Computers and Electrical Engineering 000 (2017) 1–13

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Towards an improved heuristic genetic algorithm for static content delivery in cloud storage☆

Zhigao Zheng[a,1], Zunxin Zheng[b,2,*]

[a] Luoyu Road 1037, Wuhan, Hubei, 430074, China
[b] Nanhai Ave 3688, Shenzhen, Guangdong, 518060, China

ARTICLE INFO

Article history:
Received 5 January 2017
Revised 8 June 2017
Accepted 11 June 2017
Available online xxx

MSC:
00-01
99-00

Keywords:
Cloud storage
Static content delivery
Resource management model
Cost model
Genetic algorithm

ABSTRACT

A key challenge in computer networking is how to organize network topology effectively among a large number of servers in the cloud storage system. In a cloud environment, the topology, which is different from the underlying topology, may be established in any form at any potential edge peers. The cloud content delivery network (CDN) always faces problems of complex distributed path creation, cache update, load balancing, etc. To address the problem as a static content delivery, we propose an Improved Heuristic Genetic Algorithm for Static Content Delivery in Cloud Storage (IHGA-SCDCS) based on a resource management model and cost model. The static content delivery in cloud storage is abstracted into mathematical model for set solving problem, which is then solved by an improved Genetic Algorithm (GA). Finally, the optimal solution is reduced to an optimal content delivery program. The simulation experiment, based on CloudSim, shows that IHGA-SCDCS can effectively obtain optimal solution while reducing delivery cost.

1. Introduction

Reasonable network topology and resource management model can not only improve network performance but also guarantee effectiveness and load balancing for resource allocation, thus improving the performance of cloud storage services. Cloud storage is an Internet service itself, which emphasizes cloud data center providing resource while weakening hardware and software capabilities of terminals [1]. With cloud storage service, all content needed is kept in distance data centers and users access it via network. Thus, the cloud storage service may also use CDN in an accelerated manner, like other Internet services, to achieve higher access efficiency and better user experience [2]. The cloud CDN reduces its own cost using competitive price provided by different cloud. Combined with on-demand service of cloud, the cloud CDN can easily adjust its own storage and bandwidth usage according to the requirement. It may also reduce cost by reducing the quality of service.

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Associate Editor Dr. R. Varatharajan.

* Corresponding author.
E-mail addresses: zhengzhigao@hust.edu.cn (Z. Zheng), zxzheng@szu.edu.cn (Z. Zheng).
[1] Zhigao Zheng is with Services Computing Technology and System Lab/Cluster and Grid Computing Lab/Big Data Technology and System Lab, School of Computer Science and Technology, Huazhong University of Science and Technology.
[2] Zunxin Zheng is with College of Economics, Shenzhen University. Dr. Zunxin Zheng is a Professor in Finance and he is the Vice Dean of College of Economics

http://dx.doi.org/10.1016/j.compeleceng.2017.06.011
0045-7906/

Many scholars have made preliminary studies on CDN programs based on cloud storage. The most representative one is MetaCDN proposed by Broberg et al.[3], which is a low-cost CDN using a storage cloud resource. The system provides a mechanism to place content on network provided by a different storage cloud service supplier and regularly replies appropriate copy answering requests from users. However, the system does not incorporate a new caching strategy and load balancing algorithm. The F_cache content acceleration cache technique developed by FastWeb just caches and finds optimal match on web acceleration content according to strategy. Within a period of time, it does not access file entity from the source website for repeated access but rather it copies content directly from cache to users, thus effectively improving response and saving bandwidth [4]. The cost of cloud CDN includes bandwidth and storage cost. Nevertheless, existing content delivery methods have not arrived at a reasonable solution to a pricing mechanism of cloud CDN. Meanwhile, the network topology of cloud storage is quite different from that of traditional CDN [5]. To operate cloud CDN, it is necessary to study an efficient content delivery program and reasonable load balancing strategy in cloud storage. The CDN is a whole system made up of four elements: content delivery, load balancing, content management and distributed storage after strategic analysis and implementation. The content delivery strategy is one of the key factors in CDN network planning, whose design directly determines whether the core idea as nearest service of CDN can be realized. In accordance with nearest service principle and edge server load balancing strategy, CDN ensures providing service for resource requests from users with an extremely efficient way.

The cost model in existing content delivery technologies has been developed to include one or more types of costs as download, storage and upload. As to reducing cost in content retrieval, research has shown that replica placement in general network topology is an NP-complete problem [6,7] and the optimal solution for tree topology has also been determined. Some heuristic algorithms were evaluated in [8] to find a greedy algorithm providing optimal performance. A heuristic content delivery algorithm based on fan-out was proposed by Radoslav et al. [9] and Jamin et al. [10]. Except for retrieval cost, the upload cost was further added [11]. The cost for storage was also supplemented [12,13]. In addition, the fee for retrieval, upload and storage were comprehensively considered in [14], which also provided a solution for tree topology. Other research shifted focus to adding service quality requiring all user requests to arrive at the edge server within certain network distance. The algorithm for optimizing overall storage and update cost was brought out in [15], where it is assumed that the request starts from any peer and the retrieval cost is ignored. In [16], the limitation on server capacity was added while simultaneously optimizing storage and retrieval cost.

The essence of the above solutions is static delivery. Some methods assume requests initiated by all peers evenly. Some algorithms use past request modes to customize delivery strategies. In [17], the delivery strategy was modeled as a Markov decision process and a centralized heuristic algorithm was proposed. In [18], the distributed heuristic algorithm was investigated further. The content delivery and traffic redirection was optimized in [19] to achieve request load balancing in the content delivery process while the problem of transferring a group of cached copy was solved in [20].

The traditional content delivery technologies in CDN have been widely studied. However, existing results cannot directly apply to cloud storage CDN in that many researchers in the past assumed network topology is provided as tree stored in the source server. In the present cloud environment, it is possible to establish any topology among all potential edge cloud storage peers, which may be different from the underlying network topology. Therefore, content delivery has complex problems of distributed path building, cache update and load balancing in cloud CDN. Additionally, the edge is usually undirected in traditional CDN. However, the cost for upload and download in cloud storage is different and needs a directional edge. This means that it is not enough to consider a pricing mechanism in a single direction, which asks for more reasonable pricing strategies.

Load balancing is an integral part of content delivery technology. Any content delivery algorithm should determine an appropriate load balancing strategy. Once the content delivery is completed, the user requests are redirected to a corresponding edge server using random methods in CDN like Uniform Resource Locator (URL) rewriting or transparent interception of requests to implement reasonable uniform delivery of users' requests.

To address static content delivery problem in CDN, the paper provides a resource management model based on a union tree and delivery cost model. An improved heuristic genetic algorithm for static content delivery in cloud storage is proposed. The rest of the paper is organized as follows: Section 2 designs the resource management model and cost model of CDN. In Section 3, the improved static content delivery algorithm is put forward. In Section 4, we carry out simulation experiments and compare the performance of the proposed algorithm with CloudSim simulator. We conclude this paper in Section 5.

2. Resource management model and content delivery cost model

2.1. Tree resource management model

Traditional cloud storage system manages data and resource with central indexing. The centralized management has a simple structure and is easy to design and management. Combining existing proxy mechanisms developed in [21], we build a tree resource management model based on P2P.

The tree structural model of cloud storage is a hierarchical structure that divides peers into several regions corresponding to the physical distance among peers, which are managed by region center peers [22]. The whole network is made up of center peers and normal peers. To improve the efficiency of cloud storage services, the CDN technology can be used among

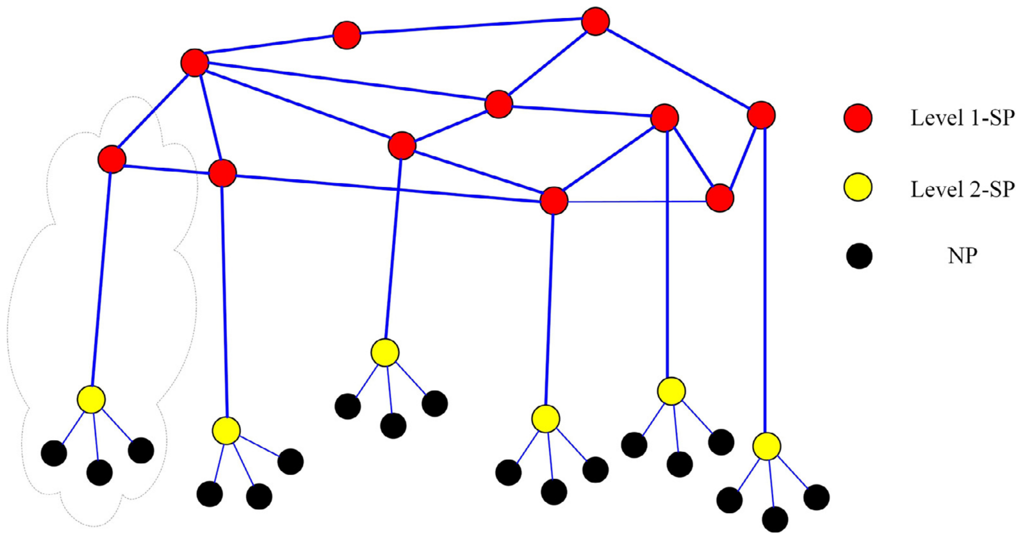

Fig. 1. Tree resource management model.

different regions. In this way, the content delivery function of CDN can be used in case of resource access across regions, thus improving access efficiency of cloud storage to the maximum extent [23]. The P2P tree resource management model designed in the paper just improves the traditional tree structure, which has similar structural features to the Internet, as shown in Fig. 1. Based on network bandwidth and processing capability, the peers in the network are classified into Super Peer (SP) and Normal Peer (NP). The NP just shares its own resource and accesses other resources in the system. The SP is undertaken by peers online for a long time and has a higher bandwidth and better performance. Except to access other resources in the system and share its own resource, the SP also maintains normal system operation management. The SPs are further divided into Level 1-SP and Level 2-SP according to their performance level. The SP is responsible for collecting and keeping resource load of peers within a local region and then allocating and managing resource in accordance with current peer load status.

The P2P tree resource management model firstly divides peers into several regions based on physical distance among peers. All peers within the region are physically close to each other. If the queried resource can be found within a local region, it can reduce network delay and bandwidth consumption considerably. Secondly, select a peer within each region as Level 2-SP. In the centralized manner, other NPs just connect directly with Level 2-SP. This centralized management enables efficient resource location. Thirdly, the region is managed by one Level 1-SP, which connects to all Level 2-SPs within this region. This peer is responsible for region management and forward resource location messages as well as response requests from peers within the region. Finally, all SPs that manage each region are connected to achieve P2P tree resource management model. The management model combines both feature of P2P and C/S structure. The resource location message is managed by centralized and distributed methods. It locates message forwarding using a parent-child relationship, namely the Level 2-SPs centralized manages messages within the region. On the other hand, the Level 1-SPs broadcast requests. If the resource is not located within the region, it needs a message delivery between Level 2-SP and Level 1-SP, as well as that among Level 1-SPs, to locate other regions.

2.2. Cloud storage content delivery process

Taking a scene of dynamic content delivery as example, the content delivery process in cloud storage is shown in Fig. 2. Assume the *NP N*2 are source server. The so-called source server is the original storage location of resource. The user is allocated to the closest region by communicating with load balance Domain Name System (DNS). It obtains the address of Level 1-SP in this region. From the resource meta-data list in the Level 1-SP, the user inquires whether there is copy to request resource. If so, it obtains the address of the peer that store copy from the list. By communicating with it, the user can access to resource content. If there is no copy to request resource within the region, the Level 1-SP send request to Level 1-SP where source server located by parse URL of user to access its address. Then, it selects some server within the region to send resource request to source server, which will delivery content to local server to provide resource service.

2.3. Content delivery cost model

Cost model is an important index that determines content delivery technology in the cloud storage. The paper brings out content delivery cost model combining with CDN delivery process and cloud storage charging mechanism. The static content delivery model shown in Fig. 3 and Fig. 4 is used as example. Assume each peer has a path to a random user. Then users can

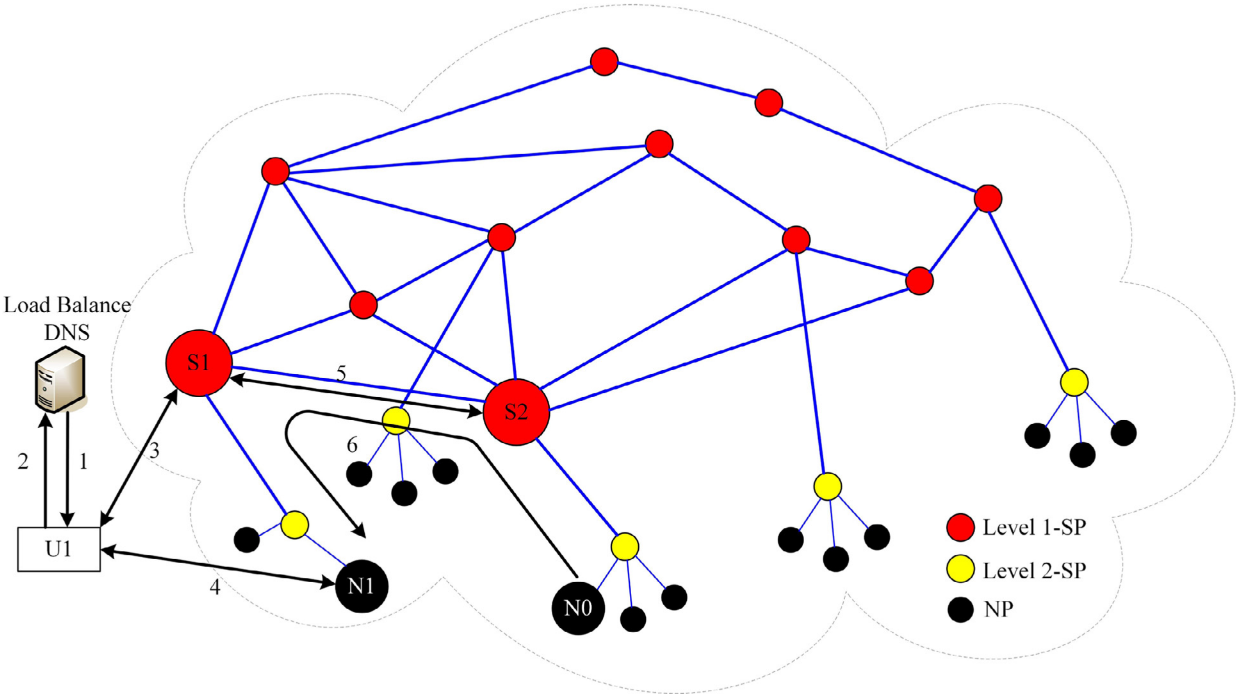

Fig. 2. Dynamic content delivery process.

access a close SP that responses to its service request by communicating with load balance DNS. In the delivery program, S_1 is responsible to deal with service requests from U_1 and U_2 and S_2 for service requests from U_2. The S_4 is used to deal with requests from U_4 and to forward resource copy from source server N_0 to the region server where S_1 and S_5 located. The total cost of this program includes the input, download and storage costs of normal copy peers within the region of Level 1-SP S_1, S_2, S_3, S_4 and S_5. As to a normal copy peer that provides resource download service for users, its input cost comes from the input traffic generated by its own request for resource to source server, while the download cost comes from the output traffic generated by providing resource download service for users. The resource server N_0 is regarded as a NP in the cloud storage. Its upload cost comes from the upload traffic generated from content supplier to cloud storage, while the download cost comes from the output traffic generated by content delivery from N_0 to N_1. As to an SP that forwards copy to other peers, its download cost also comes from the output traffic generated from forwarding copies to other SP_s. The storage cost comes from storage of copies, including storage on source server N_0 and all copy peers.

Thus, the content delivery cost model in the cloud storage can be obtained as follows. The Level 1-SPs in cloud storage are $S = S_1, S_2, \ldots, S_n$ and end-users are $U = U_1, U_2, \ldots, U_m$. The size of copy is W. The cost for storage on a cloud peer is C_j per G byte. The cost for output of peer j is D_j per G byte. The cost for input of peer j is P_j per G byte. The cost for content forwarding of peer u and peer v is V_{uv}.

If U is source server and V is Level 1-SP in the cloud storage, the computation method of the cost in forwarding content between U and V is given by formula 1 below.

$$V_{uv} = (C_u + D_u + D_v)W \tag{1}$$

If U and V are Level 1-SPs in the cloud, the cost for content delivery between U and V is given by formula 2 below.

$$V_{uv} = (D_u + P_v)W \tag{2}$$

If V is a cloud storage peer and U is an end-user, the cost for content delivery between U and V is given by formula 3 below.

$$V_{uv} = (C_u + D_u)W \tag{3}$$

After the above variables have been defined, the optimized objective function can be obtained by formula below. 4.

$$min \sum_{(u,v)\in E} y_{uv}V_{uv}, \tag{4}$$

where y_{uv} indicates whether to establish a delivery path between u and v.

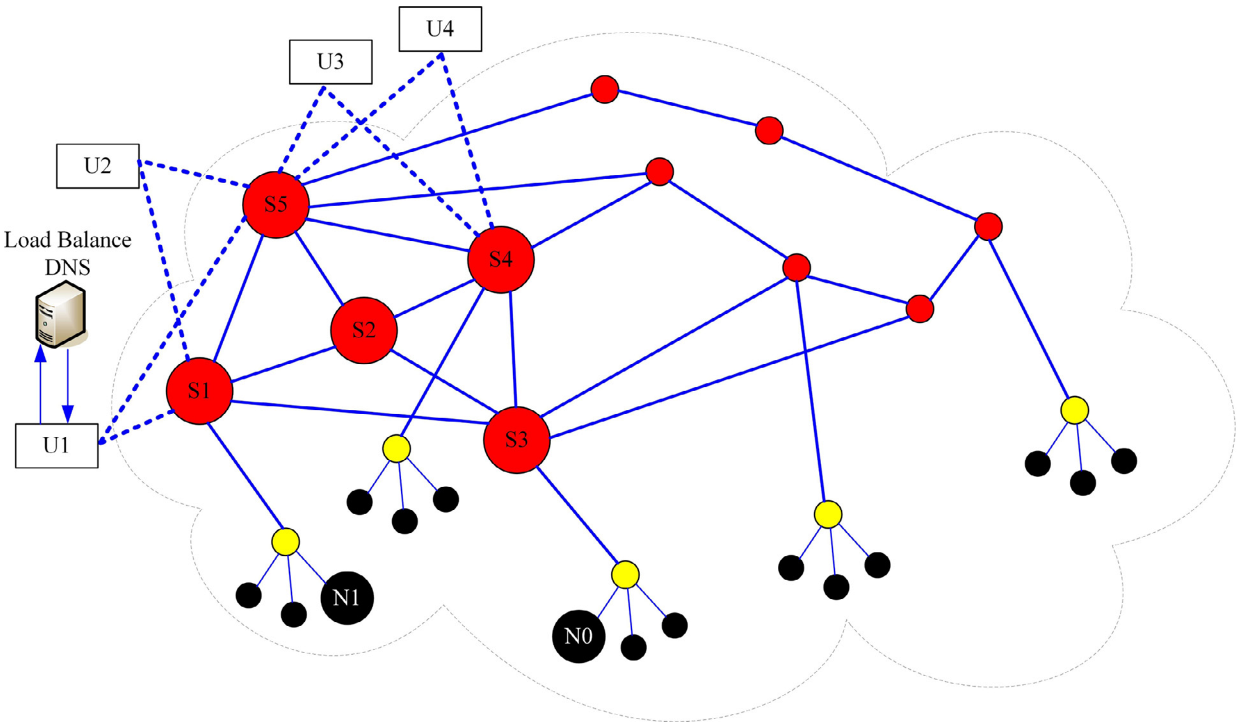

Fig. 3. User membership.

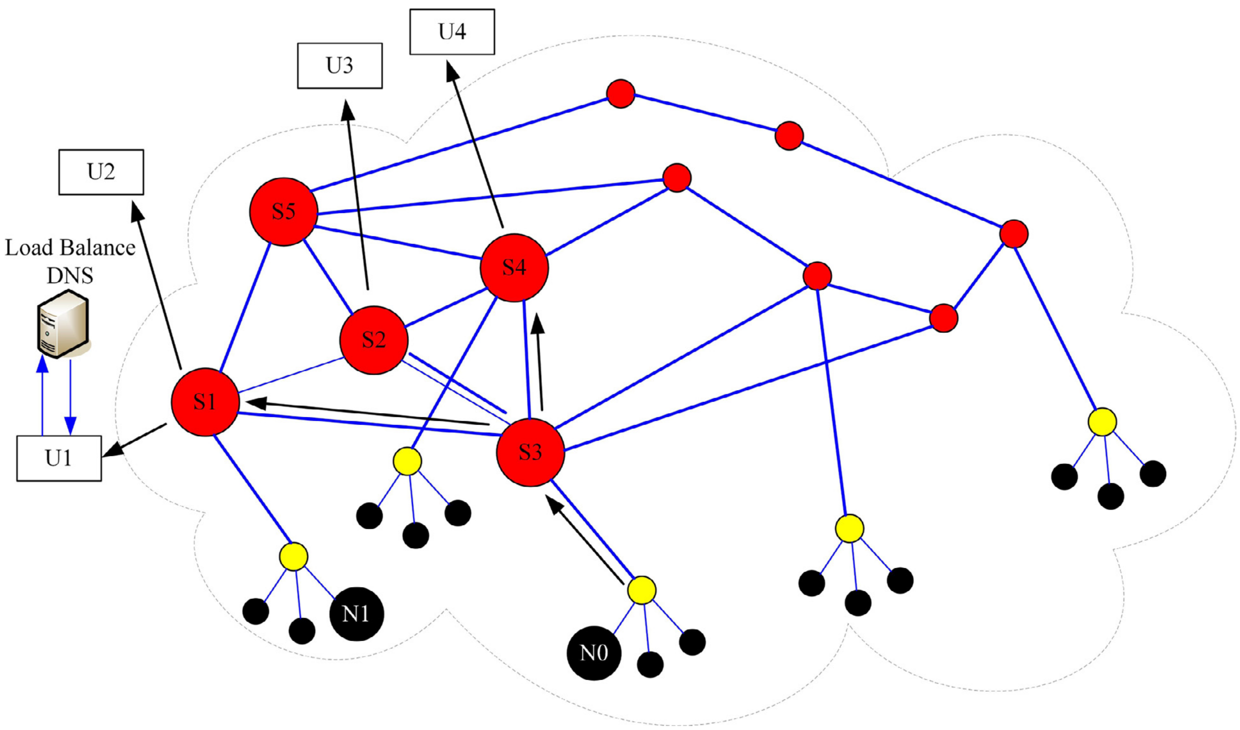

Fig. 4. Content delivery program.

3. Improved static content delivery algorithm

The static content delivery in cloud storage is a Set Covering Problem (SCP) which can be like the following solving process: there are different types of goods that are not sold separately. According to different collocation, the goods are packaged into different packages with different prices. To purchase all types of goods, it is easy to find a solution to purchase all packages. However, it is difficult to find the solution to purchase minimum packages at least cost, which is an NP-hard problem or weighted SCP. The SCP has been proved to be an NP-hard problem [24]; namely, the polynomial time complexity cannot be solved. The local optimal phenomenon also introduces a big challenge for NP-hard problem solving. Currently, many researchers have proposed some heuristic algorithms to solve this problem. For example, Bersley et al. [25] put forward a genetic algorithm-based heuristic for non-unicast set covering problem (GA-heuristic) to solve SCP without weight. Liang et al. [26] improved GA on individuals in the population heuristics and adjusted genetic parameters appropriately to propose genetic algorithm for weighted set covering problem (GA-WSCP), developing a new approach for SCP. Liang et al. [26] established a programming model of emergency service location problem. The improved GA was used to solve such SCP. However, the above algorithms are not very effective in solving large-scale SCP and not easy to obtain global optimal solution, which cannot properly apply to cloud storage static content delivery characterized by mass data. Therefore, it is necessary to study and design cloud storage static content delivery algorithm capable of solving large scale SCP.

The performance of existing static content delivery algorithms to solve large scale SCPs is not satisfactory. Therefore, the paper proposes an improved GA to solve the SCP of static content delivery in cloud storage. It has significant improvements in aspects of maintaining population diversity, avoiding local optimum and enhancing solving.

3.1. Mathematical model

Set $A=(a_{ij})_{m\times n}$ as a $0-1$ matrix with m rows and n columns. A row in A corresponds to users of static content delivery network topology in the cloud storage and so the row set represents the set of total users in the network topology, which is denoted as $i\in M$, $M=1,2,\ldots,m$. A column corresponds to Level 1-SPs on the edge of network topology; the column set is expressed as $j\in N$, $N=1,2,\ldots,n$. The cost of column is $C=(c_j),\ j\in N$, where, c_j is the cost of column j, i.e., the cost for content delivery to the edge Level 1-SPs denoted by column j, $c_j>0$, $j\in N$. If $a_{ij}=1$, it indicates the row i covered by column j, i.e., the end-users expressed by row i provided services by edge Level 1-SPs denoted by column j. The equation $a_{ij}=0$ means row i has not been covered by column j, i.e., end-users of row i are provided services by edge Level 1-SPs denoted by column j. As the target of static content delivery in cloud storage is to deliver content to appropriate edge Level 1-SPs, so all end-users are directly provided services by edge Level 1-SPs and reduce delivery cost as possible. Therefore, the abstracted SCP in the paper should arrive at an optimal solution $X(X\subseteq N)$. It is the set of a group of columns, i.e., a static content delivery program so that each row in M can be covered by one column in X at least. It means that each end-user will be served by at least one edge Level 1-SP. Furthermore, the summation of cost in solution X should be the minimum, i.e., minimizing the delivery cost. The abstracted SCP mathematical model follows from formula (5).

$$\begin{aligned} &min\sum_{j=1}^{n}c_jx_j \\ &\qquad\qquad s.t.\sum_{j=1}^{n}a_{ij}x_j\geq 1, i=1,2,\ldots,m. \\ &\qquad\qquad x_j\in 0,1, j=1,2,\ldots,n. \end{aligned} \tag{5}$$

The objective formula (5) gives the minimized total delivery cost. The constraint condition

$$\sum_{j=1}^{n}a_{ij}x_j\geq 1, i=1,2,\ldots,m,$$

means each end-user is covered by one edge Level 1-SP at least. In the condition $x_j\in 0,1,\ j=1,2,\ldots,n$, $x_j=1$ means the column j is included in solution X, i.e., delivery content to edge Level 1-SP denoted by column j. If $x_j=0$, it indicates that the column j is not included in X, i.e., it does not deliver content to peers represented by column j. If all $c_j(j\in N)$ are the same, then it is the set covering problem without weight. If all $c_j(j\in N)$ are not the same, then it is the weighted SCP. The weight c_j corresponds to the cost of content delivery at edge Level 1-SP denoted by column j in the cloud storage.

3.2. Improved algorithm

Due to the advantages of GA, it has been widely applied in the fields of computer science, social science and engineering [26,27]. However, traditional GA has a slow convergence speed in solving large-scale set covering problems like static content delivery in cloud storage. The solution is not ideal and it is likely to converge to a local optimum. Based on abstracted mathematical model of static content delivery, we propose the algorithm IHGA-SCDCS, which introduces heuristic initial population generation method. The linear transformation idea is introduced to enlarge result difference in fitness function

Please cite this article as: Z. Zheng, Z. Zheng, Towards an improved heuristic genetic algorithm for static content delivery in cloud storage, Computers and Electrical Engineering (2017), http://dx.doi.org/10.1016/j.compeleceng.2017.06.011

to optimize parent selection. The repair operation is introduced and the repeat operation is modified to increase population diversity to avoid local optimization. The heuristic multi-point crossover selection is used to save high-quality genes as much as possible. Meanwhile, a new mutation operator is put forward to determine sudden probability in accordance with individual fitness.

3.2.1. Individual encoding

The static content delivery is essentially an SCP. Firstly, the individual encoding manner in the population should be determined. Here, the binary encoding method is used to represent the individual in the form of a vector S. For example, $S = [1, 1, 0, 0, 1, 1, 0, 0]$ is an individual that has 7 genes, which corresponds to the 1st, 2nd, 5th and 6th column in the current solution. It represents a delivery program corresponding to static content delivery model in cloud storage. In other words, there are 7 potential edge Level 1-SPs for delivery but only the 1st, 2nd, 5th and 6th edge Level 1-SPs are selected.

3.2.2. Initial population generation

The initial population is generated by heuristic algorithm and random algorithm corporately. It can accelerate convergence and reduce operation time to generate half initial population. Another half population is generated randomly, so as to ensure population diversity. The size of initial population is R_s. Because of static content delivery, the population size is fixed.

Each individual S in the initial population is an initial solution. Assume $B(i)$ is the set of all columns covering row $i(i \in M)$, namely all potential edge Level 1-SPs that provide direct services for user i. The $D(j)$ is set of all rows covered by column $j(j \in N)$, namely the set of all potential users that may serve by edge Level 1-SPs. The X is set of columns in current solution, namely edge Level 1-SPs needed to delivery content in current program. The U is set of rows not been covered by current solution, namely the users not been covered by edge Level 1-SPs in current solution. The $W(i)$ is number of columns that can cover row $i(i \in M)$ in the solution X, namely the number of edge Level 1-SP server covering user i in the current plan. The random generation method uses existing random function. Here gives a specific process generated by heuristic algorithm as Algorithm 1.

Algorithm 1 The proposed IHGA-SCDCS algorithm.

1: Preparation. Set current X as empty and $W(i) = 0$, $\forall i \in M$. Assume an individual S is the zero vector with 1 row and n columns.
2: **while** $N \neq 0$ **do**
3: Perform subsequent operations on each row in M. Randomly select one column j from $B(i)$ and add it into current solution X, namely $X = X + j$. Then, let $W(i) = W(i) + 1$ to $\forall i \in D(j)$.
4: To $\forall j \in X$, if $\forall i \in D(j)$, there is $W(i) \geq 2$, namely each row i is covered by two columns at least, let $X = X - (j)$. Then, to $\forall i \in D(j)$, let $W(i) = W(i) - 1$.
5: **end while**
6: Set value of the j^{th} bit in individual vector S as 1 to each column j in X.

1

3.2.3. Individual fitness computation

As to fitness function selection, the objective function is usually to evaluate fitness of individuals. However, the objective function in the algorithm takes the minimum, so we should firstly define fitness P_1 based on objective function shown as formula 6.

$$p_1 = \begin{cases} p'_{max} - \psi p, & Individual\ match\ condition\ in\ 5 \\ 0, & Not\ match\ consist\ condition. \end{cases} \tag{6}$$

where, $p = \sum_{j=1}^{n} c_j x_j$ is the function body in the objective function, which refers to cost of current individual on behalf of delivery cost. The $p'_{\max}$ takes a random value larger than the maximum value of f. The individual with a larger P_1 has better fitness. Thus, the fitness function P_1 can better evaluate an individuals fitness. The algorithm uses fitness proportional selection method in the selection phase, namely roulette method. It may encounter problems when there are many poor individual in the population, so it is difficult to select better individuals. In order to enlarge differences among individuals and highlight advantages, the idea of GA is to use linear transformation to define the fitness function P_2 in the selection phase as shown in Eq. (7) below.

$$p_2 = \begin{cases} 1 - 1.2 \times \frac{p_{avg}}{p_{max}+p_{min}}, & 0 < \frac{p_{avg}}{p_{max}+p_{min}} < 0.5 \\ 0.4, & else \end{cases} \tag{7}$$

where $p_{\max}$, $p_{\min}$ and p_{avg} are maximum, minimum and average of fitness P_1 of this individual in the current population. In the algorithm execution process, if the value of $p_{avg}/(p_{\max} + p_{\min})$ is less than 0.5, it indicates the fitness of most individuals in the population is less than the average value, and so it should improve the value of P_2 properly and introduce new child generation to improve the fitness of individuals. If the value is larger than 0.5, it means the individuals in the population are reasonable in accordance with distribution of fitness. At this moment, it should properly reduce the value of P_2 to effectively slow down individual loss. However, it should also avoid the algorithm converging too early to ensure the value of P_2 is large enough. The probability in formula 7 can meet the above requirements and ensure the value in the range 0.4 to 0.99, which is the advised probability in traditional GA.

3.2.4. Repair operation

The individual whose fitness is zero will not produce a next generation, resulting in part gene loss, which is not conducive to a preservation of population diversity. Referring to algorithm in [26], the repair operation is used to change a non-feasible solution S into a feasible solution. Here are steps of the repair operation:

Step 1 Calculate the covering status of all rows. As to the ith row, $W(i) = \{|X \bigcap B(i)|,\ \forall i \in M\}$.
Step 2 Statistics row set not covered by and column, denoted as $U = \{i | W(i) = 0,\ \forall i \in M\}$.
Step 3 As to all row in U, perform subsequent operations in the row order. Firstly, find column j with minimum $c_j / |U \bigcap D(j)|$ in $B(i)$. Namely, find rows not been covered as possible from the set of columns denoted row number i. At the same time, the cost of column j is relatively smaller. The goal is to find column the highest potential performance-cost ratio. Secondly, add column j with highest performance-cost ratio to current solution X. Let $X = X + j$ and $w(i) = W(i) + 1$ $(\forall i \in D(j))$. It means remove uncovered rows before from column j. At least, adjust gene of individual S in accordance with change of X, namely set $S(j) = 1$. In this way, S changes from non-feasible solution to feasible one.

3.2.5. Repeat individual modification

In the iterative process of GA, a wide variety of individuals is conductive for generation of an optimal solution. The generation of repeat individuals not only occupies population resource but also reduces population diversity. If the repeat individuals are directly removed, the population size will reduce, thus affecting the convergence result of the algorithm. In this paper, we modify repeat individuals heuristically, thus keeping population numbers and strengthening its local searching. The specific steps of the modification are as follows.

Step 1 As to all columns corresponding to repeat individual S in the current solution X, conduct subsequent operations in accordance with column order. Firstly, if there is $W(i) \geq 2$ to each row i covered by column j, it means it is at least covered by one column. Set $X = X - j$ and $w(i) = W(i) - 1,\ (\forall i \in D(j))$. Finally, set $S(j) = 0$ to delete redundant columns.
Step 2 If Step 1 does not achieve ideal modification effect, randomly delete certain columns corresponding to repeated individual S from current X. The operation of each column refers to Step 1, we can use adjustable parameter W_u to represent number of deleted columns.

3.2.6. Selection operation

The paper adapts a fitness proportional selection method, namely the roulette method. The probability of an individual being selected equals the ratio of its fitness to the total fitness of the population. Here we use the fitness function P_2 to represent individual fitness; the selection probability is then given by $p_2(i) / \sum_i^{R_s} p_2(i)$.

3.2.7. Crossover operation

Based on existing multi-point crossover, the paper further proposes heuristic multiple point crossover method. The method can save quality gene fragment and reduce occurrence of repeated individuals effectively, thus optimizing searching in the solution space. Assume two parent individuals to be carried out with crossover operation are denoted as E and F. The fitness function P_1 is used to compute their fitness p_E and p_F, respectively. The gene length of individual chromosome is L; crossover point number as N_u; child individual as G. Here give specific steps for crossover operation.

Step 1 Generate a group of random integer Y different from each other from 2 to L/k, the total number of which is V_u. Where, k is the adjustable parameter. Each one represents the cross-point location on the chromosome. There are in total V_u cross points. Thus, the chromosome of a parent individual is divided into $V_u + 1$ gene fragments. Assume the ith gene fragment in a parent individual E is represented as $E(i)$, where, $1 \leq i \leq V_u + 1$. The $E(1)$ is the gene fragment between first parent individual E and first cross-point, while $F(V_u + 1)$ is the gene fragment between F chromosome and the N_uth cross-point.
Step 2 For each random number i in Y, if $E(i) = F(i)$, two gene fragments are the same and the gene fragment of parent individual takes random location of a corresponding one on the chromosome of child individual G, i.e., $G(i) = E(i) = F(i)$. If $E(i) \neq F(i)$, randomly generate a number r between 0 and 1. If $r > Q$, the gene fragment on the corresponding location of the child individual G chromosome is $G(i) = F(i)$. If $r < Q$, the gene fragment on the corresponding location is $G(i) = E(i)$. The variable Q is defined to adjust the probability that parent generation brings to next generation and it is given by $Q = 0.5 + (p_F - p_E)/(p_E + p_F)$. We can see that the higher the fitness of the parent generation, the more likely it brings its genes to an offspring.

Table 1
Experiment result comparison.

Example	Scale	Cost of user greedy algorithm	Cost of GA-heuristic	Cost of IHGA-SCDCSc
Scp39	1000200	838.9	584.5	567.5
Scp40	1000200	854.2	575.8	566.0
Scp41	1000200	854.2	575.6	575.6
Scp42	1000200	854.2	566.8	566.8
Scp43	1000200	854.2	566.0	566.0
Scp45	3000300	963.9	624.5	594.6
Scpb1	3000300	424.2	296.0	280.8
Scpb2	3000300	428.1	298.8	298.8
Scpb3	3000300	454.2	324.0	324.0
Scpb4	3000300	443.9	354.5	341.6

3.2.8. Mutation operation

A common mutation operator is to randomly reverse gene on chromosome per a fixed mutation probability. As the actual mutation probability is generally small, the mutation has only little contribution to ensure population diversity and enlarge searching space. Therefore, we introduce an adaptive mutation operator with variable mutation probability based on individual fitness. The mutation probability f_m of individual E is defined as formula (8).

$$f_m = mg \times (1 - p_E) \tag{8}$$

where p_E can be obtained from the fitness function P_2; mg is the adjustable parameter. It can be seen that the individual with lower fitness has a larger mutation probability, which benefits population diversity and global optimization. The specific operation is as follows.

Step 1 Compute mutation probability of individual E with formula (8).
Step 2 Generate a random number with uniformly distribution within interval [0, 1] for each gene at this individual chromosome. If the random number corresponding to some gene is less than the mutation probability, send inverse, namely change 0 to 1 and 1 to 0.

3.2.9. Optimal reserve strategy

To guarantee that high-quality genes brought to child individuals as possible but not be mutated or abandoned, the paper uses optimal reserve strategy. The individual with highest fitness in the parent population replaces individuals with lowest fitness in the child population to avoid accidentally damaging quality gene in the crossover and mutation operations, effectively improve algorithm convergence and optimize solution structure.

3.3. Algorithm implementation

The flow of IHGA-SCDCS algorithm is shown in Fig. 5.

4. Simulation experiment and result analysis

4.1. Experiment environment

The algorithm was carried out as a simulation experiment on CloudSim simulator. The system environment is Intel(R) Core(TM)2 Duo, 2.10 GHz and 3 GB memory. Several examples on OR-Library were used to simulate content delivery. The column in the examples corresponds to edge Level 1-SP of static content delivery network topology in the cloud storage and the row corresponds to users. The weight corresponds to delivery cost. All these examples provide optimal solution for reference. The example data can be downloaded from http://people.brunel.ac.uk/mastjjb/jeb/orlib/files/. The test examples are all SCP with weight. When the iteration number exceeds 100, it determines whether the structure of continuous iteration solution in recent 50 times has changed. If there is no change, the algorithm terminates.

The parameters involved in the algorithm are initialized as follows. The initial population size R_s is 100. The adjustable parameter to represent number for deleting columns in repeat individual modification is set to 10. The cross ratio Q_c, representing the probability that a parent generation individual brings to the next generation in the crossover operation, is 0.99. The parameter k is set to 2 but set to 1 in Scpcyc06 and Scpcyc07. The parameter to express the cross point number V_u has a different value for different examples; it is 300 from Scpbl to Scpb5, 100 from Scp41 to Scp45 and 40 in Scpcyc06 and Scpcyc07. The parameter to adjust mutation probability in mutation operation mg is set to 0.02.

4.2. Result analysis

The results of the experiment for 10 groups is shown in Table 1.

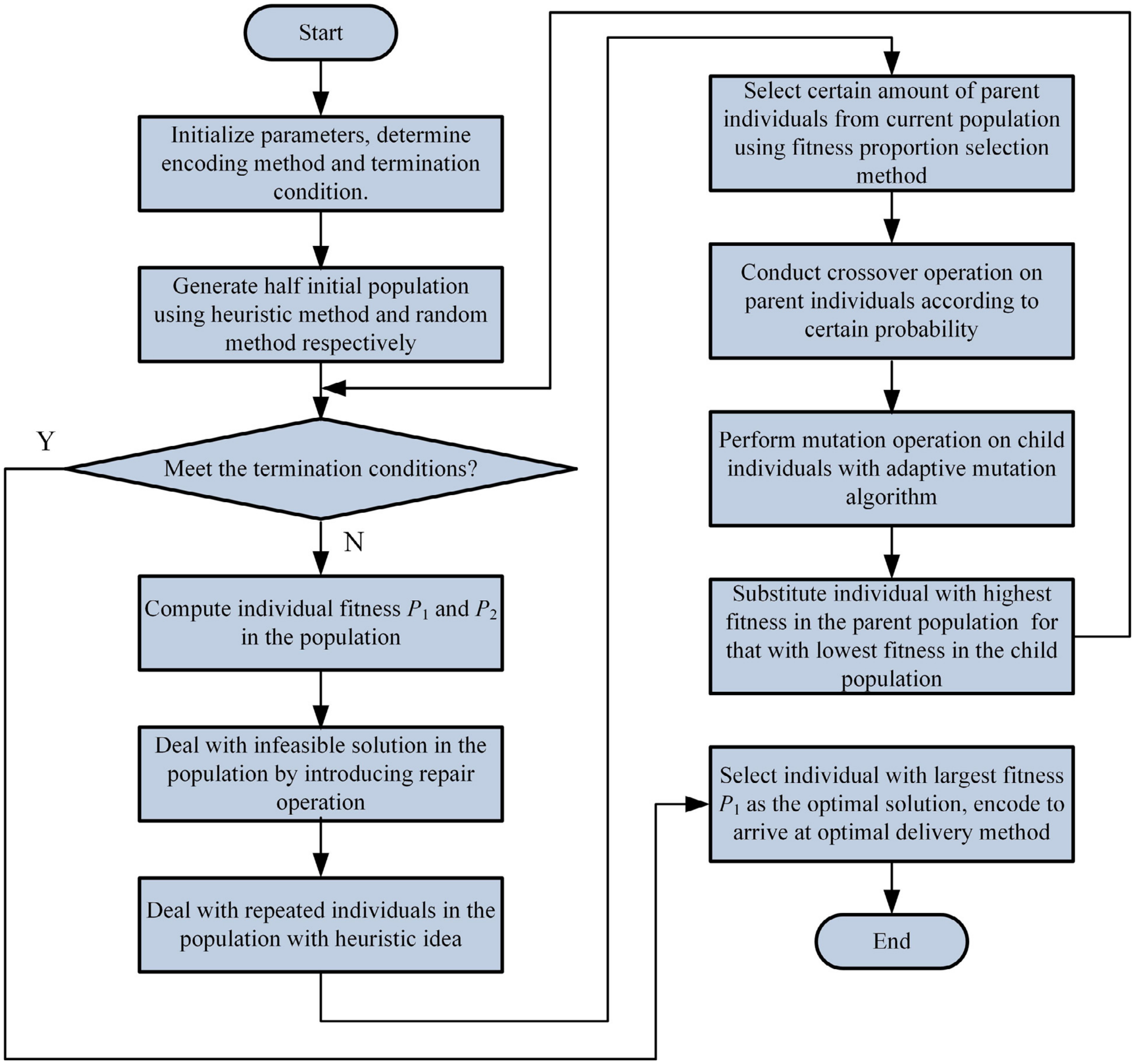

Fig. 5. IHGA-SCDCS algorithm flow.

As shown in Table 1, the 5 examples from 10 groups show that the IHGA-SCDCS has lower cost than that of the GA-heuristic by Beasley. Furthermore, the experimental results of all the examples are superior to the performance of the user greedy algorithm. In order to compare computation efficiency between IHGA-SCDCS and GA-heuristic, the GA-heuristic was implemented on CloudSim simulator. After simulation, it was found that only under larger scale cases, particularly in static content delivery of cloud storage, does the GA-heuristic spend more operation time than IHGA-SCDCS to obtain the cost of final column. The reason for this is that its mutation probability has a higher value in the latter period of convergence, so that the algorithm cannot converge, thus affecting algorithm efficiency. In summary, the IHGA-SCDCS algorithm exhibits superior and better adaptability in solving static content delivery set covering problem compared to traditional greedy algorithms and the GA-heuristic algorithm. As to delivery cost, large-scale test examples in [26] were used for performance comparison. The IHGA-SCDCS algorithm and GA-WSCP [26] solved the above examples on CloudSim simulator respectively. The operation results were compared for analysis as shown in Fig. 6. It can be seen from the figure that IHGA-SCDCS has great advantages in reducing delivery cost compared with GA-WSCP. This is because the proposed algorithm can achieve optimal solution on these large-scale examples, while GA-WSCP cannot obtain optimal solution in case of large test examples but can only arrive at an approximate solution. Therefore, the cost of IHGA-SCDCS algorithm is far below that of GA-WSCP. It indicates that the algorithm is better adapted to solve static content delivery problem in cloud storage.

Please cite this article as: Z. Zheng, Z. Zheng, Towards an improved heuristic genetic algorithm for static content delivery in cloud storage, Computers and Electrical Engineering (2017), http://dx.doi.org/10.1016/j.compeleceng.2017.06.011

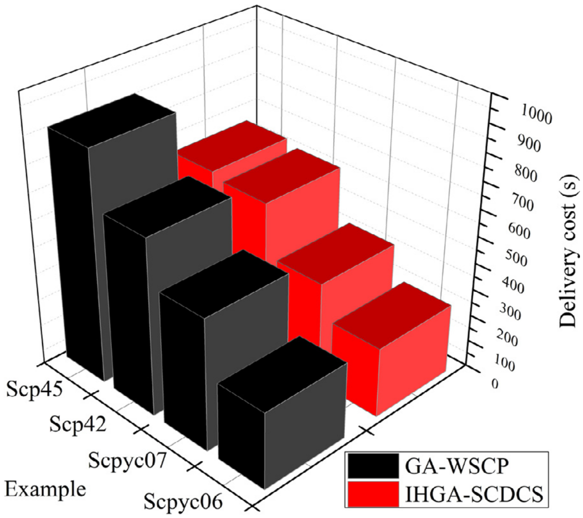

Fig. 6. Delivery cost comparison.

5. Conclusion

In this paper, a resource management model and a cost model for cloud storage were proposed. Based on the models, an improved genetic algorithm for static content delivery set covering problem was constructed and a validated simulation experiment was undertaken. The experiment results showed that the IHGA-SCDCS can solve static content delivery problem in cloud storage. In the future, the research will be focused on problems of metadata copy storage and life cycle to ensure the consistency and integrity of all cache copies in the network.

References

[1] Hu M, Luo J, Wang Y, Veeravalli B. Practical resource provisioning and caching with dynamic resilience for cloud-based content distribution networks. IEEE Trans Parallel Distrib Syst 2014;25(8):2169–79. doi:10.1109/TPDS.2013.287.
[2] Inzinger C, Nastic S, Sehic S, Vgler M, Li F, Dustdar S. Madcat: a methodology for architecture and deployment of cloud application topologies. In: 2014 IEEE 8th international symposium on service oriented system engineering; 2014. p. 13–22. doi:10.1109/SOSE.2014.9.
[3] Broberg J, Buyya R, Tari Z. Metacdn: harnessing storage clouds for high performance content delivery. J Netw Comput Appl 2009;32(5):1012–22. http://dx.doi.org/10.1016/j.jnca.2009.03.004. Next Generation Content Networks.
[4] Beijing FastWeb Technology Co. L. Website solution. http://www.fastweb.com.cn/a/solution/.
[5] Lu ZH, Gao XH, Huang SJ, Huang Y. Scalable and reliable live streaming service through coordinating cdn and p2p. In: 2011 IEEE 17th international conference on parallel and distributed systems; 2011. p. 581–8. doi:10.1109/ICPADS.2011.113.
[6] Krishnan P, Raz D, Shavitt Y. The cache location problem. IEEE ACM Trans Netw 2000;8(5):568–82. doi:10.1109/90.879344.
[7] Jinming W, Zhengchun Z, Jian W, Xiaohu T, Qun M. A sharp condition for exact support recovery of sparse signals with orthogonal matching pursuit. IEEE Trans Sig Process 2016;PP(99):1. doi:10.1109/TSP.2016.2634550.
[8] Qiu L, Padmanabhan VN, Voelker GM. On the placement of web server replicas. In: Proceedings IEEE INFOCOM 2001. Conference on computer communications. Twentieth annual joint conference of the IEEE computer and communications society (Cat. No.01CH37213), 3; 2001. p. 1587–96. doi:10.1109/INFCOM.2001.916655.
[9] Radoslavov P, Govindan R, Estrin D. Topology-informed internet replica placement. Comput Commun 2002;25(4):384–92. doi:10.1016/S0140-3664(01)00410-8.
[10] Cronin E, Jamin S, Jin C, Kurc AR, Raz D, Shavitt Y. Constrained mirror placement on the internet. IEEE J Sel A Commun 2006;20(7):1369–82. doi:10.1109/JSAC.2002.802066.
[11] Xiaohua J, Deying L, Xiaodong H, Weili W, Dingzhu D. Placement of web-server proxies with consideration of read and update operations on the internet. Comput J 2003;46(4):378–90. doi:10.1093/comjnl/46.4.378.
[12] Cidon I, Kutten S, Soffer R. Optimal allocation of electronic content. Comput Netw 2002;40(2):205–18. http://dx.doi.org/10.1016/S1389-1286(02)00251-7.
[13] Jinming W, Xiao-Wen C. Success probability of the babai estimators for box-constrained integer linear models. IEEE Trans Inf Theory 2017;63(1):631–48. doi:10.1109/TIT.2016.2627082.
[14] Kalpakis K, Dasgupta K, Wolfson O. Optimal placement of replicas in trees with read, write, and storage costs. IEEE Trans Parallel Distrib Syst 2001;12(6):628–37. doi:10.1109/71.932716.
[15] Tang X, Xu J. Qos-aware replica placement for content distribution. IEEE Trans Parallel Distrib Syst 2005;16(10):921–32. doi:10.1109/TPDS.2005.126.
[16] Rodolakis G, Siachalou S, Georgiadis L. Replicated server placement with qos constraints. IEEE Trans Parallel Distrib Syst 2006;17(10):1151–62. doi:10.1109/TPDS.2006.145.
[17] Bartolini N, Presti FL, Petrioli C. Optimal dynamic replica placement in content delivery networks. In: The 11th IEEE international conference on networks, 2003. ICON2003.; 2003. p. 125–30. doi:10.1109/ICON.2003.1266178.
[18] Presti FL, Petrioli C, Vicari C. Dynamic replica placement in content delivery networks. In: 13th IEEE international symposium on modeling, analysis, and simulation of computer and telecommunication Systems; 2005. p. 351–60. doi:10.1109/MASCOTS.2005.29.
[19] Vicari C, Petrioli C, Presti FL. Dynamic replica placement and traffic redirection in content delivery networks. ACM SIGMETRICS Perform Eval Rev 2007;35(3):66–8.
[20] Loukopoulos T, Tziritas N, Lampsas P, Lalis S. Implementing replica placements: feasibility and cost minimization. In: 2007 IEEE international parallel and distributed processing symposium; 2007. p. 1–10. doi:10.1109/IPDPS.2007.370316.

[21] Xingye H, Xinming L, Yinpeng L. Research on resource management for cloud computing based information system. In: 2010 international conference on computational and information sciences; 2010. p. 491–4. doi:10.1109/ICCIS.2010.127.
[22] You W, Mathieu B, Simon G. How to make content-centric networks interwork with cdn networks. In: 2013 fourth international conference on the network of the future (NoF); 2013. p. 1–5. doi:10.1109/NOF.2013.6724511.
[23] Xu HM, Shi YJ, Liu YL, Gao FB, Wan T. Integration of cloud computing and p2p: a future storage infrastructure. In: 2012 international conference on quality, reliability, risk, maintenance, and safety engineering; 2012. p. 1489–92. doi:10.1109/ICQR2MSE.2012.6246507.
[24] Kolodner EK, Tal S, Kyriazis D, Naor D, Allalouf M, Bonelli L, et al. A cloud environment for data-intensive storage services. In: 2011 IEEE third international conference on cloud computing technology and science; 2011. p. 357–66. doi:10.1109/CloudCom.2011.55.
[25] Lemke C, Salkin H, Spielberg K. Set covering by single-branch enumeration with linear-programming subproblems. Oper Res 1971;19(4):998–1022. doi:10.1287/opre.19.4.998.
[26] Liang C, Shi-jun R. Study on genetic algorithm application in set covering problem. J Harbin Univ Commerce 2006;22(2):67–70.
[27] Wen J, Li D, Zhu F. Stable recovery of sparse signals via l_p-minimization. Appl Comput Harmon Anal 2015;38(1):161–76. http://dx.doi.org/10.1016/j.acha.2014.06.003.

Zhigao Zheng obtained the Master's Degree from Peking University in 2011. Now he is a Ph.D Candidates of Services Computing Technology and System Lab/Cluster and Grid Computing Lab/Big Data Technology and System Lab, School of Computer Science and Technology, Huazhong University of Science and Technology. His main research interest is focused on parallel and distributed computing.

Zunxin Zheng is with College of Economics, Shenzhen University. He is a full Professor in Finance and he is the Vice Dean of College of Economics