2023 19th International Conference on Network and Service Management (CNSM)

Online Utilization Maximization in Resource Allocation with Minimum Service Guarantees

Dor Harris
Technion, Israel
dorharris@cs.technion.ac.il

David Naori
Technion, Israel
dnaori@cs.technion.ac.il

Danny Raz
Technion, Israel
danny@cs.technion.ac.il

***Abstract*—The natural objective of resource allocation algorithms is twofold: On one hand, to maximize utilization and on the other hand to allow a fair share to all users. The actual meaning of "fair" in this context is manifold; we propose to address fairness in a simple and natural way by guaranteeing a minimum level of service to every user.**

We develop new competitive online algorithms for this new resource allocation with mandatory service problem and analyze their performance guarantees both in the adversarial-order and random-order online models. We also show that having prior knowledge about the request distribution can be beneficial. We accomplish this by analyzing a probabilistic relaxation of the mandatory service criterion.

We study the practical implementation of these theoretical algorithms in the context of online cell selection in access networks. In this setting, mobile users request service and the network needs to assign a relevant cell (or cells) to provide it. We conduct extensive simulations to evaluate the performance of our algorithms in realistic conditions. The results suggest that our new algorithms perform better than applicable adaptations of the commonly used heuristics.

I. Introduction

Efficient utilization of resources is becoming a critical aspect of many systems due to economical drives to reduce costs and be more competitive. Resource allocation algorithms have two natural goals. The first is allowing all users to receive a fair amount of service. The second goal is maximizing the utilization, the overall amount of usage, or the revenue for the service operator.

Satisfying both these goals simultaneously is challenging and in many cases, optimizing one aspect cannot be done without compromising the other. This problem is even more challenging in online settings, where service requests are not known in advance, but rather arrive over time and the provider must decide immediately and irreversibly if and how to serve each request.

While the definition of utilization and the rigorous meaning of maximizing it are straightforward, this is not the case for fairness. One common interpretation is based on *max-min fairness*, where the goal is to maximize the minimal allocation of a user. More generally, an allocation is "max-min fair" if it is impossible to improve the allocation of a user by hurting only users with larger allocations (see for example [1]).

A well-known approach to address this trade-off between efficiency and fairness is *proportional fairness* obtained by maximizing a logarithmic utility function [2]. Another common way to combine both goals is to consider all feasible allocations that satisfy a set of fairness constraints and to select an allocation that optimizes utilization (see for example [3]–[5]). In this approach, the overall outcome strongly depends on the exact definition of the fairness constraints, and the restrictions it puts on the overall utilization.

Nevertheless, in various practical scenarios, existing solutions are insufficient. In many applications, a minimum service level is required by users, or due to regulated obligations and the service provider must allocate resources for these occurrences. This is the case, for example, for voice calls where a call cannot be made with insufficient bandwidth. Thus, getting an allocation below that minimal level will not allow the user to make an emergency call (911).

To account for these common practical scenarios, we consider a new way of combining fairness and utilization optimization. The main idea is to guarantee a minimal service level to each customer, which we call *mandatory service*, and to decide on the allocation of the remaining available resources in a way that maximizes utilization.

We apply this approach to study a generalization of the fundamental online maximum fractional matching problem in bipartite graphs, which has many applications from ad allocation [6] to maximizing throughput in multi-queue switches [7] and virtual machine placement [8]. On one side of the graph, we have users or clients, and on the other side, we have servers. Each server has a capacity (the amount of service it can provide), and each client has a *mandatory demand* describing the minimal amount of service it requires and also a *total demand* representing the amount of service it would like to have. Clients arrive online and the goal is to provide the mandatory demand to all clients and allocate the remaining service capacity in a way that maximizes the overall utilization.

We distinguish between two possible service types, *serve-by-one* in which each client can be served by one server only (this is the case in many current services), and *serve-by-many* that allows each client to be served by multiple servers (and is expected to be deployed in future services). Naturally, the serve-by-many setting provides more flexibility which can be utilized to achieve improved performance.

We begin by developing online algorithms that reserve resources throughout their execution to accommodate for *all* mandatory demands. We call such algorithms *fully-compliant*.

978-3-903176-59-1 ©2023 IFIP

For the serve-by-one case, we follow a greedy approach that provides service from a server that can provide the largest portion of the request's demand from its free and unreserved capacity (among the servers that can provide service to this request). We refer to this algorithm by *Greedy with Reservations* (GwR). For the serve-by-many case, we use a water-filling based algorithm that splits the service among the adjacent servers with the largest fraction of free and unreserved capacity. We refer to this algorithm by *Water-Filling with Reservations* (WFwR).

To allow for fully-compliant algorithms, the serving capacity of each server must be at least the overall amount of mandatory demands. It turns out that the competitive-ratio of GwR and WFwR depend on the ratio between the minimum capacity of a server and the overall amount of mandatory demands. We call this ratio the *capacity-ratio* and denote it by ρ. We show that as the value of ρ increases, the competitive-ratios of the algorithms improve, and approach their respective performance in the setting without mandatory demands. More concretely, we prove a worst-case competitive-ratio of $\frac{1}{2}\left(1-\frac{1}{\rho}\right)$ for GwR, and $\left(1-\frac{e^{1/\rho}}{e}\right)$ for WFwR.

While the standard worst-case competitive analysis provides robust performance guarantees, it might be too pessimistic for many realistic conditions. Therefore, we also analyze our algorithms in the random order model – a prominent relaxation of the worst-case model, in which the requests arrive in a uniformly random order instead of an adversarial order. We show that GwR and WFwR achieve improved random-order competitive-ratios. In particular, both algorithms obtain constant random-order competitive-ratios for the interesting case of $\rho = 1$. In contrast, we show that no online algorithm can achieve a constant worst-case competitive-ratio for $\rho = 1$.

In many practical cases, the operator has prior knowledge of the expected requests. Such knowledge can be acquired by processing past data or building statistical models of the demand. Using prior knowledge can be very beneficial since we can reduce the amount of capacity we keep to serve the mandatory demands of future requests. However, to prove rigorous bounds, we need to relax the fully-compliant requirement. That is, we replace the requirement that all mandatory demands must be satisfied with a probabilistic requirement, which we call *approximately-compliant*, saying that all mandatory demands must be satisfied with high probability.

We formally analyze this case using the online i.i.d. model [6]. In this model, the distribution of client requests is known in advance, and the actual requests are drawn from this distribution (see Section IV for a formal description). We develop an algorithmic scheme that takes an online algorithm A for the simpler problem without the requirement for mandatory services, (for example A may be greedy or water-filling), and generate an online algorithm A', based on A, that serves all mandatory demands with high probability, and pays a small cost in the competitive-ratio of A. Also, A' maintains the service type of A (serve-by-one or serve-by-many).

To study the applicability of our theoretical results, we evaluate the expected performance of our algorithms in access network scenarios. This is done using extensive simulations over realistic data. The results indicate that our new algorithms perform well over a variety of settings, and outperform applicable adaptations of commonly used heuristics. Unsurprisingly, the performance in the serve-by-many case is better than in the serve-by-one case, and using prior knowledge can be beneficial in these settings.

Our main contributions are as follows.

- We define an optimization criterion that provides a new trade-off between fairness and utilization maximization.
- We provide proven performance guarantees for two online algorithms both in the adversarial and random order arrival models.
- We define the notion of approximately-compliant algorithms and show that statistical prior knowledge regarding the expected requests can provide better guarantees for online algorithms.
- We show that our new algorithms can be used in practical realistic scenarios (cell selection in access networks) resulting in better performance.

II. Preliminaries

In the *Resource Allocation with Mandatory Service* problem (RAMS), we are given a bipartite graph $\mathrm{G} = (\mathrm{C}, \mathrm{A}, \mathrm{E})$, the vertices in C are called *clients* and the vertices in A are called *servers*. We denote the number of clients by $n = |\mathrm{C}|$ and the number of servers by $m = |\mathrm{A}|$. Each server $a_j \in \mathrm{A}$ has a capacity $c_j \in \mathbb{N}$. Each client $u_i \in \mathrm{C}$ has a mandatory demand $\mu_i \in (0, 1]$ and a total demand $s_i \geq \mu_i$. A server $a_j \in \mathrm{A}$ can allocate resources to its neighbors in G, i.e., to clients in $N(a_j) = \{u_i \in \mathrm{C} : (u_i, a_j) \in \mathrm{E}\}$. We denote by $x_{i,j}$ the amount of resources server a_j allocates client u_i (where $u_i \in N(a_j)$). The total amount of resources a_j allocates must not exceed its capacity c_j, that is, $\sum_{i\in N(a_j)} x_{i,j} \leq c_j$. For each client $u_i \in \mathrm{C}$, the amount of resources allocated to u_i must be at least μ_i (mandatory service) and must not exceed its total demand s_i, that is, $\sum_{j\in N(u_i)} x_{i,j} \in [\mu_i, s_i]$. Hence, we assume that $N(u_i) \neq \emptyset$ for all $u_i \in \mathrm{C}$. The goal is to find a feasible allocation that maximizes the total amount of allocated resources, that is, $\sum_{a_j\in \mathrm{A}} \sum_{u_i\in \mathrm{C}} x_{i,j}$.

We use the following LP formulation of RAMS:

$$\begin{aligned}
\text{maximize:} \quad & \sum\nolimits_{(u_i,a_j)\in \mathrm{E}} x_{i,j} && \text{(P)}\\
\text{subject to:} \quad & \sum\nolimits_{u_i\in N(a_j)} x_{i,j} \leq c_j, \quad a_j \in \mathrm{A} \\
& \sum\nolimits_{a_j\in N(u_i)} x_{i,j} \leq s_i, \quad u_i \in \mathrm{C} \\
& \sum\nolimits_{a_j\in N(u_i)} x_{i,j} \geq \mu_i, \quad u_i \in \mathrm{C} \\
& x_{i,j} \geq 0, \qquad (u_i, a_j) \in \mathrm{E}.
\end{aligned}$$

And the dual LP:

$$\begin{aligned}
\text{minimize:} \quad & \sum_{a_j\in \mathrm{A}} c_j\alpha_j + \sum_{u_i\in \mathrm{C}} s_i\beta_i - \sum_{u_i\in \mathrm{C}} \gamma_i\mu_i && \text{(D)}\\
\text{subject to:} \quad & \alpha_j + \beta_i - \gamma_i \geq 1, \qquad (u_i, a_j) \in \mathrm{E}
\end{aligned}$$

$$\alpha_j, \beta_i, \gamma_i \geq 0, \qquad\qquad a_j \in \mathrm{A}, u_i \in \mathrm{C}.$$

We consider two variants of RAMS. The first is the *serve-by-many* RAMS in which a client may be served by multiple servers (as formulated in the LP (P) above). The second is the *serve-by-one* RAMS, in which each client must be served by one server alone. We note that in the offline version of the problem, the serve-by-many RAMS can be solved optimally in polynomial time, but it is NP-hard to optimally solve the serve-by-one RAMS. [1]

In the online version of the problem, the online player is given the server set A and the number of clients n upfront. Then, the clients in C arrive online one by one. Let $(u_1, \ldots, u_n)$ denote the online sequence. When a client $u_\ell \in \mathrm{C}$ arrives (at online round ℓ), its demands μ_ℓ, s_ℓ and its incidents edges are revealed. Then, the online player must decide immediately (before the arrival of $u_{\ell+1}$), and irrevocably, how much resources to allocate u_ℓ from each of its neighboring servers. As mentioned before, in the serve-by-one setting, only one of the neighboring servers can allocate resources to u_ℓ, and in the serve-by-many setting, multiple servers may allocate resources to u_ℓ.

Let ALG be an online algorithm for RAMS. For an input instance I, let $\mathrm{ALG}(I)$ be the amount of resources ALG allocates on I, and let $\mathrm{OPT}(I)$ be the amount of resources allocated in an optimal (offline) solution for I (that is, the value of an optimal solution for the LP (P)). We define the *minimum-capacity* of I by $c_{\min}(I) = \min_{a_j \in \mathrm{A}} c_j$ and the *capacity-ratio* of I by $\rho(I) = c_{\min}(I)/n$. We express the competitive-ratio of our algorithms in terms of the capacity-ratio.[2]

In this work we study algorithms in three different online models. The first is the standard worst-case competitive-analysis (also known as the adversarial-order model). In this model, a deterministic online algorithm ALG for RAMS is called $\lambda(\rho)$-competitive, if for any input instance I (and any arrival order of the clients), $\mathrm{ALG}(I) \geq \lambda(\rho(I)) \cdot \mathrm{OPT}(I)$.[3] The second model that we consider is the random-order model, in which the adversary cannot choose the arrival order of the clients. Instead, the clients arrive in a uniformly random order. In this model, an algorithm ALG is called λ-competitive, if on any input instance I, $\mathbb{E}\left[\mathrm{ALG}(I)\right] \geq \lambda \cdot \mathrm{OPT}(I)$, where the expectation is taken over the random arrival order of the clients. Finally, to account for prior knowledge, we consider the online i.i.d. model. We defer the definition of the problem in this model to Section IV.

III. Fully-Compliant RAMS

For $r \in \mathbb{R}_{\geq 0}$, we say that an online algorithm ALG for RAMS is *fully-compliant* for capacity-ratio r, if on any input instance I with $\rho(I) \geq r$, ALG always serves the mandatory demands of all clients (with probability 1).

A. Worst-case competitive-analysis (adversarial order)

We begin by deriving a lower bound on the amount of reserved resources that any fully-compliant online algorithm for RAMS must maintain. The following observation guides us in the design of fully-compliant online algorithms for RAMS.

Lemma III.1. *Let* ALG *be a fully-compliant online algorithm for* RAMS *for capacity-ratio r. Then, on any input instance I with n clients and $\rho(I) \geq r$, at the end of round $1 \leq \ell \leq n$,* ALG *leaves at least $n - \ell$ free resources in each of the servers (with probability* 1*).*

Proof. Assume by contradiction that there is an input instance I with $\rho(I) \geq r$ and server $a_j \in \mathrm{A}$ for which ALG leaves less than $n - \ell$ free resources in a_j at the end of online round ℓ with positive probability $p > 0$. Then, we can construct an instance I' identical to I on the first ℓ clients, and with the same server set, but in I', the clients that arrive at online rounds $\ell+1, \ldots, n$ are adjacent only to a_j and each client has a mandatory demand of 1. Thus, ALG on I' fails to serve all mandatory demands of the clients that arrive after round ℓ with probability at least $p > 0$. Also note that $\rho(I') = \rho(I) \geq r$ and so, ALG is not fully-compliant for capacity-ratio r. □

An immediate corollary of Lemma III.1 is that there are no fully-compliant online algorithms for capacity-ratio $r < 1$. In what follows, we design fully-compliant online algorithms for capacity-ratio $r \geq 1$.

*1) **Serve-by-One** **RAMS**:* In the *serve-by-one* RAMS, each client must be served by only one server. We modify the classical greedy algorithm to guarantee that the mandatory demand of each client is served. To this end, we reserve resources throughout the execution of the algorithm to accommodate the mandatory demand of future clients. More concretely, the algorithm begins by reducing the server capacities and reserving a unit of resources for each client from each server. Clearly, this requires that each server has a capacity of at least n, i.e., $c_{\min} \geq n$ and thus $\rho \geq 1$. Then, when a client arrives, the algorithm releases one unit of resource from the reserved resources of each server and executes the standard greedy algorithm to serve the client with the new, modified server capacities. For a formal description see Algorithm 1.

Theorem III.2. *Algorithm* GwR *is fully-compliant for capacity-ratio* 1*, and is $\frac{1}{2}\left(1 - \frac{1}{\rho}\right)$-competitive for serve-by-one* RAMS.

Proof. First, to see that the algorithm is fully-compliant for capacity-ratio 1, observe that when a client u_ℓ arrives, each server has free capacity of at least $1 \geq \mu_\ell$. Additionally, $N(u_\ell) \neq \emptyset$, and since the algorithm serves u_ℓ with a server that maximizes the amount of resources allocated to u_ℓ, u_ℓ is served with at least μ_ℓ resources.

To prove the competitive-ratio, we follow the online dual-fitting technique by Buchbinder et al. [9]. When a client u_i is assigned an amount of $x_{i,j}$ by a server a_j, we increase the dual variables α_j and β_i by $\Delta\alpha_j = (1/c_j) \cdot (x_{i,j}/2)$ and

[1] Given a polytime algorithm for RAMS, one can construct a polytime algorithm for the known NP-hard subset sum problem. We omit the details due to lack of space.

[2] When I is clear from the context, we drop I from the notation and write, for example, ρ instead of $\rho(I)$.

[3] In this work we analyze only deterministic online algorithms for RAMS.

Algorithm 1: GREEDY WITH RESERVATIONS (GWR)

```
for each server $a_j \in A$ do
    $c'_j \leftarrow c_j - n$    // reduced capacities (reserving resources)
for each client $u_\ell$ that arrives at round $\ell$ do
    for each server $a_j \in A$ do
        /* assuring mandatory service (releasing resources) */
        $c'_j \leftarrow c'_j + 1$
    For $a_j \in N(u_\ell)$, let $q(\ell, j) = \min\left\{s_\ell, c'_j - \sum_{k=1}^{\ell-1} x_{k,j}\right\}$ be the maximum amount of resources $a_j$ can allocate $u_\ell$
    Choose $a_{j_\ell} \in \arg\max_{a_j \in N(u_\ell)}\{q(u_\ell, j)\}$    // arbitrarily
    $x_{\ell,j_\ell} \leftarrow q(\ell, j_\ell)$
```

$\Delta\beta_i = (1/s_i) \cdot (x_{i,j}/2)$. We also set $\gamma_i = 0$ for all $i \in$ C. Observe that according to this update rule, the dual objective D and the primal objective P are always equal. Therefore, in order to show that GWR is λ-competitive, it suffices to show that $\alpha_j + \beta_i - \gamma_i \geq \lambda$ for all $(u_i, a_j) \in$ E.

Fix $(u_\ell, a_j) \in$ E and consider the point in time right before u_ℓ is served. Let $w_j = \sum_{k=1}^{\ell-1} x_{k,j}$ be the water level (the load) of a_j at this point in time. If u_ℓ is completely served, we have $\alpha_j + \beta_\ell \geq \beta_\ell = \frac{1}{s_\ell} \cdot \frac{s_\ell}{2} = \frac{1}{2}$. Otherwise, if only a p-fraction of the total demand of u_ℓ is served, i.e., $p \cdot s_\ell$, then we have $w_j \geq c_j - n - p \cdot s_\ell$ (otherwise, u_ℓ would have been served by a_j), and therefore, $\alpha_j + \beta_\ell \geq \frac{1}{c_j} \frac{c_j - n - p \cdot s_\ell}{2} + \frac{p}{2} \geq \frac{1}{2} - \frac{n}{2c_j} + \frac{p}{2} - \frac{p \cdot s_\ell}{2c_j} \geq \frac{1}{2}\left(1 - \frac{1}{\rho}\right)$, where in the last inequality we used the fact that $s_\ell / c_j \leq 1$, and $n/c_j \leq n/c_{\text{min}} = 1/\rho$. □

Next, we show that GWR is an optimal deterministic fully-compliant algorithm for the serve-by-one RAMS.

Theorem III.3. *The competitive-ratio of any deterministic fully-compliant online algorithm for serve-by-one* RAMS *is at most* $\frac{1}{2}\left(1 - \frac{1}{\rho}\right) + o(1)$.

Proof. Fix a deterministic algorithm ALG for serve-by-one RAMS. Consider an instance with n servers A $= \{a_1, \ldots, a_n\}$ each with a capacity of $z \geq n$ (i.e., $c_j = z$ for all $j \in [n]$), and n clients C $= \{u_1, \ldots, u_n\}$. The total demand of each client in $\{u_1, \ldots, u_{2\sqrt{n}}\}$ is z, and the total demand of each client in $\{u_{2\sqrt{n}+1}, \ldots, u_n\}$ is 1. Also, the mandatory demand of each client is 1. For $i \leq \sqrt{n}$, u_i is connected only to two servers: a_i and $a_{\sqrt{n}+i}$.

Now, since ALG must serve each client by one server, each u_i must be served by either a_i or $a_{\sqrt{n}+i}$. Let $a_{j_1}, \ldots, a_{j_{\sqrt{n}}}$ be the servers that ALG uses to serve $u_1, \ldots, u_{\sqrt{n}}$. Now, the online sequence proceeds with the arrival of $u_{\sqrt{n}+1}, \ldots, u_{2\sqrt{n}}$ where each u_i is connected only to a_{j_i}. Finally, for $i \geq 2\sqrt{n} + 1$, u_i is connected only to a_i.

By Lemma III.1, at the end of online round $1 \leq \ell \leq n$, ALG must keep at least $n - \ell$ free resources in each server. Therefore, ALG allocates to $u_1, \ldots, u_{2\sqrt{n}}$ a total of at most $\sqrt{n} \cdot (z - (n - 2\sqrt{n}))$ resources, and for $u_{2\sqrt{n}+1}, \ldots, u_n$ the algorithm allocates a total of at most $n - 2\sqrt{n} < n$ resources. Therefore, we have ALG $\leq \sqrt{n} \cdot (z - n + 2\sqrt{n}) + n$, while OPT $\geq 2\sqrt{n} \cdot z$, as $u_1, \ldots, u_{2\sqrt{n}}$ can be fully served by $a_1, \ldots, a_{2\sqrt{n}}$. Hence, we get that $\frac{\text{ALG}}{\text{OPT}} \leq \frac{\sqrt{n} \cdot \left(z - n + 2\sqrt{n}\right) + n}{2\sqrt{n} \cdot z} = \frac{1}{2}\left(1 - \frac{n}{z}\right) + \frac{3\sqrt{n}}{2z} \leq \frac{1}{2}\left(1 - \frac{1}{\rho}\right) + o(1)$ where the last inequality follows from the fact that $\rho = c_{\text{min}}/n = z/n$ and $z \geq n$. □

*2) **Serve-by-Many** RAMS:* In the *serve-by-many* RAMS, each client may be served by multiple servers. We modify the classical water-filling algorithm to guarantee that the mandatory demand of each client is served. We follow the same reservation technique we used in Algorithm GWR. For a formal description see Algorithm 2. Note that the function $g : [0, 1] \to [0, 1]$ is used only for the analysis and we explicitly define it later on. Also, the dual variables α_j, β_i, and γ_i are used only for the analysis.

Algorithm 2: WATER-FILLING WITH RESERVATIONS (WFWR)

```
for each server $a_j \in A$ do
    $c'_j \leftarrow c_j - n$                // reduced capacities
    $\alpha_j \leftarrow 0$             // used only for the analysis
for each client $u_\ell$ that arrives at round $\ell$ do
    for each server $a_j \in A$ do
        /* assuring mandatory service */
        $c'_j \leftarrow c'_j + 1$
    $x_{\ell,j} \leftarrow 0$ for all $a_j \in N(u_\ell)$
    $\beta_\ell, \gamma_\ell \leftarrow 0$             // used only for the analysis
    Let $x_\ell$ denote the total amount of resources allocated to $u_\ell$, i.e., $\sum_{a_j \in N(u_\ell)} x_{\ell,j}$
    For $a_j \in$ A, let $w_j$ be the water-level of $a_j$: $w_j = \sum_{k=1}^{\ell-1} x_{k,j}$
    while $x_\ell \leq s_\ell$ and there is $a_j \in N(u_\ell)$ with $w_j < c'_j$ do
        allocate a $dx$ amount to each $x_{\ell,j}$ for $j \in \arg\min_{a_j \in N(u_\ell)}\{w_j/c_j\}$
        if $x_{\ell,j}$ is increased by $dx$ then
            // Used only for the analysis
            Increase $\alpha_j$ and $\beta_\ell$ as follows:
            $d\alpha_j = \frac{1}{c_j} g(w_j/c_j)dx$
            $d\beta_\ell = \frac{1}{s_\ell}(1 - g(w_j/c_j))dx$
```

The proofs of Theorem III.4 and Theorem III.5 are omitted due to lack of space and are available in the full version of the paper.

Theorem III.4. *Algorithm* WFWR *is fully-compliant for capacity-ratio* 1*, and is* $\left(1 - \frac{e^{1/\rho}}{e}\right)$*-competitive for the serve-by-many* RAMS.

Observe that for $\rho = 1$, Theorem III.4 provides no guarantee on the performance of WFWR. The next theorem shows that in fact, no online algorithm can achieve a constant worst-case competitive-ratio for $\rho = 1$.

Theorem III.5. *There is an infinite sequence of instances* $I_1, I_2, \ldots$*, with* $\rho = 1$*, such that the competitive-ratio of any fully-compliant online algorithm* ALG *on* I_n *approaches* 0 *as* $n \to \infty$.

We note that the difficulty of the instances we construct in the proof of Theorem III.5 relies on the worst-case arrival order of the clients. Next, we show that when the clients arrive in random order, both GwR and WFwR provide better performance guarantees. In particular, we show that GwR and WFwR attains constant competitive-ratios for $\rho = 1$ of $1/4$ and $(3/2 - 2/\sqrt{e}) \approx 1/3.48$, respectively.

B. The random-order model

In the random-order model, the adversary cannot choose the arrival order of the clients. Instead, the clients arrive online in a uniformly random order. For an instance I with $\mathrm{C} = \{u_1, \dots, u_n\}$ let $(u_{i_1}, \dots, u_{i_n})$ denote the online sequence where $(i_1, \dots, i_n)$ is a uniformly random permutation of $[n]$, and u_{i_ℓ} arrives at online round ℓ. In the random-order model an algorithm ALG is called λ-competitive, if on any input instance I, $\mathbb{E}[\mathrm{ALG}(I)] \geq \lambda \cdot \mathrm{OPT}(I)$, where the expectation is taken over the arrival order of the clients.

Theorem III.6. *Algorithm* GwR *is fully-compliant for capacity-ratio* 1*, and is* $\frac{1}{2}\left(1 - \frac{1}{2\rho}\right)$*-competitive for the serve-by-one* RAMS *in the random-order model.*

The proof of Theorem III.6 is omitted due to lack of space and is available in the full version of the paper.

Theorem III.7. *Algorithm* WFwR *is fully-compliant for capacity-ratio* 1*, and is* $\left(\frac{1}{2} + \rho \cdot \left(1 - e^{-1/2\rho}\right) - e^{1/2\rho - 1}\right)$*-competitive for the serve-by-many* RAMS *in the random-order model.*

Proof. First, the proof of Theorem III.4 shows that Algorithm WFwR is fully-compliant for capacity-ratio 1. It remains to bound the competitive-ratio of the algorithm. Using the randomized primal-dual technique by Devanur et al. [10] (Lemma 2.1), in order to show that the algorithm is λ-competitive it suffices to show the primal and dual objectives are always equal and that for all $(u_i, a_j) \in \mathrm{E}$, $\mathbb{E}[\alpha_j + \beta_i + \gamma_i] \geq \lambda$.

By the update rules of the primal and dual variables, the dual objective D and the primal objective P are always equal. We define $g : [0,1] \to [0,1]$, by $g(x) = \min\left\{e^{x - \left(1 - \frac{1}{2\rho}\right)}, 1\right\}$.

Fix $(u_i, a_j) \in \mathrm{E}$ and round $\ell \in [n]$. When u_i arrives at round ℓ, i.e., $u_{i_\ell} = u_i$ (which happens with probability $1/n$), we have the following cases. If $w_j \geq c_j - n + \ell$, we have

$$\begin{aligned} \alpha_j &\geq \frac{1}{c_j} \int_0^{c_j - n + \ell} g(x/c_j)dx \geq \frac{1}{c_j} \int_0^{c_j - (n-\ell)c_j/c_{\min}} g(x/c_j)dx \\ &\geq \frac{1}{c_j} \int_0^{c_j\left(1 - \frac{1}{\rho} - \frac{\ell}{c_{\min}}\right)} g(x/c_j)dx \\ &\geq \begin{cases} e^{-\frac{1}{2\rho} + \frac{\ell}{c_{\min}}} - e^{\frac{1}{2\rho} - 1} & 1 - \frac{1}{\rho} - \frac{\ell}{c_{\min}} \leq 1 - \frac{1}{2\rho} \\ 1 - e^{\frac{1}{2\rho} - 1} & \text{otherwise,} \end{cases} \end{aligned} \tag{1}$$

where in the second inequality we used the fact that $c_j - n + \ell \geq c_j - (n - \ell)c_j/c_{\min}$ as $c_j/c_{\min} \geq 1$. In the last inequality we discard the (non-negative) value obtained by $x > 1 - \frac{1}{2\rho}$ from the integral. Otherwise, $w_j < c_j - n + \ell$, and therefore, u_i must be fully served by servers whose fraction of allocated resources is at most w_j/c_j (as the algorithm always allocates resources from the servers with the minimum fraction of resources allocated from their capacity). Hence, we have

$$\begin{aligned} \alpha_j + \beta_i &\geq \frac{1}{c_j} \int_0^{w_j} g(x/c_j)dx + \frac{1}{s_u} \int_0^{s_u} (1 - g(w_j/c_j))dx \\ &= \frac{1}{c_j} \int_0^{w_j} g(x/c_j)dx + 1 - g(w_j/c_j) = 1 - e^{\frac{1}{2\rho} - 1}. \end{aligned}$$

Observe that the lower bound we obtained in Inequality (1) is upper bounded by $1 - e^{\frac{1}{2\rho} - 1}$ for all $\ell \in [n]$, therefore we can use the weaker lower bound in (1) to lower bound $\alpha_j + \beta_i$ in both cases. Now, we take the expectation over the arrival order of the clients and get that

$$\begin{aligned} &\mathbb{E}[\alpha_j + \beta_i] \\ &\geq \frac{1}{n} \sum_{\ell=1}^{n/2} \left(e^{-\frac{1}{2\rho} + \frac{\ell}{c_{\min}}} - e^{\frac{1}{2\rho} - 1}\right) + \frac{1}{n} \sum_{\ell=n/2+1}^{n} \left(1 - e^{\frac{1}{2\rho} - 1}\right) \\ &= \frac{1}{n} e^{-\frac{1}{2\rho}} \cdot e^{\frac{1}{c_{\min}}} \frac{e^{\frac{n}{2c_{\min}}} - 1}{e^{\frac{1}{c_{\min}}} - 1} - \frac{1}{2} e^{\frac{1}{2\rho} - 1} + \frac{1}{2}\left(1 - e^{\frac{1}{2\rho} - 1}\right) \\ &= \frac{1}{n} \frac{e^{\frac{1}{c_{\min}}}}{e^{\frac{1}{c_{\min}}} - 1} \cdot \left(1 - e^{-\frac{1}{2\rho}}\right) + \frac{1}{2} - e^{\frac{1}{2\rho} - 1} \\ &\geq \frac{c_{\min}}{n} \cdot \left(1 - e^{-\frac{1}{2\rho}}\right) + \frac{1}{2} - e^{\frac{1}{2\rho} - 1} \\ &= \frac{1}{2} + \rho \cdot \left(1 - e^{-\frac{1}{2\rho}}\right) - e^{\frac{1}{2\rho} - 1}. \end{aligned}$$

Where the penultimate inequality follows from the fact that $e^{1/x}/(e^{1/x} - 1) \geq x$ for all $x \neq 0$. □

IV. Approximately-Compliant RAMS

In this section, we study RAMS in the online i.i.d. model. In this model, we have a set $\mathcal{Y}$ of client types and a distribution D over $\mathcal{Y}$. We can think of a client type as a left-side vertex in the bipartite *type graph* $\mathrm{G}(\mathcal{Y}, \mathrm{A})$. The type graph $\mathrm{G}(\mathcal{Y}, \mathrm{A})$, n, and the distribution D are given to the online algorithm upfront. Then, at each online round $\ell = 1, \dots, n$, a client type y is drawn from D, and a client u_ℓ of this type arrives (the mandatory demand μ_ℓ and the total demand $s_\ell \geq \mu_\ell$ of the client may be arbitrary).

For $\mathrm{G}(\mathcal{Y}, \mathrm{A})$, D and n, let $\mathcal{I}(\mathrm{G}(\mathcal{Y}, \mathrm{A}), D, n)$ denote the distribution over random instances generated by the random process described above. Also, for $y \in \mathcal{Y}$, let p_y be the probability that a client of type y is drawn from D, i.e., $p_y = \Pr_{X \sim D}[X = y]$.

Let ALG be an online algorithm for RAMS in the i.i.d. model. ALG is called λ-competitive, if for any $\mathrm{G}(\mathcal{Y}, \mathrm{A})$, D and n, $\mathbb{E}[\mathrm{ALG}(I)] \geq \lambda \cdot \mathbb{E}[\mathrm{OPT}(I)]$, where the expectation is taken over the choice of $I \sim \mathcal{I}(\mathrm{G}(\mathcal{Y}, \mathrm{A}), D, n)$. For $r \in \mathbb{R}_{\geq 0}$ we say that ALG *ε-compliant* for capacity-ratio r, if for any $\mathrm{G}(\mathcal{Y}, \mathrm{A})$, D and n with $\rho(\mathcal{I}) \geq r$, with probability at least $1 - \varepsilon$, ALG on $I \sim \mathcal{I}$ serves the mandatory demands of all

```
Algorithm 3: Approx. Reservations Scheme ARS $(\mathrm{ALG}', \varepsilon)$
for each server $a_j$ with capacity $c_j$ do
    $c'_j \leftarrow c_j - \phi_j$                    // modified capacities
    $\phi'_j \leftarrow \phi_j$                       // reserved resources
    $R_j \leftarrow 0$                                // released resources
Initialize ALG$'$ with server set A and modified capacities
    $c'_1, \dots, c'_m$
for a client $u_\ell$ that arrives at round $\ell$ do
    Let $y$ be the type of $u_\ell$
    Let $a_j = a(y)$ be the designated server for type $y$
    if $\phi'_j \geq 1$ then
        $\phi'_j \leftarrow \phi'_j - 1$; $R_j \leftarrow R_j + 1$
    else
        Failure
    Feed the client $u_\ell$ to ALG$'$ (simulation)
    Let $x'_{\ell,1}, \dots, x'_{\ell,m}$ be the resources allocated to $u_\ell$ by ALG$'$
    // add resources allocated by ALG$'$ to
        released resources
    For all $a_j \in$ A: $R_j \leftarrow R_j + x'_{\ell,j}$
    /* Serve from released resources                      */
    if serve-by-one then
        Serve $u_\ell$ by GREEDY with server capacities $R_1, \dots, R_m$
    else (serve-by-many)
        Serve $u_\ell$ by WATER-FILLING with capacities $R_1, \dots, R_m$
```

clients.[4] We also define the *mandatory service portion* of $\mathcal{I}$ by $\delta(\mathcal{I}) = n/\mathbb{E}\left[\mathrm{OPT}(I)\right]$. We express the competitive-ratio of our proposed algorithm in this section in terms of δ.

We design an algorithmic scheme for RAMS in the online i.i.d. model that uses an online algorithm ALG$'$ for RAMS without the mandatory demand constraints as a black-box, and produces a ε-compliant algorithm for RAMS. More formally, we consider the problem *RA* defined by the LP (P) without the third type of constraints: $\sum_{a_j \in N(u_i)} x_{i,j} \geq \mu_i, \forall u_i \in$ C. Note that RA generalizes the online fractional bipartite matching problem. On the other hand, RA is a special case of the Adwords problem (see [6] for example) where the bids of all advertisers on each keyword are equal.

Let ALG$'$ be an online algorithm for RA. For each client type $y \in \mathcal{Y}$ we choose an arbitrarily designated server $a(y) \in N(y)$ to be responsible of assuring the mandatory demands of type y clients are served. We reserve resources in advance on the designated servers $\{a(y) : y \in \mathcal{Y}\}$. Let ϕ_j denote the amount of reserved resources on a_j. To determine $\{\phi_j\}_{a_j \in \mathrm{A}}$, we use the prior knowledge of the distribution D, and estimate the amount of resources needed in each server to serve all mandatory demands.

When a client u_ℓ of type y arrives with designated server $a_j = a(y)$, we release one unit of resource from ϕ_j to R_j (where R_j denotes the released resources in a_j). Then, we use the algorithm ALG$'$ on an instance with reduced server capacities, $c_j - \phi_j$, for all $a_j \in$ A as follows: we simulate ALG$'$ on u_ℓ. Then, we shift the resources that ALG$'$ allocates to u_ℓ from each server and add them to the released resources of the corresponding server (i.e., to $R_1, \dots, R_m$). Next, in the serve-by-one setting, we use the GREEDY algorithm to serve u_ℓ from the released resources, and in the serve-by-many setting, we use the WATER-FILLING (WF) algorithm to serve u_ℓ from the released resources (see Algorithm 3 for a formal description).

For $y \in \mathcal{Y}$, $a_j \in$ A and online round ℓ, we define an indicator random variable $X_{y,j,\ell}$ for the event $\{u_\ell \text{ is of type } y\} \wedge \{a(y) = a_j\}$. The total amount of mandatory demands that a_j is responsible for assuring is upper bounded by $\sum_{y \in \mathcal{Y}, \ell \in [n]} X_{y,j,\ell}$. We have $\mathbb{E}\left[\sum_{y \in \mathcal{Y}, \ell \in [n]} X_{y,j,\ell}\right] = n \sum_{y \in N(a_j)} \mathbb{1}_{\{a(y)=a_j\}} p_y$. For each server $a_j \in$ A we reserve an amount of $\phi_j = \min\{\tau + n\sum_{y \in N(a_j)} \mathbb{1}_{\{a(y)=a_j\}} p_y, n\}$ resources, where $\tau = \sqrt{n \cdot \ln(m/\varepsilon)/2}$. We denote the total amount of reserved resources across all servers by $\Phi(D, \varepsilon)$. We have

$$\begin{aligned}
\Phi(D,\varepsilon) = \sum_{j=1}^{m} \phi_j &\leq \sum_{a_j \in \mathrm{A}} \left(\tau + n \sum_{y \in N(a_j)} \mathbb{1}_{\{a(y)=a_j\}} p_y \right) \\
&= m\tau + n \sum_{y \in \mathcal{Y}} \sum_{a_j \in N(y)} \mathbb{1}_{\{a(y)=a_j\}} p_y \\
&= m\tau + n \sum_{y \in \mathcal{Y}} p_y = m\tau + n.
\end{aligned}$$

Next, we show that for any algorithm ALG$'$ for RA, ARS $\left(\mathrm{ALG}', \varepsilon\right)$ serves all the mandatory demands with probability at least $1 - \varepsilon$.

Theorem IV.1. *Let* ALG$'$ *be an online algorithm for RA and let* $\varepsilon > 0$, *then Algorithm* ARS$(\mathrm{ALG}', \varepsilon)$ *is* ε*-compliant for capacity-ratio* 1.

Proof. For $a_j \in$ A, let $Z_{j,\ell} = \sum_{y \in \mathcal{Y}} X_{y,j,\ell}$ be the random variable that gets the amount of mandatory resources a_j is responsible of assuring at round ℓ. If the mandatory demand of a client is not served, then it must be the case that $\sum_{\ell \in [n]} Z_{j,\ell} > \phi_j$ for some server $a_j \in$ A. Otherwise, each time a client arrives, the designated server of its type releases one unit of resource, and the mandatory demand is served from the released resources (by GREEDY or WF). Thus, to upper bound the probability that the mandatory demand of a client is not served, it suffices to upper bound the probability that there is a server $a_j \in$ A such that $\sum_{\ell \in [n]} Z_{j,\ell} > \phi_j$. Fix $a_j \in$ A. Observe that if $\phi_j = n$, the reserved resources of a_j cannot be depleted. Hence, we can assume that $\phi_j = \tau + n \sum_{y \in N(a_j)} \mathbb{1}_{\{a(y)=a_j\}} p_y$. We have $Z_{j,\ell} \in \{0, 1\}$, for all $\ell \in [n]$, and also, $Z_{j,1}, \dots, Z_{j,n}$ are independent. Therefore, we can apply a Chernoff bound and get that $\Pr\left[\sum_{\ell \in [n]} Z_{j,\ell} > \phi_j\right] = \Pr\left[\sum_{\ell \in [n]} Z_{j,\ell} - n\sum_{y \in N(a_j)} \mathbb{1}_{\{a(y)=a_j\}} p_y > \tau\right] \leq e^{\frac{-2\tau^2}{n}}$. Now, by a union bound, we get that the probability that there is a server $a_j \in$ A with $\sum_{\ell \in [n]} Z_{j,\ell} > \phi_j$ is at most $me^{-2\tau^2/n}$. By substituting $\tau = \sqrt{n \cdot \ln(m/\varepsilon)/2}$, we get that this probability is at most ε. □

Remark IV.2. *Note that in the proof of Theorem IV.1 we use the fact that* $\rho \geq 1$ *only to guarantee that each server,* $a_j \in A$*,*

[4] Note that the capacity-ratio depends only on n and A, and thus, it is not a random variable.

has capacity at least ϕ_j. Hence, the theorem statement also holds under the (weaker) conditions where each server $a_j \in A$ has capacity at least ϕ_j (even if $\rho < 1$).

We now proceed to analyze the competitive-ratio of the algorithm in terms of δ. Recall that $\delta(\mathcal{I}) = n/\mathbb{E}[\text{OPT}(I)]$.

Theorem IV.3. *Let* ALG$'$ *be a λ-competitive algorithm for the RA problem. Then, for any constant $\varepsilon > 0$,* ARS $\left(\text{ALG}', \varepsilon\right)$ *is $\lambda \max\{\delta(1-\varepsilon), 1-\delta(1+1/\sqrt{2})\}$-competitive for* RAMS *with $\rho \geq 1$ and $n \geq m^2 \ln(m/\varepsilon)$.*

Proof. For an instance $I \sim \mathcal{I}$, let I' denote the instance with the modified, reduced server capacities $c'_1, \ldots, c'_m$. We first consider the RA problem (without mandatory demands). Consider the optimal solution for RA, OPT(I), and the solution obtained from OPT(I) by removing the minimal amount of allocated resources from each server until all allocations fit in the reduced capacities. Let y denote the resulting solution and its value. We have OPT$(I) \leq y + \Phi(D, \varepsilon)$, and since y is a feasible solution to I', we have OPT$(I') \geq y$. Hence, OPT$(I') \geq$ OPT$(I) - \Phi(D, \varepsilon)$. Since the last inequality holds for all I, it also holds in expectation over the random choice of $I \sim \mathcal{I}$, i.e., $\mathbb{E}[\text{OPT}(I')] \geq \mathbb{E}[\text{OPT}(I)] - \Phi(D, \varepsilon)$. We now substitute $\Phi(D, \varepsilon) = \sqrt{nm^2 \ln(m/\varepsilon)/2} + n$ in the last inequality, and use the fact that $n \geq m^2 \ln(m/\varepsilon)$, and get that $\mathbb{E}[\text{OPT}(I')] \geq \mathbb{E}[\text{OPT}(I)] \left(1 - \frac{n + \sqrt{n^2/2}}{\mathbb{E}[\text{OPT}(I)]}\right) \geq \mathbb{E}[\text{OPT}(I)] \left(1 - \delta(\mathcal{I})(1 + 1/\sqrt{2})\right)$.

Now, since ALG$'$ is λ-competitive, we have $\mathbb{E}\left[\text{ALG}'(I')\right] \geq \lambda \cdot \mathbb{E}[\text{OPT}(I')]$. For convenience of notation, we denote ARS$(\text{ALG}', \varepsilon)$ by ALG. Observe that ALG always serves each client with at least as much resources as ALG$'$ does. This is because the resources that ALG$'$ allocates to a client are released to $R_1, \ldots, R_m$, and then ALG uses either GREEDY or WF to serve the client from $R_1, \ldots, R_m$. Therefore, we have $\mathbb{E}[\text{ALG}(I)] \geq \mathbb{E}\left[\text{ALG}'(I')\right]$. Overall, $\mathbb{E}[\text{ALG}(I)] \geq \mathbb{E}\left[\text{ALG}'(I')\right] \geq \lambda \mathbb{E}[\text{OPT}(I')] \geq \lambda \mathbb{E}[\text{OPT}(I)] \left(1 - \delta(\mathcal{I})(1 + 1/\sqrt{2})\right)$.

Additionally, ALG serves all mandatory demands with probability at least $(1 - \varepsilon)$, and so we have the trivial bound $\mathbb{E}[\text{ALG}] \geq (1-\varepsilon) \cdot n \geq (1-\varepsilon) \cdot \delta(\mathcal{I}) \cdot \mathbb{E}[\text{OPT}(I)]$. To conclude,

$$\mathbb{E}[\text{ALG}] \geq \lambda \max\{\delta(1-\varepsilon), 1 - \delta(1 + 1/\sqrt{2})\} \mathbb{E}[\text{OPT}]. \square$$

Devanur et al. [11] showed that GREEDY is $(1 - 1/e)$-competitive for the Adwords problem in the i.i.d. model, and so, it is also $(1 - 1/e)$-competitive for the RA problem. Also, similarly to our proof of Theorem III.4, one can show that WATER-FILLING (WF) is $(1 - 1/e)$-competitive for the RA problem (even in the adversarial model, see also [7]). Hence, we get the following corollary.

Corollary IV.4. *For $\rho \geq 1$, $\varepsilon > 0$ and $n > m^2 \ln(m/\varepsilon)$,* ARS (GREEDY, ε) *and* ARS (WF, ε) *are ε-compliant for capacity-ratio 1 and $(1-1/e)\max\{\delta(1-\varepsilon), 1-\delta(1+1/\sqrt{2})\}$-competitive for the serve-by-one and serve-by-many* RAMS*, respectively.*

Remark IV.5. *Note that we use the knowledge of the distribution D only to calculate $\{\phi_j\}_{a_j \in A}$, which are independent of the demand values $\mu_1, \ldots, \mu_n$, $s_1, \ldots, s_n$ and the arrival order of the clients. Hence, our results also hold when $\mu_1, \ldots, \mu_n$, $s_1, \ldots, s_n$ are chosen by an adversary, and for adversarial arrival order.*

V. PERFORMANCE EVALUATION – CELL SELECTION

In this section, we demonstrate a practical application of our new algorithms by evaluating their performance on realistic data of the online cell selection problem [12]. In the online cell selection problem, mobile devices request service over time. When a request arrives, the network needs to decide on the amount of service to provide it, and the location of the cell (or cells if concurrent service from multi cells is allowed) to provide this service from. This decision has to be made immediately and irrevocably based only on the available information upon the request arrival.

We simulate realistic data of the online cell selection problem in New York City. To this end, we use real locations of New York City cellular antennas as the cell locations [13]. To generate the sequence of client requests, we place each client randomly in the simulation area. To model populated areas like business areas and less populated areas like residential areas, we use a Pareto distribution to draw client locations. More concretely, we partition the simulation area into sections using a 10x10 grid. We associate each section with a frequency drawn from a Pareto distribution with a shape of 2. Based on these frequencies, each client is then randomly assigned to a grid section. The actual location of the client within the section is selected uniformly at random.

Each cell has a service range that determines its connectivity with clients, that is, a client is connected to the cell if its distance to the cell is at most the cell's range. We choose the capacity of each cell to be n. To model high and low client demands, the total demand of each client is also drawn from a Pareto distribution with a shape of 2. In what follows, we use the term *load* to describe the sum of client demands divided by the total server capacities, i.e., *load* $= \sum_{i=1}^{n} s_i / \sum_{j=1}^{m} c_j$. To achieve a given load, we scale all total demands by the appropriate factor. We also choose the mandatory demand of each client to be 1.

In Figure 1, we present an example of a simulation area. The large dark blue dots represent cells, and the circles around them represent their service range. The green dots are the client locations. The heat map depicts the number of clients in each section (red represents more populated sections). Observe that the client location distribution is top-heavy with a few sections that contain most of the clients.

In addition to our fully-compliant algorithms GWR and WFWR, and our approximately-compliant algorithms, ARS(GREEDY, ε) and ARS(WF, ε), we also evaluate the performance of the following two simple compliant heuristics. The first is called *Equal Filling with Reservations* (EFWR), which equally splits the total demand of each client between its connected cells, and serves the available portion of the demand

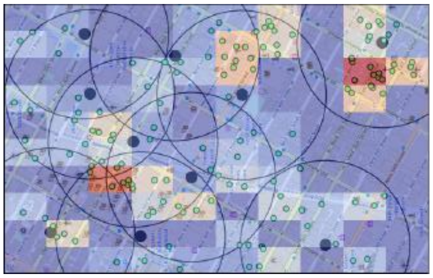

Fig. 1. Location distribution of cells and clients.

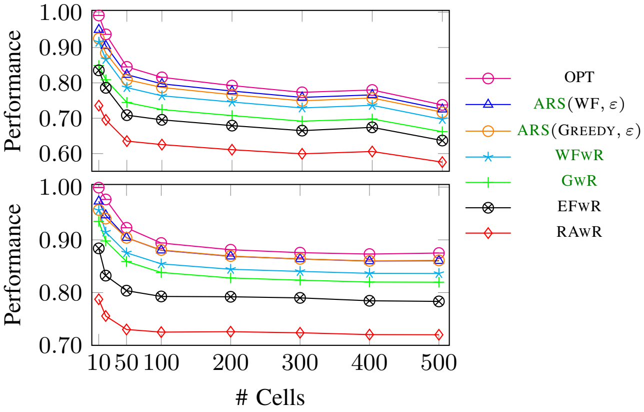

Fig. 2. Algorithms performance with respect to the number of cells. Pareto distributions are used for client locations and demands in the top graph. Uniform distributions are used in the bottom graph.

from each cell. The second is called *Random Allocations with Reservations* (RAwR), which for each client, randomly selects one of its connected cells and serves the maximum available demand from it. We also present the performance of the optimal offline solution OPT, which is calculated by solving the LP (P).

The theoretical results in Section IV assume that ARS(GREEDY, ε) and ARS(WF, ε) get the client distribution D as input. To evaluate their performance in practice, we do not use the exact distribution. Instead, we use the empirical estimation by drawing n clients from D. In practical scenarios, network operators can use historical data in a similar way to estimate the total number of clients, n, and their location distribution.

Figure 2 depicts the performance of the described algorithms as a fraction of the total demand served. For ARS(WF, ε) and ARS(GREEDY, ε) we use $\varepsilon = 0.1$. The ratio between the number of clients and the number of cells is kept constant, $n/m = 15$. The average number of connected cells per client is 6, and *load* $= 1$ (i.e., $\sum_{i=1}^{n} s_i = \sum_{j=1}^{m} c_j$). In the top graph, both client locations and total demands are drawn from a Pareto distribution, and in the bottom graph, both client locations and demands are drawn from a uniform distribution.

In the top graph, all curves follow a similar trend where the performance decreases as the number of clients increases. A possible explanation for this is that although the number of clients increases with the number of cells, due to the Pareto distribution, the popular areas become more crowded while the capacity of the cells in these areas remains the same. Observe that the relative performance of all algorithms compared to OPT is almost constant regardless of the number of clients, and that the approximately-compliant algorithms perform better than their fully-compliant counterparts.

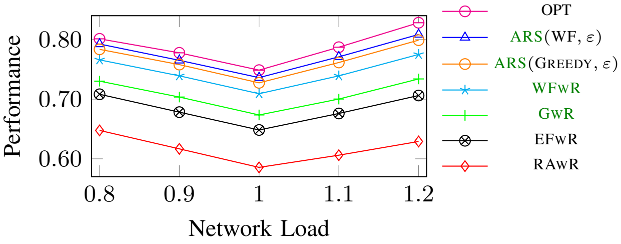

Fig. 3. Algorithms performance with respect to network load.

The trends in the bottom graph are similar. One can see that all algorithms perform better with uniform distributions (bottom graph) with Pareto distributions (top graph). This might also indicate that the decrease in performance depicted in the top graph is caused by highly populated sections in the map. Observe that in the case of uniform distributions too, ARS(GREEDY, ε) and ARS(WF, ε) perform best.

Figure 3 depicts the algorithms' performance as a function of the network load. All simulations are performed on networks that contain 400 cells, 2000 clients, and an average of 6 connected cells per client. The performance of the algorithm is measured as the ratio between the total demand served by the algorithm and the minimum between the total client demands and the total server capacities (both provide an upper bound on OPT), i.e., $\frac{\sum_{i\in[n],j\in[m]} x_{i,j}}{\min\{\sum_{i\in[n]} s_i, \sum_{j\in[m]} c_j\}}$.

One can see that the worst performance is obtained when the network load is 1. A possible explanation for this is that, on the one hand, when the network load is greater than 1, the algorithms have more possibilities to utilize the network resources. On the other hand, when the network load is less than 1, the algorithms are less likely to fully occupy the cells and therefore, more clients can be provided with a larger fraction of their total demand.

In Figure 4, we present the effect of the value of ε on the performance and the failure probability of the algorithms. The failure probability is the probability that the algorithm does not satisfy all mandatory demands. The simulations are performed on networks that contain 10 cells, 50 clients, 2 connected cells per client on average, and *load* $= 1.1$. The top graph presents the performance of the algorithms as a fraction of the optimal offline solution OPT and the bottom graph presents the failure probability. One can see that as ε increases the performance and the failure probability increases. This happens since the initial reserved capacity (in each cell) decreases. As expected ARS(WF, ε) performs better than ARS(GREEDY, ε) and its failure probability is lower. Also, observe that the actual failure probabilities of both algorithms are better than their corresponding theoretical guarantees.

VI. Additional Related Work

Cell planning, the problem of optimal planning of cellular network, is one of the basic and most studied problems in

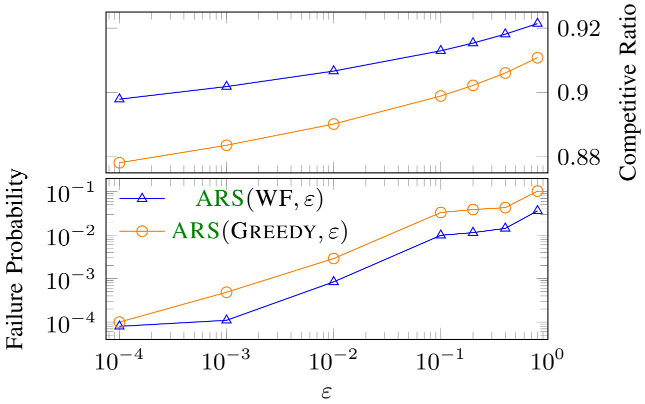

Fig. 4. ε effect on the performance of the approximated fairness algorithms.

the context of cellular systems (see for example [14]–[17]). In this paper we focus on the specific problem of cell selection, where one needs to determine the specific cell (or cells) that provides service for a certain mobile device [5], [12], [18].

Traditionally, cell selection is done by the mobile device, which selects the cell with the best signal-to-interference-plus-noise ratio (SINR) [19]. However, new generation cellular networks allow for other selection methods that potentially can improve network performance [18], [20].

The trade-off between fairness and maximal utilization is an important factor here and was studied extensively both in the specific cellular context [3]–[5] and in the more general online routing problem [1], [21].

From the theoretical viewpoint, the cell selection problem we study in this paper is closely related to the integral and fractional online matching problems, and to the online Adwords problem which were studied extensively over the last three decades. We refer the reader to [6] for an extensive survey on these problems. [6] also covers the three online models we study in this paper (adversarial-order, random-order, and i.i.d.) and the relations between them.

VII. Discussion

In this paper, we introduced the online resource allocation problem with mandatory services. We presented multiple *fully-compliant* and *approximately-compliant* algorithms and proved theoretical guarantees on their performance in various online models. Performance evaluation of these algorithms in cell selection scenarios indicates that they can indeed be used in relevant practical cellular settings to achieve a good balance between fairness and utilization.

As we showed, prior knowledge can be used to improve performance. We note that the assumption of a fixed known distribution (i.e., the i.i.d. model) can be relaxed and replaced by new models that only assume the ability to obtain a representative sample of the requests from historical data (see [8], [22], [23]). Additionally, historical data can be used to estimate the distribution of mandatory demand values. Such estimations can be incorporated in our approximately-compliant algorithms to reduce the amount of reserved resources and improve performance.

References

[1] A. Goel, A. Meyerson, and S. Plotkin, “Combining fairness with throughput: Online routing with multiple objectives,” in *Proceedings of the Thirty-Second annual ACM Symposium on Theory of Computing (STOC)*, 2000, pp. 670–679.

[2] T. Bonald, L. Massoulié, A. Proutiere, and J. Virtamo, “A queueing analysis of max-min fairness, proportional fairness and balanced fairness,” *Queueing systems*, vol. 53, no. 1, pp. 65–84, 2006.

[3] I. Chlamtac and A. Lerner, “Fair algorithms for maximal link activation in multihop radio networks,” *IEEE Transactions on Communications*, vol. 35, no. 7, pp. 739–746, 1987.

[4] C. Peng, H. Zheng, and B. Y. Zhao, “Utilization and fairness in spectrum assignment for opportunistic spectrum access,” *Mobile Networks and Applications*, vol. 11, no. 4, pp. 555–576, 2006.

[5] J. Wang, J. Liu, D. Wang, J. Pang, and G. Shen, “Optimized fairness cell selection for 3gpp lte-a macro-pico hetnets,” in *2011 IEEE Vehicular Technology Conference (VTC)*. IEEE, 2011, pp. 1–5.

[6] A. Mehta, “Online matching and ad allocation,” *Foundations and Trends in Theoretical Computer Science*, vol. 8, no. 4, pp. 265–368, 2013.

[7] Y. Azar and A. Litichevskey, “Maximizing throughput in multi-queue switches,” *Algorithmica*, vol. 45, no. 1, pp. 69–90, 2006.

[8] D. Naori and D. Raz, “Online placement of virtual machines with prior data,” in *IEEE Conference on Computer Communications (INFOCOM)*, 2020, pp. 2539–2548.

[9] N. Buchbinder, K. Jain, and J. S. Naor, “Online primal-dual algorithms for maximizing ad-auctions revenue,” in *European Symposium on Algorithms (ESA)*. Springer, 2007, pp. 253–264.

[10] N. R. Devanur, K. Jain, and R. D. Kleinberg, “Randomized primal-dual analysis of ranking for online bipartite matching,” in *Proceedings of the Twenty-Fourth annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2013.

[11] N. R. Devanur, K. Jain, B. Sivan, and C. A. Wilkens, “Near optimal online algorithms and fast approximation algorithms for resource allocation problems,” *Journal of the ACM (JACM)*, vol. 66, no. 1, pp. 1–41, 2019.

[12] D. Amzallag, R. Bar–Yehuda, D. Raz, and G. Scalosub, “Cell selection in 4g cellular networks,” *IEEE Transactions on Mobile Computing 12 (7)*, 2012.

[13] OpenCellId. (2022) Open Database of Cell Towers. [Online]. Available: https://www.opencellid.org

[14] A. Taufique, M. Jaber, A. Imran, Z. Dawy, and E. Yacoub, “Planning wireless cellular networks of future: Outlook, challenges and opportunities,” *IEEE Access*, vol. 5, pp. 4821–4845, 2017.

[15] D. Amzallag, M. Livschitz, J. Naor, and D. Raz, “Cell planning of 4g cellular networks: Algorithmic techniques and results,” in *2005 6th IEEE International Conference on 3G and Beyond*. IET, 2005, pp. 1–5.

[16] L. U. Khan, I. Yaqoob, M. Imran, Z. Han, and C. S. Hong, “6g wireless systems: A vision, architectural elements, and future directions,” *IEEE access*, vol. 8, pp. 147 029–147 044, 2020.

[17] S. Wang, W. Zhao, and C. Wang, “Budgeted cell planning for cellular networks with small cells,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4797–4806, 2014.

[18] Q. Han, B. Yang, C. Chen, and X. Guan, “Matching-based cell selection for proportional fair throughput boosting via dual-connectivity,” in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2017, pp. 1–6.

[19] J. G. Andrews, F. Baccelli, and R. K. Ganti, “A tractable approach to coverage and rate in cellular networks,” *IEEE Transactions on Communications*, vol. 59, no. 11, pp. 3122–3134, 2011.

[20] E. Coronado, S. Siddiqui, and R. Riggio, “Roadrunner: O-ran-based cell selection in beyond 5g networks,” in *NOMS IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2022, pp. 1–7.

[21] J. Wong, J. Sauve, and J. Field, “A study of fairness in packet-switching networks,” *IEEE Transactions on Communications*, vol. 30, no. 2, pp. 346–353, 1982.

[22] H. Kaplan, D. Naori, and D. Raz, “Competitive analysis with a sample and the secretary problem,” in *Proceedings of the Thirty-First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2020, pp. 2082–2095.

[23] ——, “Online weighted matching with a sample,” in *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2022, pp. 1247–1272.