DISTRIBUTED NETWORK RESOURCE ALLOCATION WITH INTEGER CONSTRAINTS

Yujiao Cheng, Houfeng Huang, Gang Wu, Qing Ling

Department of Automation, University of Science and Technology of China, Hefei, China

ABSTRACT

This paper considers the resource allocation problem defined over a hybrid data center and edge server network, where the allocations are subject to integer constraints, taking the granularity of network resources and service requests into account. We develop two efficient heuristic algorithms to solve this nonconvex program, both based on the alternating direction method of multipliers (ADMM) and a distributed integral projection scheme. The first algorithm ignores the integer constraints and solves the relaxed convex program, while the second algorithm includes the integer constraints in the optimization process. We develop a distributed integral projection scheme, which approximately projects the resulting resource allocation strategies onto the feasible set. Numerical experiments validate the effectiveness of the proposed algorithms.

Index Terms— Cloud computing, edge computing, network resource allocation, integer programming

1. INTRODUCTION

This paper considers the resource allocation problem defined over a hybrid data center and edge server network. The hybrid network, as depicted in Fig. 1, is composed of one virtual cloud center and multiple edge servers. Upon receiving service requests (for example, computation or storage tasks) from end users, the edge servers either process the requests by themselves or assign them to neighboring edge servers and the cloud center. We say that two edge servers are neighbors if they are connected with a low-latency communication link. The communication links between the edge servers and the cloud center generally have high communication latency. The role of the edge servers is enabling quick response to the requests in spite of their limited computation and storage powers. The cloud server can utilize almost infinite computation and storage resources, but it often incurs considerable communication cost. Therefore, the hybrid network infrastructure combines the advantages of cloud computing [1] and edge computing [2, 3], and provides the end users on-site, elastic and autonomous services.

The key of implementing such a hybrid network is efficient network resource allocation, which allocates network resources (for example, computation and storage resources of the edge servers and the cloud center, as well as bandwidth resources of the communication links) to handle the service requests, for the purpose of maximizing the utility of the whole network. Most of the existing network resource allocation models and algorithms only consider the collaboration either among distributed nodes [4, 5, 6], or between one centralized master node and multiple distributed slave nodes [7]. In the recent work [8], the authors propose a collaborative network resource allocation model, which allows joint optimization of the hybrid cloud center and edge server network.

Qing Ling is supported in part by NSF China grant 61573331 and NSF Anhui grant 1608085QF130.

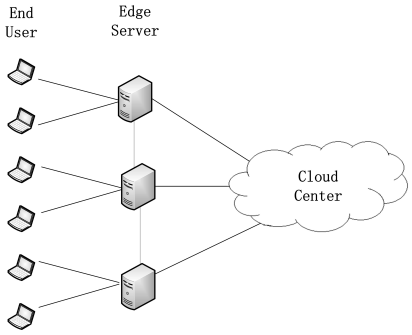

Fig. 1. Illustrative infrastructure of a hybrid cloud center and edge server network.

However, the optimization model proposed in [8] does not take the granularity of network resources and service requests into account. Therein, a request can be divided into arbitrarily small pieces, and assigned to different edge servers and the cloud center. This is not favorable for two reasons. First, a resource in the network often has been divided into blocks; hence, it is inefficient to handle those artificially generated small pieces of requests since they still occupy whole blocks. Second, an end user may also prefer that a certain batch of data is processed or stored in one, not multiple places.

This fact motivates us to introduce integer constraints into the collaborative network resource allocation model, where the assignments of requests must be integers. The integer constraints, however, bring challenges to design efficient distributed algorithms. Classical integer programming approaches, such as the branch and bound method, is computationally costly and does not fit for distributed implementation in a network environment [9].

This paper develops efficient heuristic algorithms to solve this nonconvex program, both based on the alternating direction method of multipliers (ADMM) and a distributed integral projection scheme. ADMM is a powerful operator splitting algorithm, which has found successful application in distributed optimization problems [8, 10, 11]. Though usually used to solve convex programs, it is also applicable to nonconvex programs, such as those with integer constraints [12]. The proposed distributed integral projection scheme projects a real-valued network resource allocation strategy to an integral one and guarantees its feasibility. Our contributions are as three-fold.

(i) We propose an integer-constrained network resource allocation model for the hybrid cloud center and edge server network, where the granularity of service requests and assignments is taken into consideration.

(ii) We develop two distributed network resource allocation algorithms based on ADMM. The first algorithm ignores the integer constraints and solves the relaxed convex program, while the second algorithm includes the integer constraints in the optimization process.

(iii) We devise a distributed integral projection scheme, which projects a real-valued network resource allocation strategy onto the feasible set with integral constraints.

The remainder of the paper is organized as follows. Section 2 puts forward the collaborative resource allocation model with integer constraints. Section 3 proposes two distributed resource allocation algorithms. Numerical experiments are provided in Section 4, demonstrating the effectiveness of the proposed algorithms.

2. NETWORK RESOURCE ALLOCATION MODEL

Consider the hybrid cloud edge server and cloud center network that responses to the service requests from end users, as illustrated in Fig. 1. To simplify the discussion, we assume that there is only one kind of service request (we consider computation in this paper) and the amount of every service request can be quantified as an integer (for example, an integral number of MB of data to process). The granularity of the service request is a system parameter, which is determined by the size of network resource blocks and the requirements of end users.

Given a hybrid network with one cloud center and N edge servers, we call the cloud center as node 0 and the edge servers are labelled from node 1 to node N. The network is hence represented as a bidirectionally connected communication graph. We use $\mathcal{N}_i$ to denote the set of neighbors for edge server i. The cloud center is connected to every edge server, but is not treated as a neighbor. Denote l_{ij} as the latency of link (i,j). If two edge servers i and j are neighbors (namely, $i \in \mathcal{N}_j$ and $j \in \mathcal{N}_i$), the latency l_{ij} is assumed to be small. The latency between the cloud center (namely, node 0) and edge server i, denoted by l_{0i}, is often large. The amount of service requests collected by edge server i from the end users is $s_i \in \mathcal{Z}_+$, where $\mathcal{Z}_+$ denotes the set of nonnegative integers. Edge server i can assign these service requests to node j, $j \in \mathcal{N}_i \cup i \cup 0$, and the assignment is denoted by $x_{ij} \in \mathcal{Z}_+$. To satisfy the service requests, for every edge server i it must hold $s_i = \sum_{j\in\mathcal{N}_i\cup i\cup 0} x_{ij}$.

For a single edge server i, the goal is to minimize two parts of cost: the first is the latency cost $f_i(\cdot)$ when it assigns requests to its neighboring edge servers and the cloud center; the second is the computation cost $g_i(\cdot)$ to handle the assignments received from itself and its neighbors. For the cloud center, its computation cost is defined by $g_0(\cdot)$ for handling the assignments received from the edge servers. Defining $\mathbf{x}_i = [x_{i0}; x_{i1}; \cdots; x_{iN}] \in \mathcal{Z}_+^{N+1}$ as the assignment vector of edge server i, a typical choice of $f_i(\cdot)$ is

$$f_i(\mathbf{x}_i) = q s_i \Big(\sum_{j\in\mathcal{N}_i\cup 0} \frac{x_{ij} l_{ij}}{s_i} \Big)^2,$$

which is also used in [10]. Therein, q is a weight factor that tunes the relative importance of the latency and computation costs; $\sum_{j\in\mathcal{N}_i\cup 0} x_{ij} l_{ij}/s_i$ is the average latency for edge server i. The computation costs $g_i(\cdot)$ and $g_0(\cdot)$ are properly chosen to be strictly increasing functions, with $g_i(\cdot)$ being much larger than $g_0(\cdot)$ when their arguments are the same.

Thus, the resource allocation problem is in the form of

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} f_i(\mathbf{x}_i) + \sum_{i=1}^{N} g_i\Big(\sum_{j\in\mathcal{N}_i\cup i} x_{ji} \Big) + g_0\Big(\sum_{j=1}^{N} x_{j0} \Big), \quad (1) \\ s.t. \quad & s_i = \sum_{j\in\mathcal{N}_i\cup i\cup 0} x_{ij}, \quad i = 1, \cdots, N, \\ & x_{ij} \in \mathcal{Z}_+, \quad i = 1, \cdots, N, \quad j \in \mathcal{N}_i \cup i \cup 0. \end{aligned}$$

3. ALGORITHM DEVELOPMENT

The main challenges of solving (1) are two-fold. First, the computation must be distributed to the nodes for the sake of being real-time and robust. Second, the integer constraints make the problem nonconvex such that finding the global optimal solution becomes intractable. In this section, we propose two distributed algorithms, both based on the alternating direction method of multipliers (ADMM). The first algorithm applies the idea of convex relaxation by dropping off the integer constraints during the optimization process. We call this algorithm as ADMM-CR. The second algorithm, termed as ADMM-NC, uses ADMM to directly solve the nonconvex program. In both algorithms, the solved real-valued network resource allocation strategy is projected to an integral one using a distributed integral projection scheme after termination.

3.1. Algorithm 1: ADMM-CR

Throwing away the integer constraints and introducing auxiliary variables $y_{ji} = x_{ij}, \forall i, \forall j$, (1) is relaxed to

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} f_i(\mathbf{x}_i) + \sum_{i=1}^{N} g_i\Big(\sum_{j\in\mathcal{N}_i\cup i} y_{ij} \Big) + g_0\Big(\sum_{j=1}^{N} y_{0j} \Big), \quad (2) \\ s.t. \quad & s_i = \sum_{j\in\mathcal{N}_i\cup i\cup 0} x_{ij}, \quad i = 1, \cdots, N, \\ & x_{ij} = y_{ji}, \quad i = 1, \cdots, N, \quad j \in \mathcal{N}_i \cup i \cup 0, \\ & x_{ij} \geq 0, \quad i = 1, \cdots, N, \quad j \in \mathcal{N}_i \cup i \cup 0. \end{aligned}$$

Let a_i and c_{ij} be the Lagrange multipliers of the equality constraints $s_i = \sum_{j\in\mathcal{N}_i\cup i\cup 0} x_{ij}$ and $x_{ji} = y_{ij}$, respectively. The augmented Lagrangian function of (2) is

$$\begin{aligned} L_\rho = & \sum_{i=1}^{N} f_i(\mathbf{x}_i) + \sum_{i=1}^{N} g_i\Big(\sum_{j\in\mathcal{N}_i\cup i} y_{ij} \Big) + g_0\Big(\sum_{j=1}^{N} y_{0j} \Big) \quad (3) \\ & + \sum_{i=1}^{N} a_i \Big(\sum_{j\in\mathcal{N}_i\cup i\cup 0} x_{ij} - s_i \Big) + \frac{\rho}{2} \sum_{i=1}^{N} \Big(\sum_{j\in\mathcal{N}_i\cup i\cup 0} x_{ij} - s_i \Big)^2 \\ & + \sum_{i=1}^{N} \sum_{j\in\mathcal{N}_i\cup i\cup 0} c_{ji}(x_{ij} - y_{ji}) + \frac{\rho}{2} \sum_{i=1}^{N} \sum_{j\in\mathcal{N}_i\cup i\cup 0} (x_{ij} - y_{ji})^2, \end{aligned}$$

subject to $x_{ij} \geq 0, i = 1, \cdots, N, j \in \mathcal{N}_i \cup i \cup 0$. Here, $\rho > 0$ is a positive penalty parameter.

At time k, ADMM first fixes the primal variables y_{ij} and the dual variables a_i and c_{ij} to minimize the augmented Lagrangian function with respect to x_{ij}, then fixes the primal variables x_{ij} and the dual variables a_i and c_{ij} to minimize the augmented Lagrangian function with respect to y_{ij}, and finally updates the dual variables a_i and c_{ij} with dual gradient ascent. Below we directly give the updates; readers are referred to [8] for detailed derivation.

For edge server i, its updates of x_{ij} are given by

$$\begin{aligned} x_{ij}^{k+1} := \arg \min_{\{x_{ij}\geq 0\}} \Big\{ & f_i(\mathbf{x}_i) + \frac{\rho}{2} \Big(\sum_{j\in\mathcal{N}_i\cup i\cup 0} x_{ij} - s_i + \frac{a_i^k}{\rho} \Big)^2 \\ & + \frac{\rho}{2} \sum_{j\in\mathcal{N}_i\cup i\cup 0} \Big(x_{ij} - y_{ji}^k + \frac{c_{ji}^k}{\rho} \Big)^2 \Big\}. \quad (4) \end{aligned}$$

The updates of y_{ij} are

$$y_{ij}^{k+1} := \arg \min_{\{y_{ij}\}} \Big\{ g_i\Big(\sum_{j\in\mathcal{N}_i\cup i} y_{ij} \Big)$$

$$+\frac{\rho}{2}\sum_{j\in\mathcal{N}_i\cup i}(x_{ji}^{k+1}-y_{ij}+\frac{c_{ij}^k}{\rho})^2\Big\}. \quad (5)$$

The cloud center (namely, node 0) also needs to update y_{0j} for all $j=1,\cdots,N$, by

$$y_{0j}^{k+1}:=\arg\min_{\{y_{0j}\}}\Big\{g_0(\sum_{j=1}^N y_{0j}) + \frac{\rho}{2}\sum_{j=1}^N(x_{j0}^{k+1}-y_{0j}+\frac{c_{0j}^k}{\rho})^2\Big\}. \quad (6)$$

The updates of dual variables a_i and c_{ij} are

$$a_i^{k+1}:=a_i^k+\rho(\sum_{j\in\mathcal{N}_i\cup i\cup 0}x_{ij}^{k+1}-s_i), \quad (7)$$

$$c_{ij}^{k+1}:=c_{ij}^k+\rho(x_{ji}^{k+1}-y_{ij}^{k+1}). \quad (8)$$

The above iterations are fully distributed to the edge servers and the could center.

Table 1: ADMM-CR Run at Cloud Center

0. Initialize $y_{0j}, c_{0j}=0, j=1,\cdots,N$
1. **for** $k=0,1,\cdots$ **do**
2. From all edge server j, collect $x_{j0}^{k+1}, j=1,\cdots,N$
3. Update $y_{0j}, j=1,\cdots,N$ by (6)
4. Update $c_{0j}, j=1,\cdots,N$ by (8)
5. **end for**

Table 2: ADMM-CR Run at Edge Server i

0. Initialize $a_i=0, x_{ij}, y_{ij}, c_{ij}=0, j\in\mathcal{N}_i\cup i\cup 0$
1. **for** $k=0,1,\cdots$ **do**
2. From every node $j\in\mathcal{N}_i\cup 0$, collect y_{ji}^k and c_{ji}^k
3. Update $x_{ij}^{k+1}, j\in\mathcal{N}_i\cup i\cup 0$ by (4)
4. From every neighboring edge server $j\in\mathcal{N}_i$, collect x_{ji}^{k+1}
5. Update $y_{ij}, j\in\mathcal{N}_i\cup i$ by (5)
6. Update a_i by (7)
7. Update $c_{ij}, j\in\mathcal{N}_i\cup i\cup 0$ by (8)
8. **end for**
9. Calculate $\{r_{ij}\}$ from $\{x_{ij}\}$ by distributed integral projection

Distributed Integral Projection. After terminating the ADMM iterations, we have a real-valued resource allocation strategy $\{x_{ij}\}, i=1,\cdots,N, j\in\mathcal{N}_i\cup i\cup 0$. Here we propose a distributed integral projection scheme to approximately project the real-valued strategy onto the feasible set of (1) by considering the integer constraints. First, round $\{x_{ij}\}$ down to their nearest integers to get $\{r_{ij}\}, i=1,\cdots N, j\in\mathcal{N}_i\cup i\cup 0$. Second, for every edge server i, calculate the value of $d_i=s_i-\sum_{j\in\mathcal{N}_i\cup i\cup 0}r_{ij}$. If d_i is zero, then $\{r_{ij}\}$ is feasible. Otherwise, calculate $m_i=\mathrm{mod}(d_i,|\mathcal{N}_i|+2)$, add $(d_i-m_i)/(|\mathcal{N}_i|+2)$ to every neighboring edge server, the cloud center and itself, and then add 1 to those r_{ij} with the largest m_i values of $|r_{ij}-x_{ij}|$. Eventually we get an integral resource allocation strategy (meanwhile, the request assignment strategy) $\{r_{ij}\}$, which is a satisfactory solution to (1). This integral projection scheme is also distributed to the edge servers.

The ADMM-CR algorithms run at the cloud center and every edge server i are outlined in Table 1 and Table 2, respectively. The cloud center is in charge of updating y_{0j} and c_{0j} for $j=1,\cdots,N$, while edge server i calculates a_i, x_{ij}, y_{ij} and c_{ij}, $j\in\mathcal{N}_i\cup i\cup 0$. Information exchange occurs between the cloud center and the edge servers, as well as between the neighboring edge servers.

3.2. Algorithm 2: ADMM-NC

The nonconvex ADMM approach works on (1) other than its convex relaxation. Introducing auxiliary variables $z_{ji}=y_{ji}=x_{ij},\forall i,\forall j$, (1) is equivalent to

$$\begin{aligned}\min\quad & \sum_{i=1}^N f_i(\mathbf{x}_i)+\sum_{i=1}^N g_i(\sum_{j\in\mathcal{N}_i\cup i}y_{ij})+g_0(\sum_{j=1}^N y_{0j}), \quad (9)\\ s.t.\quad & s_i=\sum_{j\in\mathcal{N}_i\cup i\cup 0}x_{ij},\quad i=1,\cdots,N,\\ & x_{ij}=y_{ji},\quad i=1,\cdots,N,\quad j\in\mathcal{N}_i\cup i\cup 0,\\ & z_{ji}=y_{ji},\quad i=1,\cdots,N,\quad j\in\mathcal{N}_i\cup i\cup 0,\\ & z_{ji}\in\mathcal{Z}_+,\quad i=1,\cdots,N,\quad j\in\mathcal{N}_i\cup i\cup 0.\end{aligned}$$

Let a_i, c_{ij} and d_{ji} be the Lagrange multipliers of the equality constraints $s_i=\sum_{j\in\mathcal{N}_i\cup i\cup 0}x_{ij}$, $x_{ji}=y_{ij}$ and $z_{ji}=y_{ji}$, respectively. The augmented Lagrangian function of (9) is

$$\begin{aligned}L_\rho=&\sum_{i=1}^N f_i(\mathbf{x}_i)+\sum_{i=1}^N g_i(\sum_{j\in\mathcal{N}_i\cup i}y_{ij})+g_0(\sum_{j=1}^N y_{0j}) \quad (10)\\ &+\sum_{i=1}^N a_i(\sum_{j\in\mathcal{N}_i\cup i\cup 0}x_{ij}-s_i)+\frac{\rho}{2}\sum_{i=1}^N(\sum_{j\in\mathcal{N}_i\cup i\cup 0}x_{ij}-s_i)^2\\ &+\sum_{i=1}^N\sum_{j\in\mathcal{N}_i\cup i\cup 0}c_{ji}(x_{ij}-y_{ji})+\frac{\rho}{2}\sum_{i=1}^N\sum_{j\in\mathcal{N}_i\cup i\cup 0}(x_{ij}-y_{ji})^2,\\ &+\sum_{i=1}^N\sum_{j\in\mathcal{N}_i\cup i\cup 0}d_{ji}(z_{ji}-y_{ji})+\frac{\rho}{2}\sum_{i=1}^N\sum_{j\in\mathcal{N}_i\cup i\cup 0}(z_{ji}-y_{ji})^2,\end{aligned}$$

subject to $z_{ji}\in\mathcal{Z}_+, i=1,\cdots,N, j\in\mathcal{N}_i\cup i\cup 0$. Again, $\rho>0$ is a positive penalty parameter.

ADMM-NC directly works on this augmented Lagrangian, first minimizing with respect to x_{ij} and z_{ij}, then minimizing with respect to y_{ij}, and finally updating a_i, c_{ij} and d_{ij} (see [11] and [12] for reference). For edge server i, its updates of x_{ij} are given by

$$x_{ij}^{k+1}=\arg\min_{\{x_{ij}\}}\Big\{f_i(\mathbf{x}_i)+\frac{\rho}{2}\sum_{j\in\mathcal{N}_i\cup i\cup 0}(x_{ij}-s_i+\frac{a_i^k}{\rho})^2 + \frac{\rho}{2}\sum_{j\in\mathcal{N}_i\cup i\cup 0}(x_{ij}-y_{ji}^k+\frac{c_{ji}^k}{\rho})^2\Big\}, \quad (11)$$

which is similar to (4) except that the nonnegativity constraints are not necessary. The updates of z_{ij} are

$$z_{ij}^{k+1}:=\mathcal{P}_{\mathcal{Z}_+}(y_{ij}^k+\frac{d_{ij}^k}{\rho}), \quad (12)$$

where $\mathcal{P}_{\mathcal{Z}_+}(\cdot)$ denotes projection onto the nonnegative integral set $\mathcal{Z}_+$. Notice that this projection is different to the distributed integral projection step introduced above. The latter projects a resource allocation strategy $\{x_{ij}\}$ onto the feasible set of (1), while the former only projects individual z_{ij} onto the nonnegative integral set $\mathcal{Z}_+$.

The updates of y_{ij} are

$$y_{ij}^{k+1}:=\arg\min_{\{y_{ij}\}}\Big\{g_i(\sum_{j\in\mathcal{N}_i\cup i}y_{ij})+\frac{\rho}{2}\sum_{j\in\mathcal{N}_i\cup i}(x_{ji}^{k+1}-y_{ij}+\frac{c_{ij}^k}{\rho})^2 + \frac{\rho}{2}\sum_{j\in\mathcal{N}_i\cup i}(y_{ij}-z_{ij}^{k+1}+\frac{d_{ij}^k}{\rho})^2\Big\}. \quad (13)$$

The updates of y_{0j} are

$$y_{0j}^{k+1} := \arg\min_{\{y_{0j}\}} \Big\{ g_0(\sum_{j=1}^{N} y_{0j}) + \frac{\rho}{2} \sum_{j=1}^{N} (x_{j0}^{k+1} - y_{0j} + \frac{c_{0j}^k}{\rho})^2 + \frac{\rho}{2} \sum_{j=1}^{N} (y_{0j} - z_{0j}^{k+1} + \frac{d_{0j}^k}{\rho})^2 \Big\}. \quad (14)$$

The updates of dual variables a_i, c_{ij} and d_{ij} are

$$a_i^{k+1} := a_i^k + \rho(\sum_{j \in \mathcal{N}_i \cup i \cup 0} x_{ij}^{k+1} - s_i), \quad (15)$$

$$c_{ij}^{k+1} := c_{ij}^k + \rho(x_{ji}^{k+1} - y_{ij}^{k+1}), \quad (16)$$

$$d_{ij}^{k+1} := d_{ij}^k + \rho(y_{ij}^{k+1} - z_{ij}^{k+1}). \quad (17)$$

The ADMM-NC algorithms run at the cloud center and every edge server i are outlined in Table 3 and Table 4, respectively. The cloud center is in charge of updating y_{0j}, z_{0j}, c_{0j} and d_{0j} for $j = 1, \cdots, N$, while edge server i calculates $a_i, x_{ij}, y_{ij}\ z_{ij}, c_{ij}$ and d_{ij}, $j \in \mathcal{N}_i \cup i \cup 0$. The final request assignment strategy of edge server i is given by r_{ij}, for all $j \in \mathcal{N}_i \cup i \cup 0$, from the distributed integral projection step.

Table 3: ADMM-NC Run at Cloud Center

0. Initialize $y_{0j}, z_{0j}, c_{0j}, d_{0j} = 0, j = 1, \cdots, N$
1. **for** $k = 0, 1, \cdots$ **do**
2. Update $\{z_{0j}\}$ by (12)
3. From all edge server j, collect $x_{j0}^{k+1}, j = 1, \cdots, N$
4. Update $y_{0j}, j = 1, \cdots, N$ by (14)
5. Update $c_{0j}, j = 1, \cdots, N$ by (16)
6. Update $d_{0j}, j = 1, \cdots, N$ by (17)
7. **end for**

Table 4: ADMM-NC Run at Edge Server i

0. Initialize $a_i = 0, x_{ij}, y_{ij}, z_{ij}, c_{ij}, d_{ij} = 0, j \in \mathcal{N}_i \cup i \cup 0$
1. **for** $k = 0, 1, \cdots$ **do**
2. From every node $j \in \mathcal{N}_i \cup 0$, collect y_{ji}^k and c_{ji}^k
3. Update $x_{ij}^{k+1}, j \in \mathcal{N}_i \cup i \cup 0$ by (11)
4. Update $\{z_{ij}\}$ by (12)
5. From every neighboring edge server $j \in \mathcal{N}_i$, collect x_{ji}^{k+1}
6. Update $y_{ij}, j \in \mathcal{N}_i \cup i$ by (13)
7. Update a_i by (7)
8. Update $c_{ij}, j \in \mathcal{N}_i \cup i \cup 0$ by (16)
9. Update $d_{ij}, j \in \mathcal{N}_i \cup i \cup 0$ by (17)
10. **end for**
11. Calculate $\{r_{ij}\}$ from $\{x_{ij}\}$ by distributed integral projection

3.3. Discussions

The primal-dual updates in the two algorithms have economic explanations. Given service request s_i, edge server i has to split it and assign to $j \in \mathcal{N}_i \cup i \cup 0$, denoted by x_{ij}. Node j allocates resource y_{ji} to meet the assignment x_{ij}, and their difference represents the scarcity of the resource. The Lagrange multiplier c_{ji} plays the role of price, which autonomously adjusts the assignments x_{ij} and the assignments y_{ji}. In ADMM-NC, there are additional variables z_{ji}, which enforce the allocations y_{ij} to be integers. Their corresponding Lagrange multipliers d_{ji} represent the prices of taking the integer constraints into account. To implement the algorithm, edge server i collects its neighbors' assignments x_{ji}, allocations y_{ji} and prices c_{ji}, which help it to make its own decisions x_{ij} and y_{ij}, as well as estimate the prices c_{ij}. In ADMM-NC, the integer constraints are handled locally. Therefore, the decisions z_{ij} and prices d_{ij} do not need to be exchanged with its neighbors.

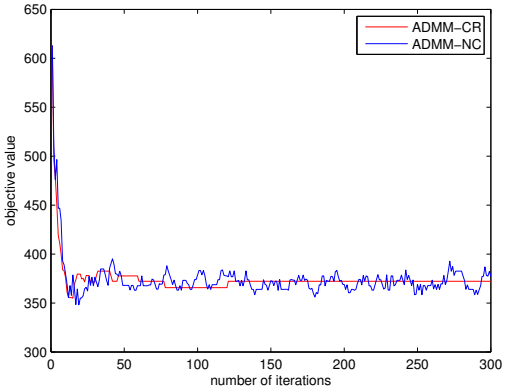

Fig. 2. Comparison of the proposed algorithms.

4. NUMERICAL EXPERIMENTS

In this section, we validate the two proposed algorithms through numerical experiments.

Simulation Setup. We generate a hybrid network with one cloud center and $n = 40$ edge servers. Out of 780 possible communication links between the edge servers, 187 of them are uniformly randomly chosen to be connected. The two edge servers i and j at the two ends of communication link (i, j) are neighbors of each other, and the communication latency is set to be $l_{ij} = 1$. The latency between every edge server i and the cloud center is $l_{0i} = 5$. In the cost function $f_i(\cdot)$ that corresponds to the communication cost, the value of q is set to be 1. The cost function regarding the computation cost is $g_i(x) = k_i x^2$, where $k_i = 1, i = 1, \cdots, N$ and $k_0 = 0.01$. The amounts of service requests at the edge servers are drawn from i.i.d. uniform distribution within $[0, 40]$, followed by rounding down to their nearest integers.

In the two algorithms, we let the ADMM parameter $\rho = 1$ and the total number of iterations be 300. For fair comparison of the two algorithms, we let them run the distributed integral projection step at the end of every iteration, to obtain the resource allocation strategy $\{r_{ij}\}$ from the intermediate values of $\{x_{ij}\}$. Therefore, the two algorithms both yield feasible solutions and we are able to compare their objective function values.

Simulation Results Fig. 2 demonstrates the evolution of the two algorithms. Observe that for both algorithms, the objective value goes down sharply from the initial ~ 600 to ~ 370 within 50 iterations, showing the effectiveness of the proposed algorithms. The curves have fluctuations, which are reasonable because of the nonconvex nature of the integer constrained problem formulation. Comparing with ADMM-CR, ADMM-NC has more frequent small fluctuations; we conjecture that they are from the projections of z_{ij} in (12), which make the iterations unstable. An immediate observation is that, if we allow the cloud center to calculate the overall objective function, then we can record the best result and hence obtain steady curves. This approach, however, incurs higher communication cost and cooperation burden among the cloud center and the edge servers.

5. REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," Communications of the ACM, vol. 53, no. 4, pp. 50–58, 2010

[2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of things," In: Proceedings of MCC, 2012

[3] M. Chiang, "Fog networking: An overview on research opportunities," Manuscript at https://arxiv.org/ftp/arxiv/papers/1601/1601.00835.pdf

[4] R. Johari and J. Tsitsiklis, "Efficiency loss in a network resource allocation game," Mathematics of Operations Research, vol. 29, no. 3, pp. 407–435, 2004

[5] D. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," IEEE Journal on Selected Areas in Communications, vol. 24, no. 8, pp. 1439–1451, 2006

[6] A. Beck, A. Nedic, A. Ozdaglar, and M. Teboulle, "Optimal distributed gradient methods for network resource allocation problems," IEEE Transactions on Control of Network Systems, vol. 1, no. 1, pp. 64–74, 2014

[7] M. Hale, A. Nedic, and M. Egerstedt, "Cloud-based centralized/decentralized multi-agent optimization with communication delays," In: Proceedings of CDC, 2015

[8] H. Huang, Q. Ling, W. Shi, and J. Wang, "Collaborative resource allocation over a hybrid cloud center and edge server network", Manuscript at http://home.ustc.edu.cn/~qingling/pdf/NRA_ADMM.pdf

[9] E. Lawler and D. Wood, "Branch-and-bound methods: A survey," Operations Research, vol. 14, no. 4, pp. 699–719, 1966

[10] C. Feng, H. Xu, and B. Li, "An alternating direction method approach to cloud traffic management," Manuscript at http://arxiv.org/pdf/1407.8309v2.pdf

[11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," Foundations and Trends in Machine Learning, vol. 3, no. 1, pp. 1–122, 2011

[12] S. Diamond , R. Takapoui, and S. Boyd, "A general system for heuristic solution of convex problems over nonconvex sets," Manuscript at http://arxiv.org/pdf/1601.07277v1.pdf