2014 10th International Conference on Mobile Ad-hoc and Sensor Networks

Online Algorithms for Optimal Resource Management in Dynamic D2D Communications

Alan Kuhnle, Xiang Li, and My T. Thai
Department of Computer and Information Science and Engineering, University of Florida
Gainesville, FL 32611
Email: {kuhnle, xixiang, mythai}@cise.ufl.edu

***Abstract*—Device-to-device (D2D) communications has recently emerged as a promising technology for boosting the capacity of cellular systems. D2D enables direct communication between mobile devices over the cellular band without utilizing infrastructure nodes such as base stations, thereby reducing the load on cellular base stations and increasing network throughput through spatial reuse of radio resources. Hence it is important to optimally allocate these radio resources. Furthermore, since the composition of a cellular macrocell is highly dynamic, it is critical to adaptively update the resource allocation for D2D communications rather than recomputing it from scratch. In this work, we develop the first online algorithm, namely ODSRA, for dynamic resource allocation while maximizing spatial reuse. At the core of the resource allocation problem is the online set multicover problem, for which we present the first deterministic $O(\log n \log m)$-competitive online algorithm, where n is the number of elements, and m the number of sets. By simulation, we show the efficacy of ODSRA by analyzing network throughput and other metrics, obtaining a large improvement in running time over offline methods.**

I. INTRODUCTION

The proliferation of tablets, smartphones, and other highly capable mobile devices has significantly increased the demand for high-speed wireless access [7]. This exponential growth in bandwidth-intensive applications, especially in mobile social networks, has strained modern-day wireless networks and motivated the need for new technologies to optimize the usage of the scarce radio resources. In particular, device-to-device (D2D) communications over the licensed spectrum bands has recently emerged as a promising technology for boosting the capacity of cellular systems [4]. Using D2D, mobile users can communicate directly over cellular spectrum bands while bypassing the base stations. In contrast to conventional D2D over short-range, limited capacity technologies such as Bluetooth, D2D over cellular provides longer transmission ranges, improved spectrum sharing, higher capacities, guaranteed quality-of-service, and a broader range of applications and services. Owing to its promising potential, D2D is now viewed as a cornerstone technology in emerging 5th generation (5G) wireless systems [5].

However, effectively leveraging D2D communications faces several fundamental challenges. Most notably is optimized management of cellular interference levels and allocation of spectral resources. In D2D communication, a cellular base station offloads some of the communication traffic to devices within the cell. In this way, devices can reduce power consumption, as D2D transmission would be over shorter range than communication with base station, and also D2D communication can reuse radio resources within the cell. Owing to this power reduction, it is possible for more than one D2D communication link to spatially reuse the same radio resource block. Therefore, one of the problems that D2D design need to handle is to optimally allocate the resource blocks to devices in order to maximize this spatial reuse.

The two main challenges for this allocation are first, determining which devices are able to communicate concurrently on a resource block without interference, and second, the cellular system is dynamic; new devices enter the system, devices leave the system, and devices move around within the system.

Spatial reuse of radio resource has been studied in a D2D context [12]; recently Lee et. al. published an algorithm that outperformed earlier methods [8]. This algorithm finds maximal sets of devices that can communicate without interference among the D2D links, and uses the greedy set multicover (SMC) algorithm to perform the allocation. However, the SMC algorithm must be run every time the cellular system changes. For example, when a new device joins the cell, the entire allocation procedure must be redone.

Recomputing the allocation every time the system changes has the following drawbacks. First, the time taken for the allocation is large. If the base station is under heavy load and devices are changing quickly, allocation based on SMC may not be able to be performed fast enough. Second, reallocation means that old devices will often be assigned new resource blocks – this creates overhead in the form of control signals that must be sent to each device every time the system changes.

To address these problems, in this work we consider online algorithms for the resource allocation problem. An online algorithm receives the problem instance in small pieces and updates its solution according to the received instance, while always maintaining a feasible solution with a theoretical performance bound, called *competitive ratio*. The *competitive ratio* compares the worst-case quality of an online algorithm's solution over all possible sequences of possible input compared to the quality of an offline optimal algorithm, which is allowed to know the input in advance.

Along this direction, we develop an online solution to the D2D resource allocation problem. At the core of this problem is the online set multicover (OSMC) problem. For OSMC, we provide an online algorithm with competitive ratio of $O(\log n \log m)$, where n is the number of elements and

978-1-4799-7394-1/14 $31.00
DOI 10.1109/MSN.2014.24

m is the number of sets. First, we produce a randomized algorithm with this competitive ratio. We then derandomize the algorithm, which requires nonstandard derandomization techniques because of the online nature of the problem. To the best of our knowledge, this is the first deterministic algorithm with this competitive ratio that has been given for the online set multicover problem.

The contributions of this paper are as follows.

- We present a deterministic, online algorithm for OSMC with competitive ratio $O(\log n \log m)$, where n is the total number of elements, and m is the number of sets. This algorithm also solves the online set cover with repetitions (OSCWR) problem with the same competitive ratio $O(\log n \log m)$. This result improves a deterministic bicriteria $O(\log n \log m)$-competitive algorithm for OCSWR that had been given in [2].
- Based on OSMC, we develop an online algorithm for D2D spatial reuse that quickly updates the allocation in response to changes in the system, leaving previously allocated devices using the same resources as before the change.
- We compare our algorithm to the algorithm in [8] by running experiments on a simulated cellular system. Our algorithm maintains a similar spatial reuse (usually within 10%) while improving the running time by a factor of usually at least 50, and often 100 or higher.

The remainder of this paper is organized as follows. In section II, we define the model of the cellular system and the radio resources, discuss D2D interference, and define the maximization of spatial reuse problem. We consider online solutions to this problem in section III, in which we give the online algorithms for OSMC and maximization of spatial reuse and prove the competitive ratios. We evaluate the algorithms experimentally in section IV. In section V, we discuss related work, and finally section VI concludes the paper.

II. Model and Problem Definition

A. Network Model

The cellular system will be modeled as a downlink orthogonal frequency division multiple access (OFDMA) wireless cell. The cell consists of a single base station (BS), together with user equipments (UEs). The UEs will be divided into a set E of *primary devices*: user equipment which communicates directly with BS, and a set F of *secondary devices*: user equipment which does not communicate with BS, but communicates with devices in E via D2D communication.

Each secondary device $s \in F$ communicates only with a primary device $e(s) \in E$ via a D2D link; $e(s)$ itself communicates directly with the base station as well as with possibly many secondary devices in F. One primary device together with its dependent secondary devices comprise a D2D subnetwork. A D2D link consists of a primary device and a secondary device. The secondary device receives data transmission indirectly from BS through the primary device.

The sets of primary and secondary devices, E and F, vary with time. At any time, devices may enter, leave or move around within the network, thereby changing the resource allocation problem.

B. Radio Resource

As in the typical OFDMA model, the frequency band is divided into orthogonal subcarriers, which are organized into N equal-sized subchannels. A single subchannel is composed of multiple subcarriers. Furthermore, for the resource allocation, there is a division of time into frames. The basic unit for resource allocation is the resource block (RB), which is one subchannel during one frame.

In this paper, we do not consider how the D2D links form; we consider only the allocation of resources in response to a change in D2D links.

Each primary device $e \in E$ determines how many resource blocks are required for communication on its D2D subnetwork, and also the transmission power required in each block, and notifies BS of its resource requirements.

C. D2D interference

In this section, we briefly present the interference analysis among D2D transmissions. Details can be found in [8]. Consider two D2D links, i and j transmitting on the same RB. The channel gain between transmitter t_i of i and receiver r_j of j is defined as $g_{i,j} = z_{i,j}\eta_{i,j}\psi_{i,j}$, where $z_{i,j}$ is the path-loss component, $\eta_{i,j}$ the shadowing component, and $\psi_{i,j}$ the multi-path fading component between t_i and r_j.

We make the following assumptions, as in [8]:

- from the position of the nodes, BS can determine path-loss components,
- path-loss component $z_{i,j}$ remains constant during transmission,
- shadowing components η of all D2D links are iid with distribution log-normal with mean zero and standard deviation σ_η, and
- multi-path components of all links over all RBs are independent of each other and have exponential distributions with mean μ_ψ.

Then, the logarithm of $g_{i,j}$ has normal distribution with mean $10\log_{10} z_{i,j} + 10\log_{10}\mu_\psi - 2.5$ and standard deviation $\sqrt{\sigma_\eta^2 + 5.57^2}$ [8] [6].

Let S_D denote all D2D links sharing RB D. Let p be the maximum allowed transmission power on each D2D link. The interference $I(r_j)$ on receiver r_j is the sum of all channel gains from other devices i sharing D:

$$I(r_j) = \sum_{i\in S_D, i\neq j} g_{i,j}p.$$

By [6], we can approximate $I(r_j)$, a sum of log-normal variables, with another log-normal variable. Thus, we get $\log I(r_j)$ approximately follows normal distribution with

$$\sigma = \sqrt{10\log_{10}\left(\alpha\frac{\sum_{i\in S_D, i\neq j}(\zeta_{i,j}^b)^2}{(\sum_{i\in S_D, i\neq j}\zeta_{i,j})^2} + 1\right)},$$

$$\mu = 10\log_{10}\left(\sum_{i\in S_D, i\neq j}\zeta_{i,j}\right) + \beta - \frac{\sigma^2}{2},$$

where $\alpha = 10^{\frac{\sigma_\eta^2+5.57^2}{10}} - 1, \zeta_{i,j} = z_{i,j}\mu_\psi 10^{-0.25}p$, and $\beta = \frac{\sigma_\eta^2+5.57^2}{2}$ [8].

D. Problem Definition

Each primary device $e \in E$ determines how many RBs are needed for communication on its D2D subnetwork. Denote the resource requirement of e by k_e, which will be a positive integer. Device e, after determining the value of k_e, requests k_e RBs from the base station. Thus, it is necessary for the base station to assign resource blocks in order to maximize the spatial reuse of the radio resource.

In order to maximize spatial reuse, we consider maximal sets of primary devices which may all communicate on the same RB without interference. Such a set of devices is termed a maximal interference-free set (MIFS). Denote the collection of all MIFSs as $\mathcal{S}$. The determination of $\mathcal{S}$ is not the focus of this paper; however, in section IV we discuss a method to approximate it.

Notice that each MIFS $S_j \in \mathcal{S}$ may be assigned a single RB for communication, as the devices in S_j do not interfere. Therefore, in order to maximize the spatial reuse of RBs, we must choose a minimum subcollection $\mathcal{W} \subset \mathcal{S}$ such that each primary device e is in k_e elements of $\mathcal{S}$. Thus, we have the classical *constrained set multicover* problem; the maximization of spatial reuse can be determined by solving this classical problem in an online fashion.

The changes to the cellular system with respect to time will change the resource requirement r_e for each primary device e managing D2D connections. These changes will require the solution to be updated. Therefore, we consider the problem in an online context, defined as follows.

Problem 1 (Online Maximization of Spatial Reuse of Frequency Spectrum (OMSRFS))**.** *Given a cellular system consisting of a base station, primary devices E, secondary devices F, which may change with time, and a collection $\mathcal{S}$ of maximal interference-free subsets of E, the problem asks us to maintain an allocation of resource blocks, i.e. elements in $\mathcal{S}$, satisfying for each device e its coverage requirement k_e, and minimizing the number of elements of $\mathcal{S}$ employed, thereby maximizing the spatial reuse of radio resource.*

III. Solutions

First, we will approximate the online set multicover (OSMC) problem by defining a randomized algorithm with competitive ratio $O(\log n \log m)$, where $m = |S|$, $n = |E|$. The randomized algorithm is then derandomized while preserving the competitive ratio. Next, we formulate algorithms based on OSMC algorithm to solve the OMSRFS problem.

We define the concept of competitive ratio for any online algorithm. Intuitively, the competitive ratio quantifies how badly the online algorithm can perform compared to an algorithm that is allowed to know the input in advance. The online algorithm may return different solutions based on the order it receives the input, so we must consider all orderings of an input.

Definition (Competitive ratio). Let $\mathcal{X}$ be the collection of all potential inputs to online algorithm B for a minimization online problem. For $X \in \mathcal{X}$, denote the set of all orderings of elements in X by $O(X)$. Let $O \in O(X)$, let $A_{opt}(X)$ be an offline optimal solution to X, and let $A(O)$ be the solution returned by B. Denote the cost of a solutions A by $c(A)$. Then the *competitive ratio* is defined as

$$\max_{X \in \mathcal{X}} \max_{O \in O(X)} \frac{c(A(O))}{c(A_{opt}(X))}.$$

A. Online Set Multicover

Let us separate the OSMC problem from the D2D context.

Problem 2 (Online Set Multicover)**.** *Given a universe E and a collection of sets $\mathcal{S} = \{S_1, \ldots, S_m\}$, elements from a subset $X \subset E$ are given to the algorithm one-by-one in a specific order. The algorithm does not know the set X beforehand. Each element e has a coverage requirement k_e, which is a positive integer. The algorithm must cover each element $e \in X$ with k_e sets as it arrives, minimizing the total number of sets used. Let A denote the sets picked by the algorithm.*

First, we formulate the set multicover problem as an integer program:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{m} w_i & & (1) \\ \text{subject to} \quad & \sum_{i:e \in S_i} w_i \geq k_e, & e \in E & \\ & w_i \in \{0, 1\}, & i \in [m]. & \end{aligned}$$

Here, w_i is 1 if set S_i is picked, and 0 otherwise. This IP has linear relaxation

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{m} w_i & & (2) \\ \text{subject to} \quad & \sum_{i:e \in S_i} w_i \geq k_e, & e \in E & \\ & -w_i \geq -1, & i \in [m], & \\ & w_i \geq 0, & i \in [m]. & \end{aligned}$$

Notice the constraints $-w_i \geq -1$. These are necessary to ensure that each set may only be picked once. In the corresponding linear program for the set cover problem, these constraints are unnecessary, as there is no benefit to picking a set more than once. However, for the set multicover problem, since each element e may be required to be covered more than once, these constraints are necessary.

The corresponding dual program is

$$\begin{aligned} \text{maximize} \quad & \sum_{e \in E} k_e y_e - \sum_{i=1}^{m} z_i & & (3) \\ \text{subject to} \quad & \left(\sum_{e:e \in S_i} y_e \right) - z_i \leq 1, & i \in [m] & \\ & y_e \geq 0, & e \in E, & \\ & z_i \geq 0, & i \in [m]. & \end{aligned}$$

As a starting point, we will give an online solution to LP (2) with competitive ratio $O(\log m)$.

1) Online Fractional Algorithm: Define $k = \max_{e \in E} k_e$. We define an online fractional algorithm in Alg. 1 for LP (2). Let $m = |S|$, $n = |E|$. For each $i \in [m]$, we let $w_i \in [m]$ are the primal variables in LP (2). In order to establish the competitive ratio, the algorithm also updates values for dual variables z_i and y_e.

Alg. 1 works as follows. Initially z_i is set to 0 for all i, w_i is set to $1/m^2$ for all i. The variables y_e are initialized to 0. The input is unknown in advance, and each element arrives one at a time. As each element e arrives, the algorithm must ensure that the corresponding constraint in LP (2) is satisfied. To satisfy this constraint, the algorithm considers all of the sets in S that contain e. For each such set S_i, it first checks if $w_i < 1$. If so, it sets w_i to $\min\{2w_i, 1\}$, otherwise it increments z_i.

Algorithm 1: An online fractional algorithm for LP (2) with competitive ratio $O(\log m)$.

```
Input: A set multicover instance. Elements from an unknown set X, one at a
       time.
Output: Primal values w, dual values y, z
for i ∈ [m] do
    z_i = 0, w_i = 1/m^2
for e ∈ E do
    y_e = 0
while e arrives do
    while Σ_{i:e∈S_i} w_i < k_e do
        for w_i : e ∈ S_i do
            if w_i < 1 then
                w_i := min{1, 2w_i}
            else
                z_i := z_i + 1
        y_e := y_e + 1
```

Theorem 1. *Algorithm 1 yields an online fractional solution for set multicover with $O(\log m)$-competitive ratio.*

Proof.

Claim 1.

$$\sum_{i=1}^{m} w_i \leq \sum_{e \in E} k_e y_e - \sum_{i=1}^{m} z_i + \frac{1}{m}. \tag{4}$$

Proof. The proof will be by induction on the elements given to the algorithm. Before the first element arrives, the inequality holds. Now say e arrives, and consider one iteration of the while loop.

Let $\mathcal{S} = \{i : e \in S_i\}$. Let $\mathcal{T} = \{i : e \in S_i$ and w_i does not increase during the iteration$\}$. Let $|\mathcal{T}| = l$.

Then the change in the LHS of the inequality is at most $k_e - l$. To see this, recall that

$$\sum_{i \in \mathcal{S}} w_i < k_e.$$

Now, if w_i doesn't increase, then $w_i = 1$, hence

$$\sum_{i \in \mathcal{S}-\mathcal{T}} w_i < k_e - l. \tag{5}$$

The sum on the LHS of (5) is not less than the increase in the LHS of (4). The RHS of (4) increases by exactly $k_e - l$. Hence, inequality (4) holds. □

Claim 2. The solution to the dual obtained by dividing the dual values assigned during the algorithm by $2 \log m$ is feasible.

Proof. Consider the constraint of the dual (3) corresponding to set S_i. Notice that $\sum_{e \in S_i} y_e - z_i$ increases iff. w_i doubles. The initial value of w_i is $\frac{1}{m^2}$ and the final value is at most 1. Hence w_i doubles at most $2 \log m$ times. □

By Claims 1 and 2, the cost of the feasible primal solution $\{w_i\}$ is at most $O(\log m)$ of a feasible dual solution. Thus, the theorem is proved by the well-known LP-duality theorem [11]. □

2) Online Integer Solution: To convert the fractional algorithm above to an integer algorithm, a randomized rounding step is added. Define new variables $\hat{x}_i = \min(1, 6 \log n \cdot w_i)$.

Suppose an element e arrives; Denote $S_e = \{S_i : e \in S_i\}$. After the weight augmentation process is finished, for each $S_i \in S_e$, choose S_i for the cover with probability equal to the increase in $\hat{x}_i$. Let X_i be the random variable indicating whether set S_i was picked either in this iteration or a previous one, and let $Y = \sum_{S_i \in S_e} X_i$. So element e is left uncovered with probability $Pr[Y < k_e]$.

A version of Chernoff's bound states for any $\delta > 0$,

$$Pr[Y < (1-\delta)\mu] \leq \exp(-\mu\delta^2/2).$$

Now, the mean μ of Y satisfies $\mu \geq 6 \log n \cdot k_e$. Thus, using Chernoff's bound with $\delta = 1 - \frac{1}{6 \log n} \geq 5/6$, we have

$$Pr[Y < k_e] \leq \exp(-6 \log n \cdot \frac{25}{72}) \leq \frac{1}{n^2}.$$

Finally, it follows that the probability that a single element is uncovered is at most $\frac{1}{n}$, and the expected cost of the integral solution is at most $6 \log n$ times the cost of the fractional solution produced by Algorithm 1.

Theorem 2. *Algorithm 1 combined with the rounding procedure outlined above yields a randomized algorithm for the online set multicover problem with competitive ratio*

$$O(\log n \log m).$$

3) Derandomization: Finally, we derandomize the randomized algorithm. This is accomplished through the use of a potential function Φ.

For each $e_i \in E$, let c_i denote the number of times e_i has been covered, k_i denote the coverage requirement of e_i, and C the set of elements which have had their coverage requirements met. For each set $s \in S$, let $\chi_C(s) = 1$ if s has been picked for the cover, and 0 otherwise. Define

$$g(e_i) = \exp\left(\sum_{s:e_i \in s} \alpha w(s) - \alpha(c_i + 1)\right).$$

Define potential function $\Phi = \Phi_1 + \Phi_2$, where

$$\Phi_1 = \sum_{e_i \notin C} g(e_i),$$

$$\Phi_2 = \frac{\exp\left(\sum_{s \in S}((\log 2) \cdot \chi_C(s) - \alpha w(s)) - 1\right)}{2}.$$

Lemma 1. *The following properties are satisfied by Φ.*

Algorithm 2: The deterministic, online algorithm for set multicover.

begin
Run the online fractional Algorithm 1. Every time the weight of a set is augmented, add it to the cover if the potential function Φ does not increase by doing so.

(1) *Choosing $\alpha = \log(2n)$ ensures that initially $\Phi \leq 1$, and Φ is always positive.*
(2) *Every time the weight of a set s increases, either including s in the cover, or leaving it out will not increase Φ.*

Proof. (1) First, let us consider the initial value of Φ_1. If we can choose α such that $g(e_i) < 1/(2n)$, then $\Phi_1 \leq 1/2$. Initally $c_i = 0$ for all i, and $w_s = 0$ for all s. Hence, we require

$$\exp(-\alpha) <= 1/(2n)$$
$$\iff \qquad \alpha >= \log 2n.$$

Thus, choosing $\alpha = \log 2n$ ensures $\Phi_1 \leq 1/2$.
Since the initial value of $\Phi_2 = \exp(-1)/2 \leq 1/2$, we have the inital value of $\Phi \leq 1$.
It is clear that Φ is always positive.

(2) Suppose the set s has its weight augmented by δ_s, with $0 \leq \delta_s \leq 1$. We will show through a probabilistic argument that choosing the set or not choosing it causes Φ to not increase from its value before the augmentation. Let $p = \exp(-\alpha\delta_s)$. We will choose set s with probability $(1-p)$ and leave it out with probability p; we show that $E[\Phi_1]$ and $E[\Phi_2]$ do not increase. Hence, there exists a choice such that Φ does not increase.
First, we consider $E[\Phi_2]$. Let Φ_2^i denote the value of Φ_2 before the weight augmentation of s.

$$\begin{aligned} E[\Phi_2] &= \Phi_2^i((1-p)\exp(\log 2 - \alpha\delta_s) + p^2) \\ &= \Phi_2^i((1-p)2p + p^2) \\ &= \Phi_2^i p(2-p) \\ &\leq \Phi_2^i, \end{aligned}$$

since $x(2-x) \leq 1$ for any $x \in [0,1]$.
Next, we consider $E[\Phi_1]$. Let Φ_1^i denote the initial value of Φ_1.
Let $e \in s$. There are two cases: the case when $c_e = k_e - 1$ and the case when $c_e < k_e - 1$.
Suppose $c_e < k_e - 1$. Then, after inclusion of s, e will remain active; that is $e \notin C$. Denote by $g''(e)$ the value of g after augmentation and inclusion of s, $g'(e)$ the value of g after augmentation but without inclusion of s, and $g_i(e)$ the value of g before augmentation.
Claim 3. (a) $g''(e) \leq g_i(e)$
(b) $p \cdot g'(e) \leq g_i(e)$

Proof. (a) In $g''(e)$, the argument of the exponential function in $g_i(e)$ is decreased by α, since c_i has increased by 1, and increased by at most α, since $\delta_s \leq 1$.
(b) This follows directly from the definition of $g(e)$. □

Hence

$$\begin{aligned} E[g(e)] &= (1-p)g''(e) + pg'(e) \\ &\leq (1-p)g_i(e) + pg_i(e) \\ &= g_i(e) \end{aligned}$$

Now, suppose $c_e = k_e - 1$. Then, the inclusion of s will make e inactive. Hence

$$\begin{aligned} E[g(e)] &= (1-p)\cdot 0 + p(g'(e)) \\ &\leq g_i(e) \end{aligned}$$

Hence $E[\Phi] = \sum_{i \notin C} E[g(e_i)] \leq \Phi_i$. □

Theorem 3. *The solution produced by Alg. 2 is feasible and has competitive ratio $O(\log m \log n)$.*

Proof. First, we will show Alg. 2 produces a feasible solution. Since the fractional solution is feasible, we know after element e is given to the algorithm, $\sum_{s:e\in s} w_s \geq k_e$. Hence, if $g(e)$ is included in Φ_1, that is if $e \notin C$, then $\Phi_1 \geq 1$ and thus $\Phi > 1$. This contradicts Lemma 1, which shows that Φ is initially at most 1 and does not increase during Alg. 2.

Let OPT denote the optimal value of the fractional, offline problem. As shown above, the fractional algorithm produces a solution within factor $O(\log m)$ of OPT.

At the end of Alg. 2, $\Phi \leq 1$ by Lemma 1. Hence, $\Phi_2 \leq 1$, which gives

$$\begin{aligned} &\exp\left(\sum_{s\in S}(\log 2 \cdot \chi_C(s) - \alpha w_s) - 1\right) \leq 2 \\ &\Rightarrow \sum_{s\in S}\chi_C(s) \leq \log 2 + \frac{\alpha}{\log 2}\sum_{s\in S} w_s + \frac{1}{\log 2} \\ &\Rightarrow \sum_{s\in S}\chi_C(s) = O(\log n \log m) OPT. \end{aligned}$$

□

Remark. The analysis in this section with minor modifications applies in the case where elements presented to the algorithm are allowed to repeat; that is, we could use this algorithm for the the Online Set Multicover with Repetitions problem, where the coverage requirement for an element is allowed to increase as an element is presented more than once. The same competitive ratio would hold for this case.

B. Algorithm for OMSRFS

We adapt the above OSMC algorithm for the OMSRFS problem in Algorithm 3, the Online D2D Spatial Reuse Algorithm (ODSRA). As input, the ODSRA takes $\mathcal{S}$, the collection of all MIFSs in the cell. In Section IV, we will discuss a way to approximate $\mathcal{S}$. However, computing $\mathcal{S}$ is not the focus of this paper. As long as the collection $\mathcal{S}$ does not change, we can decompose changes to the cellular system into a sequence of additions and deletions to the system. We consider here a period of constant $\mathcal{S}$, and input additions and deletions one at a time to ODSRA. According to these changes, the algorithm updates its solution A.

At an abstract level, as devices enter the network, ODSRA uses the derandomized OSMC algorithm to determine to which MIFS resource blocks should be assigned. Hence, it must keep track of the value of Φ, the potential function defined in Section III. Each MIFS is assigned a weight, which is initially set to $1/m^2$, where m is the number of MIFSs. When a primary device d arrives, the algorithm checks if its resource requirement is satisfied by the sum of the weights of the MIFSs containing this device. Until this the resource requirement is satisfied by the weight sum, the weight of each MIFS containing d is doubled, and the potential function Φ determines which MIFSs should be employed in RB allocation: if inclusion of MIFS does not increase Φ, the MIFS is included, and otherwise not. When a device leaves the network, the resource requirement of k_e of primary device e may decrease. If so, it reports to the base station that the earliest RB assigned to it is no longer needed for D2D communication. When an RB is no longer in use by any of the devices in the MIFS to which it was initially assigned, it is freed. The algorithm updates the weights of the freed MIFSs and the value of Φ. Pseudocode for ODSRA is presented in Alg. 3.

Algorithm 3: Adaptation of OSMC algorithm for OM-SRFS. This algorithm is referred to as Online D2D Spatial Reuse Algorithm (ODSRA) in the text.

```
Input: A cell, with primary devices E.
Collection C of all MIFSs.
Output: A subcollection A ⊂ C.
Each element a ∈ A is assigned one RB.
begin
    A = ∅.
    X = ∅.
    C = {S_1, ..., S_m}.
    Φ = Φ_0.
    for i ∈ [m] do
        w_i = 1/m^2.
    while change c arrives to device e do
        if the resource requirement k_e of device e increases then
            X = X ∪ {e}.
            while Σ_{i:e∈S_i} w_i < k_e do
                for w_i : e ∈ S_i do
                    if w_i < 1 then
                        w_i := min{1, 2w_i}
                        if choosing S_i does not increase Φ then
                            A = A ∪ {S_i}.
        else
            // the resource requirement k_e of e decreases
            if k_e = 0 then
                X = X − {e}.
            for i ∈ [m] do
                if S_i ∈ A then
                    if S_i is no longer in use by any device in S_i then
                        A = A − {S_i}
                        w_i = 1/m^2
```

IV. Experimental results

A. Setup

The synthesized D2D system consists of a geometrical disc representing the range of BS and locations within the disc where devices are present. Initially, the system contains no devices; primary devices are added one at a time uniformly randomly to the disc. As each device d arrives, it must be assigned enough RBs to meet its communication requirements. This number, k_d, is chosen uniformly randomly from $1, \ldots, k_{max}$.

To calculate the MIFSs, we perform the following procedure. We create a graph G, with every vertex representing a device. Initially, G has no edges. Then, every pair of (i, j) of primary devices is considered, and it is determined whether i and j will interfere if each communicate with a secondary device on the same RB, according to the calculation shown in Section II-C. If there is interference, then an edge (i, j) is added to the graph G.

The MIFSs are then calculated as the maximal independent sets of G. This is only an approximation of the actual MIFSs, since the inteference of multiple signals will sum to create the total interference on device j. However, the calculation of MIFSs is discussed in [8] and is not the focus of this paper; to illustrate our algorithm, the maximal independent sets of G suffice. All experiments were run on an AMD A-Series A8-4500M at 1.9 GHz with 4 gigabytes of RAM.

B. Addition of Devices Only

1) Performance: We evaluate the performance of the algorithm ODSRA on cellular systems starting with 0 primary devices and adding one device at a time up to a maximum of 60 primary devices. We compare with the offline set multicover algorithm presented in [8].

To evaluate the quality of the allocations, we compare to an assignment with no spatial reuse – as allocation without D2D communication is currently performed within a single cell [10]. We calculate the percentage of RBs freed when compared to the no spatial reuse.

More explicitly, let N_{nsr} be the number of RBs required for the system with no spatial reuse of RBs. Let N_A be the number of RBs required after using spatial reuse algorithm A. Then, we consider the following percentage

$$P = \frac{N_{nsr} - N_A}{N_{nsr}} \cdot 100\%.$$

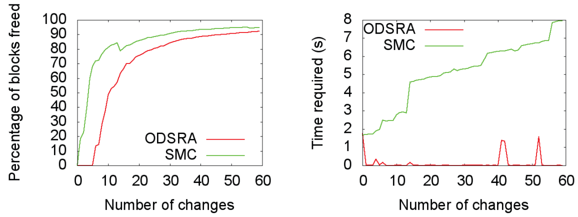
This metric is an indication of how efficiently the algorithm is spatially reusing the RBs.

The number of RBs a primary device requests is picked from a uniform distribution on $\{1, \ldots, 50\}$.

In Figure 1a, we show the number of RBs allocated by ODSRA and SMC as devices are added to the system. As expected, the SMC algorithm gives slightly higher spatial reuse. For very small numbers of devices, SMC frees up to 60% more RBs then ODSRA, but as the number of additions continues, ODSRA comparatively performs better, with by the 50th addition, the difference between the two is less than 5%. This result can be interpreted as ODSRA needing a few changes to the system in order to adapt and start giving good results.

For addition of devices, the competitive ratio of $O(\log m \log n)$ to the optimal is in effect, as in this case, the algorithm ODSRA follows the deterministic OSMC algorithm described above. This means that the algorithm is guaranteeed to perform within this ratio of an optimal algorithm which knows ahead of time which devices will be added to the system. Since in the context of a cellular system, this requires knowing the future, such an optimal algorithm cannot be implemented.

2) Running time: In Figure 1b, we show the running time of ODSRA and SMC during this experiment. Notice that the running time of ODSRA generally decreases as more devices are added, and is usually roughly 0.01 seconds, with rare spikes up to 1.5 seconds. However, the running time of SMC increases roughly linearly with the size of the system; ODSRA runs from 5 to 795 times faster than SMC, and in most cases runs hundreds of times faster.



(a) Allocation performance (b) Running time

Fig. 1: Results for addition of primary devices from 0 devices to 50.

C. Addition and Removal of Devices

Next, we consider a cellular system in which devices are both added and removed from the network. Again, the system is started from an empty state with 0 devices. Devices are added until 65 devices are in the system. Then, for each change of the system, a device is added or removed with equal probability.

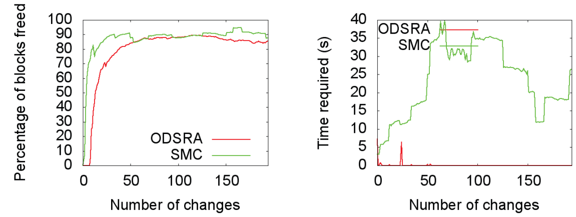
The number of secondary devices assigned to a primary device is uniformly randomly picked from $\{1, \ldots, 50\}$.

1) Performance: We use the same performance metric P defined above. In Figure 2a, we see P plotted against the number of changes made to the network. Again, as in the case of additions only, we see the allocation performance of ODSRA becomes closer to that of SMC as the number of changes increases. After the initial addition of 65 devices, the performance of ODSRA stays within 10% to the performance of SMC.

2) Running time: With the larger system of 65 primary devices, the SMC algorithm takes much longer, as seen in Figure 2b. However, ODSRA continues to run very quickly, with most iterations taking less than 0.025 seconds, improving the runtime of SMC by factors of as much as 1000.

D. Nonuniform distribution of resource requirements

Next, we relax the requirement that the number of secondary devices each primary device manages is picked from a uniform distribution. We examine cellular systems where k_d was chosen from a geometric distribution with success probability



(a) Allocation performance (b) Running time

Fig. 2: Results for addition and removal of primary devices.

$p = 0.1$, instead of a uniform distribution on $\{1, \ldots, k_{max}\}$. Thus, in this scenario, the probability that $k_e = k$ is

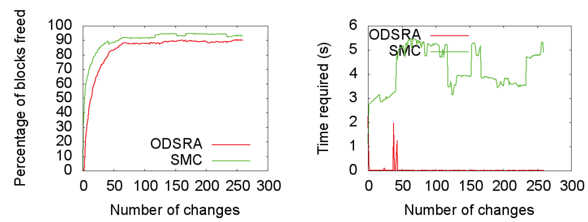
$$P[k_e = k] = (1 - p)^{k-1} p.$$

We chose this distribution as it is the discrete analogue of the exponential distribution, which is a reasonable assumption for the distribution of secondary devices.

Starting from 0 primary devices, primary devices are added until 65 devices are in the system – thereafter, primary devices are added or removed with equal probability.

1) Performance: In Figure 3a, we show the performance with the metric P. The performance of ODSRA stays within 10% of that of SMC.

2) Running time: The running time of ODSRA once again is close to 0 seconds while SMC takes from 3 to 6 seconds for each change, as shown in Figure 3b. Twice, the running time of ODSRA exceeded 1 second; this is most likely because of resetting weights of MIFSs after removal of primary of devices. If these MIFSs are reallocated, it can take some time for their weights to be increased.



(a) Allocation performance (b) Running time

Fig. 3: Results from choosing number of secondary devices from geometric distribution.

E. Network throughput

We examine the performance of the algorithms ODSRA and SMC from the perspective of network throughput. Our model is an LTE system with 10 MHz of channel bandwidth. Then, there are 50 RBs to be allocated. We calculate the throughput as the sum of the data rates on all receiving devices. Each device allocated an RB receives at a rate of 1 Mbps through that RB.

Figure 4 shows the throughput achieved by the allocation of ODSRA, SMC, and an allocation that uses no spatial reuse,

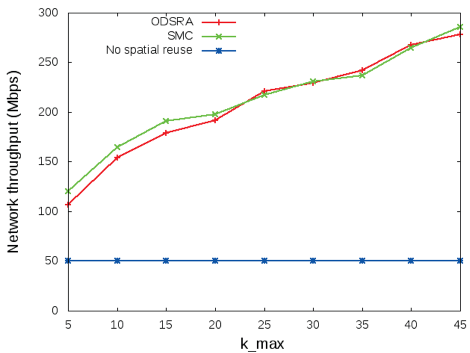


Fig. 4: Network throughput plotted against the maximum number of secondary devices a primary device is allowed to manage.

averaged over 10 randomly generated cellular systems, plotted against the maximum number of secondary devices that a primary device may service, k_{max}. The number of primary devices is fixed at 40; there are no online changes to the systems after they have been constructed. Since the number of secondary devices is uniformly randomly chosen from $\{1, \ldots, k_{max}\}$, increasing k_{max} has the effect of increasing the expected number of D2D links.

If no spatial reuse is employed, there is no benefit from D2D communication, as expected. However, we see that ODSRA and SMC obtain similar throughput values from their spatial-reuse allocations, ODSRA even exceeding SMC for some values of k_{max}.

V. Related Work

Spatial reuse of radio resources has been studied in the context of cellular systems. Without D2D communication, spatial reuse has been studied among different cells [10] [9]. The online allocation algorithm given in this work may find application in spatial reuse among heterogeneous cells. D2D communication within a single cell was introduced in [4]. Maximizing the spatial reuse for D2D communication has been studied in [12], where a mixed integer nonlinear programming was formulated for the problem, and a greedy heuristic for the nonlinear programming was presented. In [8], approximation for the problem was presented using a greedy set multicover algorithm. This improved the results of [12], and is the approach to which we compare ODSRA in this work.

An online, deterministic algorithm for the set cover problem, where each element must be covered only once, was presented in [1] with competitive ratio $O(\log n \log m)$. It uses a potential function for the derandomization, but this potential function is quite different from the one in this work. Our potential function is adapted from the one in [3] for the online set cover problem. In [3], an online fractional algorithm is given for covering linear programs with box constraints. However, the online fractional algorithm for the set multicover LP presented here differs from the more complicated algorithm in [3].

In [2], Alon et. al. present an online, bicriteria algorithm for the online set cover with repetitions problem. This problem is similar to online set cover, but the same element can be given to the algorithm multiple times. The bicriteria algorithm in [2] achieved an $O(\log n \log m)$-competitive ratio while violating the coverage requirements by an $(1 - \epsilon)$ factor, where $\epsilon > 0$. Our deterministic, $O(\log n \log m)$-competitive algorithm for online set multicover will approximate online set multicover with repetitions with this same ratio, without violating any constraints.

VI. Conclusions

We considered the problem of spatial reuse of radio resources in D2D context in an online setting. This led us to consider the OSMC problem, for which we formulated both a randomized and a deterministic algorithm with $O(\log n \log m)$-competitive ratio. Using the OSMC algorithm as a baseline, we formulated an online algorithm ODSRA to approximate the maximization of spatial reuse of resource blocks. We compared the performace of ODSRA to the offline set multicover approach in [8], and observed decreases in running time by factors of up to 1000 together with similar performance in terms of total throughput and efficiency of RB allocation.

Our algorithm ODSRA requires knowledge of the MIFSs $\mathcal{S}$ is required beforehand, and changes to $\mathcal{S}$ are not incorporated in the algorithm. Future work would include modifying ODSRA to update $\mathcal{S}$ as the system changes.

VII. Acknowledgement

This work partially supported by NSF CAREER 0953284.

References

[1] N. Alon, B. Awerbuch, Y. Azar, N. Buchbinder and J. Naor. "The online set cover problem". *SIAM J. Comput.* 39 (2009), 361-370.

[2] N. Alon, S. Gutner and Y. Azar. "Admission control to minimize rejections and online set cover with repetitions". *ACM Transactions on Algorithms*. 6(1), 2009.

[3] N. Buchbinder and J. Naor. "Online Primal-Dual Algorithms for Covering and Packing Problems." *Mathematics of Operations Research*, Vol. 34, No. 2, May 2009, p 270-286.

[4] K. Doppler, M. Rinne, C. Wijting, C. B. Ribeiro, and K. Hugl, "Device-to-device communication as an underlay to LTE-advanced networks," *IEEE Commun. Mag.*, vol 47, no. 12, pp. 42–49, Dec. 2009.

[5] W. H. Chin, Z. Fan, and R. J. Haines,"Five disruptive technology directions for 5G," IEEE Communications Society Publications Contents Digest, Mar. 2014.

[6] H. Fu and D. I. Kim, "Analysis of Throughput and Fairness with Downlink Scheduling in WCDMA Networks," *IEEE Transactions on Wireless Communications*, Vol. 5, No. 8, August 2006.

[7] Krogfoss, B., Hanson, G. and Vale, R.J., "Impact of consumer traffic growth on mobile and fixed networks: Business model and network quality impact." Bell Labs Tech. J., 16: 105120. May 2011.

[8] Lee, D. H., Choi K. W., Jeon, W. S., Jeong, D. G. "Resource Allocation Scheme for Device-to-Device Communication for Maximizing Spatial Reuse". *IEEE Wireless Communications and Networking Conference*, 2013.

[9] D. L. Lopez-Perez, A. Valcarce, G. de la Roche, and J. Zhang. "OFDMA Femtocells: A Roadmap on Interference Avoidance".

[10] T. D. Novlan, R. K. Ganti, A. Ghosh, and J. G. Andrews, "Analytical Evaluation of Fractional Frequency Reuse of OFDMA Cellular Networks," *IEEE Transactions on Wireless Communications*, Vol. 10, no. 12, December 2011.

[11] V. V. Vazirani, *Approximation Algorithms*. Springer-Verlag, 2003.

[12] M. Zulhasnine, C. Huang, and A. Srinivasan, Efficient resource allocation for device-to-device communication underlaying LTE network, *Proc. IEEE WiMob 2010*, Oct. 2010, pp. 368375.