Received February 28, 2019, accepted March 22, 2019, date of publication April 11, 2019, date of current version April 22, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2909084*

# Disaster-and-Evacuation-Aware Backup Datacenter Placement Based on Multi-Objective Optimization

**XIAOLE LI [1], HUA WANG [2], SHANWEN YI [1], SHUAI LIU [3], LINBO ZHAI [4], AND CHUANQI JIANG [1]**

[1]School of Computer Science and Technology, Shandong University, Jinan 250100, China
[2]School of Software, Shandong University, Jinan 250100, China
[3]School of Computer, Beihang University, Beijing 100083, 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 under Grant NSFC 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** Backup datacenters provide massive data storage and access services, and their failure may result in huge economic losses. So their location selection requires low damage risk and high evacuation capability simultaneously. But previous works on backup datacenter placement have not jointly considered these two factors from the viewpoint of traffic engineering and might result in the unnecessary loss in case of disaster. In this paper, with the global view of network resources in the software defined network scenarios, we propose a new disaster-and-evacuation-aware backup datacenter placement strategy. To reduce backup loss risk and apply rapid post-disaster evacuation, we jointly consider expected disaster loss and evacuation latency and formulate a new disaster-and-evacuation-aware facility location problem (NP-hard) which is multi-objective. To obtain the solution according to the disaster situation assessment, we propose a disaster-and-evacuation-aware multi-objective optimization algorithm. We optimize multiple objectives owning different coefficients in different disaster situations. We introduce location-output-capability, backup-evacuation-latency, Pareto-recommendation-degree, and node-damage-loss to guide solution searching. We prune the external set according to fitness-deviation-ratio to improve convergence speed and computation efficiency of the algorithm. Through extensive simulations, we demonstrate that our algorithm is efficient and promising with less expected disaster loss and higher evacuation capability simultaneously.

**INDEX TERMS** Disaster-and-evacuation-aware facility location, multi-objective optimization, expected disaster loss, evacuation capability.

## I. INTRODUCTION

More and more geographically distributed datacenters are mega-centers of computing and storage resources and becoming increasingly important components to support various wide spreading cloud computing services [1], [2]. In order to obtain sufficient data redundancy and provide safe and reliable storage for critical information and applications, we need to leverage periodic data backup in some specified backup datacenters [3]. Consisting of such massive high-value data, backup datacenters are facing more and more potential large scale disasters (e.g., weapons of mass destruction (WMD) attacks, earthquakes, hurricanes, etc.) and their failure may result in huge economic loss [4], [5]. For instance, the Tohoku earthquake and tsunami in 2011 caused many companies to file bankruptcy due to critical backup data loss in enterprise datacenters [3]; in 2012, cascading failures caused by Hurricane Sandy damaged some backup datacenters in the Northeastern US [6]. Therefore, to prevent data damage in case of disaster, backup datacenter location selection problem is of great significance [7], and the disaster tolerance capability should be fully considered.

From the viewpoint of disaster tolerance, disaster risk and evacuation capability are two important issues that need to be

The associate editor coordinating the review of this manuscript and approving it for publication was Siddhartha Bhattacharyya.

carefully considered in backup datacenter location selection. The disaster risk is usually evaluated by expected disaster loss [7], [8]. Given a set of geographically distributed datacenters as candidate locations, we present risk analysis to estimate how much, in terms of cost or penalty, the cloud datacenter network might lose probabilistically in case of a possible disaster. Therefore, a reasonable strategy should deploy backup datacenters in the zones of least expected disaster loss to reduce potential damage in case of disaster. On the other hand, when disaster(s) occurs, the evacuation activity should transfer the data as soon as possible from the endangered datacenters in disaster zones to other datacenters in safe zones within evacuation deadline (e.g., before the depletion of uninterruptible power supplies). But unfortunately, there is still no effective evaluation metric to predict post-disaster evacuation capability for datacenters from the viewpoint of traffic engineering in face of disasters that may occur at any time. In this paper, we use evacuation latency to denote the time needed to evacuate all the data from disaster zones to safe zones. For the same amount of endangered data, lower latency means more efficient evacuation activity. So we use the evacuation latency to evaluate evacuation capability, and aim to guarantee low latency for endangered backup datacenters to evacuate bulk data once disaster(s) occurs. Therefore, in order to obtain reliable disaster-resistant backup datacenter placement strategy, we should jointly consider disaster risk distribution and evacuation latency. It is worth noting that when faced with different disaster situations, the impact and importance of these two factors may change significantly. In the case of frequent disasters, we should mainly focus on disaster risk distribution to reduce expected disaster loss in emergency backup activity [7], [8]. But in the case of severe disasters with low probability of occurrence, we should pay more attentions to improving evacuation capability. And we should also take into account that lower evacuation latency in location selection of backup datacenters will also help to reduce backup time and backup cost in regular backup activity [3], [9]. Therefore, we should adopt optimization correspondingly to deal with multiple objectives owning different coefficients in different disaster situations.

For properly optimizing the backup datacenter placement strategy in the large-scale cloud datacenter network, we still need a new network paradigm supporting global view of network resources. Fortunately, the Software Defined Network (SDN) matches our requirement well. SDN is becoming the leading technology behind many traffic engineering solutions both for backbone network and datacenter network [10], [11]. In our earlier works about traffic engineering for disaster backup and evacuation [6], [9], [12], we leverage SDN to construct powerful control environment for network resource management based on centralized visibility including global network information (e.g., network resource limitations or dynamically changing the network status) and global application information (e.g., quality of service requirements). In this paper, we still choose the SDN scenarios as backup datacenter placement research environment.

## II. RELATED WORKS

### A. DISASTER-RESISTANT DATECENTR PALCEMENT

For disaster-resistant datacenter placement, some new mechanisms have been proposed, mainly including resource allocation for backup storage [13], shared location with least backup datacenters [10], content placement and management to provide high disaster survivability with less expected loss [7], minimum failure probability [14], high content connectivity and lower resource consumption [15], backup path selection and content replica placement for disaster survivability [16], [17], minimum expected data loss with limited primary-to-backup distance [18]–[20].

In [13], Bianco *et al.* discuss resource allocation algorithms to support remote backup storage and live virtual machines migration. They present algorithms trading-off the minimization of the maximum number of hops between every virtual machine and its backup disk, and the minimization of the overload caused by site failures on backup sites chosen to host virtual machines after migration. In [10], Couto *et al.* propose a strategy to place datacenters in geographically distributed areas avoiding simultaneous failure of backup and primary servers. They try to reduce the amount of required backup datacenters by using virtualization. [7], [14] and [15] study placement and management of contents and their replicas among multiple datacenters considering disaster risks. In [7], Ferdousi *et al.* propose a disaster-aware datacenter placement and content management strategy to mitigate disaster loss by avoiding placement of contents and their replicas in given disaster vulnerable locations. In [14], Ma *et al.* consider the placement of datacenters and contents for datacenter failure probability minimization against a region failure. In [15], Li *et al.* define k-node (edge) content connectivity to measure reachability of content from any point of a network after disaster failures and apply it to optical datacenter networks. In [16], Habib *et al.* consider content placement, routing, and protection of paths and content together. This objective tends to place a content in those datacenters closer to its popular region, which can reduce resource usage by primary and backup paths while routing connection requests. But to simplify the model, they have not considered the constraints on storage and computing capacity of datacenters. In [17], Zhou *et al.* note that appropriate virtual machine placement could save considerable amount of time and network resources in failure recovery mode. They aim at reducing the lost time and the network resource consumption when the k-fault-tolerance requirement must be satisfied, to reduce network resource consumption in addition to enhancing cloud service reliability. However, these researches above focus on a single criterion, and none of them has jointly considered disaster risk distribution and evacuation capability for backup datacenter placement in different disaster situations. When determining backup datacenter location, the ignorance of evacuation capability might result in huge backup data loss in case of disaster.

Some works involve multi-objective function with expected data loss and evacuation capability [18]–[20]. However, there are still some challenges. When determining the location of primary-backup pairs, they limit primary-to-backup distance to facilitate the transmission by manpower or vehicles. They have not considered available network transmission capability which is essential or even preferred way of data evacuation in case of disasters [6], [21]. And it's worth noting that closer physical location between primary-backup pair does not always mean more efficient post-disaster evacuation, especially for multiple geographically distributed datacenters [6], [9], [12]. In addition, they leverage primary-to-backup distance as evacuation capability constraint, without optimizing it according to different disaster situations. And therefore, the obtained solution may not be the most desired one with less expected data loss and higher evacuation capability simultaneously.

### B. OUR SOLUTIONS

Multi-objective optimization in evolutionary algorithms usually uses population-based approach to find Pareto optimal solutions. The majority of existing works to deal network problems use the concept of dominance during selection [22]–[24]. In this paper, we leverage multi-objective optimization to obtain disaster-and-evacuation-aware backup datacenter placement solution applicable for disaster-and-evacuation-aware scenarios. Our contributions can be summarized as follows:

- We propose new evaluation metric jointly considering expected disaster loss and evacuation capability for backup datacenter placement in SDN, and as far as we know, this is the first work to proactively optimize post-disaster data evacuation capability from the viewpoint of traffic engineering in disaster-aware backup datacenter placement phase.
- We add expected disaster loss and evacuation capability into facility location problem, and propose a new Disaster-and-Evacuation-Aware Facility Location (DEA-FL) problem which is NP-hard.
- We design a Disaster-and-Evacuation-Aware Multi-Objective Optimization (DEA-MO) algorithm, which sets unique pheromone and heuristic information for every backup datacenter, and introduces location-output-capability, backup-evacuation-latency, Pareto-recommendation-degree and node-damage-loss to guide solution searching. We optimize multiple disaster backup objectives owning different coefficients in different disaster situations. This algorithm is applicable for practical networks of large-scale.
- By extensive simulations we demonstrate that our algorithm achieves good performance in terms of reducing total expected disaster loss and implementing more efficient data evacuation in case of disaster compared with the state-of-the-art algorithms.

The rest of the paper is organized as follows. In Section III, we give an overview of disaster-and-evacuation-aware backup datacenter placement, and formulate the DEA-FL problem. In Section IV, we design DEA-MO algorithm to solve the DEA-FL problem. In Section V, we evaluate the performance of our solution through extensive simulations. At last, we draw our conclusion in Section VI.

## III. PROBLEM FORMULATIONS

The facility location problem with an input for $m$ backup datacenters to place in $n$ candidate locations, is NP-hard [25], [26]. As shown in Section I, expected disaster loss and evacuation capability are two key parameters in the backup datacenter placement process. To minimize expected disaster loss and improve evacuation capability, we should add them into facility location problem and therefore obtain a new DEA-FL problem. As a special case of facility location problem, the DEA-FL problem is also NP-hard.

In the disaster-and-evacuation-aware backup datacenter placement process, we propose analysis on expected disaster loss in a given network with a set of candidate locations to estimate how much a network operator might lose probabilistically in case of a possible disaster and define it as expected disaster loss. Risk maps of disasters can be obtained and matched with the physical topology of a network to determine its possible risky zones [8], [27]. For instance, according to the information on possible locations of different major facilities from various public sources in [7], we generate a simple risk map for attacks on datacenter nodes in a US-Backbone topology by considering possible attacks in Fig. 1 to help us to develop and test disaster-aware backup datacenter placement. Besides, for post-disaster evacuation, we also consider the evacuation capability in term of evacuation latency between endangered backup datacenters and the application datacenters within their functioning ranges. From the viewpoint of traffic engineering, even though network traffic is dynamic, lower evacuation latency is always beneficial to shorten evacuation time of bulk backup data among datacenters. After the determination of backup datacenter location, we compute the latency value from backup datacenters to their application datacenters by the transmission strategy proposed in our earlier works [6], [9], [12] considering the scenarios of both regular backup and emergency backup.

We consider DEA-FL problems on a network with a symmetric directed graph $G = (V, E)$, where $V$ is the set of nodes, and $E$ is the set of physical links between nodes. We denote the link from node $u$ to node $v$ as $(u, v)$.

We use $DC = \{dc_1, dc_2, \ldots, dc_n\}$ to denote the set of application datacenters with various contents to backup. It is worth noting that in the cloud datacenter network, some datacenter nodes always play dual roles as application datacenter and backup datacenter [3], [9], [12]. Therefore, we can consider any application datacenter node as candidate location for backup datacenter placement. We use $BD = (bd_1, bd_2, \ldots, bd_m)$ to denote the set of backup datacenters that should be placed in the network $G$. We assume that

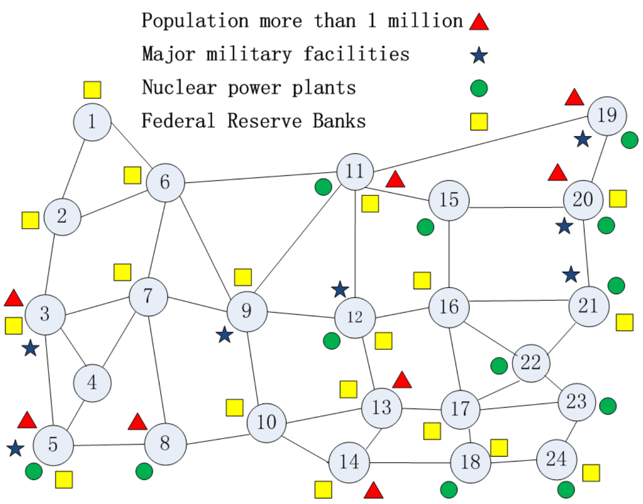



**FIGURE 1.** Simple risk map for attacks on datacenters in US-Backbone topology.

backup datacenter $bd_i$ has a maximum backup load capacity $C(bd_i)$ and the total amount of backup contents in the application datacenter nodes (here we call them backup loads) within the functioning range of $bd_i$ cannot exceed this limit. $\theta(bd_i)$ is used to denote the set of all the application datacenters within the functioning range of the individual backup datacenter $bd_i$. Because the datacenter nodes are always geographically distributed, they usually fall into different disaster areas. Then the whole network $G$ is divided into $m$ clusters: $\theta(bd_1)$, $\theta(bd_2), \ldots, \theta(bd_m)$. The clusters should satisfy the following constraints:

$$\theta(bd_i) \neq \phi, \quad \forall bd_i \in BD \tag{1}$$

$$\cup_{i=1}^{m}\theta(bd_i) = DC \tag{2}$$

$$\theta(bd_i) \cap \theta(bd_j) = \phi, i \neq j, \quad \forall bd_i,\ bd_j \in BD \tag{3}$$

Using (1), we ensure that every $bd_i$ plays the role of backup datacenter. Using (2), we ensure that the set of backup datacenters cover all application datacenters. Using (3), we ensure that every application datacenter belongs to only one backup datacenter.

We use $dc_j \in \theta(bd_i)$ to represent that the application datacenter node $dc_j \in DC$ is within the functioning range of $bd_i$. We use $bl(dc_j)$ to represent the backup load of $dc_j$ , and use $bl(bd_i) = \sum_{dc_j \in \theta(bd_i)} bl(dc_j)$ to represent the total backup load of $bd_i$. In a partition cluster we should make sure that:

$$bl(bd_i) \leq C(bd_i), \quad \forall bd_i \in BD \tag{4}$$

Using (4), we ensure that the backup loads given by the application datacenter nodes within the functioning range of $bd_i$ should not exceed $bd_i's$ backup load capacity.

To prevent data damage in case of disaster, we add expected disaster loss and evacuation capability to backup datacenter location selection process. From the viewpoint of expected disaster loss, we aim to deploy backup datacenters in least risk zones to avoid data damage as much as possible. We define $loss_{avg}$ as average expected disaster loss of backup datacenters and $loss_{bd_i}$ as expected disaster loss of $bd_i$ in the following:

$$loss_{avg} = \frac{1}{m} \sum_{bd_i \in BD} loss_{bd_i} \tag{5}$$

$$loss_{bd_i} = \sum_{dc_j \in \theta(bd_i)} bl(dc_j) \cdot \sum_{v \in V} \left(x_v^{bd_i} \cdot P_v\right), \quad \forall bd_i \in BD \tag{6}$$

$$P_v = \begin{cases} 1 & \text{if} \sum_{s \in S} P_v^s \geq 1 \\ \sum_{s \in S} P_v^s & \text{otherwise} \end{cases}, \quad \forall v \in V \tag{7}$$

$$P_v^s = \frac{intensity_s}{dist(v, s)}, \quad \forall v \in V, \ \forall s \in S \tag{8}$$

$$\sum_{v \in V} x_v^{bd_i} = 1, \quad \forall bd_i \in BD \tag{9}$$

Here the $P_v^s$ denotes the probability that node $v$ is damaged by the disaster $s$. The $x_v^{bd_i} \in \{0, 1\}$ denotes whether $bd_i$ is placed in $v$. The $P_v^s$ is directly proportional to the intensity of disaster $s$ (denoted as $intensity_s$), and inversely proportional to the distance from $v$ to the center of $s$ (denoted as $dist(v, s)$). The $P_v = \sum_{s \in S} P_v^s$ means the sum of damage probability on $v$ by all disasters. We limit its value to no more than 1 by (7).

Once under the threat of disaster, massive data in backup datacenters, including application data and historical backup data, should be evacuated as soon as possible. Considering compatibility and availability of backup data in $bd_i$, it is a good choice to take $\theta(bd_i)$ as evacuation destinations in case of disaster. We focus on latencies among backup datacenters and the application datacenters within their functioning ranges. In (10), we define average evacuation latency as $laten_{avg}$ to evaluate evacuation capability of backup datacenters in case of disaster:

$$laten_{avg} = \frac{1}{m} \sum_{bd_i \in BD} time_{bd_i} \tag{10}$$

Here the $time_{bd_i}$ represents the evacuation time from $bd_i$ to the application datacenters within its functioning range. In the unpredictable situation of network traffic, lower $laten_{avg}$ expresses higher evacuation capability.

To jointly minimize the expected disaster loss and maximize evacuation capability, we can formulate the optimization objective of DEA-FL problem as follows:

$$minimize(loss_{avg}, laten_{avg}) \tag{11}$$

As shown in (11), we aim to simultaneously minimize the average expected disaster loss and average evacuation latency. However, the DEA-FL problem is NP-hard that cannot be solved in polynomial time. Owing to computational impracticality of exact algorithms to produce solutions for practical networks of large scale, we consider Multi-Objective Optimization algorithm based on ant colony optimization (ACO) [28] to improve time efficiency and obtain optimal or near-optimal solution within acceptable computing time in the following sections.

## IV. ALGORITHM DESIGN

ACO metaheuristic is inspired by operating principles of ants. Its central mechanism is to probabilistically construct solutions using a parameterized probability model, which is indicated by the pheromone trails. In the context of ACO, the solution component is usually associated with pheromone trail. Artificial ants probabilistically add the solution component to the partial solution until they generate a completely feasible solution. During these iterations, the pheromone values are dynamically updated based on the information derived from some high quality solutions to force the search to concentrate on regions containing high quality solutions in the solution space. It is a powerful algorithm to solve NP problems in the field of computing intelligence [28]. Here we design DEA-MO algorithm based on ACO to solve the DEA-FL problem.

To find the solution for backup datacenter location, we initially place every backup datacenter in a random datacenter node, respectively. To jointly minimize the expected disaster loss and maximize evacuation capability, we introduce location-output-capability, backup-evacuation-latency, Pareto-recommendation-degree and node-damage-loss to adjust the location selection searching for solution. By using external set, we can realize information sharing among different non-dominated solutions and guide the evolution of new solution.

### A. PHEROMONE AND HEURISTIC INFORMATION

In previous works such as [22], if there are multiple non-dominated solutions in the external set after the algorithm is finished, they will randomly choose one as optimal solution. However, the randomly selected solution might not be the most suitable one for specific disaster situation (e.g., in face of frequent disasters, or infrequent but severe disasters). Therefore, in DEA-MO, to deal with multiple objectives owning different coefficients in different disaster situations, we leverage factors $\omega_1$ and $\omega_2$ to express the importance of the parameters for expected disaster loss and evacuation capability, respectively. In the following paragraphs, we assign related parameters with $\omega_1, \omega_2 \in [0, 1]$ and $\omega_1 + \omega_2 = 1$. Solution $sl$ is represented by a $m \times n$ adjacency matrix. If $sl[i][j] = 1$, backup datacenter $bd_i$ is placed in the location $dc_j$, otherwise not. To obtain higher efficiency of location searching, we set unique pheromone and heuristic information for every $bd_i$. We define two normalized values $p_v^{'}$ and $cap_v^{'}$ as follows:

$$P_v^{'} = \frac{P_v}{\frac{1}{n}\sum_{v \in DC} P_v} \tag{12}$$

$$cap_v^{'} = \frac{\sum_{u \in Adj(v)} cap(v,u)}{\frac{1}{n}\sum_{v \in DC}\sum_{u \in Adj(v)} cap(v,u)} \tag{13}$$

Here the $adj(v)$ is the set of nodes connecting directly to node $v$. the $cap(v, u)$ denotes the available capacity of link $(v, u)$. We define the location-output-capability of $v$ as $\sum_{u \in adj(v)} cap(v, u)$, which means the sum of available capacity in all links from $v$ to $Adj(v)$. In DEA-MO, we set the initial pheromone intensity as $\tau_v^i(t_0) = \frac{\left(cap_v^{'}\right)^{\omega_2}}{1+\left(P_v^{'}\right)^{\omega_1}}$.

After the $tth$ iteration, we put additional pheromones on the nodes used by the best current solution $cur\_place$ and non-dominated solutions in external set. The update of pheromone for $bd_i$ placement on node $v$ is as follows:

$$\tau_v^i(t+1) = (1-\rho)\tau_v^i(t) + \Delta\tau_v(t) \tag{14}$$

$$\Delta\tau_v(t) = \lambda_1\chi(t) + \lambda_2\delta(t) \tag{15}$$

$$x_{cur}^{i,v} = \begin{cases} 1 & \text{if } bd_i \text{ is placed on node } v \text{ in } cur\_place \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

$$y_{es}^{i,v} = \begin{cases} 1 & \text{if } bd_i \text{ is placed on node } v \text{ in external set} \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

The $\rho$ represents evaporating parameter to control the evaporating speed of pheromone. The $\lambda_1$ and $\lambda_2$ express the influence of $cur\_place$ and external set on the increment of pheromone intensity respectively in the $(t+1)th$ iteration. The $bl(bd_i)$ expresses the total backup loads in $bd_i$. We use it to divide the location-output-capability to estimate the backup-evacuation-latency of $bd_i$. The $bl(bd_i) \cdot P_v$ denotes node-damage-loss of $v$. We define the node-damage-loss as an expected value, which means multiplying backup loads in $bd_i$ with the sum of damage probability on $v$ by all disasters in $S$. We prefer to choose the node with less node-damage-loss which means less expected disaster loss. It is worth noting that the value of backup-evacuation-latency and node-damage-loss might be different in (16) and (18), as shown at the top of the next page.

The $\mu_1$ and $\mu_2$ adjust the value of backup-evacuation-latency and node-damage-loss to express their influences on the increment of pheromone intensity. We define $(1 + sn)$ as the Pareto-recommendation-degree. The $sn$ represents the number of non-dominated solutions which place $bd_i$ on $v$ in external set.

The heuristic information on $v$ for $bd_i$ placement depends on location-output-capability and total disaster risk. And then, we can obtain heuristic information $\eta_v^i(t+1)$ for $bd_i$ placement as follows:

$$\eta_v^i(t+1) = \kappa.\frac{\left(\sum_{u \in Adj(v)} cap(v,u)\right)^{\omega_2}}{1+(P_v)^{\omega_1}} \tag{20}$$

Here the parameter $\kappa$ is used to adjust the value of heuristic information $\eta_v^i(t+1)$. We prefer to choose the node with larger location-output-capability and smaller total

$$\chi(t) = \frac{x_{cur}^{i,v}}{\mu_1 \cdot \left( bl(bd_i) / \sum_{u \in Adj(v)} cap(v,u) \right)^{\omega_1} + \mu_2 \cdot (bl(bd_i) \cdot P_v)^{\omega_2}} \tag{16}$$

$$\delta(t) = \frac{y_{es}^{i,v} \cdot (1+sn)}{\mu_1 \cdot \left( bl(bd_i) / \sum_{u \in Adj(v)} cap(v,u) \right)^{\omega_1} + \mu_2 \cdot (bl(bd_i) \cdot P_v)^{\omega_2}} \tag{18}$$

damage probability because the former will provide more available bandwidth in post-disaster evacuation and the latter guarantees low probability of being destroyed by disasters.

### B. TRANSITION PROBABILITY

In DEA-MO, while constructing a solution, an ant selects some node $v$ to place $bd_i$. The choice of $v$ depends on pheromone intensity $\tau_v^i(t+1)$ and heuristic information $\eta_v^i(t+1)$. Here we choose nodes using the roulette wheel selection procedure of evolutionary computation. The transition probability to $v$ for $bd_i$ placement is:

$$R_v^i(t+1) = \frac{(\tau_v^i(t+1))^{\phi} \cdot (\eta_v^i(t+1))^{\varphi}}{\sum_{w \in CN} (\tau_w^i(t+1))^{\phi} \cdot (\eta_w^i(t+1))^{\varphi}} \tag{21}$$

The parameters $\phi$ and $\varphi$ express the influence of pheromone and heuristic factors in transition probability respectively, and $CN$ denotes the candidate node set can be used to host backup datacenters under the current network status.

### C. FITNESS EVALUATION

The optimization process of multiple objectives may conflict with each other. Therefore, to obtain non-dominated solution, we need to evaluate solutions with different fitness functions according to different objectives respectively as follows.

$$fitness_{loss}(sl) = \frac{1}{1 + \left(loss_{avg}\right)^{\omega_1}} \tag{22}$$

$$fitness_{laten}(sl) = \frac{1}{\left(laten_{avg}\right)^{\omega_2}} \tag{23}$$

$fitness_{loss}(sl)$ is the fitness function related to expected disaster loss and its value is inversely proportional to the average expected disaster loss of backup datacenters. $fitness_{laten}(sl)$ is the fitness function related to evacuation capability and its value is inversely proportional to the average evacuation latency of backup datacenters.

According to fitness values, we will eliminate dominated solutions and reserve non-dominated solutions in external set. For example, with two solutions $sl_i$ and $sl_j$, $sl_i$ dominates $sl_j$ if $sl_i \prec sl_j$, and otherwise $sl_j$ dominates $sl_i$. If they do not dominate each other, they both will be added to external set. We define $sl_i \prec sl_j$ as follows.

$$\begin{aligned}
&\big(fitness_{loss}(sl_i) > fitness_{loss}(sl_j)\big) \textit{ and}\\
&\big(fitness_{laten}(sl_i) \geq fitness_{loss}(sl_j)\big)\\
&\textit{or}\\
&\big(fitness_{loss}(sl_i) \geq fitness_{loss}(sl_j)\big) \textit{ and}\\
&\big(fitness_{laten}(sl_i) > fitness_{loss}(sl_j)\big)
\end{aligned} \tag{24}$$

### D. EXTERNAL SET UPDATE

We use external set to store non-dominated solutions and coordinate different objectives. After every iteration, we compare fitness function values of solutions according to (22) and (23), select the non-dominated solution to be $cur\_place$. If there are multiple non-dominated solutions, we select one as $cur\_place$ randomly and add all non-dominated solutions into external set. Then we compare newly added solutions with original solutions in external set, and eliminate dominated solution(s) if any.

It is noteworthy that the number of solutions in external set increases with the increase of iteration number. In [22], they have not limited the capacity of external set, but simply remove the dominated solutions. However, too many elements (although all are non-dominated solutions) in the external set will result in slower convergence speed and lower computation efficiency. Therefore, we limit the capacity of external set and design new pruning strategy to remove redundant non-dominated solution(s) if necessary. We use the maximum number of non-dominated solutions that can be accommodated to denote the maximum capacity $Cap_{ES}$ of external set $ES$, and eliminate some solution(s) if the number of solutions exceeds $Cap_{ES}$ in $ES$. We sort all the solutions according to our newly defined metric called fitness-deviation-ratio (FDR) as follows:

$$FDR(sl) = \sqrt{\left(\frac{FD_{loss}(sl)}{avgfitness_{loss}(ES)}\right)^2 + \left(\frac{FD_{laten}(sl)}{avgfitness_{laten}(ES)}\right)^2} \tag{25}$$

$$FD_{loss}(sl) = \begin{cases} 0 & \text{if } fitness_{loss}(sl) \geq avgfitness_{loss}(ES) \\ avgfitness_{loss}(ES) - fitness_{loss}(sl) & \text{otherwise} \end{cases} \quad (26)$$

$$FD_{laten}(sl) = \begin{cases} 0 & \text{if } fitness_{laten}(sl) \geq avgfitness_{laten}(ES) \\ avgfitness_{laten}(ES) - fitness_{laten}(sl) & \text{otherwise} \end{cases} \quad (27)$$

$$avgfitness_{loss}(ES) = \frac{1}{Cap_{ES}} \sum_{sl \in ES} fitness_{loss}(sl) \quad (28)$$

$$avgfitness_{laten}(ES) = \frac{1}{Cap_{ES}} \sum_{sl \in ES} fitness_{laten}(sl) \quad (29)$$

For a solution $sl$, we use $\left(\frac{FD_{loss}(sl)}{avgfitness_{loss}(ES)}\right)$ and $\left(\frac{FD_{laten}(sl)}{avgfitness_{laten}(ES)}\right)$ to represent loss-deviation-ratio and latency-deviation-ratio, respectively. We use $FD_{loss}(sl)$ and $FD_{laten}(sl)$ to denote the fitness deviation of $sl$ relative to the average fitness value in $ES$ from the aspects of expected disaster loss and evacuation latency, respectively. We use $avgfitness_{loss}(ES)$ and $avgfitness_{laten}(ES)$ to denote the average value of $fitness_{loss}()$ and $fitness_{laten}()$ respectively for all non-dominated solutions in $ES$. To jointly reduce the expected disaster loss and improve evacuation capability, we would like to control $loss_{avg}$ and $laten_{avg}$ as small as possible simultaneously. The solution with larger FDR value certainly owns larger $loss_{avg}$ or larger $laten_{avg}$ than the average fitness value in $ES$. Therefore, when the number of solutions in $ES$ exceeds $Cap_{ES}$, we will eliminate the solution(s) with the largest FDR value until the solution number not exceeds $Cap_{ES}$. In the next section, we will verify the effectiveness of pruning strategy for the redundant non-dominated solution. In the following experiments, if there lie several non-dominated solutions in $ES$, we will choose the one with minimum FDR value as solution for comparison. If there are multiple solutions with the same minimum FDR value, we select one randomly.

### E. PSEUDO CODE OF DEA-MO

The pseudo code of DEA-MO algorithm is as follows:

In DEA-MO, we set the number of ants as $num$. For the placement of every backup datacenter, at most $num$ solutions are generated, so the time complexity of DEA-MO is approximately $O(n \cdot m \cdot num)$. Through extensive simulations, we get reasonable values of simulation parameters. Eventually, we set $\phi = 0.8, \varphi = 0.5, \rho = 0.25, \lambda_1 = 3, \lambda_2 = 7$ on the basis of experience.

**Algorithm 1** DEA-MO Algorithm

**Input**: $G = (V, E)$; $DC = \{dc_1, dc_2, \ldots, dc_n\}$; $BD = \{bd_1, bd_2, \ldots, bd_m\}$; probability distribution map of disaster events
**Output**: backup datacenter placement solution

Set parameters, initialize pheromone and transition probability, etc.
**while** termination condition not met **do**
  **for** $num$ ants **do**
    Initialize available location set $ALS = DC$
    **for** every $bd_i$ **do**
      Calculate heuristic information by (20)
      Select its location $v$ from $ALS$ using the roulette wheel selection procedure according to (21)
      Remove $v$ from $ALS$
    **end for**
    **for** every $dc_j$ **do**
      Find the shortest path to every backup datacenter
      Assign it to the nearest backup datacenter $bd_i$
      Compute backup load of $bd_i$
      **if** $bd_i$ exceeds its backup load capacity **do**
        Assign $dc_j$ to the next nearest backup datacenter
      **end if**
    **end for**
    Obtain non-dominated solution(s) in current population by (24), and select $cur\_place$
    **for** every non-dominated solution in current population **do**
      **if** it is not dominated by the solutions in external set **do**
        Add it into external set
        Eliminate solution(s) dominated by it from external set
      **end if**
    **end for**
    **if** the number of solutions exceeds $Cap_{ES}$ in external set **do**
      **repeat**
        Eliminate the solution with largest FDR value by (25)
      **until** the number of solutions not exceeds $Cap_{ES}$
    **end if**
    Update pheromone by (14)
  **end for**
**end while**

## V. PERFORMANCE EVALUATION

### A. ENVIRONMENT AND CONFIGURATION

We implement algorithms in a DELL OPTIPLEX 9020 server with 8 Intel(R) Core(TM) i7-4790 3.60 GHz CPUs and 8 GB

RAM. We perform experiments over the US-Backbone topology [29]. As in Fig. 2, it has 24 application datacenter nodes and we will place a certain number of backup datacenters. For convenient display, we only mark the distance between datacenter nodes. Due to limited network resources, some nodes may play dual roles as application datacenter and backup datacenter. Every datacenter is connected with high-bandwidth links. We set the available bandwidth uniformly distributed on each link within [500, 1000] (Gbps).

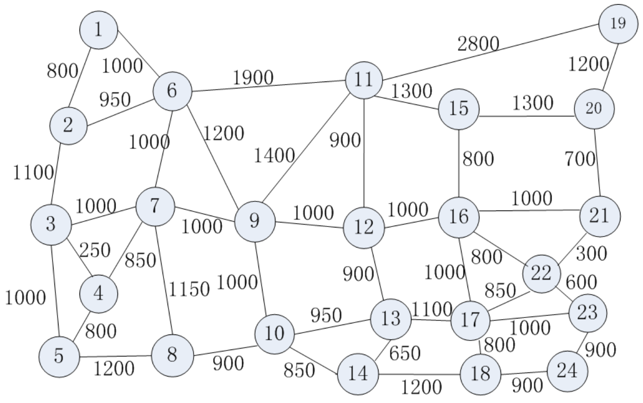


**FIGURE 2.** US-Backbone topology denoted with distance value.

From the viewpoint of disaster backup location, we consider disaster risk distribution on nodes. Similar to previous work [7], we consider the increasing disaster risk, such as WMD attacks, earthquakes, hurricanes [27]. To evaluate performance of these algorithms, we consider possible locations of disaster events according to risk map in Fig. 1, set the value of four risks as 0.25 respectively, and then generate 10 disaster instances. In every instance, we select 7 different datacenter nodes as the locations to place disaster events. Here we apply a pseudo random selection rule [30]. We generate a random number $q \in [0, 1]$. If $q \leq q_0$, we will choose the first 7 datacenter nodes with the largest total disaster risks. Otherwise, we choose 7 locations using roulette wheel selection procedure according to total disaster risks in every location. We set $q_0 = 0.7$ on the basis of experience. For every disaster instance, we model disaster events on datacenter nodes as in Fig. 3. We repeat performance evaluation with an independent run for every disaster instance, and report the average results for comparison among different algorithms.

To evaluate damages by disaster events, we consider not only primary disaster damages but also correlated effects [7]. Based on information in [31] and considering large-scale disaster and multiple correlated effects, we assume a failure span of 1000 kilometers around the targeted areas. The probabilities of failure on nearby datacenter nodes are estimated with reasonable assumptions (between 0 and 1) based on their distances from the target's epicenter. For example, in the case of disaster event $s_1$ (shown with corresponding disaster zone), node 3 is estimated to be damaged with probability 1, and node 4 and node 7 have estimated damage

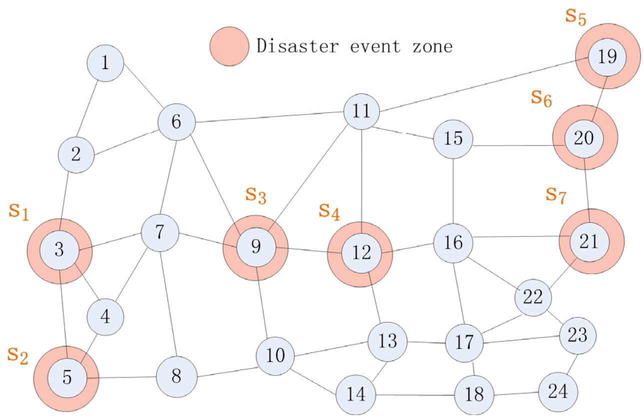



**FIGURE 3.** US-Backbone topology denoted with possible disaster events.

probabilities of 0.2 and 0.05, respectively (decreasing with the distance).

We compare DEA-MO with some representative algorithms. First, we choose the average distance limiting with Branch-and-Bound (ADL-BB) algorithm and the maximum distance limiting with Branch-and-Bound (MDL-BB) algorithm in [18]. To ensure rapid evacuation in case of disasters, they consider geographical distance between primary-backup pair instead of available network transmission capability. To evaluate evacuation capability over the US-Backbone topology denoted with possible disaster zones, we need to calculate evacuation latency with network parameters. Therefore, as in [3], we leverage hop number to represent the geographical distance in ADL-BB and MDL-BB. In the following experiments, we set the average hop number constraint as 3 for ADL-BB, and set the maximum hop number constraint as 5 for MDL-BB. For convenience of performance comparison, we modify the backup storage capacity setting and ensure that every backup datacenter can receive backup data from multiple application datacenters if it has enough storage capacity. Second, we choose the disaster-aware datacenter placement (DADP) algorithm [7] and the TwoStep-ILP algorithm [3]. DADP mitigates disaster loss by avoiding placement in given disaster vulnerable locations with the objective to minimize failure risk. But it has not considered evacuation capability in case of disasters. TwoStep-ILP minimizes total hop number between all the application datacenters and their backup datacenters in one-to-one mutual backup model. But it has not considered disaster risk distribution to reduce expected disaster loss. In experiments, we leverage its first integer linear programming to determine backup datacenter locations.

For DEA-MO, we can dynamically assess different disaster situations and observe the influence of coefficients on solutions by changing values of $\omega_1$ and $\omega_2$. With larger $\omega_1$, we will focus on the expected disaster loss and therefore obtain higher data integrity. With larger $\omega_2$, we will pay more attentions to evacuation capability of backup datacenters. In that case, DEA-MO prefers to choose the nodes with higher transmission capacity to cover application datacenters, not the ones with low disaster risk but lie in out-of-the-way locations. Such choices will surely reduce the data integrity factor, especially as the increase of backup datacenter number. In practice, we can dynamically set values of $\omega_1$ and $\omega_2$, and search solutions according to different disaster-resistant scenarios. Especially in the backup activity faced with frequent disasters, we can increase the value of $\omega_1$ to reduce expected disaster loss. Of course, backup datacenter placement strategy based on such parameter setting might lead to longer backup time and higher backup cost in regular backup activity. If $\omega_2$ plays a more significant role, DEA-MO will perform better in reducing evacuation latency. In regular backup activity faced with infrequent disasters, we can increase the value of $\omega_2$ to reduce backup time. Furthermore, we can also optimize backup cost effectively if the evacuation latency is replaced by unit cost in the solution searching process.

In the following experiments, we observe algorithm performance with increase of backup datacenter number and backup load amount, respectively. In the former case, we set the total backup load amount as 1.5 PB, set the amount of backup data in every application datacenters ranging from 50 TB to 100 TB, set the maximum load capacity in every backup datacenter ranging from 100 TB to 500 TB, and set the backup datacenter number ranging from 5 to 10. In the latter case, we set the total backup datacenter number as 8, set the total backup load amount ranging from 0.5 PB to 3 PB, adjust the amount of backup data in every application datacenter and the maximum load capacity in every backup datacenter with increase of total backup load amount.

## B. SIMULATION RESULTS

### 1) COMPARISON OF ALGORITHM EFFECTIVENESS

To improve convergence speed and computation efficiency, we limit the capacity of external set and remove redundant non-dominated solution(s) if necessary. To evaluate algorithm effectiveness, we should focus on the uniformity of solution distribution to see whether all solutions are equally spaced from one another. Therefore, we choose Spacing (SP) [32] as evaluation criteria. SP refers to the variance of the distance from every solution to its closest neighbor. We compute SP as follows:

$$SP = \sqrt{\frac{1}{|Pt|-1}\sum_{i=1}^{|Pt|}(\overline{d}-d_i)^2} \tag{30}$$

$$d_i = \min_j\left\{\left|\frac{f_1^i - f_1^j}{f_1^{\max} - f_1^{\min}}\right| + \left|\frac{f_2^i - f_2^j}{f_2^{\max} - f_2^{\min}}\right|\right\}, \quad i,j = 1,2,\ldots,|Pt|,\ i \neq j \tag{31}$$

Here we use $|Pt|$ to denote the obtained Pareto front, $||_c$ to denote the cardinality. We use $f_k^i$ and $f_k^j (k = 1, 2)$ to represent the value of the *kth* objective in the *ith* and *jth* solution, use $f_k^{\max}$ and $f_k^{\min} (k = 1, 2)$ to represent the maximum value and the minimum value of the *kth* objective, respectively. We use $\overline{d}$ to denote the average value of all $d_i$. Obviously, a good solution set should have SP value close to 0. For comparison, we implement the DEA-MO without capacity limitation of external set (denoted as DEA-MO-NL). For DADP and TwoStep-ILP, it's meaningless to compare SP value because neither of them jointly considers expected disaster loss and evacuation capability.

Table 1 shows SP performance comparison with increase of backup datacenter number. We can see that ADL-BB and MDL-BB performs better than DEA-MO-NL benefiting from their distance limitation constraints to obtain more evenly distributed evacuation latency value. Especially for MDL-BB, limiting the maximum distance results in more gentle fluctuation of evacuation capability. DEA-MO outperforms other algorithms with a more uniform solution distribution, because we keep eliminating the solution(s) with the largest FDR when the number of solutions exceeds $Cap_{ES}$.

**TABLE 1. Comparison of SP performance with increase of backup datacenter number.**

| Number of backup datacenter | Algorithm | SP |
|---|---|---|
| 5 | ADL-BB | 0.28 |
| | MDL-BB | 0.25 |
| | DEA-MO | 0.21 |
| | DEA-MO-NL | 0.43 |
| 6 | ADL-BB | 0.23 |
| | MDL-BB | 0.19 |
| | DEA-MO | 0.16 |
| | DEA-MO-NL | 0.35 |
| 7 | ADL-BB | 0.21 |
| | MDL-BB | 0.20 |
| | DEA-MO | 0.13 |
| | DEA-MO-NL | 0.29 |
| 8 | ADL-BB | 0.17 |
| | MDL-BB | 0.15 |
| | DEA-MO | 0.11 |
| | DEA-MO-NL | 0.27 |
| 9 | ADL-BB | 0.15 |
| | MDL-BB | 0.11 |
| | DEA-MO | 0.07 |
| | DEA-MO-NL | 0.26 |
| 10 | ADL-BB | 0.13 |
| | MDL-BB | 0.08 |
| | DEA-MO | 0.05 |
| | DEA-MO-NL | 0.23 |

Table 2 shows SP performance comparison with increase of backup load amount. The comparison results are similar to those in Table 1, showing good optimization effects for two objectives simultaneously in DEA-MO.

### 2) COMPARISON OF DATA INTEGRITY

It is found in experiments that the value of total damage risk faced by backup datacenters in some algorithms is relatively small (even be less than 0.1 in some cases). Similar to previous works [18]–[20], we aim to display and compare data availability and then introduce data integrity factor (*DIF*) as follows:

$$DIF = 1 - \frac{\sum\limits_{bd_i \in BD}\left(\sum\limits_{dc_j \in \theta(bd_i)} bl(dc_j) \cdot \sum\limits_{v \in V}\left(x_v^{bd_i} \cdot P_v\right)\right)}{\sum\limits_{bd_i \in BD}\sum\limits_{dc_j \in \theta(bd_i)} bl(dc_j)} \tag{32}$$

**TABLE 2.** Comparison of SP performance with increase of backup load amount.

| Backup load amount (PB) | Algorithm | SP |
|---|---|---|
| 0.5 | ADL-BB | 0.13 |
| | MDL-BB | 0.11 |
| | DEA-MO | 0.07 |
| | DEA-MO-NL | 0.22 |
| *1* | ADL-BB | 0.15 |
| | MDL-BB | 0.12 |
| | DEA-MO | 0.09 |
| | DEA-MO-NL | 0.25 |
| *1.5* | ADL-BB | 0.17 |
| | MDL-BB | 0.15 |
| | DEA-MO | 0.10 |
| | DEA-MO-NL | 0.27 |
| *2* | ADL-BB | 0.21 |
| | MDL-BB | 0.17 |
| | DEA-MO | 0.12 |
| | DEA-MO-NL | 0.32 |
| 2.5 | ADL-BB | 0.26 |
| | MDL-BB | 0.20 |
| | DEA-MO | 0.16 |
| | DEA-MO-NL | 0.41 |
| *3* | ADL-BB | 0.33 |
| | MDL-BB | 0.25 |
| | DEA-MO | 0.19 |
| | DEA-MO-NL | 0.51 |

Here we use *DIF* as the data integrity factor of backup datacenters. Since backup datacenters of different data amounts might be faced with different disaster risks, we use weighted average value to measure data integrity with a global view.

In Fig. 4 and Fig. 5, we illustrate the comparison of *DIF* with increase of backup datacenter number and backup load amount, respectively. Different from other algorithms, TwoStep-ILP leverages one-to-one mutual backup model for all datacenter nodes, and therefore its backup datacenter number is fixed as 24 in the US-Backbone topology. In Fig. 4, we set the total backup load amount as 1.5 PB, implement TwoStep-ILP separately for 10 times and finally obtain its average value of *DIF* as 0.72 which means much lower data integrity than that of other algorithms. In Fig. 5, the *DIF* of TwoStep-ILP ranges from about 0.70 to 0.73 with increase of backup load amount. DADP focuses on reducing expected disaster loss and keeps the *DIF* value steadily above that of other algorithms. But the ignorance of evacuation capability would lead to its poor performance in terms of evacuation latency in case of disaster. In Fig. 4 and Fig. 5, DEA-MO slightly outperforms ADL-BB and MDL-BB, benefiting from its disaster-aware unique pheromone and heuristic information for the location selection of every $bd_i$.

### 3) COMPARISON OF AVERAGE EVACUATION LATENCY

As mentioned in Section III, we use average evacuation latency $laten_{avg}$ to evaluate evacuation capability of backup datacenters in case of disaster. To jointly reduce the expected disaster loss and improve evacuation capability, we would like to control *DIF* as large as possible and $laten_{avg}$ as small as possible. For the convenience of comparison, we compute the evacuation latency from every backup datacenter to its application datacenters in a unified way by the transmission strategy proposed in our earlier work [6].

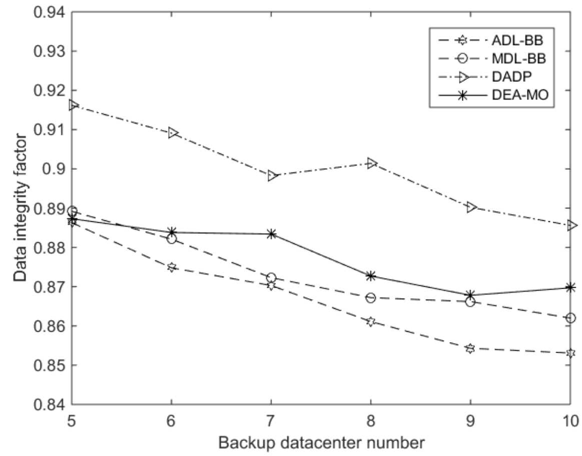



**FIGURE 4.** Comparison of ***DIF*** with increase of backup datacenter number.

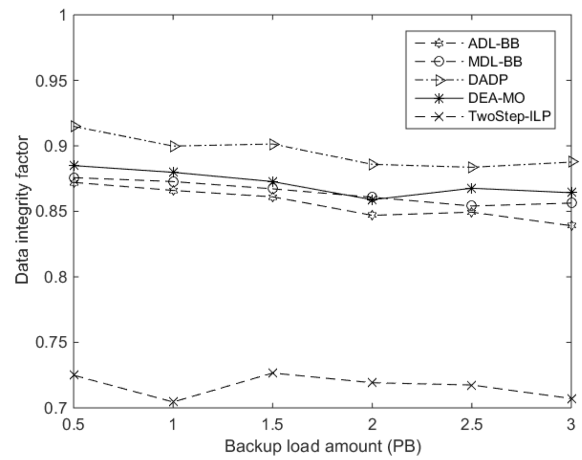



**FIGURE 5.** Comparison of ***DIF*** with increase of backup load amount.

In Fig. 6 and Fig. 7, we illustrate the comparison of $laten_{avg}$ with increase of backup datacenter number and backup load amount, respectively. DADP obtains larger $laten_{avg}$ due to its ignorance of evacuation capability during backup datacenter location selection process. ADL-BB and MDL-BB outperform DADP because of selecting nearer available locations from every application datacenter to place related backup datacenter. With minimum total hop number, TwoStep-ILP performs even better than ADL-BB and MDL-BB. But the smaller total hop number does not always mean better network transmission capability. DEA-MO considers not only hop number but also location-output-capability for the placement of every $bd_i$, and therefore obtains the best optimization effect in terms of evacuation latency.

Furthermore, we compute network utilization with increase of backup load amount to compare the network transmission capability utilization in evacuation process among these algorithms. There is no network utilization comparison with increase of backup datacenter number, because TwoStep-ILP leverages one-to-one mutual backup model for

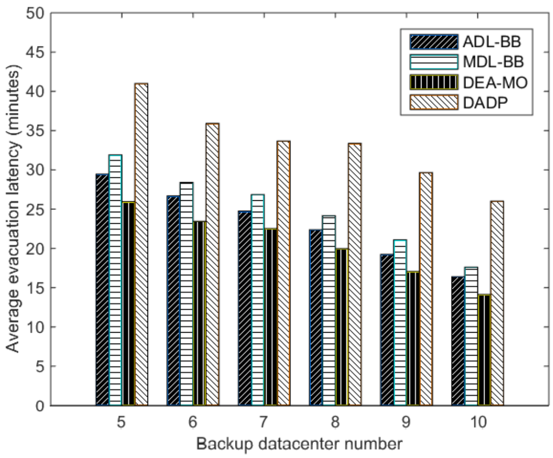



**FIGURE 6.** Comparison of $laten_{avg}$ with increase of backup datacenter number.

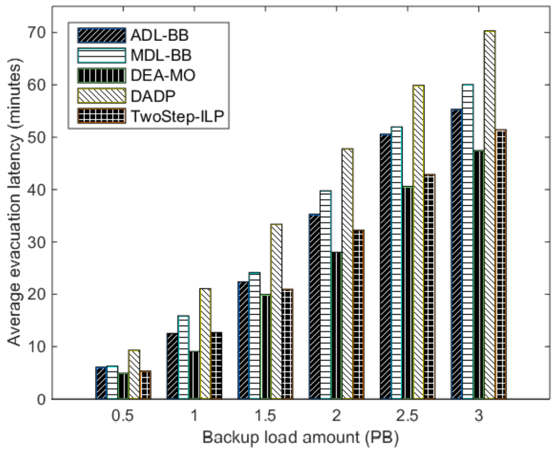



**FIGURE 7.** Comparison of $laten_{avg}$ with increase of backup load amount.

all datacenter nodes. Since we evacuate data with the same proportional bandwidth allocation strategy for concurrent transfers [6], the network utilization directly reflects the usage of network transmission capability. Higher network utilization means more full use of network transmission capacity, so as to achieve more efficient post-disaster evacuation.

We first compute maximum flow from application datacenters to their backup datacenters in every algorithm, and denote the largest one as *MaxFlow*. Then, we run ADL-BB, MDL-BB, DEA-MO, DADP, and TwoStep-ILP respectively to get their throughput as $Throughput_{ADL}$, $Throughput_{MDL}$, $Throughput_{DEA}$, $Throughput_{DADP}$ and $Throughput_{ILP}$. We compute the NT [33] for these five algorithms as follows:

$$NT_{ADL} = Throughput_{ADL} / MaxFlow \tag{33}$$
$$NT_{MDL} = Throughput_{MDL} / MaxFlow \tag{34}$$
$$NT_{DEA} = Throughput_{DEA} / MaxFlow \tag{35}$$
$$NT_{DADP} = Throughput_{DADP} / MaxFlow \tag{36}$$
$$NT_{ILP} = Throughput_{ILP} / MaxFlow \tag{37}$$

In Fig. 8, we represents the comparison of NT among ADL-BB, MDL-BB, DEA-MO, DADP, and TwoStep-ILP with increase of backup load amount. ADL-BB (with NT ranging from about 86% to 74%)and MDL-BB (with NT ranging from about 83% to 72%) outperform DADP (with NT ranging from about 71% to 56%) because their primary-to-backup distance constraints reduce total hops among every backup datacenter and their application datacenters. TwoStep-ILP performs even better because it minimizes total hop number and its one-to-one mutual backup model obtains more full use of network transmission capacity with higher NT ranging from about 87% to 79%. But higher network utilization does not necessarily mean more efficient transmission activity as shown in our earlier works [6], [9], [12]. DEA-MO achieves the highest NT ranging from about 93% to 86%, benefiting from its joint consideration of hop number and post-disaster evacuation capability from the viewpoint of traffic engineering (e.g., location-output-capability and backup-evacuation-latency, etc.).

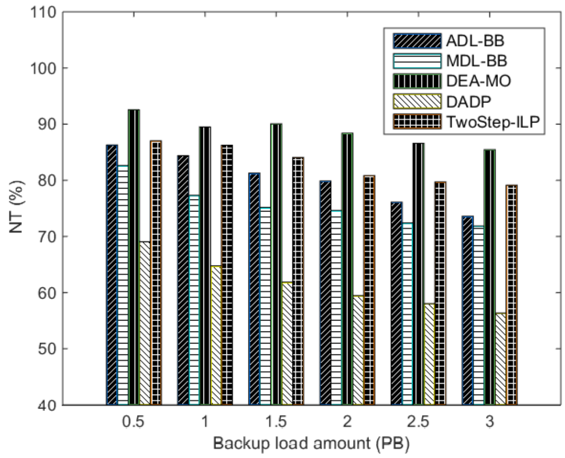



**FIGURE 8.** Comparison of NT with increase of backup load amount.

### 4) COMPARISON OF COMPUTATION TIME

We illustrate the comparison of computation time with increase of backup datacenter number and total datacenter number. In Fig. 9, we leverage the US-Backbone topology with 24 datacenter nodes and set the backup datacenter number ranging from 5 to 10. In Fig. 10, we leverage the Waxman model [34] to generate topologies with datacenter node number ranging from 20 to 100. We set the number of backup datacenters as 30% of the total number of nodes (except for TwoStep-ILP which leverages mutual backup strategy). We can see rising curves in Fig. 9 and Fig. 10. The computation time of ADL-BB grows more rapidly than other algorithms. Although its performance in reducing average evacuation latency is relatively good, it takes more time to obtain solutions. Similarly, DADP has good performance in reducing expected disaster loss, but its computation time grows very fast. TwoStep-ILP leverages the ellipsoid or interior point algorithm to determine backup datacenter location with fast speed, but its data integrity is relatively low without considering disaster risk distribution. Compared with MDL-BB, DEA-MO obtains faster convergence speed. Because it tends

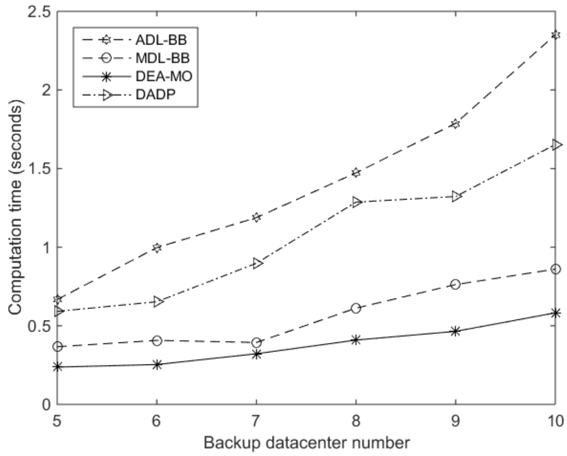



**FIGURE 9.** Comparison of computation time with increasing backup datacenter number.

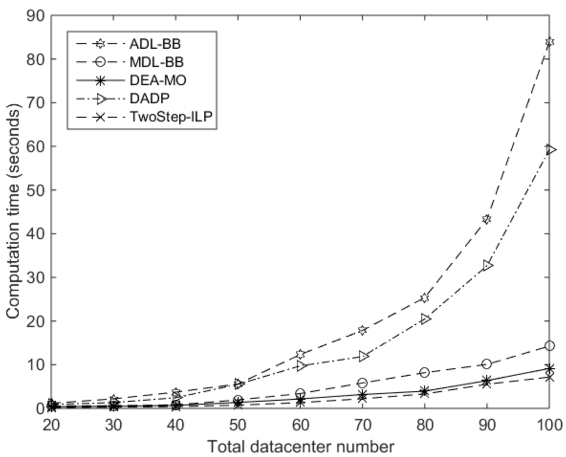



**FIGURE 10.** Comparison of computation time with increasing total datacenter number.

to make better choice through continuous pheromone updating after every iteration and improves computation efficiency by pruning strategy to remove redundant non-dominated solution(s).

In all, the simulation results above indicate that DEA-MO is an efficient and promising algorithm, since among the five algorithms for comparison, it reduces expected disaster loss and improves evacuation capability simultaneously with a relatively short computation time.

## VI. CONCLUSION

Backup datacenters holding massive high-value data are faced to increasing disaster risks. To reduce damage loss in case of disasters, reasonable backup datacenter placement requires less expected disaster loss and higher evacuation capability. Hence, with global view of network resources in the SDN scenarios, we propose a new disaster-and-evacuation-aware backup datacenter placement strategy and then design multi-objective optimization algorithm to realize it. As far as we know, this is the first research that optimizes post-disaster evacuation capability from the viewpoint of traffic engineering in backup datacenter placement phase. The innovation points mainly embody in dynamically optimizing multiple objectives in different disaster situations, and unique pheromone and heuristic information to adjust location searching. Especially, we use location-output-capability and backup-evacuation-latency to jointly evaluate evacuation capability of candidate node, leverage Pareto-recommendation-degree to express the influence of non-dominated solutions on location selection, and define node-damage-loss to estimate expected disaster loss.

Through extensive simulations, we demonstrate that our algorithm outperforms state-of-the-art algorithms with less expected disaster loss and higher post-disaster evacuation capability. Based on this placement strategy, we will aim to optimizing the subsequent backup transmission phase for proportional bandwidth allocation to backup datacenters and load balance on critical links.

## ACKNOWLEDGMENT

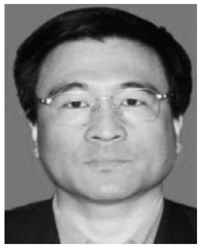
The authors would like to thank the editors and the reviewers for their valuable comments and suggestions.

## REFERENCES

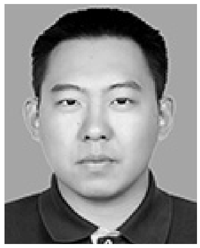
[1] K. Chen *et al.*, "DAC: Generic and automatic address configuration for data center networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 84–99, Feb. 2012.
[2] M. F. Bari *et al.*, "Data center network virtualization: A survey," *IEEE Commun. Surv. Tuts.*, vol. 15, no. 2, pp. 909–928, 2nd Quart., 2013.
[3] 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.
[4] Y. Ran, "Considerations and suggestions on improvement of communication network disaster countermeasures after the Wenchuan earthquake," *IEEE Commun. Mag.*, vol. 49, no. 1, pp. 44–47, Jan. 2011.
[5] T. Adachi, Y. Ishiyama, Y. Asakura, and K. Nakamura, "The restoration of telecom power damages by the Great East Japan Earthquake," in *Proc. INTELEC*, Amsterdam, The Netherlands, Oct. 2011, pp. 1–5.
[6] 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.
[7] 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.
[8] F. Dikbiyik, M. Tornatore, and B. Mukherjee, "Minimizing the risk from disaster failures in optical backbone networks," *J. Lightw. Technol.*, vol. 32, no. 18, pp. 3175–3183, Sep. 15, 2014.
[9] 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.
[10] 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.
[11] S. Agarwal, M. Kodialam, and T. V. Lakshman, "Traffic engineering in software defined networks," in *Proc. IEEE INFOCOM*, Turin, Italy, Apr. 2013, pp. 2211–2219.
[12] X. Li, H. Wang, S. Yi, X. Yao, F. Zhu, and L. Zhai, "Redundancy-guaranteed and receiving-constrained disaster backup in cloud data center network," *IEEE Access*, vol. 6, pp. 47666–47681, 2018.
[13] A. Bianco, L. Giraudo, and D. Hay, "Optimal resource allocation for disaster recovery," in *Proc. GLOBECOM*, Miami, FL, USA, Dec. 2010, pp. 1–5.
[14] 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.
[15] 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.

[16] 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.
[17] 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.
[18] T. Nakamura, S. Matsumoto, and H. Muraoka, "Discreet method to match safe site-pairs in short computation time for risk-aware data replication," *IEICE Trans. Inf. Syst.*, vol. E98-D, no. 8, pp. 1493–1502, 2015.
[19] T. Nakamura, S. Matsumoto, M. Tezuka, S. Izumi, and H. Muraoka, "Comparison of distance limiting methods for risk-aware data replication in urban and suburban area," *J. Inf. Process.*, vol. 24, no. 2, pp. 381–389, 2016.
[20] S. Matsumoto, T. Nakamura, and H. Muraoka, "Redundancy-based iterative method to select multiple safe replication sites for risk-aware data replication," *IEEJ Trans. Electr. Electron. Eng.*, vol. 11, no. 1, pp. 96–102, 2016.
[21] 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, 2015.
[22] Y. Gao, H. Guan, Z. Qi, Y. Hou, and L. Liu, "A multi-objective ant colony system algorithm for virtual machine placement in cloud computing," *J. Comput. Syst. Sci.*, vol. 79, no. 8, pp. 1230–1242, 2013.
[23] M. H. Ferdaus *et al.*, "Multi-objective, decentralized dynamic virtual machine consolidation using ACO metaheuristic in computing clouds," in *Concurrency and Computation: Practice and Experience*. Hoboken, NJ, USA: Wiley, 2016, pp. 1–40.
[24] A. Ashraf and I. Porres, "Multi-objective dynamic virtual machine consolidation in the cloud using ant colony system," *Int. J. Parallel, Emergent Distrib. Syst.*, vol. 33, no. 1, pp. 103–120, 2018.
[25] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit, "Local search heuristics for k-median and facility location problems," *SIAM J. Comput.*, vol. 33, no. 3, pp. 544–562, 2004.
[26] C. Gao, H. Wang, F. Zhu, L. Zhai, and S. Yi, "A particle swarm optimization algorithm for controller placement problem in software defined network," in *Proc. ICA3PP*, Zhangjiajie, China, 2015, pp. 44–54.
[27] B. Mukherjee, M. Habib, and F. Dikbiyik, "Network adaptability from disaster disruptions and cascading failures," *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 230–238, May 2014.
[28] M. Dorigo and G. Di Caro, "Ant colony optimization: A new meta-heuristic," in *Proc. CEC99*, Washington, DC, USA, 1999, pp. 1470–1477.
[29] M. Batayneh, D. A. Schupke, M. Hoffmann, A. Kirstaedter, and B. Mukherjee, "On routing and transmission-range determination of multi-bit-rate signals over mixed-line-rate wdm optical networks for carrier ethernet," *IEEE/ACM Trans. Netw.*, vol. 19, no. 5, pp. 1304–1316, Oct. 2011.
[30] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Apr. 1997.
[31] F. Dikbiyik *et al.*, "Minimizing the disaster risk in optical telecom networks," in *Proc. Opt. Fiber Commun. Conf.*, Los Angeles, CA, USA, Mar. 2012, Paper OTh4B-2.
[32] J. Schott, "Fault tolerant design using single and multicriteria genetic algorithm optimization," M.S. thesis, Dept. Aeronaut. Astronaut., Massachusetts Inst. Technol., Cambridge, MA, USA, 1995.
[33] 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.
[34] M. Naldi, "Connectivity of Waxman topology models," *Comput. Commun.*, vol. 29, no. 1, pp. 24–31, 2005.

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



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



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

**SHUAI LIU** received the master's degree from Shandong University, China, in 2016. He is currently pursuing the Ph.D. degree with Beihang University. His current research interest includes wireless networks.

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

**CHUANQI JIANG** received the bachelor's degree from Information Engineering University, China, in 2010. He is currently pursuing the master's degree with Shandong University. His current research interests include network optimization, network measurement, deep learning, and network energy-saving.