Deadline-Sensitive Mobile Data Offloading via Opportunistic Communications

Guoju Gao†, Mingjun Xiao*†, Jie Wu‡, Kai Han† and Liusheng Huang†
†School of Computer Science and Technology / Suzhou Institute for Advanced Study,
University of Science and Technology of China, Hefei, P. R. China
‡Department of Computer and Information Sciences, Temple University
*Correspondence to: xiaomj@ustc.edu.cn

***Abstract*—With the explosive proliferation of smartphones, many mobile cloud computing applications have emerged in recent years. These applications generally involve many data transmissions between mobile users and the cloud side. In order to reduce the monetary cost of these data transmissions, an effective approach is to offload partial data traffic from cellular networks to WiFi networks, when mobile users pass by some WiFi Access Points (APs). In this paper, we focus on the problem of offloading many deadline-sensitive data items to some WiFi APs with capacity constraints; that is, how to schedule each data item to the WiFi APs, so that we can offload as many data items before their deadlines as possible, while taking the constraints of transmission capacity into consideration. This problem involves a probabilistic combination of multiple 0-1 knapsack constraints, which differs from existing problems. To solve this problem, we propose a greedy oFfline Data Offloading (FDO) algorithm, and prove that this algorithm can achieve an approximation ratio of 2. Moreover, we extend our data offloading strategy to the online decision case, and propose an oNline Data Offloading (NDO) algorithm, which has a competitive ratio of 2. Finally, we demonstrate the significant performances of our algorithms through extensive simulations.**

***Index Terms*—Mobile data offloading, deadline-sensitive, opportunistic offloading, offline and online algorithms.**

I. INTRODUCTION

With the explosive growth of user population and their demands for bandwidth-eager multimedia content in recent years, a big challenge is raised regarding the cellular network. The Cisco VNI report [2] predicts that the number of mobile users will grow from 3.7 billion in 2011 to 4.5 billion by 2016. Furthermore, mobile data traffic is expected to reach 10.8 exabytes per month by 2016, an 18-fold increase over 2011. To cope with the unprecedented traffic load, mobile network operators need to increase their cellular network capacities significantly. However, this is expensive and inefficient. One promising solution to this problem is to offload part of traffic to other coexisting networks, while leaving the capacities of cellular networks unchanged. Some recent research efforts have been focused on offloading cellular traffic to other forms of networks, such as WiFi networks [3, 4, 7–9, 15, 19] and Delay Tolerant Networks (DTNs) [5, 11, 16, 17, 20].

In this paper, we focus on the mobile data offloading based on WiFi networks in mobile cloud computing [6]. Consider the scenario in which a mobile user is performing some mobile cloud computing applications and needs to upload some data items to the cloud side. In order to ensure the quality of the mobile cloud computing applications, each data item needs to be uploaded before a deadline. On the other hand, when the

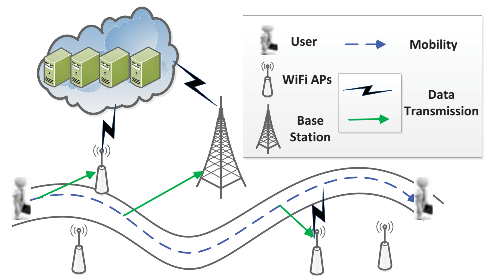


Fig. 1. Data offloading scenario: the mobile user uploads data items onto the cloud side through WiFi networks when it encounters WiFi APs during Time-To-Lives (TTLs) of data items, or via cellular networks when the TTLs of data items expire, respectively.

user conducts the mobile cloud computing applications, it can access cellular networks at any time, anywhere. Meanwhile, the user also might pass by some WiFi APs. Hence, the user can transmit the data items through cellular networks directly, or offload some data to WiFi networks, when it visits a WiFi AP, as shown in Fig. 1. In general, the data transmission via cellular networks has the advantage of instantaneity, but it will lead to a large monetary cost. In contrast, data being offloaded to WiFi networks can save a significant monetary cost, but the instantaneity cannot be ensured. There is a trade-off between the two transmission modes, especially when the transmission capacity of WiFi APs is taken into consideration. Our concern is how to schedule data items between the two transmission modes, so that we can minimize the total monetary cost, while ensuring that each data item be uploaded before its deadline.

The proposed data offloading is different from existing offloading problems [1, 18, 20]. These works in [5, 11, 16, 17] mainly focus on offloading data from cellular networks to DTNs, which is formulated as a target-set selection problem. In addition, the works in [4, 9] study the economic benefits and load balance problem of traffic offloading between cellular networks and WiFi networks from the perspective of Network Services Providers (NSPs). In contrast, we consider the data offloading problem from the user's side. Moreover, our problem can be deduced as an optimization problem with multiple 0-1 knapsack constraints, in which each knapsack is related to a WiFi AP. Adding a data item into a knapsack means offloading this data item via the corresponding WiFi AP. Since the accessibility of each WiFi AP is uncertain, it is a probabilistic event to add a data item into a knapsack. Furthermore, each data item is allowed to be added into multiple knapsacks. Hence, these data items share a combinatorially probabilistic optimization objective. Meanwhile, each data item also needs to be subject to a different deadline constraint. It is because of these features that our problem differs from the existing trivial Multiple Knapsack Problems (MKP) [14].

To this end, we design a special utility function. Furthermore, we propose a greedy offline data offloading algorithm and an online algorithm to solve the aforementioned problem, respectively. The offline algorithm indicates that the mobile user makes the data offloading decisions before it encounters any WiFi AP, while the online algorithm means that the mobile user dynamically makes the immediate data offloading decisions at each time when it visits a WiFi AP. More specifically, our major contributions are summarized as follows:

- We introduce a problem of offloading many deadline-sensitive data items to some WiFi APs with capacity constraints. Then, we formalize it as an optimization problem with multiple 0-1 knapsack constraints, sharing a combinatorially probabilistic optimization objective. Moreover, we prove the NP-hardness of this problem.
- We propose an offline data offloading algorithm, i.e., FDO, to solve the above problem. A greedy strategy is adopted in this algorithm. We prove that this greedy strategy can achieve the approximation ratio of 2.
- We also propose an online data offloading algorithm, i.e., NDO. It is composed of a series of greedy offloading decisions, each of which is made when the mobile user visits a WiFi AP. Moreover, we derive that this algorithm has the competitive ratio of 2.
- We conduct extensive simulations to evaluate the performances of the proposed algorithms. The results show that they can achieve better performances, compared with other algorithms.

The remainder of the paper is organized as follows. We describe the network model, and formulate the optimization problem in Section II. The offline and online algorithms are proposed in Sections III and IV, respectively. In Section V, we evaluate the performances of our algorithms through extensive simulations. After reviewing related work in Section VI, we conclude the paper in Section VII.

II. MODEL & PROBLEM FORMULATION

In this section, we first present our data offloading model, and then formally formulate the problem.

A. *Offloading Model*

We consider that a mobile user is conducting some mobile cloud computing applications, in which the user needs to upload some data to the cloud side. The data can be denoted by a set $\boldsymbol{D} = \{d_1, \cdots, d_i, \cdots, d_n\}$, where $d_i = \langle s_i, t_i \rangle$ $(1 \le i \le n)$, in which s_i and t_i denote the size and Time-To-Live (TTL) of the i-th data item, respectively. Without loss of generality, we assume that these data items are organized in the ascending order of their TTLs, that is, $t_1 \le t_2 \le \ldots \le t_n$. At the same time, each data item is assumed to be indivisible. Moreover, the data item needs to be uploaded successfully before the time when its TTL expires, called the transmission *deadline* of this data item.

On the other hand, the mobile user is assumed to move around in an urban area, so that it can upload these data items to the cloud side, by using cellular networks at any time, anywhere. However, if the user transmits all of these data items through cellular networks, it generally needs to pay many fees for these data transmissions. In this paper, we assume that there are many WiFi APs distributed in the urban area, and the NSP is willing to provide the WiFi-based offloading service, so as to alleviate the load of cellular networks. Hence, in order to reduce the monetary costs, the mobile user can offload some data items via WiFi networks. Since most WiFi APs cannot be accessed for free, the traffic offloading will also produce some costs, but they will be much lower than the transmission cost via cellular networks. In this paper, we use C and c to denote the transmission costs per unit data traffic via cellular networks and WiFi networks, respectively.

In real scenarios, not all WiFi APs can provide the offloading service. It is subject to many factors, such as when the mobile user enters the communication range of a WiFi AP, whether the WiFi AP is accessible, and so on. Moreover, since the time that the user stays in the communication range of one WiFi AP is restricted, the data items that the user can transmit via this WiFi AP are generally limited. That is to say, the transmission capacity is also limited. To this end, we use a triple $w = \langle \tau, p, q \rangle$ to describe the *offloading opportunity* from a WiFi AP, where τ (>0) is the time of the user visiting the WiFi AP, p $(\in (0, 1])$ is the probability of the WiFi AP providing the offloading service, and q (>0) is the transmission capacity of this WiFi AP. In this paper, we assume that NSP has recorded the historical offloading transactions, including the offloading time, transmission rate, and so on. This is reasonable since all offloading operations are conducted via NSP. Based on these historical offloading records and the mobile behavior, each mobile user can derive the offloading opportunity $w = \langle \tau, p, q \rangle$ (from NSP) for each given WiFi AP. More specifically, the probability p can be estimated by the corresponding frequency of historical offloading transactions. The transmission capacity q can be calculated by using the transmission rate and the time that the user stays in the communication range of each WiFi AP.

In addition, we use $\boldsymbol{W} = \{w_1, w_2, \cdots, w_m\}$ to denote all offloading opportunities, where $w_j = \langle \tau_j, p_j, q_j \rangle$ $(1 \le j \le m)$, and $\tau_1 < \tau_2 < \cdots < \tau_m$. Here, if the user visits a WiFi AP more than one time, it can offload data items multiple times, each of which is seen as an offloading opportunity in $\boldsymbol{W}$.

B. *Problem Formulation*

In this paper, we focus on the data items scheduling problem in the above offloading model, that is, how to schedule the data items in $\boldsymbol{D}$ to the offloading opportunities in $\boldsymbol{W}$, so as to minimize the total transmission cost, while ensuring that each data item to be uploaded before its deadline.

Before the problem formulation, we define two terms, for the simplicity of the following description:

Definition 1: [*Data Offloading Operation*] A data offloading operation, denoted by (d_i, w_j), indicates that the i-th data item

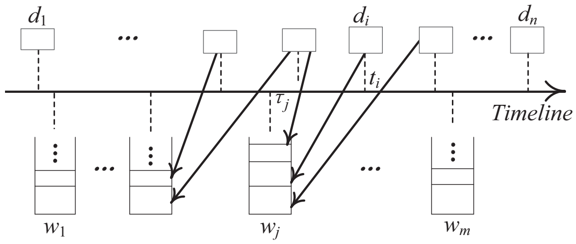


Fig. 2. Data items might be offloaded to multiple WiFi APs.

d_i will be offloaded to the j-th offloading opportunity w_j.

***Definition** 2:* [*Data Offloading Strategy*] A data offloading strategy, denoted by $\mathbf{\Phi}$, is defined as a set of data offloading operations, i.e.,

$$\mathbf{\Phi}=\{(d_i,w_j)|(d_i,w_j)\in \boldsymbol{D}\times\boldsymbol{W}\}. \tag{1}$$

In light of the uncertainty of each offloading opportunity, it is important to note that we allow each data item to be scheduled to multiple offloading opportunities, as shown in Fig. 2, so as to improve the probabilities of being offloaded. If the data item still fails to be uploaded after these offloading opportunities, it will have to be transmitted by using cellular networks, to ensure it be uploaded to the cloud side before its deadline.

For a given data offloading strategy $\mathbf{\Phi}$, we can derive the successful probability of a data item d_i being offloaded to WiFi networks. It is the probability of the data item d_i being offloaded via any one offloading opportunity in $\mathbf{\Phi}$, which is defined as follows:

***Definition** 3:* [*Successful Offloading Probability*] For a given data offloading strategy $\mathbf{\Phi}$, the successful offloading probability of data item d_i, denoted by $\rho_i(\mathbf{\Phi})$, satisfies:

$$\rho_i(\mathbf{\Phi})=1-\prod_{j:(d_i,w_j)\in\mathbf{\Phi}}(1-p_j). \tag{2}$$

Then, according to the probabilities, we can derive the total expected transmission cost of all data items being uploaded, defined as follows:

***Definition** 4:* [*Total Expected Cost*] The total expected transmission cost is the sum of the expected costs of all data items in $\boldsymbol{D}$ being uploaded for a given data scheduling strategy, denoted by $f_{cost}(\mathbf{\Phi})$, which satisfies:

$$f_{cost}(\mathbf{\Phi})=\sum_{i=1}^{n} s_i\big(c\rho_i(\mathbf{\Phi})+C(1-\rho_i(\mathbf{\Phi}))\big). \tag{3}$$

Now, we can formalize our problem as follows:

$$\begin{aligned} &Minimize: && f_{cost}(\mathbf{\Phi})\\ &Subject\ to: && \sum_{i:(d_i,w_j)\in\mathbf{\Phi}} s_i \le q_j,\ 1\le j\le m; \qquad (P1)\\ & && t_i\ge\tau_j,\ \text{for } \forall(d_i,w_j)\in\mathbf{\Phi}\subseteq\boldsymbol{D}\times\boldsymbol{W}. \end{aligned}$$

Here, $\sum_{i:(d_i,w_j)\in\mathbf{\Phi}} s_i \le q_j$, called the *capacity constraint*, means that the total size of data items that are offloaded to the j-th WiFi AP should be no larger than the capacity of the WiFi AP; and, $t_i\ge\tau_j$, called the *deadline constraint*, indicates that each data item d_i can be offloaded via the offloading opportunity w_j, only when the TTL of this data item is no less than the time of the offloading opportunity w_j.

TABLE I
DESCRIPTION OF MAJOR NOTATIONS.

Variable	Description
n, m	the numbers of data items and WiFi APs, respectively.
i, j	the indexes for data items and offloading opportunities, respectively .
$\boldsymbol{D}, \boldsymbol{W}$	the sets of data items and offloading opportunities, respectively.
$\langle s_i, t_i\rangle$	the size and TTL of i-th data item, respectively.
$\langle\tau_j, p_j, q_j\rangle$	the time, probability and capacity of j-th offloading opportunity, respectively.
C, c	transmission cost per unit data traffic via cellular networks and WiFi networks, respectively.
(d_i, w_j)	a data offloading operation (Definition 1).
$\mathbf{\Phi}$	a data offloading strategy (Definition 2).
$\rho_i(\mathbf{\Phi})$	the successful offloading probability of d_i for a given strategy $\mathbf{\Phi}$ (Definition 3).
$\mathbf{\Omega}$	universal set of deadline-satisfying data offloading operations (Definition 6).

By analyzing Eq. (3), we can obtain an equivalent expression as follows:

$$f_{cost}(\mathbf{\Phi})=C\sum_{i=1}^{n} s_i-(C-c)\sum_{i=1}^{n} s_i\rho_i(\mathbf{\Phi}). \tag{4}$$

where $C\sum_{i=1}^{n} s_i$ and $(C-c)$ are known fixed values. Then, we define an offloading utility function as follows:

***Definition** 5:* [*Offloading Utility Function*] The offloading utility function of a data offloading strategy $\mathbf{\Phi}$, denoted by $\mathcal{U}(\mathbf{\Phi})$, is the expected total size of data items that will be offloaded to WiFi networks under this data offloading strategy. Then, $\mathcal{U}(\mathbf{\Phi})$ satisfies:

$$\mathcal{U}(\mathbf{\Phi})=\sum_{i=1}^{n} s_i\rho_i(\mathbf{\Phi}). \tag{5}$$

Since $f_{cost}(\mathbf{\Phi})=C\sum_{i=1}^{n} s_i-(C-c)\mathcal{U}(\mathbf{\Phi})$, the optimization problem (P1) can be equivalently re-formalized as follows:

$$\begin{aligned} &Maximize: && \mathcal{U}(\mathbf{\Phi})\\ &Subject\ to: && \sum_{i:(d_i,w_j)\in\mathbf{\Phi}} s_i \le q_j,\ 1\le j\le m; \qquad (P2)\\ & && t_i\ge\tau_j,\ \text{for } \forall(d_i,w_j)\in\mathbf{\Phi}\subseteq\boldsymbol{D}\times\boldsymbol{W}. \end{aligned}$$

Unlike existing MKP [14], (P2) is an optimization problem with multiple 0-1 knapsack constraints, where each data item might be added into multiple knapsacks, and these data items in all knapsacks must share a combinatorially probabilistic optimization objective. For the ease of reference, we summarize the commonly used notations throughout the paper in Table I.

III. OFFLINE DATA OFFLOADING

In this section, we analyze the hardness of our problem, and then, propose an offline data offloading algorithm, followed by the performance analysis.

A. Problem Hardness Analysis

First, we prove that Problem (P2) cannot be solved in polynomial time unless $P = NP$. More specifically, we have the following theorem:

***Theorem** 1:* Problem (P2) is NP-hard.

Proof: To prove the NP-hardness of Problem (P2), we first consider the following special 0-1 knapsack problem.

$$\begin{aligned} &Maximize: && s_1x_1+s_2x_2+\cdots+s_nx_n \\ &Subject\ to: && s_1x_1+s_2x_2+\cdots+s_nx_n\le S, \qquad (P3)\\ & && x_1,x_2,\cdots,x_n\in\{0,1\}. \end{aligned}$$

where s_i is the size of the i-th item, S is the size of the knapsack, and x_i is a valuable which indicates whether the i-th item is added into the knapsack. The special 0-1 knapsack problem (P3) is NP-hard [12].

Second, we consider a special case of Problem (P2), in which there is only one WiFi AP, i.e., $\boldsymbol{W}=\{\langle\tau_1,p_1,q_1\rangle\}$, and $\tau_1\le t_1$. Such a data offloading problem can be expressed as follows:

$$\begin{aligned} &Maximize: && \sum_{i:(d_i,w_1)\in\boldsymbol{\Phi}} s_i \qquad (P4)\\ &Subject\ to: && \sum_{i:(d_i,w_1)\in\boldsymbol{\Phi}} s_i\le q_1. \end{aligned}$$

Mapping S in Problem (P3) to q_1 in Problem (P4), we can get the two problems to be equivalent. That is, the special case of Problem (P2), is a special 0-1 knapsack problem, which is NP-hard. Thus, Problem (P2) is also NP-hard. ∎

B. The Basic Solution

Since Problem (P2) has both deadline constraints and capacity constraints, we divide our solution into two phases. In the first phase, we take the deadline constraints into account, and determine a set of all data offloading operations that satisfy the deadline constraints, denoted by $\boldsymbol{\Omega}$. This set can be derived with the polynomial time complexity. When we let all data offloading operations in $\boldsymbol{\Phi}$ be selected only from $\boldsymbol{\Omega}$, the data offloading strategy $\boldsymbol{\Phi}$ will be deadline-satisfying, and will not miss any feasible data offloading operations. In this way, we have removed the deadline constraints from Problem (P2). Then, in the second phase, we focus on the optimization problem only with the capacity constraints. Since the problem is NP-hard due to the capacity constraints, we adopt a greedy strategy to approximately solve the problem. We select the data offloading operations from $\boldsymbol{\Omega}$ one by one. In each step, the data offloading operation, which can increase the offloading utility function value most quickly, while ensuring the constraints of transmission capacity, is selected. More specifically, our solution is presented as follows:

First, in order to remove the deadline constraints from Problem (P2), we define and derive the universal set of deadline-satisfying data offloading operations as follows:

***Definition** 6:* [*Universal Set of Deadline-satisfying Offloading Operations*] The universal set of deadline-satisfying offloading operations, i.e, $\boldsymbol{\Omega}$, is a set, including each possible data offloading operation (d_i,w_j) in $\boldsymbol{D}\times\boldsymbol{W}$ that satisfies the deadline constraint, i.e., $t_i\ge\tau_j$. That is:

$$\boldsymbol{\Omega}=\{(d_i,w_j)\mid\forall(d_i,w_j)\in\boldsymbol{D}\times\boldsymbol{W}:t_i\ge\tau_j\}. \qquad (6)$$

According to Definition 6, we can derive the set $\boldsymbol{\Omega}$, by a linearly scanning over the set $\boldsymbol{D}\times\boldsymbol{W}$ and adding the data offloading operations in $\boldsymbol{D}\times\boldsymbol{W}$ that satisfy the deadline constraints into the set $\boldsymbol{\Omega}$. This can be conducted in the polynomial time complexity $O(mn)$.

Algorithm 1 The FDO Algorithm

Require: $\boldsymbol{D},\boldsymbol{W}$.
Ensure: $\boldsymbol{\Phi}$.
1: Initialize $\boldsymbol{\Phi}=\phi$ and $\boldsymbol{\Omega}=\phi$;
2: **for** τ_j from τ_1 to τ_m **do**
3:   **for** t_i from t_1 to t_n **do**
4:     **if** $\tau_j\le t_i$ **then**
5:       $\boldsymbol{\Omega}=\boldsymbol{\Omega}\cup\{(d_i,w_j)\}$;
6: **while** $(\exists\,(d_i,w_j)\in\boldsymbol{\Omega})$ and $(s_i\le q_j)$ **do**
7:   $\{(d_{i_{max}},w_{j_{max}})\}=\underset{(d_i,w_j)\in\boldsymbol{\Omega}\bigwedge s_i\le q_j}{\text{argmax}}\frac{\mathcal{U}(\boldsymbol{\Phi}\cup\{(d_i,w_j)\})-\mathcal{U}(\boldsymbol{\Phi})}{s_i}$;
8:   $(d_{i^*_{max}},w_{j^*_{max}})=\max_{s_{i^*_{max}}>s_{i_{max}}}\{(d_{i_{max}},w_{j_{max}})\}$;
9:   $\boldsymbol{\Phi}=\boldsymbol{\Phi}\cup\{(d_{i^*_{max}},w_{j^*_{max}})\}$;
10:   $\boldsymbol{\Omega}=\boldsymbol{\Omega}-\{(d_{i^*_{max}},w^*_{j_{max}})\}$;
11:   $q_j=q_j-s_{i^*_{max}}$, where $s_{i^*_{max}}$ is the size of $d_{i^*_{max}}$;
12: **return** $\boldsymbol{\Phi}$;

Second, after we have derived the set $\boldsymbol{\Omega}$, we greedily select the data offloading operations from $\boldsymbol{\Omega}$, and add them into the data offloading strategy $\boldsymbol{\Phi}$. The greedy criterion in each round of selection is that the data offloading operation should increase the offloading utility function value most quickly, while ensuring the constraints of transmission capacity. Based on this criterion, we repeatedly select the data offloading operations from $\boldsymbol{\Omega}$, until the capacity constraints will be broken. The criterion in each round can be formulated as follows:

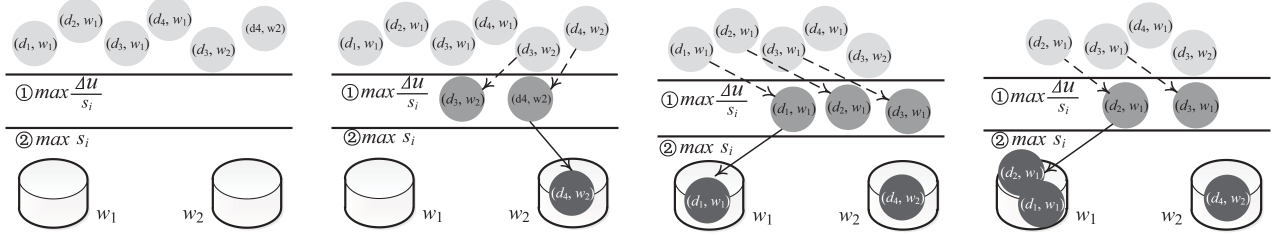
$$\{(d_{i_{max}},w_{j_{max}})\}=\underset{(d_i,w_j)\in\boldsymbol{\Omega}\bigwedge s_i\le q_j}{\text{argmax}}\frac{\mathcal{U}(\boldsymbol{\Phi}\cup\{(d_i,w_j)\})-\mathcal{U}(\boldsymbol{\Phi})}{s_i}. \qquad (7)$$

$$(d_{i^*_{max}},w_{j^*_{max}})=\max_{s_{i^*_{max}}>s_{i_{max}}}\{(d_{i_{max}},w_{j_{max}})\}. \qquad (8)$$

Here, in each round, $\boldsymbol{\Phi}$ is the data offloading strategy, only including the data offloading operations that have been previously determined, and the capacity q_j is the remaining allowed transmission capacity. If Eq. 7 produces multiple offloading operations, we will select the offloading operation, whose data item has the largest data size, as shown in Eq. 8.

C. The Detailed Algorithm

Based on the above solution, we design the greedy algorithm to approximately solve the optimization problem (P2), as shown in Algorithm 1. In Step 1, the data offloading strategy $\boldsymbol{\Phi}$ and the universal set of deadline-satisfying offloading strategy $\boldsymbol{\Omega}$ are initialized to be empty. Then, from Step 2 to Step 5, we add all deadline-satisfying data offloading operations into the set $\boldsymbol{\Omega}$. Next, in Steps 7-8, we choose each data offloading operation from $\boldsymbol{\Omega}$, which can increase the offloading utility function per unit data size most quickly, while ensuring the capacity constraints. Then, we add this data offloading operation into the set $\boldsymbol{\Phi}$ and remove this data offloading operation from the set $\boldsymbol{\Omega}$ in Steps 9 and 10, respectively. After determining a data offloading operation, the remaining capacity of each offloading opportunity is updated in Step 11.



(a) The initial state (b) Select (d_4, w_2) (c) Select (d_1, w_1) (d) Select (d_2, w_1)

Fig. 3. Example: greedily schedule the data items d_1, d_2, d_3, d_4 to offloading opportunities w_1, w_2 , and the obtained data offloading strategy is $\mathbf{\Phi} = \{(d_4, w_2), (d_1, w_1), (d_2, w_1)\}$. The dashed lines and solid lines indicate the selected offloading operations according to the greedy criterion max $\frac{\Delta \mathcal{U}}{s_i} = \frac{\mathcal{U}(\mathbf{\Phi}\cup\{(d_i,w_j)\})-\mathcal{U}(\mathbf{\Phi})}{s_i}$ and max s_i, respectively.

At last, when not existing (d_i, w_j) in the set $\mathbf{\Omega}$ satisfies the capacity constraints, the algorithm terminates and outputs the data offloading strategy $\mathbf{\Phi}$, in Step 12.

By analyzing Algorithm 1, we show that the algorithmic procedures are all pseudo-polynomial-time. More specifically, we can straightforwardly demonstrate correctness of the algorithm in the following theorem:

Theorem 2: Algorithm 1 is correct. It will terminate for sure, and will produce a feasible data offloading strategy.

Proof: Since each data offloading operation is selected from the universal set of deadline-satisfying data offloading operations, a limited set, the algorithm will terminate for sure, and the results will satisfy the deadline constraints. On the other hand, at each round of selection in Algorithm 1, the capacity constraints are ensured. Thus, the produced data offloading strategy must be feasible. ■

In addition, the computational overhead of Algorithm 1 is dominated by Step 7, which is $O(m^2n^2)$.

D. Examples

To better understand Algorithm 1, we present an example to show the data offloading procedure, in which the mobile user has the four data items $\boldsymbol{D} = \{d_i = \langle s_i, t_i\rangle | 1 \le i \le 4\}$, where $s_1 = 8, t_1 = 11, s_2 = 6, t_2 = 13, s_3 = 5, t_3 = 17, s_4 = 10, t_4 = 18$, and it wishes to offload the data items to two WiFi APs $w_1 = \langle \tau_1, p_1, q_1\rangle$ and $w_2 = \langle \tau_2, p_2, q_2\rangle$, where $\tau_1 = 10, p_1 = 0.6, q_1 = 15, \tau_2 = 15, p_2 = 0.9, q_2 = 10$. Since the deadline constraints $\tau_1 < t_1 < t_2 < \tau_2 < t_3 < t_4$ are satisfied, the $\mathbf{\Omega}$ is first determined in Fig. 3(a). Then, Algorithm 1 greedily selects offloading operations as follows:

First round, $\mathbf{\Phi} = \phi$. Algorithm 1 computes the increased offloading utility values per unit data size for each data offloading operation (d_i, w_j) in $\mathbf{\Omega}$. The results are as follows:

$$\frac{\mathcal{U}(\mathbf{\Phi}\cup\{(d_1,w_1)\})-\mathcal{U}(\mathbf{\Phi})}{s_1}=0.6, \frac{\mathcal{U}(\mathbf{\Phi}\cup\{(d_2,w_1)\})-\mathcal{U}(\mathbf{\Phi})}{s_2}=0.6,$$
$$\frac{\mathcal{U}(\mathbf{\Phi}\cup\{(d_3,w_1)\})-\mathcal{U}(\mathbf{\Phi})}{s_3}=0.6, \frac{\mathcal{U}(\mathbf{\Phi}\cup\{(d_4,w_1)\})-\mathcal{U}(\mathbf{\Phi})}{s_4}=0.6,$$
$$\frac{\mathcal{U}(\mathbf{\Phi}\cup\{(d_3,w_2)\})-\mathcal{U}(\mathbf{\Phi})}{s_3}=0.9, \frac{\mathcal{U}(\mathbf{\Phi}\cup\{(d_4,w_2)\})-\mathcal{U}(\mathbf{\Phi})}{s_4}=0.9.$$

According to the results, we select two data offloading operations: $\{(d_3, w_2), (d_4, w_2)\}$. Next, due to $s_4 > s_3$, the offloading operation (d_4, w_2) is selected finally, i.e., $\mathbf{\Phi} = \{(d_4, w_2)\}$, and it is removed from $\mathbf{\Omega}$. Accordingly, the data item d_4 will be offloaded via w_2, and the remaining capacity of w_2 is 0, as shown in Fig. 3(b).

Second round, $\mathbf{\Phi} = \{(d_4, w_2)\}$, and we have:

$$\frac{\mathcal{U}(\mathbf{\Phi}\cup\{(d_1,w_1)\})-\mathcal{U}(\mathbf{\Phi})}{s_1}=0.6, \frac{\mathcal{U}(\mathbf{\Phi}\cup\{(d_2,w_1)\})-\mathcal{U}(\mathbf{\Phi})}{s_2}=0.6,$$
$$\frac{\mathcal{U}(\mathbf{\Phi}\cup\{(d_3,w_1)\})-\mathcal{U}(\mathbf{\Phi})}{s_3}=0.6, \frac{\mathcal{U}(\mathbf{\Phi}\cup\{(d_4,w_1)\})-\mathcal{U}(\mathbf{\Phi})}{s_4}=0.06.$$

Similarly, we will select three data offloading operations from $\mathbf{\Omega}$: $\{(d_1, w_1), (d_2, w_1), (d_3, w_1)\}$. Since $s_1 > s_2 > s_3$, (d_1, w_1) is selected finally. Then, $\mathbf{\Phi} = \{(d_4, w_2), (d_1, w_1)\}$, and (d_1, w_1) is removed from $\mathbf{\Omega}$, as shown in Fig. 3(c). Now, due to the capacity constraints of w_1, only the remaining data offloading operations $\{(d_2, w_1)$ and $(d_3, w_1)\}$ can be executed.

Third round, we continue to compute

$$\frac{\mathcal{U}(\mathbf{\Phi}\cup\{(d_2,w_1)\})-\mathcal{U}(\mathbf{\Phi})}{s_2}=0.6, \frac{\mathcal{U}(\mathbf{\Phi}\cup\{(d_3,w_1)\})-\mathcal{U}(\mathbf{\Phi})}{s_3}=0.6.$$

Then, $\mathbf{\Phi} = \{(d_4, w_2), (d_1, w_1), (d_2, w_1)\}$ after selecting (d_2, w_1), as shown in Fig. 3(d). Due to the capacity constraints, the algorithm terminates after this round of selection, and $\mathbf{\Phi} = \{(d_4, w_2), (d_1, w_1), (d_2, w_1)\}$ is the final result.

E. Performance Analysis

In this subsection, we analyze the approximation ratio of Algorithm 1. First, we use $\boldsymbol{opt}_F$ to denote the optimal offline offloading strategy of optimization problem (P2). Then, we have the following theorem:

Theorem 3: Algorithm 1 has an approximation ratio of 2. That is,

$$\frac{\mathcal{U}(\boldsymbol{opt}_F)}{\mathcal{U}(\mathbf{\Phi})} < 2. \tag{9}$$

Proof: First, we consider a special solution. For this solution, we assume that all data items can be divided, and let each data item $d_i = \langle s_i, t_i\rangle$ be divided as $d_{i_1} = \langle 1, t_i\rangle$, $\cdots$, $d_{i_{s_i}} = \langle 1, t_i\rangle$. Then, we conduct our Algorithm 1 to get a solution, denoted by $\boldsymbol{opt}_F^*$. When all data items are divisible, the greedy strategy in Algorithm 1 can achieve the optimal result. This is because the problem has the property of optimal substructure, the best offloading operation is selected in each round, and the transmission capacity of each offloading opportunity is fully utilized. Additionally, $\boldsymbol{opt}_F$ is the optimal strategy where data items are indivisible. Due to the indivisible data items, $\boldsymbol{opt}_F$ cannot fully utilize the transmission capacity of each offloading opportunity in most cases. Hence, we have:

$$\mathcal{U}(\boldsymbol{opt}_F^*) \ge \mathcal{U}(\boldsymbol{opt}_F). \tag{10}$$

Second, we consider the greedy criterion Eq. 7, used in Algorithm 1. According to Definitions 3 and 5, we have:

$$\frac{\mathcal{U}(\mathbf{\Phi}\cup\{(d_i,w_j)\})-\mathcal{U}(\mathbf{\Phi})}{s_i} = p_j(1-\rho_i(\mathbf{\Phi})). \tag{11}$$

This shows that the offloading operation selection based on Eq. 7 at each round in Algorithm 1 is irrelative to the size of the data item. This means that, when we do not consider the capacity constraints, Algorithm 1 uses the same greedy criterion and produces the same result, regardless if data items are divisible or indivisible. Here, we consider another special solution for the case where data items are indivisible, i.e., a result produced by Algorithm 1 while the capacity constraint of each offloading opportunity can be broken once. Denote this solution as $\boldsymbol{opt}_F^+$. Note that $\boldsymbol{opt}_F^+$ and $\boldsymbol{opt}_F^*$ are produced by using the same greedy criterion, while $\boldsymbol{opt}_F^+$ can offload data items beyond each capacity constraint once. Thus, we have:

$$\mathcal{U}(\boldsymbol{opt}_F^+) \geq \mathcal{U}(\boldsymbol{opt}_F^*). \tag{12}$$

Now, we compare $\boldsymbol{opt}_F^+$ and $\mathbf{\Phi}$. Without loss of generality, we assume that there are totally g data offloading operations in $\mathbf{\Omega}$, which correspond to the j-th offloading opportunity w_j, denoted as $\{(d_{i_1}, w_j), \cdots, (d_{i_k}, w_j), \cdots, (d_{i_g}, w_j)\}$, in which $d_{i_k} = \langle s_{i_k}, t_{i_k} \rangle$. Moreover, we assume that these data offloading operations are organized in descending order of their data sizes, that is, $s_{i_1} \geq s_{i_2} \geq \cdots \geq s_{i_g}$. According to Algorithm 1, for a given offloading opportunity w_j, the data item with the largest size will be selected first. Without loss of generality, we assume that h $(1 \leq h \leq g)$ data offloading operations in $\mathbf{\Omega}$ are selected and added into $\mathbf{\Phi}$ by Algorithm 1. Then, $(d_{i_1}, w_j), \cdots, (d_{i_h}, w_j) \in \mathbf{\Phi}$, and $(d_{i_1}, w_j), \cdots, (d_{i_{h+1}}, w_j) \in \boldsymbol{opt}_F^+$, according to the definition of $\boldsymbol{opt}_F^+$. Since $s_{i_{h+1}} \leq s_{i_h} \leq \cdots \leq s_{i_1}$, we have:

$$2\sum_{k=1}^{h} s_{i_k} > \sum_{k=1}^{h} s_{i_k} + s_{i_{h+1}} > q_j, \quad \text{for } \forall j \in [1, m]. \tag{13}$$

Moreover, $(d_{i_1}, w_j), \cdots, (d_{i_h}, w_j)$ are selected by $\boldsymbol{opt}_F^+$ and $\mathbf{\Phi}$ with the same strategy. Thus, we can get:

$$2\mathcal{U}(\mathbf{\Phi}) > \mathcal{U}(\boldsymbol{opt}_F^+). \tag{14}$$

Based on Eqs. 10, 12, and 14, we can get that the theorem is correct. ∎

IV. ONLINE DATA OFFLOADING

In this section, we propose the online data offloading algorithm, in which the data offloading decision is made only when the mobile user encounters the WiFi APs.

A. The Basic Idea

The basic idea of online algorithm is that the mobile user makes the data offloading decisions only when it encounters the offloading opportunities. The detailed solution is presented as follows. When the mobile user encounters the j-th WiFi AP, the probability of the j-th offloading opportunity becomes 1 from p_j. Then, we replace the probability p_j by 1, and conduct the same greedy strategy as that in the offline case to produce an offloading strategy. According to each offloading operation (d_i, w_j) in this offloading strategy, that corresponds to w_j, the user makes the decision to offload d_i to w_j, while ignoring other offloading operations. Once the user makes this online offloading decision, the selected data items will be offloaded to w_j for sure. The data items will not be considered in future offloading decisions, and will be removed from $\mathbf{\Omega}$. For simplicity of description, we say these selected offloading operations *final results*. Since those ignored offloading operations might be updated when the user encounters the next WiFi AP, they are called *temporary results*. Hence, all offloading operations, which correspond to w_j or to the offloading opportunity before w_j, are final results, while others are temporary results. Particularly, when $j = m$, all offloading operations will be final results.

Algorithm 2 The NDO Algorithm

Require: $\boldsymbol{D}$, $\boldsymbol{W}$.

Ensure: $\mathbf{\Phi}^*$.

1: **for** each w_j from $j = 1$ to m **do**
2:   Initialize $\mathbf{\Omega} = \phi$;
3:   **if** the user meets w_j **then**
4:     $p_j = 1$;
5:   **else**
6:     $p_j = 0$;
7:   **for** τ from τ_j to τ_m **do**
8:     **for** t_i from t_1 to t_n **do**
9:       **if** $\tau \leq t_i$ **then**
10:         $\mathbf{\Omega} = \mathbf{\Omega} \cup \{(d_i, w_j)\}$;
11:   Initialize $\mathbf{\Phi} = \phi$;
12:   **while** $(\exists\, (d_i, w_j) \in \mathbf{\Omega})$ and $(s_i \leq q_j)$ **do**
13:     $\{(d_{i_{max}}, w_{j_{max}})\} = \underset{(d_i,w_j)\in\mathbf{\Omega} \bigwedge s_i \leq q_j}{\text{argmax}} \frac{\left(\mathcal{U}(\mathbf{\Phi}\cup\{(d_i,w_j)\}) - \mathcal{U}(\mathbf{\Phi})\right)}{s_i}$;
14:     $(d_{i^*_{max}}, w_{j^*_{max}}) = \underset{s_{i^*_{max}} > s_{i_{max}}}{max} \{(d_{i_{max}}, w_{j_{max}})\}$;
15:     $\mathbf{\Phi} = \mathbf{\Phi} \cup \{(d_{i^*_{max}}, w_{j^*_{max}})\}$;
16:     $\mathbf{\Omega} = \mathbf{\Omega} - \{(d_{i^*_{max}}, w_{j^*_{max}})\}$;
17:     $q_j = q_j - s_{i^*_{max}}$, where $s_{i^*_{max}}$ is the size of $d_{i^*_{max}}$;
18:   $\mathbf{\Phi}_j = \cup_{h=1}^{j-1}\{(d_i, w_h) | (d_i, w_h) \in \mathbf{\Phi}_h\} \cup \mathbf{\Phi}$;
19: **return** $\mathbf{\Phi}^* = \mathbf{\Phi}_m$;

B. The Detailed Algorithm

The detailed algorithm is presented in Algorithm 2. First, the universal set of deadline-satisfying offloading operations, i.e., $\mathbf{\Omega}$, is initialized to be empty in Step 2. Then, for all offloading opportunities in $\boldsymbol{W}$, if the mobile user encounters the j-th offloading opportunity w_j , the accessing probability of j-th offloading opportunity p_j is replaced by 1, otherwise $p_j = 0$, in Steps 3-6. Next, Algorithm 2 performs the same steps as Algorithm 1, to determine $\mathbf{\Omega}$ according to the deadline constraints in Steps 7-10. Moreover, the data offloading strategy $\mathbf{\Phi}$ is initialized to be empty in Step 11. In Steps 13-14, we perform the same operations as those in Algorithm 1, to select the data offloading operations which increase the offloading utility function per unit data size most. At the same time, we add the selected data offloading operation into $\mathbf{\Phi}$ and remove it from $\mathbf{\Omega}$ in Steps 15 and 16, respectively. After determining a data offloading operation, the remaining capacity of each offloading opportunity is updated in Step 17.

In Algorithm 2, we use $\mathbf{\Phi}_j$ to denote the current data offloading strategy, which includes the data offloading operations selected before w_j, and the data offloading operations determined when the user meets the j-th WiFi AP. Among them, the data offloading operations $\cup_{h=1}^{j-1}\{(d_i,w_h)|(d_i,w_h)\in\mathbf{\Phi}_h\}$ are final results that have been determined before w_j. $\mathbf{\Phi}$ includes the data offloading operations that are determined in the current round of decision. In $\mathbf{\Phi}$, only data offloading operations that correspond to w_j are final results, while the others are temporary results. Along with the increase of j, the final results will expand. Finally, when $j=m$, all data offloading operations in $\mathbf{\Phi}$ are final results, denoted by $\mathbf{\Phi}^*$. That is, $\mathbf{\Phi}^*=\mathbf{\Phi}_m$, as shown in Step 19.

In addition, the computational overhead of Algorithm 2 is dominated by Step 13, which is $O(m^3n^2)$.

C. Performance Analysis

In this subsection, we analyze the competitive ratio of our online algorithm. Assume that there is a god, who can foresee whether the mobile user will encounter each offloading opportunity. Based on this knowledge, the god can give an optimal offloading strategy, denoted by $\boldsymbol{opt}_N$. Note that, we have discussed an optimal strategy $\boldsymbol{opt}_F$ in the last section, which is actually the best offline offloading strategy based on the expected offloading opportunities. In contrast, $\boldsymbol{opt}_N$ is the best strategy based on the knowledge that each offloading opportunity can be foresaw definitely. Hence, $\boldsymbol{opt}_N$ is even better than $\boldsymbol{opt}_F$. Here, the competitive ratio is defined as the ratio of $\boldsymbol{opt}_N$ and our solution. Then, we have:

***Theorem** 4:* The competitive ratio of NDO satisfies

$$\frac{\mathcal{U}(\boldsymbol{opt}_N)}{\mathcal{U}(\mathbf{\Phi}^*)}<2. \tag{15}$$

Proof: First, we consider a special solution. That is, we assume that the god not only knows whether the mobile user will encounter each offloading opportunity, but also can divide each data item. Then, it can produce an online data offloading strategy for this case, denoted by $\boldsymbol{opt}_N^*$. Since $\boldsymbol{opt}_N^*$ not only includes all optimal offloading decisions, but

$$\mathcal{U}(\boldsymbol{opt}_N^*)\geq\mathcal{U}(\boldsymbol{opt}_N). \tag{16}$$

Second, we compare the utility values of the two offloading strategies: $\boldsymbol{opt}_N^*$ and $\mathbf{\Phi}^*$. Note that, in the online algorithms, $p_j=1$ or $p_j=0$. This implies that each data item either will be offloaded with the probability of 1, or will not be offloaded. Hence, the utility value of an online algorithm actually equals to the total size of all data items that are offloaded by this algorithm. Then, according to the definition of $\boldsymbol{opt}_N^*$, we have:

$$\mathcal{U}(\boldsymbol{opt}_N^*)=\sum_{j=1}^{m}q_j. \tag{17}$$

Next, we focus on $\mathcal{U}(\mathbf{\Phi}^*)$. Without loss of generality, we consider that the mobile user encounters an arbitrary offloading opportunity w_j, and we assume that there are g data offloading operations via w_j in $\mathbf{\Omega}$, denoted as $\{(d_{i_1},w_j),\cdots,(d_{i_k},w_j),\cdots,(d_{i_g},w_j)\}$, in which $d_{i_k}=\langle s_{i_k},t_{i_k}\rangle$ and $s_{i_1}\geq s_{i_2}\geq\cdots\geq s_{i_g}$. Based on Algorithm 2, the offloading operation whose data item with the largest size will be selected first. Without loss of generality, we assume that $h(1\leq h\leq g)$ data offloading operations in $\mathbf{\Omega}$, i.e., $\{(d_{i_1},w_j),\cdots,(d_{i_h},w_j)\}$ are selected and added into $\mathbf{\Phi}^*$ by Algorithm 2. Then, we have:

$$\sum_{k=1}^{h}s_{i_k}+s_{i_{h+1}}>q_j. \tag{18}$$

It needs to be pointed out that Eq. 18 must be correct; otherwise, the $(h+1)$-th data offloading operation will be added into $\mathbf{\Phi}^*$ by Algorithm 2. Since $s_{i_{h+1}}\leq s_{i_h}\leq\cdots\leq s_{i_1}$, we have:

$$2\sum_{k=1}^{h}s_{i_k}>\sum_{k=1}^{h}s_{i_k}+s_{i_{h+1}}>q_j. \tag{19}$$

Consider each w_j from $j=1$ to m. Then, we can get:

$$2\mathcal{U}(\mathbf{\Phi}^*)=2\sum_{j=1}^{m}\sum_{i_k:(d_{i_k},w_j)\in\mathbf{\Phi}^*}s_{i_k}>\sum_{j=1}^{m}q_j=\mathcal{U}(\boldsymbol{opt}_N^*). \tag{20}$$

Comparing Eqs. 16 and 20, we can get that the theorem holds. ∎

V. EVALUATION

We conduct extensive simulations to evaluate the performances of our algorithms. The compared algorithms, the simulation settings, and the results are presented as follows.

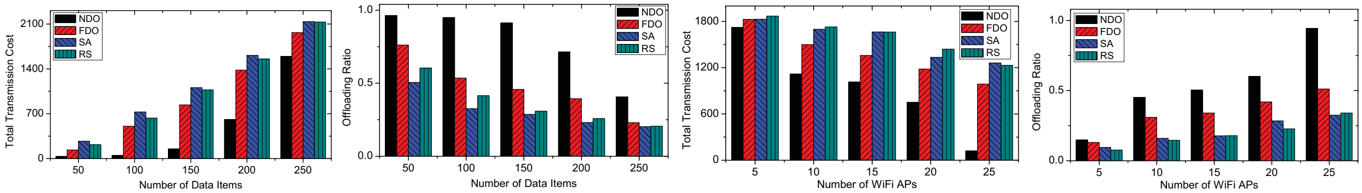
A. Compared Algorithms

In order to evaluate the performances of our algorithms, we implement two other scheduling algorithms for comparison: RS (Random Selection) and SA (Sequential Allocation). As we discussed in Section I, our problem is different from the existing works. Previous offloading algorithms cannot be applied in our problem directly. Hence, we carefully design RS and SA. In the RS algorithm, all data offloading operations are randomly selected from $\mathbf{\Omega}$, while satisfying capacity constraints of offloading opportunities. In the SA algorithm, each data offloading operation in the set $\mathbf{\Omega}$ is selected sequentially, until the total size of selected data items exceeds the capacity of the corresponding offloading opportunity.

B. Simulation Settings and Metrics

We begin by introducing the simulation settings, and let the transmission costs per unit data traffic via cellular networks and WiFi networks be $C=0.1$ and $c=0.001$, respectively. In order to evaluate the performances of our algorithms with different numbers of data items and WiFi APs, we let the numbers of data items and WiFi APs be selected from $\{50,100,\cdots,250\}$ and $\{5,10,\cdots,25\}$, respectively.

Then, we take the four attributes of data items and WiFi APs into consideration as follows. The sizes and TTLs of these data items are randomly produced in $[0,2l]$ and $[0,2t]$, where l and t are the average size and TTL of data items, respectively. In the simulations, l and t are selected from the sets $\{100,200,\cdots,500\}$ and $\{50,100,\cdots,250\}$, respectively. Since the time that WiFi APs can be accessed is generally close to the TTLs of data items, we also let them be randomly generated in $[0,2t]$. The capacities of WiFi APs are randomly and uniformly generated in $[0,2L]$, where L is the average capacity. Additionally, the L is selected from the set $\{1000,2000,\cdots,5000\}$. The probabilities of contacting WiFi networks are produced in $[0,2p]$ randomly, and p is selected from the set $\{0.1,0.15,\cdots,0.3\}$, which is used to generate contact events.



(a) Cost vs. Number of Data Items (b) Ratio vs. Number of Data Items (c) Cost vs. Number of WiFi APs (d) Ratio vs. Number of WiFi APs

Fig. 4. Performance comparisons: total transmission cost and offloading ratio vs. the number of data items (the number of WiFi APs $m = 10$) and the number of WiFi APs (the number of data items $n = 150$).

In a generic WiFi-based offloading model, the most important performance metrics include the amount of offloaded data and the offloading delay. However, in our mobile data offloading model, the offloading delay is used as the deadline constraints. Since our primitive optimization problem is to minimize the total data transmission cost, the total transmission cost of all data items is used as a most leading metric in our simulation. In addition to the total transmission cost, we also evaluated the data offloading ratio (OR) which is defined as Eq. (21):

$$OR = \frac{\sum_{i=1}^{n} s_i \rho_i(\mathbf{\Phi})}{\sum_{i=1}^{n} s_i} \tag{21}$$

C. Evaluation Results

In this subsection, we present the simulation results of the four algorithms. First, we evaluate the performances of the four algorithms with different numbers of data items and WiFi APs. In the first group of simulations, we conduct these algorithms by changing the number of data items, while keeping the number of WiFi APs fixed. The results of total transmission cost and offloading ratio are shown in Figs. 4(a) and 4(b). FDO and NDO achieve about 22.8% and 70.5% smaller total transmission costs than the two compared algorithms, respectively. Additionally, NDO has a better performance than FDO. In contrast, NDO has the biggest offloading ratio, and FDO follows. Their offloading ratio are about 145% and 43.0% larger than the compared algorithms, respectively. In the second group of simulations, we evaluate the performances of the four algorithms by changing the number of WiFi APs, while keeping the number of data items fixed. The results of the total transmission cost and offloading ratio are shown in Figs. 4(c) and 4(d). As we expected, NDO achieves the best performance, and FDO follows. The total transmission costs of FDO and NDO are about 13.5% and 44.0% smaller than those of the compared algorithms, respectively. The offloading ratio of NDO achieves the best result; the FDO and the two compared algorithms decrease stepwise. Moreover, when the number of data items increases, the total transmission costs of all algorithms increase, and the offloading ratios decrease; when the number of WiFi APs increases, the total transmission costs decrease, and the offloading ratios increase. These simulations validate our theoretical analysis results.

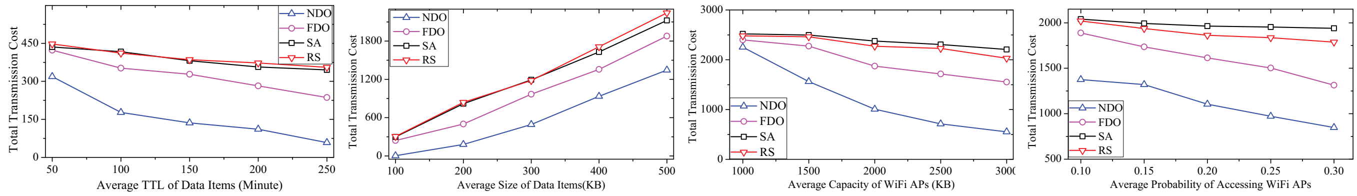
Then, we evaluate the performances of four algorithms, taking the sizes and TTLs of all data items, the capacity, the accessible probability and time of each WiFi AP into consideration. Additionally, when we change one of the parameters for evaluation, we keep the other parameters fixed. The results are shown in Figs. 5 and 6. The results also demonstrate that FDO and NDO achieve better performances than RS and SA, and NDO has the best performance among the four algorithms. In addition, along with the increase of the average TTL of data items, the average capacity of WiFi APs and average probability of mobile user visiting WiFi APs, the total transmission costs decrease. The offloading ratios of all algorithms increase simultaneously. However, along with the increase of the average size of data items, the total transmission costs increase, while the offloading ratios decrease. These simulations remain consistent with our theoretical analysis results.

VI. RELATED WORK

In this paper, we focus on the data transmission cost problem in mobile cloud computing applications, in which these offloading data items must share a combinatorially probabilistic optimization objective. By far, there has been much research on the data offloading problem, such as [4, 5, 9–11, 13, 16, 17, 20]. In a broad sense, offloading cellular traffic can be mainly classified into two categories: offloading to WiFi networks [4, 9, 10, 13] and offloading to DTNs [11, 20].

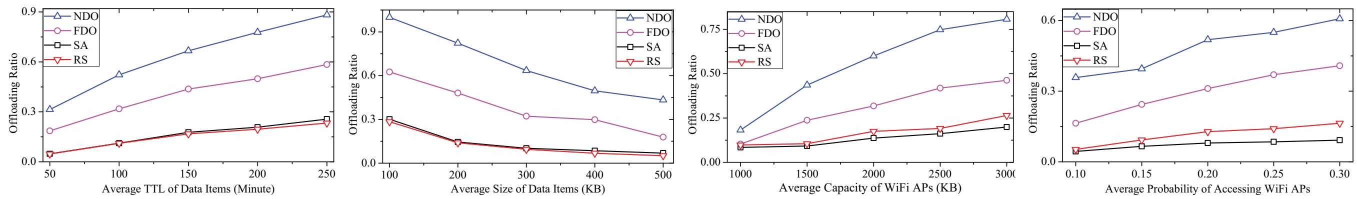
Without loss of generality, data offloading through third-party WiFi APs or femtocell APs requires the cooperation and agreement of both the mobile cellular network operators (MNOs) and AP owners (APOs). Gao *et al.* [4] developed a model to analyze the interaction among one MNO and multiple APOs by using Nash bargaining theory. Moreover, Lee *et al.* [9] studied the economic benefits generated due to delayed WiFi offloading, by analyzing the traffic load balance between cellular networks and WiFi networks. In the work [7], the heterogeneous network is responsible for collecting the network information, and decides the specific portion of traffic to be transmitted via WiFi, to maximize the per-user throughput. Different from the aforementioned work, our purpose is to minimize the total transmission cost from the perspective of mobile users. Additionally, we take the deadline constraints and the capacity constraints into consideration simultaneously.

Furthermore, our work is also different from the offloading using DTNs. For example, Zhuo *et al.* [20] mainly investigated the tradeoff between the amount of traffic being offloaded and the users' satisfaction. Then, they proposed a novel incentive offloading target where users with large offloading potential will be prioritized for traffic offloading. Additionally, the work [5] exploited the DTNs to facilitate the information dissemination, and investigated the target-set selection problem for information delivery to minimize the cellular data traffic. Different from the existing problems, we formulate the objective of achieving the minimum of data transmission cost from a mobile device to the cloud side. Then, we deduce



(a) Cost vs. TTL (b) Cost vs. Data Size (c) Cost vs. Capacity (d) Cost vs. Accessing Probability

Fig. 5. Performance comparisons on the total transmission cost with the different average TTLs of data items, average sizes of data items, average capacities of WiFi APs, or average probabilities of accessing WiFi APs.



(a) Ratio vs. TTL (b) Ratio vs. Data Size (c) Ratio vs. Capacity (d) Ratio vs. Accessing Probability

Fig. 6. Performance comparisons on the offloading ratio with the different average TTLs of data items, average sizes of data items, average capacities of WiFi APs, or average probabilities of accessing WiFi APs.

the problem as an optimization problem with a probabilistic combination of multiple 0-1 knapsack constraints, which also differs from the existing MKP [14].

VII. CONCLUSION

We have studied the problem of how to offload multiple mobile data items from cellular networks to WiFi networks, to minimize the total expected data transmission cost from the perspective of mobile users. Additionally, these data items are heterogeneous in data sizes and TTLs, and the capacities of WiFi networks are limited. We first prove the NP-hardness of our data offloading problem. Then, we propose a greedy strategy to maximize the data offloading utility at each step, satisfying the deadline constraints and capacity constraints simultaneously. Based on this greedy strategy, we design the offline algorithm FDO and the online algorithm NDO to solve our optimization problem. We prove that FDO achieves the approximation ratio of 2, and NDO achieves the competitive ratio of 2. At last, extensive simulations are conducted to verify the significant performances of our algorithms.

Acknowledgment

This research was supported by the National Natural Science Foundation of China (NSFC) (Grant No. 61572457, 61379132, 61303206, 61572342, 61502261) and the Natural Science Foundation of Jiangsu Province in China (Grant No. BK20131174, BK2009150).

References

[1] N. Cheng, N. Lu, N. Zhang, X. Shen, and J. Mark. Vehicular wifi offloading: Challenges and solutions. *Vehicular Communications*, 1(1):13–21, 2014.

[2] I. Cisco. Cisco visual networking index: Forecast and methodology. *CISCO White paper*, pages 2011–2016, 2012.

[3] H. Deng and I.-H. Hou. Online scheduling for delayed mobile offloading. In *IEEE INFOCOM*, 2015.

[4] L. Gao, G. Iosifidis, J. Huang, L. Tassiulas, and D. Li. Bargaining-based mobile data offloading. *IEEE Journal on SAC*, 32(6):1114–1125, 2014.

[5] B. Han, P. Hui, V. Kumar, M. Marathe, J. Shao, and A. Srinivasan. Mobile data offloading through opportunistic communications and social participation. *IEEE Transactions on Mobile Computing*, 11(5):821–834, 2012.

[6] Z. Hao, Y. Tang, Y. Zhang, E. Novak, N. Carter, and Q. Li. Smoc: a secure mobile cloud computing platform. In *IEEE INFOCOM*, 2015.

[7] B. H. Jung, N.-O. Song, and D. K. Sung. A network-assisted user-centric wifi-offloading model for maximizing peruser throughput in a heterogeneous network. *IEEE Transactions on Vehicular Technology*, 63(4):1940–1945, 2014.

[8] X. Kang, Y.-K. Chia, and S. Sun. Mobile data offloading through a third-party wifi access point: An operator's perspective. *IEEE Transactions on Wireless Communications*, 13(10):5340–5351, 2014.

[9] J. Lee, Y. Yi, S. Chong, and Y. Jin. Economics of wifi offloading: trading delay for cellular capacity. *IEEE Transactions on Wireless Communications*, 13(3):1540–1554, 2014.

[10] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong. Mobile data offloading: how much can wifi deliver? *IEEE Transactions on Wireless Communications*, 21(2):536–550, 2010.

[11] Y. Li, M. Qian, D. Jin, P. Hui, Z. Wang, and S. Chen. Multiple mobile data offloading through delay tolerant networks. In *ACM workshop on Challenged networks*, 2011.

[12] S. Martello, D. Pisinger, , and P. Toth. Dynamic programming and strong bounds for the 0-1 knapsack problem. In *Management Science*, 1999.

[13] F. Mehmeti and T. Spyropoulos. Is it worth to be patient? analysis and optimization of delayed mobile data offloading. In *IEEE INFOCOM*, 2014.

[14] S. Miyazaki, N. Morimoto, and Y. Okabe. Approximability of two variants of multiple knapsack problems. In *Algorithms and Complexity*, pages 365–376. Springer, 2015.

[15] H. Park, Y. Jin, J. Yoon, and Y. Yi. On the economic effects of user-oriented delayed wi-fi offloading. In *IEEE WiOpt*, 2014.

[16] W. Peng, F. Li, X. Zou, and J. Wu. The virtue of patience: Offloading topical cellular content through opportunistic links. In *IEEE MASS*, 2013.

[17] X. Wang, M. Chen, Z. Han, D. Wu, and T. Kwon. Toss: traffic offloading by social network service-based opportunistic sharing in mobile social networks. In *IEEE INFOCOM*, 2014.

[18] L. Xiang, S. Ye, Y. Feng, B. Li, and B. Li. Ready, set, go: Coalesced offloading from mobile devices to the cloud. In *IEEE INFOCOM*, 2014.

[19] O. B. Yetim and M. Martonosi. Dynamic adaptive techniques for learning application delay tolerance for mobile data offloading. In *IEEE INFOCOM*, 2015.

[20] X. Zhuo, W. Gao, G. Cao, and S. Hua. An incentive framework for cellular traffic offloading. *IEEE Transactions on Mobile Computing*, 13(3):541–555, 2014.