International Journal of Innovative
Computing, Information and Control
Volume **13**, Number **6**, December **2017**

 ISSN 1349-4198
pp. **2009–2023**

OPTIMIZING THE VIDEO PLACEMENT ON REPLICA SERVERS OF VIDEO-ON-DEMAND SYSTEMS BY BRANCH AND BOUND APPROACH

Meng-Huang Lee

Department of Information Technology and Management
Shih Chien University
No. 70, Dazhi Street, Zhongshan Dist., Taipei 10462, Taiwan
meng@g2.usc.edu.tw

Received February 2017; revised June 2017

Abstract. *To adapt to the constraints on the bandwidth and storage capacity in replica servers for optimal video replication, an exhaustive evaluation approach is traditionally used to generate all possible solutions, after which an optimal video placement solution is determined. The time complexity of this approach can reach as high as $O(n!)$. In our previous work, we proposed a heuristic algorithm to approximate the optimal video placement in replica servers while reducing the worst-case bandwidth demand at the origin server. In this paper, we further propose a deterministic algorithm, termed the potential and look ahead (PLA) algorithm, which produces the optimal solution for this optimization issue through a significant reduction in the solution space with a branch and bound approach. The PLA algorithm was proved to generate the optimal solution with less solution space required by the traditional exhaustive evaluation approach.*
Keywords: Video on demand, Replica, Origin server, Placement optimization, Look ahead, Branch and bound

1. **Introduction.** In [4,5], the origin server stored all the video programs in the video-on-demand (VOD) system, and the replica servers stored parts of copies of the videos from the origin server (Figure 1). The users of this system were grouped into numerous clusters, and each cluster was assigned a replica server that was closer to the users than the origin server. In such a system, if the replica server of the cluster already contains a replica of a user-requested video program, streaming bandwidth is required only between the user and the replica server. However, if the requested video program is not in the replica server, streaming bandwidth is required between the origin server and the replica server as well as between the replica server and the user. The higher the number of video copies stored in the replica server is, the higher the likelihood that the replica server can service the users' video requests is. This leads to a reduction in the bandwidth required from the origin server, thereby reducing the network costs. However, a large number of video program copies on the replica server increase storage costs; therefore, a tradeoff is necessary between the network and storage costs.

A replication strategy involves video selection and placement [4-6,9,10,14-18]. In the selection phase, the offered load pattern or viewing request probability are used to determine the videos that require replication to appropriately distribute the load of the origin server. During the placement phase, the constraints on the storage capacity and bandwidth of replica servers must be considered.

For system analysis, a VOD system can be modeled as a queue system. In this case, the replication strategy should shift from a video's viewing request popularity to the offered load of the VOD system. According to [17,18], service time (i.e., viewing duration) and

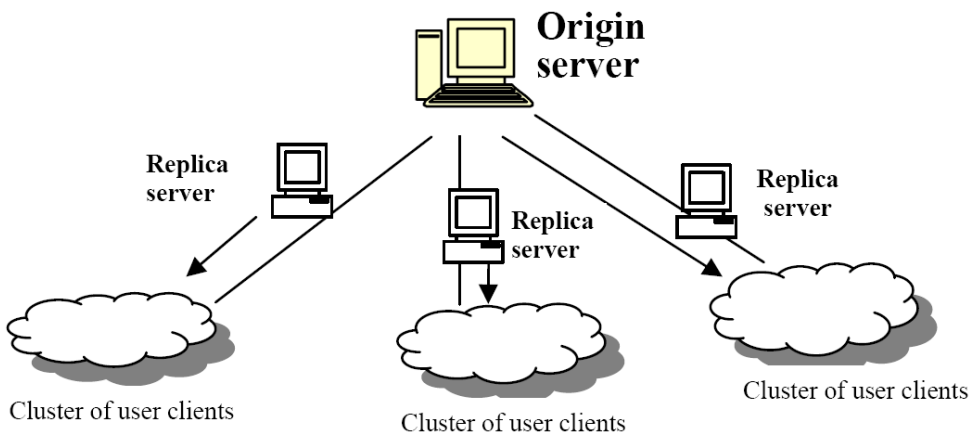

FIGURE 1. Video-on-demand system with replica servers

viewing request probability are parameters that must be considered when determining the offered load of a system. Because the viewing duration and video length are directly proportional [2,19], the video length can be used to simulate the viewing duration for the queue model.

Because video length has a significant effect on the worst-case bandwidth demand at the origin server, in our previous work [24], we investigated the combined influence of the viewing request probability and video length on the bandwidth requirements at the origin server and established a theoretical model for the selection phase of replication. This selection model was based on the offered load [17,18] rather than on the viewing request probability [4-6,9,10,14-16,25]. In addition, this selection model resulted in a reduction in the worst-case bandwidth demand at the origin server. We called this model the QD major selection model, in which Q represents the viewing request probability and D represents the video length. The value of the viewing request probability multiplied by the video length was defined as the QD value of a video program. We demonstrated that the reduction in the worst-case bandwidth demand of the origin server is associated with the sum of the QD of video programs in the replica servers. Therefore, during the video selection phase of replication, we selected the video programs with the largest QD value and placed them within the replica servers. Traditional selection models that consider only the viewing request probability are called Q major selection models. The QD major selection model was verified by simulating a VOD system with replica servers and comparing the results with those of Q major selection models. When measuring the worst-case bandwidth demand at the origin server, we found that the QD major selection model outperformed the Q major selection models. Among the 10 testing data sets, two exhibited 5% advantages, and the others exhibited 10% to 25% advantages.

All replica servers have respective resource constraints related to the storage capacity and the streaming bandwidth that they can provide. Therefore, for the video placement phase of replication, a corresponding objective function was proposed [6]. This objective function must comply with the resource constraints of the replica server and allow the placement of videos in line with the requirements of a particular optimization condition. [23] considered the resource constraints of replica servers and proposed the space-to-bandwidth ratio (SBR) as a criterion for video placement. The argument for considering the SBR is as follows: a replica server has restrictions on its space and bandwidth, and a video also has requirements for space (i.e., video size) and bandwidth (i.e., the streaming bandwidth necessary to play the video). Therefore, for optimal video placement, the selected video should match the SBR of the replica server. According to the SBR criteria

proposed in [23], the characteristics of a video (request bandwidth and video length) and a replica server (bandwidth and storage space) can be represented by rectangles with the width and height representing the bandwidth and space, respectively.

To optimize the QD major placement model, as many smaller video rectangles as possible must be placed within a larger replica server rectangle, such that the sum of the area of these video rectangles is maximized. This is similar to the bin packing problem, which is affected by weight and volume constraints. The placement of a small rectangle within a large rectangle leads to a utilization loss. Figure 2 illustrates that the placement of rectangle A divides the entire block into four blocks (A, B, C, and D). Blocks B and C can no longer accommodate any other rectangles. Block D, which is the only space that can continue to accommodate other rectangles, represents the remaining area of the replica server. Therefore, when our objective function places video rectangles into a large replica server rectangle to maximize the total area of the placed video rectangles, the remaining area must also be considered. Because the widths and heights of video rectangles cannot overlap, placing too many small video rectangles within the same replica server rectangle would result in substantial wasted space. For example, the total area of R1 and R2 in Figure 3(a) is 12. The area of R3 in Figure 3(b) is also 12 (R3 = R1 + R2). The remaining area in Figure 3(a) is 96, whereas the remaining area in Figure 3(b) is 126.5. Hence, the remaining area in Figure 3(b) is greater than that in Figure 3(a). This indicates that Figure 3(b) has higher potential for the placement of video rectangles with

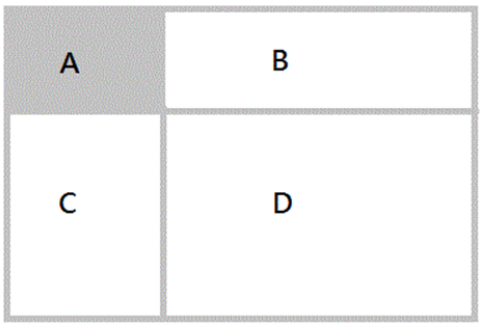

FIGURE 2. After rectangle A is placed, only block D can continue to accommodate the placement of other video rectangles. Blocks B and C are no longer eligible for the subsequent placement of video rectangles.

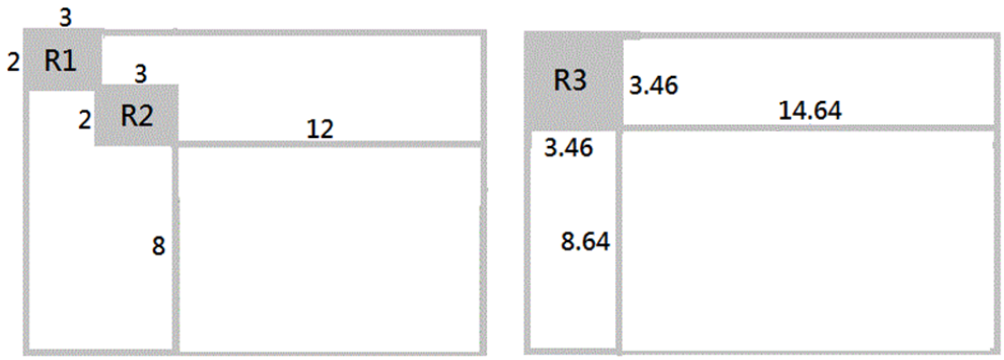

FIGURE 3. Effect on the remaining area of the replica server caused by the placement of different combinations of video rectangles with the same total area

larger areas. Therefore, video rectangles with a larger area can achieve a higher overall allocated area.

Considering the SBR placement model, in our previous work [24], we developed a heuristic algorithm for the placement issue that comprises the QD major selection model and the resource constraints of replica servers. This heuristic algorithm generated near-optimal solutions for this optimization issue. In this paper, we further construct a deterministic algorithm that generates the optimal solution within a considerably limited solution space size by using a branch and bound approach.

Because this paper concerns the video placement issues discussed in our previous work, the problem statement of our previous work is partially repeated in Section 2. In Section 3, a deterministic algorithm, called the potential and look ahead (PLA) algorithm, is proposed; this algorithm reduces the solution space by using a brand and bound approach for the optimization of the QD major placement model. The proposed algorithm is evaluated in Section 4, and conclusions are presented in Section 5.

2. **Problem Statement.** In a VOD system with replica servers, videos placed on replica servers can reduce the required bandwidth of origin server. The more videos are placed on replica servers, the less bandwidth is required from the origin server. Due to the resource constraints of replica servers, only part of videos can be placed on replica servers. How to minimize the worst-case bandwidth demand on the origin server while meeting the resource constraints of replica servers is a challenge for a video placement strategy.

The optimization problem for this video placement is a combinatorial problem. In our previous work [24], we addressed the problem with a heuristic algorithm, whereas in this paper we attempt to solve it using a deterministic algorithm. Therefore, the statements of the combinational optimization problem shown as follows are partially consistent with those of our previous work.

V: set of video programs in the VOD system

v_i: ith video program in the VOD system, where $v_i \in V$

d_i: video length of v_i

q_i: viewing request probability of v_i, where $q + q_1 + q_2 + \cdots + q_{N-1} = 1$

b_i^v: bandwidth necessary to play the video object v_i

s_i^v: size of the video object v_i

$S^{replica\_server}$: storage space of the replica server

$B^{replica\_server}$: bandwidth of the replica server

$s_{available}^{replica\_server}$: available storage space of the replica server

$s_{alloc}^{replica\_server}$: allocated storage space of the replica server

$b_{available}^{replica\_server}$: available bandwidth of the replica server

$b_{alloc}^{replica\_server}$: allocated bandwidth of the replica server

$QD^{replica\_server}$: QD value of the replica server (i.e., $B^{replica\_server} \times S^{replica\_server}$)

qd_i^v: QD value of v_i (i.e., $q_i \times d_i$)

$qd_{alloc}^{replica\_server}$: sum of the QD values of the video programs currently accumulated in the replica server, which is either smaller than or equal to $s_{alloc}^{replica\_server} \times b_{alloc}^{replica\_server}$

$qd_{available}^{replica\_server}$: $b_{available}^{replica\_server} \times s_{available}^{replica\_server}$

$$\text{Maximize} \quad \sum_{j=1}^{N} b_j^v s_j^v x_j \tag{1}$$

$$\text{Subject to} \quad \sum_{j=1}^{N} b_j^v x_j \leqq B^{replica\_server} \tag{2}$$

$$\sum_{j=1}^{N} s_j^v x_j \leqq S^{replica\_server} \tag{3}$$

The value of x_j in Equations (1) to (3) is either zero or one. When the value of x_j is 1, the video v_j is selected for replication on the replica server. The aim of this system is to minimize the worst-case bandwidth demand on the origin server.

On the basis of our previous work [24], the higher the sum of the QD values of the selected videos is, the smaller the worst-case bandwidth demand on the origin server is. Therefore, the objective in Equation (1) favors selection processes that attain relatively large QD values for the videos. However, because the replica server is constrained by its storage space and bandwidth capacity, the selection of video combinations using Equation (1) is subject to these resource constraints. The constraints on the bandwidth capacity and the storage space of the replica server are stated in Equations (2) and (3), respectively. The exhaustive evaluation approach is traditionally used for the optimization computations in Equations (1)-(3). Because the time complexity for solving this optimization is very high, a solution with other faster algorithms must be found.

In the following sections, a deterministic algorithm based on a branch and bound criterion is proposed in Section 3 for the optimization problem in Equations (1)-(3). The test case evaluation is shown in Section 4.

3. Branch and Bound Approach for Reducing the Solution Space of QD Major Placement Model.

3.1. Solution space of QD major placement model and its branch and bound criteria.

The optimization of QD major placement model is required for maximizing the QD sum in replica servers, which can effectively reduce the worst-case bandwidth demand at origin server. According to combinatorial theory, typically a brute force algorithm finds the best solution by exhaustively enumerating all the possibilities. For example, the solution space for the three video files v_0, v_1, and v_2 are (v_0), (v_1), (v_2); (v_0, v_1), (v_0, v_2), (v_1, v_2); and (v_0, v_1, v_2), respectively. The problem solving process approximates a tree topology, expanding each solution node in the order of stages. Figure 4 shows the

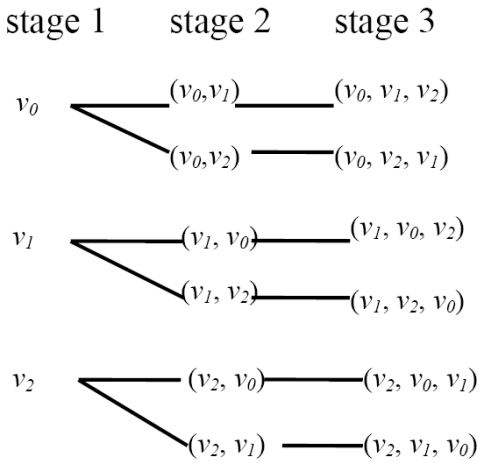

FIGURE 4. The solution space for the combination of three video files: v_0, v_1, and v_2

solution nodes at each stage. Stage 1 includes (v_0), (v_1), and (v_2), stage 2 includes (v_0, v_1), (v_0, v_2), (v_1, v_0), (v_1, v_2), (v_2, v_0), and (v_2, v_1), and stage 3 includes (v_0, v_1, v_2), (v_0, v_2, v_1), (v_1, v_0, v_2), (v_1, v_2, v_0), (v_2, v_0, v_1), and (v_2, v_1, v_0). Since the order of video placement does not affect the QD sum of placed videos, the solution space of these 15 solution nodes is equal to that of the 6 nodes, i.e., (v_0), (v_1), (v_2), (v_0, v_1), (v_0, v_2), (v_1, v_2), and (v_0, v_1, v_2).

However, sometimes we can determine if a given node in the solution space will not lead to the optimal solution – either because the given solution and all its successors are infeasible or because we have already found a solution that is guaranteed to be better than any successor of the given solution. In such cases, the given node and its successors need not be considered. In effect, we can prune the solution tree, thereby reducing the number of solutions to be considered. Since this paper aims to develop an algorithm for maximizing the QD value, if the QD sum of solution node A and its successor is proved to be less than that of solution node B, then solution A can be pruned. This technique can save considerable computation cost by such a branch and bound evaluation.

Consider video rectangles including $(1, 9)$, $(1, 6)$, $(3, 27)$, $(3, 6)$, $(7, 14)$, $(7, 35)$, $(10, 30)$, $(10, 10)$, $(13, 52)$, and $(13, 13)$ are placed into the replica server rectangle $(20, 60)$. We aim at a larger total area of the video rectangles placed into the replica server rectangle $(20, 60)$. There are more than 10! possible solutions. In the following discussions, we show examples that utilize the characteristics of the remaining area discussed in Section 1 as the branch and bound criteria such that the solution space can be effectively reduced. The examples are provided for the placement sequences that begin with video rectangles $(3, 27)$ and $(13, 52)$, respectively.

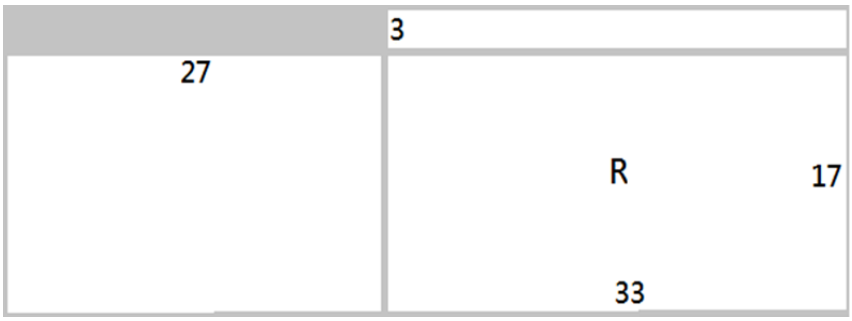

FIGURE 5. After placement of video rectangle $(3, 27)$, the remaining area R is $17 * 33$.

As shown in Figure 5, when video rectangle $(3, 27)$ is placed into the replica server rectangle $(20, 60)$, the remaining area R is $(20 - 3) * (60 - 27) = 561$ – this remaining area R represents the largest space for subsequent video rectangles to be placed after the placement of video rectangle $(3, 27)$. Similar to the previous discussions, the total area for the subsequent video rectangles that can be placed will not exceed this remaining area R, regardless of any placement sequence of video rectangles. Therefore, the maximum area of the placement sequence that begins with the video rectangle $(3, 27)$ will not exceed the total area of video rectangle $(3, 27)$ and remaining area R, i.e., $3 * 27 + 561 = 642$, which means that 642 is the upper bound on the area for any solution nodes of the placement sequence that begins with video rectangle $(3, 27)$.

In another example, the placement sequence that begins with video rectangle $(13, 52)$ is considered. As shown in Figure 6, after the video rectangle $(13, 52)$ is placed within the replica server rectangle (20,60), if we look ahead and place a video rectangle $(3, 6)$ into the remaining area, then the total area would be the area of video rectangle $(13, 52)$ added to the area of video rectangle $(3, 6)$, i.e., $13 * 52 + 3 * 6 = 694$. At this point,

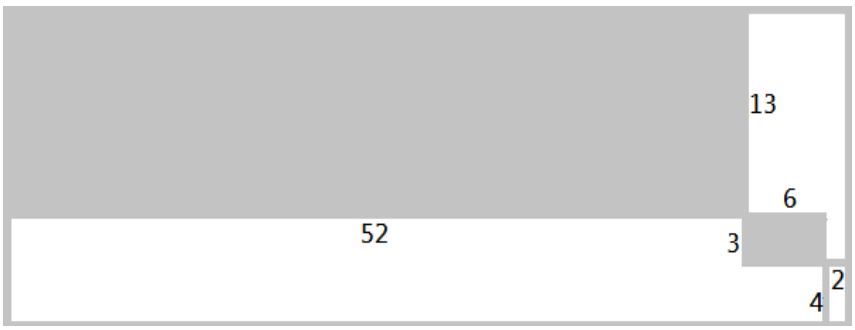

FIGURE 6. After placement of the video rectangle $(13, 52)$, a look-ahead video rectangle $(3, 6)$ is placed.

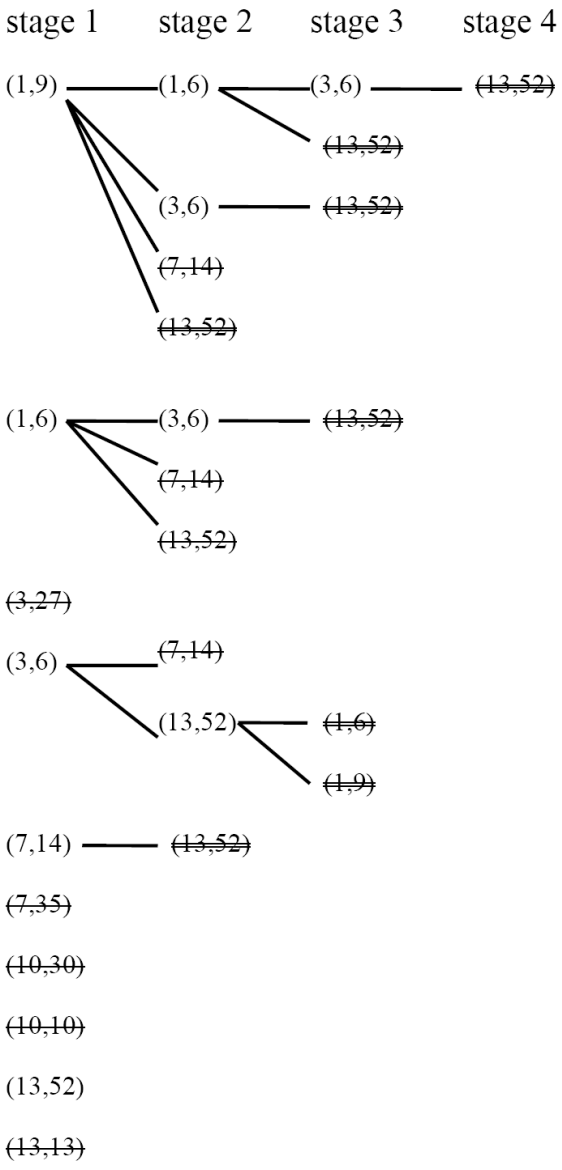

FIGURE 7. The solution space for video rectangles $(1, 9)$, $(1, 6)$, $(3, 27)$, $(3, 6)$, $(7, 14)$, $(7, 35)$, $(10, 30)$, $(10, 10)$, $(13, 52)$, and $(13, 13)$ to be placed within the replica server rectangle $(20, 60)$

there is still space in the remaining area for more rectangles to be placed. Therefore, the total area of the placement sequence that begins with video rectangle $(13, 52)$ is greater than 694, indicating that 694 is the lower bound on the area of any solution nodes for the placement sequence that begins with video rectangle $(13, 52)$.

The two examples show that the upper bound on the total area of any solution nodes for the placement sequence that begins with video rectangle $(3, 27)$ would not exceed 642, while the lower bound on the total area of any solution nodes for the placement sequence that begins with video rectangle $(13, 52)$ is greater than 694. Therefore, the total area of the placement sequence that begins with video rectangle $(3, 27)$ is less than that of the placement sequence that begins with video rectangle $(13, 52)$. Since we aim for the placement sequence of the largest total area, the placement sequence that begins with video rectangle $(3, 27)$ can be neglected in comparison with the placement sequence that begins with video rectangle $(13, 52)$. Similarly, placement sequences that begin with video rectangles $(7, 35)$, $(10, 30)$, $(10, 10)$, or $(13, 13)$ can be neglected, since every one of their total areas is smaller than 694. In this way, the size of the solution space can be reduced considerably. As a result, what previously required a total more than 10! solution nodes now requires only 17 solution nodes.

Figure 7 shows the solution space from the above examples, where the single strikethroughs represent solution nodes that require no further expansion for new solution nodes, because their upper bound areas are less than the lower bound area of a specific solution node within the stage. The double strikethroughs represent solution nodes that require no further expansion for new solution nodes because the widths or heights of the video rectangles of these solution nodes exceed the width or height of the replica server rectangle. As a result, 5 out of the 10 original solution nodes at stage 1 require no further expansion for new solution nodes, resulting in a 50% reduction in solution space. For stage 2, there are 8 video rectangles that do not exceed the width or height limit of replica server rectangle, among which there are 4 that do not require further expansion for new solution nodes because of the branch and bound criteria discussed above. Therefore, the total area of the combination of video rectangles $(3, 6)$ and $(13, 52)$ represents the maximum value in this example.

3.2. **Branch and bound approach for optimizing QD major placement on replica servers.**

Lemma 3.1. *When rectangles r_0, $r_1 \ldots r_i$ are placed within a rectangle R, the total area of these rectangles is no more than the area of R.*

Proof: Since rectangles r_0, $r_1 \ldots r_i$ are placed within a rectangle R, this is obviously true that the total area of these rectangles is no more than the area of R.

Remark 3.1. *If this rectangle R is the remaining area described in the previous sections, then the area of this rectangle R is the upper bound on the total area of all the rectangles r_0, $r_1 \ldots r_i$ that can be placed within rectangle R.*

Lemma 3.2. *When a rectangle r is placed within a rectangle R, because more rectangles can be placed within rectangle R subsequently, the total area of all the rectangles that can be placed within rectangle R is larger than or equal to the area of r.*

Proof: Since rectangle r is placed within a rectangle R, the remaining area of rectangle R may accommodate other rectangles to be placed subsequently. Therefore, this is obviously true that the total area of all the placed rectangles is larger than or equal to the area of r.

Remark 3.2. *If this rectangle r is the look-ahead rectangle described in the previous section, then there is still space in rectangle R for subsequent placement of more rectangles, with a subsequent increase in the entire allocated area. Therefore, the area of the allocated rectangles within rectangle R added to the area of rectangle r is the lower bound on the total area of all the rectangles that can be placed within rectangle R.*

Theorem 3.1. *$\underline{A}$ refers to the total area of solution node A and its look-ahead rectangle; $\overline{B}$ refers to the total area of solution node B and its remaining area. If $\underline{A} \geq \overline{B}$, then the total area of B and its successors is no more than that of A and its successors. Therefore, during the solution space expanding process, node B and its successors can be pruned.*

Proof:

If A denotes the maximum QD total area of node A and its successors, then according to Lemma 3.2, $A \geq \underline{A}$;

If B denotes the maximum QD total area of node B and its successors, then according to Lemma 3.1, $B \leq \overline{B}$.

Since $\underline{A} \geq \overline{B}$, $A \geq \underline{A} \geq \overline{B} \geq B$, during the solution space development process, node B and its successors can be pruned.

Potential and look ahead algorithm for optimizing QD major placement on replica servers

Because the proposed algorithm is a kind of deterministic algorithm, all the possible solutions are expanded in order – the solution space is a tree-like structure and every solution node is a possible solution. Each solution node is evaluated for branching or bounding based on the Potential and look ahead algorithm shown below, from which the entire solution space can be constructed. Therefore, in the following discussions, a solution node is a video program along with its associated QD information. Parameters for a video listed in the problem statement of Section 2 would become the parameters for the solution node. Two important phrases in this algorithm are worth noting: *Potential* indicates that for each solution node, the allocated QD value is added to the remaining area, which is the upper bound on the QD sum of this solution node and its future possible expanded solution nodes (Lemma 3.1). Therefore, the remaining area here is the *potential* area of the subsequent possible solution nodes derived from this solution node. The other important phrase is *Look Ahead*, which refers to the maximum QD value of the allocated QD value of each solution node in the system added to the QD value of a *look-ahead* rectangle – this denotes the lower bound on the current QD sum value of the system (Lemma 3.2). Based on this and Theorem 3.1, we can determine whether a solution node requires further expansion for new solution nodes, or whether the expansion is to be terminated.

$node_i.v_k$: the video program associated to the solution $node_i$ is v_k.

$node_i.expandable$: "true" means that solution $node_i$ is qualified to generate another solution nodes after the evaluation by the PLA algorithm. "false" means that solution $node_i$ cannot generate another solution nodes.

$node_i.solution\_Nodes$: is a collection of solution nodes from the root to the solution $node_i$. The video programs associated to the solution nodes in $node_i.solution\_Nodes$ mean the placement sequence from the root to the solution $node_i$.

$node_i.qd_{alloc}^{replica\_server}$: the QD sum value of all the video programs placed within the replica server in solution $node_i$. These video programs are associated to the solution nodes in $node_i.solution\_Nodes$.

$node_i.qd_{available}^{replica\_server}$: the QD value of the remaining area of the replica server at solution $node_i$. This area should be $QD^{replica\_server} - node_i.qd_{alloc}^{replica\_server}$.

$node_i.qd^{replica\_server}_{look\_ahead}$: $\max\{qd^v_i$: $v_i \notin node_j.solution\_Nodes\}$

$node_i.qd^{replica\_server}_{alloc\_lower\_bound}$: $node_i.qd^{replica\_server}_{alloc} + node_i.qd^{replica\_server}_{look\_ahead}$

$node_i.qd^{replica\_server}_{alloc\_upper\_bound}$: $node_i.qd^{replica\_server}_{alloc} + node_i.qd^{replica\_server}_{available}$

$stage_i.Nodes\{\}$: a collection of solution nodes in $stage_i$

$stage_i.qd^{replica\_server}_{alloc\_lower\_bound}$: $\max\{node_k.qd^{replica\_server}_{alloc\_lower\_bound}$, where $node_k$ belongs to $stage_i.Nodes\{\}\}$

The overview of the proposed potential and look ahead algorithm is as follows:

```
01 for (i = 0; i < Num_Stages; i++){
02     expand the "expandable" solution nodes in stage_i.Nodes;
03     the expanded solution nodes are appended into stage_{i+1}.Nodes;
04     determine stage_{i+1}.qd^{replica_server}_{alloc_lower_bound} of solution nodes in stage_{i+1}.Nodes;
05     for each solution node, nod_k, in stage_{i+1}.Nodes
06         if (node_k.qd^{replica_server}_{alloc_upper_bound} <= stage_{i+1}.qd^{replica_server}_{alloc_lower_bound})
07             mark node_k.expandable as "false";
08         else
09             mark node_k.expandable as "true";
10 }
```

Algorithm procedure 1: Potential and look ahead algorithm overview

Line 06 in Algorithm procedure 1 represents the branch and bound criteria for the PLA algorithm. For a given solution node, if the QD sum of the allocated area and the remaining area is less than the QD sum of the allocated area and the look-ahead QD, the "expandable" of this solution node would be marked as "false". It indicates that this solution node would not expand to any new solution nodes in the next stage as indicated by Line 07 in the Algorithm procedure 1. Line 02 expands the entire solution space stage by stage. The detailed procedure for Algorithm procedure 1 is as follows:

```
01 for (i = 0; i <Num_Stages; i++){
02     nodes_expand(stage_i.Nodes);
03     nodes_expandable_eval(stage_{i+1}.Nodes);
04 }
05 nodes_expand(stage_i.Nodes){
06    for each solution node, node_j, in stage_i.Nodes
07        if (node_j.expandable == true){
08            for each video, v_k, in V
09                if (v_k ∉ node_j.solution_Nodes){
10                    m = create_node(v_k);
11                    add m to stage_{i+1}.Nodes;
12                    add node_j.solution_Nodes to m.solution_Nodes;
13                {
14        }
15 }
16 nodes_expandable_eval(stage_{i+1}.Nodes){
17        for each expandable node, node_j, in stage_{i+1}.Nodes{
18            qd_lower_bound(node_j);
19            qd_upper_bound(node_j);
20        }
21        determine stage_{i+1}.qd^{replica_server}_{alloc_lower_bound};
22        for each solution node, node_j, in stage_{i+1}.Nodes
```

```
23          if (node_j.qd^{replica_server}_{alloc_upper_bound} <= stage_{i+1}.qd^{replica_server}_{alloc_lower_bound})
24              node_j.expandable = false;
25          else
26              node_j.expandable = true;
27 }
28 qd_lower_bound(node_j){
29      for each node, node_k, in node_j.solution_Nodes
30          node_j.qd^{replica_server}_{alloc} = node_j.qd^{replica_server}_{alloc} + node_k.qd^v_k;
31      node_j.qd^{replica_server}_{alloc_lower_bound} = node_j.qd^{replica_server}_{alloc} + node_i.qd^{replica_server}_{look_ahead};
32 }
33 qd_upper_bound(node_j){
34      for each solution node, node_k, in node_j.solution_Nodes
35          node_j.qd^{replica_server}_{alloc} = node_j.qd^{replica_server}_{alloc} + node_k.qd^v_k;
36      node_j.qd^{replica_server}_{alloc_upper_bound} = node_j.qd^{replica_server}_{alloc} + node_j.qd^{replica_server}_{available};
37 }
```

Algorithm procedure 2: Detailed procedure for the potential and look ahead algorithm

Lines 01-04 of Algorithm procedure 2 comprise the main body of the PLA algorithm. The nodes_expand() function in Line 02 expands the solution nodes stage by stage to construct the entire solution space, and the nodes_expandable_eval() function in Line 03 determines whether a solution node is qualified to continue expanding to new solution nodes. The nodes_expandable_eval() function in Lines 23 to 26 provides the branch and bound criteria for the PLA algorithm, consistent with the Line 06 in Algorithm procedure 1.

4. **Test Case Evaluation.** Our previous work [24] proposed a heuristic algorithm to solve this video placement problem. In the most cases, the results produced by this heuristic algorithm are the same as the optimal results produced by LINGO. However, there is some test case that the heuristic algorithm does not produce the optimal solution. We use this test case for the evaluation of PLA algorithm. The video parameters of this test are in Table 1 and replica server configurations of this test are in Table 2.

To solve this optimization problem, LINGO is used for the calculations for Equations (1)-(3). The replica server configuration is shown in Table 2, the lump sum of allocated QD obtained from LINGO is 9.945, and that obtained from the PLA algorithm is 9.945. The video programs allocated using the PLA algorithm are p_2, p_4, p_7, p_{10}, which are consistent with those allocated using LINGO.

Figure 8 shows a comparison of the number of solution nodes at each stage between the exhaustive evaluation approach and the PLA algorithm. Because the number of solution nodes generated through the exhaustive evaluation approach is too large, the number of solution nodes in the y-axis is processed by $\log_{10}$. As shown in Figure 8, the number of solution nodes from the PLA algorithm is less than that from exhaustive evaluation approach at every stage, and the difference increases with an increase in the number of stages. The PLA algorithm converges at stage 5, while the exhaustive evaluation approach does not converge until stage 7. It is worth noting that the infeasible nodes have been excluded for the comparison between these two algorithms. A so-called infeasible node is a solution node whose bandwidth or space exceeds that of the replica server, and therefore can no longer generate any new solution node. This restriction must be followed in PLA algorithms and brute force exhaustive evaluation approach, and therefore is not a feature of the PLA algorithm. As a result, the effect of infeasible nodes is not discussed when the algorithms are compared with each other. In addition, the number of solution nodes is

TABLE 1. Viewing request probability and length of each video program for the evaluation

Program index	Viewing request probability	Program duration	Program index	Viewing request probability	Program duration
P_0	25.3	5	P_{14}	1.7	10
P_1	12.6	5	P_{15}	1.6	45
P_2	8.4	60	P_{16}	1.5	60
P_3	6.3	30	P_{17}	1.4	10
P_4	5.1	45	P_{18}	1.3	30
P_5	4.2	15	P_{19}	1.3	10
P_6	3.6	5	P_{20}	1.2	30
P_7	3.2	60	P_{21}	1.2	120
P_8	2.8	60	P_{22}	1.1	45
P_9	2.5	90	P_{23}	1.1	90
P_{10}	2.3	30	P_{24}	1	90
P_{11}	2.1	120	P_{25}	1	45
P_{12}	1.9	60	P_{26}	0.9	30
P_{13}	1.8	10	P_{27}	0.9	45

TABLE 2. Replica server configures for the evaluation

Bandwidth	Space
Supports streaming bandwidth for 20% of viewing requests received by the replica server	200 min of video data

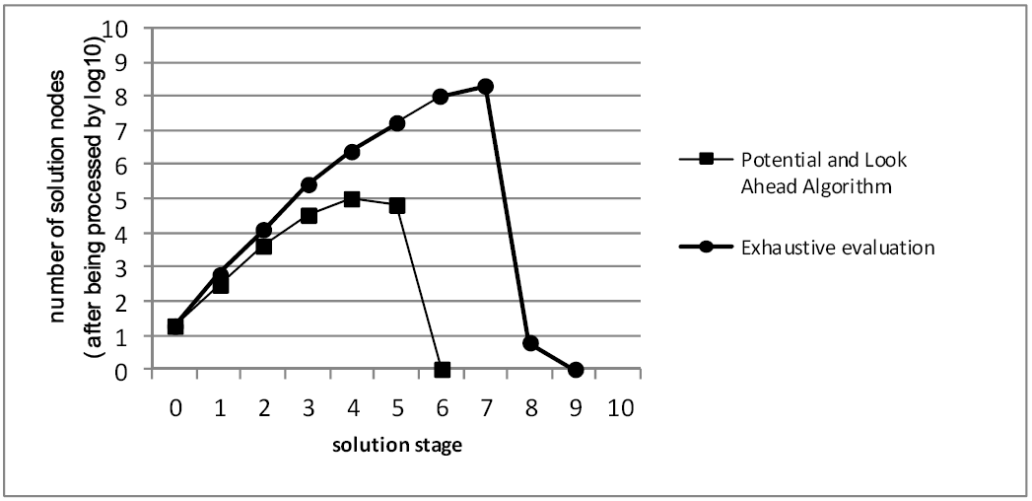

FIGURE 8. Number of solution nodes at each solution stage

157,292 for the PLA algorithm, and 420,198,025 for the exhaustive evaluation approach – the number of solution nodes derived from exhaustive evaluation is 2,671 times that of the PLA algorithm.

Figure 9 shows the QD value at each stage of the solution space for the QD values of the lower bound of allocated area and the look ahead area, as well as the upper bound of allocated area and remaining area. As expected, the QD value of the lower bound of allocated area and the look ahead area increases with increasing stages, achieving its

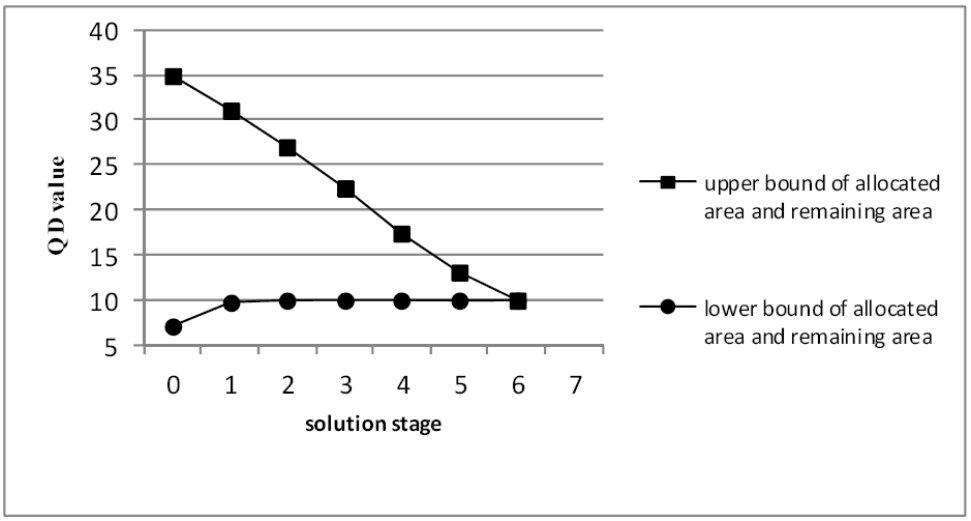

FIGURE 9. QD values of the upper bound of allocated area and remaining area, and those of the lower bound of allocated area and look ahead area, at each solution stage

maximum, 9.945, at the 2nd stage. On the other hand, the QD value of the upper bound of allocated area and the remaining area decreases with increasing stages. These two curves intersect at the 6th stage, i.e., the QD value of the lower bound of allocated area and the look ahead area will be greater than that of the upper bound of allocated area and the remaining area after this point. As a result, the PLA algorithm stops at the 6th stage, since at this stage the upper bound of allocated and remaining area of each solution node is less than 9.945, which is the lower bound of allocated area and look ahead area. Therefore, all nodes at the 6th stage are "unexpandable" nodes, and the solution space would no longer expand. This result is consistent with that shown in Figure 8, in which the number of solution nodes of the PLA algorithm drops to 0 at the 6th stage. Therefore, Figure 8 and Figure 9 are mutually verifiable.

5. **Conclusions.** In a VOD system, replicating video programs into replica servers can reduce the bandwidth requirements in an origin server. In our previous work, we proved and verified that the reduction in worst-case bandwidth demand of origin server is associated with the QD sum of video programs replicated in replica servers. However, there are bandwidth and storage capacity constraints in replica servers. Therefore, during the video placement phase of replication, one must consider the allocation of video programs that can reach a maximum QD value under these two constraints. In our previous work, we constructed a heuristic algorithm to achieve the optimization of QD major placement model, which could produce a near-optimal solution very close to that produced by LINGO. In this paper, we further investigated the optimization issue by branch and bound approach and proposed a deterministic algorithm, the PLA algorithm, for the optimization of QD major placement model. The proposed algorithm can generate an optimal solution, while generating comparatively fewer solution nodes than in the traditional exhaustive evaluation approach, thus significantly reducing the solution space. We used the test case in our previous work, in which the heuristic algorithm could not produce the same optimal solution as LINGO, as a verification case. It was demonstrated that the PLA algorithm could produce the same optimal solution as LINGO, with the solution space of only 1/2671 of that in traditional exhaustive evaluation approach.

Acknowledgement. The research is supported by Shih Chien University, Taiwan, under Grant USC 105-08-01001, and National Science Council, Taiwan, under Grant NSC 102-2221-E-158-004.

REFERENCES

[1] M.-H. Lee, Peer-to-peer content delivery network for IPTV network personal video recorder, *Journal of Current Computer Science and Technology*, vol.2, no.3, pp.76-85, 2012.
[2] M.-H. Lee, Comments on worst-case demand in a VOD system with replica servers, *ICIC Express Letters*, vol.6, no.11, pp.2855-2860, 2012.
[3] M.-H. Lee, Apply relay recording and video segment annotation for IPTV network personal video recorder, *IEEE Trans. Consumer Electronics*, vol.56, no.4, pp.2364-2372, 2010.
[4] F. Thouin and M. Coates, Equipment allocation in video-on-demand network deployments, *ACM Trans. Multimedia Computing, Communication, and Applications*, vol.5, no.1, pp.1-22, 2008.
[5] F. Thouin and M. Coates, Video-on-demand server selection and placement, *Proc. of the 20th International Tele-Traffic Conference on Managing Traffic Performance in Converged Networks*, pp.18-29, 2007.
[6] X. Zhou and C.-Z. Xu, Efficient algorithm of video replication and placement on a cluster of streaming servers, *Journal of Network and Computer Applications*, vol.30, no.2, pp.515-540, 2007.
[7] B. Ciciani, A. Santoro and P. Romano, Approximate analytical models for networked servers subject to MMPP arrival process, *Proc. of the 6th IEEE International Symposium on Network Computing and Applications*, pp.25-32, 2007.
[8] A.-E. Baert, V. Boudet, A. Jean-Marie and X. Roche, Minimization of download time variance in a distributed VOD system, *Scalable Computing: Practice and Experience*, vol.10, no.1, pp.75-86, 2009.
[9] B. Tan and L. Massoulie, Optimal content placement for peer-to-peer video-on-demand systems, *Proc. of IEEE INFOCOM*, pp.694-702, 2011.
[10] T. A. Neves, L. M. de A. Drummond, L. S. Ochi, C. Albuquerque and E. Uchoa, Solving replica placement and request distribution in content distribution networks, *Electronic Notes in Discrete Mathematics*, vol.36, pp.89-96, 2010.
[11] J. P. C. Blanc, *Queueing Models – Analytical and Numerical Methods*, Course 35M2C8, 2011.
[12] I. Adan and J. Resing, *Queueing Theory*, Department of Mathematics and Computing Science, Eindhoven University of Technology, The Netherlands, 2002.
[13] G. K. Zipf, *Human Behavior and the Principle of Least Efforts*, Addison-Wesley, Cambridge, MA, 1949.
[14] S. A. Chellouche, W. Aubry, D. Negru and Y. Chen, Home boxes support for an efficient video on demand distribution, *Proc. of 2011 IEEE International Conference on Multimedia and Expo (ICME)*, pp.1-6, 2011.
[15] J. P. Muñoz-Gea, S. Traverso and E. Leonardi, Modeling and evaluation of multisource streaming strategies in P2P VoD systems, *IEEE Trans. Consumer Electronics*, vol.58, no.4, pp.1202-1210, 2012.
[16] W. Wu and J. C. S. Lui, Exploring the optimal replication strategy in P2P-VoD systems: Characterization and evaluation, *Proc. of the 30th International Conference on Computer Communications (IEEE INFOCOM)*, pp.1206-1214, 2011.
[17] P. Mundur, R. Simon and A. Sood, End-to-end request handling in distributed video-on-demand systems, *Proc. of Communication Networks and Distributed Systems Conference*, pp.151-157, 1999.
[18] D. Villela and D. Rubenstein, A queuing analysis of server sharing collectives for content distribution, *Proc. of the 11th International Workshop on Quality of Service*, pp.41-58, 2003.
[19] H. Yu, D. Zheng, B. Y. Zhao and W. Zheng, Understanding user behavior in large-scale video-on-demand systems, *Proc. of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems*, pp.333-344, 2006.
[20] R. Bekker and A. Bruin, Time-dependent analysis for refused admissions in clinical wards, *Annals of Operations Research*, vol.178, no.1, pp.45-65, 2010.
[21] M. R. H. Mandjes and P. Zuraniewski, *M/G/infinity Transience, and Its Applications to Overload Detection*, Centrum Wiskunde & Informatica, Netherlands, 2009.
[22] *LINGO*, http://www.lindo.com/.
[23] A. Dan and D. Sitaram, An online video placement policy based on bandwidth to space ratio, *Proc. of ACM SIGMOD*, pp.376-385, 1995.

[24] M.-H. Lee, Estimation of worst-case bandwidth requirements of video-on-demand systems with replica servers using the M/G/∞ model, *Journal of Information Science and Engineering*, vol.30, no.5, pp.1365-1394, 2014.

[25] M. Ma, Z. Wang, K. Su and L. Sun, Understanding content placement strategies in smartrouter-based peer video CDN, *Proc. of the 26th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2016.