Evacuate before Too Late: Distributed Backup in Inter-DC Networks with Progressive Disasters

Xiaokang Xie, Qing Ling, *Senior Member, IEEE*, Ping Lu, Wei Xu, and Zuqing Zhu, *Senior Member, IEEE*

Abstract—Inter-datacenter (inter-DC) networks are essential for large enterprises to deliver high-quality services to end-users. Since DCs are vulnerable to natural disasters, an inter-DC network operator needs an effective emergency backup plan to evacuate the endangered data out in case of a progressive disaster whose status can be predicted by an early warning system. In this paper, we try to solve the problem of emergency backup in inter-DC networks with progressive disasters. We first utilize the time-expanded network (TEN) approach to model the time-variant inter-DC network during a progressive disaster as a variant TEN (VTEN) and convert the dynamic flow scheduling for emergency backup to a static one. Then, with the VTEN, we formulate an optimization model to maximize the profit from the emergency backup in consideration of data values and resource costs. Although this large-scale optimization can be solved in a distributed way by leveraging the alternation direction method of multipliers (ADMM), we find that one of its subproblems is nontrivial in the distributed setting. We propose a novel inexact ADMM approach to resolve the issue induced by the subproblem, and prove that the proposed algorithm can converge to the optimal solution. The results from extensive simulations confirm that our algorithm is robust and time-efficient, and outperforms several benchmarks in terms of backup profit and running time.

Index Terms—Inter-DC networks, Emergency backup, Progressive disasters, Alternating direction method of multipliers (ADMM), Time-expanded network (TEN).

1 Introduction

NOWADAYS, datacenters (DCs) have become the key IT infrastructure to support cloud computing, Big Data analytics and other data-intensive emerging applications [1]. In order to provide low-latency, high-quality and non-disruptive services to end-users, large enterprises such as Google, Facebook and Amazon have placed their DCs in a geographically distributed manner and built inter-DC networks to interconnect them. As a DC carries enormous amounts of data and runs virtual machines (VMs) to deliver services to thousands or even millions of end-users, a breakdown on it would cause unimaginable losses. Recent statistics indicated that a DC operator could lose over $9000 per minute because of unexpected DC outages [2].

However, DCs are vulnerable to natural disasters such as flood, earthquake, hurricane and tsunami, which can easily wipe them out [3]. Therefore, it is essential to have an effective data evacuation plan that can back up as much data from the endangered DCs as possible before an upcoming disaster destroys them. Note that, certain disasters, *e.g.*, hurricane and tsunami, are progressive and thus usually predictable. For instance, it took Hurricane Sandy several days since its formation to move ashore on the east coast of the United States [4]. Hence, an early warning system can provide useful information regarding such a disaster (*i.e.*, the time and corresponding impact range) to an inter-DC network operator and let it evacuate important data out before too late. Specifically, the inter-DC network operator needs to figure out an effective emergency backup plan, *i.e.*, how to back up endangered data under a rigid time constraint over a time-variant network topology [5], [6].

Note that, there are generally two types of backups in inter-DC networks, *i.e.*, regular backup and emergency backup. Regular backup runs periodically when the network is in its working state to move data around among geographically distributed DCs, for obtaining sufficient data redundancy [7]. Therefore, previous studies have treated regular backup as a problem of bulk-data transfer [8]–[10], and considered how to schedule data transfers and optimize resource allocation for minimizing bandwidth cost [11] or how to achieve coordinated data transfers to minimize the backup duration [7]. However, both the backup scheme and network model of regular backup are fundamentally different from those of emergency backup, which is triggered in response to a predictable and progressive disaster [5].

Previously, we have studied emergency backup in [5]. Specifically, we defined a utility function to quantify data value, leveraged the time-expanded network (TEN) approach [8] for data transfer scheduling, and designed a distributed algorithm based on dual decomposition to maximize the data owners' utility. However, the work in [5] still bears three drawbacks. Firstly, the proposed algorithm only considers the revenue gain from successful data backups but ignores the costs of the network resources (*i.e.*, bandwidth on links and storage in DCs) used in the backup process, which makes the revenue model less practical. Secondly, the dual decomposition suffers from slow convergence speed and/or oscillation around the optimal solution. Last and most importantly, the algorithm also calculates the backup scheme with a fixed topology but does not consider the topology change during a progressive disaster. To address these issues, we investigated the emergency backup in an

- *X. Xie, P. Lu and Z. Zhu are with the School of Information Science and Technology, University of Science and Technology of China, 230027. E-mail: zqzhu@ieee.org.*
- *Q. Ling and W. Xu are with the Department of Automation, University of Science and Technology of China, Hefei, Anhui 230027, China. E-mail: qingling@mail.ustc.edu.cn.*

Manuscript received on May 12, 2017.

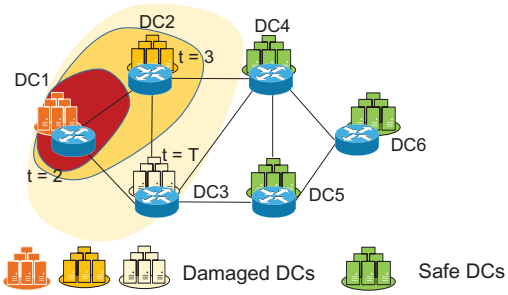


Fig. 1. Example on an inter-DC network with a progressive disaster.

inter-DC network whose topology is time-variant in [12] and used the alternating direction method of multipliers (ADMM) [13] to design a distributed algorithm. Nevertheless, the study in [12] is still preliminary, since it did not fully optimize the distributed algorithm's performance and the algorithm's convergence was not verified theoretically.

In this paper, we extend our work in [12] to solve the problem of emergency backup in inter-DC networks with progressive disasters. First of all, to reduce the complexity of dynamic flow scheduling, we utilize the TEN approach to model the time-variant inter-DC network with a progressive disaster as a variant TEN (VTEN) and convert the dynamic flow scheduling problem to a static one. With the VTEN, we formulate an optimization problem to maximize the profit from the emergency backup in consideration of data values and resource costs. We then apply ADMM to solve this large-scale optimization problem in a distributed manner. As one subproblem in ADMM is subject to network constraints and nontrivial in the distributed setting, we solve it approximately with a single primal-dual gradient step and also improve the time efficiency, which results in a novel inexact ADMM approach. We theoretically prove that the inexact ADMM based algorithm can converge to the optimal solution, and evaluate it with extensive simulations. Simulation results demonstrate that our proposal is robust and time-efficient, and outperforms several benchmarks in terms of backup profit and running time. In summary, the major contributions of this work are as follows.

- **Model**. We address the emergency backup in an inter-DC network with a progressive disaster, which is challenging since it needs to accomplish dynamic flow scheduling in a time-variant network. We adopt the VTEN technique to reformulate it to a tractable profit optimization problem over a static network.
- **Algorithm**. We propose an inexact ADMM-based algorithm to solve this problem in a distributed manner. Different from the conventional ADMM that exactly solves involved subproblems, we solve the subproblem with network constraints approximately, and thus significantly reduce the time complexity.
- **Theory**. We prove that the proposed inexact ADMM-based distributed algorithm converges to the optimal solution. This theoretical guarantee is corroborated by extensive numerical experiments.

The rest of this paper is organized as follows. Section 2 provides a brief survey on the related work. In Section 3, we formulate the optimization problem to address the emergency backup in an inter-DC network with a progressive disaster. The inexact ADMM-based distributed algorithm is proposed in Section 4, and Section 5 theoretically proves that it can converge to the optimal solution. In Section 6, we present the performance evaluation with numerical simulations. Finally, Section 7 summarizes the paper.

2 Related Work

In an inter-DC network, data backup is the most frequently used technique to maintain the survivability and integrity of data. Previous studies have addressed both regular backup and emergency backup scenarios. For regular backup [7], people usually treated it as transferring normal bulk-data in an inter-DC network [14]. In [1], [3], [7], we have considered the regular data backup schemes in inter-DC optical networks, and tried to minimize the DC backup window (*i.e.*, the overall duration of bulk-data transfers) to avoid the prolonged negative impacts on normal DC operation. Meanwhile, without particularly aiming at solving the regular backup problem, the studies in [15], [16] have also addressed how to schedule bulk-data transfers in an inter-DC network with dynamic traffic. However, compared with the emergency backup considered in this work, regular backup has two fundamental differences, *i.e.*, the data transfers do not have a rigid deadline and the inter-DC network's topology is usually not time-variant. Therefore, the algorithms proposed for regular backup can hardly be leveraged to design emergency backup schemes.

Emergency backup is triggered in response to a predictable and progressive disaster [5]. Since emergency backup has to evacuate as much endangered data out as possible under a rigid time constraint over a time-variant network topology, it involves more sophisticated data transfer scheduling. Ma *et al.* [6] proposed several algorithms to back up endangered data within a pre-determined warning time and minimized the backup costs. Nevertheless, they did not consider a progressive disaster that would generate a time-variant network topology, and treated all the endangered data equally in the backup without service differentiation. Hence, the algorithms would not give priority to the critical data when not all the data can be evacuated. The study in [17] differentiated data based on its importance and tried to back up critical data first within the warning time. However, the backup scheme was developed over a fixed topology, which did not reflect the case in progressive disasters. The authors of [18] investigated the data evacuation strategy for a wireless sensor network (WSN) that just experienced a disaster. Since the strategy was designed for the post-disaster scenario in WSNs, the network model is fundamentally different from ours. Moreover, the work in [18] only considered a fixed topology when designing the data evacuation strategy. We have also studied emergency backup in inter-DC networks in [5], [12]; but as explained in the previous section, our previous studies still bear certain drawbacks.

The problem of emergency backup in inter-DC networks can essentially be viewed as scheduling and routing of multi-source multi-destination flows over time. However, it is known that this type of dynamic flow scheduling problems are generally very complicated [19]. Specifically, the common problems of *fraction multi-commodity flows over time*

and *minimum-cost flow over time* are both NP-hard [19]. To assist the problem solving of dynamic flow scheduling, Ford *et al.* [20] proposed the TEN method, which adopts a discrete time model to expand the network by replicating its topology for each time interval and converts the dynamic network to a static one. Hence, TEN helps to simplify dynamic flow scheduling at the cost of increased network size. This has motivated several researches on how to reduce the network size obtained by TEN [15], [21]. However, their proposals usually involve relatively complicated procedure and thus cannot address the complexity issue of TEN effectively.

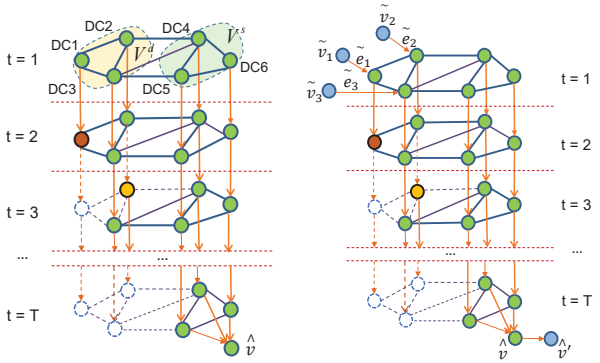
On the other hand, a large-scale optimization problem can also be solved time-efficiently with a distributed algorithm. For example, ADMM [13] has been considered as a powerful tool for this purpose since it has a naturally parallel implementation. Note that, in the standard approach of ADMM, variables are divided into sub-blocks, each of which corresponds to a subproblem that is solved exactly in an iteration. The convergence of standard ADMM has already been theoretically verified in [22]. Nevertheless, in many practical cases, solving the subproblems in ADMM exactly is costly, which promotes the studies on inexact ADMM.

It is known that an inexact ADMM adopts inexact subproblem solutions whose inaccuracy can be maintained within a given tolerance in each iteration [23]. Previously, researchers have designed a few inexact ADMM schemes that try to solve subproblems inexactly with various approximation methods [24]–[27]. However, these studies only addressed the subproblems that have no or very simple constraints (*i.e.*, the trivial ones). As we will explain in Section 4.3, we encounter a subproblem with network constraints, which is nontrivial in a distributed setting. To address this issue, we introduce an inexact primal-dual gradient step to find an approximate solution to the subproblem. This actually leads to a new type of inexact ADMM, which, to the best of our knowledge, has not been explored before.

3 Problem Formulation

3.1 Network Model

Fig. 1 illustrates an inter-DC network that is being impacted by a progressive disaster. In the network, the DCs are interconnected by high-capacity links, and the disaster can bring down both the DCs and links in a time-variant manner. The disaster starts to land on *DC* 1 at $t = 2$, and as time goes on, *DCs* 2 and 3 will be destroyed at $t = 3$ and T, respectively. After $t = T$, the disaster's impact range stops to increase and thus the remaining DCs stay unaffected. Apparently, the DCs in the inter-DC network can be classified into two categories, *i.e.*, damaged DCs and safe ones. When the disaster progresses, the network operator should try to evacuate the endangered data to safe DCs according to its importance at each time interval, before it will be wiped out by the disaster. Note that, with the development of the disaster, certain safe DCs can become damaged ones and hence they should be only used to buffer the endangered data for the time being. Hence, for each damaged DC, there is an *emergency backup window*, which refers to the duration from when an early warning is received to when it is impacted by the disaster. Note that, we only consider the predictable disasters whose ranges and timing can be forecasted precisely [28], [29], and the unpredictable disasters discussed in [28] are out of scope of this work. When the range and timing of a disaster cannot be predicted, it can be modeled with a probabilistic model [28], [30], which will be considered in our future work.



(a) VTEN (b) Modified VTEN for ADMM

Fig. 2. Examples on building VTENs.

We denote the original topology of the inter-DC network as a directed graph $G_0(V_0, E_0)$, where V_0 and E_0 are the sets of DC nodes and links before a disaster happens, respectively. When the disaster stops (*i.e.*, $t > T$), a group of DCs are damaged and the rest are safe. We denote the subset of damaged DCs as V^d and the rest as $V^s = \{i : i \in V_0 \backslash V^d\}$. Similar to previous studies in [31]–[34], we assume that the inter-DC network operates at discrete times $t = 1, 2, \cdots$. During the progressive disaster, each damaged DC $i \in V^d$ has an emergency backup window T_i [29], within which it should evacuate as much endangered data in it as possible to the safe DCs. Hence, the overall backup window of the network will be $T = \max_{i \in V^d}(T_i)$. We assume that all the emergency backup windows are known through scientific forecast [29] before the backups are triggered.

As shown in Fig. 1, the inter-DC network is time-variant due to the progressive disaster. Therefore, we leverage the TEN approach [20] to characterize its operation over time as a variant TEN (VTEN) $\mathcal{G}(\mathcal{V}, \mathcal{E})$. Here, the VTEN is a topology with T layers, each of which is denoted as $G_\theta(V_\theta, E_\theta), \theta \in \{1, \cdots, T\}$ and corresponds to the inter-DC network's topology at time $t = \theta$. The first layer $G_1(V_1, E_1)$ simply copies the original topology. With particular note, in V_1 we also denote the subset of DCs that shall be damaged at the end of the disaster as V^d and the rest as V^s, as we have done for V_0. In each layer of the VTEN, the bandwidth of a link $e_\theta \in E_\theta$ is represented as B_{e_θ}, which is the available bandwidth on link $e \in E_0$ at time $t = \theta$. Meanwhile, in between two adjacent layers G_θ and $G_{\theta+1}$, we insert a directed link e_{θ_i} from each DC $i \in V_\theta$ to its replica $i \in V_{\theta+1}$ to represent the available storage on the DC at time $t = \theta$. Note that, to maximize the total throughput of the emergency backup, we allow the DCs to use the store-and-forward scheme together with direct data transfer. Specifically, an intermediate DC along the backup path can buffer the incoming data for future transmission opportunities, if its outbound bandwidth is not enough for the time being. Hence, the bandwidth of e_{θ_i} (*i.e.*, $B_{e_{\theta_i}}$) is set as the available storage on DC $i \in V_0$ at time $t = \theta$.

Now, the VTEN $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is constructed, and since the

progressive disaster can wipe DCs out and convert certain safe DCs to damaged ones over time, the topology of each layer can be different. This is the reason why it is referred to as VTEN. We can verify that any dynamic data transfer in the inter-DC network over time can be represented by a unique static flow in the VTEN $\mathcal{G}$, no matter whether store-and-forward is used or not [15], [20], [21]. Finally, to further simplify the flow routing of the emergency backup, we add a virtual super DC $\hat{v}$ in $\mathcal{G}$, which terminates a directed virtual link from each safe DC in the last layer G_T, whose bandwidth is set as the available storage on its source DC.

Fig. 2(a) shows an example on how to build the VTEN based on the status of the inter-DC network illustrated in Fig. 1. At $t = 1$, all the DCs are intact and thus the first layer of the VTEN is $V_1 = V_0$ and $E_1 = E_0$. Then, *DC* 1 is wiped out by the disaster at $t = 2$. Hence, from the second layer V_2, we remove *DC* 1 and all the links that connect to it. The directed links to represent the available storage on the DCs at $t = 1$ are also added in between V_1 and V_2. The subsequent layers can be built in the same way. Finally, after $t = T$, the disaster's impact range stops to increase. V^d denotes the set of DCs that eventually will be damaged by the disaster (*i.e.*, *DCs* 1-3 in Fig. 1), while V^s denotes the set of safe ones that will survive the disaster (*i.e.*, *DCs* 4-6 in Fig. 1). Each node in V^s originates a directed virtual link to the super DC $\hat{v}$ in the last layer G_T.

3.2 Optimization Model

As the emergency backup is time-constrained, we have to consider the situation in which not all the endangered data can be backed up successfully. Hence, we have to differentiate the endangered data according to its importance in the emergency backup. The data's importance usually can be understood as its value. Thus for each DC $i \in V^d$, we define a concave utility function $f_i(s_i)$ to quantify the value of the data in it, where s_i represents the data volume. If the data gets backed up successfully, the network operator gains the revenue determined by the utility function. In the meantime, the cost to the operator is due to the resource utilization by the emergency backup. Note that, the usages of both link bandwidth and DC storage can be modeled as link bandwidth usages in the VTEN $\mathcal{G}$. Therefore, we define a unit bandwidth cost c_e for each link $e \in \mathcal{E}$, and specifically, the bandwidth costs of the virtual links that connect to $\hat{v}$ are all 0. For simplicity, we define $b_{i,e}$ as the bandwidth allocated on a link $e \in \mathcal{E}$ for transferring the data from DC $i \in V^d$. Then, the profit from the emergency backup is the total utility minus total cost, which should be maximized to achieve a cost-effective backup.

Objective:

$$Maximize \quad \sum_{i \in V^d} f_i(s_i) - \sum_{e \in \mathcal{E}} \sum_{i \in V^d} c_e \cdot b_{i,e}. \tag{1}$$

Constraints:

1) *Data volume constraint:*

$$0 \le s_i \le C_i, \quad \forall i \in V^d, \tag{2}$$

where s_i is the volume of the data that has been successfully backed up for *DC* $i \in V^d$ and C_i is the total volume of data in DC i. Eq. (2) ensures that the evacuated data cannot exceed the total data in each damaged DC.

2) *Link capacity constraint:*

$$\begin{aligned} &\sum_{i \in V^d} b_{i,e} \le B_e, \quad \forall e \in \mathcal{E}, \\ &b_{i,e} \ge 0, \quad \forall i \in V^d, e \in \mathcal{E}, \end{aligned} \tag{3}$$

where B_e is the available bandwidth on link $e \in \mathcal{E}$. Eq. (3) ensures that the total bandwidth usage on each link for the emergency backup cannot exceed its capacity.

3) *Flow conservation constraint:*

$$\sum_{e \in \mathcal{V}^+} b_{i,e} - \sum_{e \in \mathcal{V}^-} b_{i,e} = \begin{cases} s_i, & v = i, \\ -s_i, & v = \hat{v}, \\ 0, & \text{Otherwise}, \end{cases} \quad \forall i \in V^d, v \in \mathcal{V}. \tag{4}$$

where $\mathcal{V}^+$ and $\mathcal{V}^-$ represent the set of directed links that are from and to node v, respectively. Eq. (4) is the flow conservation constraint to ensure that all the data evacuated from DC i reaches the virtual super DC $\hat{v}$ (*i.e.*, the safe DCs).

By solving the aforementioned optimization problem, which is essentially to find a profit-maximized multi-commodity flow in the VTEN $\mathcal{G}(\mathcal{V}, \mathcal{E})$ [12], we can obtain an optimal solution of the emergency backup. However, the issue with this approach is the additional complexity due to the increased network size. The time complexity to solve the optimization problem can increase exponentially with the numbers of variables and constraints in it, which are $|V^d| \cdot (1 + |\mathcal{E}|)$ and $|V^d| \cdot (1 + |\mathcal{V}| + |\mathcal{E}|)$, respectively. Meanwhile, since the VTEN has T layers as shown in Fig. 2(a), the numbers of nodes and links in it, *i.e.*, $|\mathcal{V}|$ and $|\mathcal{E}|$, respectively, would in general increase with T sub-linearly. This is because the numbers of nodes and links in the inter-DC network would decrease with time due to the progressive disaster. In all, we can see that the time complexity to solve the optimization problem increases sharply with the overall backup window T [20]. To address this issue, we will design an ADMM-based algorithm to solve this large-scale optimization in a distributed manner in the next section.

4 Algorithm Development

In this section, we proposed an inexact ADMM based distributed algorithm to solve the optimization and design several benchmarks based on some existing methods. The following list defines the notations used in our algorithm.

- $G_0 = (V_0, E_0)$: the original topology of the inter-DC network.
- $\mathcal{G} = (\mathcal{V}, \mathcal{E})$: the topology of VTEN.
- $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$: the topology of modified VTEN.
- V^d: the set of damaged DCs in the disaster.
- V^s: the set of safe DCs in the disaster.
- T: backup window, namely, the early warning time of a disaster.
- $\hat{v}$: the virtual super DC in the VTEN.
- $f_i(s_i)$: the utility function of damaged DC i.
- s_i: the volume of the data that has been successfully backed up for damaged DC i.
- $b_{i,e}$: the bandwidth allocated on a link $e \in \mathcal{E}$ for transferring the data from DC $i \in V^d$.
- c_e: the bandwidth cost for link $e \in \mathcal{E}$.
- C_i: the total volume of data in DC i.
- B_e: the bandwidth of link $e \in \mathcal{E}$.
- $a_{v,e}$: the parameter that indicates the relation between a node $v \in \mathcal{V}'$ and a link $e \in \mathcal{E}'$.

4.1 Overview of ADMM

It is known that ADMM is suitable for distributed convex optimizations, especially for large-scale ones, because it combines the decomposability of dual ascent and the superior convergence properties of the method of multipliers [13]. More specifically, ADMM can solve a separable optimization problem in the following form

$$\begin{aligned} \min_{x\in S_x, z\in S_z} \quad & f(x)+g(z), \\ s.t. \quad & A\cdot x + B\cdot z = c, \end{aligned} \tag{5}$$

where $x \in \mathcal{R}^m$ and $z \in \mathcal{R}^n$ are optimization variables, $A \in \mathcal{R}^{p\times m}$, $B \in \mathcal{R}^{p\times n}$, and $c \in \mathcal{R}^p$ are given, $f(\cdot)$ and $g(\cdot)$ are (convex) functions, while $\mathcal{S}_x$ and $\mathcal{S}_z$ are (convex) sets. In order to solve Eq. (5), ADMM searches for the saddle point of the following augmented Lagrangian function

$$\begin{aligned} L_\rho(x,z,\varphi) =& f(x)+g(z)+\varphi^T\cdot[A\cdot x+B\cdot z-c] \\ &+\frac{\rho}{2}\cdot\|A\cdot x+B\cdot z-c\|_2^2, \end{aligned} \tag{6}$$

where $\varphi \in \mathcal{R}^p$ is the Lagrange multiplier, $\rho > 0$ is the penalty parameter, and $\frac{\rho}{2}\cdot|A\cdot x+B\cdot z-c\|_2^2$ is the penalty term. Given the augmented Lagrangian function in Eq. (6), ADMM runs the following steps at every iteration k.

$$\begin{aligned} x^{k+1} &:= \operatorname*{argmin}_{x\in\mathcal{S}_x}\left[L_\rho(x,z^k,\varphi^k)\right], \\ z^{k+1} &:= \operatorname*{argmin}_{z\in\mathcal{S}_z}\left[L_\rho(x^{k+1},z,\varphi^k)\right], \\ \varphi^{k+1} &:= \varphi^k+\rho\cdot(A\cdot x^{k+1}+B\cdot z^{k+1}-c). \end{aligned} \tag{7}$$

For convenience, below we call the steps in Eq. (7) as x-optimization, z-optimization, and dual variable update, respectively. As the variables x and z are updated independently in each iteration, the term *alternating direction* is used.

4.2 Adaption of ADMM Form

Unfortunately, the optimization problem in Section 3.2 cannot be directly solved by ADMM in a distributed manner, since the constraint Eq. (4) couples the variables $\{s_i\}$ and $\{b_{i,e}\}$ and has to be dualized in ADMM. It inevitably brings cross-terms in the form of $b_{i,e}\cdot b_{i,e'}$ to the augmented Lagrangian function, where e and e' are two edges (*cf.* Eq. (6)). The existence of these cross-terms requires joint optimization involving multiple edges, and thus prohibits distributed computation. To address this issue, we adopt the following techniques to reformulate the optimization problem to a tractable one.

1) **Modification of VTEN:** Note that, for the flow conservation constraints in Eq. (4), variables $\{s_i\}$ and $\{b_{i,e}\}$ are coupled only when DC v is the source (*i.e.*, DC i) or the destination (*i.e.*, the virtual super DC $\hat{v}$). Hence, we first add $|V^d|$ virtual nodes in the VTEN, each of which is $\tilde{v}_i, i \in V^d$ and connects to a damaged DC i in the first layer of the VTEN with a virtual directed link $\tilde{e}_i$. Then, we insert another virtual super DC $\hat{v}'$ to connect to $\hat{v}$ in the last layer of the VTEN. Finally, the VTEN is modified as $\mathcal{G}'(\mathcal{V}',\mathcal{E}')$, where all the nodes in the original VTEN $\mathcal{G}$ become intermediate nodes. Fig. 2(b) shows the VTEN for ADMM, which is modified from the one in Fig. 2(a). For convenience, we introduce the adjacent matrix $A \triangleq [a_{11},\cdots a_{1|\mathcal{E}'|};\cdots,a_{\mathcal{V}'1},\cdots a_{\mathcal{V}'|\mathcal{E}'|}] \in \mathcal{R}^{|\mathcal{V}'|\times|\mathcal{E}'|}$ of the VTEN, whose (v,e)-th entry $a_{v,e}$ indicates the relation between a node $v \in \mathcal{V}'$ and a link $e \in \mathcal{E}'$ and is set as

$$a_{v,e} = \begin{cases} 1, & e\in\mathcal{V}^+, \\ -1, & e\in\mathcal{V}^-, \\ 0, & \text{Otherwise}, \end{cases} \quad v\in\mathcal{V},\ e\in\mathcal{E},$$

and $a_{v,e}=0$ for $v \in \mathcal{V}'\setminus\mathcal{V}, e\in\mathcal{E}'\setminus\mathcal{E}$. Then the original constraint in Eq. (4) is transformed into

$$\sum_{e\in\mathcal{E}'} a_{v,e}\cdot b_{i,e}=0, \quad \forall i\in V^d,\ v\in\mathcal{V}'. \tag{8}$$

Note that although these virtual nodes become new source and destination nodes, their existence does not affect the variables for the original nodes and edges.
Next, we merge variables $\{s_i\}$ into variables $\{b_{i,e}\}$. We set the available bandwidth of a newly-added virtual link $\tilde{e}_i, i \in V^d$ as the total data volume on DC i (*i.e.*, $B_{\tilde{e}_i}=C_i$). Then, Eqs. (2) and (3) can be unified as

$$\begin{aligned} &\sum_{i\in V^d} b_{i,e}\le B_e, \quad \forall e\in\mathcal{E}', \\ &b_{i,e}\ge 0, \quad \forall i\in V^d,\ \forall e\in\mathcal{E}'. \end{aligned} \tag{9}$$

We should emphasize that Eq. (9) is different from Eq. (3) in the sense that it holds for all $e \in \mathcal{E}'$, but not just for those $e \in \mathcal{E}$. We then introduce a parameter h_e to indicate whether a link is newly-added or not, *i.e.*, if e is a newly-added link, we have $h_e = 1$, and $h_e = 0$ otherwise. The objective in Eq. (1) can be rewritten as

$$Minimize \sum_{i\in V^d}\sum_{e\in\mathcal{E}'}\left[c_e\cdot b_{i,e}-h_e\cdot f_i(b_{i,e})\right]. \tag{10}$$

2) **Introduction of auxiliary variables:** Now the optimization problem in Eqs. (8)-(10) only has one set of variables $\{b_{i,e}\}$. To divide and conquer the summands in the objective function as well as the two constraints, we introduce auxiliary variables $\{z_{i,e}\}$ to duplicate $\{b_{i,e}\}$:

$$z_{i,e}=b_{i,e},\ \forall i\in V^d, e\in\mathcal{E}'. \tag{11}$$

Finally, the optimization problem is reformulated as

$$\begin{aligned} Minimize\ & \sum_{i\in V^d}\sum_{e\in\mathcal{E}'}[c_e\cdot z_{i,e}-h_e\cdot f_i(b_{i,e})], \\ s.t.\ & \sum_{i\in V^d} z_{i,e}\le B_e, \quad \forall e\in\mathcal{E}', \\ & \sum_{e\in\mathcal{E}'} a_{v,e}\cdot b_{i,e}=0, \quad \forall i\in V^d,\ v\in\mathcal{V}', \\ & z_{i,e}\ge 0,\ b_{i,e}\ge 0, \quad \forall i\in V^d, \forall e\in\mathcal{E}' \\ & z_{i,e}=b_{i,e}, \quad \forall i\in V^d, e\in\mathcal{E}', \end{aligned} \tag{12}$$

which takes the standard form of ADMM in Eq. (5) and can be solved in a distributed manner as shown next.

4.3 Inexact ADMM-based Distributed Algorithm

Below we leverage the ADMM approach described in Section 4.1 and develop a distributed algorithm to solve Eq. (12). We first write the augmented Lagrangian function as

$$\begin{aligned} L=&\sum_{i\in V^d}\sum_{e\in\mathcal{E}'} c_e\cdot z_{i,e}-h_e\cdot f_i(b_{i,e})+\langle\mu_{i,e},z_{i,e}-b_{i,e}\rangle \\ &+\frac{\rho}{2}\cdot\|z_{i,e}-b_{i,e}\|_2^2, \end{aligned} \tag{13}$$

where $\{\mu_{i,e} \in \mathcal{R}\}$ are the dual variables and $\rho > 0$ is the penalty parameter. Note that, the first three constraints in Eq. (12) remain inexplicit. To make the formulation of Eq. (13) more compact, we use the vector expression and define the partial augmented Lagrangian function of each link e as

$$L_e(z_e, b_e, \mu_e) = c'_e \cdot z_e + f_e(b_e) + \langle \mu_e, z_e - b_e \rangle + \frac{\rho}{2}\|z_e - b_e\|_2^2, \quad (14)$$

where $z_e \triangleq \left[z_{1,e}; \cdots; z_{|V^d|,e}\right] \in \mathcal{R}^{|V^d|}$, $b_e \triangleq \left[b_{1,e}; \cdots; b_{|V^d|,e}\right] \in \mathcal{R}^{|V^d|}$, $c'_e \triangleq [c_e; \cdots; c_e]^{\mathrm{T}} \in \mathcal{R}^{|V^d|}$, $f_e(b_e) = -\sum_{i \in V^d} h_e \cdot f_i(b_{i,e})$, and $\mu_e \triangleq \left[\mu_{1,e}; \cdots; \mu_{|V^d|,e}\right] \in \mathcal{R}^{|V^d|}$ are the dual variables. Then, the augmented Lagrangian function in Eq. (13) can be rewritten as

$$L(z, b, \mu) = \sum_{e \in \mathcal{E}'} L_e = c \cdot z + f(b) + \langle \mu, z - b \rangle + \frac{\rho}{2} \cdot \|z - b\|_2^2, \quad (15)$$

where $z \triangleq \left[z_1; \cdots; z_{|\mathcal{E}'|}\right] \in \mathcal{R}^{(|V^d| \cdot |\mathcal{E}'|)}$, $b \triangleq \left[b_1; \cdots; b_{|\mathcal{E}'|}\right] \in \mathcal{R}^{(|V^d| \cdot |\mathcal{E}'|)}$, $c \triangleq \left[c'_1; \cdots; c'_{|\mathcal{E}'|}\right]^{\mathrm{T}} \in \mathcal{R}^{(|V^d| \cdot |\mathcal{E}'|)}$, $f(b) = \sum_{e \in \mathcal{E}'} f_e(b_e)$, and $\mu \triangleq \left[\mu_1; \cdots; \mu_{|\mathcal{E}'|}\right] \in \mathcal{R}^{(|V^d| \cdot |\mathcal{E}'|)}$. With Eq. (15), we solve the subproblems of z-optimization, b-optimization and dual variable update in iterations, as described in Section 4.1.

4.3.1 z-optimization

We remove the terms that are irrelevant to z from the augmented Lagrangian function in Eq. (15), and obtain

$$\begin{aligned} z^{k+1} &= \underset{z \geq 0, z \in \mathcal{Z}}{\operatorname{argmin}} \left(L(z, b^k, \mu^k) \right) \\ &= \underset{z \geq 0, z \in \mathcal{Z}}{\operatorname{argmin}} \left(c \cdot z + \langle \mu^k, z - b^k \rangle + \frac{\rho}{2} \cdot \|z - b^k\|^2 \right), \end{aligned} \quad (16)$$

where $z \in \mathcal{Z}$ means that for every $e \in \mathcal{E}', z_e \in \mathcal{Z}_e \triangleq \{z_e, \sum_{i \in V^d} z_{i,e} \leq B_e\}$. With Eq. (16), we get the update rule of each z_e separately as

$$z_e^{k+1} = \underset{z_e \geq 0, z_e \in \mathcal{Z}_e}{\operatorname{argmin}} \left(c'_e \cdot z_e + \langle \mu_e^k, z_e - b_e^k \rangle + \frac{\rho}{2} \cdot \|z_e - b_e^k\|^2 \right). \quad (17)$$

Hence, z-optimization can be solved by tackling the independent optimization in Eq. (17) for each link in $\mathcal{E}'$, which is referred to as a per-link subproblem. Since Eq. (17) is just a simple quadratic problem, its solution can be obtained analytically with the help of Karush-Kuhn-Tucker (KKT) conditions [35], as explained in Appendix A.

4.3.2 b-optimization

After obtaining the optimal solution of z^{k+1}, we proceed to solve the subproblem of b and get

$$\begin{aligned} b^{k+1} &= \underset{b \geq 0}{\operatorname{argmin}} \left(L(z^{k+1}, b, \mu^k) \right) \\ &= \underset{b \geq 0}{\operatorname{argmin}} \left(f(b) + \langle \mu^k, z^{k+1} - b \rangle + \frac{\rho}{2} \cdot \|z^{k+1} - b\|^2 \right) \\ &\quad s.t. \sum_{e \in \mathcal{E}'} a_{v,e} \cdot b_e = 0, \ \forall v \in \mathcal{V}'. \end{aligned} \quad (18)$$

Note that, b-optimization here can not be decomposed based on the links, since the flow conservation constraints (*i.e.*, $\sum_{e \in \mathcal{E}'} a_{v,e} \cdot b_e = 0, \ \forall v \in \mathcal{V}'$) couple the variables across the links. Hence, solving this problem exactly, which is subject to the network constraints, would require global coordination over the network. Instead of pursuing its optimal solution, we adopt an approximate solution, which can be obtained in a distributed manner. Specifically, we apply dual decomposition to Eq. (18), but only run it for a single iteration. We write the Lagrangian function of Eq. (18) as

$$\begin{aligned} \mathcal{L}(b, \nu) = &\sum_{e \in \mathcal{E}'} \left[f_e(b_e) + \langle \mu_e^k, z_e^{k+1} - b_e \rangle + \frac{\rho}{2} \cdot \|z_e^{k+1} - b_e\|^2 \right] \\ &+ \sum_{v \in \mathcal{V}'} \langle \nu_v, \sum_{e \in \mathcal{E}'} a_{v,e} \cdot b_e \rangle, \end{aligned}$$

where $\nu_v \triangleq \left[\nu_{1,v}; \cdots; \nu_{|V^d|,v}\right] \in \mathcal{R}^{|V^d|}, \forall v \in \mathcal{V}'$ are the Lagrangian multipliers. The iteration updates b and ν as

$$b^{k+1} = \underset{b \geq 0}{\operatorname{argmin}} \left(\mathcal{L}\left(b, \nu^k\right) \right), \quad (19)$$

$$\nu^{k+1} = \nu^k + \lambda \cdot \nabla_{\nu^k} \left(\mathcal{L}\left(b^{k+1}, \nu^k\right) \right), \quad (20)$$

where $\nu \triangleq \left[\nu_1; \cdots; \nu_{|\mathcal{V}'|}\right] \in \mathcal{R}^{(|V^d| \cdot |\mathcal{V}'|)}$, λ is the dual decomposition step size, and $\nabla(\cdot)$ is the gradient of the Lagrangian function. Note that, Eqs. (19) and (20) only run for one iteration to approximate the optimal solution of Eq. (18). Then, b-optimization can be decomposed based on the links in $\mathcal{E}'$, and b_e is updated as

$$\begin{aligned} b_e^{k+1} = \underset{b_e \geq 0}{\operatorname{argmin}} \Big(&f_e(b_e) + \langle \mu_e^k, z_e^{k+1} - b_e \rangle \\ &+ \frac{\rho}{2} \cdot \|z_e^{k+1} - b_e\|^2 + \langle \nu_{e^-}^k - \nu_{e^+}^k, b_e \rangle \Big), \end{aligned} \quad (21)$$

where (e^-, e^+) is the end-node pair of the directed link e, $\nu_{e^-} \triangleq \left[\nu_{1,e^-}; \cdots; \nu_{|V^d|,e^-}\right] \in \mathcal{R}^{|V^d|}$, and $\nu_{e^+} \triangleq \left[\nu_{1,e^+}; \cdots; \nu_{|V^d|,e^+}\right] \in \mathcal{R}^{|V^d|}$. Due to the separable structure of $\mathcal{L}(b, \nu)$ with respect to the nodes in $\mathcal{V}'$, we can update ν_v in Eq. (20) for each node v as

$$\nu_v^{k+1} = \nu_v^k + \lambda \cdot \sum_{e \in \mathcal{E}'} a_{v,e} \cdot b_e^{k+1}. \quad (22)$$

The optimization in Eq. (21) is relatively easy to solve, if $f_e(b_e)$ is either linear or quadratic. For other forms of $f_e(b_e)$, a link e needs to leverage a local iterative algorithm to find its solution. Eq. (22) only involves arithmetic operations.

4.3.3 Dual Variable Update

With the solutions of z^{k+1} and b^{k+1}, we update the dual variables as

$$\mu^{k+1} = \mu^k + \rho \cdot (z^{k+1} - b^{k+1}). \quad (23)$$

4.3.4 Distributed Implementation

The overall procedure of the inexact ADMM-based distributed algorithm is shown in *Algorithm* 1, which allows for parallel implementation in an inter-DC network. Specifically, *Step* 1 initializes the parameter sets $b_{i,e}^0$ and $\mu_{i,e}^0$ on each link and $\nu_{i,v}^0$ on each DC. In the meantime, the utility functions of all the damaged DCs are broadcasted to the entire network. In each iteration, each link solves the per-link subproblem of z_e in *Step* 2. This can be easily implemented in parallel, since it only uses local variables b_e^k, μ_e^k and c_e to update variable z_e^{k+1}. Similarly, in *Step* 3, each link solves the per-link subproblem of b_e. To obtain the values of b_e^{k+1}, a link communicates with its end-nodes

to get the stored variables ν_{e^-}, ν_{e^+}, and uses them together with the local variables z_e^{k+1} and μ_e^{k+1}. In *Step* 4, each DC updates ν_v based on the values of b_e^{k+1}, which can be get by asking its adjacent links. Besides, each link updates the dual variable μ_e in *Step* 5. Note that, the per-link subproblems in *Steps* 2 and 3 are small-scale and convex, whose complexity would be low, while the variable updates in *Steps* 4 and 5 only involve arithmetic operations.

As each flow in VTEN corresponds to a data transfer scheme in the inter-DC network over time, our algorithm solves the emergency backup problem in a distributed manner. Specifically, each DC stores and updates the variables z_e^{k+1}, b_e^{k+1} and μ_e^{k+1} for each of its adjacent links by solving the per-link subproblems, which can be realized in parallel. Meanwhile, each DC also stores and updates ν_v.

4.4 Benchmarks

Meanwhile, in order to solve the problem of emergency backup, we also leverage the existing approaches to design several benchmarks as follows, which will be used in the performance evaluations in Section 6.

- **Highest utility data first (HUDF)**: This algorithm first prioritizes the damaged DCs based on their data utilities, and then uses maximum flows in the inter-DC network to evacuate data on the DCs out in sequence and repeat the procedure in each time interval.
- **VTEN based HUDF (VTEN-HUDF)**: The algorithm first constructs a VTEN with the procedure described in Section 3.1 and then applies HUDF in the VTEN.
- **Dual**: It is the standard dual decomposition method, which solves the original optimization problem in Section 3.2 using the dual gradient-ascent method [35].
- **Optimal**: It solves the original optimization in Section 3.2 directly with the optimization toolkit CVX.

Algorithm 1: Inexact ADMM-based Distributed Algorithm

1 Each link e initializes $b_{i,e}^0 = 0$ and $\mu_{i,e}^0 = 0$; Each DC initializes $\nu_{i,v}^0 = 0$ and notifies its adjacent links about the results; $f_i(\cdot)$ is broadcasted to all the links;
2 Given $b_e^k = [b_{1,e}^k, b_{2,e}^k, \cdots]$ and $\mu_e^k = [\mu_{1,e}^k, \mu_{2,e}^k, \cdots]$, each link $e \in \mathcal{E}'$ solves per-link subproblem in Eq. (17) and gets the values of z_e^{k+1} ;
3 Given $z_e^{k+1} = [z_{1,e}^{k+1}, z_{2,e}^{k+1}, \cdots]$, $\mu_e^k = [\mu_{1,e}^k, \mu_{2,e}^k, \cdots]$, $\nu_{e^-}^k$ and $\nu_{e^+}^k$, each link $e \in \mathcal{E}'$ solves per-link subproblem in Eq. (21) and gets the values of b_e^{k+1} ;
4 Each DC $v \in \mathcal{V}'$ collects the variables of its adjacent links, *i.e.*, $b^{k+1} = [b_1^{k+1}, b_2^{k+1}, \cdots]$, updates dual variables $\nu_v^{k+1} = [\nu_{1,v}^{k+1}, \nu_{1,v}^{k+1}, \cdots]$ with Eq. (22), and sends the results to its adjacent links ;
5 Each link $e \in \mathcal{E}'$ updates dual variables $\mu_e^{k+1} = [\mu_{1,e}^{k+1}, \mu_{2,e}^{k+1}, \cdots]$ with Eq. (23);
6 Return to *Step* 2 until convergence;

5 Convergence Analysis and Improvement

In this section, we first prove the convergence of the proposed inexact ADMM, which includes three steps. Then we make improvement to make the algorithm perform better.

Note that, our inexact ADMM is different from the existing ones due to the inexact primal-dual steps with Eqs. (19) and (20), which makes the convergence analysis challenging. The proof uses the following two assumptions.

Assumption 1. There exists a Slater point $(\bar{b}, \bar{z})$ of the original problem in Eq. (12), *i.e.*, $\bar{z} > 0$, $\bar{b} > 0$, and $\bar{z} = \bar{b}$, subject to $\sum_{i \in V^d} z_{i,e} < B_e, \forall e \in \mathcal{E}'$, $\sum_e a_{v,e} \cdot \bar{b}_e = 0,\ \forall\ v \in \mathcal{V}'$.

Assumption 2. The function $f(b) \triangleq \sum_e f_e(b_e)$ is strongly convex with constant M_f, *i.e.*, for any two arbitrary points b and $\tilde{b}$ in the domain of $f(b)$, we have

$$f(b) \geq f(\tilde{b}) + \langle \nabla(f(\tilde{b})), b - \tilde{b} \rangle + \frac{M_f}{2} \cdot \|b - \tilde{b}\|^2. \tag{24}$$

Assumptions 1 and 2 are commonly used in convex optimization and can usually be satisfied in practice. Roughly speaking, *Assumption* 1 means that the inequality constraints in Eq. (12) are not tight. To satisfy *Assumption* 2, we only need to design a strongly convex cost function $f(b)$.

Theorem 1. Under *Assumptions* 1 and 2, and given that the positive parameters ρ and λ satisfy

$$\begin{aligned} &M_f - \lambda \cdot \delta \geq 0,\ M_f + \frac{\rho}{2} - \frac{\lambda \cdot \delta}{2} \geq 0, \\ &M_f + \rho - \lambda \cdot \delta - 2 \cdot \lambda^2 \cdot \delta \geq 0, \end{aligned} \tag{25}$$

where δ is $|V^d|$ times the maximum singular value of A (the adjacent matrix of VTEN), the sequence (z^k, b^k, μ^k, ν^k) that is generated iteratively by *Algorithm* 1 with Eqs. (17) and (21)-(23) converges to the optimal solution of Eq. (12), *i.e.*, (z^*, b^*, μ^*, ν^*).

Theorem 1 tells how to choose the parameters for our algorithm. The penalty coefficient ρ should be sufficiently large, while the step size λ of dual decomposition in the subproblem of b should be small enough. Their tradeoff is determined by M_f, which characterizes the property of the objective function, and δ, which characterizes the VTEN's connectivity. Our proof of *Theorem* 1 includes three steps. Firstly, we prove that any optimal primal-dual solution of the optimization is bounded. Then, we verify that the ADMM iterations converge to a stationary point. Finally, we leverage the KKT condition to finish the proof by showing that the stationary point is just the optimal solution.

5.1 Boundness of Optimal Solution

Lemma 1. Under *Assumptions* 1 and 2, any optimal primal-dual solution (z^*, b^*, μ^*, ν^*) of Eq. (12) is bounded.

Proof: First of all, the constraints $z \in \mathcal{Z}$, $z = b$, $z \geq 0$, and $b \geq 0$ confine z and b to a bounded area. Thus, the optimal primal solution (z^*, b^*) is bounded. Then, we consider a Lagrangian function of Eq. (12) that dualizes the constraints $z - b = 0$ and $\sum_e a_{v,e} \cdot b_e = 0$ with dual variables $\mu \in \mathcal{R}^{(|V^d| \cdot |\mathcal{E}'|)}$ and $\nu \in \mathcal{R}^{(|V^d| \cdot |\mathcal{V}'|)}$, respectively, as

$$\begin{aligned} \widetilde{\mathcal{L}}(z, b, \mu, \nu) = &\sum_e f_e(b_e) + c \cdot z + \langle \mu, z - b \rangle \\ &+ \sum_v \langle \nu_v, \sum_e a_{v,e} \cdot b_e \rangle,\ z \in \mathcal{Z}, z \geq 0, b \geq 0. \end{aligned} \tag{26}$$

For any dual variables $\bar{\mu} \in \mathcal{R}^{(|V^d|\cdot|\mathcal{E}'|)}$ and $\bar{\nu} \in \mathcal{R}^{(|V^d|\cdot|\mathcal{V}'|)}$, we define $q(\bar{\mu},\bar{\nu}) = \min_{z\geq 0, b\geq 0}\left[\widetilde{\mathcal{L}}(z,b,\bar{\mu},\bar{\nu})\right]$. Then, for any optimal primal-dual solution (z^*,b^*,μ^*,ν^*), we have

$$\begin{aligned} q(\bar{\mu},\bar{\nu}) &\leq \widetilde{\mathcal{L}}(z^*,b^*,\bar{\mu},\bar{\nu}) \leq \widetilde{\mathcal{L}}(z^*,b^*,\mu^*,\nu^*) \\ &\leq \widetilde{\mathcal{L}}(\bar{z},\bar{b},\mu^*,\nu^*), \end{aligned} \tag{27}$$

where $(\bar{z},\bar{b})$ is any Slater point of Eq. (12). In Eq. (27), the first inequality comes from the definition $q(\bar{\mu},\bar{\nu}) = \min_{z\geq 0, b\geq 0}\left(\widetilde{\mathcal{L}}(z,b,\bar{\mu},\bar{\nu})\right)$, the second inequality is because (μ^*,ν^*) maximizes $\tilde{\mathcal{L}}(z,b,\mu,\nu)$ when $z=z^*$ and $b=b^*$, and the last inequality is due to that (z^*,b^*) minimizes $\tilde{\mathcal{L}}(z,b,\mu,\nu)$ when $\mu=\mu^*$ and $\nu=\nu^*$.

To bound ν^*, we define a vector $s_{\nu^*} \triangleq \left[s_{\nu_1^*};\cdots;s_{\nu^*_{|\mathcal{E}'|}}\right] \in \mathcal{R}^{(|V^d|\cdot|\mathcal{E}'|)}$, where $s_{\nu_e^*} \triangleq a_{v,e}\cdot\nu_v^* \in \mathcal{R}^{|V^d|}$. According to *Assumption* 1, we know that the Slater point $(\bar{b},\bar{z})$ satisfies $\bar{z}\in\mathcal{Z}$, $\bar{b}>0$, $\bar{z}>0$, $\bar{z}=\bar{b}$ and $\sum_e a_{v,e}\cdot\bar{b}_e = 0$. There exists a sufficiently small positive constant κ such that $\bar{z}-\kappa\cdot s_{v^*}\in\mathcal{Z}$, $\bar{z}-\kappa\cdot s_{v^*}>0$ and $\bar{b}-\kappa\cdot s_{\nu^*}>0$. Therefore, with Eq. (27), we have

$$\begin{aligned} q(\bar{\mu},\bar{\nu}) \leq& \widetilde{\mathcal{L}}(\bar{z}-\kappa\cdot s_{v^*},\bar{b}-\kappa\cdot s_{v^*},\mu^*,\nu^*) \\ =& \sum_e f_e(\bar{b}_e-\kappa\cdot s_{v_e^*}) + c\cdot(\bar{z}-\kappa\cdot s_{v^*}) \\ &-\kappa\cdot\sum_v\langle\nu_v^*,\sum_e a_{v,e}\cdot s_{v_e^*}\rangle. \end{aligned} \tag{28}$$

Substituting $s_{v_e^*} = a_{v,e}\cdot\nu_v^*$ into Eq. (28), we have

$$\begin{aligned} \kappa\cdot\sum_v\|\nu_v^*\|^2\cdot\sum_e\|a_{v,e}\|^2 \leq& \sum_e f_e(\bar{b}_e-\kappa\cdot s_{v_e^*}) \\ &+c\cdot(\bar{z}-\kappa\cdot s_{v^*}) - q(\bar{\mu},\bar{\nu}), \end{aligned} \tag{29}$$

and then get

$$\sum_v\|\nu_v^*\|^2 \leq \frac{\sum_e f_e(\bar{b}_e-\kappa\cdot s_{v_e^*}) + c\cdot(\bar{z}-\kappa\cdot s_{v^*}) - q(\bar{\mu},\bar{\nu})}{\kappa\cdot\min_v\left(\sum_e\|a_{v,e}\|^2\right)}. \tag{30}$$

In Eq. (30), if we let $\bar{\mu}=0$ and $\bar{\nu}=0$, the following equation holds

$$q(\bar{\mu},\bar{\nu}) = \min_{z\geq 0,b\geq 0}\left[\sum_e f_e(b_e)+c\cdot z\right], \tag{31}$$

which is bounded. Therefore, the right hand side of Eq. (30) is bounded, namely, ν_v^* is bounded and so as ν^*.

Next, we define a sign vector $s_{\mu^*}\in\mathcal{R}^{(|V^d|\cdot|\mathcal{E}'|)}$ whose element is 1 if the corresponding element of μ^* is positive, -1 if the corresponding element of μ^* is negative, and 0 otherwise. Apparently, we have $\langle\mu^*,s_{\mu^*}\rangle = \|\mu^*\|_1$. Given a Slater point $(\bar{z},\bar{b})$ of Eq. (12), there exists a sufficiently small positive constant κ such that $\bar{z}-\kappa\cdot s_{\mu^*}>0$, $\bar{z}-\kappa\cdot s_{\mu^*}\in\mathcal{Z}$ $\bar{z}=\bar{b}$, and $\sum_e a_{v,e}\cdot\bar{b}_e=0$, and hence from Eq. (27), we have

$$q(\bar{\mu},\bar{\nu}) \leq \widetilde{\mathcal{L}}(\bar{z}-\kappa\cdot s_{\mu^*},\bar{b},\mu^*,\nu^*). \tag{32}$$

By expanding the right side of Eq. (32) and substituting the equations $\sum_e a_{v,e}\cdot\bar{b}_e=0$ and $\bar{z}=\bar{b}$ into it, we have

$$\begin{aligned} q(\bar{\mu},\bar{\nu}) &\leq \sum_e f_e(\bar{b}_e) + c\cdot(\bar{z}-\kappa\cdot s_{\mu^*}) - \kappa\cdot\langle\mu^*,s_{\mu^*}\rangle \\ &= \sum_e f(\bar{b}_e) + c\cdot(\bar{z}-\kappa\cdot s_{\mu^*}) - \kappa\cdot\|\mu^*\|_1. \end{aligned} \tag{33}$$

In Eq. (33), if we let $\bar{\mu}=0$ and $\bar{\nu}=0$, the following inequality holds

$$\begin{aligned} \|\mu^*\|_1 \leq \frac{1}{\kappa}\cdot\Bigg\{&\sum_e f(\bar{b}_e) + c\cdot(\bar{z}-\kappa\cdot s_{\mu^*}) \\ &-\min_{b\geq 0,z\geq 0}\left[\sum_e f_e(b_e)+c\cdot z\right]\Bigg\}. \end{aligned} \tag{34}$$

Finally, we complete the proof. □

5.2 Convergence to Stationary Point

Next, we show an inequality that is critical to the convergence analysis.

Lemma 2. Under *Assumptions* 1 and 2, given that the parameter λ satisfies $M_f-\lambda\delta\geq 0$, the sequence (x^k,y^k,μ^k,ν^k) satisfies

$$\begin{aligned} &\frac{1}{2\rho}\cdot\|(\mu^{k+1}-\mu^k)\|^2 + (M_f+\frac{\rho}{2}-\frac{\lambda\cdot\delta}{2})\|b^{k+1}-b^k\|^2 \\ \leq&\frac{1}{2\lambda}\cdot\left(\|\nu^k-\nu^*\|^2-\|\nu^{k+1}-\nu^*\|^2\right) \\ &+\frac{1}{2\lambda}\cdot\left(\|\nu^k-\nu^{k-1}\|^2-\|\nu^{k+1}-\nu^k\|^2\right) \\ &+\frac{1}{2}\cdot(M_f+\rho-\lambda\cdot\delta)\cdot\left(\|b^*-b^k\|^2-\|b^*-b^{k+1}\|^2\right) \\ &+\frac{1}{2\rho}\cdot\left(\|\mu^*-\mu^k\|^2-\|\mu^*-\mu^{k+1}\|^2\right), \end{aligned} \tag{35}$$

Proof: The proof is given in Appendix B. □

Lemma 3. Under *Assumptions* 1 and 2, the iterations in *Algorithm* 1 converge to a stationary point, given that the parameters ρ and λ are set to satisfy $M_f-\lambda\delta\geq 0, M_f+\frac{\rho}{2}-\frac{\lambda\cdot\delta}{2}\geq 0$ and $M_f+\rho-\lambda\cdot\delta-2\cdot\lambda^2\cdot\delta\geq 0$.

Proof: By summing up Eq. (35) for $k\in[1,2,\cdots,K-1]$ and using telescopic cancelation, we get

$$\begin{aligned} &\frac{1}{2\rho}\cdot\sum_{k=1}^{K-1}\left[\|(\mu^{k+1}-\mu^k)\|^2+(M_f+\frac{\rho}{2}-\frac{\lambda\cdot\delta}{2})\cdot\|b^{k+1}-b^k\|^2\right] \\ \leq&\frac{1}{2\lambda}\cdot\left(\|\nu^1-\nu^*\|^2-\|\nu^K-\nu^*\|^2\right) \\ &+\frac{1}{2\lambda}\cdot\left(\|\nu^0-\nu^1\|^2-\|\nu^{K-1}-\nu^K\|^2\right) \\ &+\frac{1}{2}(M_f+\rho-\lambda\cdot\delta)\left(\|b^1-b^*\|^2-\|b^K-b^*\|^2\right) \\ &+\frac{1}{2\rho}\cdot\left(\|\mu^1-\mu^*\|^2-\|\mu^K-\mu^*\|^2\right) \\ \leq&\frac{1}{2\lambda}\cdot\|\nu^1-\nu^*\|^2+\frac{1}{2\lambda}\cdot\|\nu^0-\nu^1\|^2 \\ &+\frac{1}{2}(M_f+\rho-\lambda\cdot\delta)\|b^1-b^*\|^2+\frac{1}{2\rho}\cdot\|\mu^1-\mu^*\|^2. \end{aligned} \tag{36}$$

The second inequality holds because $\lambda>0$, $\rho>0$, and $M_f+\rho-\lambda\cdot\delta\geq 0$ by hypothesis.

Note that, it is easy to ensure that (z^1,b^1,μ^1,ν^1) is bounded with the proper initialization of (z^0,b^0,μ^0,ν^0). In the meanwhile, we have already proved in Section 5.1 that (z^*,b^*,μ^*,ν^*) is bounded. Hence, the right hand side of Eq. (36) can be replaced by a constant. According to [27], for $k\to\infty$, the following equalities must hold, *i.e.*,

$$\mu^{k+1}=\mu^k \text{ and } b^{k+1}=b^k, \text{ if } k\to\infty. \tag{37}$$

By combining the update rules in Eqs. (16) and (37), we get

$$z^{k+1} = z^k, \text{ if } k \to \infty. \tag{38}$$

Then, we show that ν^k also converges to a stationary point such that

$$\nu^{k+1} = \nu^k, \text{ if } k \to \infty. \tag{39}$$

Since we have

$$\|\nu_v^{k+1} - \nu_v^k\|^2 = \lambda^2 \cdot \| \sum_e a_{v,e} \cdot b_e^{k+1}\|^2 \text{ and } \sum_e a_{v,e} \cdot b_e^* = 0,$$

which lead to

$$\begin{aligned} \|\nu_v^{k+1} - \nu_v^k\|^2 &= \lambda^2 \cdot \| \sum_e a_{v,e} \cdot (b_e^{k+1} - b_e^*)\|^2 \\ &\leq \lambda^2 \cdot \delta \cdot \|b^{k+1} - b^*\|^2, \end{aligned} \tag{40}$$

where the inequality can be satisfied as $\| \sum_e a_{v,e} \cdot (b_e^{k+1} - b_e^*)\|^2 \leq \sum_e (a_{v,e})^2 \cdot \sum_e \|b_e^{k+1} - b_e^*\|^2 \leq \delta \cdot \|b^{k+1} - b^*\|^2$, where δ is $|V^d|$ times the largest singular value of A.

By combining Eqs. (35) and (40), we have

$$\begin{aligned} &\frac{1}{2\rho} \cdot \|(\mu^{k+1} - \mu^k)\|^2 + \|\nu_v^{k+1} - \nu_v^k\|^2 \\ &\quad + (M_f + \frac{\rho}{2} - \frac{\lambda \cdot \delta}{2}) \|b^{k+1} - b^k\|^2 \\ \leq &\frac{1}{2\lambda} \cdot \left(\|\nu^k - \nu^*\|^2 - \|\nu^{k+1} - \nu^*\|^2 \right) \\ &\quad + \frac{1}{2\lambda} \cdot \left(\|\nu^k - \nu^{k-1}\|^2 - \|\nu^{k+1} - \nu^k\|^2 \right) \\ &\quad + \frac{1}{2} \left(M_f + \rho - \lambda \cdot \delta - 2\lambda^2 \cdot \delta \right) \left(\|b^* - b^k\|^2 - \|b^* - b^{k+1}\|^2 \right) \\ &\quad + \frac{1}{2\rho} \cdot \left(\|\mu^* - \mu^k\|^2 - \|\mu^* - \mu^{k+1}\|^2 \right) \\ &\quad - \frac{1}{2} (M_f + \rho - \lambda \cdot \delta - 2\lambda^2 \cdot \delta) \cdot \|b^* - b^k\|^2. \end{aligned} \tag{41}$$

Then, we choose the positive parameters ρ and λ such that

$$M_f + \frac{\rho}{2} - \frac{\lambda \cdot \delta}{2} \geq 0, \text{ and } M_f + \rho - \lambda \cdot \delta - 2\lambda^2 \cdot \delta \geq 0.$$

Using the same method of deriving Eq. (37) from Eq. (35), we can verify Eq. (39). Finally, Eqs. (37)-(39) prove that the iterations in *Algorithm* 1 converge to a stationary point

$$\lim_{k \to \infty} (z^k, b^k, \mu^k, \nu^k) = (z^\infty, b^\infty, \mu^\infty, \nu^\infty).$$

□

5.3 Convergence to Optimal Solution

Now, we are ready to prove *Theorem* 1.

Proof: In order to prove *Theorem* 1, by *Lemma* 3, we need to verify that the stationary point $(z^\infty, b^\infty, \mu^\infty, \nu^\infty)$ satisfies the KKT conditions of Eq. (12), which means that we need to check its primal feasibility, dual feasibility, complementarity slackness and stationarity.

- **Primal feasibility**: The update rules of z and b in Eqs. (16) and (19), respectively, guarantee that $z^\infty \geq 0$, $z^\infty \in \mathcal{Z}$ and $b^\infty \geq 0$. Since the update rule of μ in Eq. (23) satisfies $\mu^\infty = \mu^\infty + \rho \cdot (z^\infty - b^\infty)$ for $z^\infty = b^\infty$, the update rule of ν_v in Eq. (22) satisfies $\nu_v^\infty = \nu_v^\infty + \sum_e a_{v,e} \cdot b^\infty$ such that $\sum_e a_{v,e} \cdot b^\infty = 0$. Therefore, z^∞ and b^∞ satisfy the constraints in Eq. (12).

Algorithm 2: Improved Inexact ADMM-based Distributed Algorithm

```
k = 0, b^(0) = 0, μ^(0) = 0, ν^(0) = 0, n = 0;
while the solution has not converged do
    solve Eq. (17) for each link e ∈ E' and get values of z^(k+1);
    b^(k+1)(0) = b^(k), ν^(k+1)(0) = ν^(k);
    while n ≤ N do
        solve Eq. (46) for each link e ∈ E' and get values of b^(k+1)(n+1);
        update dual variables of ν^(k+1)(n+1) with Eq. (47);
        n = n + 1;
    end
    b^(k+1) = b^(k+1)(n), ν^(k+1) = ν^(k+1)(n);
    update dual variables μ^(k+1) with Eq. (23);
    k = k + 1;
end
```

- **Dual feasibility**: As Lagrangian multipliers μ and ν are introduced for equality constraints, they are feasible.
- **Complementarity slackness**: The complementarity slackness condition is void because only the equality constraints in Eq. (12) are dualized.
- **Stationarity**: Basically, we need to show

$$z^\infty = \underset{z \in \mathcal{Z}}{\operatorname{argmin}} \left[\widetilde{\mathcal{L}}(z, b^\infty, \mu^\infty, \nu^\infty) \right], \tag{42}$$

$$b^\infty = \underset{b \geq 0}{\operatorname{argmin}} \left[\widetilde{\mathcal{L}}(z^\infty, b, \mu^\infty, \nu^\infty) \right]. \tag{43}$$

Then, for z, Eq. (42) is equivalent to

$$\begin{aligned} & z^\infty = \underset{z \in \mathcal{Z}}{\operatorname{argmin}} \left[c \cdot z + \langle \mu^\infty, z - b^\infty \rangle \right], \\ \Leftrightarrow \quad & z^\infty = \underset{z \in \mathcal{Z}}{\operatorname{argmin}} \left(c \cdot z + \langle \mu^\infty, z - b^\infty \rangle + \frac{\rho}{2} \cdot \|z - z^\infty\|^2 \right), \\ \Leftrightarrow \quad & z^\infty = \underset{z \in \mathcal{Z}}{\operatorname{argmin}} \left(c \cdot z + \langle \mu^\infty, z - b^\infty \rangle + \frac{\rho}{2} \cdot \|z - b^\infty\|^2 \right), \end{aligned} \tag{44}$$

where the second line is because $\|z - z^\infty\|^2$ would be 0 when $z = z^\infty$, and the last line is obtained by leveraging the primal feasibility $z^\infty = b^\infty$. Hence, it is obvious that Eq. (44) matches with the update rule of z in Eq. (16). Similarly, Eq. (43) is equivalent to

$$\begin{aligned} & b^\infty = \underset{b \geq 0}{\operatorname{argmin}} \left[\sum_e f_e(b_e) + \langle \mu^\infty, z^\infty - b \rangle \right. \\ & \qquad \left. + \sum_v \nu_v^\infty \cdot \sum_e a_{v,e} \cdot b_e \right], \\ \Leftrightarrow \quad & b^\infty = \underset{b \geq 0}{\operatorname{argmin}} \left[\sum_e f_e(b_e) + \langle \mu^\infty, z^\infty - b \rangle \right. \\ & \qquad \left. + \frac{\rho}{2} \|b^\infty - b\|^2 + \sum_v \nu_v^\infty \cdot \sum_e a_{v,e} \cdot b_e \right], \\ \Leftrightarrow \quad & b^\infty = \underset{b \geq 0}{\operatorname{argmin}} \left[\sum_e f_e(b_e) + \langle \mu^\infty, z^\infty - b \rangle \right. \\ & \qquad \left. + \frac{\rho}{2} \|z^\infty - b\|^2 + \sum_v \nu_v^\infty \cdot \sum_e a_{v,e} \cdot b_e \right]. \end{aligned} \tag{45}$$

Eq. (45) matches with the update rule of b in Eq. (19).

Therefore, we verify that the stationary point $(z^\infty, b^\infty, \mu^\infty, \nu^\infty)$ of *Algorithm* 1 satisfies the KKT conditions in Eq. (12), and thus prove *Theorem* 1. □

5.4 Algorithm Improvement

Since our ADMM-based algorithm uses the inexact approach that introduces Lagrangian multipliers in the subproblem of b-optimization and solves Eq. (18) inexactly with Eqs. (19) and (20), its convergence speed is affected by the accuracy of the inexact solution. This, however, restricts the algorithm's convergence speed, even though we have verified the convergence of *Algorithm* 1 theoretically before. Observe that if we run the primal-dual updates in Eqs. (19) and (20) for multiple times, we can obtain more accurate solutions to Eq. (18), which would improve the convergence speed of the proposed algorithm. This actually motivates us to replace the updates in Eqs. (21) and (22) to

$$b_e^{k+1}(n+1) = \underset{b_e \geq 0}{\operatorname{argmin}} \Big[f_e(b_e) + \langle \mu_e^k, z_e^{k+1} - b_e \rangle + \frac{\rho}{2} \cdot \| z_e^{k+1} - b_e \|^2 + \langle \nu_{e^-}^k(n) - \nu_{e^+}^k(n), b_e \rangle \Big], \quad (46)$$

$$\nu_v^{(k+1)}(n+1) = \nu_v^{(k+1)}(n) + \lambda \cdot \sum_{e \in \mathcal{E}'} a_{v,e} \cdot b_e^{(k+1)}(n+1), \quad (47)$$

where n is the iteration number for the inner loop to solve the subproblem of b-optimization. The improved algorithm is shown in *Algorithm* 2, in which *Lines* 5-9 is the inner loop for solving the subproblem of b-optimization iteratively and N is the preset iteration number for the inner loop.

6 Performance Evaluation

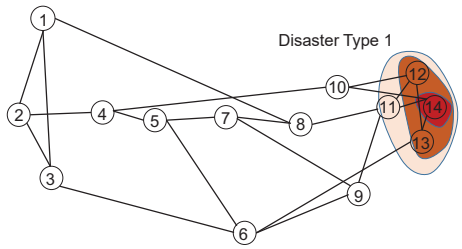
6.1 Simulation Setup

We evaluate the performance of our proposed algorithm with the NSFNET and US-Backbone topologies shown in Fig. 3. Each node in the topologies is a DC node, which can be impacted by a progressive disaster. Our simulations consider different disaster scenarios, in which the number of damaged DCs $|V^d|$ is selected within $[2, 4]$ and $[2, 6]$ for NSFNET and US-Backbone, respectively. For example, three disaster scenarios are considered in NSFNET with V^d as {14,13}, {14,13,12} and {14,13,12,11}, respectively. The emergency backup window of each damaged DC can change in different simulations. Moreover, in US-Backbone, we consider two types of progressive disasters, *i.e.*, the one that spreads from the edge of the network (Disaster Type 1) and the one that spreads from the center of the network (Disaster Type 2), as shown in Fig. 3(b).

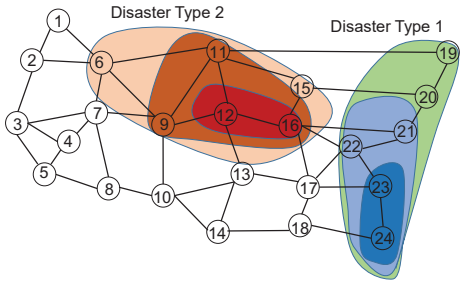
We assume that in each emergency backup, the damaged DCs need to evacuate a total amount of 400 TBytes data and the total available storage space on the safe DCs is 500 TBytes, while the actual amount of data to be backed up on each damaged DC and the available storage space on each safe DC are random. The simulations use concave utility functions that satisfy the law of diminishing marginal return [12]. Specifically, the utility function can take the form of a logarithmic function

$$f_i(s_i) = \alpha_i \cdot \log(1 + s_i), \quad (48)$$

where s_i is the amount of data that has been successfully backed up for DC i, and α_i is a constant coefficient. We



(a) NSFNET



(b) US-Backbone

Fig. 3. Inter-DC network topologies used in simulations.

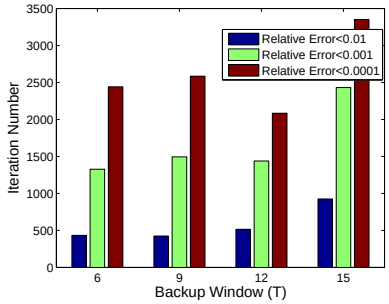
set $\alpha_i \in \{100, 120, 150, 200\}$ for *DCs* 11-14 in NSFNET and $\alpha_i \in \{100, 120, 150, 200, 160, 180\}$ for *DCs* 19-24 or {6,9,11,12,15,16} in US-Backbone. At the beginning of each disaster, the available bandwidth on each link is randomly chosen within $[30, 80]$ Gbps, and the corresponding unit bandwidth cost (*i.e.*, c_e) ranges within $[0.001, 0.0015]$ unit per TByte. The simulations run in MATLAB on a computer with 3.1 GHz Intel Core i3-2100 CPU and 8 GB RAM.

6.2 Evaluation of Convergence Performance

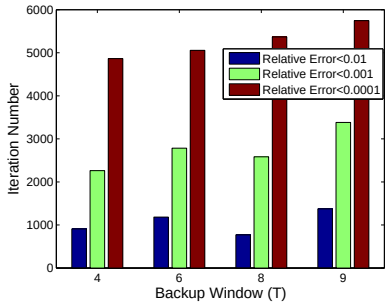
We first evaluate the convergence performance of our proposed algorithm. We use the relative error as the performance metric, which represents the difference between the algorithm's outcome and the optimal solution. Specifically, it is defined as $(S^* - S)/S^*$, where S is the backup profit obtained by our ADMM-based algorithm and S^* is the optimal profit that is obtained by directly solving the original optimization problem in Section 3.2. The simulations consider the disaster scenarios in Fig. 3 and use different combinations of $|V^d|$ and T. Without loss of generality, we select the most complicated backup cases and change backup window T. Specifically, the simulations use $|V^d| = 4$ and change T within $\{6, 9, 12, 15\}$ in NSFNET, while they have $|V^d| = 6$ and select T from $\{4, 6, 8, 9\}$ and $\{3, 6, 9, 12\}$ for Types 1 and 2 disasters in US-Backbone, respectively[1].

Fig. 4 plots the results on the number of iterations that *Algorithm* 1 uses to reach certain relative error requirements in different disaster scenarios. The results indicate that to address the most complicated backup case in NSFNET (*i.e.*, $|V^d| = 4$ and $T = 15$), the algorithm achieves a relative error less than 10^{-4} within 3500 iterations. Meanwhile, for the most complicated Type 1 (*i.e.*, $|V^d| = 6$ and $T = 9$) and Type 2 (*i.e.*, $|V^d| = 6$ and $T = 12$) disasters in US-Backbone,

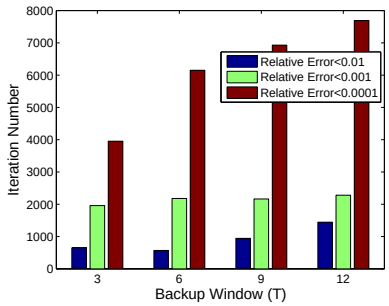
1. We select the largest value of T for each disaster scenario such that by directly solving the original optimization in Section 3.2, we can obtain the optimal profit within reasonably long time (*e.g.*, 2 hours).



(a) Disasters in NSFNET with $|V^d| = 4$



(b) Type 1 disasters in US-Backbone with $|V^d| = 6$



(c) Type 2 disasters in US-Backbone with $|V^d| = 6$

Fig. 4. Number of iterations for *Algorithm* 1 to meet certain relative error requirements in different disaster scenarios.

the algorithm achieves the same relative error within 6000 and 8000 iterations, respectively.

We then run simulations to investigate the performance of *Algorithm* 2 and the influence of N on its convergence speed. To show the influence intuitively, we plot the results on the absolute error (*i.e.*, defined as $S^* - S$) *versus* the algorithm's running time for different N in Fig. 5. The results in Fig. 5 are obtained when we consider the most complicated backup case in response to Type 1 disasters in US-Backbone (*i.e.*, $|V^d| = 6$ and $T = 9$)[2]. It can be seen that by increasing N from 1 (*i.e.*, when *Algorithm* 2 becomes *Algorithm* 1) to 5, the convergence speed gets improved obviously. Hence, the running time required to ensure relative error $< 10^{-4}$ gets reduced significantly, *i.e.*, from 86.86 seconds to 34.77 seconds. However, if we keep increasing N to 10, there is

2. Note that, for the evaluations in Sections 6.2 and 6.3, we also simulate the most complicated backup cases in NSFNET and for Type 2 disasters in US-Backbone, and confirm that the results follow the similar trends. However, due to the page limit, we omit those results.

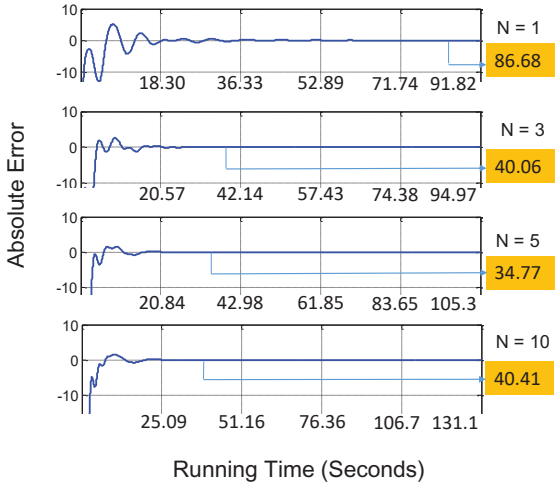


Fig. 5. Convergence speed of *Algorithm* 2 (Type 1 disaster in US-Backbone with $T = 9$ and $|V^d| = 6$).

no noticeable decrease on the running time. This is because with $N = 5$, *Algorithm* 2 can already get a reasonably accurate solution to the subproblem of b-optimization. Therefore, if we keep increasing N, the reduction on the running time will be offset by the increase of the time spent on the inner loop for the b-optimization, which increases with N. To this end, we can see that $N = 5$ is a reasonably good setting for *Algorithm* 2. In the subsequent simulations, we set $N = 5$.

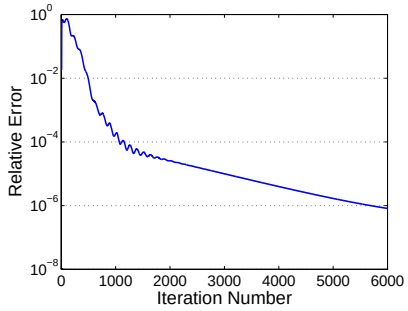
6.3 Analysis of Generality and Robustness

To verify that *Algorithm* 2 can handle different practical situations, we analyze its generality and robustness. Firstly, we change the expression of the utility function. Instead of using the logarithmic utility function in Eq. (48), we test the quadratic function in Eq. (49), and a mixture of Eqs. (48) and (49) (*i.e.*, the utility functions on certain DCs take the form of Eq. (48) while others use Eq. (49)) too.

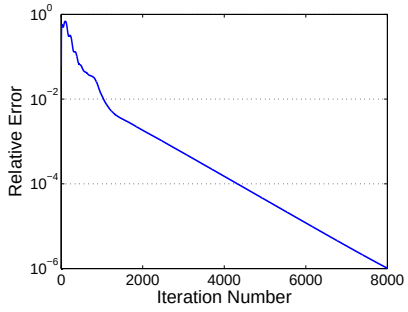
$$f_i(s_i) = -\frac{\alpha_i}{2C_i} \cdot s_i^2 + \alpha_i \cdot s_i, \tag{49}$$

where C_i is the total amount of data to be backed up in a damaged DC i. Here, we still consider the most complicated backup case for Type 1 disasters in US-Backbone (*i.e.*, $|V^d| = 6$ and $T = 9$). In Fig. 6(a), we observe that when only the quadratic utility function is used, the relative error results decrease quickly and become smaller than 10^{-4} after 1157 iterations, which confirms the convergence performance of *Algorithm* 2 in this scenario. In Fig. 6(b), when the DCs use the two utility functions mixedly (*i.e.*, *DCs* 19-21 use the logarithmic function in Eq. (48) and *DCs* 22-24 take the quadratic function in Eq. (49)), the algorithm converges slower and uses 4323 iterations to achieve a relative error $< 10^{-4}$. These results verify that our proposed algorithm can handle different utility functions well.

Then, we investigate the robustness of *Algorithm* 2 (ADMM) by comparing it with a standard dual decomposition method (Dual), which is a classical distributed method to solve large-scale optimization problems [35]. Specifically, the dual decomposition method solves the original problem Eq. (12) in Section 3.2 using the dual gradient-ascent



(a) Quadratic utility function



(b) Mixed utility functions

Fig. 6. Convergence performance of *Algorithm* 2 with different utility functions (Type 1 disaster in US-Backbone with $T = 9$ and $|V^d| = 6$).

method, where the subproblems for variables $\{s_i\}$ and $\{b_{i,e}\}$ are solved iteratively to converge to the optimal solution [5]. The robustness of the algorithms can be evaluated by changing the value of the dual decomposition step size. Specifically, in ADMM, we choose the step size λ as a fixed value, while in Dual, we use the diminishing step size $\lambda_k = \beta/\sqrt{k}$, where k is the iteration number and β is an adjustable coefficient [5]. We still use US-Backbone with $T = 9$ and $|V^d| = 6$, and check the results of iteration number and running time used to reach certain relative error requirements. Specifically we set the relative error requirement for ADMM and Dual as 10^{-4} and 10^{-2}, respectively. Then, we change the values of λ and β and evaluate the two algorithms.

In the simulations, we set the maximum iteration numbers of ADMM and Dual as 6000 and 40000, respectively. As shown in Table 1, we observe that for $\lambda \in [0.008, 0.1]$, ADMM can always converge to meet its relative error requirement, and to achieve this, it takes fewer iterations and less running time than Dual, even though its relative error requirement is tighter. On the other hand, Dual can only meet its relative error requirement when $\beta \in [0.003, 0.006]$, which suggests that it is very sensitive to the step size. When β is too small, it will take very long running time to converge, while if β is a litter bit larger, its results tend to oscillate. Fig. 7 shows such a comparison of ADMM and Dual, when we choose $\beta = 0.003$ and 0.0055. With $\beta = 0.003$, Dual takes 38864 iterations to meet the relative error requirement, while with $\beta = 0.0055$, the results from Dual oscillate a lot. On the contrary, ADMM takes 3078 and 1656 iterations to meet the relative error requirement of 10^{-4} with $\lambda = 0.008$ and 0.1, respectively, and the relative error of ADMM always stays at a stable value after it has converged.

TABLE 1
Comparisons on Convergence Performance of ADMM and Dual (Type 1 Disaster in US-Backbone with $T = 9$ and $|V^d| = 6$)

λ (ADMM)	0.008	0.01	0.02	0.05	0.1
Iteration number	3078	2763	2278	1683	1656
Time (seconds)	64.47	57.68	49.77	37.59	35.05
β (Dual)	0.003	0.004	0.005	0.0055	0.006
Iteration number	38864	26647	19825	19075	22708
Time (seconds)	170.6	112.0	82.30	78.94	98.89

Here we would like to point out that the proposed algorithm performs better with quadratic and mixed functions (*cf.* Fig. 6), which have better strong convexity properties. This observation corroborates the theoretical analysis. Finally, we can conclude that ADMM has reasonably robust and its performance would not be affected by λ significantly.

6.4 Comparison with Benchmarks

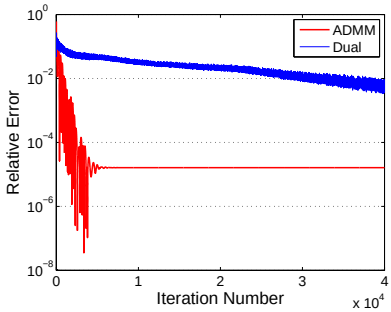
Finally, we compare our proposed algorithms with the benchmarks discussed in Section 4.4. The results of simulations using NSFNET and US-Backbone are shown in Tables 2 and 3, respectively. The results in Table 2 indicate that ADMM can obtain the profits that are the closest to those from Optimal, and it is followed by Dual. In NSFNET, the profits from VTEN-HUDF are 4.2% higher than HUDF on average, since building the VTEN helps to improve the utilization of DC storage space[3]. On average, the running time of ADMM is only 21.9% and 37.9% of that of Dual and Optimal, respectively, which indicates that ADMM is much more time-efficient. HUDF takes the shortest running time, and it is followed by VTEN-HUDF. In US-Backbone, we consider the two types of disasters in Fig. 3(b) and get the results in Table 3 (*i.e.*, Type 1 disasters with $T = \{6, 9, 12\}$ and Type 2 disasters with $T = \{15, 20\}$). The results show the similar trends as those obtained in NSFNET.

Therefore, we can conclude that among the algorithms, ADMM achieves the best tradeoff between backup profit and running time for all the simulated disaster scenarios. Meanwhile, we hope to point out that the running time of ADMM can still be reduced in two aspects. Firstly, instead of using MATLAB, we can implement ADMM in C/C++ platforms, which are known to be much more time-efficient. Secondly, the distributed ADMM algorithm is now implemented in a serial manner, but since it can solve the per-link subproblems in parallel, the running time would be reduced significantly with a distributed implementation.

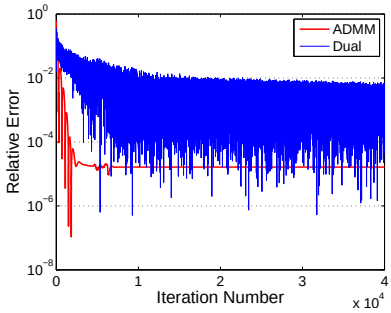
7 Conclusion

In this paper, we addressed the emergency backup in inter-DC networks with progressive disasters. We first utilized the TEN approach to model the time-variant inter-DC network with a progressive disaster as a VTEN and converted the dynamic flow scheduling in the network to a static one.

3. Note that, since we define the optimization objective of emergency backup as maximizing the total profit (*i.e.*, in Eq. (1)), a higher profit suggests a better emergency backup scheme, which means that more critical data is evacuated with fewer network resources. Also, when we set $|V^d| = 4$ and $T = 30$ in NSFNET or $T > 9$ in US-Backbone, Optimal cannot get a solution within reasonably long time (*e.g.*, 2 hours).



(a) $\lambda = 0.008, \beta = 0.003$



(b) $\lambda = 0.1, \beta = 0.0055$

Fig. 7. Convergence speeds of ADMM and Dual with different λ and β (Type 1 disaster in US-Backbone with $T = 9$ and $|V^d| = 6$).

TABLE 2
Performance Comparisons using NSFNET

	Type 1 Disasters				
$\|V^d\|$	2	3		4	
T	6	9	15	20	30
	Backup Profit (units)				
Optimal	303.47	539.66	641.96	912.66	–
HUDF	292.97	529.71	593.73	855.67	1186.8
VTEN-HUDF	302.31	535.79	639.11	908.01	1222.3
Dual	303.13	538.52	640.96	902.87	1224.6
ADMM	303.17	539.12	641.32	911.76	1228.5
	Running Time (seconds)				
Optimal	6.069	28.976	58.825	29.057	–
HUDF	0.008	0.016	0.126	0.063	0.076
VTEN-HUDF	0.07	0.206	1.667	1.691	5.446
Dual	8.475	30.571	64.502	74.01	762.2
ADMM	3.145	9.126	14.42	12.3	25.463

Then, with the VTEN, we formulated an optimization to maximize the profit from the emergency backup, and designed an ADMM-based distributed algorithm to solve it time-efficiently. The convergence of the proposed algorithm was also verified theoretically. Finally, we conducted extensive simulations to demonstrate that our proposed algorithm is robust and time-efficient, and outperforms several benchmarks in terms of backup profit and running time.

TABLE 3
Performance Comparisons using US-Backbone

	Type 1 Disasters			Type 2 Disasters	
$\|V^d\|$	2	3	4	5	6
T	6	9	12	15	20
	Backup Profit (units)				
Optimal	267.01	572.14	–	–	–
HUDF	238.76	563.77	869.95	1039.3	1382.3
VTEN-HUDF	263.18	571.74	895.21	1046.2	1399.4
Dual	266.82	571.80	896.33	1071.6	1406.3
ADMM	267.01	572.13	897.06	1073.4	1408.9
	Running Time (seconds)				
Optimal	12.232	84.498	–	–	–
HUDF	0.058	0.068	0.126	0.185	0.123
VTEN-HUDF	0.202	1.055	4.705	4.94	12.15
Dual	19.171	28.364	143.33	109.66	450.87
ADMM	5.7	18.278	26.289	29.76	64.44

APPENDIX A
SOLUTION OF EQUATION (17)

With KKT conditions, we can obtain the solution of Eq.(17) [36]. Specifically, For each link $e \in \mathcal{E}'$, we consider all flows from damaged DCs $i \in V^d$. For DC i that satisfies $\rho \cdot b_{i,e}^k - \mu_{i,e}^k - c_e \leq 0$, we have $z_{i,e}^{k+1} = 0$. Then, we denote the set of remaining DCs, *i.e.*, $\{i : i \in V^d, \rho \cdot b_{i,e}^{(k)} - \mu_{i,e}^k - c_e > 0\}$, as Ψ_e^{k+1}, and obtain

$$z_{i,e}^{(k+1)} = \begin{cases} b_{i,e}^{(k)} - \dfrac{\mu_{i,e}^{(k)} + c_e}{\rho}, \sum\limits_{i \in \Psi_e^{k+1}} \left(b_{i,e}^{(k)} - \dfrac{\mu_{i,e}^{(k)} + c_e}{\rho} \right) \leq B_e, \\ \max\left(b_{i,e}^{(k)} - \dfrac{\mu_{i,e}^{(k)} + \gamma_e^{(k+1)} + c_e}{\rho}, 0 \right), \text{ Otherwise,} \end{cases} \tag{50}$$

where the parameter $\gamma_e^{k+1} \geq 0$ is determined by $\sum\limits_{i \in V^d} z_{i,e}^{k+1} = B_e$.

APPENDIX B
PROOF OF LEMMA 2

For z, according to its update rule in Eq. (16), we can obtain

$$\begin{aligned} z^{k+1} &= \underset{z \in \mathcal{Z}}{\operatorname{argmin}} \left(c \cdot z + \frac{\rho}{2} \cdot \| z - b^k + \frac{\mu^k}{\rho} \|^2 \right) \\ &= \underset{z \in \mathcal{Z}}{\operatorname{argmin}} \left(c \cdot z + \langle \mu^k, z \rangle + \frac{\rho}{2} \cdot \| z - z^k \|^2 + \rho \cdot \langle z, z^k - b^k \rangle \right). \end{aligned} \tag{51}$$

Then, we define an auxiliary function as

$$\phi(z) = c \cdot z + \langle \mu^k, z \rangle + \frac{\rho}{2} \cdot \| z - z^k \|^2 + \rho \cdot \langle z, z^k - b^k \rangle.$$

Since $z^{k+1} = \underset{z \in \mathcal{Z}}{\operatorname{argmin}} \phi(z)$, where $\phi(z)$ is strongly convex with constant ρ, we have $\langle \nabla_{z^{k+1}} \left[\phi(z^{k+1}) \right], z - z^{k+1} \rangle \geq 0, \ \forall z \in \mathcal{Z}$. Then, we can get

$$\begin{aligned} \phi(z^*) &\geq \phi(z^{k+1}) + \langle \nabla \left[\phi(z^{k+1}) \right], z^* - z^{k+1} \rangle + \frac{\rho}{2} \cdot \| z^* - z^{k+1} \|^2 \\ &\geq \phi(z^{k+1}) + \frac{\rho}{2} \cdot \| z^* - z^{k+1} \|^2, \end{aligned}$$

which comes from the strong convexity of $\phi(z)$. By expanding $\phi(z^*)$ and $\phi(z^{k+1})$, we can obtain

$$\begin{aligned} &c\cdot z^* + \langle \mu^k, z^*\rangle + \frac{\rho}{2}\cdot\|z^* - z^k\|^2 + \rho\cdot\langle z^*, z^k - b^k\rangle \\ \geq& c\cdot z^{k+1} + \langle \mu^k, z^{k+1}\rangle + \frac{\rho}{2}\cdot\|z^{k+1} - z^k\|^2 \\ &+ \rho\cdot\langle z^{k+1}, z^k - b^k\rangle + \frac{\rho}{2}\cdot\|z^* - z^{k+1}\|^2. \end{aligned} \tag{52}$$

which can be reorganized as

$$\begin{aligned} &c\cdot(z^{k+1} - z^*) + \frac{\rho}{2}\cdot\|z^{k+1} - z^k\|^2 \\ \leq& \frac{\rho}{2}\cdot\|z^* - z^k\|^2 - \frac{\rho}{2}\cdot\|z^* - z^{k+1}\|^2 + \langle \mu^k, z^* - z^{k+1}\rangle \\ &+ \rho\cdot\langle z^* - z^{k+1}, z^k - b^k\rangle. \end{aligned} \tag{53}$$

Next, we consider b and get the following equation according to its update rule in Eq. (19).

$$\begin{aligned} b^{k+1} &= \underset{b\geq 0}{\operatorname{argmin}} \left(\sum_e f_e(b_e) + \langle \mu^k, z^{k+1} - b\rangle + \frac{\rho}{2}\cdot\|z^{k+1} - b\|^2 \right. \\ &\qquad \left. + \sum_v \langle \nu_v^k, \sum_e a_{v,e}\cdot b_e\rangle \right) \\ &= \underset{b\geq 0}{\operatorname{argmin}} \left(\sum_e f_e(b_e) - \langle \mu^k, b\rangle + \frac{\rho}{2}\cdot\|b - b^k\|^2 \right. \\ &\qquad \left. - \rho\cdot\langle b, z^{k+1} - b^k\rangle + \sum_v \langle \nu_v^k, \sum_e a_{v,e}\cdot b_e\rangle \right). \end{aligned} \tag{54}$$

We then define an auxiliary function as

$$\begin{aligned} \theta_{k+1}(b) = &\sum_e f_e(b_e) + \langle \mu^k, z^{k+1} - b\rangle + \frac{\rho}{2}\cdot\|z^{k+1} - b\|^2 \\ &+ \sum_v \langle \nu_v^k, \sum_e a_{v,e}\cdot b_e\rangle. \end{aligned} \tag{55}$$

$b^{k+1} = \underset{b\geq 0}{\operatorname{argmin}}\left(\theta_{k+1}(b)\right)$ determines that we have $\langle \nabla_{b^{k+1}}\left[\theta_{k+1}(b^{k+1})\right], b - b^{k+1}\rangle \geq 0,\ \forall b \geq 0$. Then, since $\theta(b)$ is strongly convex with constant $\rho + M_f$, we have

$$\begin{aligned} \theta_{k+1}(b^*) &\geq \theta_{k+1}(b^{k+1}) + \frac{\rho + M_f}{2}\cdot\|b^* - b^{k+1}\|^2 \\ &\quad + \langle \nabla_{b^{k+1}}\left[\theta_{k+1}(b^{k+1})\right], b^* - b^{k+1}\rangle \\ &\geq \theta_{k+1}(b^{k+1}) + \frac{\rho + M_f}{2}\cdot\|b^* - b^{k+1}\|^2, \end{aligned}$$

which can be expanded as

$$\begin{aligned} &\sum_e \left[f_e(b_e^{k+1}) - f(b_e^*)\right] + \frac{\rho}{2}\cdot\|z^{k+1} - b^k + b^k - b^{k+1}\|^2 \\ &\leq \frac{\rho}{2}\cdot\|z^{k+1} - b^k + b^k - b^*\|^2 + \langle \mu^k, b^{k+1} - b^*\rangle \\ &\quad + \sum_v \left\langle \nu_v^k, \sum_e a_{v,e}\cdot(b_e^* - b_e^{k+1})\right\rangle - \frac{\rho + M_f}{2}\cdot\|b^* - b^{k+1}\|^2. \end{aligned}$$

And, we further have

$$\begin{aligned} &\sum_e \left[f_e(b_e^{k+1}) - f(b_e^*)\right] + \frac{\rho}{2}\cdot\|b^{k+1} - b^k\|^2 \\ &\leq \frac{\rho}{2}\cdot\|b^* - b^k\|^2 + \langle \mu^k, b^{k+1} - b^*\rangle - \frac{\rho + M_f}{2}\cdot\|b^* - b^{k+1}\|^2 \\ &\quad + \sum_v \left\langle \nu_v^k, \sum_e a_{v,e}\cdot(b_e^* - b_e^{k+1})\right\rangle + \rho\cdot\langle b^{k+1} - b^*, z^{k+1} - b^k\rangle. \end{aligned} \tag{56}$$

Similarly, we also have

$$\theta_{k+1}(b^{k+1}) \leq \theta_{k+1}(b^k) - \frac{\rho + M_f}{2}\cdot\|b^{k+1} - b^k\|^2, \tag{57}$$

$$\theta_k(b^k) \leq \theta_k(b^{k+1}) - \frac{\rho + M_f}{2}\cdot\|b^{k+1} - b^k\|^2. \tag{58}$$

Eqs. (57) and (58) can be expanded as

$$\begin{aligned} &\sum_e \left[f_e(b_e^{k+1}) - f(b_e^k)\right] + \sum_v \left\langle \nu_v^k, \sum_e a_{v,e}\cdot(b_e^{k+1} - b_e^k)\right\rangle \\ &\leq \langle \mu^k, b^{k+1} - b^k\rangle + \frac{\rho}{2}\cdot\|z^{k+1} - b^k\|^2 \\ &\quad - \frac{\rho}{2}\cdot\|z^{k+1} - b^{k+1}\|^2 - \frac{\rho + M_f}{2}\cdot\|b^{k+1} - b^k\|^2, \end{aligned} \tag{59}$$

$$\begin{aligned} &\sum_e \left[f_e(b_e^k) - f(b_e^{k+1})\right] + \sum_v \left\langle \nu_v^{k-1}, \sum_e a_{v,e}\cdot(b_e^k - b_e^{k+1})\right\rangle \\ &\leq \langle \mu^{k-1}, b^k - b^{k+1}\rangle + \frac{\rho}{2}\cdot\|z^k - b^{k+1}\|^2 \\ &\quad - \frac{\rho}{2}\cdot\|z^k - b^k\|^2 - \frac{\rho + M_f}{2}\cdot\|b^{k+1} - b^k\|^2. \end{aligned} \tag{60}$$

By summing up Eqs. (59) and (60), we can obtain

$$\begin{aligned} &\sum_v \left\langle \nu_v^k - \nu_v^{k-1}, \sum_e a_{v,e}\cdot(b_e^{k+1} - b_e^k)\right\rangle \\ &\leq \langle \mu^{k+1} - \mu^k, b^{k+1} - b^k\rangle - M_f\cdot\|b^{k+1} - b^k\|^2. \end{aligned} \tag{61}$$

Thirdly, we consider the Lagrangian function in Eq. (26). Since z^{k+1} stays in $\mathcal{Z}$ and $b^{k+1} \geq 0$, it is obvious that

$$\widetilde{\mathcal{L}}(z^*, b^*, \mu^*, \nu^*) \leq \widetilde{\mathcal{L}}(z^{k+1}, b^{k+1}, \mu^*, \nu^*), \tag{62}$$

which indicates that

$$\begin{aligned} &c\cdot(z^* - z^{k+1}) + \sum_e \left[f_e(b_e^*) - f_e(b_e^{k+1})\right] \\ &\leq \langle \mu^*, z^{k+1} - b^{k+1}\rangle + \sum_v \left\langle \nu_v^*, \sum_e a_{v,e}\cdot(b_e^{k+1} - b_e^*)\right\rangle. \end{aligned} \tag{63}$$

Fourthly, the update rule of ν_v^{k+1} is $\nu_v^{k+1} = \nu_v^k + \lambda\cdot\sum_e a_{v,e}\cdot b_e^{k+1}$ in Eq. (22), which can be combined with $\nu_v^* = \nu_v^* + \lambda\cdot\sum_e a_{v,e}\cdot b_e^*$ to get

$$\begin{aligned} &\left(\mathbf{1}^T\cdot\nu_v^{k+1} - \mathbf{1}^T\cdot\nu_v^*\right)^2 \\ &= \left[\mathbf{1}^T\cdot\nu_v^k - \mathbf{1}^T\cdot\nu_v^* + \mathbf{1}^T\cdot\lambda\cdot\sum_e a_{v,e}\cdot(b_e^{k+1} - b_e^*)\right]^2, \end{aligned} \tag{64}$$

where $\mathbf{1}^T \in \mathcal{R}^{|V^d|}$ is a unit row matrix. Thereafter,

$$\begin{aligned} \sum_v \left(\mathbf{1}^T\cdot\nu_v^{k+1} - \mathbf{1}^T\cdot\nu_v^*\right)^2 = &\sum_v \left[\mathbf{1}^T\cdot\nu_v^k - \mathbf{1}^T\cdot\nu_v^* \right. \\ &\left. + \mathbf{1}^T\cdot\lambda\cdot\sum_e a_{v,e}\cdot(b_e^{k+1} - b_e^*)\right]^2. \end{aligned} \tag{65}$$

Eq. (65) can be expanded to get

$$\begin{aligned} &\sum_v \left\langle \nu_v^k - \nu_v^*, \sum_e a_{v,e}\cdot\left(b_e^* - b_e^{k+1}\right)\right\rangle \\ =&\frac{1}{2\lambda}\cdot\left(\|\nu^k - \nu^*\|^2 - \|\nu^{k+1} - \nu^*\|^2\right) \\ &+ \frac{\lambda}{2}\cdot\sum_v \left(\sum_e a_{v,e}\cdot\mathbf{1}^T\cdot b_e^{k+1} - \sum_e a_{v,e}\cdot\mathbf{1}^T\cdot b_e^*\right)^2 \\ \leq&\frac{1}{2\lambda}\cdot\left(\|\nu^k - \nu^*\|^2 - \|\nu^{k+1} - \nu^*\|^2\right) + \frac{\lambda\cdot\delta}{2}\cdot\|b^{k+1} - b^*\|^2, \end{aligned} \tag{66}$$

where the last inequality is because δ is $|V^d|$ times the largest singular value of the adjacent matrix A.

Similarly, we have

$$\sum_v \left(\mathbf{1}^T \cdot \nu_v^{k+1} - \mathbf{1}^T \cdot \nu_v^k\right)^2 = \sum_v \Big[\mathbf{1}^T \cdot \nu_v^k - \mathbf{1}^T \cdot \nu_v^{k-1} + \lambda \cdot \mathbf{1}^T \cdot \sum_e a_{v,e} \cdot (b_e^{k+1} - b_e^k)\Big]^2. \tag{67}$$

Eq. (67) can be expanded to get

$$\begin{aligned}
&\sum_v \left\langle \nu_v^k - \nu_v^{k-1}, \sum_e a_{v,e} \cdot \left(b_e^k - b_e^{k+1}\right)\right\rangle \\
=&\frac{1}{2\lambda} \cdot \left(\|\nu^k - \nu^{k-1}\|^2 - \|\nu^{k+1} - \nu^k\|^2\right) \\
&+ \frac{\lambda}{2} \cdot \sum_v \left(\sum_e a_{v,e} \cdot \mathbf{1}^T \cdot b_e^{k+1} - \sum_e a_{v,e} \cdot \mathbf{1}^T \cdot b_e^k\right)^2 \\
\leq&\frac{1}{2\lambda} \cdot \left(\|\nu^k - \nu^{k-1}\|^2 - \|\nu^{k+1} - \nu^k\|^2\right) + \frac{\lambda \cdot \delta}{2} \cdot \|b^{k+1} - b^k\|^2.
\end{aligned} \tag{68}$$

Finally, by summing up Eqs. (53), (56), (61), (63), (66) and (68), we can get

$$\begin{aligned}
&\frac{\rho}{2}\cdot\|z^{k+1} - z^k\|^2 + \frac{\rho}{2} \cdot \|b^{k+1} - b^k\|^2 - \langle \mu^{k+1} - \mu^k, b^{k+1} - b^k\rangle \\
\leq&\frac{\rho}{2} \cdot \left(\|z^* - z^k\|^2 - \|z^* - z^{k+1}\|^2\right) \\
&+ \frac{1}{2\lambda} \cdot \left(\|\nu^k - \nu^*\|^2 - \|\nu^{k+1} - \nu^*\|^2\right) \\
&+ \frac{1}{2\lambda} \cdot \left(\|\nu^k - \nu^{k-1}\|^2 - \|\nu^{k+1} - \nu^k\|^2\right) \\
&+ \frac{1}{2} \cdot (M_f + \rho - \lambda \cdot \delta) \left(\|b^* - b^k\|^2 - \|b^* - b^{k+1}\|^2\right) \\
&- \frac{1}{2} \cdot (M_f - \lambda \cdot \delta) \cdot \|b^* - b^k\|^2 - \left(M_f - \frac{\lambda \cdot \delta}{2}\right) \|b^{k+1} - b^k\|^2 \\
&+ \langle \mu^* - \mu^k, z^{k+1} - b^{k+1}\rangle + \rho \cdot \langle b^{k+1} - b^*, z^{k+1} - b^k\rangle \\
&+ \rho \cdot \langle z^* - z^{k+1}, z^k - b^k\rangle.
\end{aligned} \tag{69}$$

Then, we proceed to handle the following terms in Eq. (69):

$$\begin{aligned}
&\langle \mu^* - \mu^k, z^{k+1} - b^{k+1}\rangle + \rho \cdot \langle b^{k+1} - b^*, z^{k+1} - b^k\rangle \\
&+ \rho \cdot \langle z^* - z^{k+1}, z^k - b^k\rangle.
\end{aligned} \tag{70}$$

Since the update rule of μ is $\mu^{k+1} = \mu^k + \rho \cdot (z^{k+1} - b^{k+1})$ in Eq. (23), the first term in Eq. (70) is equivalent to

$$\begin{aligned}
&\langle \mu^* - \mu^k, z^{k+1} - b^{k+1}\rangle = \frac{1}{\rho} \cdot \langle \mu^* - \mu^k, \mu^{k+1} - \mu^k\rangle \\
&= -\frac{1}{2\rho} \cdot \left(\|\mu^* - \mu^{k+1}\|^2 - \|\mu^k - \mu^*\|^2\right) + \frac{1}{2\rho} \cdot \|\mu^{k+1} - \mu^k\|^2.
\end{aligned} \tag{71}$$

The second term in Eq. (70) is

$$\begin{aligned}
\rho \cdot \langle b^{k+1} - b^*, z^{k+1} - b^k\rangle =& \rho \cdot \langle b^{k+1} - b^*, z^{k+1} - b^{k+1}\rangle \\
&+ \rho \cdot \langle b^{k+1} - b^*, b^{k+1} - b^k\rangle.
\end{aligned} \tag{72}$$

Similarly, the last term in Eq. (70) can be transformed into

$$\begin{aligned}
&\rho \cdot \langle z^* - z^{k+1}, z^k - b^k\rangle = \rho \cdot \langle z^* - z^{k+1}, z^k - z^{k+1}\rangle \\
&+ \langle z^* - z^{k+1}, \mu^{k+1} - \mu^k\rangle + \rho \cdot \langle z^* - z^{k+1}, b^{k+1} - b^k\rangle.
\end{aligned} \tag{73}$$

As $z^* = b^*$, we consider the equations' right hand sides, add the first term in Eq. (72) to the second one in Eq. (73) as

$$\begin{aligned}
&\rho \cdot \langle b^{k+1} - b^*, z^{k+1} - b^{k+1}\rangle + \langle z^* - z^{k+1}, \mu^{k+1} - \mu^k\rangle \\
&= -\frac{1}{\rho} \cdot \|\mu^{k+1} - \mu^k\|^2,
\end{aligned} \tag{74}$$

and sum up the second term in Eq. (72) with the third term in Eq. (73) as

$$\begin{aligned}
&\rho \cdot \langle b^{k+1} - b^*, b^{k+1} - b^k\rangle + \rho \cdot \langle z^* - z^{k+1}, b^{k+1} - b^k\rangle \\
&= -\langle \mu^{k+1} - \mu^k, b^{k+1} - b^k\rangle.
\end{aligned} \tag{75}$$

Hence, by summing up Eqs. (71)-(73) and substituting the corresponding terms with Eqs. (74) and (75), we have

$$\begin{aligned}
&\langle \mu^* - \mu^k, z^{k+1} - b^{k+1}\rangle + \rho \cdot \langle b^{k+1} - b^*, z^{k+1} - b^k\rangle \\
&+ \rho \cdot \langle z^* - z^{k+1}, z^k - b^k\rangle \\
=&\frac{\rho}{2} \cdot \left(\|z^* - z^{k+1}\|^2 - \|z^* - z^k\|^2\right) + \frac{\rho}{2} \cdot \|z^k - z^{k+1}\|^2 \\
&+ \frac{\rho}{2} \cdot \|b^{k+1} - b^k\|^2 + \frac{1}{2\rho} \cdot \left(\|\mu^* - \mu^k\|^2 - \|\mu^* - \mu^{k+1}\|^2\right) \\
&- \frac{1}{2} \cdot \|\frac{1}{\sqrt{\rho}} \cdot (\mu^{k+1} - \mu^k) + \sqrt{\rho} \cdot (b^{k+1} - b^k)\|^2.
\end{aligned} \tag{76}$$

By substituting Eq. (76) into Eq. (69), we obtain

$$\begin{aligned}
&\frac{1}{2\rho} \cdot \|(\mu^{k+1} - \mu^k)\|^2 + (M_f + \frac{\rho}{2} - \frac{\lambda \cdot \delta}{2}) \cdot \|b^{k+1} - b^k\|^2 \\
\leq&\frac{1}{2\lambda} \cdot \left(\|\nu^k - \nu^*\|^2 - \|\nu^{k+1} - \nu^*\|^2\right) \\
&- \frac{1}{2}(M_f - \lambda \cdot \delta) \cdot \|b^* - b^k\|^2 \\
&+ \frac{1}{2\lambda} \cdot \left(\|\nu^k - \nu^{k-1}\|^2 - \|\nu^{k+1} - \nu^k\|^2\right) \\
&+ \frac{1}{2} \left(M_f + \rho - \lambda \cdot \delta\right) \left(\|b^* - b^k\|^2 - \|b^* - b^{k+1}\|^2\right) \\
&+ \frac{1}{2\rho} \cdot \left(\|\mu^* - \mu^k\|^2 - \|\mu^* - \mu^{k+1}\|^2\right).
\end{aligned} \tag{77}$$

If we choose the positive parameters ρ and λ such that

$$M_f - \lambda \cdot \delta \geq 0, \tag{78}$$

the terms $-\frac{1}{2}(M_f - \lambda \cdot \delta) \cdot \|b^* - b^k\|^2 \leq 0$ in the right hand side of Eq. (77) can be removed to get

$$\begin{aligned}
&\frac{1}{2\rho}\cdot\|(\mu^{k+1} - \mu^k)\|^2 + (M_f + \frac{\rho}{2} - \frac{\lambda \cdot \delta}{2}) \cdot \|b^{k+1} - b^k\|^2 \\
\leq&\frac{1}{2\lambda} \cdot \left(\|\nu^k - \nu^*\|^2 - \|\nu^{k+1} - \nu^*\|^2\right) \\
&+ \frac{1}{2\lambda} \cdot \left(\|\nu^k - \nu^{k-1}\|^2 - \|\nu^{k+1} - \nu^k\|^2\right) \\
&+ \frac{1}{2} \left(M_f + \rho - \lambda \cdot \delta\right) \left(\|b^* - b^k\|^2 - \|b^* - b^{k+1}\|^2\right) \\
&+ \frac{1}{2\rho} \cdot \left(\|\mu^* - \mu^k\|^2 - \|\mu^* - \mu^{k+1}\|^2\right),
\end{aligned} \tag{79}$$

which completes the proof of *Lemma* 2.

ACKNOWLEDGMENTS

This work was supported in part by the NSFC Project 61371117, the SPR Program of the CAS (XDA06011202), the Key Research Project of the CAS (QYZDY-SSW-JSC003), and the NGBWMCN Key Project under Grant No. 2017ZX03001019-004.

REFERENCES

[1] 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-datacenter networks," *IEEE Netw.*, vol. 29, pp. 36–42, Sept./Oct. 2015.

[2] How effective is your data center's disaster recovery plan? [Online]. Available: http://www.lifelinedatacenters.com/data-center/effective-disaster-recovery-plan/

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

[4] Hurricane Sandy. [Online]. Available: https://en.wikipedia.org/wiki/Hurricane_Sandy

[5] P. Lu, Q. Ling, and Z. Zhu, "Maximizing utility of time-constrained emergency backup in inter-datacenter networks," *IEEE Commun. Lett.*, vol. 20, pp. 890–893, May 2016.

[6] L. Ma, W. Su, B. Wu, T. Taleb, X. Jiang, and N. Shiratori, "E-time early warning data backup in disaster-aware optical interconnected data center networks," *J. Opt. Commun. Netw.*, vol. 9, pp. 536–545, Jun. 2017.
[7] 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, pp. 3005–3015, Jul. 2015.
[8] N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez, "Inter-datacenter bulk transfers with netstitcher," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, pp. 74–85, Aug. 2011.
[9] W. Lu and Z. Zhu, "Malleable reservation based bulk-data transfer to recycle spectrum fragments in elastic optical networks," *J. Lightw. Technol.*, vol. 33, pp. 2078–2086, May 2015.
[10] P. Lu and Z. Zhu, "Data-oriented task scheduling in fixed- and flexible-grid multilayer inter-DC optical networks: A comparison study," *J. Lightw. Technol., in Press*, 2017.
[11] A. Bianco, L. Giraudo, and D. Hay, "Optimal resource allocation for disaster recovery," in *Proc. of GLOBECOM 2010*, pp. 1–5, Mar. 2010.
[12] X. Xie, Q. Ling, P. Lu, and Z. Zhu, "ADMM-based distributed algorithm for emergency backup in time-variant inter-DC networks," in *Proc. of ICC 2017*, pp. 1–6, May 2017.
[13] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, pp. 1–122, Jan. 2011.
[14] W. Lu, Z. Zhu, and B. Mukherjee, "Optimizing deadline-driven bulk-data transfer to revitalize spectrum fragments in EONs," *J. Opt. Commun. Netw.*, vol. 7, pp. B173–B183, Dec. 2015.
[15] Y. Wang, S. Su, A. Liu, and Z. Zhang, "Multiple bulk data transfers scheduling among datacenters," *Comput. Netw.*, vol. 68, pp. 123–137, Aug. 2014.
[16] K. Wu, P. Lu, and Z. Zhu, "Distributed online scheduling and routing of multicast-oriented tasks for profit-driven cloud computing," *IEEE Commun. Lett.*, vol. 20, pp. 684–687, Apr. 2016.
[17] S. Ferdousi, M. Tornatore, F. Habib, and B. Mukherjee, "Rapid data evacuation for large-scale disasters in optical cloud networks," *J. Opt. Commun. Netw.*, vol. 7, pp. B163–B172, Dec. 2015.
[18] M. Liu, H. Gong, Y. Wen, G. Chen, and J. Cao, "The last minute: Efficient data evacuation strategy for sensor networks in post-disaster applications," in *Proc. of INFOCOM 2011*, pp. 291–295, Apr. 2011.
[19] B. Klinz and G. Woeginger, "Minimum-cost dynamic flows: The series-parallel case," *Networks*, vol. 43, pp. 153–162, May 2004.
[20] L. Ford and D. Fulkerson, "Constructing maximal dynamic flows from static flows," *Oper. Res.*, vol. 6, pp. 419–433, Jun. 1958.
[21] L. Fleischer and M. Skutella, "Quickest flows over time," *SIAM J. Comput.*, vol. 36, pp. 1600–1630, Feb. 2007.
[22] J. Eckstein and D. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Program.*, vol. 55, pp. 293–318, Apr. 1992.
[23] B. He, L. Liao, D. Han, and H. Yang, "A new inexact alternating directions method for monotone variational inequalities," *Math. Program.*, vol. 92, pp. 103–118, Mar. 2002.
[24] Q. Ling and A. Ribeiro, "Decentralized linearized alternating direction method of multipliers," in *Proc. of ICASSP 2014*, pp. 5447–5451, May 2014.
[25] Q. Ling, W. Shi, G. Wu, and A. Ribeiro, "DLM: Decentralized linearized alternating direction method of multipliers," *IEEE Trans. Signal Process.*, vol. 63, pp. 4051–4064, May 2015.
[26] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, "Decentralized quadratically approximated alternating direction method of multipliers," in *Proc. of GlobalSIP 2015*, pp. 795–799, Dec. 2015.
[27] T. Lin, S. Ma, and S. Zhang, "An extragradient-based alternating direction method for convex minimization," *Foun. Comput. Math.*, vol. 17, pp. 1–25, Feb. 2017.
[28] F. Habib, M. Tornatore, F. Dikbiyik, and B. Mukherjee, "Disaster survivability in optical communication networks," *Comput. Commun.*, vol. 36, pp. 630–644, Mar. 2013.
[29] B. Mukherjee, F. Habib, and F. Dikbiyik, "Network adaptability from disaster disruptions and cascading failures," *IEEE Commun. Mag.*, vol. 52, pp. 230–238, May 2014.
[30] P. Agarwal, A. Efrat, S. Ganjugunte, D. Hay, S. Sankararaman, and G. Zussman, "Network vulnerability to single, multiple, and probabilistic physical attacks," in *Proc. of MILCOM 2010*, pp. 1824–1829, Nov. 2010.
[31] W. Lu and Z. Zhu, "Dynamic service provisioning of advance reservation requests in elastic optical networks," *J. Lightw. Technol.*, vol. 31, pp. 1621–1627, May 2013.
[32] H. Xu, C. Feng, and B. Li, "Temperature aware workload management in geo-distributed data centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, pp. 1743–1753, Jun. 2015.
[33] P. Lu, Q. Sun, K. Wu, and Z. Zhu, "Distributed online hybrid cloud management for profit-driven multimedia cloud computing," *IEEE Trans. Multimedia*, vol. 17, pp. 1297–1308, Aug. 2015.
[34] W. Lu, Z. Zhu, and B. Mukherjee, "On hybrid IR and AR service provisioning in elastic optical networks," *J. Lightw. Technol.*, vol. 33, pp. 4659–4669, Nov. 2015.
[35] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, Apr. 2004.
[36] H. Xu and B. Li, "Joint request mapping and response routing for geo-distributed cloud services," in *Proc. of INFOCOM 2013*, pp. 854–862, Apr. 2013.