Video Server Deployment Using a Genetic Algorithm with Deterministic Initialization Strategy

Xin-Yi Hu, Yue-Jiao Gong(✉), Xin-Yuan Zhang, Yi-Ning Ma, and Jun Zhang

School of Computer Science and Engineering,
South China University of Technology, Guangzhou 510006, China
gongyuejiao@gmail.com, junzhang@ieee.org

Abstract. Video server deployment problem (VSDP) is a crucial problem in enhancing video viewing experience. It is difficult for traditional deterministic optimization algorithm to solve VSDP because it is non-polynomial hard (NP-hard). Current server deployment patterns are challenged by the growing scale of networks, and it is therefore desirable to design appropriate optimization algorithm to cope with this issue. Owing to the promising performance in solving NP-hard problems, a genetic algorithm (GA)-based optimizer is developed in this paper. We propose a novel deterministic initialization strategy to deploy the GA population in promising initial positions for performance enhancement. In addition, by embedding the maximum cost minimum cost flow (MCMF) in the fitness evaluation, the algorithm optimizes the server deployment and the bandwidth allocation concurrently. This way, the proposed GA is comprehensive, which deals with different aspects involved in VSDP. Experimental results and the comparisons with other algorithms indicate that the proposed algorithm achieves superior performance in terms of solution quality and convergence rate.

Keywords: Video server deployment · Genetic algorithm (GA)
Deterministic initialization strategy · Improved fitness evaluation function
Maximum cost minimum cost flow (MCMF)

1 Introduction

Video server deployment (VSD) determines the viewing experience of end users and the cost of service providers. For video service providers, the deployment and bandwidth rental cost is expected to be minimized with constraints of corresponding bandwidth requirements. As for end users, they focus on viewing experience, i.e., sufficient bandwidth allocation. As the growing number of Internet users, the networks scale cannot meet users' requirements. Additional video servers are expected to be imbedded in the existing network. Due to the resources limitation of available resources, it becomes an urgent issue of how to deploy video servers in an efficient and profitable way. More concretely, the optimization is conducted at video service provider side ensuring that the cost of video server deployment and bandwidth rental is as

T. Huang et al. (Eds.): ISNN 2018, LNCS 10878, pp. 459–467, 2018.
https://doi.org/10.1007/978-3-319-92537-0_53

low as possible. The optimization also takes viewing experience of user side into consideration. User side requirement serves as an essential constraint of the optimization process.

Recent work on server deployment can be classified into two categories: (1) order allocation of multiple resources by a single server, and (2) allocation of multiple servers. The former focuses on minimizing the delay by arranging the order of the resources allocated to the server. Tassiulas and Ephremides [1] propose a queueing model to solve it. When given identical arrival and service statistics of different queues, it maximizes the throughput and minimizes the delay by using an allocation policy. As for the allocation of multiple servers, Nakrani and Tovey [2] propose a distributed bee algorithm. Shanthikumar and Yao [3, 4] propose two methods of server allocation in multiple center manufacturing systems. They summarize the nature of closed-queuing networks, and distribute servers and traffic by these natures, not by actual network needs.

Recent years, more work on server allocation policies have been done. Cameron et al. [5] propose a high-density model for server allocation and placement, which is a random deployment of servers without the limitations of existing network topologies. Palmer and Mitrani [6] pose a method that optimizing cloud providers revenues via energy efficient server allocation. This approach takes into account the server and traffic distribution, reducing costs by minimizing the cost of power consumption. Pussep and Abboud [7] also put forward an adaptive server allocation for peer-assisted video-on-demand, which focusses on a connected P2P network where users' service demand can be partially satisfied by other users.

Today, most of the work in this area is after the allocation of the servers, such as ten cooling solutions to support high-density server deployment [8].

The Genetic Algorithm (GA) has a good global search ability, which can quickly search out the global solution in the solution space without falling into the trap of rapid decline of the local optimal solution. Moreover, by virtue of its inherent parallelism, it is easy to carry out distributed computing, speed up the solution speed.

In this paper, a GA with deterministic initialization strategy (GA-DIS) is developed according to the formulation of VSDP. A novel initialization strategy is proposed in order to assign the GA population to promising original positions. A special fitness function is finely designed to speed up the convergence rate. We also employ the maximum cost minimum cost flow (MCMF) in the fitness evaluation, so as to concurrently optimize the server deployment and the bandwidth allocation. Nine practical VSDP instances are adopted to test the performance of the proposed algorithm.

The paper is structured as follows: formulation of VSDP is presented in Sect. 2. In Sect. 3, we introduce the server allocation approach and the details of GA-DIS. Performance investigation is conducted in Sect. 4. Finally, conclusions are drawn in Sect. 5.

2 Model

Video server deployment problem (VSDP) is highly correlated with user's viewing experience. The topology of video server network can be described as follows: the network structure is modeled as an undirected graph including several network nodes

(e.g. routers, switches). Each node is connected to at least one other node via a network link. One node can transmit the received data over the network link to another connected node. The total bandwidth of each link is different to each other. The video transmission carried by each link charges the corresponding network rental fee according to the occupied bandwidth, while the rental fee per unit is different for each link. The total bandwidth occupied by a link cannot exceed the total bandwidth of the link.

Now we use a case to define the VSDP problem. Figure 1 shows a network topology; the number of nodes represents the node index. Red marked nodes are user nodes, the remaining nodes are potential server nodes that can be chosen to be VS. The link directly to the user's node is labeled $need_j$, indicates the video bandwidth consumption requirement of the jth user. Other links are marked (cap_i, $cost_i$), where cap_i indicates the link capacity limit for the ith link, $cost_i$ indicates the rental cost per unit of traffic on the ith link. VSDP is to select k nodes to deploy VS, in a network topology with m nodes, n users and e links, and meet the following requirements:

$$\min\left(k * deploymentCost + \sum cost_i * flow_i\right) \quad \text{s.t.} \begin{cases} flow_i < cap_i \\ \sum flow_j \geq need_j \end{cases} \tag{1}$$

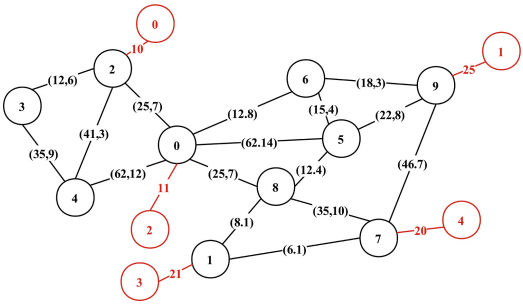

Fig. 1. A case of VSDP

Deployment scheme need to include not only the location of the nodes where the video content server is deployed, but also the network path between each consuming node and all video servers and the bandwidth consumed on the path.

3 Algorithm

We use a GA-DIS based algorithm to solve VSDP. After analysis of VSDP, we found that, the crux of the problem is to minimize the total VS deployment costs and bandwidth lease costs, while meeting users' traffic needs. At the same time, we found that, in a network where each edge has two attributes, capacity and cost, the maximum cost minimum cost flow (MCMF) can find a solution that maximizes the total flow

which flow from source A to destination B, while keeping the total cost minimized, by reasonably choosing the path and flow through each edge. It can be noted that the minimum cost maximum flow algorithm is well suited to solve VSDP.

The core idea of the algorithm is that GA-DIS randomly generate possible solutions to determine the location of VS. MCMF is used as the fitness function of GA-DIS, in order to determine the amount of bandwidth each VS allocates to each node. The fitness function is also used to calculate the minimum cost and maximum flow of the possible solution, to determine whether it is the optimal solution.

3.1 Deterministic Initialization Strategy

In GA-DIS, each individual contains an array of chromosomes, a cost value, and a bandwidth value. Among them, the chromosome array represents a feasible solution. We denote the *i*th chromosome as (2), where N is the total number of nodes in the network. If $x_{i,\ j}$ is 1, *j*th node is chosen to be a video server, else if $x_{i,\ j}$ is 0, there is no video server on *j*th node.

$$X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,N}) \tag{2}$$

By analyzing the experimental data, we find that the servers are tend to be deployed on the network nodes which are directly connected to the user nodes. Figure 2 plots the best solution of case1, which represents an extreme situation that only one VS is ordinary network nodes, and other VSs all deployed on those nodes which directly connected to the user points. In order to speed up the convergence rate, the first individual is given a promising position where all the servers are deployed on the network nodes which are directly connected to user nodes.

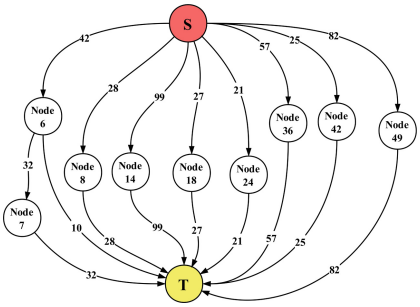

Fig. 2. The best solution result chart of case1

3.2 Preprocessing of the Input Map

As can be noted from Fig. 1, the number of VSs and user nodes are more than one. However, MCMF can only be applicable to single source point and single sink point graphs. Therefore, we transform the original network structure into a new one so that MCMF can be applicable to this scenario.

The transformation consists of three steps and is given as follows: First, build super source point S, super sink point T. Then, for each user node ID, establish a ID → T side. The cost of the side is 0, and the maximum capacity is the demand of this user node; for each VS node NID, establish a S → NID side. The cost of the side is 0, and the maximum capacity is the capacity of this VS node. After the transformation, user nodes and VS nodes can be regarded as ordinary nodes. The problem turns into finding the maximum cost minimum cost flow from S to T (Fig. 3).

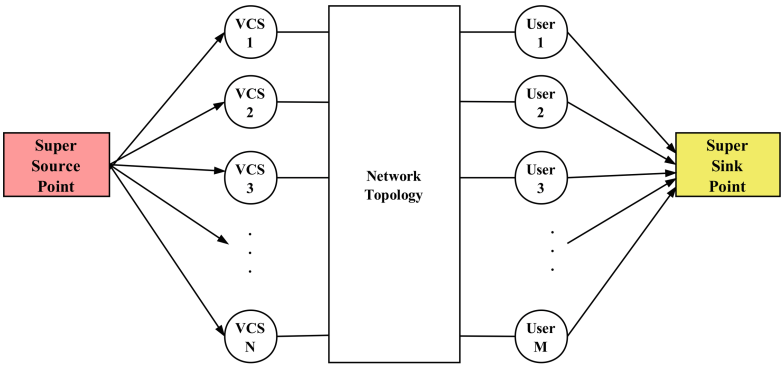

Fig. 3. Super source point and super sink point model

3.3 Design of Fitness Function

We use MCMF as the fitness function of GA-DIS. When using MCMF algorithm, we first find the minimum cost flow first, then try to get the maximum flow by expanding the chain. The specific steps can be seen in [9].

The first step of MCMF algorithm is to find the "shortest path", that is, the minimum cost flow. As Dijkstra algorithm is limited to the situation that the weight of the edge is not negative, the shortest path found by Dijkstra may be wrong. Bellman Ford can effectively find the "shortest path" in MCMF, but its efficiency is not high because of the excessive number of relaxation operations.

Therefore, we use SPFA, the improved algorithm for Bellman Ford. SPFA is based on the Bellman ford algorithm, but plus a queue optimization that reduces redundant relaxation operations. The idea is, since relaxation operations can only occur in nodes where "distance" changes, the scope for relaxation operations can be narrowed down, so we use queues record nodes to relaxation. The specific steps can be seen in [10].

We call the MCMF algorithm which use SPFA to find the "shortest path" as the fitness function1 (f_1). After that, we find a new way to replace SPFA to determine which path should be augmented, and the complexity of this new way is much lower. We call the MCMF algorithm with this new way as the fitness function2 (f_2).

Now we first analyze the principle of finding the shortest path algorithm. Take D_i be the shortest path from S to i. All the shortest path algorithm guarantees two conditions when algorithm ends.

$$\text{For any existing arc } (i, j), D_i + c_{i,j} \geq D_j \text{ is satisfied} \tag{3}$$

$$\text{There is at least one } i \text{ for each } j \text{ that can satisfied } D_i + c_{i,j} = D_j. \tag{4}$$

After the end of the algorithm, the edge just on the shortest path satisfies $D_j = D_i + c_{i,i}$. In the calculation of the minimum cost flow, we augment the path along $D_j = D_i + c_{i,j}$ each time, which will reduce flow and make some of the arc becomes no flow. This operation will not break (3) but may break (4). The usual cost-flow approach is to recalculate D for each augmentation, using SPFA, etc. This is undoubtedly a waste.

As in (3), $D_i + c_{i,j} \geq D_j$ can be translated into $D_i - D_j + c_{i,j} \geq, 0$. Also, in (4), $D_i + c_{i,j} = D_j$ can be translated into $D_i - D_j + c_{i,j} = 0$. For an identifier D, we can continue finding $D_i - D_j + c_{i,j} = 0$ augmented path by DFS. If DFS is failed, points collected by DFS are recorded as point sets V. Calculate

$$\Delta = \min\{D_i - D_j + c_{i,j}\} | i \in V, j \notin V, flow_{i,j} > 0 \tag{5}$$

Then for those points which is not in V,

$$D_i^{\pi} = D_i + \Delta \tag{6}$$

The new method only needs to augment, modify labels, but doesn't need BFS, queue, SPFA. As the new method saves the SPFA computing time, its complexity is very low.

4 Experiment

To thoroughly evaluate the performance of the proposed algorithm, we conducted a comparative experiment with five algorithms, such as BPSO-f_1, BPSO-DIS- f_1, GA- f_1, GA-DIS-f_1, GA-DIS-f_2, where A-f_1 and A-DIS represents the objective function of algorithm A is f_1 and initialization strategy of algorithm is the proposed deterministic initialization strategy. The number of iteration is set to 300, and the population size is fixed to 10. Table 1 shows the other parameter settings.

The main information of the nine test cases is shown in Table 2. These nine test cases can be divided into three dimensions according to the node number. The final results are shown in Table 3. The results show that GA-DIS-f_2 can get the best solution and run shorter time. The two GA models are more stable than the two BPSO models, while the two BPSO models run faster than the two GA models.

Table 1. Parameter configurations

Algorithm	Parameter configurations
BPSO-f_1	$\omega = 0.5 \sim 2$, $c_1 = 2 \sim 3$, $c_2 = 1 \sim 2$
BPSO-DIS-f_1	$\omega = 0.5 \sim 2$, $c_1 = 2 \sim 3$, $c_2 = 1 \sim 2$
GA-f_1	$pc = 0.7$, $pm = 0.07$
GA-DIS-f_1	$pc = 0.7$, $pm = 0.07$
GA-DIS-f_2	$pc = 0.8$, $pm = 0.09$

Table 2. Test cases information

Instance	Case	N	E_n	U_n
D1	C0	50	96	9
	C1	50	97	9
	C2	50	113	9
	C3	50	97	9
	C4	50	99	9
D2	C5	160	374	64
	C6	160	368	64
D3	C7	300	794	120
	C8	300	804	120

Here's some more explanation of the experimental results of case5–case8. The reason why BPSO-f_1, BPSO-DIS-f_1, GA-f_1, and GA-DIS-f_1 cannot get a solution is that SPFA is unable to solve the problem of cost flow with negative cycle. In a graph with n points, if a point gets into the queue more than n time, then there is a negative cycle in this graph. For instance, the source and sink points are isolated points, there is another negative cost, positive capacity circle. No matter what the initial label, simply by extension cannot eliminate this negative circle. Therefore, the SPFA algorithm will not terminate. In this case, the fitness function considers that the solution does not meet the user's needs and thus returns INF. So, the optimal solution is not available.

Table 3. Experiment results

Function		BPSO-f_1	BPSO-DIS-f_1	GA-f_1	GA-DIS-f_1	GA-DIS-f_2
Case 0	average time	7919.52	7803.96	8757.23	8106.04	5782.3
	minimum cost	2172	2172	2172	2092.6	2042
	average cost	2200.24	2109.45	2107.6	2172	2042
	maximum flow	303	303	303	303	303
Case 1	average time	8913.66	8739.42	9596.03	9035.33	6546.84
	minimum cost	2400	2400	2400	2400	2136
	average cost	2417.43	2411.1	2408.75	2405.66	2136
	maximum flow	382	382	382	382	382
Case 2	average time	9985.7	9539.3	11365.2	9720.8	7203.4
	minimum cost	1890	1890	1890	1890	1692
	average cost	1915.6	1910.2	1908.6	1894.8	1692
	maximum flow	431	431	431	431	431
Case 3	average time	14586.5	14501.9	15694.2	14589.7	13342.78
	minimum cost	2543	2543	2543	2543	2111
	average cost	2603.4	2593.1	2552.7	2547.86	2111
	maximum flow	340	340	340	340	340

(*continued*)

Table 3. (*continued*)

Function		BPSO-f_1	BPSO-DIS-f_1	GA-f_1	GA-DIS-f_1	GA-DIS-f_2
Case 4	average time	7846.4	7834.2	8727.3	8358.8	7433.8
	minimum cost	2160	2160	2160	2160	1967
	average cost	2270.4	2258.9	2263.7	2241.4	1967
	maximum flow	284	284	284	284	284
Case 5	average time	235056.7	269152.3	294624.8	253132.4	73045.3
	minimum cost	NA	NA	NA	NA	8961
	average cost	NA	NA	NA	NA	9323.7
	maximum flow	NA	NA	NA	NA	1975
Case 6	average time	255326.1	285910.7	269407.7	251439.2	68022.7
	minimum cost	NA	NA	NA	NA	9596
	average cost	NA	NA	NA	NA	9607.4
	maximum flow	NA	NA	NA	NA	1861
Case 7	average time	214724.8	247285.9	252467.6	253479.2	136392.5
	minimum cost	NA	NA	NA	NA	17974
	average cost	NA	NA	NA	NA	18992.9
	maximum flow	NA	NA	NA	NA	3698
Case 8	average time	263619.7	220637.4	212683.2	251239.4	145965.8
	minimum cost	NA	NA	NA	NA	19155
	average cost	NA	NA	NA	NA	19296.8
	maximum flow	NA	NA	NA	NA	3732

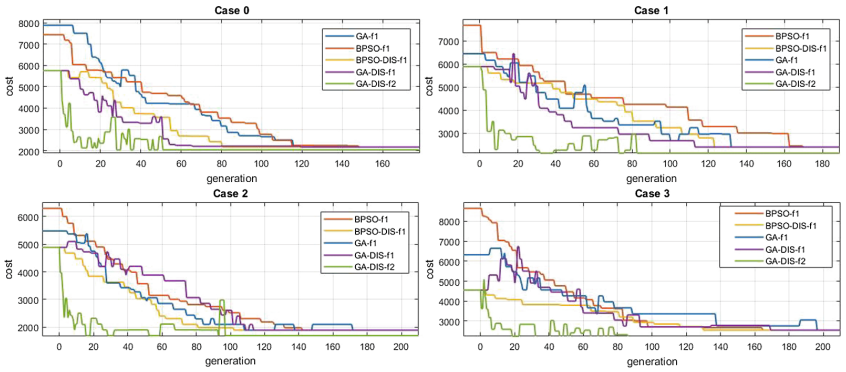

Fig. 4. Convergence curves of the five algorithms

The solve method in function2 is, forcing all negative cost side full flow, known as "backflow" operation. After doing so we destroyed the equilibrium conditions but satisfied the optimal conditions. Then we can seek feasible flow, and then seek the maximum flow. During this period, the optimal conditions have been maintained, the balance conditions and the maximum flow conditions could be gradually solved.

When comparing the convergence curves of the five algorithms, we find the reason why GA-DIS-f_2 can be the best one. For the three algorithms which has DIS, the DIS is equivalent to indicating a possible convergence direction when they initialize. From Fig. 4, it is easy to find that, BPSO-DIS-f_1, GA-DIS-f_1 and GA-DIS-f_2 start at the same number, while the other two algorithms may start from a number much larger.

5 Conclusions

In this paper, we give a rigorous analysis and formulation of VSDP. After that, we propose a GA based optimizer (GA-DIS) to cope with this problem. In order to speed up the convergence rate, a deterministic initialization strategy is proposed. A novel fitness function is designed to serve as a better guidance of the searching behavior. Experimental results indicate the effectiveness of the proposed deterministic initialization strategy and the new fitness function. GA-DIS shows superior performance in terms of solution quality and convergence rate.

References

1. Tassiulas, L., Ephremides, A.: Dynamic server allocation to parallel queues with randomly varying connectivity. IEEE Trans. Inf. Theory **39**, 466–478 (1993)
2. Nakrani, S., Tovey, C.: On honey bees and dynamic server allocation in internet hosting centers. Adapt. Behav. – Anim. Animat. Softw. Agents Robots Adapt. Syst. Arch. **12**, 223–240 (2004)
3. Shanthikumar, J.G., Yao, D.D.: On server allocation in multiple center manufacturing systems. Oper. Res. **36**, 333–342 (1988)
4. Shanthikumar, J.G., Yao, D.D.: Optimal server allocation in a system of multi-server stations. Manag. Sci. **33**, 1173–1180 (1987)
5. Cameron, C.W., Low, S.H., Wei, D.X.: High-density model for server allocation and placement. ACM SIGMETRICS Perform. Eval. Rev. – Meas. Model. Comput. Syst. **30**, 152–159 (2002)
6. Palmer, J., Mitrani, I.: Optimizing cloud providers revenues via energy efficient server allocation. Sustain. Comput.: Inform. Syst. **2**, 1–12 (2005)
7. Pussep, K., Abboud, O.: Adaptive server allocation for peer-assisted video-on-demand. In: IEEE International Symposium on Parallel & Distributed Processing, pp. 1–8. IEEE Press, New York (2010)
8. Hannaford, P.: Ten cooling solutions to support high-density server deployment. APC white paper #42 (2006)
9. Smith, D.K.: Network flows theory, algorithms, and applications. J. Oper. Res. Soc. **45**, 1340 (1994)
10. Duan, F.: A faster algorithm for the shortest-path problem called SPFA. J. Southwest Jiaotong Univ. **29**, 207–212 (1994)