Optimal Video Placement Scheme for Batching VOD Services

Wallace K. S. Tang, Eric W. M. Wong, Sammy Chan, and K.-T. Ko

***Abstract*—Advances in broadband technology are generating an increasing demand for video-on-demand (VOD) applications. In this paper, an optimal video placement scheme is proposed for a batching VOD system with multiple servers. Given a specified requirement of the blocking probability, an optimal batching interval is derived and the corresponding file placement is obtained by hybrid genetic algorithm. It is demonstrated that the specified requirement on blocking probability is satisfied, while both batching interval and server capacity usage are minimized simultaneously.**

***Index Terms*—Batching, blocking probability, genetic algorithm, video-on-demand system.**

I. Introduction

WITH the advanced technologies recently developed in the areas of high-speed networks and multimedia, video-on-demand service (VOD) is considered as the emerging trend in home entertainment, as well as in education, banking, home shopping, and interactive games [7], [24]. Such multimedia experience is expected to further blossom when broadband customer access networks become more popular.

Many VOD services have been proposed in the past few years. In general, they can be classified into three major types:

1) Unicast Scheme: Each video request is served by a video stream. Since the stream is dedicated to a single user, interactive VCR functions can be easily implemented. However, the solution does not scale well and hence the number of customers that can be served simultaneously is limited.

2) Multicast Scheme: A number of requests for the same video are grouped together and served by a video stream. This approach can reduce the bandwidth loading of the system and improve the blocking probability [5]. Examples of this scheme are:

> *Batching* [9]: video request is delayed for a period of time so that more requests for the same video are collected. The batch of requests is then served by a multicast video stream. The major drawback of this scheme is that the customers have to wait for a batching interval until the video is started to play. Hence, it may increase the customer dissatisfaction if the waiting interval is too long.
>
> *Patching* [18]: video request is firstly served by a unicast stream and then joined back to a multicast stream. This approach not only enjoys the saving of bandwidth as batching but also introduces zero startup delay. However, it requires a more complicated control system and is unfavorable for high request bursts. A similar scheme was also proposed in [13].

3) Broadcast Scheme: The video is broadcast on a dedicated channel with a pre-defined schedule. This approach can support an unlimited number of requests for popular video content with a constant amount of bandwidth. However, the bandwidth is wasted if the popularity of the video is low. In addition, customers have to wait until the scheduled time is reached. To reduce this delay, some improved systems have been suggested such as pyramid [31], permutation-based pyramid [1] and, skyscraper [17]. However, the operations of those systems are quite complicated at the receiver ends. In particular, the client is responsible for tuning the appropriate channel in order to download the video content.

Recently, the start-up delay or/and the bandwidth requirement for multicast/broadcast VoD schemes have been further improved. In [6], two techniques were proposed to improve the multicast and broadcast services in order to reduce the bandwidth requirement with little buffering and low delay. To provide interactivity in the multicast VoD scheme, a split and merge protocol was suggested in [23]. In [27], a multicast delivery scheme was developed to support full VCR functionality and true interactive VoD services while the number of streams required for interactive customers is minimized.

On the other hand, different combinations of services are also suggested to achieve cost-performance tradeoffs. Lee [22] combined unicast and broadcast services while Poon [28] combined unicast, multicast, and broadcast services together.

To realize any kinds of these VOD schemes, the video content system should have a huge amount of storage capacity and sufficient bandwidth (or I/O streams). For example, a 1.5-hour movie encoded at 1.5 Mbit/s (Mpeg-1) will require 1 Gbyte storage capacity. Therefore, a video server with a hundred on-line movies requires a 100 Gbyte storage capacity. Moreover, if all the movies are to be played concurrently, the I/O rate of the storage medium must be at least up to 150 Mbit/s. As a result, a multi-server system is usually adopted for a reasonable size VOD system in order to offer the necessary storage capacity as well as the I/O access rate.

Given a content system, it is always a challenge to allocate the thousand of videos with the specified quality of services (QoS). The video placement exercise is considered as a multiobjective and constrained problem which is currently handled by an experienced operator, meaning that a sub-optimal and personnel dependent solution is usually obtained. The complexity of such a problem exponentially increases with the numbers of videos

Manuscript received March 26, 2003; revised October 10, 2003. This work was supported by the City University of Hong Kong under the Grant 7 001 125.

The authors are with the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong (e-mail: kstang@ee.cityu.edu.hk).

Digital Object Identifier 10.1109/TBC.2003.822983

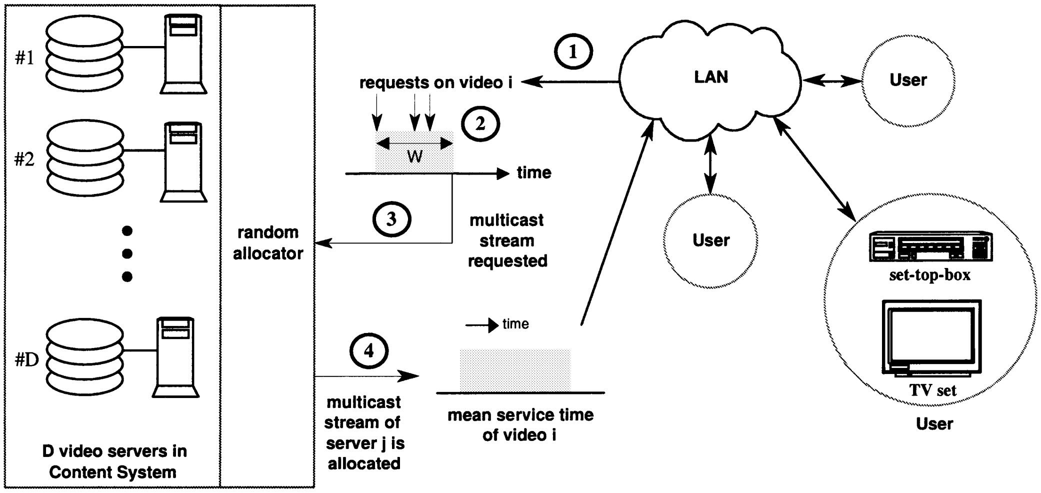
#1
#2
#D
D video servers in Content System
random allocator
1
requests on video i
2
W
time
3
multicast stream requested
4
time
multicast stream of server j is allocated
mean service time of video i
LAN
User
User
set-top-box
TV set
User

Fig. 1. Basic configuration of the Batching VOD system.

and servers in the system. The level of difficulty thereby escalates, and this unsatisfactory situation will give rise to dissatisfaction of customers.

In this paper, a hybrid genetic approach is designed to handle the video placement problem for a batching VOD service in a multi-server content system. Batching VOD is considered as an effective scheme in terms of resource utilization and the system control. With the proposed placement algorithm implemented, it is demonstrated that a guaranteed system blocking probability at the expense of minimized waiting time can be easily achieved. The proposed approach can also be extended to other systems with minor modifications.

The hybrid genetic approach is firstly proposed in [30] for a true VOD system in order to place the video contents on a set of servers with minimal blocking probability. In this paper, we have improved the algorithm to minimize the waiting interval of the batching service, and obtain the corresponding video placement scheme satisfying a specified blocking probability.

The organization of this paper is as follows: The operation of a batching VOD system is briefly introduced in the next section. The minimum blocking probability of such a system is then derived and the corresponding batching interval is determined. In Section III, the placement problem is re-formulated as a modified bin-packing problem which can be effectively solved by a proposed heuristic method. The overall video placement scheme based on hybrid genetic algorithm is then explained in detail in Section IV. In Section V, the performance of the proposed scheme is illustrated. Finally, conclusion remarks are drawn in Section VI.

II. Operation of a Batching VOD System

Fig. 1 shows a centralized content system [2] with multiple video servers for a batching VOD service. In order to provide a large variety of program contents, thousands of videos, encoded in MPEG format, are stored on one or multiple storage servers. Interleaving amongst servers is not suggested because of its adverse effect on reliability with a single server failure [14].

The videos can be obtained on demand by a large group of geographically distributed customers. When the first request arrives, a batching window will be started for a time interval W. Requests for the same video arrived within the window will be grouped together and a multicast stream is assigned randomly from one of the servers which contains the particular video [5]. It is assumed that the multicast stream can support infinite (or sufficiently large) number of users in the batch.

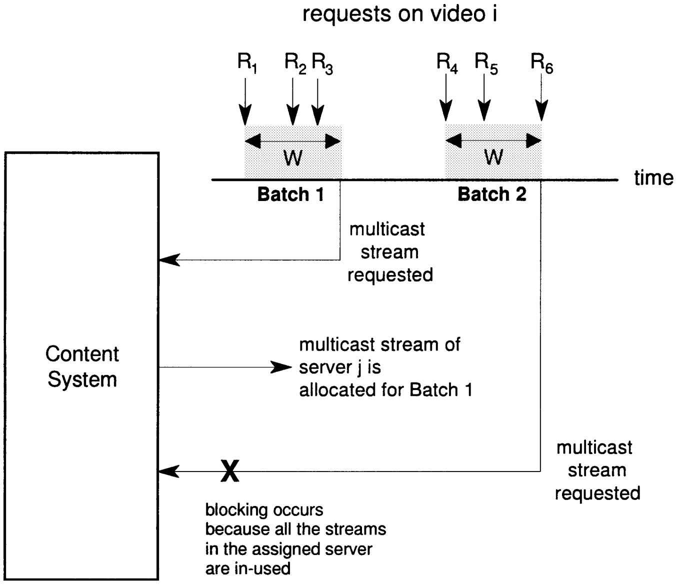
requests on video i
R1 R2 R3
R4 R5 R6
W
W
time
Batch 1
Batch 2
multicast stream requested
Content System
multicast stream of server j is allocated for Batch 1
multicast stream requested
X
blocking occurs because all the streams in the assigned server are in-used

Fig. 2. Batch scheme in the batching VOD system.

The number of concurrent accesses to a server is limited by the number of supportable multicast streams. If all the multicast streams of the allocated server are in use, blocking occurs as depicted in Fig. 2 for Batch #2 and the requests R_4, R_5, and R_6 are then rejected.

In practice, customers may renege or leave if they have waited for too long before the video playback. However, the reneging time or the exact length of time that the customers will wait before reneging, is hard to determine [29]. Obviously, the longer a client has waited, the greater the probability of reneging. In addition, the reneging time may also depend upon the channel scheduling policy. For example, the channel scheduler can attempt to influence reneging behavior through prior negotiation, like granting a maximum delay. In the studied system, it is assumed that the batching time is acceptable and customers are willing to wait for at most one batching interval without reneging.

A. Blocking Probability

Let p_i be the popularity of video i and the user requests are modeled as a Poisson arrival rate λ, the effective request rate of video i with batching can be computed as

$$\lambda_i^B = \frac{\lambda p_i}{1 + \lambda p_i W} \tag{1}$$

where W is the batching interval and $(1 + \lambda p_i W)$ is the average number of requests in a batch.

The effective request rate for the system can be expressed as

$$\begin{aligned} \lambda_e &= \sum_{i=1}^{M} \lambda_i^B \\ &= \sum_{i=1}^{M} \frac{\lambda p_i}{1 + \lambda p_i W} \end{aligned} \tag{2}$$

where M is the total number of videos available in the system.

The probability of video i being requested for a multicast stream allocation is

$$\alpha_i = \frac{\lambda_i^B}{\lambda_e} \tag{3}$$

and the system mean service time is

$$\frac{1}{\mu_e} = \sum_{i=1}^{M} \frac{\alpha_i}{\mu_i} = \frac{1}{\lambda_e} \sum_{i=1}^{M} \frac{\lambda_i^B}{\mu_i} \tag{4}$$

where $1/\mu_i$ is the mean service time of video i.

The effective traffic, A_e, coming into the system can be computed as

$$A_e = \frac{\lambda_e}{\mu_e} = \sum_{i=1}^{M} \frac{\lambda p_i}{(1 + \lambda p_i W)\mu_i} \tag{5}$$

Assuming that the interarrival time of the batched requests is exponentially distributed[1], an $M/G/n/n$ queueing system model can be applied for each video server j where n is the number of streams supportable by the server j. Assuming that q_j portion of the effective traffic A_e is allocated to server j, the blocking probability of server j having L_j multicast streams can be computed using Erlang B Formula [3]:

$$B_{q_j} = \frac{(A_e q_j)^{L_j}/L_j!}{\sum_{i=0}^{L_j} (A_e q_j)^i/i!} \tag{6}$$

where $q_j \geq 0$.

The blocking probability of the overall system is then derived [15] by

$$B = \sum_{j=1}^{D} q_j B_{q_j} = \sum_{j=1}^{D} q_j \frac{(A_e q_j)^{L_j}/L_j!}{\sum_{i=0}^{L_j} (A_e q_j)^i/i!} \tag{7}$$

where $\sum_{j=1}^{D} q_j = 1$ and D is the total number of video servers in the content system.

B. Optimal Traffic Load Sharing

Referring to (7) and using the method of Lagrange multipliers, we can define

$$G = \sum_{j=1}^{D} q_j \frac{(A_e q_j)^{L_j}/L_j!}{\sum_{i=0}^{L_j} (A_e q_j)^i/i!} + K\left(1 - \sum_{j=1}^{D} q_j\right) \tag{8}$$

where K is a constant.

Differentiate G with respect to q_j, and set the result to zero for minimization, we have

$$\frac{\partial G}{\partial q_j} = -K + \frac{\partial}{\partial q_j} q_j \frac{(A_e q_j)^{L_j}/L_j!}{\sum_{i=0}^{L_j} (A_e q_j)^i/i!} = 0 \tag{9}$$

Hence, the condition for minimization is to have $q^{\text{opt}} = [q_1^{\text{opt}}, q_2^{\text{opt}}, \ldots, q_D^{\text{opt}}]$ such that

$$\frac{\frac{\left(A_e q_j^{\text{opt}}\right)^{L_j}}{L_j!} \sum_{i=0}^{L_j} \frac{\left(A_e q_j^{\text{opt}}\right)^i (L_j+1-i)}{i!}}{\left(\sum_{i=0}^{L_j} \frac{\left(A_e q_j^{\text{opt}}\right)^i}{i!}\right)^2} = K \quad \text{or}$$

$$B_{q_j^{\text{opt}}} \cdot \left[L_j + 1 - A_e q_j^{\text{opt}} \cdot \left(1 - B_{q_j^{\text{opt}}}\right)\right] = K \tag{10}$$

with $B_{q_j^{\text{opt}}} = ((A_e q_j^{\text{opt}})^{L_j}/L_j!)/(\sum_{i=0}^{L_j} (A_e q_j^{\text{opt}})^i/i!); j = 1, 2, \ldots, D$.

Defining

$$\varphi(q_j, L_j) = B_{q_j} \cdot \left[L_j + 1 - A_e q_j \cdot \left(1 - B_{q_j}]\right)\right], \tag{11}$$

it can be proven that $\varphi(q_j, L_j)$ is monotonically increasing with q_j. Hence, a unique solution for the load sharing vector q^{opt} can be found by the binary searching method, based on the flow diagram designed in Fig. 3. The small positive constant δ is used to govern the accuracy of the solution q_j^{opt}.

C. Tradeoff Between Batching Interval and Blocking Probability

Referring to (5), a larger interval W will have a lower effective traffic density, and hence the lower blocking probability as given in (7). Fig. 4 depicts the typical relationship between batching intervals and the blocking probabilities. However, the increase of batching interval will introduce dissatisfaction to customers because of the longer waiting time.

For comparison purpose, the blocking probabilities of a simulated system with different W are also depicted in Fig. 4. It can be observed that the simulated blocking probabilities are always less than the computed values, and they are close when W is small.

[1] Under this assumption, the computed blocking probability provides an upper bound on the minimum blocking probability for a near VOD service with batching scheme. It is because the coefficient of variation [21] for the interarrival time of the batched requests is less than the one with exponential distribution (see Appendix I), implying that the actual distribution is more deterministic.

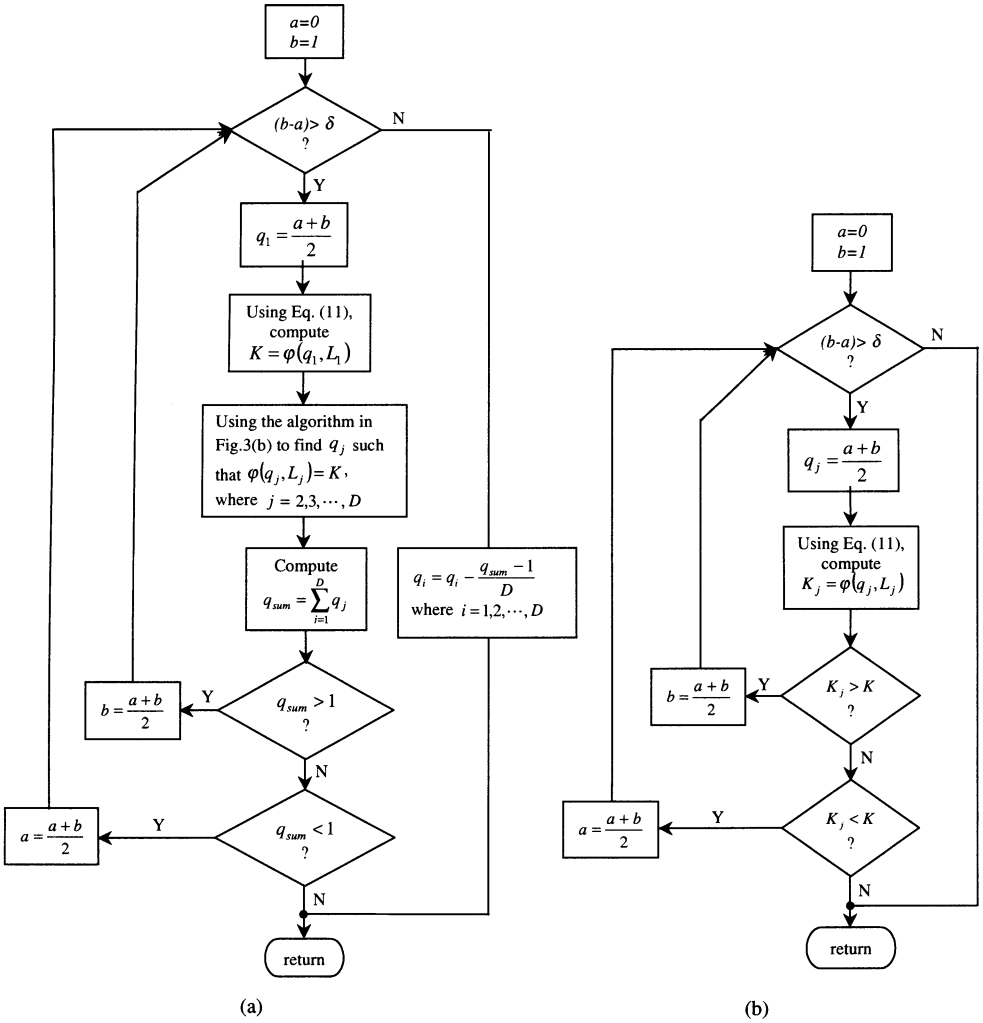
a=0
b=1
(b-a)> δ ?
N
Y
$q_1=\frac{a+b}{2}$
Using Eq. (11), compute $K=\varphi(q_1,L_1)$
Using the algorithm in Fig.3(b) to find q_j such that $\varphi(q_j,L_j)=K$, where $j=2,3,\cdots,D$
Compute $q_{sum}=\sum_{i=1}^{D}q_j$
$q_i=q_i-\frac{q_{sum}-1}{D}$ where $i=1,2,\cdots,D$
$b=\frac{a+b}{2}$
$q_{sum}>1$?
$a=\frac{a+b}{2}$
$q_{sum}<1$?
return
(a)
a=0
b=1
(b-a)> δ ?
$q_j=\frac{a+b}{2}$
Using Eq. (11), compute $K_j=\varphi(q_j,L_j)$
$b=\frac{a+b}{2}$
$K_j>K$?
$a=\frac{a+b}{2}$
$K_j<K$?
return
(b)

Fig. 3. (a) Flow diagram to find the optimal set $q^{\rm opt}$; (b) Flow diagram to find q_j such that $\varphi(q_j, L_j) = K$.

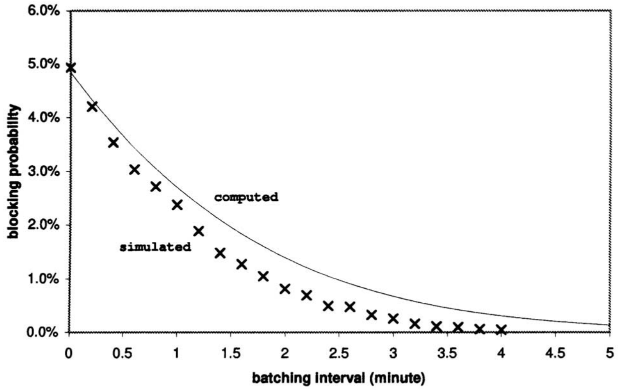

Fig. 4. Tradeoff between batching interval and blocking probability.

III. Heuristic Placement Method

If it is assumed that the set of copies of each video is known, it is possible to design an effective heuristic placement scheme so that the minimum blocking probability is approximated.

A. *Formulation of Problem*

Assuming that a random allocation scheme is employed amongst the servers with the same video stored, the probability of accessing each copy of video i can be computed as α_i/n_i where n_i is the number of copies of video i and α_i is the probability that video i requested in a batch as given in (3).

If retry is omitted in the system, the probability in accessing a server j can be computed as

$$P_{d_j} = \sum_{i\in d_j} \frac{\alpha_i}{n_i} \tag{12}$$

where $i \in d_j$ means that video i is stored in server j.

The mean service time $1/\mu_j$ and the arrival rate λ_j of server j can then be derived by

$$\lambda_j = \lambda_e P_{d_j}$$
$$\frac{1}{\mu_j} = \frac{1}{P_{d_j}} \sum_{i\in d_j} \frac{\alpha_i}{n_i\mu_i}$$

Hence, given a set of the number of copies of the videos, $\{n_1, n_2, \ldots, n_M\}$, minimum blocking probability is achieved when

$$\lambda_e \sum_{i\in d_j} \frac{\alpha_i}{n_i\mu_i} = \frac{\lambda_j}{\mu_j} = A_j = A_e q_j^{\rm opt} \quad \text{or}$$
$$\sum_{i\in d_j} \frac{\alpha_i}{n_i\mu_i} = \frac{q_j^{\rm opt}}{\mu_e} \quad \text{for } j = 1, 2, \ldots D \tag{13}$$

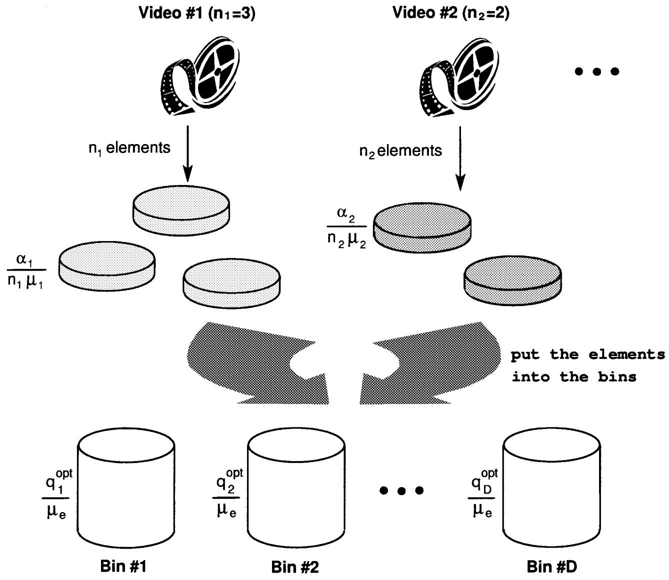
Video #1 (n1=3)
Video #2 (n2=2)
n1 elements
n2 elements
$\frac{\alpha_2}{n_2\mu_2}$
$\frac{\alpha_1}{n_1\mu_1}$
put the elements
into the bins
$\frac{q_1^{opt}}{\mu_e}$
$\frac{q_2^{opt}}{\mu_e}$
$\frac{q_D^{opt}}{\mu_e}$
Bin #1
Bin #2
Bin #D

Fig. 5. Modified bin packing problem in batching VOD system.

where $1/\mu_e$ is given in (4) and $q_j^{\rm opt}$ is the optimal load sharing of server j obtained in Section II.B.

The placement problem is hence formulated as putting the elements of size $\alpha_i/n_i\mu_i$ into the bins with space $q_j^{\rm opt}/\mu_e\ \forall j = 1, 2, \ldots, D$, as shown in Fig. 5.

B. HLF Placement Algorithm

The newly formulated problem is similar to the classical one-dimensional bin packing, for which many solutions have been suggested [8], [10], [19], [20]. The original bin-packing problem is to find the minimum number of fixed-size bins so that a collection of elements is packed without overflow. In our problem, however, the number of bins D is fixed (which is equal to the number of video servers) and the ideal space of each bin $q_j^{\rm opt}/\mu_e$ is determined based on (10). Therefore, our modified bin packing problem is to allocate the elements $\alpha_i/n_i\mu_i$ to D bins so that the bins are "nearly" full or "just" overflow. The ideal case is that all the bins are just "full" which means that (13) is satisfied.

In [30], a simple heuristic packing strategy, known as Highest-Load-First (HLF) scheme, has been suggested for this problem:

For each element to be put into the bin, the bin with maximum available space is selected providing that the capacity constraint is fulfilled and the copy of the same video has not been put into this bin before.

With HLF, the element should be arranged in a descending order, meaning that the largest size will be placed first. The descending approach is more effective by considering the discretization effect of the element's size. In general, the number of elements is relatively large as thousands of videos are to be put into the content system. Hence, the size of the elements, and also the discretization effect, introduced by the low popular videos will be very small.

Given a set of the number of copies of the videos, $\{n_1, n_2, \ldots, n_M\}$, the overall algorithm can be summarized as follows:

```
1) Compute the $1/\mu_e$ using (4) with batching interval $W$;
2) Obtain the value of $q_j^{\rm opt}$ based on the binary searching technique in Fig. 3, and determine the bin
     space of server $j$ which is equal to $q_j^{\rm opt}/\mu_e\ \forall j = 1, 2, \ldots, D$;
3) For each video $i$, it introduces $n_i$ elements with size $\alpha_i/(n_i\mu_i)$;
4) Sort the elements in descending order according to the size;
5) While (not finished)
   {
     • Select the element according to the sorted order
     • Choose the bin $j$ with maximum available space providing that the capacity of the server
       is not exceeded and the same video has not been put onto it before. If no suitable bin
       $j$ can be assigned, a constraint violation flag is issued and stop
     • Allocate the corresponding video $i$ into $j$ and reduce the available space of the bin $j$ by
       $\alpha_i/(n_i\mu_i)$
   {
```

IV. Hybrid Genetic Approach for Video Placement

In order to set up the content system, the overall placement scheme should identify the number of copies of each video and their corresponding locations for storage so that

1) the total capacity usage of the servers is minimized, and
2) the batching interval is minimized;

with the constraints that

- the capacity of each server must not be exceeded, and
- the specified blocking probability is to be fulfilled.

Fig. 6 shows the operational flow diagram of the proposed hybrid genetic approach. Given a specified blocking probability for the batching VOD service, the optimal batching interval W can be found by simple iterative approach (see Fig. 6). However, the computed blocking probability derived in Section II.B is based on the situation of optimal load sharing. It should be stressed that it may not be achieveable due to constraints of storage capacity.

Hence, a genetic algorithm is adopted to find the optimal placement solution for a given W such that a minimum blocking probability is to be met. If no feasible solution is found, W will be incremented and the GA loop will restart again. From the simulations, it is demonstrated that only a few iterations are needed due to the effectiveness of the proposed algorithm.

A. Genetic Algorithm

Genetic algorithms (GAs) [16], [25], [26] have been demonstrated as a powerful optimization tool for multiobjective problems [12], [25]. It is inspired by the mechanism of natural selection where stronger individuals would likely be the winners in a competing environment. The flow of the GA cycle can be referred to Fig. 6.

The potential solution of the problem, known as a chromosome, is structured in an integer string $I = \{n_1, n_2, \ldots n_M\}$ where M is the number of videos. The gene, n_i, represents the number of copies of video i. To ensure that the size of the element is smaller than the bin size, the lower bound of the copy

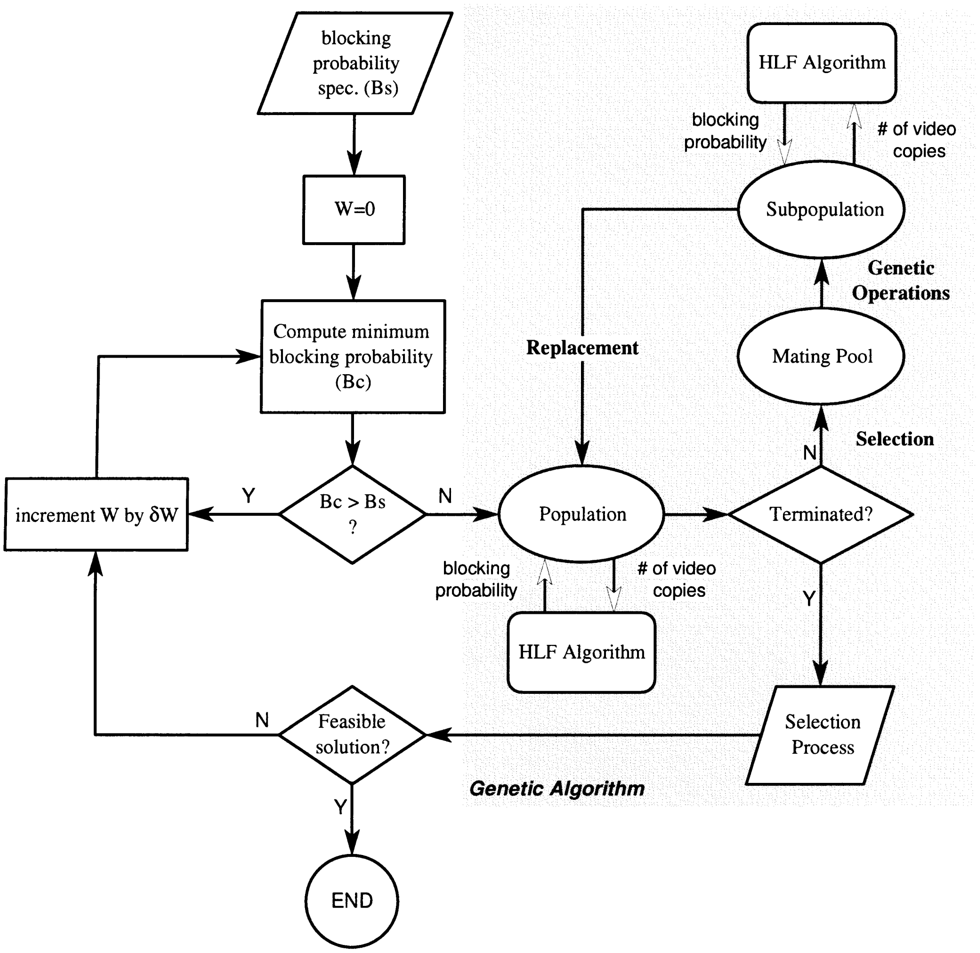

Fig. 6. Operational flow of the proposed hybrid genetic approach.

of video $i(n_i^{\mathrm{lb}})$ is the smallest positive integer satisfying the following condition:

$$\frac{p_i}{n_i^{\mathrm{lb}}\mu_i} \leq \max_j \frac{q_j^{\mathrm{opt}}}{\mu_e} \tag{14}$$

The upper bound of the number of copies (n_i^{ub}) is D which is the total number of servers (or bins) in the system.

Initially, a population of chromosomes is randomly generated. The total capacity usage of each chromosome can be found out based on the file size of each video and the corresponding number of copies. The blocking probability is obtained by applying the HLF algorithm on each chromosome. A fitness value is then assigned to each chromosome according to their rank in the population pool based on their objective values, i.e., blocking probability and capacity usage.

In order to evolve, parents are selected using a fitness proportionate selection scheme. The genes of the parents are then mixed and recombined for the production of offspring by two major genetic operations: crossover and mutation.

Multi-point crossover is adopted in our system. An example of four point crossover depicted in Fig. 7 where the crossover points are randomly selected. The portions of the two parental chromosomes are then exchanged to form their offspring.

As in mutation, the aim is to introduce genetic variation into the chromosome. Each gene of the chromosome n_i is randomly altered within the searching domain $[n_i^{\mathrm{lb}}, n_i^{\mathrm{ub}}]$ with a small probability.

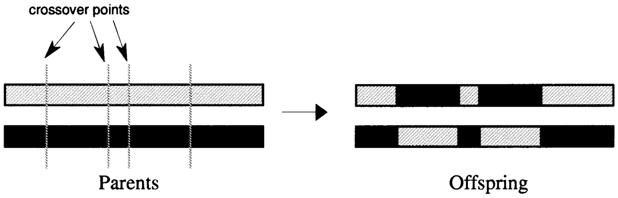

Fig. 7. Four-point crossover.

From this process of evolution (manipulation of genes), it is expected that the "better" chromosome will create a larger number of offspring, and thus has a higher chance of surviving in the subsequent generations, emulating the survival-of-the-fittest mechanism in nature.

The cycle of evolution is repeated until a desired termination criterion is reached. The criterion can be set by the number of generations, or the amount of variations of individuals between different generations, or a pre-defined value of fitness.

1) Fitness Value: The fitness value is used to reflect the "goodness" of the chromosome for the problem. In this placement problem, the solution is feasible only if the specified blocking probability (B_s) is satisfied. Using the multi-objective pareto ranking approach [11], let f_1 and f_2 be the blocking

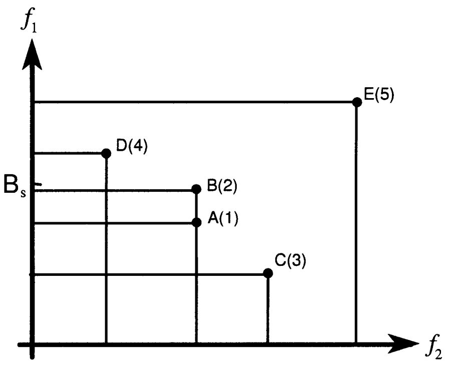

Fig. 8. Pareto-based fitness with goal information.

TABLE I
SYSTEM CONFIGURATION

Video Server	no. of multicast streams supported	capacity
No. 1-15	50	120 GByte
No. 16-20	30	70 GByte

probability and the total capacity usage, respectively, chromosome I is dominated by chromosome I' if

$$\begin{aligned}
&[\text{C1}]: \quad f_1(I) > B_s \quad \text{and} \quad f_1(I) > f_1(I')\\
&[\text{C2}]: \quad f_1(I) = f_1(I') > B_s \quad \text{and} \quad f_2(I) > f_2(I')\\
&[\text{C3}]: \quad f_1(I), f_1(I') \le B_s \quad \text{and} \quad f_2(I) > f_2(I')\\
&[\text{C4}]: \quad f_1(I') < f_1(I) \le B_s \quad \text{and} \quad f_2(I) = f_2(I')
\end{aligned}$$

The fitness of chromosome I can be determined by its rank in the population computed as

$$\text{rank}(I) = 1 + p \tag{15}$$

if chromosome I is dominated by other p chromosomes in the population.

Fig. 8 shows an example of 5 chromosomes in a population. Chromosome A is ranked as 1 and considered as the best (non-dominated) for minimizing f_1 and f_2. Chromosome B is ranked as 2 because it is dominated by chromosome A due to the condition [C4] while C is ranked as 3 because it is dominated by chromosomes A and B due to the condition [C3]. Chromosome D is ranked as 4 due to the condition [C1] since its blocking probability is larger than B_s and that of chromosomes A, B, and C.

2) Constraint Handling: If the server capacity constraint is violated when HLF is performed, a penalty value is assigned to the objectives so as to reflect the condition of the low performers.

3) Selection of Solutions: The fittest chromosome obtained in the final population is considered as a solution if the specified blocking probability is satisfied. Otherwise, the batching interval W is incremented by δW and a new GA cycle will be initiated.

V. EXPERIMENTAL RESULTS

The configuration of the content system is listed as follows:

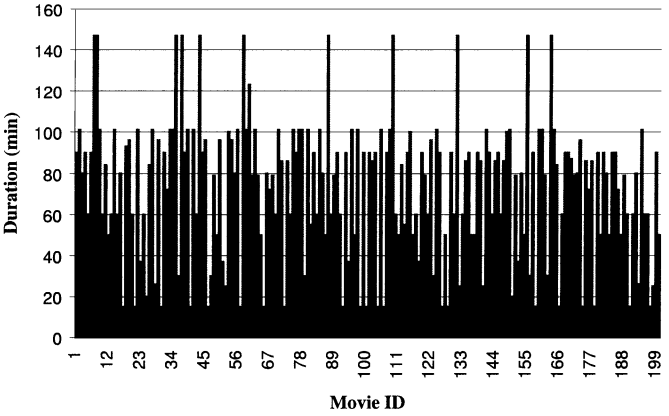

Fig. 9. Duration of videos with their ranks in popularity.

The popularities of the videos are assumed to be governed by Zipf-like distribution [4] as given below:

$$p_i = \frac{c}{i^\zeta} \quad i = 1, 2, \ldots, M \tag{16}$$

where $c = (\sum_{i=1}^{M}(1/i^\zeta))^{-1}$ and ζ is a constant describing the popularity distribution of the videos.

A. Test 1: Multiobjective Optimization

In the content system, it is assumed that there are 200 videos, randomly selected from several categories, forming a list with scale similar to some local VOD services. Their popularities are assigned based on their ID and Zipf's law with $\zeta = 0.271$ is applied [2]. The duration of videos are depicted in Fig. 9.

Assuming that the arrival rate $\lambda = 10$ and the desired blocking probability of the system is less than 1%, then, the minimum batching interval $W = 1.96$ minutes with the blocking probability equals to 0.9996%.

Fig. 10 shows the blocking probability and the total capacity usage of the best chromosome against the generation. The blocking probability and the capacity usage of the final solution are 0.999 962% and 405.334 GByte, respectively, with a batching interval of 1.96 minutes.

B. Test 2: Duration and Popularity

Since the relationship of the popularity and the duration may vary from case to case, three different situations are studied:

- Case 1: the videos with shorter duration have higher popularity
- Case 2: the videos with longer duration have higher popularity
- Case 3: the ranks of the popularity are randomly assigned (same as Test 1)

All the conditions in Test 1 are kept, and the best solutions on three different cases are tabulated as Table II. It should be noticed that only a single copy is needed for each video in both Cases 1 and 2.

C. Test 3: Popularity Distribution

The distribution of the video popularities implies different customer perference. If ζ is small, the popularity is more uniform while requests are concentrated onto the few videos if ζ is large, as illustrated in Fig. 11.

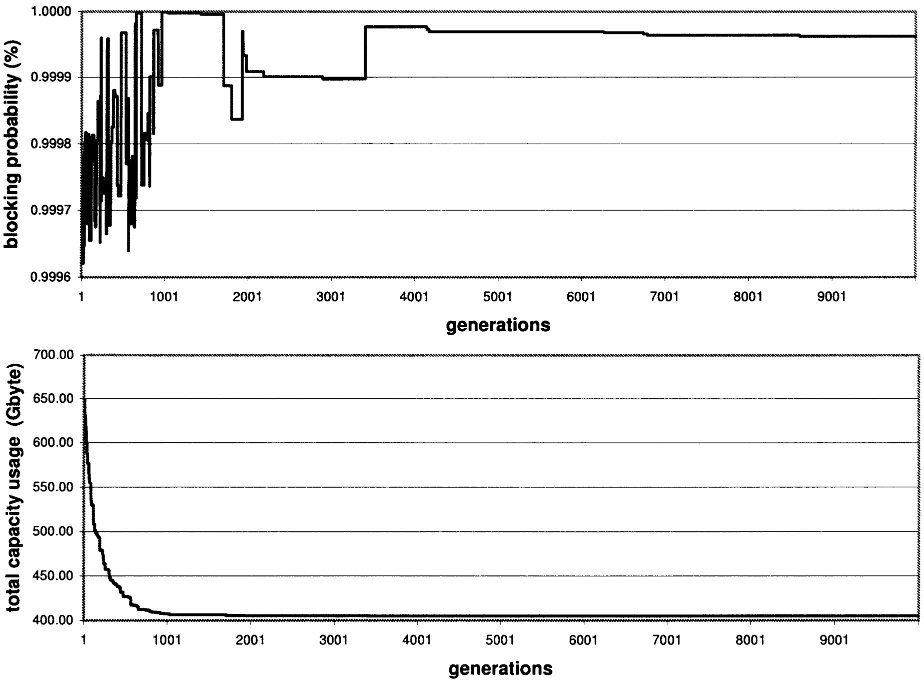

Fig. 10. Blocking probability and the capacity usage of the best chromosome.

TABLE II
SOLUTION OBTAINED IN TEST 2

	batching interval	blocking probability	capacity usage
Case 1	2.09 min	0.995068%	403.92 GByte
Case 2	1.95 min	0.996397%	403.92 GByte
Case 3	1.96 min	0.999962%	405.334 GByte

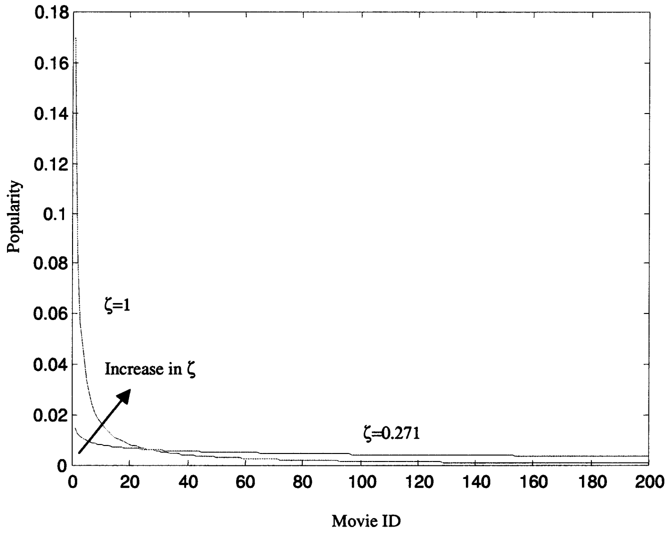

Fig. 11. Popularity distributions with different ζ.

Considering three different cases:

- Case 1: uniform distribution, $\zeta = 0$
- Case 2: $\zeta = 0.271$ (same as Test 1)
- Case 3: $\zeta = 1.0$ which means the distribution is concentrated on a few videos.

Assuming that 1% of blocking probability is required, the solutions obtained are tabulated as Table III.

TABLE III
SOLUTION OBTAINED IN TEST 3

	batching interval	blocking probability	capacity usage
Case 1	2.17 min	0.998183%	403.92 GByte
Case 2	1.96 min	0.999962%	405.334 GByte
Case 3	0.28 min	0.973950%	413.195 GByte

TABLE IV
OPTIMAL PLACEMENT WITH SINGLE COPY ONLY

	batching interval	blocking probability	capacity usage
Case 1	2.17 min	0.998183%	403.92 GByte
Case 2	1.96 min	1.000652%	403.92 GByte
Case 3	0.28 min	9.658357%	403.92 GByte

TABLE V
OPTIMAL PLACEMENT WITH TWO COPIES

	batching interval	blocking probability	capacity usage
Case 1	2.17 min	0.995442%	807.84 GByte
Case 2	1.96 min	1.00026%	807.84 GByte
Case 3	0.28 min	1.784181%	807.84 GByte

Table IV shows the best obtainable blocking probability if only one copy is assumed for each video. Comparing Tables III and IV, it can be found that a small increase in the capacity usage can cause a significant improvement by the proposed method, as demonstrated in Case 3.

Table V tabulates the best solution if exactly two copies for each video are assumed. It shows that the design specification can only be achieved in Case 1. Together with Tables III and

IV, it is concluded that proper number of copies and their corresponding locations should be made in order to minimize the blocking probability as well as the capacity usage.

VI. Conclusion

In this paper, an optimal video placement scheme is proposed for a batching VOD multi-server system. By formulating the video placement problem as a modified bin-packing problem, the problem can be effectively solved by the proposed hybrid genetic approach. Given a specified blocking probability, it is demonstrated that the minimum batching interval can be determined and the copies of each video are allocated with the server capacity usage minimized.

Appendix I
Coefficient of Variation for Batched Request

The interarrival time Y of the batched requests is equal to the batching interval W plus the original interarrival time X of the user requests with mean $1/\lambda$.

Let $\overline{Y}$ be the mean of Y, we have

$$\begin{aligned}\overline{Y} &= \int_0^\infty (W+x)\lambda e^{-\lambda x}\,dx \\ &= W + \frac{1}{\lambda}\end{aligned}$$

and

$$\begin{aligned}\overline{Y^2} &= \int_0^\infty (W+x)^2\lambda e^{-\lambda x}\,dx \\ &= W^2 + \frac{2W}{\lambda} + \frac{2}{\lambda^2}\end{aligned}$$

The standard deviation (σ_Y) of Y is computed as

$$\begin{aligned}\sigma_Y &\equiv \sqrt{\overline{Y^2} - \overline{Y}^2} \\ &= \frac{1}{\lambda}\end{aligned}$$

Hence, the coefficient of variation (C_Y) of Y is equal to

$$\begin{aligned}C_Y &\equiv \frac{\sigma_Y}{\overline{Y}} \\ &= \frac{1}{\lambda W + 1} \\ &< 1 \quad \forall W > 0\end{aligned}$$

Since the coefficient of variation of X (C_X) is equal to 1, $C_Y < C_X$ for all positive W.

References

[1] C. C. Aggarwal, J. L. Wolf, and P. S. Yu, "A permutation-based pyramid braodcasting schem for video-on-demand systems," in *Proc.IEEE Int. Conf. Multimedia Computing and Systems*, June 1996, pp. 253–258.

[2] S. A. Barnett and G. J. Anido, "A cost comparison of distributed and centralized approaches to video-on-demand," *IEEE J. Select. Areas Commun.*, vol. 14, no. 6, pp. 1173–1183, Aug. 1996.

[3] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed: Prentice Hall, 1992, p. 179.

[4] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *INFOCOM'99*, vol. 1, Mar. 1999, pp. 126–134.

[5] S. H. G. Chan, F. Tobagi, and T. M. Ko, "Providing on-demand video services using request batching," in *IEEE Int. Conf. Communication*, vol. 3, 1998, pp. 1716–1722.

[6] S. H. G. Chan and S. H. I. Yeung, "Client buffering techniques for scalable video broadcasting over broadband networks with low user delay," *IEEE Trans. Broadcast.*, vol. 48, no. 1, pp. 19–26, Mar. 2002.

[7] K. Cleary, "Video on demand—Competing technologies and services," in *Int. Broadcasting Convention*, 1995, pp. 432–437.

[8] J. E. G. Coffman, M. R. Garey, and D. S. Johnson, "Approximation algorithms for bin-packing—An updated survey," in *Algorithm Design for Computer System Design*, NY: Springer-Verlag, 1984, pp. 49–99.

[9] A. Dan, P. Shahabuddin, D. Sitaram, and D. Towsley, "Channel allocation under batching and VCR control in movie-on-demand servers," *J. Parallel Distrib. Comput.*, vol. 30, pp. 168–179, Nov. 1995.

[10] W. Fernandex de la Vega and G. S. Lueker, "Bin packing can be solved within $1+\epsilon$ in linear time," *Combinatorica*, vol. 1, pp. 349–355, 1981.

[11] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms—Part I: A unified formulation," *IEEE Trans. Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 28, no. 1, pp. 26–37, Jan. 1998.

[12] ——, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms—Part II: Application example," *IEEE Trans. Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 28, no. 1, pp. 38–47, Jan. 1998.

[13] L. Gao and D. Towsley, "Threshold-based multicast for continuous media delivery," *IEEE Trans. Multimedia*, vol. 3, no. 4, pp. 405–414, Dec. 2001.

[14] D. J. Gemmell, H. M. Vin, D. D. Kandlur, P. V. Rangan, and L. A. Rowe, "Multimedia storage servers: A tutorial," *IEEE Computer*, pp. 40–49, May 1995.

[15] A. Girard, *Routing and Dimensioning in Circuit-Switched Networks*: Addison-Wesley, 1990.

[16] J. H. Holland, *Adaption in Natural and Artificial Systems*: MIT Press, 1975.

[17] K. A. Hua and S. Sheu, "Skyscraper broadcasting: A new broadcasting scheme for metropolitan video-on-demand systems," in *SIGCOMM'97*, France, Sept. 1997, pp. 899–910.

[18] K. A. Hua, Y. Cai, and S. Sheu, "Patching: A multicast technique for true video-on-demand services," in *Proc. ACM Multimedia Conf.*, Bristol, U.K., Sept. 1998.

[19] D. S. Johnson, A. Demers, J. D. Ullman, M. R. Garey, and R. L. Graham, "Worst-case performance bounds for simple one-dimensional packing algorithms," *SIAM J. Computing*, vol. 3, pp. 299–325, 1974.

[20] N. Karmarkar and R. M. Karp, "An efficient approximation scheme for the one-dimensional bin packing problem," in *Proc. 23rd Annual FOCS*, 1982, pp. 312–320.

[21] L. Kleinrock, *Queueing Systems, Volume I: Theory*: John Wiley & Sons, Inc., 1975, p. 381.

[22] J. Y. B. Lee, "On a unified architecture for video-on-demand services," *IEEE Trans. Multimedia*, vol. 4, no. 1, pp. 38–47, Mar. 2002.

[23] W. Liao and V. O. K. Lee, "The split and merge (SAM) protocol for interactive video-on-demand systems," in *Proc. IEEE Infocom'97*, vol. 3, 1997, pp. 1349–1356.

[24] T. D. C. Little and D. Venkatesh, "Prospects for interactive video-on-demand," *IEEE Multimedia*, vol. 1, pp. 14–24, 1994.

[25] K. F. Man, K. S. Tang, and S. Kwong, *Genetic Algorithms: Concepts and Designs*: Springer Verlag London, 1999.

[26] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Program*, 3rd ed: Springer-Verlag, 1996.

[27] W. F. Poon and K. T. Lo, "Design of multicast delivery for providing VCR functionality in interactive video-on-demand systems," *IEEE Trans. Broadcast.*, vol. 45, no. 1, pp. 141–148, Mar. 1999.

[28] W. F. Poon, K. T. Lo, and J. Feng, "A hybrid delivery strategy for a video-on-demand system with customer reneging behavior," *IEEE Trans. Broadcast.*, vol. 48, no. 2, pp. 140–150, June 2002.

[29] D. Sitaram and A. Dan, *Multimedia Servers: Applications, Environments, and Design*: Morgan Kaufmann Publishers, 2000.

[30] K. S. Tang, K. T. Ko, S. Chan, and E. W. M. Wong, "Optimal file placement in VOD system using genetic algorithm," *IEEE Trans. Industrial Electronics*, vol. 48, no. 5, pp. 891–897, Oct. 2001.

[31] S. Viswanathan and T. Imielinaki, "Metropolitan area video-on-demand service using pyramid broadcasting," *Multimedia Syst.*, vol. 4, no. 4, pp. 197–208, Aug. 1996.

Wallace K. S. Tang obtained his B.Sc. from the University of Hong Kong in 1988, and both the M.Sc. and Ph.D. from the City University of Hong Kong in 1992 and 1996, respectively. He is currently an Associate Professor in the Department of Electronic Engineering, City University of Hong Kong. He has published over 30 journal papers and book chapters, and coauthored two books in the area of genetic algorithms published by Springer-Verlag. He is a member of IFAC Technical Committee on Optimal Control (Evolutionary Optimization Algorithms) and a member of Intelligent Systems Committee in IEEE Industrial Electronics Society. His research interests include evolutionary algorithms and network optimization.

Eric W. M. Wong (S'87–M'90–SM'00) received the B.Sc. and M.Phil. degrees in Electronic Engineering from the Chinese University of Hong Kong, Hong Kong, in 1988 and 1990, respectively, and the Ph.D. degree in Electrical and Computer Engineering from the University of Massachusetts at Amherst, U.S.A., in 1994.

He joined the City University of Hong Kong in 1994, where he is now an Associate Professor in the Department of Electronic Engineering and a member of the Networking Group of the department. His research interests are in content delivery networks, video-on-demand, optical networks, and dynamic routing. The most notable of these involved the analytical modeling of the least loaded routing scheme in circuit-switched networks. The model drastically reduces the computational complexity of designing and dimensioning telephone systems. The model has also served as a core for the analysis of many other advanced routing schemes, such as the Real-Time Network Routing currently implemented in AT&T telephone network. His email address is ewong@ee.cityu.edu.hk.

Sammy Chan received his B.E. and M.Eng.Sc. degrees in Electrical Engineering from the University of Melbourne, Australia, in 1988 and 1990, respectively, and a Ph.D. degree in Communication Engineering from the Royal Melbourne Institute of Technology, Australia, in 1995. From 1989 to 1994, he was with Telecom Australia Research Laboratories, first as a Research Engineer, and between 1992 and 1994 as a Senior Research Engineer and Project Leader. Since December 1994, he has been with the Department of Electronic Engineering, City University of Hong Kong, where he is currently an Associate Professor.

King-Tim Ko was born in Hong Kong, and graduated from The University of Adelaide, South Australia with a First Class Honors B.Eng. degree in Electrical Engineering in 1978. With the Australian Commonwealth Postgraduate Scholarship, his Ph.D. degree in Communication Engineering was completed in 1982.

He joined Telecom Australia Research Laboratories in Melbourne. His main research topics included the design and dimensioning of telecommunication networks. In 1986, he joined the Department of Electronic Engineering of the City University of Hong Kong—China, and currently an Associate Professor of the same Department. In research, he is involved in the designs and congestion control in broadband communication networks as well as multimedia applications.