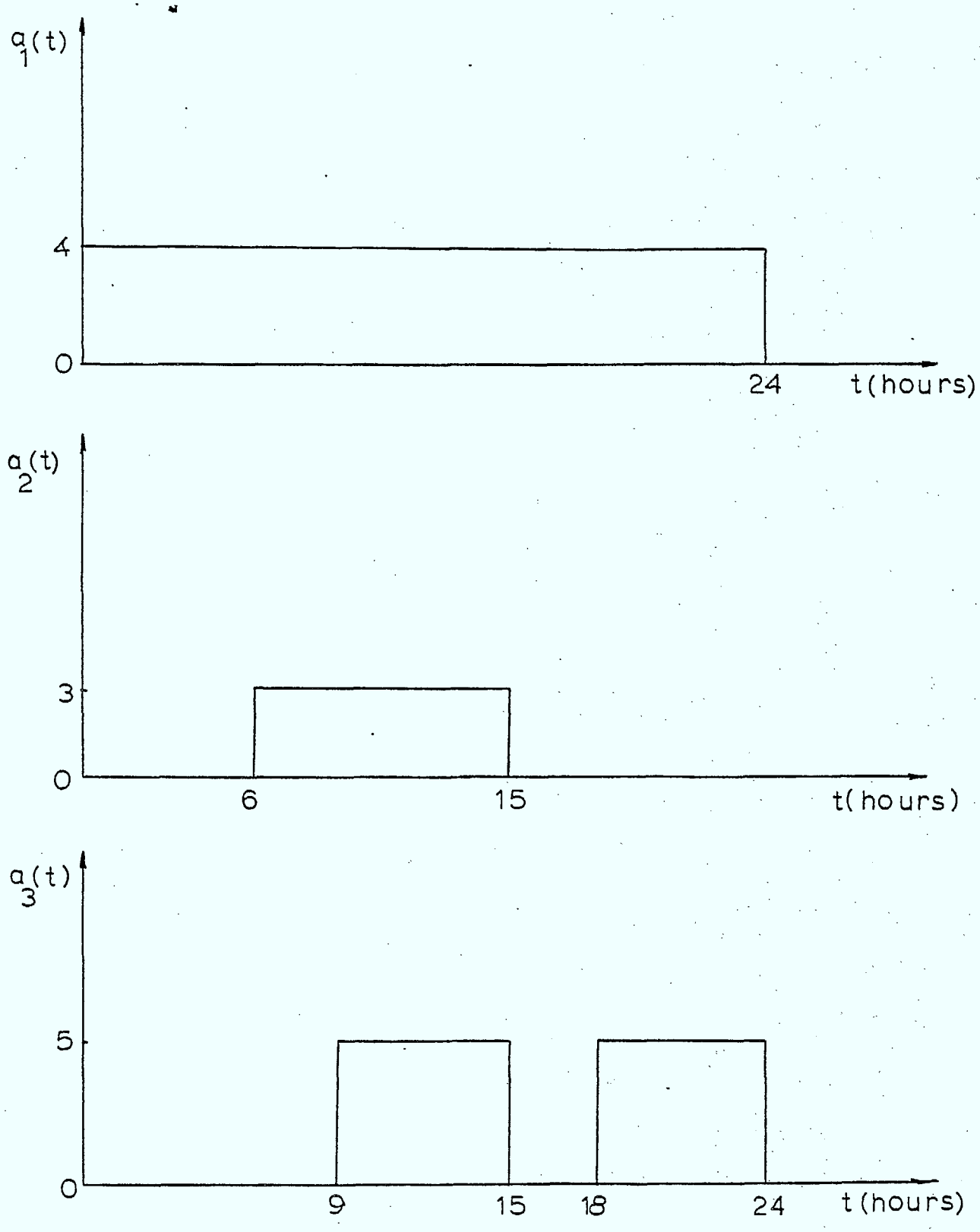


Figure 4.1.3 -Request rates for Example 4.2

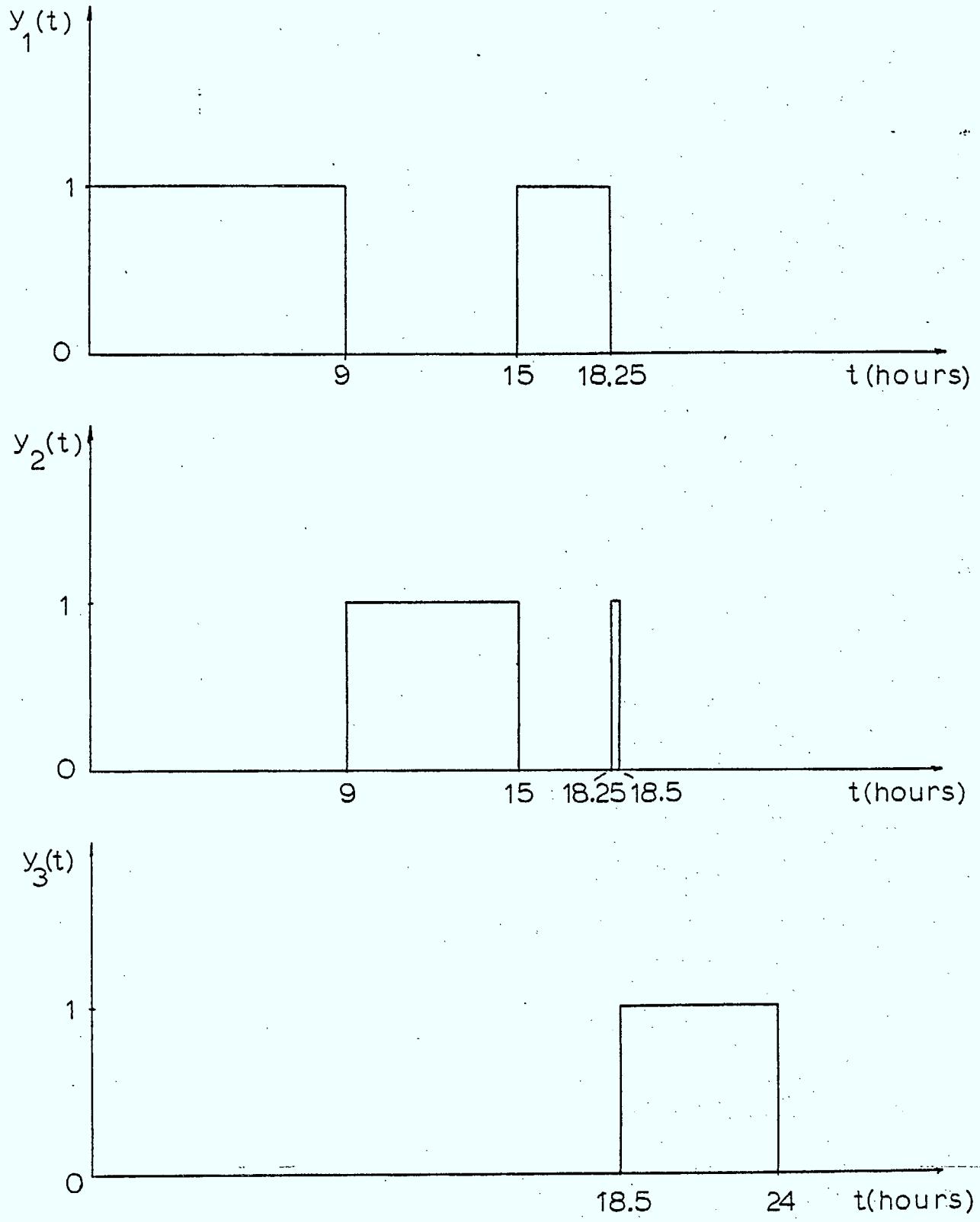


Figure 4.1.4 -Optimal file allocation for Example 4.2

By examining the results for the second example, we can see that for the period 0-6 hours, the central contrtoller decides that the file has to be stored in the memory of computer 1, since computer 1 is the only computer that requests the file during that period of time. For the period 6-9 hours the central controller decides that the file has to be stored also in the memory of computer 1. During that period, computer 2 also requests the file, but its request rate is lower than the request rate of computer 1 for the same period of time. The most interesting part of this example is following after. During the period 9-15 hours, all the computers may request the file, and the central controller decides that the file has to be stored in the memory of computer 2, though it requests the file with the lowest rate. The same situation is examined in following examples. During the period 15-18.5 hours, the file is stored in the memory of computer 1, and during the period 18.5-24 hours, it is stored in the memory of computer 3, since the request rate of computer 3 is higher than that of computer 1 for this period of time. Note that the file is stored in the memory of computer 2 for one time instant (15 minutes) as it is transferred from computer 1 to computer 3.

Example 4.3

In this example, we consider that the situation where all the computers may request the file at the same time occurs more frequently. The pattern of the request rates for each of the three computers is shown in Figure 4.1.5 and the values of the other parameters of the

system are the same as in the previous examples (Table 4.1.1). The optimal allocation of the file is given in Figure 4.1.6. In this case the minimum expected total cost is $129.27.

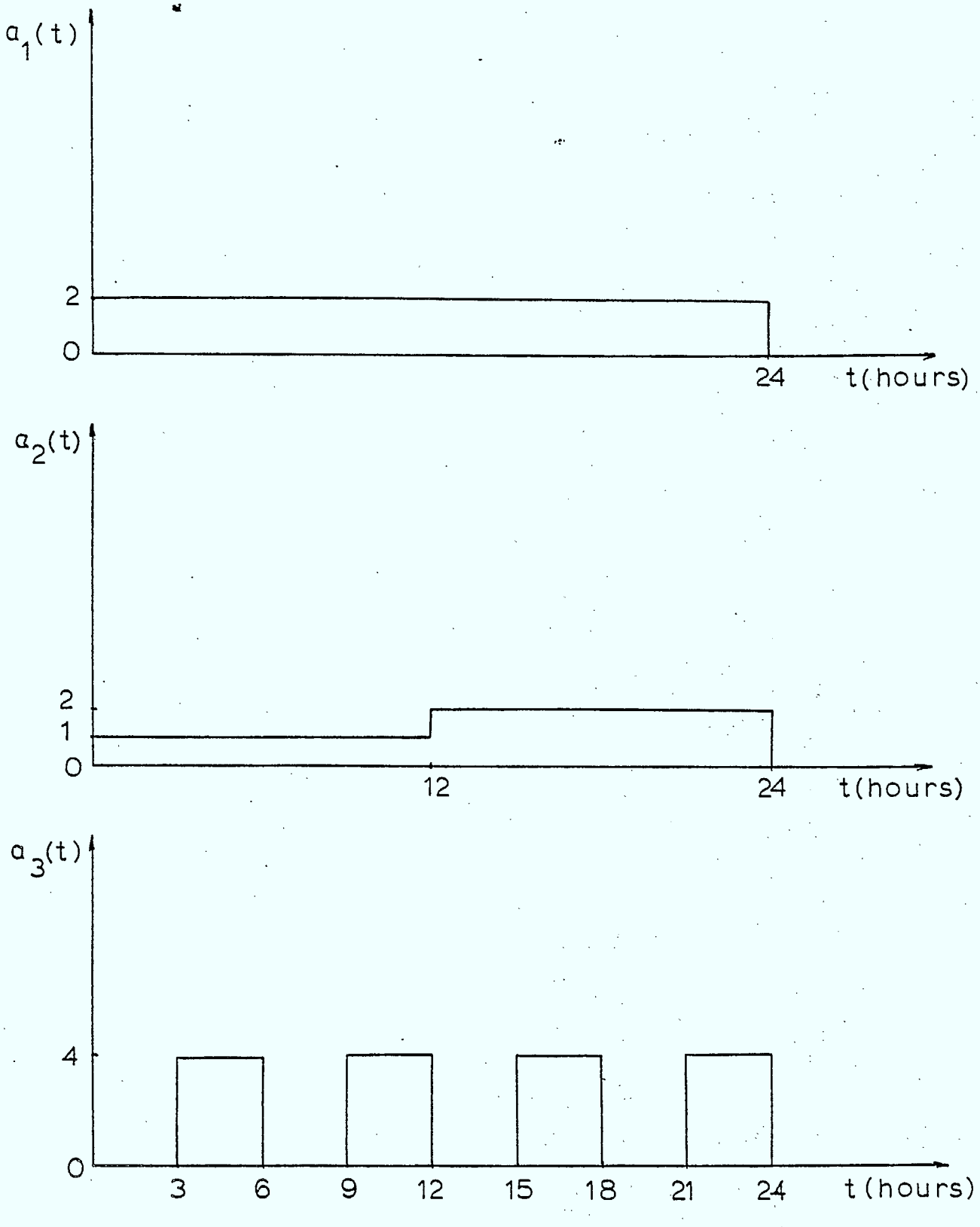


Figure 4.1.5 -Request rates for Example 4.3

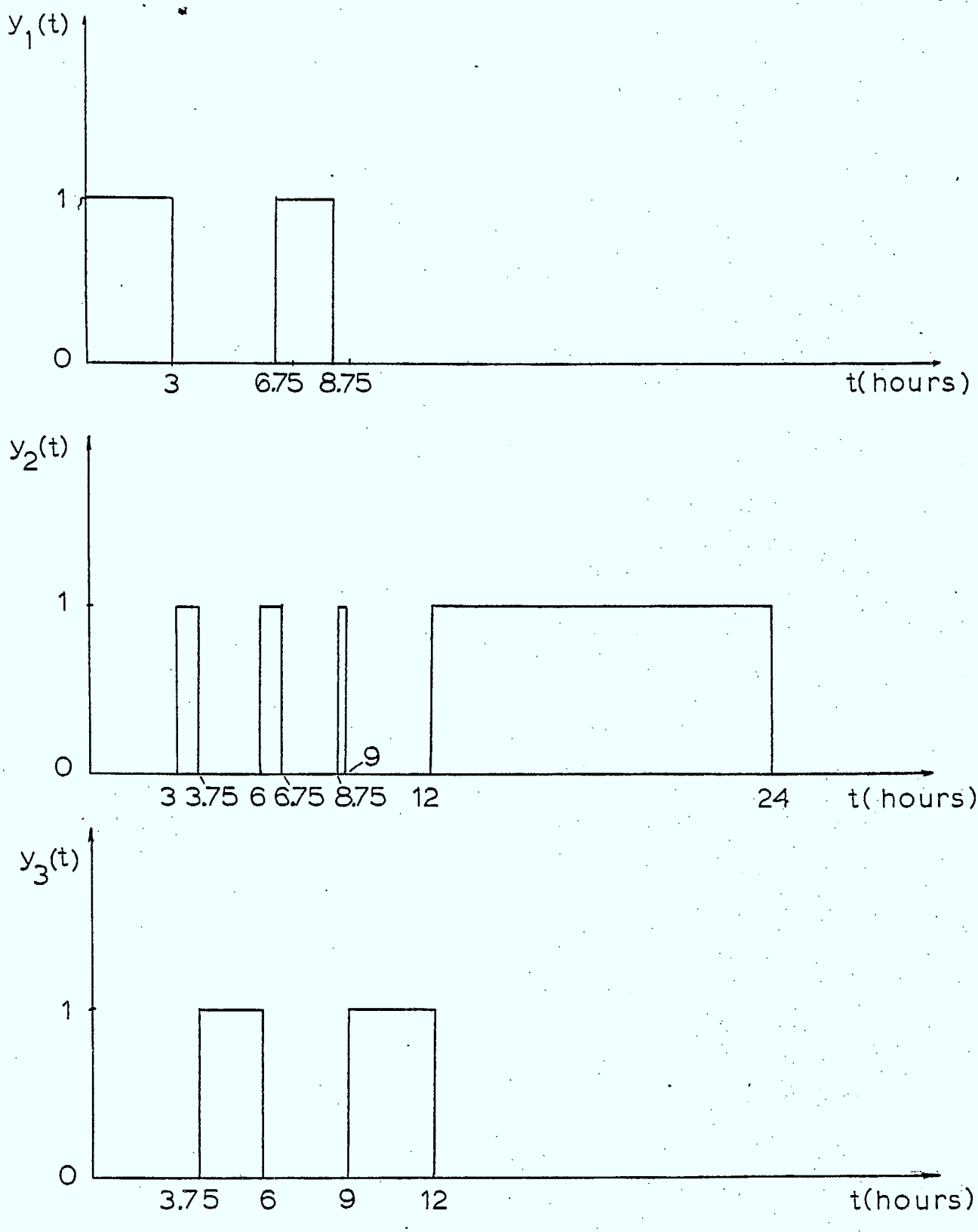


Figure 4.1.6 -Optimal file allocation for Example 4.3

In this example, all the computers may request the file at the same time during four different periods of time. The central controller decides that the file has to be stored in the memory of computer 3 during the periods 3.75-6 hours and 9-12 hours. We can see that for these periods the request rates (per hour) of the three computers 1, 2 and 3 are $a_1(t) = 2$, $a_2(t) = 1$, $a_3(t) = 4$ respectively. For the period 12-24 hours, the central controller decides that the file has to be stored in the memory of computer 2. For this period of time, the request rates (per hour) of the three computers 1, 2 and 3 are $a_1(t) = 2$, $a_2(t) = 2$, $a_3(t) = 4$ respectively. By examining these results, one can conclude that, in the case where all the computers may request the file at the same time, the central controller always decides that the file has to to be stored in the memory of computer 2, except for the case where the request rate of computer 1 (computer 3) is higher than the sum of the request rates of the two other computers. Then, the file has to be stored in the memory of computer 1 (computer 3). Note that this holds only in the case where all the other parameters of the system remain the same as in Table 4.1.1.

Example 4.4

As a last example for this part, we consider that the pattern of the request rates for each of the three computers is given in Figure 4.1.7 and the values of the other parameters of the system are the same as in Table 4.1.1. The optimal allocation of the file for this case is given in Figure 4.1.8 and the minimum expected cost is found to be $156.40.

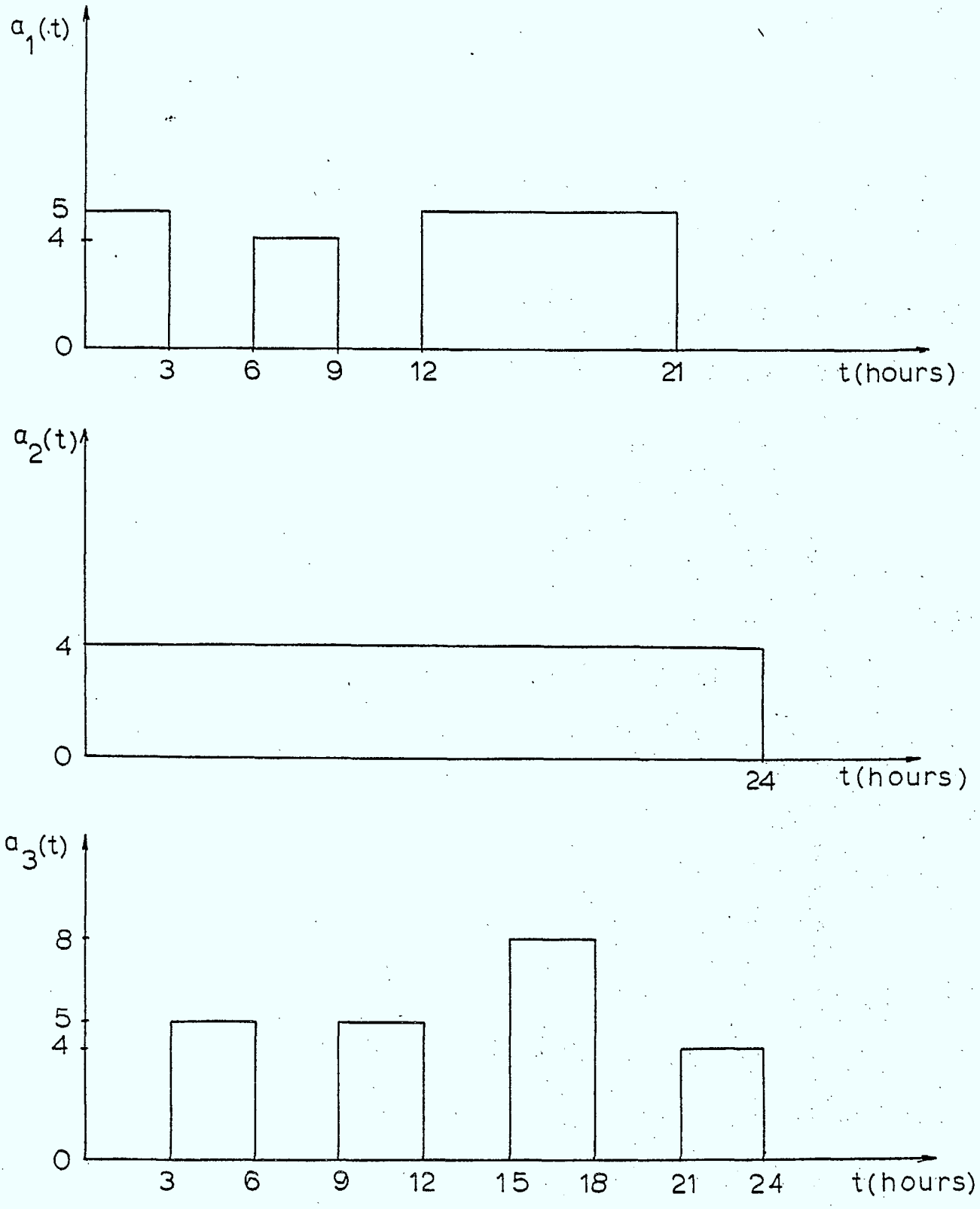


Figure 4.1.7 -Request rates for Example 4.4

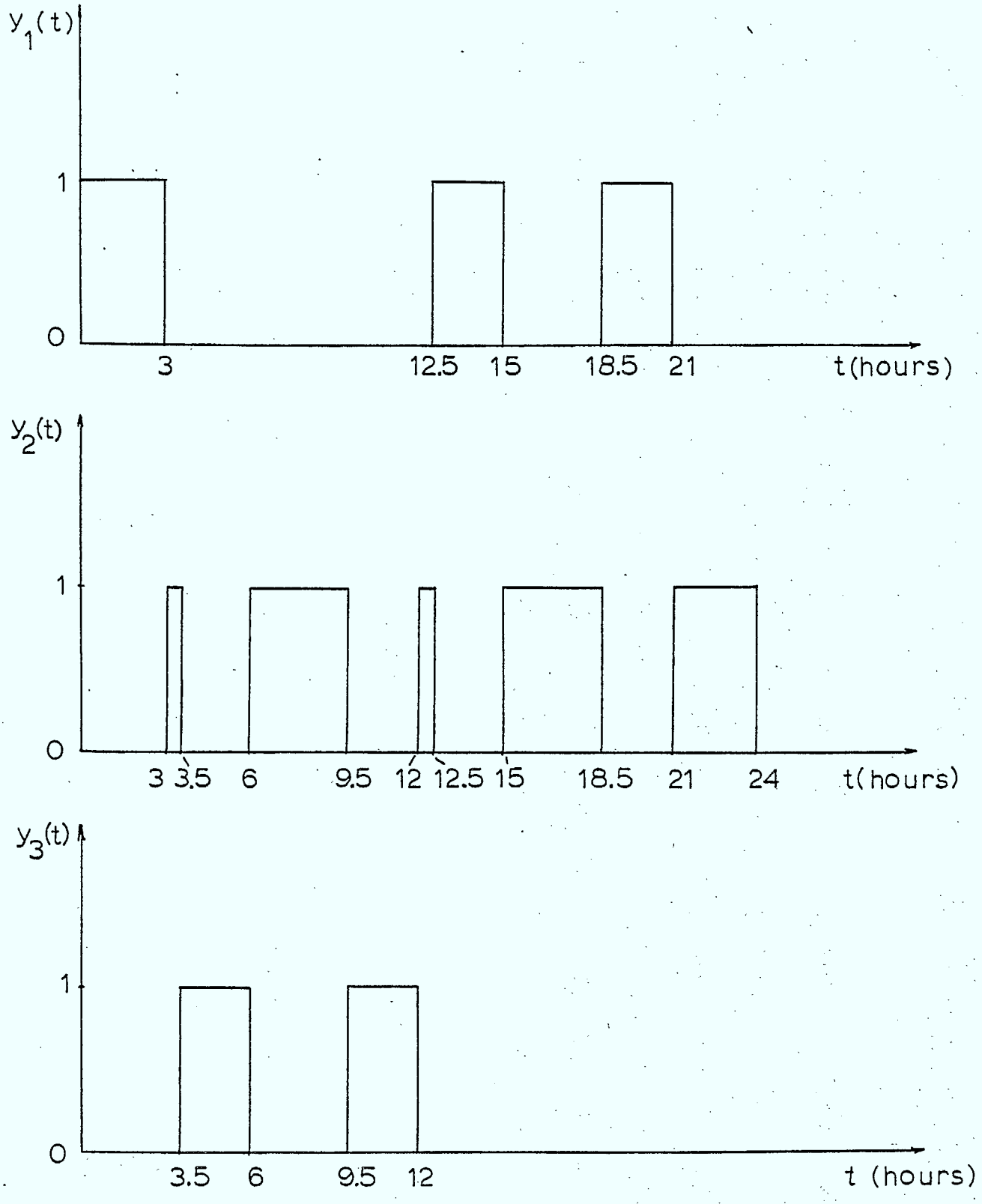


Figure 4.1.8 -Optimal file allocation for Example 4.4

By examining the results of Figure 4.1.8, we can see that these are in accordance to our previous conclusions. It is interesting to notice that during the periods 6-9 hours and 15-18 hours it is decided that the file has to be stored in the memory of computer 2. This supports our conclusion made in the discussion of Example 4.3

4.2 The Effect of the Number of Time Instants N on the Total Cost

In this section the effect of the number of time instants N for the discrete analysis of the problem is studied, by giving several examples with different N and keeping the same values for the other parameters of the system. A discussion about these examples follows.

Example 4.5

Let us suppose that the request rates pattern for each of the three computers is given in Figure 4.2.1. We also suppose that the number of the discrete instants of time within the period of 24 hours, is 48. This means that the location of the file is decided every 30 minutes. The optimal allocation of the file is given in Figure 4.2.2 and the minimum expected cost for this example is \$114.15.

Table 4.2.1 -Summary of data for Example 4.5

$N = 48$	Initial location : Computer 1
$C_1 =$ \$ 0.001 per second	$C_{12} =$ \$ 0.5 per transmission
$C_2 =$ \$ 0.001 per second	$C_{21} =$ \$ 0.5 per transmission
$C_3 =$ \$ 0.001 per second	$C_{23} =$ \$ 0.5 per transmission
	$C_{32} =$ \$ 0.5 per transmission

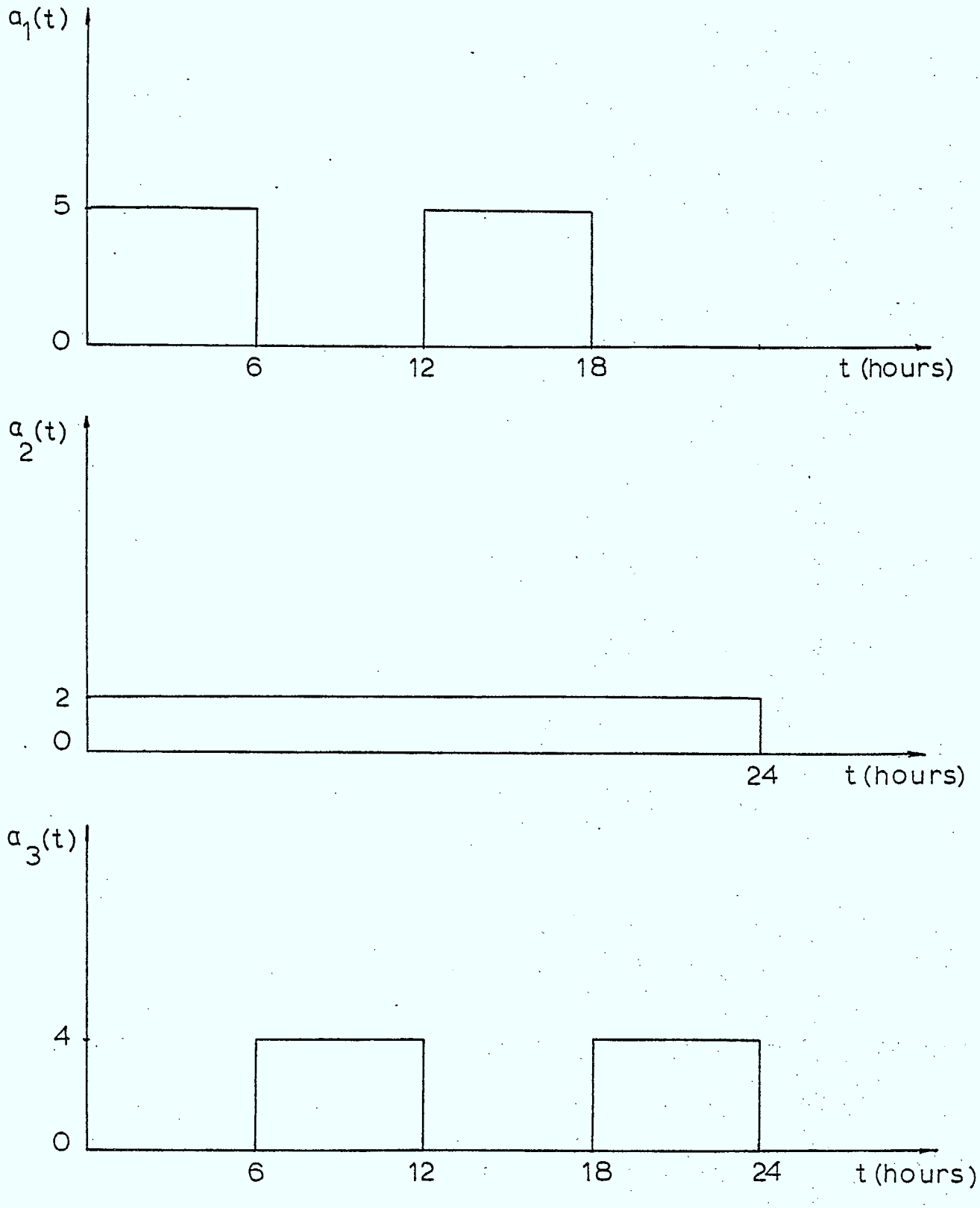


Figure 4.2.1 -Request rates for Example 4.5

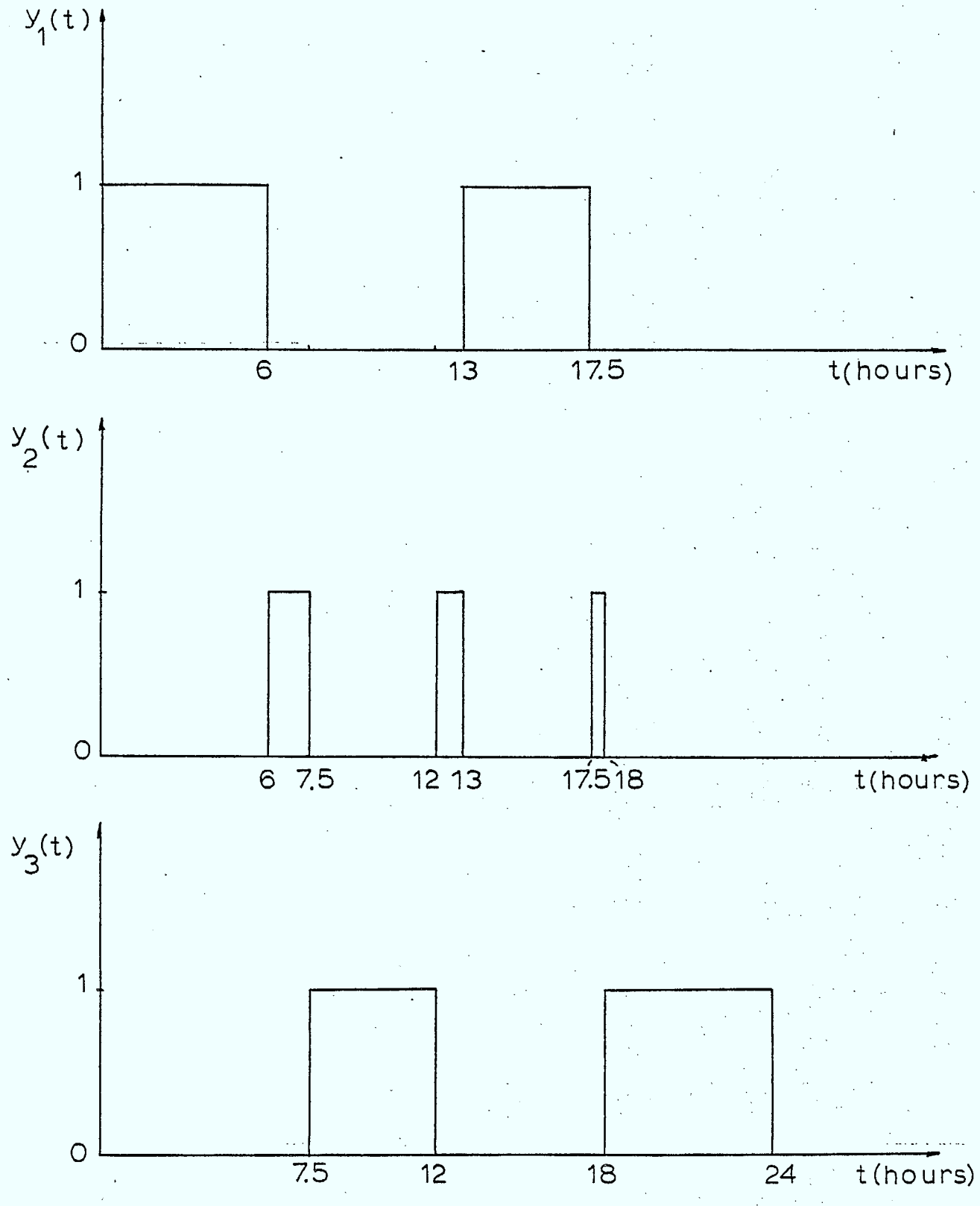


Figure 4.2.2 -Optimal file allocation for Example 4.5

By examining the results of Figure 4.2.2, we can see that the file remains in the memory of computer 2 for three instants of time during 6-7.5 hours as it is transferred from computer 1 to computer 3. It is also remains in the memory of computer 2 during the periods 12-13 hours (2 instants of time) and 17.5-18 hours (1 time instant) as it is transferred from computer 3 to computer 1 and vice versa.

Example 4.6

The pattern of the request rates for each of the three computers is the same as in Figure 4.2.1. We now suppose that $N = 96$. In this case, the location of the file is decided every 15 minutes. The optimal allocation of the file is given in Figure 4.2.3 and the total cost is found to be \$112.27. The values of the other parameters of the system are the same as in Table 4.1.1.

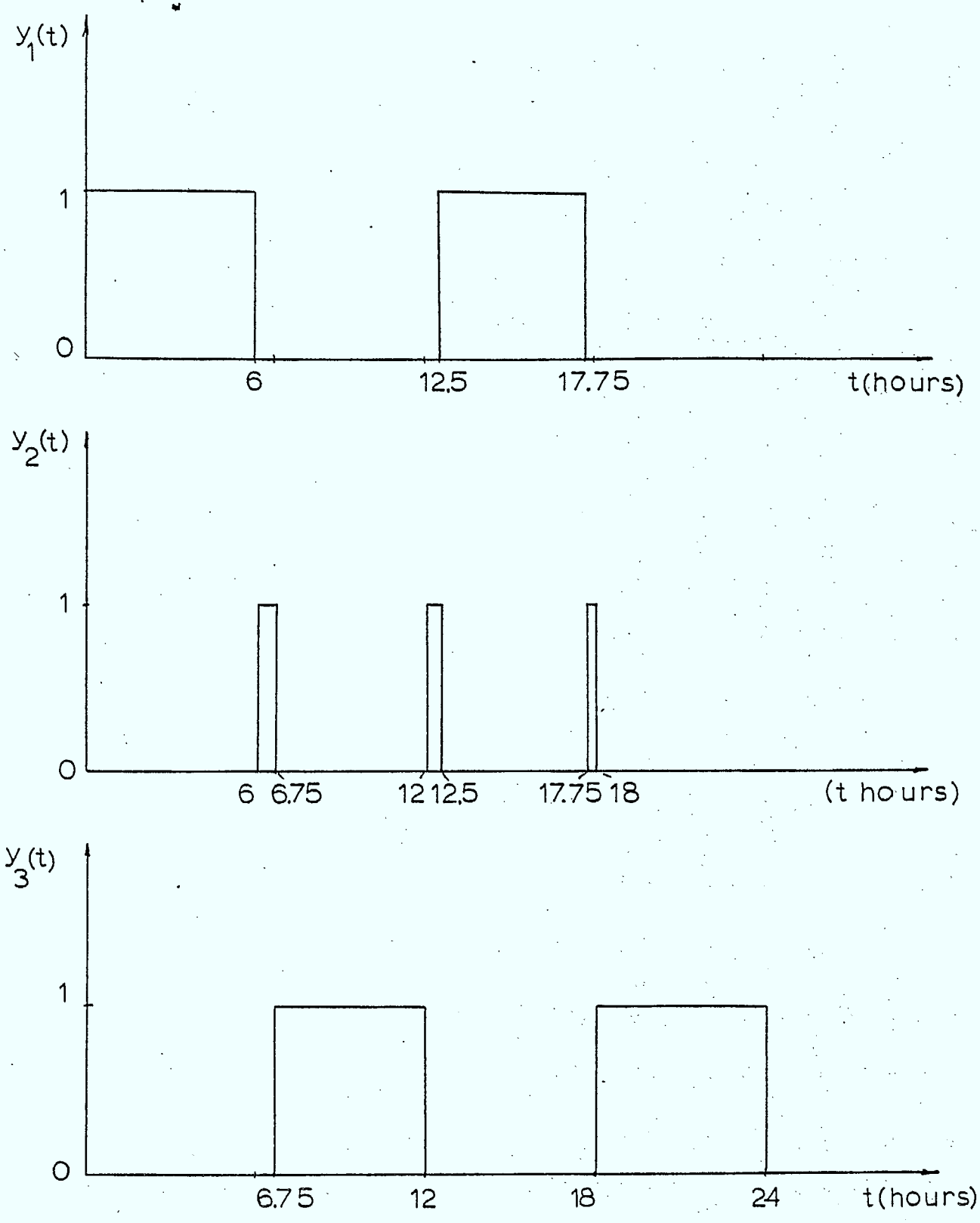


Figure 4.2.3 -Optimal file allocation for Example 4.6

If we compare the results of Figure 4.2.3 with those of Figure 4.2.2, we can see that the only difference is that, as N increases, the file is stored in the memory of computer 2 for a shorter period of time. Also, the total cost is lower in the case of Example 4.6.

Example 4.7

The pattern of the request rates for each of the three computers is given in Figure 4.2.1. We now suppose that $N = 288$. This means that the location of the file is decided every 5 minutes. The optimal allocation of the file is given in Figure 4.2.4 and the total cost for this example is found to be \$111.02.

Table 4.2.2 -Summary of data for Example 4.7

$N = 288$	Initial location : Computer 1
$C_1 = \$$ 0.001 per second	$C_{12} = \$$ 0.5 per transmission
$C_2 = \$$ 0.001 per second	$C_{21} = \$$ 0.5 per transmission
$C_3 = \$$ 0.001 per second	$C_{23} = \$$ 0.5 per transmission
	$C_{32} = \$$ 0.5 per transmission

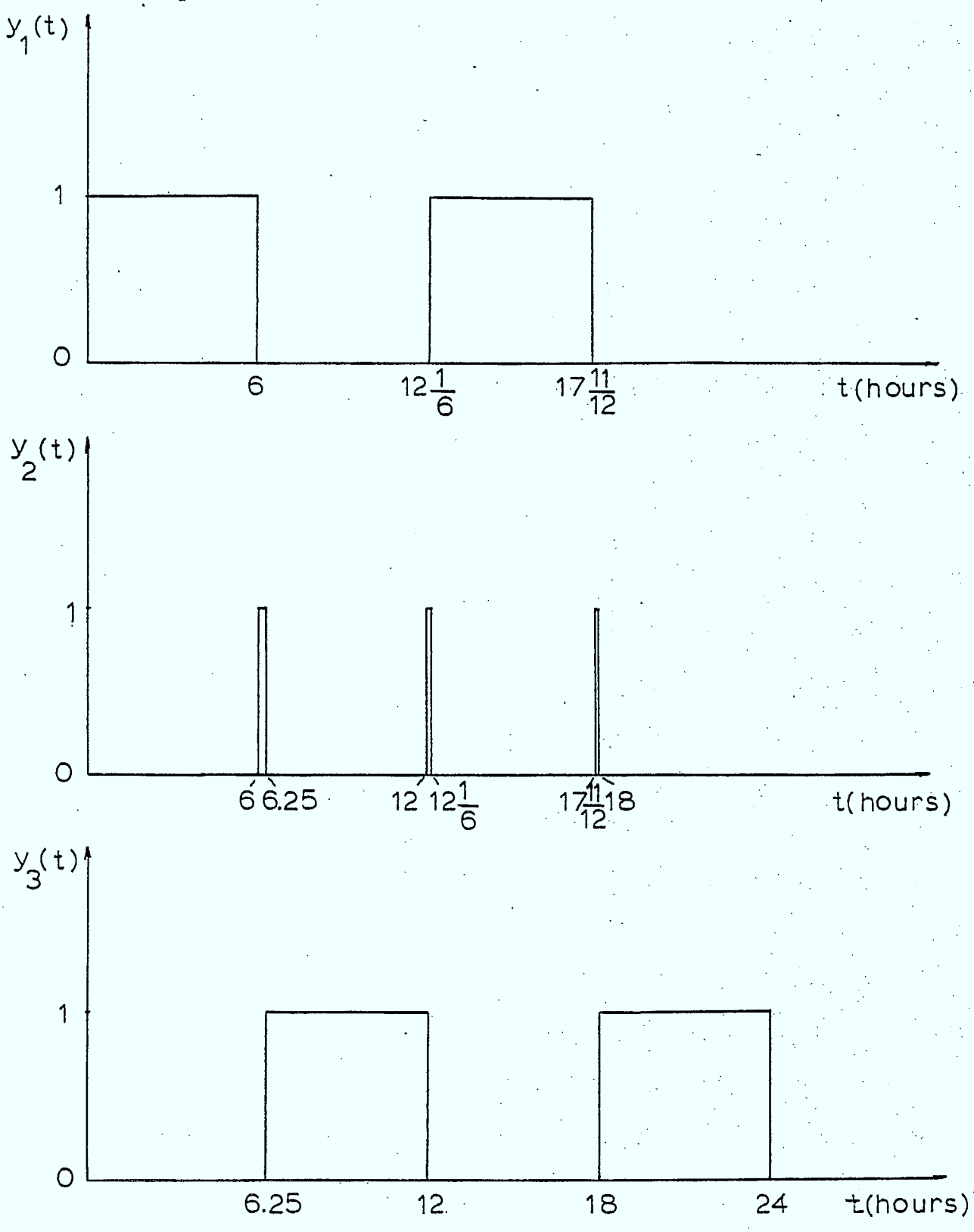


Figure 4.2.4 -Optimal file allocation for Example 4.7

If compare the results of Figure 4.2.4 with those of Figures 4.2.2 and 4.2.3, we can see that the file is stored in the memory of computer 2 for a shorter period of time. We can also notice a further decrease of the total cost. Here, we may mention that, as N increases, the total operation cost decreases, but the central controller has to decide about the optimal location of the file more frequently. So, there is a trade-off between the total operation cost of our application and the central controller operation cost.

4.3 The Effect of the Initial Location of the File on the Optimal Allocation

In this section the effect of the initial location of the file is studied, by giving several examples with different initial locations and keeping the same values for the other parameters of the system. A discussion about the results of these examples follows.

Example 4.8

Suppose that the pattern of the request rates for each of the three computers is the same as in Figure 4.1.1. We now suppose that the file is initially located in the memory of computer 2. The optimal allocation of the file is given in Figure 4.3.1. The minimum expected operation cost is found to be $87.77.

Table 4.3.1 -Summary of data for Example 4.8

$N = 96$	Initial location : Computer 2
$C_1 = \$\ 0.001$ per second	$C_{12} = \$\ 0.5$ per transmission
$C_2 = \$\ 0.001$ per second	$C_{21} = \$\ 0.5$ per transmission
$C_3 = \$\ 0.001$ per second	$C_{23} = \$\ 0.5$ per transmission
	$C_{32} = \$\ 0.5$ per transmission

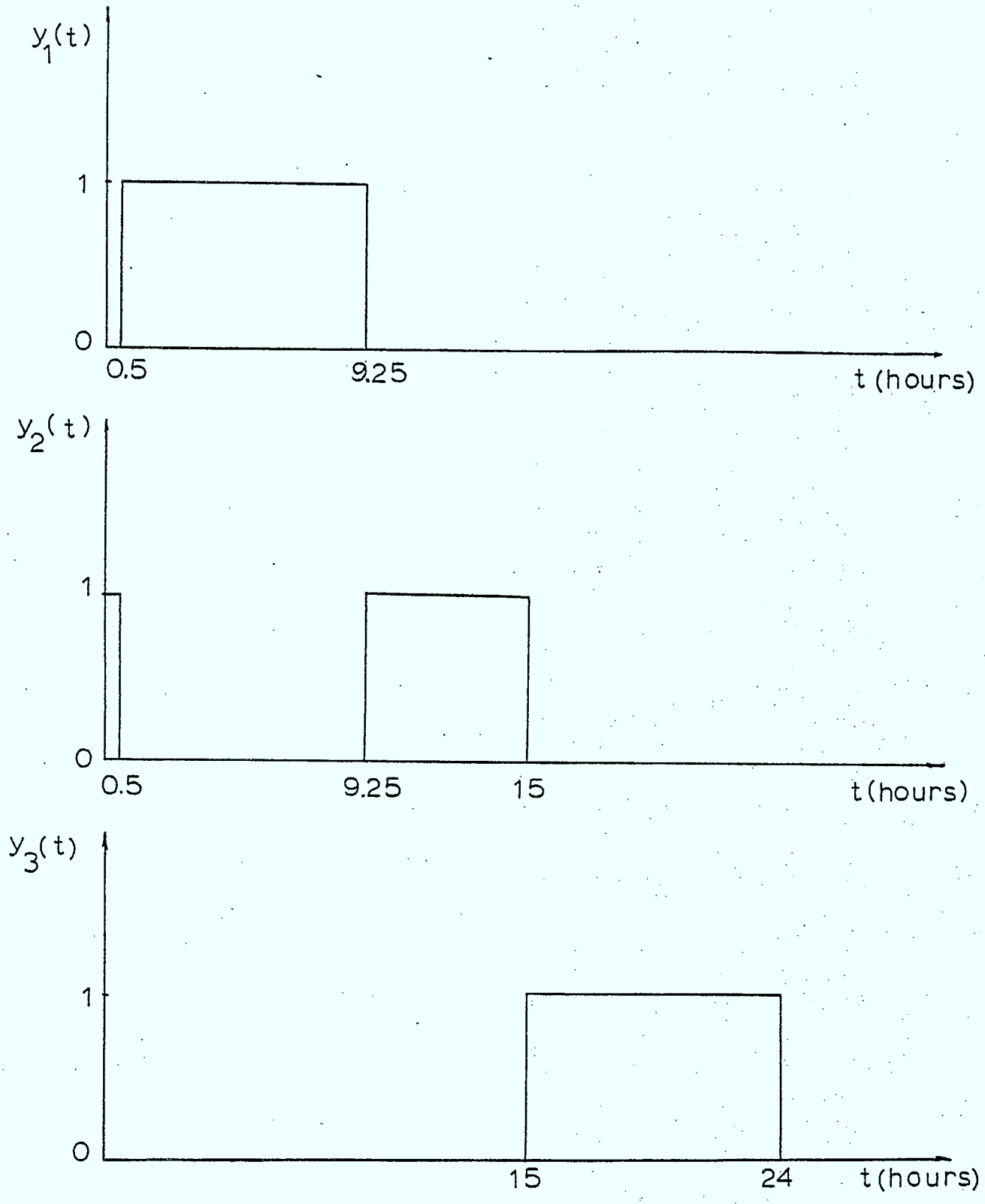


Figure 4.3.1 -Optimal file allocation for Example 4.8

We can see in Figure 4.3.1 that the file is stored in the memory of computer 2 for two time instants and then it is transferred to the memory of computer 1. After this, the results are the same as those in Figure 4.1.2. Note that this initial change results in a higher total operation cost.

Example 4.9

Suppose that the pattern of the request rates for each of the three computers is the same as in Figure 4.1.1. Now we suppose that the file is initially located in the memory of computer 3. The optimal allocation of the file is given in Figure 4.3.2. The minimum expected cost is found to be $90.27.

Table 4.3.2 -Summary of data for Example 4.9

$N = 96$	Initial location : Computer 3
$C_1 = \$\ 0.001$ per second	$C_{12} = \$\ 0.5$ per transmission
$C_2 = \$\ 0.001$ per second	$C_{21} = \$\ 0.5$ per transmission
$C_3 = \$\ 0.001$ per second	$C_{23} = \$\ 0.5$ per transmission
	$C_{32} = \$\ 0.5$ per transmission

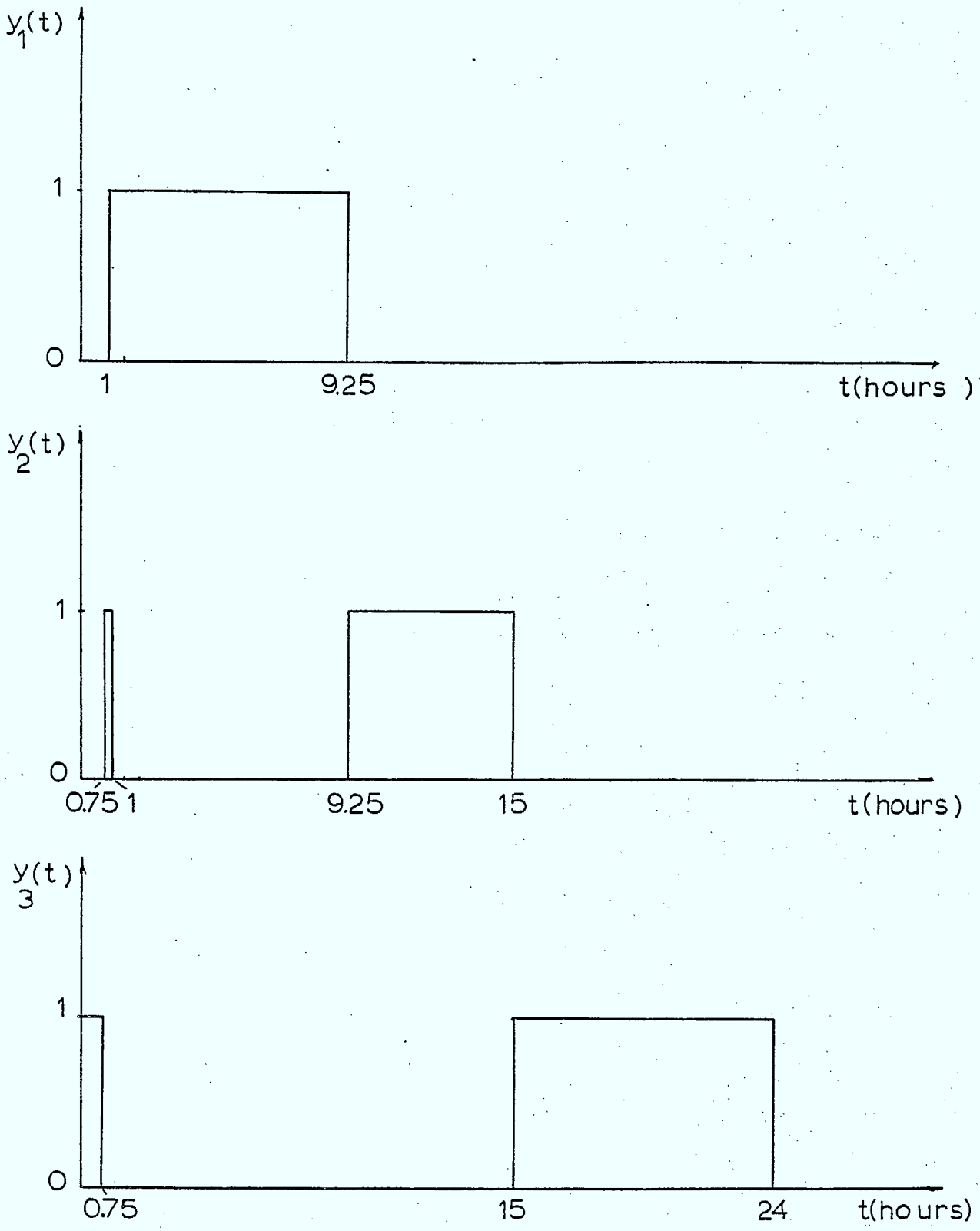


Figure 4.3.2 -Optimal file allocation for Example 4.9

By examining the results of Figure 4.3.2, we can see that the file is stored in the memory of computer 3 for the first three time instants and then it is transferred to computer 1 through computer 2. After this, the results are the same as those in Figure 4.1.2. Note that the total cost is now higher than that of Example 4.8.

4.4 The Effect of the Storage Cost on the Optimal Allocation

In this section the effect of the storage costs is studied, by giving several examples with different values of storage costs and keeping the same values for the other parameters of the system. A discussion about the results of these examples follows.

Example 4.10

We suppose that the request rates pattern for each of the three computers is the same as in Figure 4.1.1. The values of the other parameters of the system are shown in Table 4.4.1. Note that in this example the storage cost in computer 1 is three times more than the storage cost in the other computers. The optimal allocation of the file is given in Figure 4.4.1 and the total cost is found to be \$109.60.

Table 4.4.1 -Summary of data for Example 4.10

$N = 96$	Initial location : Computer 1
$C_1 = \$$ 0.003 per second	$C_{12} = \$$ 0.5 per transmission
$C_2 = \$$ 0.001 per second	$C_{21} = \$$ 0.5 per transmission
$C_3 = \$$ 0.001 per second	$C_{23} = \$$ 0.5 per transmission
	$C_{32} = \$$ 0.5 per transmission

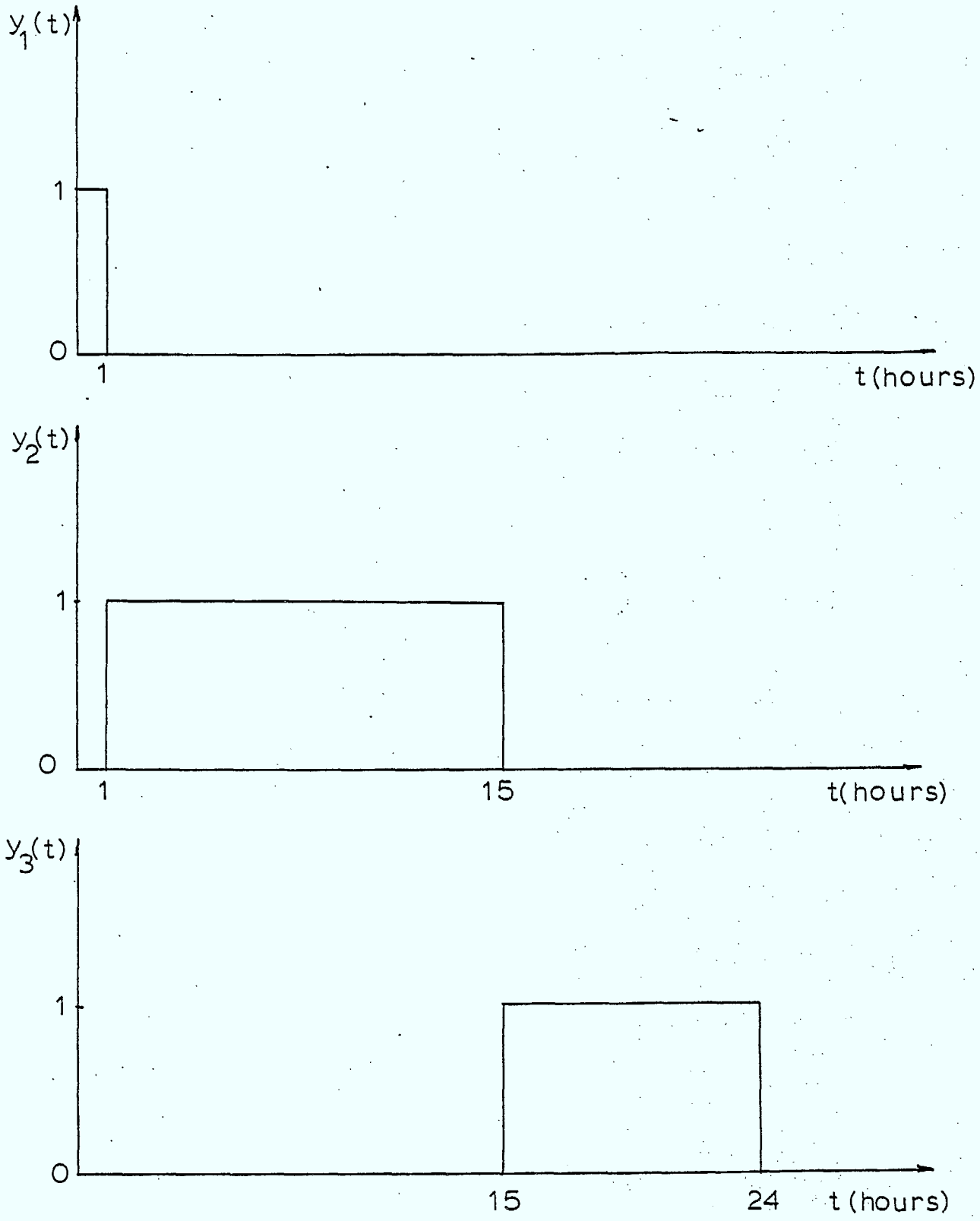


Figure 4.4.1 -Optimal file allocation for Example 4.10

By examining the results of Figure 4.4.1, we can see that the file is stored in the memory of computer 1 only for one hour, and then it is transferred to computer 2. This is reasonable, since the storage cost is higher in computer 1.

Example 4.11

We use the same request rates pattern as before and suppose that the storage cost in computer 2 is three times more than the storage cost in the other computers. The optimal allocation of the file is given in Figure 4.4.2 and the minimum expected total cost is \$97.82.

Table 4.4.2 -Summary of data for Example 4.11

$N = 96$	Initial location : Computer 1
$C_1 = \$\ 0.001$ per second	$C_{12} = \$\ 0.5$ per transmission
$C_2 = \$\ 0.003$ per second	$C_{21} = \$\ 0.5$ per transmission
$C_3 = \$\ 0.001$ per second	$C_{23} = \$\ 0.5$ per transmission
	$C_{32} = \$\ 0.5$ per transmission

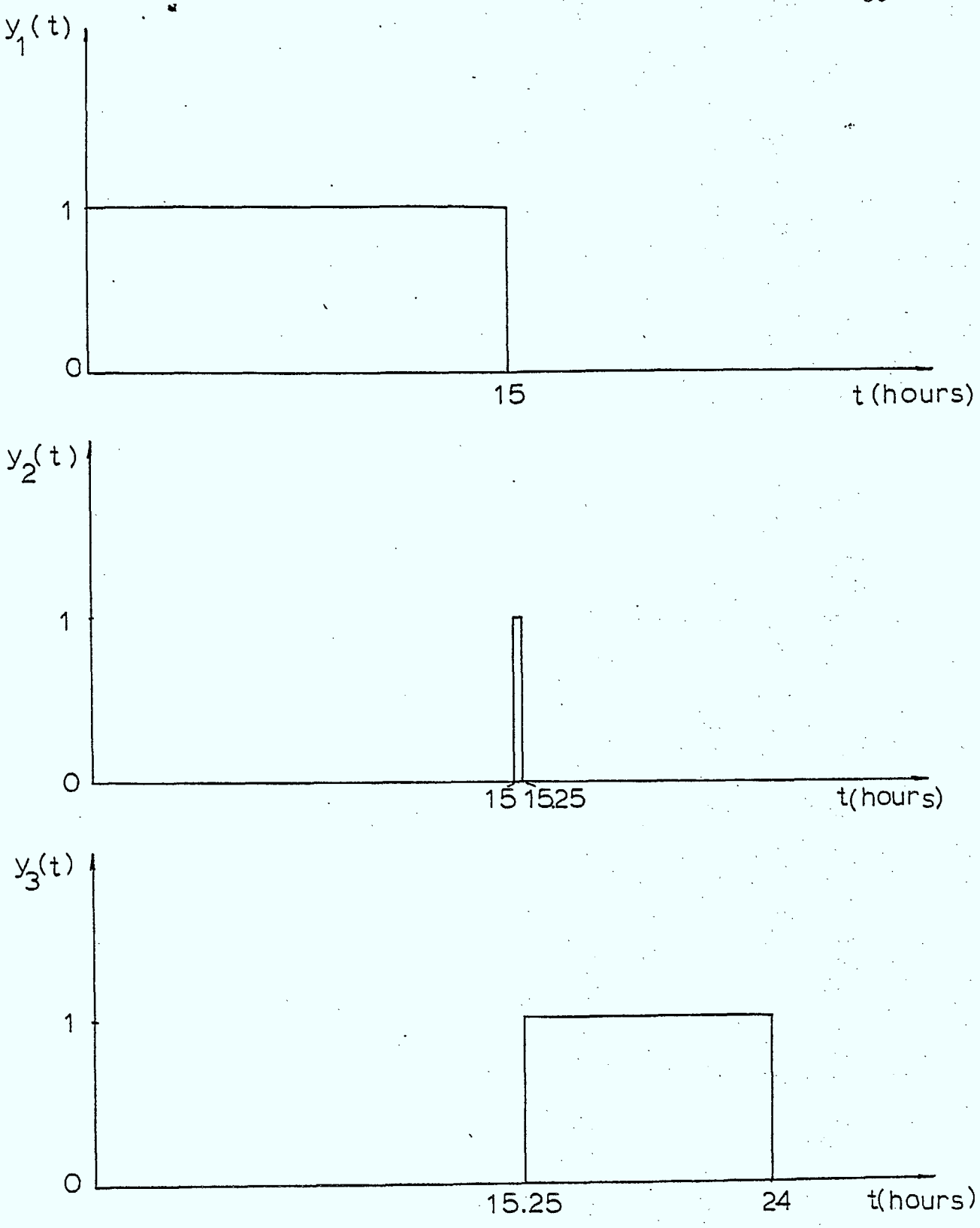


Figure 4.4.2 -Optimal file allocation for Example 4.11

We can see in Figure 4.4.2 that the file is transferred from computer 1 to computer 3 through computer 2, where it remains only for one time instant.

Example 4.12

We use again the same pattern for the request rates for each of the three computers as in the previous example. Now, we suppose that the storage cost in computer 3 is three times more than the storage cost in the other computers. The optimal allocation of the file is shown in Figure 4.4.3 and the total cost is found to be \$108.90.

Table 4.4.3 -Summary of data for Example 4.12

$N = 96$	Initial location : Computer 1
$C_1 = \$\ 0.001$ per second	$C_{12} = \$\ 0.5$ per transmission
$C_2 = \$\ 0.001$ per second	$C_{21} = \$\ 0.5$ per transmission
$C_3 = \$\ 0.003$ per second	$C_{23} = \$\ 0.5$ per transmission
	$C_{32} = \$\ 0.5$ per transmission

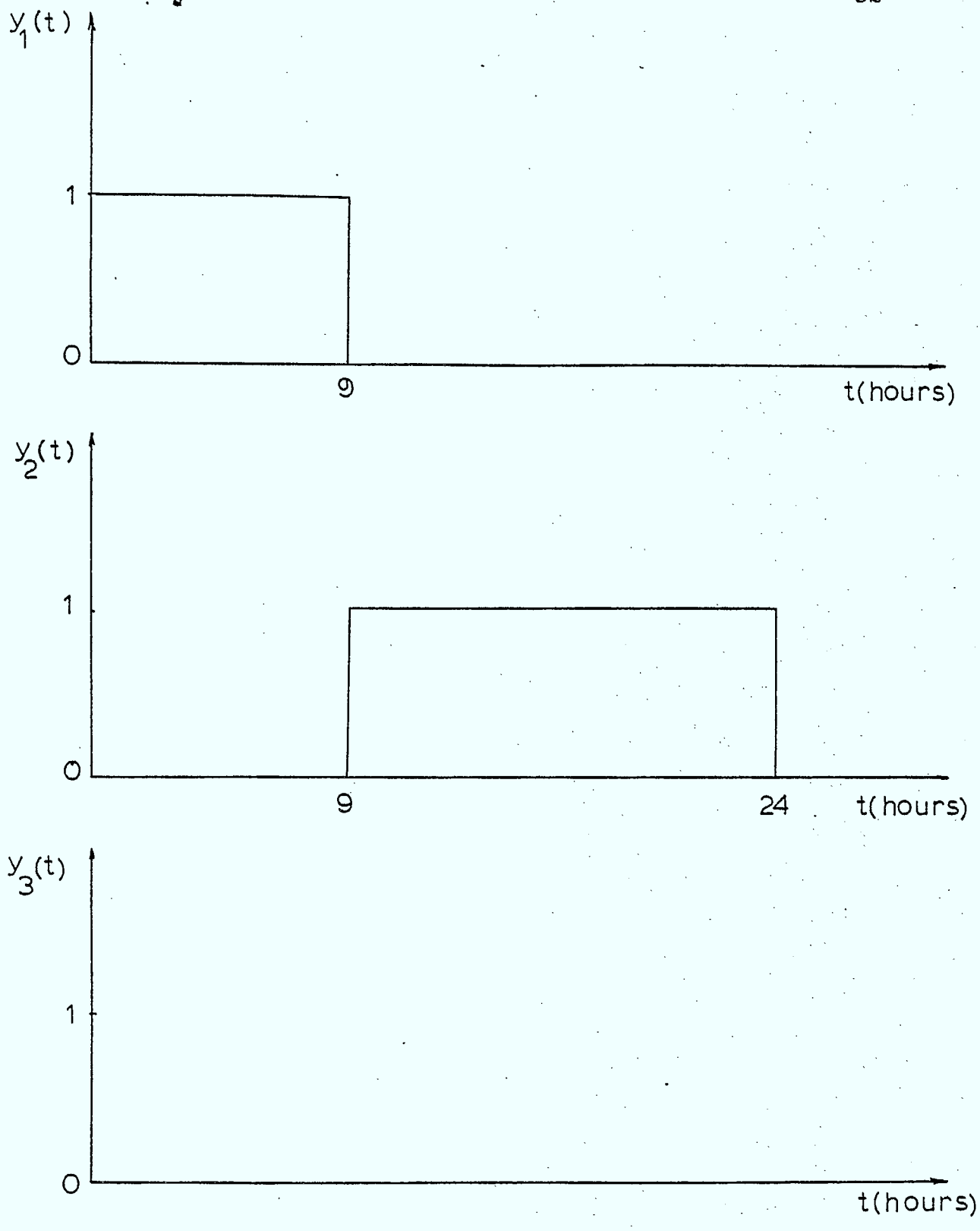


Figure 4.4.3 -Optimal file allocation for Example 4.12

By examining the results of Figure 4.4.3, we can see that the file is not stored in the memory of computer 3 at all. It remains in the memory of computer 2 during the period 9-24 hours. Note that during the period 18-24 hours only computer 3 may request the file, while the probability of a request from computer 2 is zero.

Example 4.13

In this example, we examine the case where both request rate and storage cost in a computer have high values. In Figure 4.4.4 we can see that during the period 9-15 hours the request rate of computer 2 is much higher than the request rates of the other computers. The values of the other parameters of the system are the same as in Table 4.4.3. The optimal allocation of the file is given in Figure 4.4.5 and the total cost is found to be $129.67.

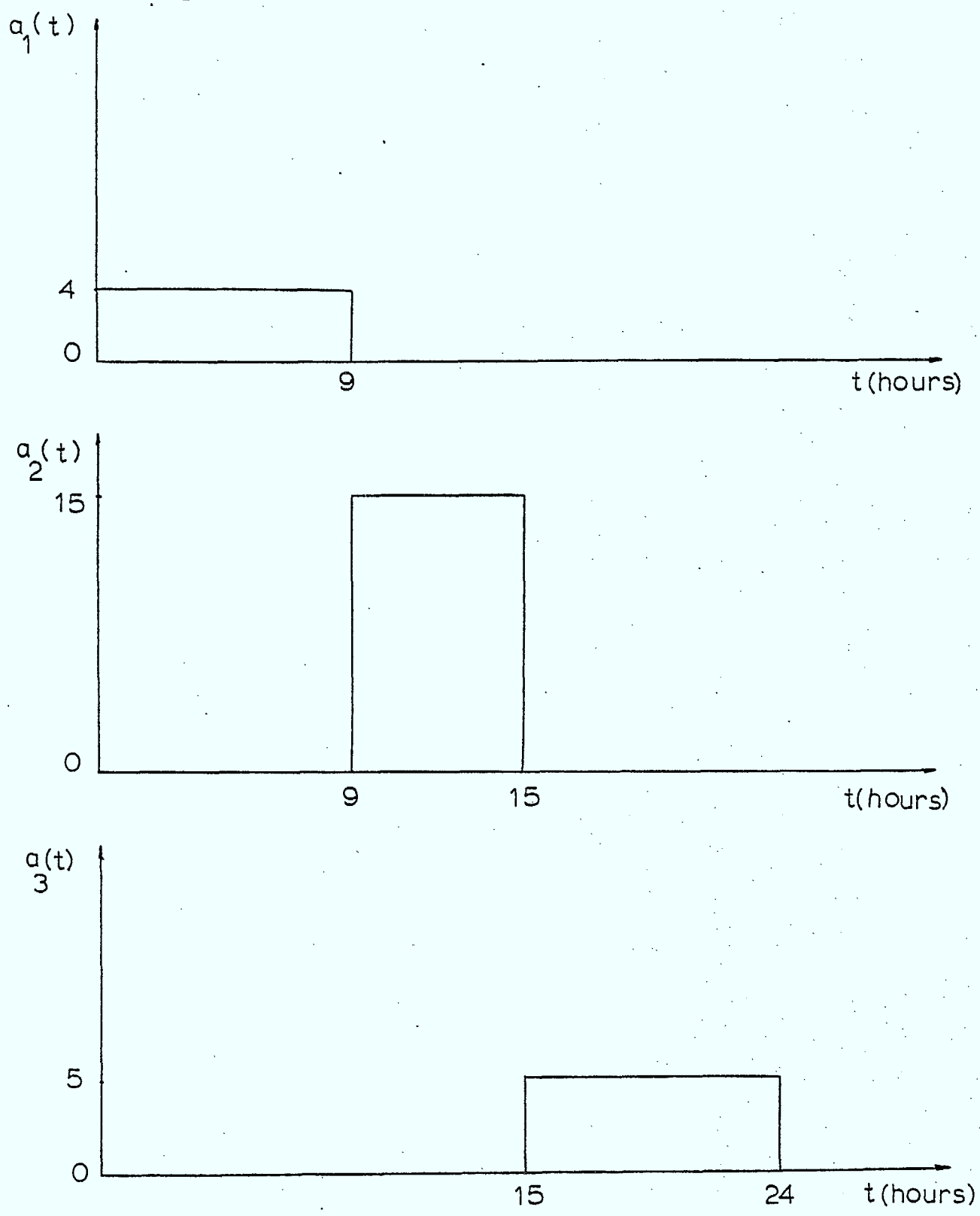


Figure 4.4.4 -Request rates for Example 4.13

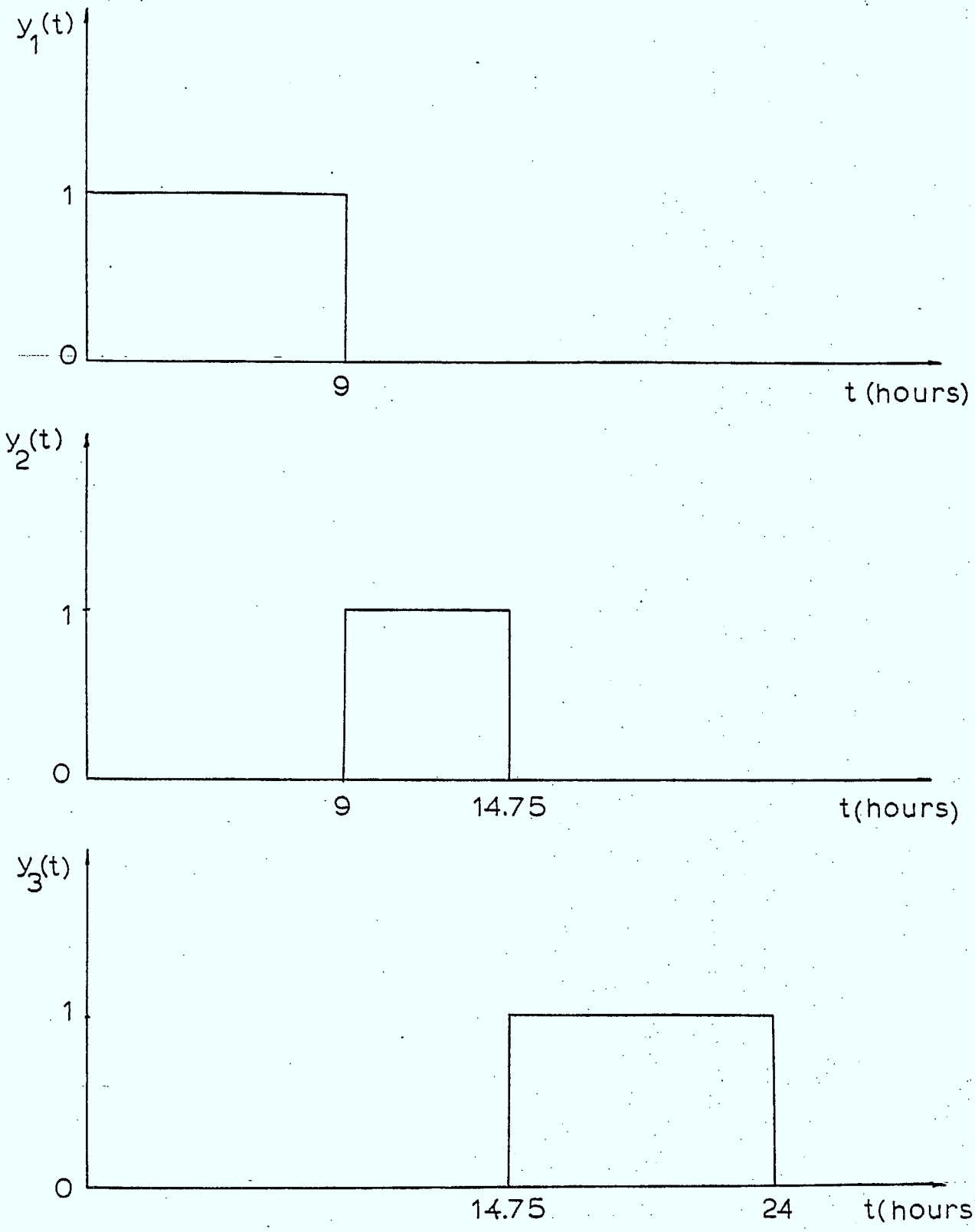


Figure 4.4.5 -Optimal file allocation for Example 4.13

By examining the results of Figure 4.4.5 we can see that they are close enough to the results of Example 4.1. This means that, if a computer requests the file with a very high rate (in comparison with the others), the central controller decides that the file has to be stored in the memory of this computer, even if its storage cost is very high (in comparison with the storage costs of the other computers).

Chapter 5
Conclusions

5.1 Summary of Results

In this report the dynamic file assignment problem in a three-computer system with a linear topology is formulated and solved as a discrete-time optimal control problem. It is the first time that such a problem is studied under the assumption that the computer network is not completely connected. A detailed theoretical analysis is given, followed by a presentation of the simulation results and a discussion of the effects of the parameters of the system on the problem solution.

5.2 Suggestions for Further Research

(1) In this report we have presented the model describing the dynamics of the file whose location is desided by a central controller. The same problem can be extended to find the optimal decentralized decisions about the location of the file. This requires application of decentralized control techniques on the particular network that we have studied.

(2) The model described in Chapter 3 can also be extended to include unknown rates of demand for each of the three computers. This requires further equations for estimating the unknown request rates,

using prior statistics available for these rates.

(3) There are also many other directions in which the results of this report have to be extended, like the investigation of the same problem in which multiple copies of the file are considered. Moreover, the effects on the results of finite storage and finite channel capacity will have to be studied.

(4) Finally, we may mention that this work could be the motivation for further research on computer networks with a star topology, which is a generalization of the linear topology of three computers studied in the present report.

Appendix A
Discrete-Time Point Processes

A discrete-time point process $\{n(t), t=1,2,...\}$ is simply a binary sequence describing the occurences of some type of events. Here $\{n(t)=1\}$ shows that such an event occurs at time t, and $\{n(t)=0\}$ shows that there is no occurence at time t. It is also assumed that no more than one event can occur at any given time. The simplest case is when $\{n(t)\}$ is a Bernoulli sequence of independent random variables with

$$Pr\Big\{n(t)=1\Big\} = 1-Pr\Big\{n(t)=0\Big\} = a(t) \tag{A.1}$$

Here one assumes also that the (possibly time-varying) parameters $a(t)$ are exactly known. The expected value of the variables $n(t)$ can be expressed as:

$$E\Big\{n(t)\Big\} = 1\cdot Pr\Big\{n(t)=1\Big\} + 0\cdot Pr\Big\{n(t)=0\Big\}$$

$$= 1\cdot a(t) + 0\cdot(1-a(t)) = a(t) \tag{A.2}$$

Because of this simple but critical relationship of equation (A.2), the quantity $a(t)$ is sometimes called the (time-varying) "rate" of occurence at time t. The role of the discrete-time point processes as well as of other stochastic processes in estimation theory, is studied in [Segall 1976a] and [Segall 1976b].

Appendix B
Backward Dynamic Programming

The purpose of this Appendix is to provide some background knowledge of the background dynamic programming technique used in Chapter 3. Suppose that we study a dynamic system described by

$$\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t)) \tag{B.1}$$

where $\mathbf{x}(t)$ is the state vector and $\mathbf{u}(t)$ the control vector. Also suppose that the objective is to minimize the cost function

$$J = \sum_{t=1}^{N} L_t(\mathbf{x}(t),\mathbf{u}(t)) + h(\mathbf{x}(N+1)) \tag{B.2}$$

Note that the function L_t is time-varying. The case where this function is time invariant is studied in [Bellman 1957], where the forward dynamic programming technique is analyzed in detail. The goal in our problem is to find the optimal controls $\mathbf{u}(t)$ that minimize (B.2) and satisfy (B.1). To solve this problem we follow this procedure : Define

$$Q_i(\mathbf{x}) = \min_{\mathbf{u}(i),\ldots,\mathbf{u}(N)} \left\{ \sum_{t=i}^{N} L_t(\mathbf{x}(t),\mathbf{u}(t)) + h(\mathbf{x}(N+1)) \right\} \tag{B.3}$$

Clearly, $Q_1(\mathbf{x}) = J_{\min}$. We also assume that $\mathbf{x} = \mathbf{x}(i)$ is given. Then, by applying (B.3) for $i = N$ we get

$$Q_N(\mathbf{x}) = \min_{\mathbf{u}(N)} \left\{ L_N(\mathbf{x},\mathbf{u}(N)) + h\,(\mathbf{f}(\mathbf{x},\mathbf{u}(N)) \right\} \tag{B.4}$$

Also (B.3) can be writte in a new form

$$Q_i(\mathbf{x}) = \min_{\mathbf{u}(i)} \left\{ L_i(\mathbf{x},\mathbf{u}(i)) + \min_{\mathbf{u}(i+1),\ldots,\mathbf{u}(N)} \left[\sum_{t=i+1}^{N} L_t(\mathbf{x}(t),\mathbf{u}(t)) + h\,(\mathbf{x}(N+1)) \right] \right\}$$

But, according to (B.3) the second term within the braces is simply the term $Q_{i+1}(\mathbf{x}(i+1))$, so the equation (B.3) may be written

$$Q_i(\mathbf{x}) = \min_{\mathbf{u}(i)} \left\{ L_i(\mathbf{x},\mathbf{u}(i)) + Q_{i+1}(\mathbf{f}(\mathbf{x},\mathbf{u}(i))) \right\} \tag{B.5}$$

where $i = N-1, N-2, \ldots, 1$. Thus, the optimal controls $\mathbf{u}(i)$ can be found by solving backwards the set of equations (B.5) and considering the equation (B.4) as the initial step. The minimum cost function $J_{\min}$ is the quantity $Q_1(\mathbf{x})$, calculated during the last step of the backward procedure.

Bibliography

Bellman R. (1957). *Dynamic Programming*. Princeton Univ. Press, Princeton NJ

Chu W. W. (1969). Optimal file allocation in a multiple computer system. *IEEE Trans. Comput.* **C-18**, 885

Chu W. W. (1973). Optimal file allocation in a computer network. *Computer Communication Networks*, N.Abramson and F.F.Kuo,Eds. Englewood Cliffs,NJ:Prentice-Hall

Frank H. (1966). Optimal location on a graph with probabilistic demands. *Oper. Res.* **vol. 14**, 409

Hakimi S. L. (1964). Optimum locations of switching centers and the absolute centers and medians of a graph. *Oper. Res.* **vol. 12**, 450

Hakimi S. L. (1965). Optimum distribution of switching centers in a communication network and some related problems. *Oper. Res.* **vol. 13**, 462

Howard R. A. (1960). *Dynamic Programming and Markov Processes*. New York, Wiley

Ros F. S. (1976). M. thesis, Massachusetts Inst. Technol., Cambridge, MA

Segall A. (1976a). Dynamic file assignment in a computer network. *IEEE Trans. Auto. Control* **AC-21**, 161

Segall A. (1976b). Stochastic processes in estimation theory. *IEEE Trans. Info. Theory* **IT-22**, 275

Segall A. (1976c). Recursive estimation from discrete-time point processes. *IEEE Trans. Info. Theory* **IT-22**, 422

Segall A., N. R. Sandell (1979). Dynamic file assignment in a computer network-part II: decentralized control. *IEEE Trans. Auto. Control* **AC-24**, 709

Tanenbaum A. S. (1981). *Computer Networks*. Englewood Cliffs NJ, Prentice-Hall

VIDYASAGAR, M.
--The dynamic file allocation...