This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2912963, IEEE Transactions on Wireless Communications

Resource Allocation in Highly Dynamic Device-to-Device Communication: An Adaptive Set Multi-cover Approach

Xiang Li *Member, IEEE*, Tianyi Pan, Md Abdul Alim, and My T. Thai *Senior Member, IEEE*

***Abstract*—Device-to-device (D2D) communication has recently gained much attention for its potential to boost the capacity of cellular systems. D2D enables direct communication between devices while bypassing a base station (BS), hence decreasing the load of BSs and increasing the network throughput via spatial reuse of radio resources. However, the cellular system is highly dynamic, an optimal allocation plan of radio resource to D2D links at one time point can easily become suboptimal when devices move. Thus, to maximize spatial reuse in cellular systems, it is crucial to update the resource allocation adaptively to reflect the current system status. In this paper, we develop the first adaptive solution framework to the dynamic resource problem for maximizing spatial reuse. At the core of the framework, we present the two algorithms for the adaptive set multicover problem with approximation ratio f and $\log n$ respectively, where f is the frequency of the most frequent element and n is the total number of elements. Experimental results not only show that our solutions have a significant improvement in running time, compared with optimal or approximated offline methods, but also demonstrate their good performance through the resource usage, network throughput and other metrics.**

***Index Terms*—Approximation algorithms, Adaptive algorithms, Device-to-device communication.**

I. Introduction

In recent years, mobile data traffic has been rapidly growing due to the booming market of mobile devices such as tablets and smart phones. Globally, mobile data traffic will increase 6-fold per user between 2016 and 2021 [1], which exerts great pressure to modern day wireless networks and therefore draws attention to new technologies that can optimize the usage of rare radio resources. Particularly, device-to-device (D2D) communications operated on the licensed spectrum bands has gained much attention for its potential to boost the capacity of cellular systems [2]. Equipped with D2D, direct communication between devices is enabled while bypassing a base station (BS). D2D underlaying a cellular infrastructure can provide increase in system throughput, improve energy efficiency, decrease the load of BSs and guarantee a planned environment with licensed spectrum [3].

One fundamental problem in D2D communication is the allocation of spectral resources. In contrast to the conventional cellular network in which resource blocks (RBs) are dedicated to the devices, multiple D2D communication links may spatially reuse an RB. The overall network performance is thus greatly impacted by the resource allocation scheme. Therefore, how to optimally allocate the RBs to devices to maximize spatial reuse becomes an essential problem in D2D design.

The resource allocation problem has been studied in literature [4]–[13]. Some of the works assumed that D2D links cannot share RBs among themselves but only with a cellular link [4]–[7], which limit the full potential of spatial reuse. Most of the other works allow more flexible spatial reuse [8]–[11], however, either they cannot guarantee optimality via game theoretical approaches [8], [9] or the approaches are too complicated to be applied directly [10], [11] that preprocessing [10] or relaxation [11] are required. More importantly, all the approaches fail to deal with the highly dynamic mobile network: an allocation may soon become invalid once a change occurs in the cell. Due to mobility, devices may enter/leave the cell so that the D2D communication links requiring RBs can rapidly change overtime.

Therefore, it is natural to pursue a solution that can be efficiently adapted to the status quo in a practical resource allocation scenario, as solving the whole problem from scratch can be costly and resulting in a delayed allocation decision that may be improper if the devices moved. Among the literature mentioned above, the approach in [10] has the most potential to be further developed into an adaptive algorithm, as its core is a set multicover (SMC) problem, which is much simpler comparing with other algorithms, despite the complicated preprocessing process that identifies the maximal interference-free sets (MIFS). Each MIFS is a subset of D2D links that can communicate concurrently using the same RB without having much interference. The MIFSs are used as sets in the SMC problem while the links corresponds to the elements. Kuhnle et al. [12] proposed an online algorithm that can partially handle the scenario when the links are established or ended. The algorithm guarantees a competitive ratio when links can only appear at fixed locations (hence the MIFSs are fixed) and the D2D links appear one-by-one. However, the setting is still not close to reality as when devices move, the MIFSs can easily change. Also, the algorithm has no guarantee when the links are ended.

In order to solve the practical resource allocation problems, we have to face the challenges resulted from mobility. As mentioned above, the D2D links and the resource requirements can change overtime. Thus, in order to apply the SMC based

The first two authors contributed equally to this paper.

Xiang Li is with the Computer Engineering Department, Santa Clara University, Santa Clara, CA, 95050.

Tianyi Pan, Md Abdul Alim and My T. Thai are with the Department of Computer & Information Science & Engineering, University of Florida, Gainesville, FL, 32611.

Email: xli8@scu.edu, { tianyi,alim,mythai}@cise.ufl.edu

approach, the MIFSs must be updated dynamically, which is complicated as the existing technique [10] requires enumerating all maximal independent sets in the interference graph based on all devices. If we allow updating the MIFSs with each change, the time complexity will be too high. However, if we cannot update the MIFSs, the allocation may not even be feasible as possibly some D2D links are not considered in all MIFSs. More importantly, even if the MIFSs are available after each system change, no solution exists for the dynamic SMC problem in which both set/elements can arrive/depart over time.

To tackle all the challenges, we first revisit the creation of MIFSs and propose an approach that the generated MIFSs can be kept fixed without impacting the solution quality. This approach solves the mobility issue related to MIFS generation and lays a solid foundation for the SMC based resource allocation approaches. Considering the fact that changes in a cellular network can be too frequent that we need to handle multiple changes instead of one when doing reallocation, we propose an adaptive approximation algorithm with f approximation ratio for the SMC problem that can deal with batches of changes, unlike the online algorithms that need to handle changes iteratively. Our adaptive approach has the merit of online approaches that only the changes in the system are required to update the solution, which needs much less running time than recomputing the whole problem. Also, it is capable of handling all kinds of dynamics in the system, including devices entering, leaving and moving within the cell. As an alternative, we also propose a $\log n$ adaptive approximation algorithm. Both algorithms are the first of their kind.

The contributions of this paper are as follows.

- We provide an adaptive framework for the D2D resource allocation problem that is able to efficiently update the resource allocation with batch changes, including devices entering, leaving and moving within the system.
- We propose an f-adaptive approximation algorithm for ASMC. To the best of our knowledge, this is the first adaptive algorithm with f ratio for adaptive set multicover problem that considers both elements entering/leaving.
- We propose the first $\log n$-adaptive approximation algorithm for ASMC under the same conditions.
- We run extensive experiments on cellular systems generated with both actual and simulated mobility traces. The proposed algorithms are scales of magnitude faster than the optimal solution. They also have comparable performance with the optimal solution, in terms of metrics like number of RBs used and the network throughput.

The rest of this paper is organized as follows. We first discuss about the related works in Section II. Next, we define the model of cellular system, discuss D2D interference and define the adaptive resource allocation problem in Section III. In Section IV, we propose the solution framework, approaches to calculate stable MIFSs and our two algorithms to solve the ASMC problem. The experiment results are shown in Section V. Finally, Section VI concludes the paper.

II. Related Work

The idea of D2D communication as an underlay of the cellular network was introduced in [2]. Control of interference among D2D links within a single cell was studied in [14], [15]. A resource sharing problem was proposed in [16] and solved using a game theoretical approach. Resource sharing problems in D2D or vehicular networks were studied in [17]–[19] via graph theoretic approaches. In [20], the problem of maximizing the spatial reuse for D2D communication has been presented. The problem was formulated in MIP and solved via a greedy heuristic. The results of [20] were later improved in [10] by using a greedy set multicover algorithm. Noticing the dynamic nature of D2D communication, online algorithms was proposed in [12], [13], which achieved a huge improvement in running time compared to algorithm in [10] while maintaining similar performance. However, the existing set multicover based solutions all assumed a fixed set of MIFSs, which limits their applicability since the future use cases of wireless communications in the 5G-era can be of high mobility [21]–[23].

From the theoretical side, the approximation algorithms for the set cover related problem were also proposed. An f-approximation algorithm for the set multicover problem was presented in [24]. Its approximation ratio was proved by the primal-dual schema. However, this algorithm was designed to solve a single SMC instance, but not an online/adaptive solution that can work under dynamic situation. For the set cover problem, various online algorithms exist in literature [25]–[28]. Among the algorithms, [25], [28] considered both addition and removal of elements while the others focused on adding elements one by one. Nonetheless, the proposed algorithms are not readily extended to the SMC problem, nor the tailored instance in the D2D resource allocation context.

III. Model and Problem Definition

A. The Cellular Network

We study a cellular system with a single BS B and set of RBs $\mathcal{R}$. To represent the system dynamics and device mobility, we denote $\mathcal{G} = (G^0, G^1, ..., G^T)$ as the system snapshots, where $G^t = (V^t, L^t, Q^t)$ with V^t, L^t as the set of all devices and the set of D2D links at time $t \in [0, T]$, respectively. The resource requirements for the links are denoted by $Q^t : L^t \rightarrow \mathbb{N}^+$.[1] For each link $l \in L^t$, its resource requirement is $Q^t(l)$, the minimum number of RBs required for the link at time t. We assume the set V^t of devices to serve is determined by the BS. Further, we assume the knowledge of c_{min}, the minimum allowable channel quality indicator (CQI), which in turn defines the minimum data rate r_{min} that any D2D link can gain from an RB. The set L^t and resource requirements Q^t are then determined by the BS based on the location of the devices, the content requirements/availability and r_{min}. We consider V^t, L^t and Q^t as inputs and the determination of the devices, links and requirements are beyond the scope of this paper.

[1] $\mathbb{N}^+$ is the set of positive integers.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2912963, IEEE Transactions on Wireless Communications

Cellular Resources. In this paper, we consider the resource sharing model discussed in [4], [11], [29]. In the model, the D2D and cellular links use disjoint portion of the licensed band. Therefore, interference only exists among D2D links and we mainly focus on the resource allocation problem for the D2D links. However, we will also discuss how our model and approaches can be adapted to solve the problem when cellular and D2D links can share RBs.

Data Rates. For a D2D link $(j,k) \in L^t$, we need to consider interference from other D2D links when calculating data rate $r(j,k)$. Denote $\mathcal{L}$ as the set of D2D links that shares an RB with (j,k), we can calculate $r(j,k)$ under distance-dependent path loss, multipath Rayleigh fading and log-normal shadowing using (1). W^d is the portion of band assigned to D2D links and $\gamma(j,k)$ is the Signal to Interference and Noise Ratio (SINR). In $\gamma(j,k)$, d_{jk} is the distance between devices j and k, α is the path loss exponent, $|m_0|^2$ is the fading component and is a constant within the BSs coverage area following [11], N_0 is the additive white Gaussian noise, ψ is the log-normal shadowing component and $p_j, p_{j'}$ are the transmit powers for UEs j, j' respectively.

$$r(j,k) = W^d \log_2(1+\gamma(j,k)) \tag{1}$$

$$\gamma(j,k) = \frac{p_j d_{jk}^{-\alpha}|m_0|^2 10^{\psi/10}}{\sum_{(j',k')\in\mathcal{L}} p_{j'} d_{j'k}^{-\alpha}|m_0|^2 10^{\psi/10} + N_0} \tag{2}$$

If we want to consider resource sharing between cellular and D2D links (in uplink) and obtain their corresponding data rates, we can simply add the interference between cellular and D2D links to the denominator of (2) (and also change the numerator for cellular link data rate calculation). This model can also be extended to handle multiple BSs by adding intercell interference to the denominator part of (2).

B. Problem Definition

There are two major ways to define the resource allocation problem, 1) minimize number of RBs while satisfying all resource requirements and 2) maximize network throughput with a fixed number of RBs. We choose to proceed with the minimization objective: at each time point t, the BS needs to determine the allocation of RBs to the D2D links: $F^t(l) \subseteq \mathcal{R}.\forall l \in L^t$, so that the number of RBs used, $|\cap_{l\in L^t} F^t(l)|$, is minimized while all the resource requirements are satisfied: $|F^t(l)| \geq Q^t(l), \forall l \in L^t$. The main reason for this choice is our goal to develop an adaptive approach to avoid extensive recomputing, in response to a highly dynamic environment. If the objective is maximizing network throughput, at each time point, all the RBs are allocated to the current D2D links. When new D2D links arrive, we have to re-allocate the RBs and likely recomputing from scratch, which is not desirable. Also, the D2D links will experience huge fluctuation in service. Instead, with the minimization objective, we satisfy all the requirements of current D2D links and maintain the largest number of free RBs that can serve future D2D links. When new D2D links arrive, we can allocate those RBs to the new links without having much impact to the service to old D2D links. The objective also facilitates an adaptive approach. The solution at time point t should be based on solution at time point $t-1$ and the changes in the network from G^{t-1} to G^t, denoted as Δ^t, rather than calculating from scratch. We assume that $\mathcal{R}$ is large enough to accommodate all resource requirements.

The definition of our problem, Adaptive Mobility-Aware Resource Allocation (AMARA), is as follows.

Definition 1 (AMARA)**.** *Given a dynamic cellular system $\mathcal{G} = (G^0, G^1, ..., G^T)$ where $G^t = (V^t, L^t, Q^t)$, the set of RBs $\mathcal{R}$, the problem is to find the allocation $F^t(l) \subseteq \mathcal{R}.\forall l \in L^t$, so that $|F^t(l) \geq |Q^t(l), \forall l \in L^t$ and the number of used RBs, $|\cap_{l\in L^t} F^t(l)|$, is minimized. Also, F^t should be derived adaptively, using only F^{t-1} and Δ^t.*

IV. Solution

A. Overview

For solving AMARA, based on the previous analysis, seemingly we have to face the two aforementioned challenges: dynamically updating MIFSs and solving the SMC problem adaptively, considering element/set arrivals/departures. However, the challenges are based on the fact that the MIFSs are generated as in [10], [12], [13]. To deal with this challenge, we will derive a new approach of generating MIFSs in Sect. IV-B that major updates are not necessary no matter how the devices move. Thus, the second challenge is simplified to an adaptive SMC problem with only element arrivals/departures. We propose two variations of its approximation solution in Sect. IV-C and Sect. IV-D respectively, which are first such algorithms and also completes the solution to AMARA. The overview of solution to AMARA is as in Alg. 1.

Algorithm 1 Solution to AMARA

Input: G^0, Δ^t, $0 < t \leq T$
Output: F^t, $0 < t \leq T$
Calculate all MIFSs as discussed in Sect. IV-B, denote the sets as $\mathcal{S}$.
for $t = 0$ to T **do**
Solve the adaptive SMC problem defined in Sect. IV-B to obtain F^t using the algorithm in Sect. IV-C or Sect. IV-D.

B. Calculation of MIFS

In [10], [12], [13], MIFSs are for the links, defined as the set of D2D links that can share the same resource block without introducing much interference. Therefore, whenever a device moves, the MIFSs must be updated accordingly as the interference for links in all MIFSs related to the device will change. As calculating the MIFSs is an enumerative process hence time-consuming, it is preferable to do it only once. Thus, we avoid using the links for constructing MIFSs. Instead, we rely on the stable components of the cell: locations.

The main idea of this approach is to split the area covered by the BS as grids and place an artificial link at the center of each grid. Then, we map all links and their requirements to the grids by proximity and use the corresponding artificial links to represent the actual links. Each artificial link has both

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2912963, IEEE Transactions on Wireless Communications

its transmitter and receiver located at the center of the grid, yet the distance between them is a positive constant d_0. The RB requirement of an artificial link is the sum of all RB requirements of actual links mapped to the grid. To obtain an MIFS, we can iteratively pick links until the minimum SINR among those links sharing a single RB is close to γ_{min}, which we set to 15 dB in this paper. Note that we consider distance between links, path loss exponent, fading component and shadowing component in MIFS calculation as we calculate SINR using equation (2). After the mapping, we can obtain an altered instance of AMARA that has a fixed universal set of artificial links. Those links may arrive, depart or have varying resource requirements overtime, based on the dynamics of the actual links. The calculated MIFSs are stable as the grids are fixed. By this approach, it is not necessary to update the MIFSs, only with a minor overhead of mapping the links to grids at each time t.

In order to ensure that each actual link can be close to the center of some grid, we use overlapping grids. Specifically, denote the maximum D2D distance as d, we introduce square grids (we also assume a square cell) with size $2d \times 2d$ and place one such grid by its center at coordinates $(\frac{pd}{2}, \frac{qd}{2}), p, q \in \mathbb{Z}$ (and the grids are within the cell). The following lemmas gives bounds on the approximation from actual links to the artificial links. Since the results are solely based on the center of each grid, they can be applied to arbitrary grid shapes.

Lemma 1. *For a link whose transmitter and receiver are both at least d away from the boundary of the cell, the maximum distance from its midpoint to the center of its assigned grid is at most* $\frac{\sqrt{2}}{4}d$.

Proof. Despite the boundary grids, each grid will obtain all links whose midpoints fall into the square with side length $\frac{d}{2}$, centered at the grid center. This is due to the way we select the grid centers. Hence, the maximum distance is from the vertex of the square to the center, which is $\frac{\sqrt{2}}{4}d$. □

Lemma 2. *Given two links* $(t_1, r_1), (t_2, r_2)$ *assigned to the same grid, let* $dist(t, r)$ *be the distance between the devices, we have*

$$\max(dist(t_1, r_2), dist(t_2, r_1)) = \frac{\sqrt{6}}{2}d.$$

Proof. Based on Lemma 1, the two links both fall into the square with side length $\frac{d}{2}$. Hence, the center of the two links must be at two end points of a diagonal of the square or the distance cannot be maximum. In order to achieve maximum distance, the links should both be orthogonal to the diagonal, as depicted in Fig. 1 and the distance in this case is $\frac{\sqrt{6}}{2}d$. □

Lemma 3. *Given two links* $(t_1, r_1), (t_2, r_2)$ *assigned to two different grids, we have*

$$\min(dist(t_1, r_2), dist(t_2, r_1)) \geq 2d.$$

when the centers of the two grids are at least $(3 + \frac{\sqrt{2}}{2})d$ *away.*

Proof. The maximum distance from the center of a grid to a transmitter/receiver whose link is assigned to the grid is $d_m = (\frac{\sqrt{2}}{4} + \frac{1}{2})d$, which is achieved when the center of the link is at one vertex of the square and the link is collinear with the diagonal, as in Fig. 2. Hence, to guarantee a minimum distance of $2d$ between transmitter and receiver from different links, the centers of the corresponding grids must be $2d+2\times d_m$ away, which yields the result. □

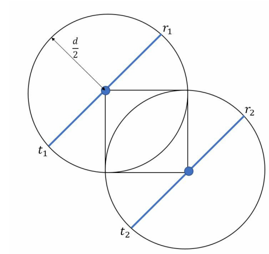


Fig. 1: Example for Lemma 2

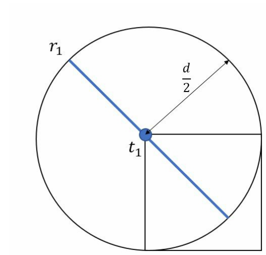


Fig. 2: Example for Lemma 3

The Lemmas ensures that using artificial links give a good approximation of the actual links. Lemma 1 shows the proximity between the actual and artificial links. Lemma 2 makes sure that the links assigned to the same grid are close enough so that no resource sharing is possible among them. From another perspective, Lemma 3 guarantees that the links assigned to different grids will not be too close, so that there exists limited impact from interference when we allocate the same RB to those links. We will illustrate the performance of link-grid mapping in the experiments, especially in Fig. 7.

Allowing variation in transmit power. In the above description of MIFS calculation, we assume a fixed transmit power for all links. Variation in transmit power can be allowed by adding copies of each artificial link with different transmit power levels. Then, an MIFS can contain links with different transmit power. However, the cost is the increased complexity for enumerating MIFSs.

Resource sharing among cellular links and D2D links. In order to consider more generalized resource sharing, we need to calculate the interference at the BS, as we only share RBs in uplink. If we want to include a cellular link into a MIFS, we can map the BS as a special grid and map the cellular transmitters as the D2D links. Then the SINRs can be calculated accordingly.

For clarity, we also define the adaptive SMC (ASMC) problem here.

Definition 2 (ASMC)**.** *Given a universal set* $\mathcal{E}$, *a collection of sets* $\mathcal{S} = \{S_1, S_2, ..., S_m\}$ *and a collection of sets of elements* $(E^0, E^1, ..., E^T)$, *where* $E^t \subseteq \mathcal{E}$ *contains the elements having coverage requirements at time point t. The coverage*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2912963, IEEE Transactions on Wireless Communications

requirement of element $e \in E^t$ is denoted as $k_e \in N^+$. The problem is to adaptively find a set multicover $C^t \subseteq \mathcal{S}$ with the minimum size so as to satisfy all the coverage requirements at every time point t.

Based on the mapping, each element corresponds to a grid and each set corresponds to an MIFS. Then, $\mathcal{E}$ is to the set of grids, and $\mathcal{S}$ is the set of all MIFSs. Elements in E^t are those grids containing D2D links at time t, the coverage requirement is the summation of requirements of all links in the grid. We can use a simple summation as the links grouped into the same grid are close, they are not able to share RBs due to high interference. To obtain the allocation F^t, we can assign an RB to each selected MIFS in C^t and they can be easily mapped to the links in each grid.

C. f-Approximation to ASMC

Our first approach to solve ASMC is based on the primal-dual framework, which guarantees an f approximation ratio for the SMC problem without any changes [24], where f is the maximum frequency of elements among all sets in $\mathcal{S}$. In order to solve the problem adaptively, however, we have to carefully design procedures to handle updates in order to maintain the f ratio. In the following, we first overview the offline algorithm for SMC with f ratio and then present our adaptive algorithm.

1) Algorithm for SMC: As we have to solve the whole problem at time $t = 0$ since there is no prior information, we need the algorithm **Base_Alg** for the offline version of SMC. We can use any existing solutions to SMC for this base case. However, we want a solution that can pick an arbitrary element to cover in each iteration, so that the later snapshots can be solved adaptively. Here we briefly overview the algorithm in [24]. For notational convenience, we introduce the IP formulations and also the details of the **Base_Alg** in Alg. 2.

We denote the selected set multicover be C and let $S_j \in \mathcal{S}$ denote an arbitrary set. Denote x_j as a binary variable for each set such that $x_j = 1$ if $S_j \in C$ and $x_j = 0$ otherwise. The IP formulation for SMC is as follows. The objective (3) makes sure the number of selected sets in the multicover C is minimized, and constraint (4) ensures each element i is covered for at least k_i times. Constraint (5) guarantees the x_js are binary.

$$\min \quad \textstyle\sum_{j=1}^{m} x_j \tag{3}$$
$$s.t. \quad \textstyle\sum_{j:i \in S_j} x_j \geq k_i \quad \forall i \in \{1, \ldots, n\} \tag{4}$$
$$x_j \in \{0, 1\} \quad \forall j \in \{1, \ldots, m\} \tag{5}$$

The corresponding LP relaxation $\mathcal{P}$:

$$\min \quad \textstyle\sum_{j=1}^{m} x_j \tag{6}$$
$$s.t. \quad \textstyle\sum_{j:i \in S_j} x_j \geq k_i \quad \forall i \in \{1, \ldots, n\} \tag{7}$$
$$-x_j \geq -1 \quad \forall j \in \{1, \ldots, m\} \tag{8}$$
$$x_j \geq 0 \quad \forall j \in \{1, \ldots, m\} \tag{9}$$

The dual program $\mathcal{D}$, in which y_i corresponds to constraint (7), z_j corresponds to constraint (8).

$$\max \quad \textstyle\sum_{i=1}^{n} k_i y_i - \sum_{j=1}^{m} z_j \tag{10}$$
$$s.t. \quad \textstyle\sum_{i \in S_j} y_i - z_j \leq 1 \quad \forall j \in \{1, \ldots, m\} \tag{11}$$
$$y_i \geq 0 \quad \forall i \in \{1, \ldots, n\} \tag{12}$$
$$z_j \geq 0 \quad \forall j \in \{1, \ldots, m\} \tag{13}$$

The following complementary slackness conditions are necessary for the f-approximation under primal-dual schema.

$$\forall j \in [1, m]\text{: either } x_j = 0 \text{ or } \sum_{i \in S_j} y_i - z_j = 1 \tag{14}$$
$$\forall i \in [1, n]\text{: either } y_i = 0 \text{ or } \sum_{j:i \in S_j} x_j \leq f \tag{15}$$
$$\forall j \in [1, m]\text{: either } z_j = 0 \text{ or } x_j = 1 \tag{16}$$

Also an additional condition:

$$\sum_{j=1,\ldots,m} z_j \leq \sum_{i=1,\ldots,n} (k_i - 1) y_i \tag{17}$$

Lemma 4. *[24] The primal solution* **x** *is a f-approximation to SMC if the corresponding dual solution* **y, z** *is feasible and conditions* (14)-(17) *hold.*

We summarize the algorithm in [24] as **Base_Alg**. This algorithm works as follows. It starts from a dual feasible solution **y, z** = **0** and a primal infeasible solution **x** = **0**. Each time an element $e_i \in \mathcal{E}$ whose coverage requirement is not met is picked and y_i is set to 1. Then z_j is modified for all sets to maintain primal complementary slackness conditions. All sets that can cover element e_i are included in the result C if they are not included yet. The algorithm stops when all coverage requirements are satisfied by the set C.

Algorithm 2 Base_Alg

Input: $\mathcal{E}$, $\mathcal{S}$
Output: C
1: $x_j = 0, z_j = 0, \quad \forall S_j \in \mathcal{S}$
2: $y_i = 0, \quad \forall e_i \in \mathcal{E}$, $A = \mathcal{E}$
3: **while** A is not empty **do**
4: Arbitrarily pick $e_i \in A$, Set $y_i = 1$
5: **for** $\forall S_j : i \in S_j$ **do**
6: **if** $S_j \notin C$ **then**
7: $x_j = 1$, $C = C \cup \{S_j\}$
8: Set $z_j = \sum_{i \in S_j} y_i - 1$
9: **for** $e_l \in S_j$ **do**
10: **if** $\sum_{j:l \in S_j} x_j \geq k_l$ **then**
11: Remove e_l from A

Theorem 1. *[24] Algorithm 2 is an f-approximation algorithm for SMC.*

Theorem 2. *Algorithm 2 has a time complexity of $O(f|\mathcal{E}|^2)$ where $f = \max_i |\{S_j | i \in S_j\}|$.*

Proof. The initialization of Algorithm 2 takes $O(|\mathcal{S}| + |\mathcal{E}|)$ time. The while loop in line 3 is executed $O(|\mathcal{E}|)$ times. The for loop in line 5 runs $O(f)$ times with each element picked

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2912963, IEEE Transactions on Wireless Communications

in line 5. The time for line 9 can go up to $O(|\mathcal{E}|)$. The time complexity follows as $|\mathcal{S}| = O(f|\mathcal{E}|)$. □

2) The Adaptive Approximation Algorithm: Based on Lemma 4, if we can maintain a solution that satisfies conditions (14)-(17), the f approximation ratio naturally holds. Thus, the goal of the adaptive algorithm is then efficiently update the primal/dual solutions so that the conditions are satisfied.

We first consider ΔE^t, the changes in the system at time t. The changes can be classified into four categories: addition/removal of elements and increase/decrease of requirements for existing elements. Each change $\delta \in \Delta E^t$ is a tuple $\delta = (TYPE, e, c)$. $TYPE$ can be $ADD/RM/INC/DEC$, which specifies the four types of changes. e is the element associated with the change and c is the amount of the change, c is positive for ADD/INC and negative for RM/DEC. The changes are handled differently based on their types in the following algorithm.

Adaptive_Alg. (Alg. 3) In this algorithm, we first call function **Remove_Elem** for removed nodes and update C^t and ΔE^t accordingly. Then we call **Partial_Cover** to satisfy all requirements in ΔE^t.

Algorithm 3 Adaptive_Alg

Input: ΔE^t,C^{t-1}, $\mathcal{S}$
Output: C^t
$C^t = C^{t-1}$
for all $\delta = (TYPE, e, c) \in \Delta E^t$ **do**
  if $TYPE = RM$ **then**
    C^t = **Remove_Elem**(δ, C^t, ΔE^t)
C^t = **Partial_Cover**(ΔE^t, C^t)

Remove_Elem. (Alg. 4) The algorithm only considers the elements that are removed. For a removed element e, we check all sets S_j such that $e \in S_j$. If $\sum_{e_i \in S_j} y_i - z_j = 0$, we first try to recover this complementary slackness condition by decreasing z_j. If z_j is already 0, we need to remove S_j from C^t and set $x_j = 0$. Also, we check all the other elements covered by S_j. If some elements need more coverage because of removal of S_j, we add the change to ΔE^t. Notice that we may add multiple records of the same element to ΔE^t. In line 10 of Alg. 4, we can search for the element in ΔE^t before addition. If the element is already in ΔE^t, we can integrate the new change with the existing one.

Partial_Cover. (Alg. 5) This algorithm takes ΔE^t and considers only requirement increases. We ensure this by handling all element removal in Alg. 4 and filter out all requirement decreases at the beginning of Alg. 5. Then the requirement increases in ΔE^t are transformed to an SMC with smaller size than the original problem. The remaining structure is similar to **Base_Alg**.

We are now ready to prove the approximation ratio of **Approx-ASMC**.

Lemma 5. ***Remove_Elem*** *keeps dual feasibility as well as conditions* (14)-(17).

Algorithm 4 Remove_Elem

Input: $\delta = (TYPE, e, c)$, C^t, ΔE^t, E^t
Output: C^t, ΔE^t
1: Remove all records of e from ΔE^t
2: **if** $y_e = 1$ **then**
3:   $y_e = 0$
4:   **for** $\forall S_j : e \in S_j$ **do**
5:     **if** $\sum_{e_i \in S_j} y_i - z_j = 0$ **then**
6:       **if** $z_j > 0$ **then**
7:         $z_j = z_j - 1$
8:       **else**
9:         Remove S_j from C^t, $x_j = 0$
10:         **for all** $e_i \in S_j$ **do**
11:           **if** $\sum_{S_p : e_i \in S_p} x_p < k_{e_i}$ AND $e_i \in E^t$ **then**
12:             Let $\delta' = (INC, e_i, 1)$, Add δ' to ΔE^t

Algorithm 5 Partial_Cover.

Input: ΔE^t, C^t, $\mathcal{S}$
Output: C^t
Let $A = \emptyset$
for $\forall \delta = (TYPE, e, c) \in \Delta E^t$ **do**
  if $TYPE = ADD$ **then**
    Add e to A, $y_e = 0$
  else if $TYPE = INC$ **then**
    Add e to A
while A is not empty **do**
  Arbitrarily pick $e_i \in A$, Set $y_i = 1$
  for all $S_j : i \in S_j$ **do**
    if $S_j \notin C^t$ **then**
      $x_j = 1$, $C^t = C^t \cup \{S_j\}$
    Set $z_j = \sum_{i \in S_j} y_i - 1$
    for all $e_l \in S_j$ **do**
      if $\sum_{j : l \in S_j} x_j \geq k_l$ **then**
        Remove e_l from A

Proof. It is trivial to remove an element e_i with $y_i = 0$. The dual constraints and the conditions will not change. When we remove an element e_i with $y_i = 1$, notice that changing y_i to 0 will not impact dual feasibility. Condition (14) is maintained by line 5-7 of Alg. 4. Condition (16) is maintained by line 8-9. Condition (15) always holds.

For condition (17), we need a more detailed analysis of how the difference of rhs and lhs of (17) changes when changing y_i in Alg. 2. We group the collection of sets that can cover e_i into two subcollections, $\mathcal{S}_i^1, \mathcal{S}_i^2$ where $\mathcal{S}_i^1$ is the collection of sets that $\forall S \in \mathcal{S}_i^1, S \cap E^t = \{e_i\}$ (the sets that can only cover e_i in the current set of elements) and $\mathcal{S}_i^2$ contains all the remaining sets that can cover e_i. Denote $|\mathcal{S}_i^1| = a_i^1$ and $|\mathcal{S}_i^2| = a_i^2$, we have $a_i^1 + a_i^2 \geq k_i$ when the problem is feasible. Clearly, when element i is removed and we set $y_i = 0$, the lhs of (17) decreases by a_i^2 as z_j values for those in $\mathcal{S}_i^1$ are 0. When $a_i^2 \geq k_i - 1$, (17) trivially holds as the rhs will decrease by $k_i - 1 \leq a_i^2$. If $a_i^2 < k_i - 1$, we can write $a_i^2 = k_i - 1 - b_i, b_i > 0$. Consider the time When y_i is set to 1 in Alg. 2 or Alg. 5, the lhs of (17) is increased by at most $k_i - 1 - b_i$ and the rhs is increased by $k_i - 1$. Therefore, the gap between the two

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2912963, IEEE Transactions on Wireless Communications

sides increased by at least b_i by this step. When setting $y_i = 0$, however, the gap between two sides will decrease by b_i. Since the two facts hold for all e_i, we can see the b_i increase in gap when setting $y_i = 1$ as the budget to pay the b_i decrease when we set $y_i = 0$. Thus, the feasibility of (17) is always ensured. □

Lemma 6. ***Partial_Cover*** *yields a primal feasible solution at any time point t for* E^t.

Proof. We prove this by induction. At time 0, Theorem 1 shows that **Base_Alg** can give a feasible solution. At time t, before we call **Adaptive_Alg**, assume SMC^{t-1} is a feasible cover for E^{t-1}. After all runs of **Remove_Element** at time t, we may remove some sets from SMC^{t-1}. The elements that used to be covered by those sets are added to ΔE^t if more coverages are required. Therefore, what **Partial_Cover** handles are all elements with unsatisfied coverage requirements in E^t. Based on the same reasoning as in Theorem 1, **Partial_Cover** can satisfy all those requirements and then yield a primal feasible solution. □

Theorem 3. *At any time point t in the dynamic network, the set multicover* C^t *found by* ***Approx-ASMC*** *satisfies* $C^t \leq fC^t_{opt}$ *where* f *is the maximum frequency of any elements in* $\mathcal{S}$ *and* C^t_{opt} *is the optimal solution at time t. Thus,* ***Approx-ASMC*** *is an f-approximation algorithm for ASMC.*

Proof. To prove the f-approximation ratio, we need to show conditions (14)-(17) are maintained in **Partial_Cover** and the resulting primal/dual solutions are feasible. Based on Lemma 6, the primal solution x_j and dual solution y_i, z_j before running **Partial_Cover** satisfy all conditions and the dual solution is feasible. Since the main execution part of **Partial_Cover** is the same as **Base_Alg**, we can use the same reasoning in Theorem 1 to show the feasibility of the dual solution and all conditions hold. From Lemma 6, primal feasibility is also ensured. Therefore, we get the desired f ratio for **Approx-ASMC** based on primal-dual schema. □

In the remaining part of this section, we present the analysis of time complexity of **Approx-ASMC**. In Theorem 2, we already proved the time complexity of **Base_Alg**, which is $O(f|E^{(0)}|^2)$.

The time complexity of **Approx-Adaptive** can be separated to two parts: **Remove_Elem** and **Partial_Cover**. For **Remove_Elem**, we first consider its worst case complexity:

Lemma 7. *The worst case time complexity for* ***Remove_Elem*** *is* $O(f_{\Delta E^t}|E^t|)$. *Where* $f_{\Delta E^t}$ *is the maximum frequency of all elements in* ΔE^t. *Also,* $O(|E^t|)$ *elements can be added to* ΔE^t *in the worst case.*

Proof. The worst case happens when we need to remove every set covers the element e from SMC^t and each of the removed sets are of size $O(|E^t|)$. In this case, all the $O(|E^t|)$ elements in those sets will be added to ΔE^t. The time complexity for Alg. 4 is then $O(f_{\Delta E^t}|E^t|)$. □

Since the time complexity of Alg. 4 can be as good as $O(f_{\Delta E^t})$ and not adding any elements to ΔE^t, we will apply amortized analysis to better depict the overall behavior of Alg. 4.

Lemma 8. *Alg. 4 has an amortized time complexity of* $O(f_{\Delta E^t})$ *and it adds* $O(f_{\Delta E^t})$ *elements to* ΔE^t.

Proof. Each time an element e is removed, the z_j values of at most $f_{\Delta E^t}$ sets covering e are checked. Notice that handling each of the sets S_j needs $O(1)$ time when $z_j > 0$ and $O(|S_j|)$ time when $z_j = 0$. When the case $z_j = 0$ is met and S_j has to be removed, there must exist $|S_j| - 1$ operations of complexity $O(1)$ to decrement z_j. Therefore, the amortized time to check a set is $O(1)$ and $O(1)$ elements are added to ΔE^t per check. Thus the lemma follows as Alg. 4 checks at most $f_{\Delta E^t}$ sets. □

Theorem 4. *Algorithm* ***Approx-Adaptive*** *has amortized time complexity of* $O(f^3_{\Delta E^t}|\Delta E^t|^2)$.

Proof. Since we run Alg. 4 $O(|\Delta E^t|)$ times, based on Lemma 8, the total amortized time complexity of all runs of Alg. 4 operations in Alg.3 is $O(f_{\Delta E^t}|\Delta E^t|)$ and the size of ΔE^t can at most change by a factor of $f_{\Delta E^t}$.

For Alg. 5, the time complexity analysis is the same as Alg.2. Therefore, setting $|\mathcal{E}| = f_{\Delta E^t}|\Delta E^t|$, Alg. 5 has time complexity of $O(f^3_{\Delta E^t}|\Delta E^t|^2)$.

Thus the overall time complexity of Alg.3 is $O(f^3_{\Delta E^t}|\Delta E^t|^2)$. □

Notice that the time complexity of **Adaptive_Alg** does not include any global parameters such as E^t or $\mathcal{S}$, which means our algorithm to ASMC has a much better time complexity than offline algorithms.

3) Solution to ASMC: Based on Alg. 2 and 3, we can build the solution for ASMC as **Approx-ASMC** in Alg. 6. The algorithm calls **Base_Alg** in initialization at time $t = 0$, in order to get the initial set multicover $C^{(0)}$. At time point $t > 0$, **Approx-ASMC** calls algorithm **Adaptive_Alg** to calculate C^t based on C^{t-1} and ΔE^t.

Algorithm 6 Adaptive Approximation Algorithm for ASMC (**Approx-ASMC**)

Input: E^0, ΔE^t, $0 < t \leq T$, $\mathcal{S}$
Output: C^t, $0 \leq t \leq T$
  C^0 = **Base_Alg**$(E^0, \mathcal{S})$
  for $t = 1$ to T **do**
    C^t =**Adaptive_Alg** $(\Delta E^t, \mathcal{S}, C^{t-1})$

From Theorem 1 and 3, Alg. 6 keeps an f adaptive approximation ratio which is one of the best ratios for offline set multicover problem [30]. Therefore, Alg. 6 can achieve one of the best ratios with improved time complexity in an adaptive setting, which is highly desirable.

D. A $\log n$*-approximation Algorithm*

A recent paper [28] introduced the first $O(\log n)$-competitive solution to the online set cover problem. In this section, we briefly recap the existing solution in [28] and discuss how it can be extended to solve ASMC. Then, we refine the result by utilizing special features in the problem originated from D2D resource allocation.

1) The algorithm for online set cover: In [28], the authors introduced a $\log n$-competitive algorithm for online set cover, which has $O(f \log n)$ update time per element arrival. The main idea of the algorithm is to assign each element to a single set and place the sets in different levels based on the density of the set, which is defined as cost per covered element. The algorithm maintains a *stable solution* as the result. In a stable solution, the sets cannot be substituted by unselected sets to obtain lower density.

The approximation ratio of the algorithm is obtained from the fact that the majority of the density levels falls in the range of $[\frac{OPT}{n}, COPT]$, where C is a constant (2^{10} in [28]). Also, the total cost in each density level is $O(OPT)$. Since the size of density levels increases exponentially, the total number of non-trivial density levels is $O(\log n)$ and hence the total cost of all sets is $O(\log nOPT)$.

2) Adaptation and new results.: As each element may need to be covered multiple times, the definition of coverage and density in [28] no longer holds. In calculating the density of a collection of sets, instead of using the number of covered elements, we can use the number of effective covering times. For density calculation, it is equivalent to creating k_e copies for each element e. Then, total cost of sets in each density level is still $O(OPT)$. However, in ASMC, the lower bound on the density level is now $\frac{OPT}{\sum_{e\in\mathcal{E}} k_e}$ instead of $\frac{OPT}{n}$. Thus, the number of density levels, as well as the approximation ratio, is $O(\log(\sum_{e\in\mathcal{E}} k_e)) = O(\log n + \log f)$ and is $O(\log n)$ when all k_es are small.

For the update time, in [28], each element carries $O(\log n)$ credits to pay for level changes. In ASMC, however, all k_e copies of each element e may change levels, thus the amount of credits is now $O(k_e \log n)$. The complexity for each level change is still $O(f)$ and the overall update time is $O(f\bar{k}\log n)$ where $\bar{k}$ is the average requirement. When $\bar{k}$ is $O(1)$, the time complexity is $O(f \log n)$, the same as [28].

3) Refinement of the result.: In [28], the key structures that contribute to the $O(\log n)$ ratio are the $O(\log n)$ density levels, as the possible densities of the sets are lower bounded by $\frac{OPT}{n}$ and upper bounded by $2^{10}OPT$. However, the specific ASMC problem we consider in the D2D context has some intrinsic features that allows better analysis of the density levels. First of all, the set costs are uniform and thus we can use any arbitrary cost c for all the sets. Also, we keep the volume of each element at 1. Thus, in any solution, the highest possible density is then c, instead of the unknown value $2^{10}OPT$. Similarly, denote the cardinality of the largest set as s_{max}, then the lowest possible density is $\frac{c}{s_{max}}$. Notice that in general s_{max} is also unknown. However, in the D2D context, this number denotes the maximum number of devices that can transmit using the same RB, which can be approximated by the size of largest MIFS. If we set $c = s_{max}$, then the range of densities becomes $[1, s_{max}]$ and the number of levels is $\log(s_{max})$, a constant. Which means that we can obtain a constant ratio approximation algorithm under this specific setting.

V. Experiments

A. Experimental Settings

1) Algorithms: In the experiments, we compare the performance of the following algorithms. We do not consider other algorithms as they are not compatible with fully dynamic resource allocation scenarios.

- **f-adaptive**: the adaptive algorithm with f approximation ratio, described in Sect. IV-C.
- **f-offline**: the algorithm with f approximation ratio but solves the resource allocation problem from scratch for each snapshot.
- **$\log n$-adaptive**: the adaptive algorithm with $\log n$ approximation ratio, described in Sect. IV-D.
- **$\log n$-offline**: the algorithm with $\log n$ approximation ratio but solves the resource allocation problem from scratch for each snapshot.
- **optimal**: the algorithm that optimally solves the IP (3) - (5) for each snapshot.

2) Parameters: We focus on a single square cell in the experiments and the main network parameters are summarized in Table I. As we only consider the resource allocation of D2D links in this paper, we ignore all parameters related to the base station. Also, we do not limit the number of RBs we may use, but we set the bandwidth for each RB for data rate calculation.

TABLE I: Main Wireless Network Parameters

Notation	Description
Cell dimension	100 x 100 m^2
Channel Model	Multipath Rayleigh fading
Fading Component	1
Path Loss Exponent	3
Shadowing std. dev.	10 dB
Noise spectral density	−174 dBm/Hz
D2D transmit power	23 dBm
Maximum D2D distance	30 m
RB bandwidth	200 kHz

3) Datasets: As we focus on resource allocation problems overtime, only the datasets with mobility traces are considered in this paper. We first consider the CRAWDAD datasets [31], [32] that provides actual [31] and simulated [32] traces in real sites. We further consider mobility data generated by the SLAW model [33].

In the datasets, only the traces of the devices are provided, so we generate the D2D links using the following method. The steps for generating the links for one snapshot are detailed in Alg. 7. For each snapshot, we iteratively pick a random unselected device and construct a D2D link between it and its nearest unselected neighbor within D2D transmission range. At most one link is allowed per device at any point of time, so we mark the two nodes as selected once the link is established. We will also mark the randomly picked node if there exists no device within its D2D transmission range. We stop the generation when the number of generated links in that snapshot reaches the predefined upper bound B (set to 80 in our experiments), or when all nodes are selected. We also consider the duration of the D2D links. For each link, we randomly assign it a duration within 1 to 5 snapshots. We will

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2912963, IEEE Transactions on Wireless Communications

Algorithm 7 Link Generation

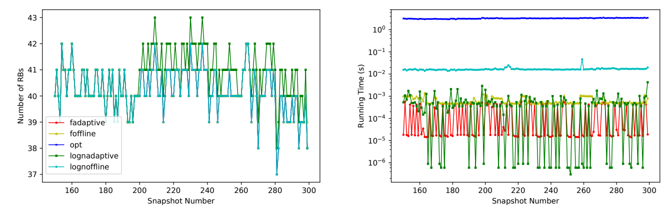
Input: V^t, L^{t-1}, B
Output: L^t
$L^t = \emptyset$
for $\forall l = (u, v) \in L^{t-1}$ **do**
  if $l.duration > 1$ and $distance(u, v) < d$ **then**
    Add l to L^t with duration $l.duration - 1$
    Mark u, v as selected
$count = 0$
while $count < B$ and exists some unselected nodes **do**
  Randomly pick an unselected $u_0 \in V^t$
  Find the unselected node $v_0 \in V^t$ nearest to u_0.
  if Both v_0 exists and $distance(u_0, v_0) < d$ **then**
    Uniformly randomly generate dur within $[1, 5]$
    Add (u_0, v_0) to L^t with duration dur
    Mark v_0 as selected
    $count+ = 1$
  Mark u_0 as selected

bring the link in link set L^{t-1} to L^t with one less duration if the two devices are still within D2D transmission range. This step is executed prior to new link generations, so it is possible that we have more links than the upper bound in some snapshots.

The size of the cells in the original datasets may be large. For illustration purpose, however, we set the cells to be $100 \times 100\ m^2$ squares and scale the traces correspondingly in order to increase the number of D2D links we may have, as the mobility traces in original datasets can be sparse, preventing us from generating the desired amount of D2D links. We also normalize the number of snapshots we consider. In the KAIST dataset, the time between two snapshots is 30 seconds. For the simulated datasets, the time difference is one second. In the experiments, we only use the results for the 151th to the 300th snapshots in all experiments. The first 150 snapshots are used as a "warm-up" period to smooth the initial fluctuations. We summarize the datasets in Table II. In the table, the average is taken over the 151th to the 300th snapshots and a change can be adding a new D2D link or removing an old D2D link. (A link that moves around can be modeled as the combination of one addition and one removal.)

B. Performance with Actual Mobility

We first consider the results with actual mobility traces in KAIST campus [31]. This dataset has the least amount of D2D links and very limited mobility. In Fig. 3, we can observe that all the algorithms outputs almost the same number of RBs (hence the results for some algorithms are not visible), demonstrating the accuracy of the approximation algorithms in this dataset. As for the running time, all approximation algorithms are scales of magnitude faster than the optimal solution. When comparing the adaptive algorithms and their corresponding offline algorithms, the adaptive ones are generally faster. It is notable that the f-approximation offline algorithm also has comparable running time with both adaptive algorithms. This phenomenon can be due to two reasons: 1) The size of the offline problem is moderate and it is easier to consider only



(a) KAIST - Performance (b) KAIST - Running Time

Fig. 3: Performance and running time with actual mobility traces

the addition of new links. 2) Maintaining the data structures and performing the link removal operation can be costly for both approximation algorithms.

C. Performance with Simulated Mobility

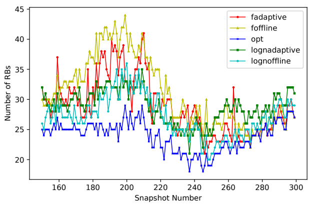
In this section, we present the results with three simulated mobility traces: the subway and downtown scenario from [32] and one dataset generated by the SLAW model [33] with default settings. The subway and downtown scenarios simulates the mobility traces of walkers in a subway station and in downtown Stockholm respectively, while the SLAW model simulates traces in common gathering places, so there's no restriction to where a walker can move within the range. In those datasets, the average numbers of D2D links are higher than that of the KAIST dataset. Also, those datasets are much more dynamic: they have more changes over time. Thus, when applying the algorithms, we can observe the obvious differences among them. In Fig. 4, there exists some performance gap between the optimal algorithm and the approximation ones. However, the approximation ratio of all approximation algorithms at all times are upper bounded by 2.3, demonstrating the good performance of them.

Among the approximation algorithms, The offline algorithm with $\log n$ theoretical approximation ratio constantly performs the best. The two adaptive algorithms have comparable performance, while, interestingly, the offline algorithm with f ratio uses the most number of RBs. We may conclude from the result that the two algorithms with $\log n$ ratio have more stable behavior, that starting from scratch grants some advantage in terms of practical performance (number of selected RBs). The advantage is from the fact that the offline algorithm only needs to consider addition of elements. So it solves an easier problem comparing with the adaptive one, which needs to consider both addition and removal of elements. On the contrary, the behavior of the algorithms with f ratio seems controversial, that the adaptive algorithm works better than the offline one. Yet, it can be explained as follows. The two primal-dual based f-ratio algorithms are "coarse" comparing with the $\log n$ ratio ones, in the sense that they consider all requirements of each element at the same time, while the $\log n$ ratio algorithms considers each unit of requirement separately. The removal of elements, which only happens with the adaptive algorithms, not only creates extra complicacy for the algorithm, but also provides a chance for the f-ratio adaptive approximation

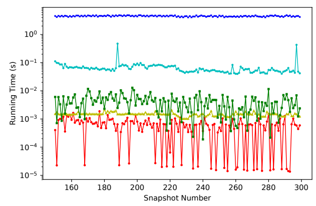
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2912963, IEEE Transactions on Wireless Communications

TABLE II: Summary of the Datasets

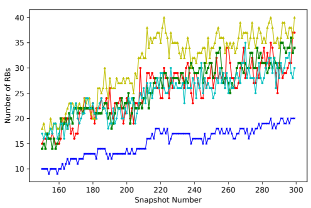
Dataset	Description	Average # of devices	Average # of links	Average # of Changes
KAIST	Actual, KAIST campus [31]	92.0	41.4	1.1
Subway	Simulated, subway scenario [32]	114.8	48.0	5.1
Downtown	Simulated, downtown scenario [32]	184.3	71.5	8.6
SLAW	Simulated, SLAW model [33]	300.0	131.6	7.4



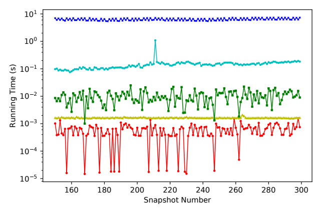
(a) Subway - Performance



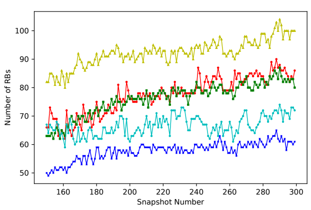
(b) Subway - Running Time



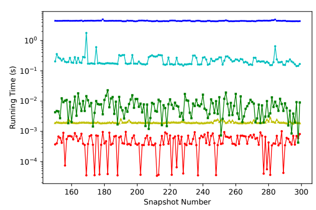
(c) Downtown - Performance



(d) Downtown - Running Time



(e) SLAW - Performance



(f) SLAW - Running Time

Fig. 4: Performance and running time of simulated mobility traces

Algorithm 8 RB Assignment

Input: R^t, L^t, dB_0

Output: $F^t : R^t -> L^t$. Assignment of RBs in R^t to links in L^t

for $\forall r \in R^t$ **do**

  $F(r) = \emptyset$

  for $\forall g \in r$ **do**:

    Arbitrarily pick an unselected link l in L^t from grid g

    Let $S = F(r) \cup \{l\}$

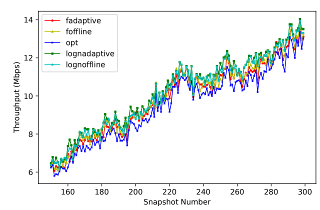
    Calculate SINR for all links in S when they are sharing one RB

    if The minimum SINR of links in S is smaller than dB_0 **then**:

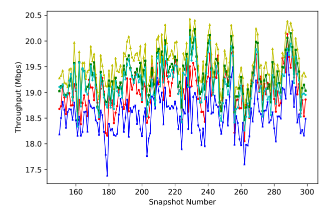
      break

    $F(r) = S$

    Mark l as selected.



(a) Downtown



(b) SLAW

Fig. 5: Throughput

algorithm to refine its result, resulting in a better performance than the offline counterpart.

In terms of running time, the f ratio algorithms demonstrate their advantage of being "coarse": it is not necessary for them to consider individual requirements. With more element addition/removal happening in the cell, this advantage significantly saves time and both f ratio algorithms are generally faster than the $\log n$ ratio ones. The $\log n$ ratio offline algorithm is much slower than the other approximation algorithms, despite that it uses the least number of RBs among them.

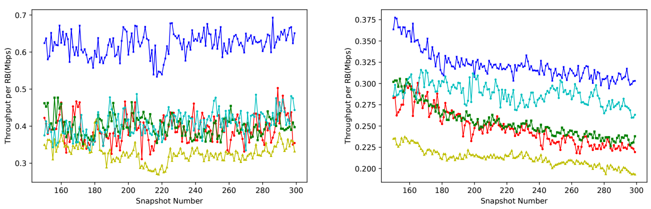
D. Throughput

In this paper, the resource allocation we obtained cannot be directly applied to realistic scenarios, since we use the grids, instead of the individual links as elements in the algorithms. Therefore, For each RB we selected, the only piece of information available to us is the set of grids the RB will be assigned to. In order to have an explicit assignment of RBs to the links, we use the following heuristic algorithm for each snapshot.

In Alg. 8, we iterate through the selected RBs R^t. For each set, we iteratively add an unselected link per each grid that the MIFS can cover. Before each addition, we check for the interference by calculating the SINR of all previously selected links and the current addition when they shares an RB. If any SINR falls below a predefined threshold dB_0 (we set the value to $15dB$, the same as the one used for generating MIFSs), we will not add any more links to this RB. Using the algorithm, we can obtain the RB assignment to all links and thus can calculate the network throughput, defined as the sum of the data rates of all links.

Fig. 5 illustrates the throughput calculated using Alg. 8 for all algorithms, in the downtown and the SLAW scenarios. In the downtown scenario, the throughput for all algorithms are almost identical while the optimal algorithm appears to have a bit inferior throughput. For the SLAW scenario, however, the throughput of the optimal algorithm is notably lower than all other algorithms. The results for the other two scenarios are similar to the one in the downtown scenario and are omitted here. The result that the optimal being the worst is of interest and we will answer it in two steps.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2912963, IEEE Transactions on Wireless Communications



(a) Downtown (b) SLAW

Fig. 6: Average Throughput per RB

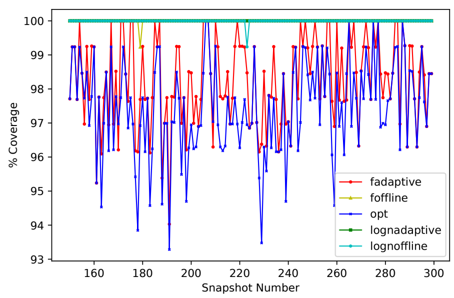
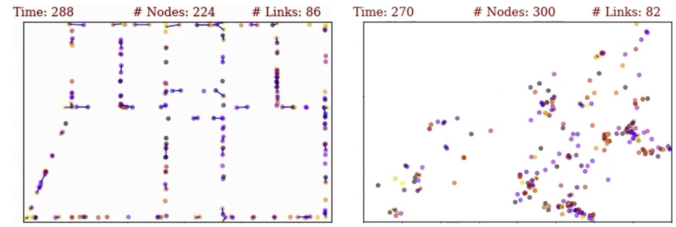


Fig. 7: Coverage in the SLAW scenario

The first explanation is relatively simple. In the previous sections, we demonstrated that the optimal algorithm needs less RBs than the other algorithms. Thus, when we are assigning the RBs to the links, each RB in the optimal solution will need to serve more links on average, comparing with the approximated solutions. When it goes to throughput calculation, clearly the links will have lower SINR when they need to share the RB with more other links. This is the primary reason why the optimal solution tends to have lower throughput. To reveal the true advantage of the optimal solution, we calculate the average throughput generated by each RB in Fig. 6. In this figure, we can observe that each single RB in the optimal solution is more efficient: it generates a higher throughput comparing with RBs in approximated solutions. Again, we omit the result in the other two scenarios as they are similar. Fig. 5 and Fig. 6 combined imply that although the approximation algorithms may have a higher throughput, it is mainly so at the cost of using more RBs.

The second explanation reveals one possible issue in the RB assignment algorithm Alg. 8. As we discussed in Sect. IV-B, the way we assign the links to grids is an approximation and may not always be accurate. Thus, in Alg. 8, it is possible that some links are not served by any RB at the end. It may happen in the case that we originally wanted to use an RB to serve m grids, but it turns out that the links in the grids are closer than expected and some SINR falls below the threshold. To support this claim, we calculate the percentage of links served after running Alg. 8 for each algorithm. It turns out that the percentage is maintained at 100% for the KAIST, downtown and subway scenarios. However, as depicted in Fig. 7, the RBs generated by several algorithms fail to cover all links in the SLAW model, with the optimal solution having the lowest coverage percentage. This finding corresponds to the fact that



(a) Downtown (b) SLAW

Fig. 8: Snapshots of typical device/link locations

the optimal solution has a notably lower throughput in the SLAW model.

Fortunately, the issue in RB assignment can be fixed. Assigning links to grids and then solve the SMC problem enables the theoretically efficient algorithms. As we explained, the resulting RBs will be able to serve all links in the majority of scenarios (three out of four in our experiments.) When the number of RBs is not enough, we can simply solve a smaller version of the resource allocation problem specifically for the links that are not served. As we can see from Fig. 7, at most 7% of the links are not served and it will be efficient to obtain a feasible, even optimal resource allocation for those links.

E. Impact of Mobility Patterns

In this section, we would like to analyze how the mobility patterns, specifically, how the typical locations of links and devices may impact the performance of the algorithms. As we already seen in Fig. 4, the approximation algorithms are farther away from the optimal one in the downtown scenario than in the SLAW scenario, while the average number of changes in the two scenarios are not too different. To see this, we plot the devices and links in the two scenarios, as in Fig. 8. For each scenario, we pick a typical snapshot, which can be used as a representative of the device/link locations.

What we can observe from Fig. 8 is a sheer difference between the two scenarios. For the downtown scenario, the traces are generated with the consideration of actual restrictions in a city. Thus, the devices/links are aligned on certain lines, which corresponds to the roads in the downtown area. For the SLAW scenario, the devices move more freely yet they tend to be more clustered. With the figures, we can explain why the approximation algorithms in the downtown scenario has inferior performance than those in the SLAW scenario. In the downtown scenario, the links are more sparse, so that it is more likely to use one RB to serve multiple links and the problem is more complicated. In the SLAW scenario, however, there exists less options to assign the RBs to the links as they are more clustered. The performance of the approximation algorithms may move closer to the theoretical approximation ratio with a harder problem and the gap between them and the optimal solution will increase.

VI. Conclusion

In this paper, we proposed an adaptive solution framework to the dynamic resource allocation problem in D2D communication. Within the framework, we first introduced an approach that can generate stable MIFSs, which simplified our core problem ASMC. Then, we proposed two adaptive approximation algorithms for ASMC, with approximation ratios f and $\log n$ respectively. In the experiments, we used actual and simulated mobility traces to evaluate the algorithms. The results demonstrated that the adaptive solutions are much faster than the optimal or approximated offline methods. Also, the performance of the adaptive algorithms are still comparable with the optimal solution, indicating its applicability in realistic scenarios.

Acknowledgement

This work was supported by a start up grant from Santa Clara University, NSF CNS-1443905 and DTRA HDTRA1-14-1-0055.

References

[1] Cisco Visual Networking Index. Forecast and methodology, 2016-2021 white paper. *Technical Report, Cisco, Tech. Rep.*, 2017.

[2] Klaus Doppler, Mika Rinne, Carl Wijting, Cássio B Ribeiro, and Klaus Hugl. Device-to-device communication as an underlay to lte-advanced networks. *IEEE Communications Magazine*, 47(12), 2009.

[3] Bin Wang, Li Chen, Xiaohang Chen, Xin Zhang, and Dacheng Yang. Resource allocation optimization for device-to-device communication underlaying cellular networks. In *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*, pages 1–6. IEEE, 2011.

[4] Chia-Hao Yu, Klaus Doppler, Cassio B Ribeiro, and Olav Tirkkonen. Resource sharing optimization for device-to-device communication underlaying cellular networks. *IEEE Transactions on Wireless communications*, 10(8):2752–2763, 2011.

[5] Peng Cheng, Lei Deng, Hui Yu, Youyun Xu, and Hailong Wang. Resource allocation for cognitive networks with d2d communication: An evolutionary approach. In *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2671–2676. IEEE, 2012.

[6] Feiran Wang, Lingyang Song, Zhu Han, Qun Zhao, and Xiaoli Wang. Joint scheduling and resource allocation for device-to-device underlay communication. In *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 134–139. IEEE, 2013.

[7] Tuong Duc Hoang, Long Bao Le, and Tho Le-Ngoc. Energy-efficient resource allocation for d2d communications in cellular networks. *IEEE Transactions on Vehicular Technology*, 65(9):6972–6986, 2016.

[8] Rongqing Zhang, Lingyang Song, Zhu Han, Xiang Cheng, and Bingli Jiao. Distributed resource allocation for device-to-device communications underlaying cellular networks. In *2013 IEEE International Conference on Communications (ICC)*, pages 1889–1893. IEEE, 2013.

[9] Chen Xu, Lingyang Song, Zhu Han, Qun Zhao, Xiaoli Wang, Xiang Cheng, and Bingli Jiao. Efficiency resource allocation for device-to-device underlay communication systems: A reverse iterative combinatorial auction based approach. *IEEE Journal on Selected Areas in Communications*, 31(9):348–358, 2013.

[10] Dong Heon Lee, Kae Won Choi, Wha Sook Jeon, and Dong Geun Jeong. Resource allocation scheme for device-to-device communication for maximizing spatial reuse. In *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 112–117. IEEE, 2013.

[11] Yong Li, Depeng Jin, Pan Hui, and Zhu Han. Optimal base station scheduling for device-to-device communication underlaying cellular networks. *IEEE Journal on Selected Areas in Communications*, 34(1):27–40, 2016.

[12] Alan Kuhnle, Xiang Li, and My T Thai. Online algorithms for optimal resource management in dynamic d2d communications. In *2014 10th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, pages 130–137. IEEE, 2014.

[13] Alan Kuhnle, Xiang Li, J David Smith, and My T Thai. Online set multicover algorithms for dynamic d2d communications. *Journal of Combinatorial Optimization*, 34(4):1237–1264, 2017.

[14] C-H Yu, Olav Tirkkonen, Klaus Doppler, and Cássio Ribeiro. Power optimization of device-to-device communication underlaying cellular communication. In *2009 IEEE International Conference on Communications (ICC)*, pages 1–5. IEEE, 2009.

[15] Pekka Janis, YU Chia-Hao, Klaus Doppler, Cassio Ribeiro, Carl Wijting, Hugl Klaus, Olav Tirkkonen, and Visa Koivunen. Device-to-device communication underlaying cellular communications systems. *International Journal of Communications, Network and System Sciences*, 2(03):169, 2009.

[16] Feiran Wang, Chen Xu, Lingyang Song, Qun Zhao, Xiaoli Wang, and Zhu Han. Energy-aware resource allocation for device-to-device underlay communication. In *2013 IEEE International Conference on Communications (ICC)*, pages 6076–6080. IEEE, 2013.

[17] Rongqing Zhang, Xiang Cheng, Qi Yao, Cheng-Xiang Wang, Yang Yang, and Bingli Jiao. Interference graph-based resource-sharing schemes for vehicular networks. *IEEE Transactions on Vehicular Technology*, 62(8):4028–4039, 2013.

[18] Rongqing Zhang, Xiang Cheng, Liuqing Yang, and Bingli Jiao. Interference graph-based resource allocation (ingra) for d2d communications underlaying cellular networks. *IEEE Transactions on Vehicular Technology*, 64(8):3844–3850, 2015.

[19] Tinghan Yang, Rongqing Zhang, Xiang Cheng, and Liuqing Yang. Graph coloring based resource sharing (gcrs) scheme for d2d communications underlaying full-duplex cellular networks. *IEEE Transactions on Vehicular Technology*, 66(8):7506–7517, 2017.

[20] Mohammad Zulhasnine, Changcheng Huang, and Anand Srinivasan. Efficient resource allocation for device-to-device communication underlaying lte network. In *2010 IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 368–375. IEEE, 2010.

[21] Xiang Cheng, Liuqing Yang, and Xia Shen. D2d for intelligent transportation systems: A feasibility study. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):1784–1793, 2015.

[22] Xiang Cheng, Chen Chen, Wuxiong Zhang, and Yang Yang. 5g-enabled cooperative intelligent vehicular (5genciv) framework: When benz meets marconi. *IEEE Intelligent Systems*, 32(3):53–59, 2017.

[23] Fanhui Zeng, Rongqing Zhang, Xiang Cheng, and Liuqing Yang. Channel prediction based scheduling for data dissemination in vanets. *IEEE Communications Letters*, 21(6):1409–1412, 2017.

[24] Nicholas G Hall and Dorit S Hochbaum. A fast approximation algorithm for the multicovering problem. *Discrete Applied Mathematics*, 15(1):35–40, 1986.

[25] Giorgio Gambosi, Marco Protasi, and Maurizio Talamo. *Dynamic maintenance of approximated solutions of min-weighted node cover and min-weighted set cover problems*. International Computer Science Institute, 1993.

[26] Noga Alon, Baruch Awerbuch, and Yossi Azar. The online set cover problem. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 100–105. ACM, 2003.

[27] Niv Buchbinder and Joseph Naor. Online primal-dual algorithms for covering and packing problems. In *European Symposium on Algorithms*, pages 689–701. Springer, 2005.

[28] Anupam Gupta, Ravishankar Krishnaswamy, Amit Kumar, and Debmalya Panigrahi. Online and dynamic algorithms for set cover. pages 537–550, 2017.

[29] Lei Lei, Zhangdui Zhong, Chuang Lin, and Xuemin Shen. Operator controlled device-to-device communications in lte-advanced networks. *IEEE Wireless Communications*, 19(3), 2012.

[30] Vijay V Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.

[31] Injong Rhee, Minsu Shin, Seongik Hong, Kyunghan Lee, Seongjoon Kim, and Song Chong. CRAWDAD dataset ncsu/mobilitymodels (v. 2009-07-23). Downloaded from https://crawdad.org/ncsu/mobilitymodels/20090723, July 2009.

[32] Sylvia Todorova Kouyoumdjieva, lafur Ragnar Helgason, and Gunnar Karlsson. CRAWDAD dataset kth/walkers (v. 2014-05-05). Downloaded from https://crawdad.org/kth/walkers/20140505, May 2014.

[33] Kyunghan Lee, Seongik Hong, Seong Joon Kim, Injong Rhee, and Song Chong. Slaw: A new mobility model for human walks. In *IEEE INFOCOM 2009*, pages 855–863. IEEE, 2009.

Xiang Li (M'18) received her Ph.D. degree in Computer and Information Science and Engineering department of University of Florida. She is an Assistant Professor at the Department of Computer Engineering of Santa Clara University. Her research interests are centered on the large-scale optimization and its intersection with cyber-security of networking systems, big data analysis, and cyber physical systems. She has published 25 articles in various prestigious journals and conferences such as IEEE Transactions on Mobile Computing, IEEE Transactions on Smart Grids, IEEE INFOCOM, IEEE ICDM, including one Best Paper Award in IEEE MSN 2014, Best Paper Nominee in IEEE ICDCS 2017, and Best Paper Award in IEEE International Symposium on Security and Privacy in Social Networks and Big Data 2018.

She has served as Publicity Co-Chair of International Conference on Computational Data Social Networks 2018, Session Chair of ACM SIGMETRICS International Workshop 2018, and on TPC of many conference including IEEE ICDCS, IEEE ICDM workshop, COCOA etc., and also served as a reviewer for several journals such as *IEEE Transactions* on *Mobile Computing*, *IEEE Transactions* on *Networks Science* and *Engineering* and *Journal* of *Combinatorial Optimization*, etc.

Tianyi Pan is currently working toward the PhD degree in the Department of Computer and Information Science and Engineering, University of Florida, under the supervision of Dr. My T. Thai. His research focuses on approximation algorithms of optimization problems and vulnerability analysis in interdependent networks, including online social networks, smart grid and communication networks.

Md Abdul Alim received the B.Sc. degree in computer science and engineering from Bangladesh University of Engineering and Technology, Bangladesh. He received his Ph.D. degree in computer science from the University of Florida in 2016. His research interests include network vulnerability and community structure analysis in complex networks including large-scale biological, wireless, and online social networks; social-aware device-to-device communication; influence propagation in online social networks and approximation algorithms and its application in combinatorial optimization. Currently, he is focusing on applying combinatorial optimization techniques to solve scheduling problems involving large-scale real-world networks arising in industrial transportation domain.

My T. Thai (M'06) is a UF Research Foundation Professor at the Computer and Information Science and Engineering department, University of Florida. Her current research interests are on scalable algorithms, big data analysis, cybersecurity, and optimization in network science and engineering, including communication networks, smart grids, social networks, and their interdependency. The results of her work have led to 6 books and 150+ articles, including IEEE MSN 2014 Best Paper Award, 2017 IEEE ICDM Best Papers Award, 2017 IEEE ICDCS Best Paper Nominee, and 2018 IEEE/ACM ASONAM Best Paper Runner up.

Prof. Thai has engaged in many professional activities. She has been a TPC-chair for many IEEE conferences, has served as an associate editor for *Journal* of *Combinatorial Optimization* (*JOCO*), *Journal* of *Discrete Mathematics*, *IEEE Transactions* on *Parallel and Distributed Systems*, *IEEE Transactions* on *Network Science* and *Engineering*, and a series editor of *Springer Briefs* in *Optimization*. She is a founding EiC of the *Computational Social Networks* journal. She has received many research awards including a UF Provosts Excellence Award for Assistant Professors, UFRF Professorship Award, a Department of Defense (DoD) Young Investigator Award, and an NSF (National Science Foundation) CAREER Award.