This article has been accepted for publication in IEEE Transactions on Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIT.2024.3399760

Joint Beam Allocation and Scheduling for mmWave Cellular Networks

Swaroop Gopalam, Iain B. Collings, Stephen V. Hanly, Hazer Inaltekin
School of Engineering, Macquarie University, Sydney, Australia
{swaroop.gopalam, iain.collings, stephen.hanly, hazer.inaltekin}@mq.edu.au

***Abstract*—This paper provides capacity results for multi-user mm-wave hybrid-beamforming, and presents optimal joint beam allocation and user scheduling algorithms. We characterize the downlink capacity of a practical system with quantized analog beamforming code-books under the constraint that users cannot be scheduled at the same time if they are closer together than a beam width in angle. We show that the capacity region is determined by a small number of linear inequality constraints. We also present capacity-achieving scheduling algorithms that provide beam allocations guaranteeing that user rate requirements are met within each resource block. In particular, we propose "sand-filling" algorithms that are provably optimal and which have linear complexity. Intuitively, our schemes can be viewed in terms of filling containers with coloured sand, in such a way that the colours at any given height do not conflict with the colours in the other containers at the same height, where the containers represent the RF chains (*i.e.* the beamforming resources), and the coloured sand represents the users (and their rate requirements). We show a numerical example where the capacity of our scheme is 82% higher than a traditional resource partitioning scheme.**

I. Introduction

Millimeter wave (mmWave) communications offers enormous potential for delivering high data rate services to large numbers of mobile users in 5G and 6G cellular networks. Achieving wide area coverage in this band requires beamforming from antenna arrays with large numbers of elements, due to the small element sizes at mmWave. The resulting beams are comparatively narrow, providing signal gain as well as the opportunity to spatially multiplex multiple beams to serve multiple users simultaneously. The number of beams is determined by the number of RF chains, which is significantly fewer than the number of antenna elements, in practical hybrid-beamforming mmWave implementations. This is due to a number factors including the physical size of amplifiers, their heat dissipation, and the need to lock carrier frequencies across all RF chain mixers. Multiplexing the beams (of each RF chain) will result in inter beam interference if the user equipments (UEs) are not sufficiently separated in angle. Consequently, there is a significant user-scheduling challenge when a cell contains large numbers of UEs, non-uniformly located in angle around the base station, and where each UE has their own data rate requirements.

In this paper we consider the question of cell capacity on the downlink, for a practical hybrid-beamforming system that can form a given number of simultaneous beams (one for each RF chain, using analog beamforming, which has been shown to be close-to-optimal in single user scenarios [1]), and where there is a quantized beamforming code-book that divides the transmission angle range into a finite number of segments along which each beam can be pointed. Beam-steering in this way is a practical scheme for mmWave channels, that requires minimal feedback compared to digital beamforming which requires full channel state information. We provide capacity results under the constraint that UEs in neighbouring segments that are closer than a beam width in angle cannot be scheduled at the same time, in order to avoid interference between the scheduled UEs. We show that the capacity region can be determined by a linear set of constraints, as opposed to computing a convex hull of exponentially many points (corresponding to all the beamforming and UE scheduling choices). We also propose two optimal scheduling algorithms (that achieve capacity), in a number of important general practical scenarios.

In [2]–[4], the multi-user multi-input multi-output (MU-MIMO) cellular user-selection problem with a quantized beamforming codebook was tackled from a signal-to-interference ratio (SIR) perspective. They considered a scenario with limited channel state information (CSI) feedback, and semi-randomly selected users that achieve an overall high SIR. These approaches did not consider UE rate requirements, nor provide any rate guarantees.

A network utility maximization approach to user scheduling was taken in [5], [6], where ZF based hybrid beamforming for MU-MIMO was considered in [5], and SU-MIMO hybrid beamforming was considered in [6] where UEs were scheduled in time sequence. Both works involve solving a convex optimization along with a max-weight scheduling problem in each slot. The max-weight scheduling problem in [5] is NP-hard and hence convex relaxation and heuristic algorithms were proposed. In contrast, the single user max-weight problem was shown to have a pseudo-convex structure in [6] for the case of low side-lobes.

Sum rate optimization for mmWave link scheduling was used in [7], [8]. In [7] a graph based approach was taken, where scheduling of interfering links was avoided and a max-weight scheduling approach was proposed for a fixed number of RF chains. In [8] user scheduling and analog beam selection was done from a code-book, where the UE rate included the interference from simultaneous beams. Due to NP-hardness of the problem, a heuristic user-clustering based scheduling

This work was supported by the Australian Research Council's Discovery Project Funding Scheme (Project number DP230102252).

This article has been accepted for publication in IEEE Transactions on Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIT.2024.3399760

algorithm was proposed.

Scheduling in multi-cell scenarios was considered in [9]–[13]. In [9], [10], inter-cell interference was avoided by scheduling beams in adjacent cells by constructing an interference graph. The objective of minimizing the number of slots required to satisfy SINR requirements subject to the limit on RF chains was considered for mmWave backhaul links in [11], where greedy sub-optimal solutions were proposed due to the hardness of the problem. In [12], slot number minimization was done subject to rate requirements and constraints on transmit power and RF chains. A heuristic solution based on user-clustering was proposed due to the hardness of the problem. In [13], a heuristic algorithm is proposed for sum rate maximization with hybrid beamforming.

The problem of scheduling a given set of tasks (with a conflict graph) on a fixed number of processors has been considered in operations research as the Mutual Exclusion Scheduling (MES) problem (*e.g.* see [14]). The problem is known to be NP-hard in general (and for many classes of graphs). Polynomial time algorithms exist for *only* a few classes of conflict graphs such as trees and forests [14]. The MES problem with unit size indivisible jobs was considered in [15], for perfect circular arc graphs, and obtained polynomial time algorithms. The beamforming setup we consider in this paper *can* be modeled as a perfect circular graph, however, as we will see, a major difference from [15] is that our model has real valued infinitely divisible jobs (*i.e.* the rate requirements of UEs), which necessitates a completely different solution. We prove, perhaps somewhat surprisingly, that this problem can be solved in linear time.

In this paper, we present a capacity-achieving scheduling algorithm that provides beam allocations guaranteeing that UE rate requirements are met within each resource block. The combinatorics of allocating K beams at a time, across N segments, and meeting the real-valued rate requirements of all users, appears to be, on the face of it, prohibitive. Our "sand filling" approach to scheduling, however, has linear complexity. Intuitively, our approach can be viewed in terms of filling containers with coloured sand, in such a way that the colours at any given height do not clash with the colours in the other containers at the same height; where the containers represent the RF chains (*i.e.* the beamforming resources), and the coloured sand represents the UEs (and their rate requirements).

We present two sand-filling algorithms, both of which have linear complexity. Our first algorithm takes a greedy approach to filling the containers, and we show it to be optimal in two important, practically relevant scenarios. Our second algorithm modifies the greedy approach and it is optimal across a much wider range of practically relevant scenarios. Finally, we provide simple capacity characterizations for the practical scenarios where we have established the optimality of our algorithms. We show that the capacity is characterized by only $N+1$ linear inequalities even though there can be exponentially many possible beam-segment/UE combinations. We also show a numerical example where the capacity of our scheme is 82% higher than a traditional resource partitioning scheme.

II. System Model and Capacity Region

In this section, we present our mmWave MU-MIMO base station model and introduce the neighboring segment scheduling constraints. We define the notion of feasible sets and then characterize the capacity region achievable by scheduling feasible sets.

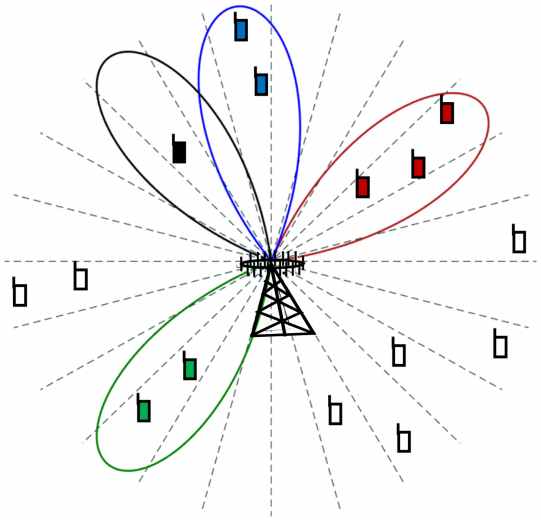


Fig. 1. Illustration with $\theta = 360^\circ, N = 24, I = 1, K = 4$.

A. System Model

Consider a mmWave gNB (Next Generation Node Base) in a single cell scenario communicating on the down-link with a set of user equipments (UEs) distributed within an arc of angular-width θ degrees. We consider the practical case for mmWave communications, where the gNB RF front-end has a limited number of RF chains, and uses analog beamforming (one for each RF chain) to serve up to K UEs simultaneously, where K is the number of RF chains. Note that analog beamforming has been shown to be close-to-optimal in single user scenarios [1]). We consider the practical case of having a quantized beamforming code-book, where the 360° plane from the gNB is divided into N non-overlapping segments. The mid-point angles of these segments are the N possible directions in which the K analog beams can be directed. See Fig. 1 which shows an example for $\theta = 360^\circ$. We label the segments 0 to $N-1$ going in the clockwise direction. The UEs located in a segment n are served by the beam pointing in the direction of segment n (which is the beam closest in direction to these UEs). Beam-steering in this way is a practical scheme for mmWave channels, that requires minimal feedback compared to digital beamforming which requires full channel state information.

Depending on the beam-width, a beam can be wider than a single segment which leads to significant interference (for the UEs) in the adjacent segments. A beam pointing in segment n causes significant interference to (the UEs in) $2I$ segments, $\{n \pm I \mod N, \ldots, n \pm 1 \mod N\}$, *i.e.* I adjacent segments in each direction. In Fig. 1, $I = 1$. Hence, it is necessary to avoid scheduling two UEs that are closer than I segments. We

assume that UEs that are more than I segments apart, have nearly orthogonal channels and suffer only negligible residual interference due to the separation in angular domain and the directional nature of mmWave and THz communication channels.

Note that the beamwidth is independent from the number of RF chains. It is determined only by the number of antenna elements. Also, the number of segments (or equivalently, the number of beamforming vectors in the codebook) is independent of both the number of RF chains and the beamwidth. It is a system design choice that is only limited by the resolution of the phase shifters used in the analog beamforming hardware.

We consider the general situation where there can be multiple UEs within each segment, and where those UEs are scheduled in a time-shared manner when a beam is pointed to the given segment. We denote the *instantaneous rate* of UE u, to be R_u bits/sec[1]. This is the physical layer rate at which the gNB sends data to a UE u while it is scheduled. The instantaneous rate will naturally change over time due to fading of the wireless channel, and as with all resource allocation algorithms, whenever the rates change appreciably the scheduling algorithm will be re-run. For each resource allocation block, the UE rate R_u is a fixed constant.

We denote the *flow rate* of UE u, to be ν_u bits/sec. This is the long-term target rate at which data of UE u is served by the gNB (*i.e.* averaged over time, including the times when it is not scheduled). It is also a fixed constant.

For a segment n, we define the set of m adjacent segments in the clockwise direction, including n, using the following notation: $[n : n+m] := \{n + x \mod N,\ \forall x = 0, \ldots, m\}$ for each $n \in \{0, \ldots, N-1\}$ and $m \in \mathbb{Z}_+$. We call this a *clockwise neighbour set*.

In this setup, a set $S \subseteq \{0, \ldots, N-1\}$ of segments can be scheduled without interference between beams, if and only if,

$$|S| \leq K \tag{1}$$

$$S - \{n\} \bigcap [n - I : n + I] = \emptyset,\ \forall n \in S. \tag{2}$$

The number of simultaneous beams are limited by the number of RF chains, which is given in the RF chains constraint (1). The constraint (2) ensures that selected segments are separated from each other enough, such that there is no interference, and can therefore feasibly be scheduled at the same time.

Definition 1. *A set $S \subseteq \{0, \ldots, N-1\}$ is called a feasible set if it satisfies (1) and (2).*

Let $\mathcal{S}$ denote the set of all feasible sets. With no loss of generality[2], we consider the case where $\max_{S \in \mathcal{S}} |S| = K$.

B. Capacity Region

Let U_n denote the set of UEs that lie in segment n. Hence, $U := \bigcup_{n=0}^{N-1} U_n$ is the set of all the UEs. Since UEs within a segment are time-shared, at any given time each segment will only have a single scheduled (*i.e.* active) UE. Let $\mathcal{V} := \{\{u_n\}_{n=0}^{N-1} : u_n \in U_n\}$ be the set of all possible UE scheduling choices (*i.e.* choices of N UEs, with the constraint of having only a single UE per segment).

For a given feasible set of scheduled segments, $S \in \mathcal{S}$, and a UE choice $V \in \mathcal{V}$, the corresponding rate vector $\boldsymbol{r}_{S,V} \in \mathbb{R}_+^{|U|}$ is defined as

$$\boldsymbol{r}_{S,V} := [R_u \mathbb{I}_{u \in V} \mathbb{I}_{n \in S} : u \in U_n,\ n \in \{0, \ldots, N-1\}] \tag{3}$$

where $\mathbb{I}$ is the indicator function. Let

$$\mathcal{R} := \{\boldsymbol{r}_{S,V}\}_{S \in \mathcal{S}, V \in \mathcal{V}} \tag{4}$$

be set of all the rate vectors.

We define the *capacity region* of flow rate vectors achievable by scheduling (*i.e.* time-sharing) of the feasible sets and UE choices as

$$\mathcal{C} := Conv(\mathcal{R}) \tag{5}$$

where $Conv(\cdot)$ is the convex-hull.

In this paper, we consider the following two key challenges; 1) To determine whether a given flow rate vector $\boldsymbol{\nu} = [\nu_u]_{u \in U}$ is achievable, *i.e.* $\boldsymbol{\nu} \in \mathcal{C}$, and 2) To provide an optimal beam association and resource allocation scheme which will achieve a given $\boldsymbol{\nu} \in \mathcal{C}$, which we do by determining time fractions f_S for feasible sets S which achieve $\boldsymbol{\nu}$.

Determining whether a given $\boldsymbol{\nu}$ is in $\mathcal{C}$ is not straightforward due to the number of feasible sets under consideration, *i.e.* the size of the set $\mathcal{S}$. Note that $\mathcal{S}$ has in the order of $\sum_{k=0}^{K} \binom{N}{k}$ elements due to the combinatorial nature of feasible sets. In this paper, we provide optimal resource allocation algorithms with complexity linear in N. As such, our algorithms show that by exploiting the inherent structure in the problem, it is not necessary to perform a brute force enumeration of all the feasible sets.

III. Problem Formulation

In this section, we formulate the minimum time allocation problem as a linear program and show that it provides an equivalent characterization of the capacity region. We then show how to generate time allocations from a solution of the linear program.

We define a load vector $\boldsymbol{\tau} := [\tau_n]_{n=0}^{N-1}$, where $\tau_n := \sum_{u \in U_n} \nu_u / R_u$ is the *utilization* of segment n, which is the sum of the utilizations of all UEs in U_n. Segment n must be scheduled at least τ_n proportion of the time (subject to the scheduling constraints (1), (2)) to meet its load. Within segment n, the UEs $u \in U_n$ are time-shared proportional to $\frac{\nu_u}{R_u}$. We note that each $u \in U_n$ is scheduled for at least $\frac{\nu_u}{R_u}$ net proportion of the time (which leads to a flow rate of at least ν_u) provided that segment n's load is met.

It is useful to consider an illustration of beam allocations to the segments, achieved by time-sharing of feasible sets, shown in Fig. 2. The horizontal axis represents the segments, and the vertical time axis of interval $[0, 1)$ represents the proportion of time. There are $N = 24$ segments in this example (directions in which the beams can be pointed). The figure shows a

[1]This rate can include a margin for residual interference. For example, the R_u's could be determined based on the signal-to-interference ratio allowing for interference from a beam pointing $I + 1$ segments away.

[2]Note that when $\max_{S \in \mathcal{S}} |S| < K$, at most $\bar{K} := \max_{S \in \mathcal{S}} |S|$ segments can be scheduled at a given time and the remaining RF chains are redundant. As such, in this situation, we would simply set $K := \bar{K}$, ignoring the excess RF chains.

This article has been accepted for publication in IEEE Transactions on Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIT.2024.3399760

scenario with $K = 4$ RF chains (which can simultaneously create 4 beams). The allocation of these beams is shown with the shaded regions. Feasible sets are shown in the horizontal rows (labelled on the right hand side of the figure), where the shading shows the segments that are in that feasible set. For example, in feasible set S_1, corresponding to the interval $[0, f_{S_1})$, the four RF chains (forming 4 beams) are allocated to segments 0, 4, 8 and 20. Clearly, these segments are separated more than I segments apart, and therefore do not interfere with each other, rendering them feasible.

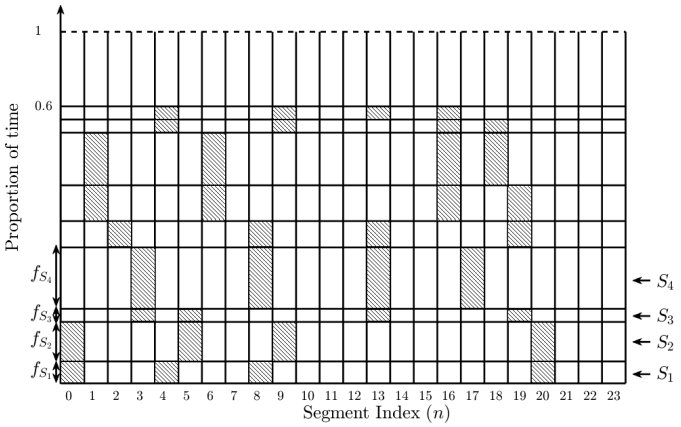


Fig. 2. Resource Allocation using LP (6)-(7) for $N = 24, I = 1, K = 4$.

In this example, at time f_{S_1}, there is a switch to a new feasible set $S_2 = \{0, 5, 9, 20\}$, in which two of the segments continue to be served (0 and 20), while the other two beams switch to segments 5 and 9. Note that each horizontal line in the figure is a switching time, when at least one beam switches to a different segment.

The challenge in determining whether a given flow rate vector $\boldsymbol{\nu}$ is achievable involves considering time (proportion) allocations f_S for feasible sets $S \in \mathcal{S}$, in order to meet the utilization requirements for each segment. For example, if we consider segment 0 in Fig. 2, it is scheduled across two feasible sets S_1, S_2. In order to meet segment 0's load τ_0, the total proportion of time $f_{S_1} + f_{S_2}$ must be greater than or equal to τ_0. The same requirement holds for all other segments, noting that many of them are scheduled across more than two feasible sets.

We seek the most efficient feasible sets and time allocations to achieve a given load vector $\boldsymbol{\tau}$. Consider the problem of minimizing the total proportion of time (system utilisation) $\sum_{S\in\mathcal{S}} f_S$ required to meet all the segment utilisation requirements, as stated in the following linear program (LP):

$$\min_{f_S \geq 0} \sum_{S\in\mathcal{S}} f_S \tag{6}$$

s.t.

$$\sum_{S:n\in S} f_S \geq \tau_n, \ \forall n = 0, \ldots, N-1 \tag{7}$$

where, f_S is the proportion of time allocated to a feasible set S. The objective in (6) is to minimize the *total* utilized time proportion $\sum_{S\in\mathcal{S}} f_S$. The constraints in (7) require that the proportion of time allocated to a segment n is greater than or equal to τ_n. We call LP (6)-(7) the *minimum time allocation problem*, with the understanding that time is normalized so that a value of 1 refers to the base station resource being fully utilized.

The optimal solution $\{f_S^*\}_{S\in\mathcal{S}}$ has the minimum possible *total* time proportion, $f^* := \sum_{S\in\mathcal{S}} f_S^*$, to meet the segment utilization requirements $[\tau_n]_{n=0}^{N-1}$, or equivalently to achieve the given flow rate vector $\boldsymbol{\nu}$. Since the proportion of time must always be less than unity, a question arises as to the physical interpretation if the optimal value f^* is greater than 1. The following theorem shows that if $f^* > 1$ then the flow rate vector $\boldsymbol{\nu}$ is outside the capacity region $\mathcal{C}$, and hence cannot be achieved by any allocation scheme.

Theorem 1. *$\boldsymbol{\nu} \in \mathcal{C}$ if and only if $f^* \leq 1$.*

Proof. See Appendix A. □

From Theorem 1, it is clear that the LP (6)-(7) is an equivalent capacity characterization. It is also clear that the solution to the minimum time allocation problem yields an optimal resource allocation scheme for scheduling beams, which will achieve any $\boldsymbol{\nu} \in \mathcal{C}$. As mentioned previously, in a segment n, the UEs in U_n can be scheduled by time-sharing.

The LP (6)-(7) is challenging to solve due to its dimensionality, *i.e.* the number of variables involved. Note that the LP contains $|\mathcal{S}|$ variables, which can grow as $O(\min(N^K, 2^N))$. For moderately large values of N and K, the resource allocation problem involves solving a potentially highly complex LP.

In this paper, we propose optimal linear-time algorithms which solve the minimum time allocation problem in practical scenarios. These will be presented in Section IV and Section VII. We first discuss the practical implementation issue of how to relate *proportions* of time to *actual* time allocations in a resource block.

A. Allocation of time in a resource allocation block

In this section, we take a solution of the minimum time allocation problem (from the previous section, which deals with *proportions* of time), and show how to generate *actual* time allocations in a resource allocation block.

As illustrated in Fig. 2, a disjoint interval in $[0, 1)$ of length f_S^* can be allocated to each feasible set S, under a solution of the minimum time allocation problem. Note that the allocations for sets lie in $[0, f^*)$ and the rest of the interval $[f^*, 1)$ is empty. Feasible allocations for a time interval $[0, t)$ can be produced by scaling the solution by t. We use this scaling approach to address allocation of time intervals to segments (and UEs) in a resource allocation block.

To be concrete, assume that time resource allocation blocks of duration B seconds are available to be scheduled. During each resource allocation block, time intervals are allocated to each segment by the scheduler. To get the time allocations in the time resource block, we can scale time intervals produced by the factor B. All time intervals allocated to beams in a resource block are completed by time f^*B, (as can be seen from Fig. 2) leaving an unused time interval of duration $(1 -$

This article has been accepted for publication in IEEE Transactions on Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIT.2024.3399760

$f^*)B$ for additional tasks, such as beam search or channel estimation.

The approach to resource allocation described above is sensible when data packet arrivals occur in a regular manner with uniform inter-arrival times between packets. When packets arrive irregularly with a constant arrival rate (e.g., as a stationary random process), the packet queues can get large due to traffic bursts. Providing higher flow rates by dedicating more time resources reduces the queueing latency.

One option for the random packet arrival case is to scale the time allocations by B/f^*. In this case, all the time intervals allocated to beams are completed at time B, the end of the resource block. This leaves no extra time for additional tasks, such as beam search or channel estimation which must then take place outside of the resource blocks used for data transmission. This proposed time scaling provides *more than sufficient* time to each beam in each resource block, thereby reducing the queueing latency experienced by flows, as compared to a time scaling by B, which was used in the regular packet arrivals case.

More generally, one can scale time proportions by a factor c seconds, where $B < c \leq B/f^*$, which provides f^*c seconds for data transmission and $B - f^*c$ seconds for additional tasks. Suppose that packet arrivals of a flow (corresponding to a UE) occur as a Poisson process, then the resulting M/D/1 queuing process for the flow has an expected stationary queue length of $\frac{1}{2}\frac{B}{c}(\frac{c}{B}-1)^{-1}$. When c is reduced, more of the resource block can be used for beam searching or channel estimation at the cost of higher queueing latency experienced by the traffic flows.

Whether for regular or random packet arrivals, the time allocation found by solving the LP (6)-(7) tells the base station how to allocate time to the beams to meet the user traffic flow demands in the system.

B. Lower Bounds

Below we derive two lower bounds on the optimal value f^* of LP (6)-(7), i.e., *minimum allocation time*. We will show that one of these bounds is tight in most cases which are of practical interest. We provide linear time optimal algorithms that are based on achieving this lower bound.

For the sake of convenience, we define $\tau_A := \sum_{i\in A}\tau_i$ for each $A \subseteq \{0,\ldots,N-1\}$. We introduce two key parameters τ_{avg} and $\tilde{\tau}_{\max}$ (where $\tilde{}$ indicates that it is a *clockwise neighbour set* load as opposed to an individual segment load) defined as follows:

$$\tau_{avg} := \frac{\sum_{n=0}^{N-1}\tau_n}{K} \tag{8}$$

$$\tilde{\tau}_{\max} := \max_{n\in[0:N-1]} \tau_{[n:n+I]}. \tag{9}$$

Lemma 1. *The optimal value of LP (6)-(7), f^*, is greater than or equal to* $\max\{\tau_{avg}, \tilde{\tau}_{\max}\}$.

Proof. See Appendix A. □

IV. Greedy Sand-Filling algorithm

In this section, we present our first algorithm to solve the minimum time allocation LP (6)-(7), and we make a "*sand-filling*" interpretation of the algorithm's operation. We first describe an alternate way to graphically depict the beam allocations, compared to the depiction we presented in Fig. 2.

In Fig. 3(a) we provide a representation from the point of view of the RF chains, using K columns; each column k corresponding to an RF chain. In this diagram, a label and colour are used to represent the segment to which the RF chain is allocated. The vertical axis represents time and the horizontal lines represent switching times, just as with Fig. 2. If two beams switch segments at the same time, the horizontal lines will align. For example, in Fig. 3(a), the first RF chain is pointing a beam in segment 0 initially, and switches to segment 1 at time h_0. It is clear that Fig. 2 and Fig. 3(a) are equivalent representations of a beam schedule.

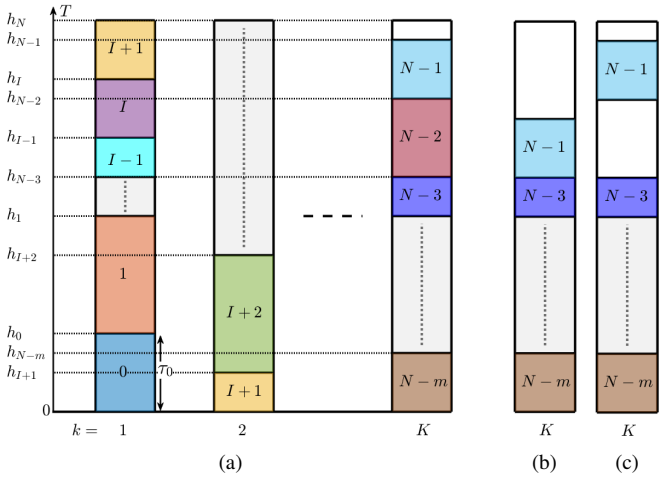


Fig. 3. (a) An illustration of the Greedy Sand-Filling algorithm. (b) An illustration of the Kth column when $\tau_{N-2} = 0$. (c) An illustration of the Kth column when $\tau_{N-2} = 0$ with proposed sand-shifting.

The sand filling analogy of Fig. 3(a) is as follows. Treat each container $k \in \{1,\ldots,K\}$ as having a height of $T = \max\{\tilde{\tau}_{\max}, \tau_{avg}\}$ and a base of unit area. Treat each segment $n \in [0:N-1]$ as having τ_n units of colored sand, where the color of the sand is unique to n. The sand-filling algorithm greedily fills the containers with sand, in the increasing order of segments, and in the increasing order of containers. Once a segment n's sand is finished, the filling continues with $n+1$'s sand. Similarly, once a container k is full, the filling continues in container $k+1$. Note that the container height T is sufficient since the total volume of the containers KT is greater than or equal to the volume of sand $\sum_{n=0}^{N-1}\tau_n = K\tau_{avg}$, since $T = \max\{\tilde{\tau}_{\max}, \tau_{avg}\}$.

This Greedy Sand-Filling algorithm can be used to generate a (potentially optimal) solution to the linear program (6)-(7) with objective value T as follows. Consider the height h_n where a segment n's sand-filling ends (which is a switching point). A subset of segments S_n is identified by the colors of sand across the containers just below height level h_n. By construction, there will be at most $N+1$ such subsets, say $\{S_i\}_{i=0}^{N}$, by letting $h_N = T$. These generated subsets are associated with disjoint height intervals, sub-intervals of $[0,T)$, *e.g.* the interval $[0,h_{I+1})$ (and $[h_{I+1},h_{N-m})$) is

associated with S_{I+1} (and S_{N-m} resp.) in Fig. 3(a). Let f_{S_i} denote the length of the interval for a subset S_i.

Definition 2. *We say that the Greedy Sand-Filling algorithm is optimal if and only if the generated solution $\{f_{S_i}\}_{i=0}^{N}$ is an optimal solution of LP* (6)-(7).

The generated solution $\{f_{S_i}\}_{i=0}^{N}$ is an optimal solution of the linear program (6)-(7) provided that all the subsets S_i generated by the algorithm are *feasible* sets, *i.e.* $S_i \in \mathcal{S}$. If each subset S_i is a feasible set, then the feasible solution $\{f_{S_i}\}_{i=0}^{N}$ to (6)-(7), generated by Greedy Sand-Filling, has an objective value $\sum_{i=0}^{N} f_{S_i} = T = \max\{\tau_{avg}, \tilde{\tau}_{\max}\}$ (since the interval $[0, T)$ is partitioned to obtain S_i's). By Lemma 1, this solution is optimal. We state this result as the following Lemma 2.

Lemma 2. *The Greedy Sand-Filling algorithm is optimal if the set of segments scheduled at each height $h \in [0, T)$, excluding switching points, is a feasible set.*

Note that under Greedy Sand-Filling, an RF chain serves a segment completely before switching to a new segment (or beam pointing direction), as seen in Fig. 3(a). This means that Greedy Sand-Filling minimizes the switching cost associated with changing beam directions.[3] Note that for the LP solution in Fig. 2, segment 4 (and also segments 4, 9, 13 & 19) are scheduled in two disjoint time intervals, which requires two beam direction switchings for these segments. This does not happen in our Greedy Sand-Filling approach.

We show in this paper that the Greedy Sand-Filling algorithm, or the modified version of it (the Generalized Sand-Filling algorithm in Section VII), generates optimal allocations in most practical cases. To do this, we introduce the necessary notation to describe the algorithms mathematically.

Definition 3. *We define position $\boldsymbol{p} \in \{1, \ldots, K\} \times [0, T)$ at a height h in a container k as the tuple $\boldsymbol{p} = (k; h)$.*

Definition 4. *The magnitude $|\boldsymbol{p}| \in [0, KT)$ of a position $\boldsymbol{p}$ is defined as $|\boldsymbol{p}| := (\boldsymbol{p}(1) - 1)T + \boldsymbol{p}(2)$, where $\boldsymbol{p}(i)$ is the ith component of $\boldsymbol{p}$.*

Note that the magnitude $|\boldsymbol{p}|$ is the total amount of sand needed to reach position $\boldsymbol{p}$. We say that $\boldsymbol{p}_1 \lesseqgtr \boldsymbol{p}_2$ if and only if $|\boldsymbol{p}_1| \lesseqgtr |\boldsymbol{p}_2|$ respectively. Hence, we order the positions $\boldsymbol{p}$ in the ascending order of magnitudes $|\boldsymbol{p}|$. Also note that each level in each container has a unique position $\boldsymbol{p}$ and magnitude $|\boldsymbol{p}|$. Hence, $|\cdot|$ is a bijective mapping from $\{1, \ldots, K\} \times [0, T)$ to $[0, KT)$. We denote the inverse mapping from magnitudes to positions by $|\cdot|_{inv}$.

Definition 5. *We define the displacement, $\boldsymbol{p} \oplus d$, of a position $\boldsymbol{p}$ to the right by a distance $d \geq 0$ as $\boldsymbol{p} \oplus d := ||\boldsymbol{p}| + d|_{inv}$. Similarly, the displacement to the left, $\boldsymbol{p} \ominus d := ||\boldsymbol{p}| - d|_{inv}$.*

Note that $|\boldsymbol{p} \oplus d| - |\boldsymbol{p}| = d$ and $\boldsymbol{p} \oplus d \geq \boldsymbol{p}$. Since $|\boldsymbol{p}|$ is the total amount of sand required to reach position $\boldsymbol{p}$, $\boldsymbol{p} \oplus d$ is the unique position obtained from $\boldsymbol{p}$ by adding d units of sand. Similarly, $\boldsymbol{p} \ominus d$ is the unique position obtained from $\boldsymbol{p}$ by removing d units of sand.

The Greedy Sand-Filling algorithm can now be described by the pseudo-code presented in Algorithm 1. In the following sections, we discuss the application of the Greedy Sand-Filling algorithm, *i.e.* where it leads to optimal allocations.

Algorithm 1 Greedy Sand-Filling algorithm

```
T = max{τ̃_max, τ_avg}                    // Container height
p = (1; 0).                               // Position at height 0 in container 1
for n = 0 to N − 1 do
    Allocate the positional range from p to p ⊕ τ_n to segment n.   // Sand-filling for n.
    p := p ⊕ τ_n.
end for
```

V. Applications of the Greedy Sand-Filling algorithm

In this section, we present two important practical scenarios 1) Balanced Scenario and 2) Partial Angular Coverage Scenario. We show that the Greedy Sand-Filling algorithm is optimal in both of these scenarios. In the next section, we consider the only remaining case of general Unbalanced Scenarios.

A. Balanced Loads: $\tau_{avg} \geq \tilde{\tau}_{\max}$

We define a scenario to be *balanced* if the $\boldsymbol{\tau}$ is such that $\tau_{avg} \geq \tilde{\tau}_{\max}$. The following theorem shows that the Greedy Sand-Filling algorithm is optimal (solves the LP (6)-(7)) for all balanced scenarios.

Theorem 2. *The Greedy Sand-Filling algorithm (Algorithm 1) is optimal if $\tau_{avg} \geq \tilde{\tau}_{\max}$. Further, $f^* = T = \tau_{avg}$.*

Proof of Theorem 2. Since $\tau_{avg} \geq \tilde{\tau}_{\max}$, it is clear that $T = \tau_{avg}$ by definition. We will show below that for an arbitrary segment $n \in [0, N-1]$, that n's allocation does not overlap (in height) with allocation of any of $[n+1 : n+I]$. The optimality of the algorithm follows from Lemma 2, which also implies that $f^* = T = \tau_{avg}$.

Let $(k; t)$ for some $1 \leq k \leq K,\ t \in [0, T)$ be the starting position $\boldsymbol{s}_n$ of an arbitrary segment $n \in [0 : N - I - 1]$ in Algorithm 1. Note that the segments $[n + 1 : n + I]$ are allocated between positions $\boldsymbol{s}_n \oplus \tau_n$ to $\boldsymbol{s}_n \oplus \tau_{[n:n+I]}$. Since $\tau_{[n:n+I]} \leq \tilde{\tau}_{\max} \leq T$, it follows that $\boldsymbol{s}_n \oplus \tau_{[n:n+I]} \leq (k+1; t)$, *i.e.* the allocations of segments $[n+1 : n+I]$ can only lie above height $t + \tau_n$ in container k and below height t in container $k + 1$. Hence, n's allocation from $(k; t)$ to $(k; t) \oplus \tau_n$ does not overlap in height with any of $[n + 1 : n + I]$, for each $n \in [0 : N - I - 1]$.

We now consider the other case $n \in [N - I, N - 1]$. Note that all the containers are full since $KT = K\tau_{avg} = \sum_{i=0}^{N-1} \tau_i$. Hence, $\boldsymbol{s}_{N-1} \oplus \tau_{N-1} = (K; T)$. It is now clear that the allocation of segments $[n : N - 1]$ lie between positions $(K; T - \tau_{[n:N-1]})$ and $(K; T)$ in the same container and hence no overlap. The allocations of $[N : n+I]$ (*i.e.* $\{0, \ldots, n+I-$

[3]Note that in Fig. 3(a), the top segment load of each container overflows into the bottom of the next container, and this may appear to imply that there are extra beam switches taking place. However, this is not the case when considering that the subsequent resource allocation block can circularly shift the RF chains one container to the left, which will ensure a continuity of segment loads across the resource block boundary.

$N\}$) lie between positions $(1;0)$ and $(1;\tau_{[N:n+I]})$. There can be no overlap since $T-\tau_{[n:N-1]} \geq \tau_{[N:n+I]}$, which is due to $T=\tau_{avg} \geq \tilde{\tau}_{\max} \geq \tau_{[n:N-1]}+\tau_{[N:n+I]}$. □

B. Scenario with Partial Angular Coverage $\theta < 360°$

In this section, we present another important special case where the Greedy Sand-Filling algorithm is optimal. This case can be either balanced ($\tau_{avg} \geq \tilde{\tau}_{\max}$) or unbalanced ($\tau_{avg} < \tilde{\tau}_{\max}$).

Consider a setup where the angular coverage region of a gNB is a sector of width θ, and not the full circle, *i.e.* $\theta < 360°$, as shown in Fig. 4. There are $N'=22$ segments in Fig. 4 which is less than $N=24$ segments for the full circle in Fig. 1. The framework for the full circle case developed in the previous section also applies to quite general sectored scenarios, such as the one depicted in Fig. 4, using the following theorem.

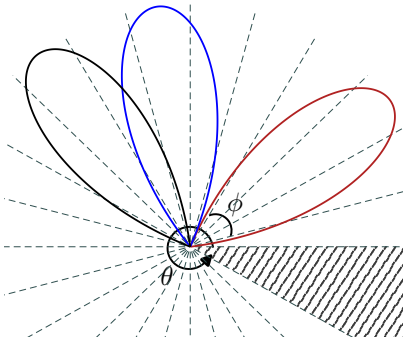


Fig. 4. Illustration with $\theta < 360°$ coverage, i.e., not full circle.

Theorem 3. *The Greedy Sand-Filling algorithm (Algorithm 1) is optimal if* $\tau_{[N-I:N-1]}=0$.

Proof. For segments $n \in [0:N-I-1]$, the proof from the same arguments as given for Theorem 1. Since the $\tau_n=0$ for other segments $n \in [N-I:N-1]$, no allocation is needed. □

Theorem 3 shows that the Greedy Sand-Filling algorithm is always optimal when the final I consecutive segments have zero load. This is the case in Fig. 4 where $I=1$ and the 2 shaded consecutive segments have no UEs. We propose that the clock-wise labelling of segments from 0 to $N-1$ start with the first segment after the shaded region which ensures that the final segments $N-I$ to $N-1$ have a zero load.

More generally, let ϕ denote the angle subtended by a beam, i.e., the beam-width. We will now use Theorem 3 to show that the Greedy Algorithm with the proposed labelling is optimal for any sector of angle $\theta \leq 360-\phi/2$.

Note that $\frac{\phi}{2I+1}$ is the angle subtended by a single segment, since beam-width ϕ is the angle subtended by $2I+1$ consecutive segments under our model. Hence, the number of segments, N, in a full circle is given by

$$N=(2I+1)\frac{360°}{\phi} \tag{10}$$

and the number of segments, N', in an arc of angle θ by

$$N'=(2I+1)\frac{\theta}{\phi} \tag{11}$$

For any arc such that $\theta \leq 360°-\phi/2$, we obtain

$$N' \leq (2I+1)\frac{360°}{\phi}-(I+0.5) \tag{12}$$

$$< N-I \tag{13}$$

and hence the Greedy Sand-Filling algorithm is optimal by Theorem 3. We state this now as a Corollary of Theorem 3.

Corollary 1. *The Greedy Sand-Filling algorithm is optimal for any arc of angle* $\theta \leq 360°-\phi/2$ *with the segment labels for the coverage arc going from* 0 *to* $N'(<N-I)$.

Since the beam-width ϕ in mmWave systems is small (due to the large number of antenna elements), the Greedy Sand-Filling algorithm applies to a wide range of arc coverage scenarios and practically all sector based implementations, from Corollary 1.

VI. Unbalanced Scenarios

In this section, we consider sand-filling for general unbalanced scenarios. We prove that Greedy Sand-Filling algorithm is not optimal in some unbalanced scenarios, and motivate the development of our Generalized Sand-Filling algorithm that will be presented in the next section.

In general, it is *not true* that greedy sand-filling is optimal for *all unbalanced* scenarios, where $\tilde{\tau}_{\max} > \tau_{avg}$. To see how the Greedy Sand-Filling algorithm can fail, consider the example illustrated in Fig. 3(a) but with τ_{N-2} replaced with the value 0 and the other loads staying the same. The resulting greedy sand-filling allocation is illustrated in Fig. 3(b); segment $N-1$ now starts immediately after the completion of segment $N-3$. Note that this means that the allocation of segment $N-1$ now overlaps in height with segment $I-1$ (which is at the same height in column 1), and the greedy sand-filling solution is therefore infeasible.

In the next section, the Generalized Sand-Filling algorithm will be developed based on the shifting of starting positions to obtain a feasible allocation. Note that a feasible allocation for the above example can be constructed from the infeasible greedy solution by shifting the starting position of $N-1$ as shown in Fig. 3(c). The Generalized Sand-Filling algorithm (that we will provide in Section VII) shifts coloured sand starting positions in a similar manner, with the same container height $T=\max\{\tau_{avg},\tilde{\tau}_{\max}\}$.

Unfortunately, even the Generalized Sand-Filling algorithm is not guaranteed to find a feasible solution for *all* unbalanced scenarios, *i.e.* for arbitrary $\boldsymbol{\tau}$. This is because there are cases (*i.e.* $\boldsymbol{\tau}$) for which $f^* > \max\{\tilde{\tau}_{\max},\tau_{avg}\}=T$ as stated in the following Theorem 4, *e.g.* For $\boldsymbol{\tau}=[3,2,3,2,3,0,5,0,3,2]$ with 3 RF chains, $\tau_{avg}=23/3$, $\tilde{\tau}_{\max}=8$, whereas $f^*=8.5$. More general examples are constructed for in the proof of Theorem 4, which can be found in the Appendix.

Theorem 4. *For* $K \geq 2$ *and* $I \geq 2$, *if* $I+1 < N \leq (2K-1)I$ *then there exist* $\boldsymbol{\tau} \in \mathbb{R}_+^N$ *such that* $f^* > \max\{\tau_{avg},\tilde{\tau}_{\max}\}$.

Proof. See Appendix B. □

The Sand-filling allocations (greedy or generalized) cannot be optimal for these examples since the container height is fixed at T in both versions of the algorithm, whereas the optimal value $f^* > T$. Both sand-filling algorithms terminate with a final allocation of sand which fits into the K containers. The condition $T \geq \tau_{avg}$ ensures that there is always sufficient container space for the sand. However, the feasibility of the resulting beam segment schedule is not guaranteed. Indeed, when $f^* > T$, the beam segment schedule obtained by the algorithms (Greedy or Generalized) must be infeasible.

In the next section, we will define a Generalized Sand-Filling algorithm and show that it is optimal for any N greater than the bound in Theorem 4; *i.e.* for

$$N \geq (2K-1)I + 1, \tag{14}$$

Consequently, $f^* = T = \tilde{\tau}_{\max}$ in unbalanced scenarios when (14) also holds. We will call (14) the *large number of segments* condition.

The large number of segments condition does not limit practical implementation of the algorithms for mm-Wave hybrid beamforming architectures. As discussed in Section I, in such architectures, the number of antenna elements at the gNB is large (which leads to narrow beam-widths ϕ) and the number of RF chains, K, is small. Therefore, the product $(K-0.5)\phi$ will be small. In this situation, the large number of segments condition will hold due to the following Lemma 3, which shows that (14) holds if $(K-0.5)\phi \leq 360°$. For example, if $K = 10$ RF chains, (14) holds for beam-widths ϕ up to $38°$.

Lemma 3. *If the beam-width ϕ and the number of RF chains K are such that $(K-0.5)\phi \leq 360°$, then the large number of segments condition in* (14) *holds.*

Proof. Recall that $360° = N\phi/(2I+1)$ (from Section V-B) under our model. Hence we obtain $(K-0.5)\phi \leq N\phi/(2I+1)$. Hence $N \geq (2K-1)I + (K-0.5)$.

Since N is an integer and $K \geq 1$, $N \geq (2K-1)I+1$. □

VII. Generalized Sand Filling Algorithm for Unbalanced Scenarios

In this section, we present our Generalized Sand-Filling algorithm, which is a modified version of greedy sand-filling. We will show that it is optimal in all unbalanced scenarios satisfying the *large number of segments condition* $N \geq (2K-1)I+1$.

We use $T = \tilde{\tau}_{\max}$ for the container height since $\tilde{\tau}_{\max} > \tau_{avg}$ in unbalanced scenarios. With no loss of generality, we assume that

$$\tilde{\tau}_{\max} = \tau_{[0:I]} \tag{15}$$

and that $K \geq 2$ and $N \geq 2I+2$, since 1) $K = 1$ is always a balanced scenario, and 2) if $N \leq 2I+1$, then there is effectively only 1 useful RF chain (*i.e.* $\max_{S\in\mathcal{S}} = 1$), and if the problem is re-formulated with $K = 1$, we see that the scenario is balanced and the Greedy Sand-Filling algorithm is optimal. In fact, scenarios with $N \leq 2I+1$ violate our earlier assumption that $\max_{S\in\mathcal{S}} = K$ (see Footnote 2 on page 3).

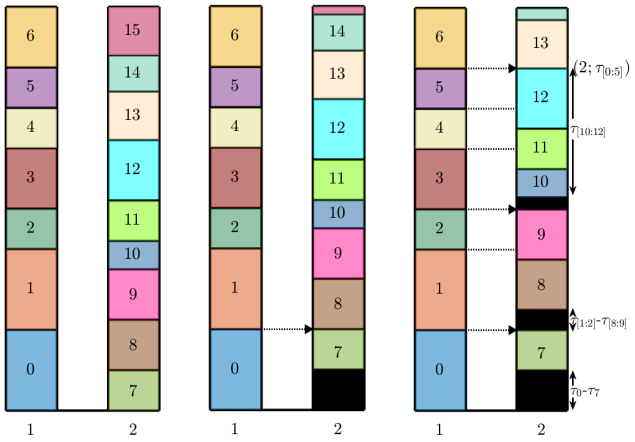
The following is the intuition behind our Generalized Sand-Filling algorithm. Since $\tilde{\tau}_{\max} > \tau_{avg}$ in this case, the total volume of the containers, $K\tilde{\tau}_{\max}$, is greater than the total sand $\sum_{n=0}^{N-1}\tau_n = K\tau_{avg}$. Hence, with the Greedy Sand-Filling algorithm (which can result in infeasible allocations in this case, due to segment allocation conflicts between containers, as mentioned previously), there is an empty space at the top of the final container, of height $K(\tilde{\tau}_{\max} - \tau_{avg}) = KT - \tau_{[0:N-1]}$.

Our key new idea in the Generalised Sand-Filling algorithm is the concept of *black* sand. This sand does not correspond to beam allocations, and is simply a filler. We denote the amount of black sand by ψ. The maximum amount of black sand we can use is given by the amount of empty space at the top of the final container after greedy sand-filling. We will show how to pour black sand into the containers in such a way that segment allocation conflicts between containers do not occur. In other words, we will pour it between sands of different colours, which shifts colours up (and to the right), in such a way that the resulting final allocation is feasible. The empty space in the final container is therefore reduced, and has effectively been used to resolve conflicts. We will provide specific details in the next section, but first we present three cases to illustrate the operation of black sand pouring and the resultant shifting of colored sand.

Fig. 5 shows examples for three different amounts of black sand, in each case showing the first two columns of a larger system with $I = 6$. Fig. 5(a) shows the greedy sand-filling solution when no black sand is used. Fig 5(b) shows an example where $\psi = \tau_0 - \tau_7$ and the sand has been poured in between the 6-th and the 7-th segments, shifting all segments above 6 upwards. In this case the top of the 7th segment aligns with the top of the 0 segment in the first column. Note that it is not possible to pour any more black sand between these two segments because a conflicting overlap would then exist between segment 7 and segment 1 in column 1. Fig 5(c) shows an example where more black sand is added into the system. In this example, the total amount of black sand added is $\psi = \tau_{[0:5]} - \tau_{[7:12]}$, which includes the initial amount of $\tau_0 - \tau_7$ between segments 6 and 7 (as above). Additional black sand of height $\tau_{[1:2]} - \tau_{[8:9]}$ is poured between segments 7 and 8, shifting all segments above 7 upwards. In this case, 9 is at its upper limit, due to the potential for conflict with segment 3 in column 1. The remaining black sand of height $\tau_{[3:5]} - \tau_{[10:12]}$ is added between segments 9 and 10. For larger values of ψ, additional amounts of black sand can be poured in a similar fashion between higher index segments, always taking account of any potential conflicts.

As illustrated, the black sand is inserted by shifting (*i.e.* increasing) the starting positions of segments in $[I+1 : N-I-1]$. It is clear that there are upper bounds on the shifted starting positions. We denote the upper bound for segment i, $\boldsymbol{u}_i$, for $i = 0, \ldots, N-I-1$. Each segment is shifted upwards by ψ unless that causes a conflict, in which case they are shifted up to their upper bound $\boldsymbol{u}_i$. Let $\boldsymbol{s}_i$ denote the starting position of i under greedy sand-filling, which corresponds to $\psi = 0$. For $\psi > 0$, the new starting position for segment i,

This article has been accepted for publication in IEEE Transactions on Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIT.2024.3399760



(a) $\psi = 0$ (b) $\psi = \tau_0 - \tau_7$ (c) $\psi = \tau_{[0:5]} - \tau_{[7:12]}$

Fig. 5. Illustration of Black Sand Pouring and starting position upper bounds.

denoted by $\boldsymbol{s}_i^{\psi}$ is

$$\boldsymbol{s}_i^{\psi} := \min(\boldsymbol{s}_i \oplus \psi, \boldsymbol{u}_i). \tag{16}$$

Hence, if $\boldsymbol{u}_i$ is known, the shifted allocation after pouring any amount of black sand $\psi > 0$ can be calculated using (16).

In the Subsection VII-A, we determine the upper-bounds $\{\boldsymbol{u}_i\}_{i=0}^{N-I-1}$ in general. In Subsection VII-B, we provide our Generalized Sand-Filling algorithm, which chooses ψ equal to $K(T - \tau_{avg})$, which is the height of the empty space in container K after greedy sand-filling. We remark here that the purpose of black-sand pouring is not necessarily to help the segments in $[0 : N - I - 1]$. For example, in Fig 5(a), segment 7 does not conflict with segment 0, yet our choice of ψ may result in it shifting up, as illustrated in Fig 5(b). Indeed, the results in Section V-B (the partial angular coverage scenario) indicate that there will be no conflicts for segments in $[0 : N-I-1]$ with greedy sand-filling, even in unbalanced scenarios. However, conflicts can occur for the final segments $[N - I : N - 1]$ with the segments $[0 : I]$ in container 1, due to wraparound. We will show that the pouring of black sand, thereby shifting segments $[I + 1 : N - I - 1]$, provides the necessary flexibility for shifting the final segments $[N - I : N - 1]$ to resolve these conflicts.

A. Starting position upper bounds on shifting

We now determine the upper-bounds $\{\boldsymbol{u}_i\}_{i=0}^{N-I-1}$ in the general-case which enable the computation of the shifted starting positions (and hence allocations) for any $\boldsymbol{\tau}$. Since the allocations of segments 0 to I are fixed in the first column and do not shift, for $i = 0, \ldots, I$, we define

$$\boldsymbol{u}_i := \boldsymbol{s}_i = \left(1; \sum_{j=0}^{i} \tau_j - \tau_i\right) \tag{17}$$

As illustrated in Fig. 5, the upper bound for a segment $i \in [I + 1 : N - 1 - I]$ occurs either 1) due to *direct blocking* by $i-I$, (*e.g.* $i = 7$ in Fig. 5(b), where $I = 6$), or 2) due to *indirect blocking* by a segment $i+l-I$, where $i+l-I$ blocks all the segments in $[i : i+l]$ for some $l \in [0 : I-1]$. In Fig. 5(c) where $I = 6$, $i = 10$ is indirectly blocked by segment $6 = i + l - I$ (*i.e.* $l = 2$). From the figure, note that $\boldsymbol{u}_6 = (1; \tau_{[0:5]})$ and $\boldsymbol{u}_{10} = (2; \tau_{[0:5]}) \ominus \tau_{[10:12]} = \boldsymbol{u}_{i+l-I} \oplus (T - \tau_{[i:i+l]})$. For a segment $i \geq I + 1$, suppose that $i + l \leq N - I - 1$ for each $l \in [0 : I - 1]$, we define the upper bound $\boldsymbol{u}_i$ as

$$\boldsymbol{u}_i := \min_{l \in [0:I-1]} \boldsymbol{u}_{i+l-I} \oplus \left(T - \tau_{[i:i+l]}\right) \tag{18}$$

where $l = 0$ in the RHS corresponds to the bound if i is directly blocked by $i - I$, and $l \in [1 : I - 1]$ corresponds to indirect blocking by $i + l - I$. The minimum is taken over all possible cases l, since all conflicts (*i.e.* no overlap conditions) must be satisfied for feasibility.

In general, l must be less than or equal to $N - I - 1 - i$ so that the term $i+l$ which appears in the RHS of (18) is less than or equal to $N - I - 1$, since only segments $[I+1 : N-I-1]$ are under consideration for upper bounds. Hence, we define the upper bounds $\boldsymbol{u}_i$ for $i = I + 1$ to $N - I - 1$ inductively as follows.

$$\boldsymbol{u}_i := \min_{l \in [0:\min\{I-1, N-I-1-i\}]} \boldsymbol{u}_{i+l-I} \oplus \left(T - \tau_{[i:i+l]}\right). \tag{19}$$

The shifted allocations for $i \in \{0, \ldots, N - I - 1\}$ are now fully defined since the upper bounds $\boldsymbol{u}_j$ are defined. Note that $N - I - 1 \geq I + 1$ due to our assumption that $N \geq 2I + 2$.

We now present the following Lemma, which shows that the black sand pouring procedure (and the shifting) does not lead to conflicting overlaps among the segments in $[0 : N - I - 1]$. Following that, we will provide our Generalized Sand-Filling Algorithm and discuss the shifting of final segments $[N - I : N - 1]$.

Lemma 4 (Shifting Lemma)**.** *For any* $0 \leq \psi \leq KT - \tau_{[0:N-1]}$*, with the new starting positions given by (16), the allocation of a segment* $n \in [I + 1 : N - I - 1]$ *does not overlap with the allocation of* $m \in [n - I : n - 1]$.

Proof. From Lemma 8 (in Appendix C), we have

$$|\boldsymbol{u}_{i+1}| - \tau_{[0:i]} \geq |\boldsymbol{u}_i| - \tau_{[0:i-1]} \tag{20}$$

for each $i \in [0 : N - I - 1]$. Note that $|\boldsymbol{s}_i| = \tau_{[0:i-1]}$ is the magnitude of sand-filled starting position $\boldsymbol{s}_i$. Hence, $|\boldsymbol{u}_i| - \tau_{[0:i-1]}$ is the minimum shift (*i.e.* amount of black sand) required for segment i to reach its upper bound $\boldsymbol{u}_i$. Hence from (20), we observe that

1) If ψ is not large enough for segment i to reach its upper bound, then all the segments $j > i$ also do not reach their upper bounds, *i.e.* if $\psi < |\boldsymbol{u}_i| - \tau_{[0:i-1]}$ then $\boldsymbol{s}_j^{\psi} < \boldsymbol{u}_j$ for each $j \geq i$.
2) If ψ is large enough for segment i to reach its upper bound, then all the segments $j < i$ also reach their upper bounds, *i.e.* if $\psi \geq |\boldsymbol{u}_i| - \tau_{[0:i-1]}$ then $\boldsymbol{s}_j^{\psi} = \boldsymbol{u}_j$ for each $j \leq i$.

We now apply the above observations, 1) and 2), to prove the lemma. Consider an arbitrary pair of segments $n \in [I+1 : N - I - 1]$ and $m \in [n - I : n - 1]$.

Suppose first that $\boldsymbol{s}_m^{\psi} < \boldsymbol{u}_m$. Since $m < n$, from 1) we obtain $\boldsymbol{s}_n^{\psi} = \boldsymbol{s}_n \oplus \psi$ and $\boldsymbol{s}_m^{\psi} = \boldsymbol{s}_m \oplus \psi$. Let $\boldsymbol{s}_m^{\psi}$ be $(k; h)$; Then $\boldsymbol{s}_n^{\psi} = (k; h) \oplus \tau_{[m:n-1]}$ (since $|\boldsymbol{s}_n^{\psi}| - |\boldsymbol{s}_m^{\psi}| = \tau_{[m:n-1]}$). Therefore, $\boldsymbol{s}_n^{\psi} \geq (k; h) \oplus \tau_m$ (since $\tau_{[m:n-1]} \geq \tau_m$), and $\boldsymbol{s}_n^{\psi} \leq (k + 1; h) \ominus \tau_n$ (since $\tau_{[m:n-1]} \leq \tau_{[n-I:n-1]} \leq \tilde{\tau}_{\max} - \tau_n$).

Hence for n's allocation, the starting position is $\boldsymbol{s}_n^{\psi} \geq (k;h) \oplus \tau_m$ and the ending position is $\boldsymbol{s}_n^{\psi} \oplus \tau_n \leq (k+1;h)$. No overlap is possible since n's allocation lies above m's allocation in container k and it lies below m's starting height h in the next container $k+1$.

Suppose instead that $\boldsymbol{s}_m^{\psi} = \boldsymbol{u}_m$. Let $\boldsymbol{s}_m^{\psi} = \boldsymbol{u}_m = (k;h)$. By definition of $\boldsymbol{u}_n$,

$$\begin{aligned} \boldsymbol{s}_n^{\psi} &\leq \boldsymbol{u}_n \leq \boldsymbol{u}_m \oplus (T - \tau_{[n:m+I]}) \\ &\leq (k+1;h) \ominus \tau_n. \end{aligned} \tag{21}$$

If $\boldsymbol{s}_n^{\psi} = \boldsymbol{u}_n$, then $|\boldsymbol{s}_n^{\psi}| - |\boldsymbol{s}_m^{\psi}| = |\boldsymbol{u}_n| - |\boldsymbol{u}_m| \geq \tau_{[m:n-1]}$ from (20). Suppose instead $\boldsymbol{s}_n^{\psi} < \boldsymbol{u}_n$, which implies $\boldsymbol{s}_n^{\psi} = \boldsymbol{s}_n \oplus \psi = \boldsymbol{s}_m \oplus (\tau_{[m:n-1]} + \psi)$. Since $\boldsymbol{s}_m \oplus \psi \geq \boldsymbol{s}_m^{\psi}$, we obtain $|\boldsymbol{s}_n^{\psi}| - |\boldsymbol{s}_m^{\psi}| \geq \tau_{[m:n-1]}$. Thus, in either case, $|\boldsymbol{s}_n^{\psi}| - |\boldsymbol{s}_m^{\psi}| \geq \tau_{[m:n-1]} \geq \tau_m$, from which it follows that $\boldsymbol{s}_n^{\psi} \geq (k;h) \oplus \tau_m$. Also, $\boldsymbol{s}_n^{\psi} \oplus \tau_n \leq (k+1;h)$ from (21). Hence, no overlap is possible just as in the previous case, which completes the proof. □

B. Generalized Sand Filling Algorithm using Black Sand pouring

Algorithm 2 Generalized Sand Filling Algorithm

1: $T = \max\{\tilde{\tau}_{\max}, \tau_{avg}\}$.
2: $\boldsymbol{p} = (1;0)$ and $\boldsymbol{s}_j = (1;0)$ for $j = 0, \ldots, N-1$.
3: **for** $i = 0$ to $N-1$ **do** // Calculate the greedy sand-filling starting positions of segments
4: $\quad \boldsymbol{s}_i = \boldsymbol{p}$.
5: $\quad \boldsymbol{p} := \boldsymbol{p} \oplus \tau_i$.
6: **end for**
7: $\psi := KT - \tau_{[0:N-1]}$ // Amount of black sand chosen to be the maximum value
8: **for** $i = I+1$ to $N-I-1$ **do**
9: $\quad \boldsymbol{s}_i^{\psi} := \min(\boldsymbol{u}_i, \boldsymbol{s}_i \oplus \psi)$ // Shifted starting position cannot be greater than the upper bound $\boldsymbol{u}_i$.
10: $\quad$ Allocate the positional range from $\boldsymbol{s}_i^{\psi}$ to $\boldsymbol{s}_i^{\psi} \oplus \tau_i$ to segment i.
11: **end for**
12: **for** $i = N-I$ to $N-1$ **do** // Shift the final segments by ψ.
13: $\quad \boldsymbol{s}_i^{\psi} := \boldsymbol{s}_i \oplus \psi$
14: $\quad$ Allocate the positional range from $\boldsymbol{s}_i^{\psi}$ to $\boldsymbol{s}_i^{\psi} \oplus \tau_i$ to segment i.
15: **end for**

Our Generalized Sand-Filling Algorithm is presented in Algorithm 2. In lines 2-6, the initial starting positions $\boldsymbol{s}_i$ are computed according to the greedy sand-filling criterion. Of course, we know that this may not provide a feasible solution in this unbalanced scenario, in which case black sand will be required. The algorithm chooses an amount of black sand equal to the height of the empty space in the final container (in line 7). In other words, it sets ψ to be $KT - \tau_{[0:N-1]}$. In lines 8-11, the new starting positions $\boldsymbol{s}_i^{\psi}$ for $i = I+1, \ldots, N-I-1$ are computed according to the shifts induced by black sand pouring, as described in the previous section. By Lemma 4, no conflicts occur among the segments in $[0 : N-I-1]$.

C. Shifting of final segments in $[N-I : N-1]$

The black sand used between the segments in $[0 : N-I-1]$ in Algorithm 2 is $|\boldsymbol{s}_{N-I-1}^{\psi}| - |\boldsymbol{s}_{N-I-1}|$, which is smaller than or equal to ψ. The remaining black sand (*i.e.* $\psi - |\boldsymbol{s}_{N-I-1}^{\psi}| + |\boldsymbol{s}_{N-I-1}|$) is added between segments $N-I-1$ and $N-I$, which results in all the segments in $i \in [N-I : N-1]$ being shifted up by ψ (from their greedy sand-filled positions), in lines 12-15 of Algorithm 2.

The following Lemma shows that there are no conflicting overlaps for the final segments $[N-I : N-1]$ with the segments $[0 : I-1]$ (which are in the clock-wise direction).

Lemma 5. *In Algorithm 2, the allocation of a segment $n \in [N-I : N-1]$ does not overlap with the allocation of any segment $m \in [n+1 : n+I]$.*

Proof. Since the amount of black sand ψ equals the height of empty space, the new ending position of $N-1$ shifts to the top of the final container, *i.e.* $\boldsymbol{s}_{N-1}^{\psi} \oplus \tau_{N-1} = (K;T)$. It is now clear that the new allocations of segments $[n : N-1]$ lie between positions $(K;T - \tau_{[n:N-1]})$ and $(K;T)$, *i.e.* above height $T - \tau_{[n:N-1]}$ in the same (final) container, and hence no overlap. The allocations of $[N : n+I]$ (*i.e.* $\{0, \ldots, n+I-N\}$) lie between positions $(1;0)$ and $(1;\tau_{[N:n+I]})$, *i.e.* below height $\tau_{[N:n+I]}$ in the first container. There can be no overlap since $T - \tau_{[n:N-1]} \geq \tau_{[N:n+I]}$, which is due to $T = \tilde{\tau}_{\max} \geq \tau_{[n:n+I]} = \tau_{[n:N-1]} + \tau_{[N:n+I]}$. □

D. Proof of Optimality

To show that the shifted allocation from Algorithm 2 is optimal, it remains to check that there are no conflicts between any segment $n \in [N-I : N-1]$ and the segments in $[n-I : n-1]$, *i.e.* there are no conflicting overlaps for the final segments in the anti clock-wise direction. This will be shown in the following Theorem 5 by making use of the large number of segments condition. The following lemma provides the key result needed for this step.

Lemma 6. *If $N \geq (2K-1)I+1$ and $K \geq 2$, then $\boldsymbol{u}_{N-2I} \geq (K;0)$.*

Proof. If $N \geq (2K-1)I+1$ then $N-2I \geq (2K-3)I+1$. From Lemma 8 (in Appendix C), $\boldsymbol{u}_{N-2I} \geq \boldsymbol{u}_{(2K-3)I+1}$. By (60) in Lemma 10 (in Appendix C), taking $k = K-1$, we have that $\boldsymbol{u}_{(2K-3)I+1} \geq (K;0)$. □

Lemma 6 says that the upper bound for $N-2I$ is in the final container K (which we note cannot be container 1, since $K \geq 2$ is assumed in the lemma) when the large number of segments condition holds. This fact is crucial in the proof of Theorem 5 below for unbalanced scenarios.

Theorem 5. *The Generalized Sand-Filling algorithm (Algorithm 2) is optimal for any $\boldsymbol{\tau} \in \mathbb{R}_+^N$ given $N \geq (2K-1)I+1$.*

Proof. For the balanced case $\tau_{avg} \geq \tilde{\tau}_{\max}$, note that $\psi = 0$, since $\psi = K(T - \tau_{avg})$ and $T = \max\{\tau_{avg}, \tilde{\tau}_{\max}\}$. In this case, since there is no shift (*i.e.* no black sand), Algorithm 2 is equivalent to Algorithm 1. Hence, the proof follows from

This article has been accepted for publication in IEEE Transactions on Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIT.2024.3399760

Theorem 2. In the unbalanced case that we consider below, we assume without any loss of generality that $K \geq 2$.[4]

For the unbalanced case with $\psi > 0$, Lemma 4 shows that there is no conflicting overlaps among segments in $[0 : N - I - 1]$. Lemma 5 shows that no conflicting overlaps of final segments $n \in [N - I : N - 1]$ with the clock-wise segments $[n : n + I]$. We now show that the allocation of a segment $n \in [N - I : N - 1]$ does not overlap with a segment $m \in [n - I : n - 1]$.

If $m \geq N - I$, no overlap is possible since the allocations of m and n both completely lie in the final container K (as the sand of $[N-I : N-1]$ is shifted together to the top of the final container after black sand pouring). We now consider the other case $m \in [n-I : N-I-1]$. In this case, $\boldsymbol{u}_m \geq (K; 0)$ by Lemma 6, since $m \geq N - 2I$. If $\boldsymbol{s}_m^{\psi} = \boldsymbol{u}_m$, there is no overlap since allocations of both m and n lie in the final container K.

If instead $\boldsymbol{s}_m^{\psi} < \boldsymbol{u}_m$, then $|\boldsymbol{s}_m^{\psi}| = \tau_{[0:m-1]} + \psi$ from Observation 1) in the proof of Lemma 4. This implies that $|\boldsymbol{s}_m^{\psi}| = KT - \tau_{[m:N-1]}$, since $\psi = KT - \tau_{[0:N-1]}$. Hence we obtain

$$\begin{aligned} \boldsymbol{s}_m^{\psi} &= (K; T) \ominus \tau_{[m:N-1]} && (22)\\ &= (K; T - \tau_{[n:N-1]}) \ominus \tau_{[m:n-1]} && (23)\\ &= \boldsymbol{s}_n^{\psi} \ominus \tau_{[m:n-1]} && (24) \end{aligned}$$

since $\boldsymbol{s}_n^{\psi} = (K; T - \tau_{[n:N-1]})$ from line 12 of Algorithm 2.

Let $\boldsymbol{s}_n^{\psi} = (K; h)$. Hence, $\boldsymbol{s}_m^{\psi} \geq (K - 1; h) \oplus \tau_n$ from (24) (since $\tau_{[m:n-1]} \leq \tilde{\tau}_{\max} - \tau_n \leq T - \tau_n$). This shows that any part of m's allocation in container $K - 1$ is above the height of n's allocation in container K. Any part of m's allocation in container K lies below n's allocation by construction. Thus, no overlap is possible. □

VIII. Capacity Results

In this section, we present capacity characterization results arising from our sand-filling algorithms. We also illustrate the capacity gain of our approach over traditional interference avoidance schemes (specifically resource partitioning), with a numerical example.

Recall that the capacity region $\mathcal{C}$ of flow rate vectors was defined in (5) as the convex hull of rate vectors in the set $\mathcal{R}$, defined in (4), which is a set with a large number of elements, exponential in the number of segments, N. We now use Theorems 1 and 5 to provide a much simpler characterization of $\mathcal{C}$ in terms of the flow rate vectors $\boldsymbol{\nu} = [\nu_u]_{u \in U}$. Recall that $\tau_n = \sum_{u \in U_n} \nu_u / R_u$ for each $n = 0, \ldots, N - 1$, where ν_u (and R_u) is the flow-rate (and the instantaneous rate) of UE u, and U_n is the set of all the UEs in segment n.

A. Capacity Region when $N \geq (2K - 1)I + 1$

Theorem 6 provides a full characterization of the capacity region with only $N+1$ linear inequality constraints when $N \geq (2K-1)I+1$. Clearly, (25)-(26) are necessary conditions, and we have shown that they are sufficient when $N \geq (2K-1)I+1$.

[4]Recall that $K = 1$ always corresponds to a balanced scenario since $\tau_{avg} = \tau_{[0:N-1]} \geq \tilde{\tau}_{\max}$.

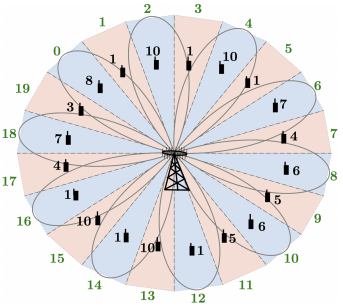


Fig. 6. Numerical Example with Resource Partitioning.

Theorem 6. *If $N \geq (2K - 1)I + 1$, then the capacity region $\mathcal{C}$ is the set of $\boldsymbol{\nu} \in \mathbb{R}_+^{|U|}$ such that*

$$\sum_{i \in [n:n+I]} \sum_{u \in U_i} \frac{\nu_u}{R_u} \leq 1, \ n = 0, \ldots, N - 1 \quad (25)$$

$$\sum_{u \in U} \frac{\nu_u}{R_u} \leq K \quad (26)$$

Proof. (25) is equivalent to $\tilde{\tau}_{\max} \leq 1$ and (26) is $\tau_{avg} \leq 1$. From Theorem 5, $f^* \leq 1$. The result now follows from Theorem 1. □

B. Numerical Example

We now use Theorem 6 to demonstrate the advantage of our capacity achieving sand-filling algorithms, over traditional resource partitioning based wireless interference avoidance approaches, using a numerical example with $K = 10$ RF chains, $N = 20$ segments, and $I = 1$. This is illustrated in Fig. 6, where the segment index is shown around the outside (in green), and the number of UEs in each segment is shown inside the segment (in black). For this example, we consider a case where all UEs are allocated the same amount of time, and as such receive a flow rate, $\nu_u = \tau R_u$, where τ is the constant of proportionality, the same for all users[5]. The value of τ is maximised on the capacity region boundary, which results in the highest flow rates.

Recall that our sand-filling approach avoids inter-beam interference by carefully managing the constraint of selecting segments that are appropriately separated in angle. In this subsection we compare our sand-filling algorithm to a traditional resource partitioning approach that avoids inter-beam interference by dividing the segments into two groups of 10 alternating segments (one segment for each RF chain/beam), and time-sharing between the two groups. The two groups are shown in blue and pink in Fig. 6. The capacity region for our sand filling approach in this numerical example, is given by $11\tau \leq 1$, from Theorem 6, where the bottle neck constraint is (25) which occurs on segment $n = 4$ (and also on segments $n = 1, 2, 3, 6, 8, 9, 10, 13, 14, 15, 17, 19$).[6]

[5]This example can also model the case of having a single user in each segment, where the users have differentiated time requirements.

[6]Note that the constraint (26) in this case is $101\tau < 10$.

This can be seen by observing that segment $n = 4$ has $\tau_{[n:n+I]} = 10\tau + 1\tau = 11\tau$ which is the $\max_{n'=0}^{N-1} \tau_{[n':n'+1]}$ over all segments. The capacity region for the traditional approach is given by $20\tau \leq 1$, which is determined by the need to serve the bottle neck segments in each group, which are $n = 2$ (or equivalently $n = 4$) for the blue group and $n = 13$ (or equivalently $n = 15$) for the pink group.

This shows that the capacity of our sand-filling scheduling algorithm is $(\frac{1}{11} - \frac{1}{20})/\frac{1}{20} = 82\%$ larger compared to the traditional partitioning scheme.

C. Capacity Region when $N < (2K-1)I + 1$

Theorem 6 characterizes the capacity region when the large number of segments condition, $N \geq (2K-1)I + 1$, holds. We claim that the large number of segments condition should always hold in a practical deployment of a mmWave base station with narrow beams, since the product of beamwidth and number of RF chains is small (see Lemma 3). Nevertheless, we can provide achievable flow rate regions, even when $N < (2K-1)I + 1$, by using Theorems 1-3, as we now show.

For $K \geq 2$, we note that N must actually lie in $[K(I+1) : (2K-1)I]$ for K RF chains to be useful. As noted in Footnote 2, let $\bar{K} := \max\{n \geq 1 : n(I+1) \leq N\}$, and if $\bar{K} < K$ then K should be reset. More specifically, if $N < K(I+1)$ (before resetting) then $\max_{S\in\mathcal{S}} |S| = \bar{K} < K$ and hence only $\bar{K}$ RF chains are useful. As in Footnote 2, we therefore reset K to be $\bar{K}$, so that all K RF chains are used. The reset causes N to be greater than or equal $K(I+1)$ and K to be equal to $\max_{S\in\mathcal{S}} |S|$.

For the regime $N \in [K(I+1) : (2K-1)I]$ (in which K RF chains are useful), we are able to characterize two achievable flow rate regions as follows. Theorem 7 provides an achievable rate region which follows from our Theorem 2 on balanced scenarios.

Theorem 7. *For $N \in [K(I+1) : (2K-1)I]$, a given $\boldsymbol{\nu} \in \mathbb{R}_+^{|U|}$ is in the capacity region $\mathcal{C}$ if*

$$\sum_{i\in[n:n+I]} \sum_{u\in U_i} \frac{\nu_u}{R_u} \leq \frac{1}{K} \sum_{u\in U} \frac{\nu_u}{R_u}, \ n = 0, \ldots, N-1 \quad (27)$$

$$\sum_{u\in U} \frac{\nu_u}{R_u} \leq K \quad (28)$$

Proof. (27) implies the balanced loads, *i.e.* $\tau_{avg} \geq \tilde{\tau}_{\max}$, and (28) implies $\tau_{avg} \leq 1$. From Theorem 2, $f^* \leq 1$. The result is now immediate from Theorem 1. □

Theorem 8 provides an achievable flow rate region, which follows from our Theorem 3 on the coverage arc case. Here, the scenario can be either balanced or unbalanced.

Theorem 8. *For $N \in [K(I+1) : (2K-1)I]$, a given $\boldsymbol{\nu} \in \mathbb{R}_+^{|U|}$ is in the capacity region $\mathcal{C}$ if $\sum_{n\in[j:j+I-1]} \sum_{u\in U_n} \frac{\nu_u}{R_u} = 0$ for some $j \in [0 : N-1]$ and*

$$\sum_{i\in[n:n+I]} \sum_{u\in U_i} \frac{\nu_u}{R_u} \leq 1, \ n = 0, \ldots, N-1 \quad (29)$$

$$\sum_{u\in U} \frac{\nu_u}{R_u} \leq K \quad (30)$$

Proof. Since $\sum_{n\in[j:j+I-1]} \sum_{u\in U_n} \frac{\nu_u}{R_u} = 0$, it is given $\sum_{n\in[j:j+I-1]} \tau_n = 0$ for some $j \in [0 : N-1]$. Re-label the segments such that i becomes $i - j - I \mod N$ for each $i \in [0 : N-1]$. Under the new labelling, $\tau_{[N-I:N-1]} = 0$ (which corresponds to $\tau_{[j:j+I-1]}$ under the original labelling). (28) is equivalent to $\tilde{\tau}_{\max} \leq 1$ and (29) is $\tau_{avg} \leq 1$. From Theorem 3, we obtain, $f^* \leq 1$. The result is now immediate from Theorem 1. □

We also note that when $N \leq 2I + 1$, $\max_{S\in\mathcal{S}} = 1$, so only 1 RF chain is useful. In this scenario, the capacity is characterized by a single linear inequality: $\sum_{n\in[0:N-1]} \tau_n \leq 1$ or equivalently $\sum_{u\in U} \frac{\nu_u}{R_u} \leq 1$, which is clearly the capacity achievable by TDMA. This is also a special case of our results when $K = 1$, since all the sand is filled in one container, the container height $T = f^* = \sum_{n\in[0:N-1]} \tau_n$.

Note that while Theorems 7 and 8 establish feasible flow rate regions when $N \in [K(I+1) : (2K-1)I]$, it is yet to be established what the full capacity region is. We have shown that f^* can be greater than $\tilde{\tau}_{\max} > \tau_{avg}$ in this regime, hence extra constraints are needed to fully characterize the capacity region.

IX. Conclusions

In this paper we have provided capacity results for a mmWave system with quantized analog beamforming codebooks. We have shown that the capacity region is determined by at most $N+1$ linear inequality constraints, under the large number of segments condition, in spite of the fact that the number of feasible beam scheduling sets is exponential in N. We have shown that the large number of segments condition applies to practical mmWave systems. We also presented capacity-achieving scheduling algorithms that provide beam allocations guaranteeing that UE rate requirements are met within each resource block. In particular, we provided provably optimal "sand filling" algorithms which have complexity that is linear in N and I.

Finally, we note that the size of the beamforming codebook, N, is a design parameter that controls the resolution of beam-pointing. Generally speaking, we would like N to be as large as possible, since this maximizes the capacity region, however implementation constraints will limit it in practice. Such constraints include the resolution of the phase shifters, limitations in the feedback channel rate, and the overhead in beam acquisition. We have provided insights into the tradeoff between N and the time taken for beam acquisition, when we observed that it will be necessary in practice to operate at less than the full capacity to provide time within the resource allocation block to do beam acquisition.

Appendix A: Proofs of Theorem 1 and Lemma 1

Proof of Theorem 1. Firstly, we show that $\boldsymbol{\nu} \in \mathcal{C}$ implies $f^* \leq 1$.

By definition of convex hull, there exists $\boldsymbol{a} := [a_{S,V}] \in \mathbb{R}_+^{|\mathcal{S}|\times|\mathcal{V}|}$ such that $\sum_{S\in\mathcal{S},V\in\mathcal{V}} a_{S,V} = 1$ and satisfying

$$\boldsymbol{\nu} = \sum_{S\in\mathcal{S}} \sum_{V\in\mathcal{V}} a_{S,V} \boldsymbol{r}_{S,V}. \tag{31}$$

Note that $\tau_n = \sum_{u\in U_n} \nu_u/R_u$. Hence from (3),

$$\tau_n = \sum_{u\in U_n} \sum_{S\in\mathcal{S}} \sum_{V\in\mathcal{V}} a_{S,V} \mathbb{I}_{n\in S} \mathbb{I}_{u\in V} \tag{32}$$

We define $\mathcal{S}_n := \{S \in \mathcal{S} : n \in S\}$ for each $n \in [0 : N-1]$ and $\mathcal{V}_u := \{V \in \mathcal{V} : u \in V\}$ for each $u \in U$. Also, let $f_S = \sum_{V\in\mathcal{V}} a_{S,V}$ for each $S \in \mathcal{S}$. Hence from (32),

$$\tau_n = \sum_{S\in\mathcal{S}_n} \sum_{u\in U_n} \sum_{V\in\mathcal{V}_u} a_{S,V} \tag{33}$$

$$= \sum_{S\in\mathcal{S}_n} f_S \tag{34}$$

where (34) follows since $\mathcal{V} = \bigcup_{u\in U_n} \mathcal{V}_u$ for each $n \in [0 : N-1]$, because each $V \in \mathcal{V}$ contains exactly one $u \in U_n$ for each n. Hence, $\{f_S\}_{S\in\mathcal{S}}$ is a feasible solution of LP (6)-(7). Since $\sum_{S\in\mathcal{S}} \sum_{V\in\mathcal{V}} a_{S,V} = 1$, we obtain $\sum_{S\in\mathcal{S}} f_S = 1$. Hence $f^* \le 1$.

For the other case, we show that $f^* \le 1$ implies $\boldsymbol{\nu} \in \mathcal{C}$.

Let $g_S := f_S^*/f^*$ for each $S \in \mathcal{S}$. Note that $\sum_{S\in\mathcal{S}} g_S = 1$ and $\sum_{S\in\mathcal{S}_n} g_S \ge \tau_n$ for each $n \in [0 : N-1]$ from LP (6)-(7).

For $n \in [0 : N-1]$ such that $|U_n| > 0$, we define

$$b^{(u)} := \frac{\frac{\nu_u}{R_u}}{\sum_{u\in U_n} \frac{\nu_u}{R_u}} = \frac{\nu_u}{R_u \tau_n} \tag{35}$$

for each $u \in U_n$.

Since $\mathcal{V} = \{[u_n]_{n=0}^{N-1} : u_n \in U_n\}$ by definition, we obtain

$$\prod_{n=0}^{N-1} \underbrace{\left(\sum_{u\in U_n} b^{(u)} \right)}_{=1 \text{ from (35)}} = \sum_{V\in\mathcal{V}} \prod_{u\in V} b^{(u)} \tag{36}$$

For each $S \in \mathcal{S}$, $V \in \mathcal{V}$, we define

$$b_{S,V} := g_S \prod_{u\in V} b^{(u)} \tag{37}$$

$$\boldsymbol{\nu}' := \sum_{S\in\mathcal{S}} \sum_{V\in\mathcal{V}} b_{S,V} \boldsymbol{r}_{S,V} \tag{38}$$

From (37) and (36), $\sum_{V\in\mathcal{V}} b_{S,V} = g_S$ for each $S \in \mathcal{S}$. Hence, $\sum_{S\in\mathcal{S}} \sum_{V\in\mathcal{V}} b_{S,V} = \sum_{S\in\mathcal{S}} g_S = 1$, which implies $\boldsymbol{\nu}' \in \mathcal{C}$. We will now show that $\boldsymbol{\nu}'$ is element-wise greater than or equal to $\boldsymbol{\nu}$ which concludes the proof.

Consider an arbitrary UE $u \in U$, which lies in segment $n \in [0 : N-1]$, *i.e.* $u \in U_n$. We have $\nu'_u = \sum_{S\in\mathcal{S}} \sum_{V\in\mathcal{V}} b_{S,V} R_u \mathbb{I}_{n\in S} \mathbb{I}_{u\in V} = \sum_{S\in\mathcal{S}_n} \sum_{V\in\mathcal{V}_u} R_u b_{S,V}$. Hence from (37),

$$\nu'_u = \sum_{S\in\mathcal{S}_n} g_S R_u \sum_{V\in\mathcal{V}_u} \prod_{u'\in V} b^{(u')} \tag{39}$$

$$= \underbrace{\left(\sum_{S\in\mathcal{S}_n} g_S \right)}_{\ge \tau_n} \underbrace{b^{(u)} R_u}_{\nu_u/\tau_n} \ge \nu_u \tag{40}$$

where (40) follows since $\mathcal{V}_u = \{[u_m]_{m=0}^{N-1} : u_m \in U_m \forall\ m \ne n, \&\ u_n = u\}$ which implies $\sum_{V\in\mathcal{V}_u} \prod_{u'\in V} b^{(u')} = b^{(u)} \prod_{m=0,m\ne n}^{N-1} \left(\sum_{u'\in U_m} b^{(u')} \right) = b^{(u)}$ ($\because \sum_{u'\in U_m} b^{(u')} = 1$ from (35)). □

Proof of Lemma 1. For 1), note that by summing the constraints of LP (6)-(7), we obtain

$$\sum_{n=0}^{N-1} \sum_{S:n\in S} f_S \ge \sum_{n=0}^{N-1} \tau_n \tag{41}$$

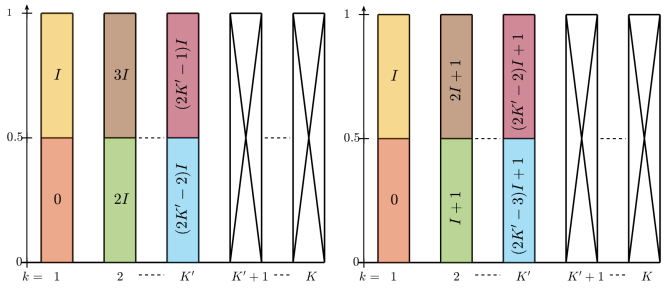
Since $|S| \le K$ for each $S \in \mathcal{S}$, each S appears at most K times in the LHS of the above sum. Hence, we obtain $K \sum_{S\in\mathcal{S}} f_S \ge \sum_{n=0}^{N-1} \tau_n$, which implies $f^* \ge \tau_{avg}$.

Let $\mathcal{S}_m \subseteq \mathcal{S}$ denote the set of all feasible sets which contain a segment $m \in [0 : N-1]$. By definition of a feasible set, each S contains at most one element from the set $[n : n+I]$. Hence, the sets $\{\mathcal{S}_m\}_{m\in[n:n+I]}$ are mutually exclusive. By summing the constraints (7) for $m \in [n : n+I]$, we obtain $\sum_{m\in[n:n+I]} \sum_{S\in\mathcal{S}_m} f_S \ge \tau_{[n:n+I]}$. From mutual exclusivity of $\mathcal{S}_m$'s, we obtain $\sum_{S\in\mathcal{S}} f_S \ge \tau_{[n:n+I]}$. Since the choice of n is arbitrary, $\sum_{S\in\mathcal{S}} f_S \ge \tilde{\tau}_{\max}$. Hence, $f^* \ge \tilde{\tau}_{\max}$. □

Appendix B: Proof of Theorem 4

Proof. If $I + 1 < N \le (2K-1)I$ then $N = k'I + \ell$ for some $k' \in [1 : 2(K-1)]$, $\ell \in [1 : I]$. Depending on whether k' is even or odd, one of the following two cases holds: 1) $N = 2K'I + \ell$ (*i.e.* $k' = 2K'$) for some $K' \in [1 : K-1]$, $\ell \in [1 : I]$ or 2) $N = (2K'-1)I + \ell$ (*i.e.* $k' = 2K'-1$) for some $K' \in [1 : K-1]$, $\ell \in [1 : I]$.

Consider case 1) $N = 2K'I + \ell$ (for $K' \ge 1$). we define the loads for the segments i in $[0 : N-1]$ as follows. $\tau_i = 0.5$ if i is a multiple of I and $\tau_i = 0$ otherwise. Note that $\tilde{\tau}_{\max} = 1$ and $\tau_{avg} = (K' + 0.5)/K < 1$ (since $K' \le K-1$), by construction. We now show that $f^* > 1$.



(a) First case (b) Second case

Fig. 7. Illustration of allocation for constructed examples

By way of contradiction, suppose that $f^* = 1$. Consider the allocation under an optimal solution of LP (6)-(7). Without loss of generality, the time allocations of segments $0, I$ under the solution can be arranged on column $k = 1$ as shown in Fig. 7(a). The ordering in this column 1 fixes the allocation of the rest of the segments, as we will show. For $K' \ge 2$, segment $2I$ must occupy the interval $[0, 0.5)$ since $\tau_{2I} = 0.5$, and segment I occupies the interval $[0.5, 1)$. It follows that the next segment, $3I$, must occupy $[0.5, 1)$. Without loss of generality,

place segments $2I, 3I$ in column 2, as shown in Fig. 7(a). For $K' \geq 3$ this inductive allocation pattern for segments $(2k-2)I, (2k-1)I$ continues for $k = 3, \ldots, K'$, where $(2k-2)I$ occupies interval $[0, 0.5)$ and $(2k-1)I$ occupies interval $[0.5, 1)$ in column k, as illustrated in Fig. 7(a).

For the final segment $N - \ell = 2K'I$, note that interval $[0.5, 1)$ is not available because $(2K'-1)I$ occupies this interval in column K' (true when $K' = 1$ as well as when $K' \geq 2$). Also interval $[0, 0.5)$ is not available for segment $N-\ell$ since this interval is occupied by segment 0 and $N-\ell$ interferes with segment 0. Hence, no interval is available for $N-\ell$ whereas $\tau_{N-\ell} = \tau_{2K'I} = 0.5$. Hence, its not possible to do all the allocations in the interval $[0, 1)$, which is a contradiction. Therefore, f^* must be larger than 1.

For the other case, 2) $N = (2K'-1)I + \ell$, we first deal with the trivial case $K' = 1$. Let $\tau_i = 1$ for each $i \in [0 : N-1]$. Note that $I + 1 < N \leq 2I$, hence every segment i with every other segment in $[0 : N-1] - \{i\}$. Hence, $f^* \geq \tau_{[0:N-1]} = N$, whereas $\tau_{avg} = \tau_{[0:N-1]}/K \leq N/2$ and $\tilde{\tau}_{\max} = \tau_{[0:I]} = I$. Hence, $f^* > \max\{\tau_{avg}, \tilde{\tau}_{\max}\}$.

For the general case, $K' \geq 2$, (hence $N - \ell \geq 3I$) we define the loads for $i \in [0 : N-\ell]$ as follows. $\tau_0 = 0.5$, $\tau_I = 0.5$, $\tau_i = 0.5$ if $i-1$ is an multiple of I, $\tau_{N-\ell} = 0.5$ and 0 otherwise. As with the previous case, $\tilde{\tau}_{\max} = 1$ and $\tau_{avg} = (K' + 0.5)/K < 1$ (since $K' \leq K-1$), by construction. We now show that $f^* > 1$. Proceeding similarly as before, the allocations for segments $0, I$ are arranged in column 1 (as shown in Fig. 7(b)) which fixes the inductive allocation pattern for segments $(2k-3)I + 1, (2k-2)I + 1$ in columns $k = 2, \ldots, K'$ (as shown in Fig. 7(b)).

For the final segment $N - \ell = (2K'-1)I$ (which must be greater than $(2K'-2)I + 1$ since $I \geq 2$), note that interval $[0.5, 1)$ is not available because this interval is occupied by $(2K'-2)I + 1$ in column K' (which interferes with $N - \ell = (2K'-1)I$). Also, interval $[0, 0.5)$ is not available for $N - \ell$ since this interval is occupied by segment 0 in column 1 (which also interferes with $N - \ell$). Hence, no time interval in $(0, 1]$ is available for $N - \ell$, whereas $\tau_{N-\ell} = 0.5$. Hence, f^* must be larger than 1. □

Remark: In our beamforming problem, the columns refer to RF chains. The constructed examples in the proof of the Theorem 4 provide a specific allocation of segments to RF chains. Other allocations are also possible but do not change our conclusions. For the example illustrated in Fig. 7(a), we have only used the first K' RF chains, and then found that there was not enough time for the final segment, even with RF chains $K'+1, \ldots, K$ completely unused. In fact, the allocation of segments $0, 1, \ldots (2K'-1)I$ to RF chains could have used more than K' RF chains, by moving the allocations horizontally across RF chains, with no effect on the scheduled feasible sets. As such, this would not solve the problem of not having enough time for the final segment.

Appendix C: Intermediate results on Upper position bounds

Lemma 7. *Assume $N \geq 2I+2$. For an $i \in [I+1 : N-I-1]$, let $l \in [0 : \min\{I-1, N-I-1-i\}]$ be the index l such that $\boldsymbol{u}_i = \boldsymbol{u}_{i+l-I} \oplus \big(T - \tau_{[i:i+l]}\big)$, then*

$$\boldsymbol{u}_{i+l'} = \boldsymbol{u}_{i+l-I} \oplus \big(T - \tau_{[i+l':i+l]}\big) \tag{42}$$

for each $l' = 0, \ldots, l$.

Proof. We use proof by contradiction. Suppose that (42) holds for $l' = 0, \ldots, k-1$, but not for $l' = k \leq l$, *i.e.* k is the smallest $l' \in [0 : l]$ for which (42) does not hold. Clearly $k \geq 1$ since it is given that

$$\boldsymbol{u}_i = \boldsymbol{u}_{i+l-I} \oplus \big(T - \tau_{[i:i+l]}\big) \tag{43}$$

By definition of $\boldsymbol{u}_{i+k}$ in (19), $\boldsymbol{u}_{i+k} \leq \boldsymbol{u}_{i+k+m'-I} \oplus \big(T - \tau_{[i+k:i+k+m']}\big)$ for all $m' \in [0 : \min\{I-1, N-I-1-i-k\}]$. Taking $m' = l-k$, we obtain that

$$\boldsymbol{u}_{i+k} \leq \boldsymbol{u}_{i+l-I} \oplus \big(T - \tau_{[i+k:i+l]}\big) \tag{44}$$

By the supposition above, the inequality must be strict, hence

$$\boldsymbol{u}_{i+k} < \boldsymbol{u}_{i+l-I} \oplus \big(T - \tau_{[i+k:i+l]}\big) \tag{45}$$

Also by definition of $\boldsymbol{u}_{i+k}$ in (19), there must exist a $m \in [0 : \min\{I-1, N-I-1-i-k\}]$ such that $\boldsymbol{u}_{i+k} = \boldsymbol{u}_{i+k+m-I} \oplus \big(T - \tau_{[i+k:i+k+m]}\big)$. Let $\alpha := k + m - I$. Hence,

$$\boldsymbol{u}_{i+k} = \boldsymbol{u}_{i+\alpha} \oplus \big(T - \tau_{[i+k:i+\alpha+I]}\big) \tag{46}$$

Note also that

$$\alpha + I \leq N - I - 1 - i, \tag{47}$$

since $I + \alpha = k + m$, and $m \leq N - I - 1 - i - k$.

Case 1: $I - k \leq m \leq I - 1$

Since $\alpha = k + m - I$, this case is equivalent to $0 \leq \alpha \leq k-1$. From the supposition that (42) holds for each $l' \in [0 : k-1]$, it must hold for α. Hence, we obtain $\boldsymbol{u}_{i+\alpha} = \boldsymbol{u}_{i+l-I} \oplus \big(T - \tau_{[i+\alpha:i+l]}\big)$ Substituting $\boldsymbol{u}_{i+\alpha}$ in (46), we obtain (48) by re-arranging terms in the sum $\tau_{[i+k:i+\alpha+I]} + \tau_{[i+\alpha:i+l]}$ as $\tau_{[i+\alpha:i+\alpha+I]} + \tau_{[i+k:i+l]}$, which is possible since $i + \alpha < i + k \leq i + l \leq i + \alpha + I$. Thus,

$$\boldsymbol{u}_{i+k} = \boldsymbol{u}_{i+l-I} \oplus \big(T - \tau_{[i+\alpha:i+\alpha+I]}\big) \oplus \big(T - \tau_{[i+k:i+l]}\big). \tag{48}$$

Then, using $T \geq \tilde{\tau}_{\max} \geq \tau_{[i+\alpha:i+\alpha+I]}$, we obtain

$$\boldsymbol{u}_{i+k} \geq \boldsymbol{u}_{i+l-I} \oplus \big(T - \tau_{[i+k:i+l]}\big) \tag{49}$$

which contradicts (45).

Case 2: $0 \leq m \leq I - k - 1$

Since $\alpha = k + m - I$, this case is equivalent to $k \leq \alpha + I \leq I - 1$. From (47), $\alpha + I \leq \min\{I-1, N-I-1-i\}$. From the definition of $\boldsymbol{u}_i$ in (19), we have that $\boldsymbol{u}_i \leq \boldsymbol{u}_{i+m'-I} \oplus \big(T - \tau_{[i:i+m']}\big)$ for all $m' \in [0 : \min\{I-1, N-I-1-i\}]$. Take $m' = \alpha + I$, we have

$$\boldsymbol{u}_i \leq \boldsymbol{u}_{i+\alpha} \oplus \big(T - \tau_{[i:i+\alpha+I]}\big).$$

But $\boldsymbol{u}_{i+\alpha} = \boldsymbol{u}_{i+k} \ominus \big(T - \tau_{[i+k:i+\alpha+I]}\big)$ from (46). Thus,

$$\begin{aligned} \boldsymbol{u}_i &\leq \boldsymbol{u}_{i+k} \ominus \tau_{[i:i+k-1]} \\ &< \boldsymbol{u}_{i+l-I} \oplus \big(T - \tau_{[i:i+l]}\big) \end{aligned} \tag{50}$$

where (50) follows from (45), but contradicts (43). Hence, the lemma must hold. □

Lemma 8. *Assume $N \geq 2I+2$. Then*

$$|\boldsymbol{u}_{i+1}| \geq |\boldsymbol{u}_i| + \tau_i \tag{51}$$

for $i = 0, \ldots, N-I-2$.

Proof. The proof is trivial for $i = 0, \ldots, I-1$ since $|\boldsymbol{u}_{i+1}| = \tau_{[0:i]}$. For $i = I$, $\exists l \in [0 : \min\{I-1, N-2I-2\}]$ such that $|\boldsymbol{u}_{I+1}| = |\boldsymbol{u}_{l+1}| + T - \tau_{[I+1:I+1+l]}$. Hence,

$$|\boldsymbol{u}_{I+1}| = \tau_{[0:I]} + T - \tau_{[l+1:I+l+1]} \tag{52}$$
$$\geq T. \tag{53}$$

where (53) holds since $\tau_{[0:I]} = \tilde{\tau}_{\max}$. Since $|\boldsymbol{u}_I| + \tau_I = \tilde{\tau}_{\max} \leq T$, the lemma holds for $i = I$.

For $i > I$, from definition of $\boldsymbol{u}_i$ in (19),

$$|\boldsymbol{u}_i| = |\boldsymbol{u}_{i+l-I}| + T - \tau_{[i:i+l]} \tag{54}$$

for some $l \in [0 : \min\{I-1, N-I-1-i\}]$.

Case 1: $l \geq 1$

From Lemma 7, we obtain $|\boldsymbol{u}_{i+1}| = |\boldsymbol{u}_{i+l-I}| + T - \tau_{[i+1:i+l]}$. Subtracting from (54), we obtain $|\boldsymbol{u}_{i+1}| - |\boldsymbol{u}_i| = \tau_i$, and the lemma holds with equality in this case.

Case 2: $l = 0$

Note that by definition of $\boldsymbol{u}_{i+1}$ in (19) for some $k \in [0 : \min\{I-1, N-I-2-i\}]$

$$|\boldsymbol{u}_{i+1}| = |\boldsymbol{u}_{i+1+k-I}| + T - \tau_{[i+1:i+1+k]}. \tag{55}$$

Suppose that $k = I-1$, then $|\boldsymbol{u}_{i+1}| = |\boldsymbol{u}_i| + T - \tau_{[i+1:i+I]}$. Hence $|\boldsymbol{u}_{i+1}| = |\boldsymbol{u}_i| + \tau_i + T - \tau_{[i:i+I]}$. Since $T \geq \tilde{\tau}_{\max} \geq \tau_{[i:i+I]}$, we obtain $|\boldsymbol{u}_{i+1}| \geq |\boldsymbol{u}_i| + \tau_i$ and the lemma holds for this case.

For the other case $0 \leq k < I-1$, note that $k+1 \in [1 : \min\{I-1, N-I-1-i\}]$. Hence, by definition of $\boldsymbol{u}_i$, we obtain (56). Then, (57) follows by substituting $|u_{i+k+1-I}|$ from (55)

$$|\boldsymbol{u}_i| \leq |\boldsymbol{u}_{i+k+1-I}| + T - \tau_{[i:i+k+1]} \tag{56}$$
$$= |\boldsymbol{u}_{i+1}| - \tau_i. \tag{57}$$

Hence, the lemma also holds for this case. □

Lemma 9. *Suppose that $N \geq 3I+1$. Then $\boldsymbol{u}_i \oplus \tau_i \geq \boldsymbol{u}_{i-2I+1} \oplus T$ for each $i \in [2I : N-I-1]$.*

Proof. By definition of $\boldsymbol{u}_j$ in (19) for $j \in [I+1 : N-2I]$, $\boldsymbol{u}_j = \boldsymbol{u}_{j+l-I} \oplus \big(T - \tau_{[j:j+l]}\big)$ for some $l \in [0 : \min\{I-1, N-I-1-j\}]$. Applying Lemma 7 for $l' = l$, we obtain $\boldsymbol{u}_{j+l} \oplus \tau_{j+l} = \boldsymbol{u}_{j+l-I} \oplus T$.

By applying Lemma 8 recursively from $j+l-1$ to $j+I$, we obtain $(\boldsymbol{u}_{j+I} \geq)\ \boldsymbol{u}_{j+I-1} \oplus \tau_{j+I-1} \geq \boldsymbol{u}_{j+l} \oplus \tau_{j+l}$, which equals $\boldsymbol{u}_{j+l-I} \oplus T$ from above. Hence,

$$\boldsymbol{u}_{j+I-1} \oplus \tau_{j+I-1} \geq \boldsymbol{u}_{j+l-I} \oplus T. \tag{58}$$

From Lemma 8, $\boldsymbol{u}_k$'s are an non-decreasing sequence of positions. Hence, $u_{j+l-I} \geq u_{j-I}$. Substituting in (58), we obtain $\boldsymbol{u}_{j+I-1} \oplus \tau_{j+I-1} \geq \boldsymbol{u}_{j-I} \oplus T$. The lemma follows by taking $i := j+I-1$. □

Lemma 10. *Suppose that $N-I-1 \geq 2(K-1)I$. Then for each $k = 1, \ldots, K-1$,*

$$\boldsymbol{u}_i \geq (k;0) \oplus \tau_{[0:i-2(k-1)I-1]}, \forall\ i \in [2(k-1)I+1 : (2k-1)I] \tag{59}$$

$$\boldsymbol{u}_i \geq (k+1;0),\ \forall\ i \in [(2k-1)I+1 : 2kI] \tag{60}$$

Proof. We use proof by induction. For the initial step $k = 1$, (59) follows from the definition $\{\boldsymbol{u}_i\}_{i=0}^{I}$ in (17).

We now show that $\boldsymbol{u}_{I+1} \geq (2;0)$ and obtain (60) for $k = 1$ from the fact that $\boldsymbol{u}_j$'s are a non-decreasing sequence of positions due to Lemma 8. By definition $\boldsymbol{u}_{I+1} = \boldsymbol{u}_{l+1} \oplus (T - \tau_{[I+1:I+l+1]})$ for some $l \in [0 : I-1]$. Note that $\boldsymbol{u}_{l+1} = (1; \tau_{[0:l]})$. Hence, $\boldsymbol{u}_{I+1} = (2;0) \oplus (\tau_{[0:l]} - \tau_{[I+l:I+l+1]}) = (2;0) \oplus (\tau_{[0:I]} - \tau_{[l+1:I+l+1]})$, which $\geq (2;0)$ since $\tau_{[0:I]} = \tilde{\tau}_{\max} \geq \tau_{[l+1:I+l+1]}$.

For the inductive step, suppose that (59)-(60) hold for all $k \leq m$. We will now show that (59)-(60) hold for $k = m+1$, which completes induction.

Firstly, note that from Lemma 9, for $i \in [2mI : (2m+1)I-1]$, $\boldsymbol{u}_i \oplus \tau_i \geq \boldsymbol{u}_{i-2I+1} \oplus T$. Since $i-2I+1 \in [2(m-1)I+1 : (2m-1)I]$, from the inductive hypothesis (59) for $k = m$; $\boldsymbol{u}_{i-2I+1} \geq (m;0) \oplus \tau_{[0:i-2mI]}$. Hence,

$$\boldsymbol{u}_i \oplus \tau_i \geq (m+1;0) \oplus \tau_{[0:i-2mI]} \tag{61}$$

for each $i \in [2mI : (2m+1)I-1]$. We use this result to show (59) for $k = m+1$. Note that $\boldsymbol{u}_i \geq \boldsymbol{u}_{i-1} \oplus \tau_{i-1}$ from Lemma 8, for each $i \in [2mI+1 : (2m+1)I]$. Further, since $i-1 \in [2mI : (2m+1)I-1]$, from (61), we obtain

$$\boldsymbol{u}_i \geq (m+1;0) \oplus \tau_{[0:i-2mI-1]} \tag{62}$$

To show (60), note that from Lemma 9, for $i \in [(2m+1)I : 2(m+1)I-1]$, $\boldsymbol{u}_i \oplus \tau_i \geq \boldsymbol{u}_{i-2I+1} \oplus T$. Since $i-2I+1 \in [(2m-1)I+1 : 2mI]$, from the inductive hypothesis (60) for $k = m$; $\boldsymbol{u}_{i-2I+1} \geq (m+1;0)$. Hence, we obtain

$$\boldsymbol{u}_i \oplus \tau_i \geq (m+2;0) \tag{63}$$

for each $i \in [(2m+1)I : 2(m+1)I-1]$. We use this to show (60) for $k = m+1$. Note that $\boldsymbol{u}_i \geq \boldsymbol{u}_{i-1} \oplus \tau_{i-1}$ from Lemma 8, for each $i \in [(2m+1)I+1 : 2(m+1)I]$. Further, since $i-1 \in [(2m+1)I : 2(m+1)I-1]$, from (63), we obtain

$$\boldsymbol{u}_i \geq (m+2;0)\,.$$

□

REFERENCES

[1] O. E. Ayach, R. W. Heath, S. Abu-Surra, S. Rajagopal, and Z. Pi, "The capacity optimality of beam steering in large millimeter wave MIMO systems," in *2012 IEEE 13th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2012, pp. 100–104.

[2] N. Jindal, "MIMO broadcast channels with finite-rate feedback," *IEEE Trans. Inform. Theory*, vol. 52, no. 11, pp. 5045–5060, Nov. 2006.

[3] T. Yoo, N. Jindal, and A. Goldsmith, "Multi-antenna downlink channels with limited feedback and user selection," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 7, pp. 1478–1491, Sep. 2007.

[4] D. J. Ryan, I. B. Collings, I. V. L. Clarkson, and R. W. Heath Jr., "Performance of vector perturbation multiuser MIMO systems with limited feedback," *IEEE Trans. Communications*, vol. 57, no. 9, pp. 2633–2644, Sep. 2009.
[5] Z. Jiang, S. Chen, S. Zhou, and Z. Niu, "Joint user scheduling and beam selection optimization for beam-based massive MIMO downlinks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2190–2204, 2018.
[6] J. Liu and E. Bentley, "Hybrid-beamforming-based millimeter-wave cellular network optimization," in *2017 15th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2017, pp. 1–8.
[7] E. Kim, J. Kwak, and S. Chong, "Exception of dominant interfering beam: Low complex beam scheduling in mmWave networks," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020, pp. 1–6.
[8] Z. Zou, S. Zhao, G. Huang, and D. Tang, "Novel design of user scheduling and analog beam selection in downlink millimeter-wave communications," *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4168–4178, 2022.
[9] Z. Sha, S. Chen, and Z. Wang, "Near interference-free space-time user scheduling for mmWave cellular network," *IEEE Transactions on Wireless Communications*, vol. 21, no. 8, pp. 6372–6386, 2022.
[10] Z. Sha, Z. Wang, S. Chen, and L. Hanzo, "Graph theory based beam scheduling for inter-cell interference avoidance in MmWave cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 3929–3942, 2020.
[11] M. Saad and S. Abdallah, "On millimeter wave 5G backhaul link scheduling," *IEEE Access*, vol. 7, pp. 76 448–76 457, 2019.
[12] P. Paul, H. Wu, and C. Xin, "BOOST: A user association and scheduling framework for beamforming mmwave networks," *IEEE Transactions on Mobile Computing*, vol. 20, no. 10, pp. 2924–2935, 2021.
[13] P. Ni, Z. Wang, H. Li, M. Li, and Q. Liu, "Joint user scheduling and hybrid beamforming design for cooperative mmwave networks," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, 2021, pp. 1–6.
[14] B. S. Baker and E. G. Coffman, "Mutual exclusion scheduling," *Theoretical Computer Science*, vol. 162, no. 2, pp. 225–243, 1996.
[15] F. Gardi, "Mutual exclusion scheduling with interval graphs or related classes, part i," *Discrete Applied Mathematics*, vol. 157, no. 1, pp. 19–35, 2009.

Swaroop Gopalam (Member, IEEE) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology (IIT) Bombay in 2014. He received the M.Res. and Ph.D. degrees in engineering from Macquarie University, Sydney, Australia, in 2017 and 2021 respectively.

He has been working as a Research Fellow in the School of Engineering at Macquarie University since 2021. His research interests include delay-Doppler domain communication, resource allocation in wireless networks, and design of distributed and low-complexity algorithms.

Iain B. Collings (Fellow, IEEE) is a Professor in the School of Engineering, Macquarie University, Sydney, Australia. He received a PhD in systems engineering from the Australian National University 1995 and has published over 300 papers in the area of wireless communications. He has served as an Editor for IEEE Transactions on Wireless Communications, was awarded the Engineers Australia IREE Neville Thiele Award 2009, and the IEEE CommSoc Stephen O. Rice Award 2011. He is a Fellow of the IEEE. Previously he spent nine years at the CSIRO, where he held a number of roles including Deputy Chief of Division, Research Program Leader, and Theme Leader, and nine years at the Universities of Melbourne and Sydney.

Stephen V. Hanly (Fellow, IEEE) is a Professor in the School of Engineering, Macquarie University, Sydney, Australia. He received a Ph.D. in mathematics from Cambridge University, U.K., in 1994. His research contributions are in wireless communications. He is a recipient of the INFOCOM Best Paper Award, the IEEE Information Theory Society and the IEEE Communication Society Joint Paper Award, and the IEEE Communications Society Tutorial Paper Award. He has been an Associate Editor of the IEEE Transactions on Wireless Communications and a Guest Editor of the IEEE Journal on Selected Areas in Communications. He has taken major roles at several IEEE conferences and workshops, including IEEE ISIT, and IEEE CTW.

Hazer Inaltekin (Member, IEEE) is a Senior Lecturer at Macquarie University. He received his B.S. degree (High Honors) in electrical and electronics engineering from Bogazici University, Istanbul, Turkey, and his M.S./Ph.D. degree in electrical and computer engineering from Cornell University, Ithaca, NY. Prior to joining Macquarie University, he held various researcher and faculty positions in Australia, Europe and United States. His research interests include airborne networks, satellite communications, fog/edge computing, IoT, wireless communications, and information theory.