

Video-on-Demand 서비스망의 자원 할당 문제를 위한 동적계획법

조 명 래†

요 약

B-ISDN(Broadband-Integrated Services Digital Network)의 출현으로 주문형 비디오(Video On Demand VOD) 서비스는 차세대 유망한 영상 정보 서비스로 관심의 대상이 되고 있으며, 대화성의 정도에 따라 크게 IVOD(Interactive VOD)와 NVOD(Near VOD) 서비스로 분류된다

다계층 구조 서비스망에서 VOD 서비스를 하기 위해서는 여러 자원이 필요하겠지만 본 연구에서는 비디오 서버, 프로그램, 그리고 저장장교로 두고, 프로그램 전송 비용, 프로그램 저장 비용, 그리고 비디오 서버 설치 비용을 최소화 하는 지원 최적화 문제를 두 가지 형태로 나누어 그 해결 방법을 제시하였다. 첫째는 VOD를 IVOD로만 서비스 하는 경우 서비스 운용 비용을 최소화 하는 비디오 서버의 위치 선정과 설치된 비디오 서버에 저장될 프로그램의 종류와 그 양을 결정하는 문제로, 이를 효과적으로 해결하기 위해 동적계획법을 이용하였다. 둘째는 VOD서비스를 보다 경제적으로 제공하기 위해, VOD를 IVOD와 NVOD의 혼합 서비스로 제공할 경우 서비스 운용 비용을 최소화 하는 NVOD프로그램의 종류와 그 양 그리고 그 때의 IVOD서비스를 위한 비디오 서버의 위치 선정 및 설치된 비디오 서버에 저장시킬 프로그램의 종류와 그 양을 결정하는 문제로 동적계획법을 확장하여 제시하였다

A Dynamic Programming for Solving Resource Allocation Problems in Video-on-Demand Service Networks

Myeong-Rai Cho†

ABSTRACT

It is strongly believed that Video on Demand(VOD) will become one of the most promising services in Broadband Integrated Services Digital Network(B-ISDN) for the next generation. VOD service can be classified into two types of services: Near VOD(NVOD) and Interactive VOD(IVOD). For both services, some video servers should be installed at some nodes(especially, at the root node for NVOD service) in the tree structured VOD network, so that each node with video server stores video programs and distributes stored programs to customers. We consider three kinds of costs: a program transmission cost, a program storage cost, and a video server installation cost. There exists a trade-off relationship among those three costs according to the locations of video servers and the kinds of programs stored at each video server.

Given a tree structured VOD network and the total number of programs being served in the network, the resource allocation problem in a VOD network providing both IVOD and NVOD services is to determine where to install video servers for IVOD service, which and how many programs should be stored at each video server for both IVOD and NVOD services, so as to minimize the total cost which is the sum of three costs for both IVOD and NVOD services.

In this paper we develop an efficient dynamic programming algorithm for solving the problem. We also implement the algorithm based on a service policy assumed in this paper.

† 정 희 원 신남도탑남도대학 김보룡신과 교수

논문접수: 2000년 8월 21일, 심사완료: 2000년 10월 4일

1. Introduction

The emergence of B-ISDN(Broadband-Integrated Service Digital Network) and the advance of several technologies such as ATM(Asynchronous Transfer Mode) technology, image compression/retrieval technology, and multimedia storage/transmission technology make it possible to provide customers with high bandwidth interactive services such as video on demand (VOD), home shopping, video conferencing, etc. Especially, VOD seems to be the most attractive service for the next generation. VOD services can be classified into two types: interactive VOD(IVOD) and near VOD(NVOD)[1, 4]

IVOD is a real time service that provides a customer with a requested program for the customer to control it. However, IVOD requires expensive and highly developed Video Server(VS)s and storage mediums to support the real time service, and incurs a large amount of program storage and transmission costs due to point-to-point connections on demand. Consequently, another interactive service such as NVOD is needed to optimize the network resource. NVOD service should be utilized from the economical VOD service point of view[2, 3].

NVOD distributes periodically some programs on several channels for each program so that customers can begin to watch their requested programs from scratch by allowing a reasonable amount of waiting time. Customers who do not want to wait the NVOD service can switch the requested service to IVOD service unless they clear it. NVOD service is not a real time service and does not depend on customers requests, but relatively cheaper VS and storage mediums than those for IVOD service can be used. Moreover, relatively small amount of program storage and transmission costs compared with IVOD service occurs because one channel can be allocated to several customers simultaneously.

In this paper we consider the resource allocation problem in a VOD network providing the mixed service of IVOD and NVOD(RAPINVOD), which determines where to install video servers for IVOD service(it is assumed a VS for NVOD service is installed only at the root node of the given tree structured VOD net-

work), which and how many programs should be stored at each video server for both IVOD and NVOD services by considering customers demands for each program, so as to minimize the sum of operating costs. There might be several costs related to the operating cost for the mixed service of IVOD and NVOD, but we just consider three kinds of costs for each service: a program transmission cost, a program storage cost, and a VS installation cost.

To the best of our knowledge, the problem RAPINVOD has not been completely analyzed by researchers yet. Hodge *et al.*[5, 7] and Ishihara *et al.*[6] just proposed a service policy for the mixed service of IVOD and NVOD such that some of popular programs are distributed through NVOD service since the total cost will be increased if all the programs are distributed only through IVOD service. In particular, Hodge *et al.*[5] analyzed technologies and costs required for IVOD and NVOD services. Kim *et al.*[12] proposed a dynamic programming algorithm for the resource allocation problem in a VOD network providing only IVOD service (RAPINVOD).

In this paper we propose a service policy for providing NVOD service and also develop a dynamic programming algorithm for solving RAPINVOD under the policy by extending the key idea of the dynamic programming algorithm proposed by Kim *et al.*[12]

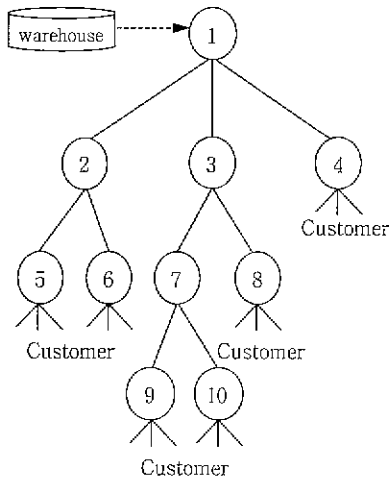
This paper is organized as follows. In Section 2, we first describe VOD network architecture and several assumptions and introduce concepts of the program vision probability and the mean service demand and also define the rate of lost service request for an NVOD program. Section 3 introduces a dynamic programming algorithm for the problem RAPINVOD. In Section 4, we propose a dynamic programming algorithm for the problem RAPINVOD by extending the algorithm given in Section 3. Section 5 concludes the paper.

2. Problem Description

2.1 VOD Network Architecture and Several Assumptions

In this section, we consider two kinds of directed and tree-structured VOD networks, one providing only

IVOD service and the other providing the mixed service of IVOD and NVOD. We assume that those networks consist of N interconnected central offices(COs) represented by nodes which are labeled in the Breadth First Search(BFS) order. It is assumed that at most one VS for IVOD service can be installed for each CO and exactly one VS for NVOD service can be installed at the root node of the network. We assume that the program warehouse containing all kinds of programs is connected to the root node of the network. The program warehouse provides some programs which are initially stored at the program storage of a video server(VS) in a CO on schedule whenever customers request those programs. We also assume that every customer is connected to exactly one of the closest leaf nodes(COs) in the network by the dedicated link so that the transmission cost from the leaf node to the customer can be ignored. Each CO corresponding to non-leaf node not only transfers IVOD programs from the CO to the immediately linked COs(i.e., its successors), but also copies NVOD programs distributed from a VS for NVOD service and multi-broadcasts those to its successors. An example of VOD network is shown in (Fig. 1)



(Fig. 1) An example of tree structured VOD network

Let $P[i, 1]$ be the set of nodes on the path from node i to the root node 1, i.e., $P[i, 1] = \{i, i_1 = PD_i, i_2 = PD_{i_1}, \dots, 1\}$ where PD_n is the predecessor of node n for

each $n = 2, 3, \dots, N$. Then it is assumed that a customer connected to a leaf node i can receive the requested IVOD program from a VS on the path $P[i, 1]$. Therefore, all of the IVOD programs requested by customers connected to the leaf node i should be stored at some VSs on the path $P[i, 1]$. We assume that the unit storage cost for every program is identical for all COs and the link capacity between two consecutive COs has no limitation

Let J be the total number of programs being served in the network. Then it is assumed that all of the programs are sorted in the decreasing order of customers preference and an IVOD program with higher preference is stored at a closer VS to customers in order to reduce the transmission cost. Moreover it is assumed that an NVOD program with high preference has the priority to be stored at a VS for NVOD service located at the root node since the more customers is served on each channel for an NVOD program.

If an NVOD program with the j -th customers preference is distributed on m_j number of channels from a VS for NVOD service located at the root node, then it is copied into all COs corresponding to successors of the root node and immediately transferred to successors of successors on m_j number of channels. This procedure is repeated until the program is copied into COs connected to customers. Then each customer can watch the requested program by choosing one of m_j channels from the CO connected to him/her[6]. Therefore, a service provider needs a standard for determining the number of channels for each NVOD program. In this paper we assume that large number of channels are allocated to NVOD programs with high customers preference so that the rate of lost service requests for NVOD programs can be reduced.

2.2 Program Vision Probability and Mean Service Demand

We assume in this paper that the demand for each program is determined by customers preference which is sorted in a decreasing order, although it varies with

several factors such as service time, service type (IVOD or NVOD service), and customers location, etc. Giovanni et al.[10] defined the *program vision probability* with the j -th customers preference (simply called it the j -th program vision probability) as follows :

$$\begin{aligned}
 P_j &= \frac{P_{j-1}}{D_{HP}}, \quad j = 2, 3, \dots, J, \\
 P_1 &= \frac{1 - (1/D_{HP})^J}{1 - (1/D_{HP})}, \quad \sum_{j=1}^J P_j = 1,
 \end{aligned}
 \tag{1}$$

where D_{HP} is the ratio between the $(j-1)$ -th and j -th program vision probabilities.

Note that $P_1 \geq P_2 \geq \dots \geq P_J$ and thus $D_{HP} \geq 1$. In this paper we also use (1) as the definition of the program vision probability. It is assumed that the same program requested by customers connected to all leaf nodes in the network has the same program vision probability.

We now define the *mean service demand* as the mean traffic volume occurred during the unit time of the busiest period. The mean traffic volume can be evaluated by the multiple of the following three values : the number of customers connected to the node n , the probability that customers will request the service during the busiest period, and the mean service time. More precisely speaking, let $\vec{T} = (V, E)$ be a directed and tree structured VOD network and $T(n) = \{q \in V \mid n \in P[q, 1]\}$ be the complete subtree of \vec{T} rooted at node n , where $V = \{1, 2, \dots, N\}$ and $E = \{(PD_i, i) \mid i = 2, 3, \dots, N\}$ are the set of nodes and the set of links (arcs), respectively. For convenience, we denote an arc (PD_i, i) as just arc i since there is point-to-point correspondence between E .

Let L_n be the set of successors of node n (i.e., $L_n = \{q \in V \mid PD_q = n\}$) and R_n be the mean service demand at node n per unit time of the busiest period. Then, if n is a leaf node (i.e., $L_n = \emptyset$), then R_n can be determined by the following value : the mean traffic volume at node $n \div$ (the unit_service time). Otherwise (i.e., if $L_n \neq \emptyset$), R_n can be obtained by $R_n = \sum_{q \in W} R_q$ where $W = \{q \in T(n) \mid L_q = \emptyset\}$.

2.3 Rate of Lost Service Requests for an NVOD Program

NVOD service distributes programs on several channels periodically. For instance, if a program with the service duration of two hours is distributed on five channels, then the program can be distributed repeatedly through NVOD service per every 24 minutes. As mentioned earlier, customers can begin to watch their requested NVOD programs from scratch by allowing reasonable amount of waiting time unless he/she cancels the request. The maximum amount of time that a customer should wait the requested NVOD program is equal to the time interval between the starts of two consecutive distributions of the same NVOD program from the VS. Therefore, if customers requested NVOD programs feel that the waiting time is too long, then they may possibly cancel their requests.

We define the *rate of lost service requests* for an NVOD program as the probability that a customer who requested the NVOD program cancels the request. To specify the rate of lost service requests for an NVOD program with the j -th vision probability, we suppose that an NVOD program with the j -th vision probability is distributed on m_j number of channels. Then, if $V(m_j)$ is the time interval between the starts of two consecutive distributions of the NVOD program with the j -th vision probability, $V(m_j)$ is also the maximum amount of time that a customer should wait the requested NVOD program with the j -th vision probability and can be obtained by

$$V(m_j) = \frac{\tau_j}{m_j}, \quad j = 1, 2, 3, \dots, J
 \tag{2}$$

where $0 < V(m_j) < \infty$ and τ_j is the service duration of the j -th program with $\tau_j > 0$.

Now, let T be a random variable of time that a customer waits for the requested NVOD program and $f(t)$ be the probability density function of T . Then the probability that a customer will wait the requested NVOD program for more than t hours is obtained by

$$P(T > t) = \int_t^\infty f(x) dx
 \tag{3}$$

Therefore, if $P_f(V(m_j))$ is the probability that a customer will wait the requested NVOD program with the j -th vision probability, $P_f(V(m_j))$ can be calculated by

$$P_f(V(m_j)) = \int_0^{V(m_j)} P(T > t) dt \quad (4)$$

Consequently, the rate of lost service requests for an NVOD program with the j -th vision probability, denoted by $\overline{P}_f(V(m_j))$, is obtained by

$$\overline{P}_f(V(m_j)) = 1 - P_f(V(m_j)) \quad (5)$$

For example, if T is exponentially distributed with parameter δ , i.e., if $f(t) = \delta \exp(-\delta t)$ with $0 < t < \infty$ and $\delta > 0$, then we have

$$P_f(V(m_j)) = \frac{m_j}{\tau_j \cdot \delta} \left(1 - \exp\left(-\frac{\tau_j \cdot \delta}{m_j}\right) \right) \quad (6)$$

Here, the parameter δ stands for the mean queueing rate that a customer will receive an NVOD service.

Since the cost of NVOD service is usually cheaper than that of IVOD service and each customer also makes a decision to wait or not to wait the requested NVOD program, we assume the following in this paper .

- (i) if a program is distributed through NVOD service, then a customer requested the program wants to receive NVOD service rather than IVOD service
- (ii) impatient customers who can not wait the requested NVOD programs receive the IVOD service in the ratio of γ , $0 \leq \gamma \leq 1$. Consequently, some customers among impatient customers who can not wait the requested NVOD programs clear their requests in the ratio of $1 - \gamma$.

3. Dynamic Programming for RAPIVOD

In this paper we consider three kinds of costs for IVOD service : a program transmission cost, a program storage cost, and a VS installation cost. Then the resource allocation problem in a VOD network providing only IVOD service(RAPIVOD) is to decide where we

should install VSs, which and how many programs should be stored at each VS, so that all the demands are satisfied with the minimum total cost. We propose a dynamic programming algorithm for solving RAPIVOD in this section

One way of finding the optimal solution for the problem RAPIVOD is to use the enumeration method which considers all the possible solutions, but it is very inefficient and almost impossible to find the optimal solution when the number of COs and programs increases, since the size of the solution space grows exponentially. Moreover, cost functions are non-linear in general and thus it is necessary to find an efficient solution technique for this kind of problem. For this purpose, we first introduce three kinds of cost functions, the cost function of the program transmission, the cost function of the program storage, and the cost function of the video server installation. Let $TC_n(k)$ be the cost function of the program transmission on arc n (PD_n, n) when k kinds of programs are stored on $T(n)$. Then the transmission cost of the $(j-k)$ number of programs on arc n should be evaluated in terms of the mean service demands for programs with the program vision probabilities from $(k+1)$ -th through J -th Therefore, $TC_n(k)$ can be expressed as follows :

$$TC_n(k) = \begin{cases} g_1 \left(C_t, D_n, \sum_{j=k+1}^J (R_n \times P_j) \right), & \text{if } n \neq 1 \\ 0, & \text{otherwise,} \end{cases} \quad (7)$$

where C_t is the unit transmission cost of an IVOD program and D_n is the distance between node n and its predecessor PD_n .

For example, if $L_n \neq \emptyset$ for $n \neq 1$ and $g_1(a, b, c)$ is defined by $(a \times b \times c)^\phi$ with $\phi_t > 0$, then $TC_n(k)$ is

expressed by $\left(C_t, D_n, \sum_{j=k+1}^J (R_n \times P_j) \right)^\phi$, where ϕ_t is

the parameter of the transmission cost. The third quantity $\sum_{j=k+1}^J (R_n \times P_j)$ in (7) represents the total amount of mean service demands on node n which is equal to the total traffic volume on arc n during the busiest

period of time when k kinds of programs are assumed to be stored on $T(n)$.

Let $SC(k, x_n)$ be the cost function of the program storage on node n when x_n kinds of programs out of k ones are stored at node n and the remaining $k - x_n$ kinds of programs are stored on $T(q)$ for all $q \in L_n$ (i.e., $k - x_n$ kinds of programs are stored at some nodes on the path $P[u, q]$ for each leaf node $u \in T(q)$ and all $q \in L_n$). Note that programs associated with the program vision probabilities from the $(k - x_n + 1)$ -th through the k -th are stored at node n because of our program storage policy assumed in this paper. Here, we assume that the unit program storage cost is the same for all programs. Let $\lceil x \rceil$ be the smallest integer larger than or equal to x . Then $SC(k, x_n)$ can be expressed as follows :

$$SC(k, x_n) = \begin{cases} g_2 \left(C_s, \sum_{j=k-x_n+1}^k \left\lceil \frac{R_n \times P_j}{h} \right\rceil \right), & \text{if } x_n \neq 0 \\ 0, & \text{otherwise,} \end{cases} \quad (8)$$

where C_s is the unit storage cost of an IVOD program and h is the number of multiple accesses for an IVOD program

For example, if $x_n \neq 0$ and $g_2(a, b)$ is defined by $(a \times b)^{\phi_s}$ with $\phi_s > 0$, then $SC(k, x_n)$ is expressed by $\left(C_s \times \sum_{j=k-x_n+1}^k \left\lceil \frac{R_n \times P_j}{h} \right\rceil \right)^{\phi_s}$, where ϕ_s is the parameter of the storage cost. The quantity $\left\lceil \frac{R_n \times P_j}{h} \right\rceil$ in (8) represents the number of programs with the j -th program vision probability stored at a VS located at node n .

If at least one program is stored at node n (i.e., if $x_n \neq 0$), then a VS should be installed in node n . Let $IC(k, x_n)$ be the cost function of the installation of a VS on node n under the same situation given for $SC(k, x_n)$. Then $IC(k, x_n)$ can be expressed as follows :

$$IC(k, x_n) = g_3(C_v, y(x_n)), \quad (9)$$

where $y(x_n) = \begin{cases} 1, & \text{if } x_n \neq 0 \\ 0, & \text{otherwise} \end{cases}$

where C_v is the installation cost of a VS for IVOD service.

For example, if the function $g_3(a, b)$ is defined by $a \times b$, then $IC(k, x_n)$ is expressed by $C_v \times y(x_n)$.

With those three cost functions, we now present an efficient dynamic programming for solving RAPIVOD. For a given node n , we assume that k kinds of programs are stored on $T(n)$ for $k = 0, 1, 2, \dots, J$. Let $f(n, k)$ be the minimum total cost related to storing k kinds of programs on $T(n)$. Suppose that we have found $f(q, k)$ for all $q \in L_n$ and $k = 0, 1, 2, \dots, J$. Then $f(n, k)$ can be determined by the following recursive formula :

$$f(n, k) = \begin{cases} \min_{0 \leq x_n \leq k} \left\{ SC(k, x_n) + IC(k, x_n) + \sum_{q \in L_n} f(q, k - x_n) \right\} \\ \quad + TC_n(k), & \text{if } L_n \neq \emptyset \\ SC(k, k) + IC(k, k) + TC_n(k), & \text{otherwise.} \end{cases}$$

It is important to notice that all the nodes in the network are labeled in BFS order and our dynamic programming incorporates a bottom-up approach which solves the restricted resource allocation problem on $T(n)$ by moving the node n in the reverse of BFS order.

We now summarize the main idea of our dynamic programming approach. We begin with the leaf node N . If k kinds of programs are stored on $T(N) = \{N\}$, then all of those programs should be stored at node N itself and the $(J - k)$ number of programs with the lower program vision probabilities than the k -th (i.e., programs with the program vision probability from $(k + 1)$ -th to the J -th) should be stored at some VSs on the path $P[PD_N, 1]$. Therefore, to find the minimum total cost $f(N, k)$ for all $k = 0, 1, 2, \dots, J$, the storage cost of storing k kinds of programs at node N , the video server installation cost at node N , and the transmission cost of the $J - k$ number of programs on arc N should be evaluated by considering the service demand for each program at node N . Consequently, $f(N, k)$ can be obtained by the sum of those cost, $SC(k, k) + IC(k, k) + TC_N(k)$, for all $k = 0, 1, 2, \dots, J$. Now, we consider the complete subtree $T(N - 1)$ of \vec{T} rooted at node $N - 1$. If

the node $N-1$ is a leaf node, then $T(N-1) = \{N-1\}$ and thus $f(N-1, k)$ can be obtained by the same argument for $f(N, k)$, which is equal to $SC(k, k) + IC(k, k) + TC_{N-1}(k)$, for all $k = 0, 1, 2, \dots, J$. Otherwise(i.e., if $L_{N-1} \neq \emptyset$), $T(N-1)$ consists of $\{N-1, N\}$ where $PD_N = N-1$. Suppose that k kinds of programs are stored on $T(N-1)$ and x_{N-1} kinds of programs out of those programs are stored at node $N-1$. Then, to find $f(N-1, k)$, it is enough to evaluate the storage cost of storing x_{N-1} kinds of programs at node $N-1$, the video server installation cost at node $N-1$, and the transmission cost of the $(J-k)$ number of programs on arc $N-1$ for each $x_{N-1} = 0, 1, \dots, k$, since $k - x_{N-1}$ kinds of programs are stored at node N and we have already found the minimum total cost $f(N, k - x_{N-1})$. Therefore, $f(N-1, k)$ can be obtained by

$$\min_{0 \leq x_{N-1} \leq k} \{SC(k, x_{N-1}) + IC(k, x_{N-1}) + f(N, k - x_{N-1}) + TC_{N-1}(k)\}$$

We continue the above procedure by visiting nodes in the reverse of BFS order until we meet the root node 1 of \vec{T} , and finally find the optimal value $f(1, J)$ of RAPIVOD.

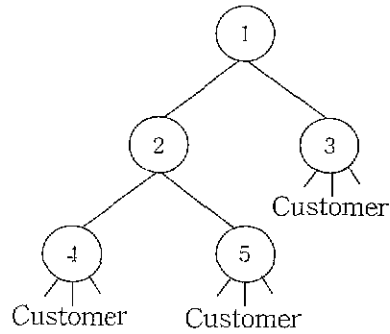
To find the optimal solution x_n^* for $n = 1, 2, \dots, N$, we first define the following value for each $n = 1, 2, \dots, N$ and $k = 0, 1, 2, \dots, J$:

$$x(n, k) = \begin{cases} \operatorname{argmin}_{0 \leq x_n \leq k} \{SC(k, x_n) + IC(k, x_n) + \sum_{q=1}^{L_n} f(q, k - x_n)\}, & \text{if } L_n \neq \emptyset \\ k, & \text{otherwise} \end{cases}$$

Then the optimal solution can be obtained by $x_n^* = x(n, J - w)$ in the BFS order for all $n = 2, 3, \dots, N$, where $w = \sum_{i \in PD_{n-1}, 1} x_i^*$ and $x_1^* = x(1, J)$. Note that the optimal solution holds the information about the video server location and the kinds of programs stored in the video server. In fact, if $x_n^* \neq 0$ for some n , then a video server should be installed at node n . Moreover, programs with the program vision probability from $(J -$

$w - x_n^* + 1)$ -th through $(J - w)$ -th should be stored at node n , since it is assumed that the program with the lower program vision probability is stored at the further node to the customer. For example, if $V = \{1, 2\}$, $J = 7$, $x_1^* = 3$, and $x_2^* = 4$, then programs with 7-th, 6-th, and 5-th program vision probabilities and programs with 4-th, 3-rd, 2-nd, and 1-st program vision probabilities should be stored at node 1 and node 2, respectively

We now use an example given in (Fig. 2) to show how our algorithm works in detail. We assume that the total number of programs being served in the example is 5. It is assumed that the cost functions of the program transmission and the program storage are nonlinear(i.e., $\phi_t \neq 1$ and $\phi_s \neq 1$) and the unit transmission cost of an IVOD program is more expensive than the unit storage cost of the program. It is also assumed that the length of each arc is equal to one, but it can be easily extended to the case of having the different arc length. All values used for the example are shown in <Table 1>.



(Fig. 2) An example of VOD service network

<Table 1> Values corresponding to parameters

Parameter	Value	Parameter	Value
J	5	h	10
D_{HP}	1.06	C_t	2
D_n	1	C_s	1
R_3	300	C_v	100
R_1	200	ϕ_t	0.5
R_5	100	ϕ_s	2

We begin with node 5. Since it is a leaf node, i.e.,

$L_5 = \emptyset$, the objective function values $f(5, k)$ and the values for decision variables $x(5, k)$ can be obtained by $SC(k, k) + IC(k, k) + TC_5(k)$ and k , respectively for all $k = 0, 1, 2, 3, 4, 5$. The detailed procedure is shown in <Table 2>.

We move into node 4. Since $T(4) = \{4\}$, we can find out the corresponding values $f(4, k)$ and $x(4, k)$ in the same way of finding $f(5, k)$ and $x(5, k)$ for all $k = 0, 1, 2, 3, 4, 5$. The detailed procedure is also shown in <Table 3>.

We then move into node 3 and <Table 4> shows the detailed procedure of finding the corresponding values $f(3, k)$ and $x(3, k)$ for all $k = 0, 1, 2, 3, 4, 5$.

We now move into node 2. Since $L_2 \neq \emptyset$ and $T(2) = \{2, 4, 5\}$, for all $k = 0, 1, 2, 3, 4, 5$, the objective function values $f(2, k)$ can be obtained by

$$\min_{0 \leq x_2 \leq k} \{SC(k, x_2) + IC(k, x_2) + f(4, k - x_2) + f(5, k - x_2)\} + TC_2(k)$$

and the values for decision variables $x(2, k)$ can be obtained by

$$\operatorname{argmin}_{0 \leq x_2 \leq k} \{SC(k, x_2) + IC(k, x_2) + f(4, k - x_2) + f(5, k - x_2)\}$$

<Table 5> shows the detailed procedure

Finally we move into the root node 1 and the optimal value $f(1, 5)$ can be obtained by

$$\min_{0 \leq x_1 \leq 5} \{SC(5, x_1) + IC(5, x_1) + f(2, 5 - x_1) + f(3, 5 - x_1)\} + TC_1(5)$$

and the values for decision variables $x(1, 5)$ can be

<Table 2> $f(5, k)$ and $x(5, k)$

k	x_5	$TC_5(k) + SC(k, x_5) + IC(k, x_5)$					$f(5, k)$	$x(5, k)$	
		0	1	2	3	4			5
0		14+0+0					14	0	
1			12+9+100				121	1	
2				11+36+100			147	2	
3					9+64-100		173	3	
4						6-100+100	206	4	
5							0+144+100	244	5

<Table 3> $f(4, k)$ and $x(4, k)$

k	x_4	$TC_4(k) + SC(k, x_4) + IC(k, x_4)$					$f(4, k)$	$x(4, k)$	
		0	1	2	3	4			5
0		20-0+0					20	0	
1			18+15+100				143	1	
2				15+100+100			215	2	
3					12+196+100		308	3	
4						8+324-100	432	4	
5							0+184+100	584	5

<Table 4> $f(3, k)$ and $x(3, k)$

k	x_3	$TC_3(k) + SC(k, x_3) + IC(k, x_3)$					$f(3, k)$	$x(3, k)$	
		0	1	2	3	4			5
0		24+0+0					24	0	
1			22+49+100				171	1	
2				18+196+100			314	2	
3					15+400+100		515	3	
4						10+676+100	786	4	
5							0+1024+100	1124	5

<Table 5> $f(2, k)$ and $x(2, k)$

k	x_2	$TC_2(k) + SC(k, x_2) + IC(k, x_2) + f(4, k-x_2) + f(5, k-x_2)$					$f(2, k)$	$x(3, k)$
		0	1	2	3	4		
0	24+0+0 +20+14						58	0
1	22+0+0 +143+121	22+49+100 +20+14					205	1
2	18+0+0 -215+147	18+49+100 +143-121	18+196-100 +20+14				348	2
3	15+0+0 -308+173	15+36+100 +215-147	15+169+100 -143+121	15+400+100 +20+14			496	0
4	10+0+0 -432+206	10+36+100 +308-173	10+144+100 +215+147	10+361+100 +143+121	10+676+100 +20+14		616	2
5	0+0+0 584+244	0+36+100 +432-206	0+144+100 -308+173	0+321+100 +215+147	0+625+100 +143+121	0+1024+100 +20+14	725	2

<Table 6> $f(1, 5)$ and $x(1, 5)$

k	x_1	$TC_1(5) + SC(5, x_1) + IC(5, x_1) + f(2, 5-x_1) + f(3, 5-x_1)$					$f(1, 5)$	$x(1, 5)$
		0	1	2	3	4		
5	0+0+0 +725+1124	0+122+100 +616+786	0+529+100 -496+515	0+1225+100 +348+314	0+230+100 +205+171	0+3844+100 +58+24	1624	1

obtained by

$$\operatorname{argmin}_{0 \leq x_1 \leq 5} \{SC(5, x_1) + IC(5, x_1) + f(2, 5 - x_1) + f(3, 5 - x_1)\}.$$

As shown in <Table 6>, the optimal value of 1624 is obtained by storing programs with the 5-th program vision probability at the root node 1.

To find the optimal solution x_n^* for $n = 1, 2, \dots, 5$, we just trace the values $x(n, k)$ given in the above tables in the BFS order. Since $x_n^* = x(1, 5) = 1$ from <Table 6>, four kinds of programs with 4-th, 3-rd, 2-nd, and 1-st program vision probabilities should be stored on $T(2) = \{2, 4, 5\}$ and $T(3) = \{3\}$, respectively. Then, by using $x_n^* = x(n, 5 - x_1^*)$ for $n = 2, 3$, we can obtain $x_2^* = x(2, 4) = 2$ and $x_3^* = x(3, 4) = 4$ from <Table 5> and <Table 4>, respectively. It means that programs with 4-th and 3-rd program vision probabilities should be stored at node 2 and programs with the program vision probabilities from 4-th through 1-st should be stored at node 3. Now we move into node 4. Since $P[PD_4, 1] = \{2, 1\}$ and $x_4^* = x(4, 5 - x_2^* - x_3^*)$, we have $x_4^* = x(4, 2) = 2$ from <Table 3>. It means that programs with

2-nd and 1-st program vision probabilities should be stored at node 4. Finally, we move into node 5 and obtain $x_5^* = x(5, 2) = 2$ from <Table 2> in the same argument for finding x_4^* . <Table 7> summarizes the optimal solutions for the example in (Fig. 2). Note that the number of programs with the j -th program vision probabilities stored at a VS located at node n can be obtained by $\lfloor \frac{R_j \times P_j}{h} \rfloor$ or all $n, j = 1, 2, \dots, 5$.

<Table 7> Optimal Solutions for the example

CO(n)	State(k)	Decision Variable (x_n^*)	Kind of stored program Amount(of program)
1	5	1	5 11
2	4	2	4 6, 3 6
3	4	4	4 6, 3 6, 2 7, 1 7
4	2	2	2 5, 1 5
5	2	2	2 3, 1 3

4. Extension to the Mixed Service of IVOD and NVOD

In this paper we also consider three kinds of costs for both IVOD and NVOD services: a program transmission cost, a program storage cost, and a VS instal-

lation cost. Then the storage allocation problem in a VOD network providing the mixed service of IVOD and NVOD(RAPINVOD) is to decide where we should install VSs for IVOD service, which and how many programs should be stored at each VS for both IVOD and NVOD services, so that all the demands are satisfied with the minimum total cost. Note that a VS for NVOD service is assumed to be installed only at the root node of the given network.

We propose a dynamic programming for solving RAPINVOD in this section. For the mixed service of IVOD and NVOD, we first need to find out an efficient rule of determining the number of channels for each NVOD program. Note that we have assumed that impatient customers who can not wait the NVOD service receive IVOD service in the ratio of γ , $0 \leq \gamma \leq 1$. For this case, we may consider several possible rules of determining the number of channels for each NVOD program, but we just propose a rule which takes the number of channels for an NVOD program such that an expected number of customers who cancel their NVOD service requests does not exceed L where L is a fixed number. In fact, let m_j be the number of channels for an NVOD program with the j -th program vision probability. Then is determined by the minimum number of channel satisfying $\overline{P}_j(V(m_j)) \times (R_1 \times P_j) \leq L$, where $\overline{P}_j(V(m_j))$ is given in (5) and $R_1 \times P_j$ stands for the expected service demand for an NVOD program with the j -th program vision probability, since a VS for NVOD service can be installed only at node 1. Note that the mean service demand for an IVOD program with the j -th program vision probability is $\overline{p}_j(V(m_j)) \times (R_1 \times P_j) \times \gamma$. The procedure of finding m_j can be described as follows.

Procedure Find_ m_j

- Step 1. (Initialization) $\overline{m}_j \leftarrow 0$;
- Step 2. $\overline{m}_j \leftarrow \overline{m}_j + 1$,
- Step 3. $\overline{P}_j(V(\overline{m}_j)) \leftarrow 1 - P_j(V(\overline{m}_j))$;
- Step 4. If $\overline{P}_j(V(\overline{m}_j)) \times (R_1 \times P_j) > L$, then go to Step 2.

Step 5. $m_j \leftarrow \overline{m}_j$, stop ;

Once we obtain the number of channels, m_j , for all $j = 1, 2, \dots, J$, we are able to decide which and how many programs should be served through IVOD and NVOD, respectively, so as to minimize the sum of operating costs of IVOD service and NVOD service. Before we formulate the problem RAPINVOD, we first introduce three kinds of cost for NVOD service, the transmission cost, the storage cost, and the video server installation cost.

Let NTC_s be the transmission cost for s kinds of NVOD programs stored at a VS on node 1 to all leaf nodes connected to customers by using m_j number of channels. For convenience, we set $m_0 = 0$. Then, NTC_s can be expressed as follows :

$$NTC_s = \sum_{\kappa=2}^N \left(\sum_{j=0}^s (nct \times D_{\kappa} \times m_j) \right)^{\phi_s} \quad (10)$$

where nct is the transmission cost for an NVOD program per unit distance and ϕ_s is the parameter for the transmission cost with $\phi_s \geq 0$.

Let NSC_s be the storage cost for s kinds of NVOD programs at a VS on node 1. Then NSC_s can be expressed as follows :

$$NSC_s = \left(\sum_{j=0}^s \left(ncs \times \left\lfloor \frac{m_j}{H} \right\rfloor + 1 \right) \right)^{\phi_s} \quad (11)$$

where ncs is the unit storage cost for an NVOD program and ϕ_s is the parameter for the storage cost with $\phi_s \geq 0$.

The quantity $\left\lfloor \frac{m_j}{H} \right\rfloor$ in (11) represents the amount of programs with the j -th program vision probability stored at a VS on node 1 for NVOD service.

Let NFC_s be the installation cost of a VS for NVOD service on node 1 when s kinds of programs are served by NVOD. Then, since a VS for NVOD service should be installed at node 1 if at least one program is served by NVOD, NFC_s can be expressed as follows :

$$NFC_s = ncv \times y_s, \tag{12}$$

where ncv is the installation cost of a VS for NVOD service and y_s is defined by

$$y_s = \begin{cases} 1, & \text{if } s \neq 0 \\ 0, & \text{otherwise} \end{cases}$$

With the above three costs related to NVOD service, the problem RAPINVOD can be formulated as follows :

$$\min_{0 \leq s \leq J} \{ NTC_s + NSC_s + NFC_s + f_s(. J) \}, \tag{13}$$

where $f_s(1, J)$ is the minimum total cost for providing IVOD programs with the program vision probabilities rearranged by considering the rate of lost service requests for s kinds of NVOD programs and can be obtained by the dynamic programming algorithm given in Section 3.

We now summarize the main idea of our dynamic programming procedure for solving RAPINVOD. Initially, the number of channels for each NVOD program is obtained by applying the procedure 'Find $_m$ '. We first evaluate the IVOD operating cost corresponding to providing only IVOD service by applying the algorithm given in Section 3 and then begin with allocating programs to the VS for NVOD service in the decreasing order of program vision probabilities and find out the total cost which is the sum of the NVOD operating cost and the IVOD operating cost. In finding the IVOD operating cost, all the programs for IVOD service should be rearranged in decreasing order of program vision probabilities because all the customers who cancel the requested NVOD programs receive IVOD service in the ratio of γ and thus the program vision probabilities of programs for IVOD service that are also allocated for NVOD service should be changed. Once we find out the maximum number, s^* , of kinds of programs which should be stored at the VS for NVOD service so that the total cost can be minimized, the locations of VSs for IVOD service and the kind and number of IVOD programs stored at each VS can be found simultane-

ously when $f_s(1, J)$ is evaluated by the algorithm given in Section 3. We now describe our dynamic programming procedure for solving (13) as follows :

Procedure Solve_RAPINVOD

- Step 1. (Initialization) Find $_m$, for all $J = 1, 2, \dots, J$;
 $TCOST \leftarrow \infty$; $s^* \leftarrow 0$; $s \leftarrow 0$;
- Step 2 $NCOST \leftarrow NTC_s + NSC_s + NFC_s$;
- Step 3. If $s = 0$, then go to Step 6 ;
- Step 4. $P_s \leftarrow P_s \times \bar{P}(V(m_s)) \times \gamma$,
- Step 5. Rearrange all the programs by following the updated program vision probabilities in decreasing order ;
- Step 6. Evaluate $f_s(1, J)$;
- Step 7. If, $NCOST + f_s(1, J) < TCOST$, then $TCOST \leftarrow NCOST + f_s(1, J)$, $s^* \leftarrow s$;
- Step 8. If $s < J$, then $s \leftarrow s + 1$ and go to Step 2 ; Otherwise, stop ;

Note that the returned values $TCOST$ and s^* from the procedure 'Solve_RAPINVOD' are the optimal value and the optimal number of kinds of programs for NVOD service, respectively. Moreover, the total amount of programs stored at a VS on node 1 for NVOD service can be obtained by $\sum_{i=0}^{s^*} \lceil \frac{m_i}{H} \rceil$.

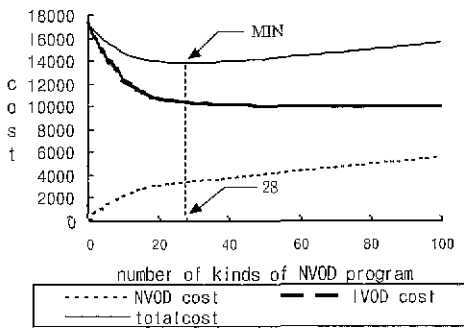
We now use the example given in (Fig. 1) to show the procedure of our algorithm. All the assumptions for IVOD service are assumed to be the same as those used in Section 3, except that the cost functions of the program transmission and the program storage for IVOD service as well as NVOD service are assumed to be linear(i.e., $\phi_t, \phi_l, \phi_s, \phi_v = 1$) here. We assume that the program transmission cost, the program storage cost, and the video server installation cost for IVOD service is more expensive than those for NVOD service. We also assume that the unit transmission cost of an NVOD program is more expensive than the unit storage cost of the program. Moreover, it is assumed that all the customers who can not wait the NVOD service

receive IVOD service(i.e., $\gamma = 1$) and the mean queuing time that a customer will wait the requested NVOD program is 20 minutes(i.e., $\delta = \frac{1}{20} = 0.05$). It is also assumed that $L = 710$ and the service duration for all programs is identically equal to 120 minutes(i.e., $\tau_j = 120$ for all $j = 1, 2, \dots, 5$). All values used for the example are summarized in <Table 8>.

<Table 8>. Values corresponding to parameter

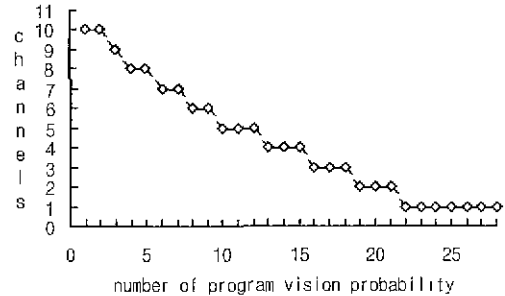
Parameter	Value	Parameter	Value
J	100	γ	1
D_{IMP}	1.06	δ	0.05
D_n	1	$\tau_j, j=1,2,\dots,J$	120
c_s / ncs	3/2	R_1	12,000
c_v / ncv	4/3	R_5	7,000
c_o / nco	200/150	R_6	9,000
h/H	10/10	R_8	9,000
ϕ_r / ϕ_s	1/1	R_9	6,000
ϕ_s / ϕ_s	1/1	R_{10}	7,000

We now apply the procedure 'Solve_RAPINVOD' to find out the optimal number, s^* , of kinds of programs for NVOD service which gives the minimum total cost. (Fig. 3) shows the change of the total cost with respect to the number of kinds of NVOD programs when $L = 710$.



(Fig. 3) The change of total cost with respect to number of kinds of NVOD program when $L = 710$

Consequently s^* is 28. (Fig 4) shows the number of channels for each s^* kind of programs for NVOD service



(Fig. 4) The number of channels for each s^* kind of programs for NVOD service

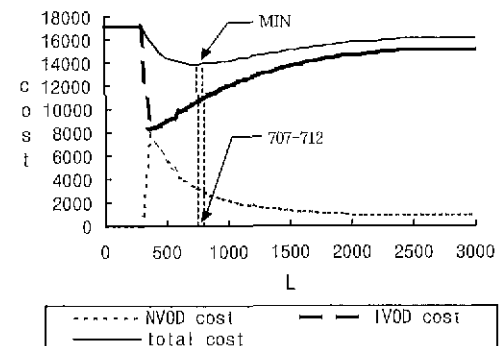
The optimal solutions including the locations of VSs for IVOD service and the kinds and number of IVOD programs stored at each VS are shown in <Table 9>

<Table 9> Optimal solutions related to IVOD service for example

CO(m)	1	2	3	4	5	6	7	8	9	10
Amount	0	0	0	711	440	546	0	546	386	440
Kind	(0)	(0)	(0)	(1-100)	(1-100)	(1-100)	(0)	(1-100)	(1-100)	(1-100)

Finally, (Fig. 5) shows the change of the total cost for the example with respect to the change of L .

It can be seen that the optimal value is obtained when $707 \leq L \leq 712$. This implies that our algorithm not only solves RAPINVOD under the predetermined service policy, but also provides the critical information to service providers when they have to prepare their service policy for providing the mixed service of IVOD and NVOD beforehand



(Fig. 5) The change of the total cost with respect to the change of L

5. Conclusions

In this paper we have introduced a dynamic programming algorithm for solving the resource allocation problem providing only IVOD service(RAPIVOD). Then we have proposed the procedure of finding the number of channels for each NVOD program under the assumption that the mean number of customers who cancel the requested NVOD service is given and also proposed an efficient dynamic programming algorithm for solving the resource allocation problem in a VOD network providing the mixed service of IVOD and NVOD (RAPINVOD) by extending the key idea of the dynamic programming algorithm for solving RAPIVOD

It is expected that our algorithm given in this paper can be applied to several optimization problems which arise in resource allocation problems in networks providing various types of multimedia services.

References

[1] Petit, G.H., Delodre, D., and Verbiest, W., "Bandwidth Resource Optimization in Video-On-Demand Network Architectures," IEEE GLOBECOM '94, pp.91-97, 1994.

[2] Gelman, A. D. and Halfin, S., "Analysis of Resource Sharing in information Providing Services," IEEE GLOBECOM'90, pp.312-316, 1990.

[3] Sincoskie, W. D., "Video on Demand : Is it Feasible?," IEEE GLOBECOM'90, pp.201-205, 1990.

[4] IGI Consulting, "Video Dialtone & Video on Demand," VDT 2000, Vol.3, 1995

[5] Hodge, W. and Milligan, C., "True Video on Demand VS. Near Video on Demand, Statistical Modeling, Cost, & Performance Trade-offs." NCTA TECHNICAL PAPERS, pp.157-172, 1994.

[6] Ishihara, T., Tanaka, J., Nakajima, I., Okuda, M,

and Yamashita, H., "Modeling and Evaluation of Broadband Access Networks for Multimedia Services." IEEE GLOBECOM'94, pp.117-125, 1994.

[7] Hodge, W., Mabon, S., and Jone T Powers, Jr., "Video On Demand : Architecture, Systems, and Applications," SMPTE Journal, pp.791-803, 1993.

[8] Sinkoskie, W. D., "System Architecture for a Large Scale Video on Demand Service," Computer Networks and ISDN Systems, Vol.22, pp.155-162, 1991.

[9] Hessenmuller, H., "The use of large CATV networks as an infrastructure for interactive video services." Australian Telecommunication Networks & Applications Conference, pp.69-74, 1994.

[10] Giovanni, L.De., Langellotti, A M., Pautucci, L.M., and Petrini, L., "Dimensioning of Hierarchical Storage for Video on Demand Services." IEEE ICC'94, pp.1739-1743, 1994.

[11] Schaffa, F. and Nussbaumer, J-P., "On Bandwidth and Storage Tradeoffs in Multimedia Distribution Networks," IEEE INFOCOM'95, pp.1020-1026, 1995.

[12] Kim, Y. G , Cho, M. R., and Kim, J. Y., "Storage Allocation in Multi-level VOD Network Using Dynamic Programming," IE Interfaces, Vol.9, No.3, pp.202-213, 1996.



조명래

e-mail : mrcho@changhung.ac.kr

1990년 전남대학교 산업공학과 (학사)

1993년 전남대학교 대학원 산업공학과(공학석사)

1997년 전남대학교 대학원 산업공학과(공학박사)

1999년~현재 전남도립 남도대학 정보통신과 전임강사, 전자계산소장

관심분야 : 통신망 설계 및 운영, 조합최적화, Genetic Algorithm, Tabu Search 의 응용