IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 11, 2018, accepted July 12, 2018, date of publication July 25, 2018, date of current version September 21, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2859427

Redundancy-Guaranteed and Receiving-Constrained Disaster Backup in Cloud Data Center Network

XIAOLE LI[1], HUA WANG[2], SHANWEN YI[1], XIBO YAO[3], FANGJIN ZHU[2], AND LINBO ZHAI[4]

[1]School of Computer Science and Technology, Shandong University, Jinan 250100, China
[2]School of Software, Shandong University, Jinan 250100, China
[3]ZhongTai Securities, Jinan 250001, China
[4]School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China

Corresponding author: Hua Wang (wanghua@sdu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61672323, in part by the Fundamental Research Funds of Shandong University under Grant 2017JC043, and in part by the Key Research and Development Program of Shandong Province under Grant 2017GGX10122 and Grant 2017GGX10142.

ABSTRACT For disaster backup in cloud data center network, existing researches have not jointly considered sufficient data redundancy and limited receiving capacity, likely to result in underutilization of network transmission capability, unfair distribution of backup load, or even lacks of disaster resistance. In this paper, we propose a new strategy to realize bandwidth-efficient and load-fair disaster backup under redundancy and capacity constraints using customized bandwidth allocation and flexible flow scheduling in software-defined networking. Based on many-to-many relationship in disaster backup, we formulate a new redundancy-guaranteed and receiving-constrained capacitated multi-commodity flow problem. By constructing flow-ratio-constrained backup transmission model, we specify flow allocation ratio among backup data centers with limited receiving capacity. Then we present a basic ratio-aware ant colony optimization algorithm satisfying backup flow constraint and rate requirement constraint. Furthermore, to obtain higher performance in redundancy guarantee and enhance bandwidth allocation fairness among massive backup transfers, we propose a fair-rotating and ratio-aware ant colony optimization (FRRA-ACO) algorithm. Especially, we use rotary routing search for multiple concurrent flows based on backup requirement cloning to approximate the upper bound of bandwidth allocation, adjust ratio of bandwidth allocation for multiple backup transfers with different requirements, and further improve flow rate according to the maximum link utilization on links if possible. Experiments demonstrate that FRRA-ACO outperforms state-of-the-art algorithms with less backup completion time, fairer backup load distribution, and higher network utilization.

INDEX TERMS Disaster backup, redundancy-guaranteed and receiving-constrained, flow-ratio-constrained, fair-rotating and ratio-aware, software-defined networking.

I. INTRODUCTION

Nowadays, the deployment of geographically distributed (geo-distributed) data centers (DCs) is becoming an increasingly popular trend. More and more large enterprises such as Amazon, Facebook, Google, Microsoft, and Yahoo!, have been building multiple global DCs in cloud DC network [1], [2]. Large-scale and distributed cloud DCs not only play an important role in guaranteeing high quality of cloud services for global users, but also contain huge amount of valuable military and economic information [3]–[5]. However, they are facing growing failure risks due to various man-made or natural disasters [6], [7]. To avoid huge economic losses in case of disaster, more and more attentions are paid to periodic disaster backup for sufficient data redundancy among geo-distributed DCs in cloud DC network.

Disaster backup will not be real-time but usually happens in a particular time period of the day (e.g., from 3 a.m. to 6 a.m.) [8]. It copies all the newly generated data in a certain past period to remote backup DCs. Large enterprises like Google can process about 100 PB data daily in geo-distributed application DCs [6]. Although not exceeding five percent of the daily data requires backup [9], the data amount is still huge and disaster backup will consume massive bandwidth and storage capacity. On the one hand, there are two

important constraints should be considered in disaster backup process. Firstly, to obtain sufficient data redundancy against disasters, the critical data in multiple DCs is eligible for the backup process among three or more geo-distributed backup DCs [10]. Therefore, the required number of replicas for all the data with backup requirement (we denote them as primary data) should be considered as data redundancy constraint. Secondly, limited by resources, one or more application DCs have to act as backup DCs simultaneously. Considering huge storage space requirement in that case, we should consider the limited storage capacity as receiving capacity constraint. On the other hand, backup completion time and backup load distribution are two important metrics for disaster backup. Bulk backup data transfers consume huge bandwidth, likely to impact other daily services in the network. So disaster backup should be completed as soon as possible [1]. The fairness of backup load distribution is also significantly important to guarantee availability of replicas and avoid overload of DCs [11], [12]. Therefore, from the traffic engineering aspect, it is of great importance to realize fast backup transmission and fair backup load distribution with considerations of sufficient data redundancy and limited receiving capacity by appropriate residual bandwidth allocation and efficient routing selection.

It is noteworthy that the current researches on disaster backup transmission face many challenges. They have not considered resource allocation from global view, flow splitting of arbitrary proportions or intelligent multipath routing selection for concurrent transfers according to transmission requirements (e.g., various backup data amount) or destination storage status (e.g., limited receiving capacity). Existing works schedule transmission order of backup transfers one by one with different priorities (e.g., backup data amount [6], [7] or specify importance factor [13]) to reduce completion time or obtain maximum utility. But these methods above might lead to underutilization of network transmission capability in some links, or unfair backup load distribution to some backup DCs [14].

Here we present an illustrative example in Fig. 1 about backup completion time with different bandwidth allocation strategies. We assume that there are two backup transfers denoted as bt_1 (delivering 20 GB data) and bt_2 (delivering 10 GB data) to backup DC set M_1 or M_2 respectively. bt_1 and bt_2 are delivered through a shared link $e_{u,v}$. We suppose that bt_1 and bt_2 have different importance factors ($bt_1's$ is lower than $bt_2's$). The available bandwidth of $e_{u,v}$ is 20 Gbps. The total bandwidth of the path set from node v to M_1 and M_2 is 30 Gbps and 10 Gbps respectively.

In the existing researches above, if one available path is assigned to a backup transfer, all the residual bandwidth on this path will be occupied by this transfer until its transmission completion. Fig. 1(a) and Fig. 1(c) illustrate two different bandwidth allocation strategies in that case. In Fig. 1(b), bt_1 occupies all residual bandwidth (20 Gbps) in $e_{u,v}$ and transfers 20 GB data to M_1 using 8 seconds. After completion of bt_1, bt_2 transfers 10 GB data to M_2 using 8 seconds. So it takes 16 seconds in total to complete the whole backup transmission in strategy 1. In Fig. 1(d), bt_2 occupies all residual bandwidth (20 Gbps) in $e_{u,v}$ and transfers 10 GB data to M_1 using 4 seconds, and then bt_1 transfers 20 GB data to M_2 using 16 seconds. So it takes 20 seconds in total to complete the whole backup transmission in strategy 2. However, the two strategies above have not fully utilized the residual bandwidth in $e_{u,v}$. For example, in strategy 1, when bt_2 transfers data after the completion of bt_1, only 50% of the residual bandwidth in $e_{u,v}$ is utilized. The same situation appears in step 2 of strategy 2.

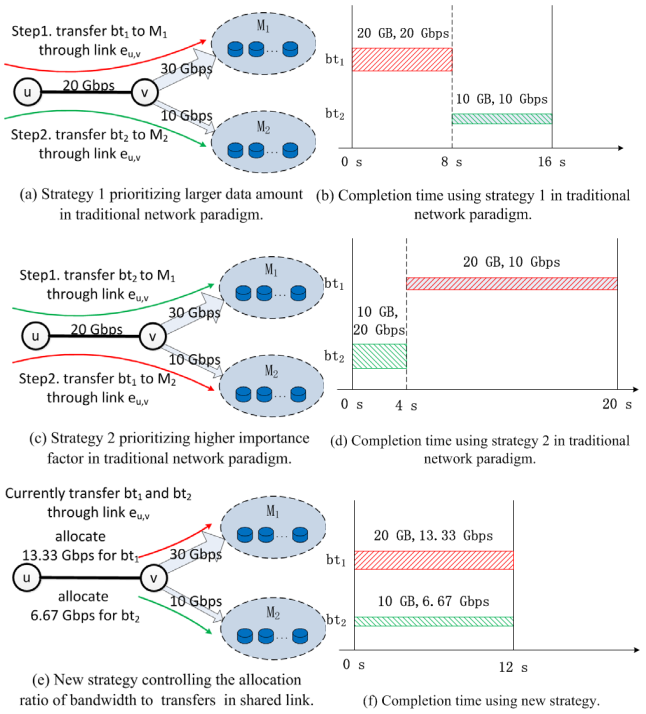


FIGURE 1. Different strategies for disaster backup transmission.

In order to further improve the utilization of network transmission capability, we should set different bandwidth ratios for these two transfers with different backup data amount, especially in their shared link $e_{u,v}$. In Fig. 1(e), we split the residual bandwidth in $e_{u,v}$ for bt_1 and bt_2 directly proportional to their data amount. In details, we assign 13.33 Gbps to bt_1 and 6.67 Gbps to bt_2 in $e_{u,v}$, and then concurrently transfer their data to M_1 and M_2 respectively. In this condition, we complete transmission for bt_1 and bt_2 simultaneously spending about 12 seconds in total in Fig. 1(f). This example illustrates that proportional bandwidth allocation for concurrent backup transfers according to their requirements (e.g., different backup data amount) can effectively improve the utilization of network transmission capability and achieve more efficient backup transmission with less total backup completion time or more transferred data before a certain deadline.

However, existing traffic management mechanisms in traditional network paradigm are not applicable for disaster

backup in cloud DC network. For example, we can use the Multi-Protocol Label Switching (MPLS) to flexibly split network flows for different network requirements according to a certain given optimization purpose [15], but face challenges in guaranteeing scalability and resilience for lack of global view and control in traffic engineering [16].

Therefore, we need a new network paradigm with global view of whole network supporting centralized management of network resources, arbitrary flow splitting optimization and multipath routing for different network service requirements. Here we choose the Software-Defined Networking (SDN) architecture. Due to its high efficiency to obtain better network capacity utilization, SDN provides an advanced supporting environment for traffic engineering of cloud DC network, and has been widely accepted to deploy and manage global DCs in large enterprises [17], [18]. In Google private cloud DC network spanning wide area networks, they leverage SDN for flexible flow splitting via multipath routing to balance link capacity, achieving almost 100% utilization rate [19]. In the SDN scenarios, we can easily optimize routing paths, improve resource utilization, and realize network traffic control. In this paper, we provide a new efficient disaster backup strategy by reducing backup completion time and achieving fair backup load distribution under redundancy constraint and receiving capacity constraint among geo-distributed DCs in SDN.

II. RELATED WORKS AND OUR SOLUTIONS

For disaster backup problem among geo-distributed DCs, many new mechanisms have been proposed, focusing on optimal resource allocation for backup storage [11], shared backup with least backup servers [12], content placement and management to provide survivability in disasters with less expected loss [20], minimum failure probability [21], high content connectivity and low wavelength consumption [22], backup path selection and content replica placement for disaster survivability [23], [24], emergency backup with maximum utility [13] or minimal cost [25], fast one-to-one backup strategy [6], [26], [27], our earlier work on rapid and fair disaster backup with receiving capacity constraints [28]. However, none of them has jointly considered data redundancy constraint and receiving capacity constraint for backup activity among geo-distributed DCs.

Most researches focus on the placement problem of DC contents and their backup replicas for disaster prevention. In [11], Bianco *et al.* trade off the minimization of maximum hop number for backup activity and the minimization of overload on backup servers after virtual machine migration in case of disasters. In [12], Couto *et al.* select locations for virtual machine servers avoiding simultaneous failure of backup and primary servers, to reduce required server amount by virtualization. References [20], [21], and [22] study placement and management of contents and their replicas among multiple DCs considering disaster risks. In [20], Ferdousi *et al.* consider disaster vulnerable location distribution and research on DC placement and data management to mitigate disaster loss. In [21], Ma *et al.* focus on DC location and content distribution to minimize failure probability. In [22], Li *et al.* define k-node (edge) connectivity to measure content reachability in case of disasters and apply it to optical DC networks. In [23], Habib *et al.* construct disaster resistant DC network jointly against path failure and node failure, and reduce backup cost by placing data close to their popular regions. But to simplify the model, they have not considered limited storage space in backup DCs. In [24], Zhou *et al.* propose three-stage algorithm including physical server selection, virtual machine placement and task assignment to reduce recovery cost under k-fault-tolerance constraint. In [13], Lu *et al.* prioritize endangered data in emergency backup according to their values and propose distributed algorithm to maximize backup profit. However, all the researches above ignore the joint consideration of appropriate routing selection and reasonable bandwidth allocation for multiple concurrent backup transfers to achieve efficient data transmission.

We note that there have been some researches about transmission path selection for disaster backup. However, none of them jointly consider redundancy constraint and receiving capacity constraint for backup activity among geo-distributed DCs. In [25], Ma *et al.* propose a theoretical framework to select backup DC nodes and evacuation routing before the arrival of predicted disasters for minimal cost of backup transmission and data storage. But they have not considered how to allocate bandwidth for concurrent backup transfers (especially in shared links) during data transmission. References [6], [26], and [27] by Yao *et al.* study mutual disaster backup strategy among multiple DCs with the object to minimize backup completion time. They design disaster-aware heuristics algorithms to select backup DC and calculate maximum flow routing for every application DC to realize rapid mutual disaster backup. However, the researches in [6], [26], and [27] are based on one-to-one backup mode which is not adaptable in practice for insufficient reliability of fault-tolerant, because critical data should have three or more replicas in geo-distributed backup DCs to obtain sufficient data redundancy [10]. In addition, they have not considered limited receiving capacity of backup DCs. We have done some earlier work about disaster backup. In [28], we consider receiving capacity constraint and propose BA-ACO algorithm to realize specified proportional bandwidth allocation and backup load distribution. However, we have not considered data redundancy constraint, and therefore there is no redundancy guarantee for solutions. Besides, after path searching stage, we have not adjusted bandwidth allocation or improve flow rate according to the maximum link utilization. Therefore, we need to further improve earlier works for sufficient data redundancy guaranteeing and higher network transmission capability utilization.

As shown above, previous works ignore the joint consideration of sufficient data redundancy and limited receiving capacity, likely to result in underutilization of network transmission capability, unfair distribution of backup load [14], or even lacks of disaster resistance.

In this paper, we newly propose Redundancy-Guaranteed and Receiving-Constrained strategy. We leverage SDN to support customized bandwidth allocation for backup transfers and flexible flow scheduling for backup DCs. We allow data transmission via multipath routing to make full use of network transmission capability. We define data redundancy constraint to guarantee sufficient replicas for primary data. We use receiving capacity constraint to fairly assign backup loads. We summarize our contributions from four aspects:

- We schedule bandwidth ratio jointly according to data redundancy requirement and receiving capacity in disaster backup for the first time, achieving more reasonable bandwidth allocation among concurrent backup transfers and fairer load distribution in geo-distributed backup DCs.
- We formulate the disaster backup problem as a new Redundancy-Guaranteed and Receiving-Constrained Capacitated Multi-Commodity Flow (RGRC-CMCF) problem based on many-to-many backup relationship.
- We build an effective Flow-Ratio-Constrained (FRC) backup transmission model to specify the ratio of flows allocated to backup DCs according to their receiving capacity for guaranteeing backup load distribution fairness.
- We design a Basic Ratio-Aware Ant Colony Optimization (BRA-ACO) algorithm satisfying backup flow constraint and rate requirement constraint. And furthermore, we propose a new Fair-Rotating and Ratio-Aware Ant Colony Optimization (FRRA-ACO) algorithm to obtain better data redundancy guarantee and enhance bandwidth allocation fairness. FRRA-ACO performs better than state-of-the-art algorithms in reducing backup completion time, balancing backup load distribution and improving network utilization.

The following chapters are organized as follows. In Section III, we present illustrative examples to explain the necessity of data redundancy constraint and receiving capacity constraint, and then give the formal specification of RGRC-CMCF problem. In Section IV, we construct a new FRC backup transmission model, and then propose and analyze BRA-ACO algorithm and FRRA-ACO algorithm in detail. In Section V, we evaluate our solution over different network topologies. At last, we summarize our work and provide suggestions for further research.

III. PROBLEM DESCRIPTION AND FORMULATION

In disaster backup, to obtain sufficient data redundancy, enough replicas of the same data (e.g., k_{rp} replicas) should be assigned to different geo-distributed backup DCs [10]. To make full use of resources, some DCs always play dual roles as application servers (i.e., application DCs) and backup servers (i.e., backup DCs) simultaneously [6]. We propose a new and practical backup model to describe many-to-many relationship between application DCs and their backup DCs in Fig. 2.

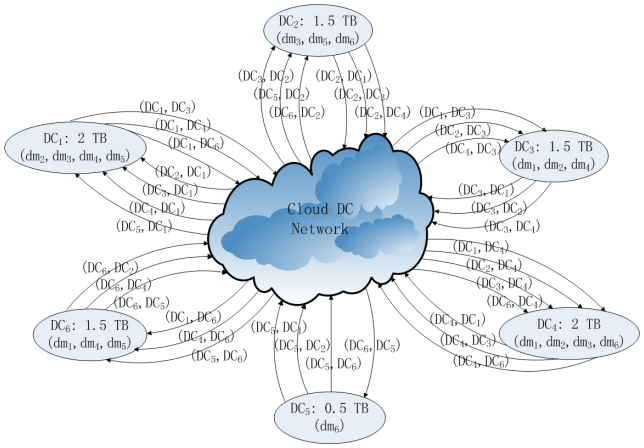


FIGURE 2. Example of many-to-many relationship between application DCs ($n = 6$) and their backup DCs ($k_{rp} = 3$).

As shown above, the primary data of $n = 6$ application DCs need to be backup. We denote the amount of primary data in DC_i as dm_i and every primary data requires $k_{rp} = 3$ replicas in geo-distributed backup DCs. The information in every ellipse represents a certain DC, its available storage space for receiving replicas, and the replicas assigned to it. We use a one-way arrow pointing to the cloud DC network to denote a backup transfer from an application DC, and a one-way arrow from the cloud DC network to denote a backup transfer to an application DC. The tuple (DC_i, DC_j) indicates the backup transfer from an application DC (DC_i) to a backup DC (DC_j). We can see the backup correspondence between every application DC and its backup DCs. For example, the application DC_1 delivers replicas to three geo-distributed backup DCs (DC_3, DC_4, and DC_6). At the same time, DC_1 acts as backup DC of four geo-distributed application DCs (DC_2, DC_3, DC_4, and DC_5) to store the replicas of their primary data (dm_2, dm_3, dm_4, and dm_5). The receiving capacity of backup DCs might be different due to their different available storage space. In Fig. 2, we assign different amount of replicas to backup DCs according to their different receiving capacity. To obtain sufficient data redundancy as shown above, we should reformulate the disaster backup problem and design more practical and efficient algorithm than the existing works.

Moreover, to obtain fast and fair backup transmission solution, flow ratio scheduling for backup DCs is a fatal factor. We have demonstrated this problem in detail in our earlier work [28]. Therefore, instead of repeating it, we just present an illustrative example here. In Fig. 3, we need to transfer 1 PB data to backup DCs $\{bd_1, bd_2, bd_3\}$. For the first strategy in Fig. 3(a), the backup transmission to bd_1 will only last for about 13.33 minutes because $bd_1's$ available storage will be fully filled with backup load at that time. The backup transmission to bd_2 will proceed for about 35.56 minutes until $bd_2's$ storage is fully filled. But the backup transmission to bd_3 will proceed for about 53.33 minutes, whereas the

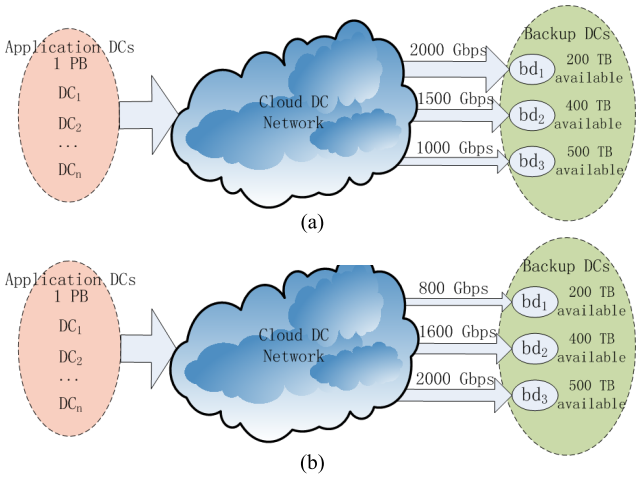


FIGURE 3. An illustrative example of flow allocation ratio in disaster backup.

bandwidth capacity allocated to bd_1 and bd_2 will lie idle for about 40 minutes and 17 minutes respectively. By contrast, the second strategy in Fig. 3(b) make better use of network transmission capability and obtain the total backup time of 30.30 minutes with more fair load distribution. We adjust flow ratio assigned to backup DCs according to their receiving capacities, and obtain more efficient and balanced disaster backup solution. As shown in Fig. 3 and Table 1, maximum total flows of backup transmission does not necessarily imply full utilization of network transmission capability. We should pay more attention to appropriate flow ratio scheduling, especially according to transmission requirement or destination storage status. Therefore, we should consider receiving capacity and backup data amount, and adapt reasonable flow scheduling to improve network transmission capability utilization and backup load distribution fairness.

TABLE 1. Comparsion of two strategies.

Compared Items	Strategy 1	Strategy 2
total available flows	4500 Gbps	4400 Gbps
total completion time	53.33 minutes	30.30 minutes
load distribution	200 TB for bd_1	181.82 TB for bd_1
	400 TB for bd_2	363.64 TB for bd_2
	400 TB for bd_3	454.54 TB for bd_3

A. PROBLEM DESTRIPTION

We assume that there is no alternate network dedicated to disaster backup because of high deployment cost especially in cloud DC network. Thus we use residual bandwidth to transfer backup data. To describe the network topology, we consider a directed and connected graph $G = (V, E)$, where V is a finite set of vertices (network nodes) and E is the set of edges (network links) representing connection of these vertices. Let $|V|$ be the number of network nodes and $|E|$ be the number of network links. We use $e_{u,v}$ to denote the link from node $u \in V$ to node $v \in V$. Let N_v^+ (N_v^-) denote the set of out-neighbor (in-neighbor) nodes of v in G. Node u is in N_v^+ (N_v^-) if there is a directed edge $e_{u,v}$ from u to v (from v to u) in E.

We use $D = \{DC_1, DC_2, \cdots, DC_n\}$ to denote the DCs with disaster backup requirements and $M = \{M_1, M_2, \cdots, M_n\}$ to denote the corresponding available backup DC set. Here, $M_i = \{bd_{i1}, bd_{i2}, \cdots\}$ denotes the available backup DC set to respectively place a piece of replica from DC_i. We define $BT_i = \{bt_{i1}, bt_{i2}, \cdots\}$ as the set of massive (tens of or even hundreds of) backup transfers from DC_i to assign replicas to M_i, and $BT = \{BT_1, BT_2, \cdots, BT_n\}$ as the total set of backup transfers for D. We assume that there are sufficient total storage capacity distributed in all geo-distributed backup DCs to place replicas transferred by BT whereas the storage capacity of every backup DC is limited.

We use a group of ants which are denoted as $ant = \{ant_1, ant_2, \cdots ant_m\}$ to search for a solution with multipath routing in every iteration. And in every iteration, we run multiple rounds to search for available paths for every backup transfer. Let $path = (path_1, path_2, \cdots, path_n)$ denote the path set for backup data delivering. Every $path_i = path_i(BT_i, M_i)$ is a sub-graph of G spanning the source node $DC_i \in D$ and the set of its destination nodes $M_i \in M$. Let $p(bt_{ij}, d)$ be the set of multiple paths for the backup transfer bt_{ij} to a destination node $d \in M_i$. We define $p_t(bt_{ij}, d)$ as the tth path for bt_{ij} to a destination node $d \in M_i$ in path searching.

There are three non-negative real value functions associated with every link $e_{u,v} \in E$: delay $d(e_{u,v}) : E \to R^+$, available bandwidth $c(e_{u,v}) : E \to R^+$, and path shared degree $psd(p(bt_{ij}, d), e_{u,v}) : E \to N$. The link delay $d(e_{u,v})$ is considered to be the sum of processing, propagation, and switch configuration delays. The link bandwidth $c(e_{u,v})$ is the residual bandwidth function. The path shared degree $psd(p(bt_{ij}, d), e_{u,v})$ stands for the number of paths in $p(bt_{ij}, d)$ through $e_{u,v}$, and is defined as follows:

$$psd(p(bt_{ij}, d), e_{u,v}) = \sum_{p_t(bt_{ij},d) \in p(bt_{ij},d)} \sum_{d \in M_i} \delta_{ij,t,d,e_{u,v}} \quad (1)$$

The binary variable $\delta_{ij,t,d,e_{u,v}}$ denotes if $e_{u,v}$ is in $p_t(bt_{ij}, d)$ to a destination node $d \in M_i$. Special to note is that the path shared degree indicates the relative importance of a link in the disaster backup routing topology for bt_{ij}. A larger value of $psd(p(bt_{ij}, d), e_{u,v})$ means that more paths share $e_{u,v}$ during backup data transmission for bt_{ij}.

We use $b(p_t(bt_{ij}, d))$ to denote the allocated bandwidth for $p_t(bt_{ij}, d)$. We define total bandwidth of $p(bt_{ij}, d)$ as:

$$b(p(bt_{ij}, d)) = \sum_{p_t(bt_{ij},d) \in p(bt_{ij},d)} b(p_t(bt_{ij}, d)) \quad (2)$$

We use binary variable $x_{bt_{ij},d}$ to denote if the primary data in bt_{ij} is backuped in a destination node $d \in M_i$. We define total rate f_{ij} of bt_{ij} by multipath routing as follows:

$$f_{ij} = \sum_{d \in M_i} x_{bt_{ij},d} \cdot b(p(bt_{ij}, d))) \quad (3)$$

We use f_{uv}^{ij} to denote the flow allocated to bt_{ij} in $e_{u,v}$. We define dm_{ij} as the amount of backup data in bt_{ij} and define dm_{ij}/dl_{ij} as rate requirement to complete bt_{ij} within its disaster backup deadline dl_{ij}. Considering data integrity, we can assign a piece of replica to only one backup DC and every backup DC with enough free storage space can hold multiple replicas from different application DCs. We define sc_d as available storage capacity for disaster backup load in $d \in M$ and use $\left(\bigcup_{i=1}^{n} M_i\right)$ to denote all the backup DCs with replicas. We use α_d to denote the ratio of backup receiving capacity in d to total backup receiving capacity of all backup DCs as follows:

$$\alpha_d = sc_d \Big/ \sum_{d \in \left(\bigcup_{i=1}^{n} M_i\right)} sc_d \tag{4}$$

We use β_{ij} to denote the ratio of total data amount in bt_{ij} to total data amount of all backup transfers as follows:

$$\beta_{ij} = \left(\sum_{d \in M_i} \left(x_{bt_{ij},d} \cdot dm_{ij}\right)\right) \Big/ \left(\sum_{BT_i \in BT} \sum_{bt_{ij} \in BT_i} \sum_{d \in M_i} \left(x_{bt_{ij},d} \cdot dm_{ij}\right)\right) \tag{5}$$

B. PROBLEM FORMULATION

The objective function of RGRC-CMCF problem is:

$$\text{maximize} \sum_{BT_i \in BT} \sum_{bt_{ij} \in BT_i} f_{ij} \tag{6}$$

We aim to maximize total available flows for disaster backup in the network to achieve fast data transmission and fair load distribution under the following constraints:

$$\sum_{BT_i \in BT} \sum_{bt_{ij} \in BT_i} \sum_{d \in M_i} \sum_{p_t(bt_{ij},d) \in p(bt_{ij},d)} \left(\delta_{ij,t,d,e_{u,v}} \cdot b(p_k(bt_{ij},d))\right) \le c(e_{u,v}) \tag{7}$$

$$\sum_{v \in V} f_{uv}^{ij} - \sum_{v \in V} f_{vu}^{ij} = \begin{cases} -b(p(bt_{ij},d)) & \text{u = d} \\ 0 & otherwise \\ b(p(bt_{ij},d)) & \text{u = DC}_i \end{cases}, \quad d \in M_i, bt_{ij} \in BT_i \tag{8}$$

$$\left| \left(\sum_{BT_i \in BT} \sum_{bt_{ij} \in BT_i} b(p(bt_{ij},d)) \Big/ \sum_{d \in \left(\bigcup_{i=1}^{n} M_i\right)} \sum_{BT_i \in BT} \sum_{bt_{ij} \in BT_i} b(p(bt_{ij},d)) \right) - \alpha_d \right| \le \omega_1, \quad d \in M_i \tag{9}$$

$$\left| \left(f_{ij} \Big/ \sum_{BT_i \in BT} \sum_{bt_{ij} \in BT_i} f_{ij} \right) - \beta_{ij} \right| \le \omega_2 \tag{10}$$

$$f_{ij} \ge dm_{ij}/dl_{ij}, \quad bt_{ij} \in BT_i,\ BT_i \in BT \tag{11}$$

$$b(p_t(bt_{ij},d)) \ge 0, \quad d \in M,\ bt_{ij} \in BT_i,\ BT_i \in BT \tag{12}$$

$$\sum_{d \in M_i} x_{bt_{ij},d} \ge k_{rp}, \quad bt_{ij} \in BT_i,\ BT_i \in BT \tag{13}$$

The link capacity constraint in (7) ensures that the aggregated traffic through any link cannot exceed its maximum capacity. The flow conservation constraint in (8) ensures that for each backup flow, the input traffic equals to the output traffic at any intermediate node in the paths to destinations. The ω_1 and ω_2 in (9) and (10) are two very small nonnegative decimals. The backup receiving capacity constraint in (9) ensures the close approximation of flow ratio to receiving capacity ratio α_d for every backup DC. Next in Section IV, we will extend original network $G(V, E)$ to construct a new FRC backup transmission model $G'(V', E')$, in order to specify the ratio of flows allocated to backup DCs according to (9). The backup flow constraint in (10) ensures the close approximation of flow ratio to data amount ratio for every backup transfer, to achieve rapid transmission of total backup data as shown in Fig. 1. In the next section, we will design two algorithms to determine the transmission paths and adjust bandwidth allocations for every backup transfer according to this constraint. The rate requirement constraint in (11) ensures sufficient bandwidth obtained via multipath routing for every backup transfer, to guarantee timely transmission completion. The flow value constraint in (12) denotes that the value of backup flows should be nonnegative. The data redundancy constraint in (13) denotes that for every backup transfer, total number of replicas assigned to multiple geo-distributed backup DCs should not be less than the required replica number k_{rp} to guarantee enough data redundancy.

In our earlier work, we have considered receiving capacity constraint and formulated the disaster backup problem to be Receiving-Capacity-Constrained Capacitated Multi-Commodity Flow (RCC-CMCF) problem which is NP-complete [28]. Obviously, RGRC-CMCF problem is a special case of RCC-CMCF problem and more challenging with the addition of data redundancy constraint. Thus RGRC-CMCF problem is NP-complete, too. In the cloud DC network of large-scale and high-complexity structure, it is extremely essential to obtain optimal or near-optimal solution within acceptable computing time. For RGRC-CMCF problem, we design algorithms based on ACO which is effective to solve the CMCF problem [29], [30] in the following sections.

IV. ALGORITHM DESIGN

We first construct FRC transmission model to specify flow ratio for backup DCs. Based on it, we propose and analyze two algorithms to solve RGRC-CMCF problem.

A. FRC TRANSMISSION MODEL FOR FLOW RATIO

To specify flow ratio of backup DCs, we extend the original network model $G(V, E)$ to a new model $G'(V', E')$

exclusively used for disaster backup transmission. In our earlier work for disaster backup [28], we have proposed RCA network model which introduces corresponding super node for every original backup DC, with the purpose of flow ratio control. But we have not fully considered the effect of data redundancy constraint on storage space and just set every super node to have infinite storage space. This setting will not affect flow ratio value which is determined by receiving capacity of backup DCs. But it still causes some bias in fitness evaluation especially for backup load distribution and might mislead the routing search of ants. Therefore, in this paper, we define super backup DCs as virtual nodes (e.g., $bd_i^{'}$ in Fig. 4) owning the same storage capacity as the original nodes (e.g., bd_i) instead of infinite storage space in [28].

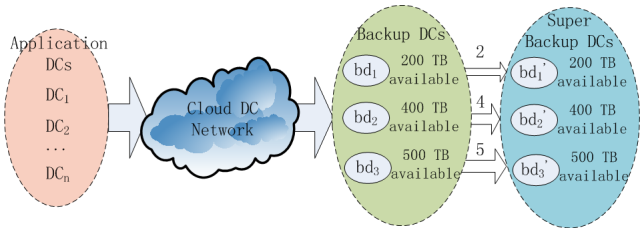


FIGURE 4. Example of FRC transmission model.

In other respects, we still maintain the original settings. Therefore, instead of repeating them, we just give a brief description and present an illustrative example here. In general, we define $V^{'} = V \cup M_{add}$ and $E^{'} = E \cup E_{add}$. We use $M_{add} = \{bd_1', bd_2', \ldots\}$ to denote the super backup DC set which will be used as destination nodes for backup data transmission instead of original backup DC set M in our algorithms. In order to distinguish between every available backup DC and its super backup DC, we call the former as original destination node which is the unique previous node of the later. We use $E_{add} = \{e_{bd_1,bd_1'}, e_{bd_2,bd_2'}, \ldots\}$ to denote the directed edges from every original destination node $bd_i \in M$ to its corresponding destination node $bd_i^{'} \in M_{add}$. To guarantee flow ratio control in algorithm implement, we specify the ratio of link capacity in $e_{bd_i,bd_i'}$ to total link capacities in E_{add} to be equal to the ratio of available storage capacity in bd_i to total available storage capacity in all backup DCs.

As in Fig. 4. We specify the link capacity ratio in directed edges to constrain flow ratio (i.e., 2: 4: 5) for backup DCs. Performing data transmission through FRC, we use flow ratio to control load distribution among backup DCs (as the BRA-ACO algorithm and FRRA-ACO algorithm in the next subsection).

B. ALGORITHM DESIGN

ACO is a global optimization evolutionary algorithm, which imitates the collective food searching behavior of ant colony in real world. Ants often travel between their nests and food sources by the guidance of shared foraging information which is called pheromone. During foraging process, ants prefer to choose, in probability, the path with higher pheromone concentration. After reaching the food source, ants evaluate their passing paths and leave clues to other ants in the form of pheromone. The pheromone trails on paths lead to effective communication of sharing information, which enables ants to find the shortest paths between their nest and food sources. Specifically, when choosing candidate node to construct available path, ACO leverages transition probability model which depends on pheromone factor and heuristic factor. Compared with other optimization algorithms, ACO has following characteristics: positive feedback mechanism to make search process converge and approach optimal solution, pheromone trail to realize indirect communication among individuals, distributed and parallel computing by many individuals simultaneously to greatly improve algorithm computing efficiency, and heuristic probabilistic search to avoid local optimum and find global optimal solution.

Based on the newly proposed FRC network model and existing ACO metaheuristic [31], we design new algorithms to determine transmission paths and bandwidth allocation to realize redundancy-guaranteed and receiving-constrained disaster backup for geo-distributed DCs in SDN. In the following subsections, we describe pheromone trail, heuristic information, fitness evaluation, and then propose two available algorithms.

1) PHEROMONE TRAIL AND HEURISTIC INFORMATION

To avoid mutual interference of different backup transfers in the path searching process, we set unique pheromone trail and heuristic information for every bt_{ij}. After the kth iteration, we choose the used links in cur_CMCF and gb_CMCF for bt_{ij}, and enhance the pheromone intensity. Here the cur_CMCF is total amount of flows for the current solution cur in the kth iteration, and the gb_CMCF is total amount of flows for the global best solution gb. We define the rules of pheromone update as:

$$\tau_{uv}^{ij}(k+1) = (1-\rho)\tau_{uv}^{ij}(k) + \Delta\tau_{uv}^{ij}(k) \tag{14}$$

$$\Delta\tau_{uv}^{ij}(k) = \big(\lambda\chi_{ij}(k) + \mu\sigma_{ij}(k)\big)\big/\big(\varepsilon_{ij}(k)+1\big) \tag{15}$$

$$\chi_{ij}(k) = \frac{(x_{u,v}^{cur,ij} \cdot cur\_CMCF) \cdot \big(psd(p(bt_{ij},d),e_{u,v})\big)^{\theta}}{UP\_CMCF} \tag{16}$$

$$\sigma_{ij}(k) = \frac{(y_{u,v}^{gb,ij} \cdot gb\_CMCF) \cdot \big(psd(p(bt_{ij},d),e_{u,v})\big)^{\theta}}{UP\_CMCF} \tag{17}$$

$$x_{u,v}^{cur,ij} = \begin{cases} 1 & \text{if } e_{u,v} \text{ belongs to } p(bt_{ij},d) \text{ for } cur\_CMCF \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

$$y_{u,v}^{gb,ij} = \begin{cases} 1 & \text{if } e_{u,v} \text{ belongs to } p(bt_{ij},d) \text{ for } gb\_CMCF \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

$$UP\_CMCF = \min\{\sum_{D_i \in D}\sum_{u \in V'} c(DC_i, u), \sum_{d \in \left(\bigcup_{i=1}^{n} M_i\right)}\sum_{v \in V'} c(v, d)\} \quad (20)$$

$$\varepsilon_{ij}(k) = \sum_{d \in M_i} z_{u,v}^{k,ij}\omega_{ij}^{k} \quad (21)$$

$$\omega_{ij}^{k} = \left|\left(cur_{bt_{ij}}/cur\_CMCF\right) - \beta_{ij}\right| \quad (22)$$

$$z_{u,v}^{k,ij} = \begin{cases} 1 \text{ if } e_{u,v} \text{ lies in } bt_{ij}\text{'s path set in the } kth \text{ iteration} \\ 0 \text{ otherwise} \end{cases} \quad (23)$$

We use ρ as evaporating parameter controlling the evaporating speed of pheromone after every iteration. We use λ and μ to express the influences of cur_CMCF and gb_CMCF on the increment of pheromone intensity in the $(k+1)th$ iteration. We define UP_CMCF as upper bound of maximum network flow in backup data transmission. We use θ to express the influence of $psd(p(bt_{ij}, d), e_{u,v})$ in $\chi_{ij}(k)$ and $\sigma_{ij}(k)$. We define ω_{ij}^{k} as the bias between β_{ij} and the ratio of bandwidth allocated for bt_{ij} to total bandwidth allocated for all backup transfers in the kth iteration. We use $cur_{bt_{ij}}$ to express total bandwidth allocated to bt_{ij} in the kth iteration.

Heuristic information $\eta_{u,v}^{ij}(k+1)$ reflects the prior and deterministic factors on $e_{u,v}$ in the $(k+1)th$ routing search process for bt_{ij}. We define it with the residual bandwidth capacity $c(e_{u,v})$, the length of shortest path from v to the nearest destination node, and the delay on $e_{u,v}$. The heuristic information in the $(k+1)th$ iteration is as follows:

$$\eta_{u,v}^{ij}(k+1) = \varpi \cdot \left(\frac{c(e_{u,v})}{(dis(v, d_i)+1) \cdot d(e_{u,v})}\right) \quad (24)$$

We use ϖ to adjust the value of $\eta_{u,v}^{ij}(k+1)$ and use $dis(v, d_i)$ to denote the length of the shortest path from node v to the nearest destination node for bt_{ij}.

2) TRANSITION PROBABILITY

We use $R_{u,v}^{ij}(k+1)$ to describe the possibility of a certain node v being chosen from the current node u in the $(k+1)th$ routing search process for bt_{ij}. We define it with pheromone intensity and heuristic information occupying different importance factors ϕ and φ respectively. We denote $N(u)$ as the neighborhood set of u. The definition is as follows:

$$R_{u,v}^{ij}(k+1) = \frac{\left(\tau_{u,v}^{ij}(k+1)\right)^{\phi} * \left(\eta_{u,v}^{ij}(k+1)\right)^{\varphi}}{\sum_{w \in N(u)}\left(\left(\tau_{u,w}^{ij}(k+1)\right)^{\phi} * \left(\eta_{u,w}^{ij}(k+1)\right)^{\varphi}\right)} \quad (25)$$

3) FITNESS EVALUATION

In every iteration, we need to evaluate the fitness of the solution. To realize redundancy-guaranteed and receiving-constrained disaster backup, we focus on the following metrics: total amount of available flows, backup load distribution variation, and network transmission capability utilization. So the evaluation of fitness is a compound function of the three above:

$$fitness(cur) = \begin{cases} 0, \quad if \left(\sum_{d \in M_i} x_{bt_{ij},d} < k_{rp}, bt_{ij} \in BT_i, BT_i \in BT\right) \\ \text{or } \left(f_{ij} < dm_{ij}/dl_{ij}, bt_{ij} \in BT_i, BT_i \in BT\right) \\ \alpha_1 \cdot e^{cur\_CMCF/gb\_CMCF} \\ +\alpha_2 \cdot e^{-loaddis(cur)/bestloaddis} \\ +\alpha_3 \cdot e^{NT(cur)/bestNT}, \quad otherwise \end{cases} \quad (26)$$

$$loaddis(cur) = \sqrt{\sum_{d \in \left(\bigcup_{i=1}^{n} M_i\right)}\left(\left(\sum_{BT_i \in BT}\sum_{bt_{ij} \in BT} x_{bt_{ij},d} \cdot dm_{ij}\right)\Big/ sc_d - avgload\right)^2} \quad (27)$$

$$NT(cur) = Throughput_{cur}/MaxFlow \quad (28)$$

We use $avgload$ to denote average ratio of backup load to the total backup receiving capacity in all backup DCs. We use $Throughput_{cur}$ to denote the throughput in cur, and $MaxFlow$ to denote the total maximum network flow from application DCs to their backup DCs. We use Normalized Throughput [32] $NT(cur)$ to evaluate network transmission capability utilization in cur. The $loaddis(cur)$ and $NT(cur)$ denote the values of total backup load distribution variation and network transmission capability utilization function in cur, and the variables $bestloaddis$ and $bestNT$ stand for the total backup load distribution variation and network transmission capability utilization function in gb, respectively. α_1, α_2, α_3 are the weighted functions to represent the importance of corresponding measurements. We define $\alpha_1, \alpha_2, \alpha_3 \geq 0$, and $\alpha_1 + \alpha_2 + \alpha_3 = 1$. In the real algorithms, different values can be set according to the requirements of the user. In the simulation, we set α_1, α_2, α_3 to be 0.7, 0.15, and 0.15 respectively by experience.

4) BRA-ACO ALGORITHM

BRA-ACO algorithm has two basic operations: path searching and pheromone updating. We set cur_CMCF to store the sum of the flows in cur and initially set $cur\_CMCF = 0$. Every ant does not search multiple available paths to obtain bandwidth as much as possible for every backup transfer one by one based on different priorities. Because this traditional way may result in unfair load distribution and more completion time. Therefore, an ant just searches for one available path and adds the path to $p(bt_{ij}, d)$ for every bt_{ij}. And we update cur_CMCF and network status. The abovementioned steps are repeated until all ants finish path searching. Then we obtain $path = (path_1, path_2, \cdots, path_n)$ and compute cur_CMCF. After obtaining a solution cur, we assess the fitness, and update gb and gb_CMCF according to fitness

if necessary. Then we update the pheromone and run the next round of iterations until BRA-ACO converges.

The pseudo code of BRA-ACO algorithm is as follows:

Algorithm 1 BRA-ACO Algorithm

Input: $G^{'}(V^{'}, E^{'})$; $BT = \{bt_1, bt_2, \cdots, bt_n\}$
Output: disaster backup solutions for every backup transfer bt_{ij}

```
Set parameters, initialize pheromone trails, etc.
/* Run iterative loop */
while termination condition not met do
    /* Construct a disaster backup solution */
    for every ant do
        /* Search for one path for every backup transfer */
        for every bt_ij do
            while not reach available backup DC do
                /* Choose next node in the transition probability */
                Calculate heuristic information according to (24)
                Select the next node according to (25)
                if reach available backup DC do
                    /* Update path set and network status */
                    Add the path to p(bt_ij, d)
                    Update network G'(V', E')
                end if
                if the next node not exist do
                    Break
                end if
            end while
        end for
    end for
    /* Obtain a solution and compute total flows */
    Obtain path and compute cur_CMCF
    /* Determine whether the solution satisfies the constraints */
    if satisfy rate requirement and redundancy constraint do
        /* Update the global best solution if necessary */
        if find a better solution do
            Update gb_CMCF
        end if
    end if
    /* Update pheromone */
    Apply updating rule (14)
end while
```

BRA-ACO is terminated when it converges or reaches maximum iteration number. In every iteration, multiple ants participate in the path set construction process for bt_{ij}. We set m as the number of ants. At most m paths are generated for bt_{ij}, so the time complexity is approximately $O(mn|V|)$.

5) FRRA-ACO ALGORITHM

Based on the FRC network model, we propose BRA-ACO algorithm. However, there are two disadvantages. First, BRA-ACO cannot guarantee sufficient replicas for every primary data. In any round of path searching for a backup transfer, there is a certain probability that the current ant cannot find available path, especially in the last round(s). Finally, BRA-ACO cannot guarantee that the already found paths can reach enough remote backup DCs to place sufficient replicas for every primary data. Second, BRA-ACO cannot control the bandwidth amount allocated to different primary data, and therefore may fail to satisfy the backup flow constraint and cannot provide reasonable bandwidth allocation according to different backup requirements (e.g., backup data amount). As in Fig. 1, this may result in transmission capability underutilization in some links or unfair load distribution to some backup DCs.

Furthermore, in order to obtain higher performance in redundancy guarantee and enhance bandwidth allocation fairness among multiple backup transfers, we propose the FRRA-ACO algorithm. Especially, we use rotary routing search for multiple concurrent flows based on backup requirement cloning to approximate the specified ratio of bandwidth allocation, and improve network transmission capability by reasonable bandwidth ratios for multiple backup transfers with different requirements, unlike conventional solutions to achieve maximum network flow for every backup requirement one by one respectively.

FRRA-ACO has three basic operations: rotary path searching, bandwidth ratio adjusting, and pheromone updating. First, we reconstruct the set of backup transfers. To ensure adequate data redundancy and fair backup load distribution, we clone the backup requirement set and expand $BT = \{BT_1, BT_2, \cdots, BT_n\}$ into k_{rp} copies, and obtain new set as $BT^{'} = \{BT^1, BT^2, \ldots, BT^{k_{rp}}\}$. Here we use every $BT^i = \{BT_1, BT_2, \cdots, BT_n\}$ to denote a copy of BT. Then we use every ant to search for a single available path for every backup transfer bt_{ij}^k in BT^k $(1 \leq k \leq k_{rp})$ one by one in every iteration.

However, we should consider that the process of path searching does not always go well. So we mark a transfer if the ant finds one available path for it in the current path searching round. For the unlucky transfer(s) (unmarked), the ant will proceed special path searching again until every backup requirement obtains one available path or the round number of re-search exceeds the maximum. This measure is for the purpose of guaranteeing routing path allocation as fair as possible for every backup transfer with sufficient redundancy assurance. After special path searching, the failed path search for some unlucky transfer(s) means that it is unable to allocate bandwidth for the unlucky transfer(s) in the current network status. However, even if we cannot allocate enough bandwidth for them, we should allocate as much as possible and add bandwidth allocation to them in subsequent

steps by adjustment rules. Therefore, we reserve the existing search result, and update *path* and $G^{'}(V^{'}, E^{'})$.

If every backup transfer has flow path, we adjust bandwidth allocations according to (10). In [14], we have proposed two bandwidth ratio adjustment rules and applied them to concurrent evacuation transfer optimization. Here, we modify the original rules and extend them to three new rules suitable for disaster backup transmission as follows:

- flow rearranging in shared links. In the links shared by multiple backup transfers, we adjust the ratios of allocated flows among them if necessary. If we have to retrieve the flow assigned to some backup transfer in shared link, we choose the one for least backup data amount.
- flow rearranging by path pruning. If $bt_j's$ original destination node d_j is an intermediate node of a certain path for bt_i, we can abandon this $bt_i's$ path, prune the links from d_j to $bt_i's$ destination node $d_i^{'}$, and then construct a new path with original destination d_j to allocate more bandwidth for bt_j if necessary.
- flow rearranging by path aggregation. If $bt_j's$ original destination node d_j is an intermediate node of a certain path for bt_i, we can abandon this $bt_j's$ path and aggregate its available transmission capability to $bt_i's$ path to allocate more bandwidth for bt_i if necessary.

The adjustment process terminates if the convergence condition is satisfied, for example that the bandwidth ratio offset is less than ω_2 or the number of adjustment iterations reaches a specified value. The path searching process and bandwidth ratio adjustment (if necessary) will continue with the next ant. After the path searching of all ants, we adjust total bandwidth allocations for every bt_{ij}^k in case of finding no path for some bt_{ij}^k (s) in certain previous round(s). Then we obtain *path*, check the maximum utilization on these links, and improve flow rate as much as possible. This measure is for the purpose of making full use of residual bandwidth. Finally we assess the fitness of *cur*, and update *gb* and *gb_CMCF* if necessary. The abovementioned steps for solution searching are repeated until FRRA-ACO converges.

The pseudo code of FRRA-ACO algorithm is as follows:

Algorithm 2 FRRA-ACO Algorithm

Input: $G^{'}(V^{'}, E^{'})$; $BT^{'} = \{BT^1, BT^2, \cdots, BT^{k_{rp}}\}$
Output: disaster backup solutions for every backup transfer bt_{ij}^k

```
1.  Set parameters, initialize pheromone trails, etc.
    /* Set priorities to facilitate bandwidth allocation adjustment */
2.  Sort transfers in every BT^k in ascending order of deadlines
    /* Run iterative loop */
3.  while termination condition not met do
      /* Construct a disaster backup solution */
4.    for every ant do
5.      Set every bt_ij^k as unmarked
        /* Search for one path for every unmarked backup transfer*/
6.      for every bt_ij^k do
7.        if bt_ij^k is marked do
8.          Continue
9.        else
10.         while not reach available backup DC do
              /* Choose next node in transition probability */
11.           Calculate heuristic information according to (24)
12.           Select the next node according to (25)
13.           if reach available backup DC do
                /* Mark bt_ij^k to avoid repeated path searching */
14.             Mark bt_ij^k
                /* Update path set and network status */
15.             Add the path to path_ij
16.             Update network G'(V', E')
17.           end if
18.           if the next node not exists do
19.             Break
20.           end if
21.         end while
22.       end if
23.     end for
        /* Re-search available path for unlucky backup transfer(s) */
24.     for every bt_ij^k do
25.       if unmarked and re-search iterations not exceed do
26.         Goto 6
27.       end if
28.     end for
        /* Approximate bandwidth ratio to data amount ratio for every backup transfer */
29.     if every bt_ij^k has flow path(s) do
30.       Adjust bandwidth allocation according to (10)
31.       Update cur_CMCF
32.       Update network G'(V', E')
33.     end if
34. end for
    /* Implement further adjustment if necessary */
35. Adjust bandwidth allocation for every bt_ij^k according to (10)
    /* Utilize residual transmission capability on selected links */
36. Obtain path and improve flow rate according to the maximum utilization on these links if possible
    /* Determine whether the solution satisfies the constraints */
```

Algorithm 2 *(Continued.)* FRRA-ACO Algorithm

```
37.  if satisfy rate requirement and redundancy
     constraint do
       /* Update the global best solution if necessary */
38.    if find a better solution do
39.      Update gb_CMCF
40.    end if
41.  end if
     /* Update pheromone */
42.  Apply updating rule (14)
43. end while
```

Similarly, we set m as the number of ants. For bt_{ij}^{k}, at most m paths are generated, so the time complexity is approximately $O(nmk_{rp}|V|)$. Because the k_{rp} is always a constant (around three usually according to actual needs), we can denote time complexity as approximately $O(nm|V|)$.

V. PERFORMANCE EVALUATION

A. ENVIRONMENT AND CONFIGURATION

We compare our algorithms with three representative algorithms. First, we implement basic maximum network flow algorithm without data redundancy constraint and capacity constraint (MF-NORC). We sort the application DCs in descending order of primary data, and calculate the maximum flow to transfer data one by one. Second, we choose the TwoStep-ILP algorithm in [6]. TwoStep-ILP divide backup process into two steps: it formulates the first integer linear programming to determine backup DC locations to minimize the total hop number between all the application DCs and their backup DCs; then it formulates the second integer linear programming to obtain maximum total network flows for backup activity. Different from our new algorithms, TwoStep-ILP has not established association for bandwidth ratios and backup requirements among concurrent backup transfers, and has not considered redundancy constraint and receiving capacity constraint. Then, we choose the BA-ACO algorithm in [28] which is our earlier work. There are obvious differences among BA-ACO, BRA-ACO, and FRRA-ACO algorithm. Only aiming at limited receiving capacity in backup DCs, BA-ACO has not considered data redundancy constraint and there is no related setting in its pheromone trail and heuristic information. After the path searching stage, BA-ACO neither adjusts bandwidth allocation, nor improves flow rate according to maximum utilization on these links.

We implement above algorithms in a DELL OPTIPLEX 9020 server with eight Intel(R) Core(TM) i7-4790 3.60 GHz CPUs and 8 GB RAM. We perform simulations over two kinds of network topologies as follows:

1) THE WAXMAN TOPOLOGY MODEL

We use Waxman topology model [33] to randomly generate network topology with $p(e_{u,v}) = \rho' \cdot e^{-\left(\frac{distance(u,v)}{L \cdot \sigma'}\right)}$. We use $p(e_{u,v})$ to denote the probability of link $e_{u,v}$, $distance(u, v)$ to denote the Euler Distance between node u and v, and L to denote the maximum value of $distance(u, v)$. We use ρ' to control the number of links and σ' to represent the number of short links. We generate random network topologies with different nodes and run algorithms on them for comparison.

In simulations, the total data amount for backup is fixed. We randomly choose 8 nodes as application DCs with backup requirement and 8 nodes as backup DCs. Obviously, some nodes simultaneously act as application DC and backup DC. We randomly choose the primary data on each application DC to make up 100 backup transfers in all. We set available bandwidth uniformly distributed on each link within [1000, 3000] (Gbps). To guarantee sufficient data redundancy, we set the required replica number $k_{rp} = 3$.

2) THE U.S. BACKBONE TOPOLOGY

The U.S. backbone topology used in simulation is in Fig. 5. Similar to [27] and [28], we modify the original topology for the convenience of representing data transmission between every pair of nodes in the disaster backup process. We set available bandwidth on every link uniformly distributed within [1000, 3000] (Gbps). Every bidirectional link is decomposed into two unidirectional links with opposite directions. Because of the inability to actually measure the bandwidth in the two directions, we assume that the two newly generated links share the bandwidth in the original link by random ratios. In this topology, we denote the nodes with thick black circles as application DC nodes, and the nodes covered by blue shade as backup DC nodes. The application DCs have backup data ranging from 100 TB to 500 TB. It is worth noting that the backup DCs at nodes 6, 8, 9, 13, 15 and 22 simultaneously act as application DC and backup DC. We randomly choose the primary data on each application DC to make up 100 backup transfers with content size ranging from 2 TB to 30 TB. The total amount of data ranges from 0.5 PB to 5 PB. To guarantee sufficient redundancy, we still set the required replica number $k_{rp} = 3$.

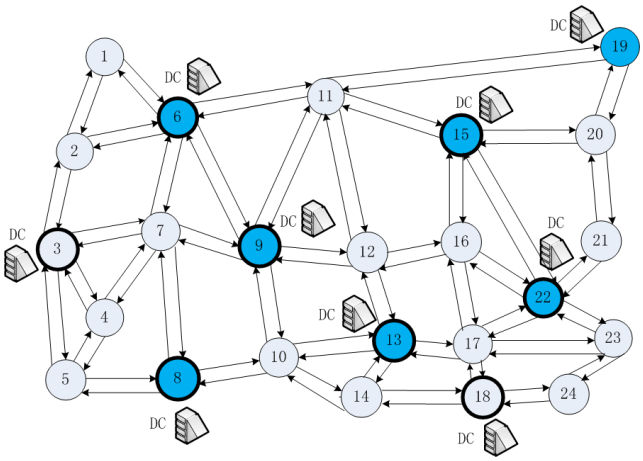


FIGURE 5. U.S. backbone topology.

Large enterprises such as Google can process about 100 PB data daily in geo-distributed application DCs [6]. Normally, not exceeding five percent of the daily data requires

backup [9]. Therefore, we can ensure that the results obtained using our parameter settings accord with the actual situation. In simulations, we consider processing delay as 10 μs, propagation delay as 5 μs per kilometer, and switch configuration delay as 15 ms [34]. In our proposed algorithms (i.e., BA-ACO in [28], BRA-ACO, and FRRA-ACO), we set the maximum iteration number as 50 and the ant number as 50 at the beginning. If there is no evolution in five consecutive iterations, we will stop the iteration.

B. SIMULATION RESULTS

We compare our algorithms with other three algorithms from the aspects of backup completion time, network transmission capability utilization, and backup load distribution.

1) BACKUP COMPLETION TIME

a) In Fig. 6, we illustrate the comparison of backup completion time with increasing number of nodes (including intermediate nodes, application DCs and backup DCs) using Waxman topology model. In MF-NORC, TwoStep-ILP and BA-ACO, every piece of primary data only has one replica while the required replica number is $k_{rp} = 3$ in BRA-ACO and FRRA-ACO. For a fair performance comparison, we modify the replica number to be three in the first three algorithms to make the total amount of data consistent, and set the total amount of primary data to be 1 PB.

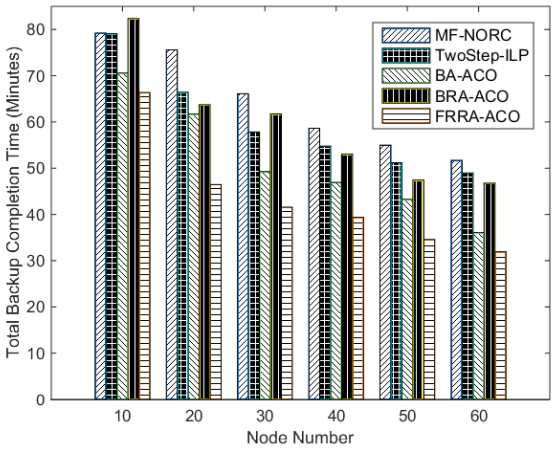


FIGURE 6. Comparison of backup completion time with increase of node number.

When the node number is relatively small (e.g., less than 20), the performance of MF-NORC is still acceptable, even better than BRA-ACO. That's because MF-NORC tries to maximize the network flow for every transfer one by one, whereas it is difficult to allocate reasonable ratios of bandwidth for concurrent transfers in BRA-ACO due to the poor choices of appropriate intermediate nodes and links. But as the increase of node number and link number, BA-ACO and BRA-ACO outperform MF-NORC more and more obviously, because they make better use of network transmission capability through more reasonable bandwidth allocation for concurrent transfers. TwoStep-ILP always performs better than MF-NORC because of its global view of least hop-counts and maximum backup throughput. FRRA-ACO performs better than other four algorithms because it optimizes the utilization of network transmission capability by reasonable and fair bandwidth ratios for concurrent transfers with a global view. In FRRA-ACO, we use rotary routing search to provide bandwidth supplement for unlucky transfer(s) to avoid hysteresis phenomenon of backup activity, and use the final flow rate improvement to make further use of residual network transmission capability as much as possible, leading to better performance.

b) In Fig. 7, we illustrate the comparison of backup completion time with increase of data amount in the U.S backbone topology. As in the Waxman topology model, we set replica number as three uniformly in these five algorithms.

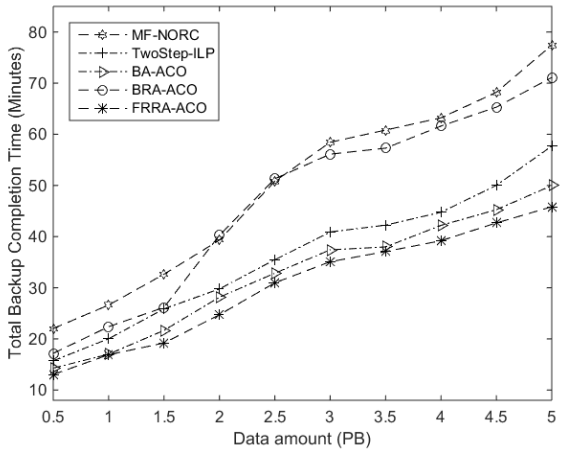


FIGURE 7. Comparison of backup completion time with increase of data amount.

Of course, backup completion time values of all these algorithms are rising while data amount increases. We can obtain more apparent growth in MF-NORC because it computes maximum flow and transfers backup data for every application DC one by one, resulting in some available link capacity lies idle in this mode (as the example shown in Fig. 1). TwoStep-ILP performs better with less completion time than MF-NORC because it takes all the data in an application DC as a whole and transfers the data to a nearest available destination to create a replica. But placing data to backup DC in nearby location does not mean full utilization of transmission capability. Instead, the optimal transmission solution may be missed by this correspondence. For BRA-ACO, completion time grows much faster than BA-ACO and FRRA-ACO, because the network transmission capability is not fully explored in its routing search process for concurrent transfers. By contrast, the rotary routing search mechanisms in BA-ACO and FRRA-ACO improve network transmission capability utilization and lead to less completion time, especially in FRRA-ACO with both further bandwidth allocation adjustment and final flow rate improvement.

2) NETWORK UTILIZATION

To compare the network transmission capability utilization, we compute network utilization. We do not intend considering MF-NORC and TwoStep-ILP, because they both use the maximum flow algorithm whereas higher network utilization does not necessarily mean more efficient transmission of backup data as shown in Fig. 3 and Table I. We aim at the comparison among BA-ACO, BRA-ACO and FRRA-ACO, because the three algorithms introduce proportional bandwidth allocation for concurrent backup transfers and the network utilization directly reflects the usage of network transmission capability.

We first compute *MaxFlow* from application DCs to their backup DCs. And then, we run BA-ACO, BRA-ACO and FRRA-ACO respectively to get their throughput as $Throughput_{BA}$, $Throughput_{BRA}$ and $Throughput_{FRRA}$. The NT [32] for three algorithms are defined as follows:

$$NT_{BA} = Throughput_{BA} / MaxFlow \tag{29}$$
$$NT_{BRA} = Throughput_{BRA} / MaxFlow \tag{30}$$
$$NT_{FRRA} = Throughput_{FRRA} / MaxFlow \tag{31}$$

a) In Fig. 8, we represents the comparison of NT among BA-ACO, BRA-ACO, and FRRA-ACO with increase of node number in the Waxman topology model.

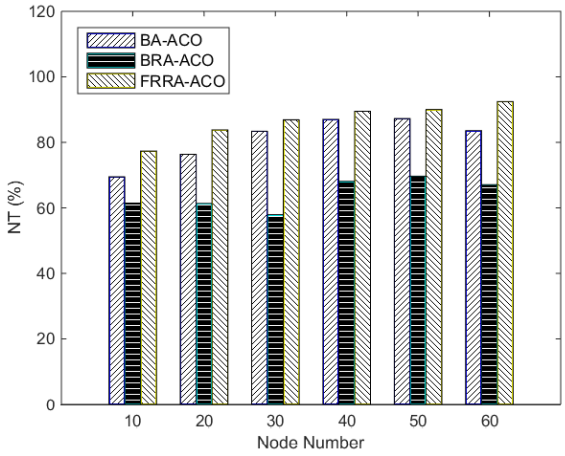


FIGURE 8. Comparison of NT with increase of node number.

BA-ACO outperforms BRA-ACO because the rotary routing search reduces idle bandwidth and therefore improves network transmission capability utilization. FRRA-ACO performs even better because the bandwidth allocation adjustment after multipath searching makes more reasonable bandwidth ratio approximately to data amount ratio of every backup transfer and final flow rate improvement plays a positive role of further optimization, leading to higher NT than the other two algorithms.

b) In Fig. 9, we illustrate the comparison of NT among BA-ACO, BRA-ACO, and FRRA-ACO with increase of data amount in the U.S. backbone topology.

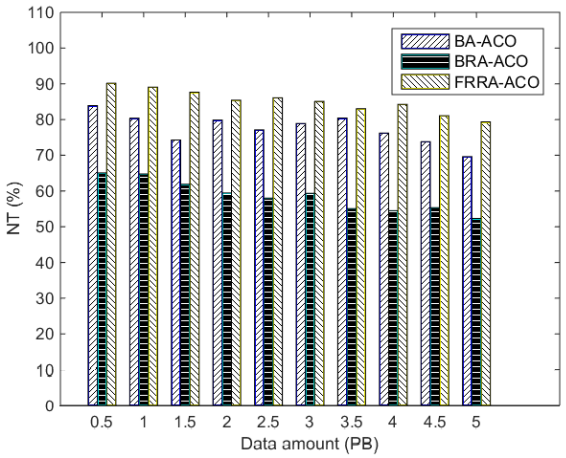


FIGURE 9. Comparison of NT with increase of data amount.

The less average amount of data carried by a single backup transfer, the more we can utilize network transmission capability by reasonable customized bandwidth allocation and flow control. The fixed network topology and transfer number lead to relatively small fluctuation of NT. But the gap between their NT still can be observed. The NT of BRA-ACO is relatively low, ranging from 52.30% to 65.05%. BA-ACO obtains relatively higher NT ranging from 69.62% to 83.89%. FRRA-ACO obtains even higher NT ranging from 79.34% to 90.17%. The comparison results in Fig. 7 and Fig. 9 jointly demonstrate that for bandwidth allocation to multiple backup transfers simultaneously under defined ratio constraint, the strategy with higher NT provides better transmission capability. This phenomenon also verifies the necessity of flow rate improvement in FRRA-ACO.

3) BACKUP LOAD DISTRIBUTION

To compare load distribution fairness for disaster backup, we first define the fair load distribution factor with γ as follows:

$$\gamma = 1 - \sqrt{\frac{1}{|\bigcup_{i=1}^{n} M_i| - 1} \sum_{d \in \left(\bigcup_{i=1}^{n} M_i\right)} (\gamma_d - \overline{\gamma})^2} \tag{32}$$

$$\gamma_d = \left(\sum_{BT_i \in BT} \sum_{bt_{ij} \in BT_i} x_{bt_{ij},d} \cdot dm_{ij} \right) \Big/ sc_d, \quad \forall d \in M \tag{33}$$

$$\overline{\gamma} = \frac{1}{|\bigcup_{i=1}^{n} M_i|} \sum_{d \in \left(\bigcup_{i=1}^{n} M_i\right)} \left(\left(\sum_{BT_i \in BT} \sum_{bt_{ij} \in BT_i} x_{bt_{ij},d} \cdot dm_{ij} \right) \Big/ sc_d \right) \tag{34}$$

We use γ_d to denote the ratio of total received data amount in a destination node d to its own backup receiving capacity. We use $\overline{\gamma}$ to denote the average value of γ_d for all

destination nodes. We compute $\sqrt{\frac{1}{|\bigcup_{i=1}^{n} M_i|-1} \sum_{d\in\left(\bigcup_{i=1}^{n} M_i\right)} (\gamma_d - \overline{\gamma})^2}$ as the standard deviation of γ_d for all destination nodes. In experiments, the value of standard deviation is relatively small, even approximating 0 in FRRA-ACO. Therefore, for the convenience of computation and comparison, we define γ in (32) to express the fairness of backup load distribution. Obviously, in these algorithms, greater value of γ implies better balance of backup load distribution among all destination nodes.

And then, we run algorithms to get their fair load distribution factor as γ_{MF}, γ_{ILP}, γ_{BA}, γ_{BRA} and γ_{FRRA}. To obtain greater clarity about the comparison of backup load distribution fairness among these algorithms, we introduce fairness ratio and define it as the ratio value of fairness between two different algorithms. We calculate the fairness ratio of FRRA-ACO to other four algorithms as follows:

$$fr_{FRRA-MF} = \gamma_{FRRA} / \gamma_{MF} \tag{35}$$

$$fr_{FRRA-ILP} = \gamma_{FRRA} / \gamma_{ILP} \tag{36}$$

$$fr_{FRRA-BA} = \gamma_{FRRA} / \gamma_{BA} \tag{37}$$

$$fr_{FRRA-BRA} = \gamma_{FRRA} / \gamma_{BRA} \tag{38}$$

a) In Fig. 10, we illustrate the comparison of fairness ratio with increasing node number in the Waxman topology model.

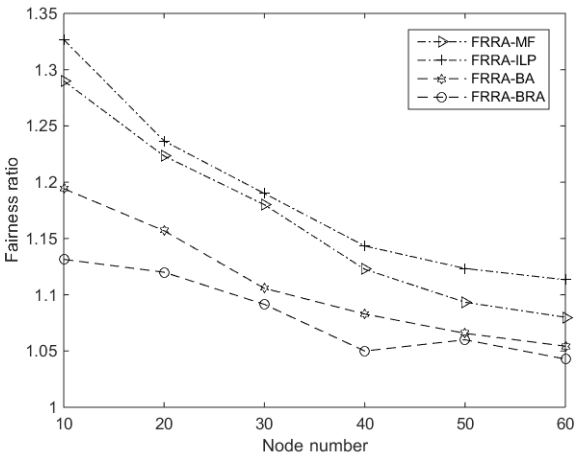


FIGURE 10. Comparison of fairness ratio with increase of node number.

Implemented in original network model, MF-NORC and TwoStep-ILP have not considered backup load balance in backup DCs and they tend to store backup loads nearby to reduce hop number in data transmission process, therefore resulting in significant imbalance of backup load distribution. BRA-ACO performs better in load distribution fairness than BA-ACO, benefiting from its further consideration of storage capacity in super backup DCs which is conducive (especially by pheromone updating) to assign backup loads in a balanced way. But on the other hand, as shown from Fig. 6 to Fig. 9, the performance of BRA-ACO in backup completion time and NT is worse than that of BA-ACO for lack of rotary routing search. Furthermore, FRRA-ACO performs even better than BRA-ACO because it jointly considers storage capacity in super backup DCs and rotary routing search in bandwidth allocation process.

b) In Fig. 11, we illustrate the comparison of fairness ratio with increase of data amount in the U.S. backbone topology.

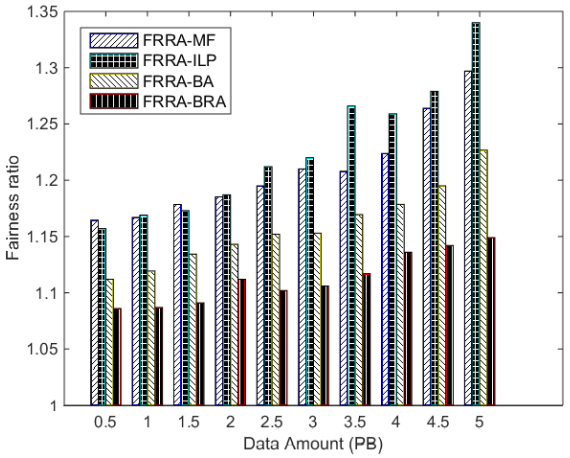


FIGURE 11. Comparison of fairness ratio with increase of data amount.

The rising trend of ratio curves implies that the advantage of FRRA-ACO in backup load distribution fairness becomes more and more obvious over other four algorithms. That's because larger data amount in transfers has greater impact on the change of load ratio in backup DCs, especially for MF-NORC and TwoStep-ILP without consideration of backup load balance. The growth trend of $fr_{FRRA-BRA}$ is slower than that of $fr_{FRRA-BA}$, benefiting from load distribution optimization by the specification of FRC transmission model in BRA-ACO. FRRA-ACO performs better than other four algorithms in load distribution fairness because of its comprehensive consideration of rotary routing search, further bandwidth ratio adjustment, and final flow rate improvement.

VI. CONCLUSION

In this paper, we investigate redundancy-guaranteed and receiving-constrained disaster backup among geo-distributed DCs in SDN. Based on problem formulation and modified network model, we further propose FRRA-ACO algorithm to obtain higher performance in data redundancy guarantee and enhance bandwidth allocation fairness. The innovation points of FRRA-ACO mainly embody in rotary routing search based on requirement cloning, reasonable proportional bandwidth allocation, further flow adjustment, and final flow rate improvement. We perform simulations to verify the superior performance of FRRA-ACO over other algorithms.

In further study, we will try to improve the strategy by reducing bandwidth consumption cost and relieving load pressure on critical paths.

ACKNOWLEDGMENT

We especially thank the editors and the reviewers for their valuable comments and suggestions.

REFERENCES

[1] Q. Xia, Z. Xu, W. Liang, and A. Y. Zomaya, "Collaboration- and fairness-aware big data management in distributed clouds," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 7, pp. 1941–1953, Jul. 2016.

[2] Z. Xu and W. Liang, "Operational cost minimization of distributed data centers through the provision of fair request rate allocations while meeting different user SLAs," *Comput. Netw.*, vol. 83, pp. 59–75, Jun. 2015.

[3] D. Applegate, A. Archer, V. Gopalakrishnan, S. Lee, and K. K. Ramakrishnan, "Optimal content placement for a large-scale VoD system," *IEEE/ACM Trans. Netw.*, vol. 24, no. 4, pp. 2114–2127, Aug. 2016.

[4] S. Agarwal, J. Dunagan, N. Jain, S. Saroiu, A. Wolman, and H. Bhogan, "Volley: Automated data placement for geo-distributed cloud services," in *Proc. CoNEXT*, San Jose, CA, USA, 2010, pp. 17–32.

[5] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: Research problems in data center networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 68–73, 2009.

[6] J. Yao, P. Lu, L. Gong, and Z. Zhu, "On fast and coordinated data backup in geo-distributed optical inter-datacenter networks," *J. Lightw. Technol.*, vol. 33, no. 14, pp. 3005–3015, Jul. 15, 2015.

[7] S. Ferdousi, M. Tornatore, M. F. Habib, and B. Mukherjee, "Rapid data evacuation for large-scale disasters in optical cloud networks," *J. Opt. Commun. Netw.*, vol. 7, no. 12, pp. B163–B172, Dec. 2015.

[8] N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez, "Inter-datacenter bulk transfers with netstitcher," in *Proc. ACM SIGCOMM*, Toronto, ON, Canada, 2011, pp. 74–85.

[9] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[10] Y. Wang, S. Su, A. X. Liu, and Z. Zhang, "Multiple bulk data transfers scheduling among datacenters," *Comput. Netw.*, vol. 68, pp. 123–137, Aug. 2014.

[11] A. Bianco, L. Giraudo, and D. Hay, "Optimal resource allocation for disaster recovery," in *Proc. GLOBECOM*, Miami, FL, USA, Dec. 2010, pp. 1–5.

[12] R. S. Couto, S. Secci, M. E. M. Campista, and L. H. M. K. Costa, "Server placement with shared backups for disaster-resilient clouds," *Comput. Netw.*, vol. 93, pp. 423–434, Dec. 2015.

[13] P. Lu, Q. Ling, and Z. Zhu, "Maximizing utility of time-constrained emergency backup in inter-datacenter networks," *IEEE Commun. Lett.*, vol. 20, no. 5, pp. 890–893, May 2016.

[14] X. Li, H. Wang, S. Yi, X. Yao, F. Zhu, and L. Zhai, "Optimizing concurrent evacuation transfers for geo-distributed datacenters in SDN," in *Proc. ICA3PP*, Helsinki, Finland, 2017, pp. 99–114.

[15] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, *Overview and Principles of Internet Traffic Engineering*, document RFC 3272, May 2002.

[16] I. F. Akyildiz *et al.*, "A new traffic engineering manager for DiffServ/MPLS networks: Design and implementation on an IP QoS testbed," *Comput. Commun.*, vol. 26, no. 4, pp. 388–403, Mar. 2003.

[17] J. Pang, G. Xu, and X. Fu, "SDN-based data center networking with collaboration of multipath TCP and segment routing," *IEEE Access*, vol. 5, pp. 9764–9773, Jun. 2017.

[18] C.-Y. Hong *et al.*, "Achieving high utilization with software-driven WAN," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 15–26, 2013.

[19] S. Jain *et al.*, "B4: Experience with a globally-deployed software defined WAN," in *Proc. ACM SIGCOMM*, Hong Kong, 2013, pp. 3–14.

[20] S. Ferdousi, F. Dikbiyik, M. F. Habib, M. Tornatore, and B. Mukherjee, "Disaster-aware datacenter placement and dynamic content management in cloud networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 7, no. 7, pp. 681–694, Jul. 2015.

[21] L. Ma, X. Jiang, B. Wu, A. Pattavina, and N. Shiratori, "Probabilistic region failure-aware data center network and content placement," *Comput. Netw.*, vol. 103, pp. 56–66, Jul. 2016.

[22] X. Li *et al.*, "Content placement with maximum number of end-to-content paths in K-node (edge) content connected optical datacenter networks," *J. Opt. Commun. Netw.*, vol. 9, no. 1, pp. 53–66, Jan. 2017.

[23] M. F. Habib, M. Tornatore, M. D. Leenheer, F. Dikbiyik, and B. Mukherjee, "Design of disaster-resilient optical datacenter networks," *J. Lightw. Technol.*, vol. 30, no. 16, pp. 2563–2573, Aug. 15, 2012.

[24] A. Zhou *et al.*, "Cloud service reliability enhancement via virtual machine placement optimization," *IEEE Trans. Services Comput.*, vol. 10, no. 6, pp. 902–913, Nov. 2017.

[25] L. Ma, X. Jiang, B. Wu, T. Talebi, A. Pattavina, and N. Shiratori, "Cost-efficient data backup for data center networks against ε-time early warning disaster," in *Proc. HPSR*, Yokohama, Japan, Jun. 2016, pp. 22–26.

[26] J. Yao, P. Lu, and Z. Zhu, "Minimizing disaster backup window for geo-distributed multi-datacenter cloud systems," in *Proc. ICC*, Sydney, NSW, Australia, Jun. 2014, pp. 3631–3635.

[27] P. Lu, L. Zhang, X. Liu, J. Yao, and Z. Zhu, "Highly efficient data migration and backup for big data applications in elastic optical inter-data-center networks," *IEEE Netw.*, vol. 29, no. 5, pp. 36–42, Sep./Oct. 2015.

[28] X. Li, H. Wang, S. Yi, and X. Yao, "Receiving-capacity-constrained rapid and fair disaster backup for multiple datacenters in SDN," in *Proc. ICC*, Paris, France, May 2017, pp. 1–6.

[29] M. Charikar and A. Karagiozova, "On non-uniform multicommodity buy-at-bulk network design," in *Proc. ACM STOC*, Baltimore, MD, USA, 2005, pp. 176–182.

[30] H. Masri, S. Krichen, and A. Guitouni, "An ant colony optimization metaheuristic for solving bi-objective multi-sources multicommodity communication flow problem," in *Proc. WMNC*, Toulouse, France, Oct. 2011, pp. 1–8.

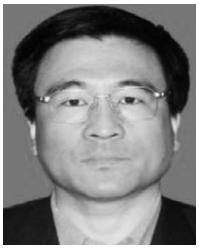
[31] M. Dorigo and G. Di Caro, "Ant colony optimization: A new meta-heuristic," in *Proc. CEC*, Washington, DC, USA, Jul. 1999, pp. 1470–1477.

[32] S. Cho, T. Elhourani, and S. Ramasubramanian, "Independent directed acyclic graphs for resilient multipath routing," *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 153–162, Feb. 2012.

[33] M. Naldi, "Connectivity of Waxman topology models," *Comput. Commun.*, vol. 29, no. 1, pp. 24–31, 2005.

[34] F. Dikbiyik, L. Sahasrabuddhe, M. Tornatore, and B. Mukherjee, "Exploiting excess capacity to improve robustness of WDM mesh networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 114–124, Feb. 2012.

XIAOLE LI received the master's degree from Guilin University of Electronic Technology, China, in 2007. He is currently pursuing the Ph.D. degree with Shandong University. His main research interests include network optimization, network algorithms, network intelligence, and network architecture and protocol.



HUA WANG received the Ph.D. degree from Nanjing University of Science and Technology, China, in 2003. From 2005 to 2008, he was a Post-Doctoral Fellow with the School of Computer Science and Technology, Shandong University, China. He is currently a Professor with the School of Software, Shandong University, where he leads the Network Optimization Research Group. His research interests include network optimization, network algorithms, network intelligence, network architecture and protocol, and network simulation. His research has been supported by the China Next Generation Internet Project, and the NSF of China.

SHANWEN YI received the master's degree from Shandong University, China, in 2010. He is currently pursuing the Ph.D. degree with Shandong University. His main research interests include network optimization, network algorithms, network intelligence, and network architecture and protocol.

XIBO YAO received the master's degree from Shandong University, China, in 2017. He is currently a Research and Development Engineer with ZhongTai Securities, China. His main research interests include network algorithms and network simulation.

FANGJIN ZHU received the Ph.D. degree from Shandong University, China, in 2011, where he is currently a Teacher. His current research interests include network optimization, intelligence algorithms, network measurement, network resource management, and network architecture and protocol.

LINBO ZHAI received the Ph.D. degree from Beijing University of Posts and Telecommunications, China, in 2010. From 2014 to 2017, he was a Post-Doctoral Fellow with the School of Computer Science and Technology, Shandong University, China. He is currently a Teacher with Shandong Normal University. His main research interests include cognitive radio, crowdsourcing, and distributed network optimization.