Number-Theoretic Sequence Design for Uncoordinated Autonomous Multiple Access in Relay-Assisted Machine-Type Communications

Yaser M. M. Fouad, Ramy H. Gohary, *Senior Member, IEEE*, and Halim Yanikomeroglu, *Fellow, IEEE*

***Abstract*—Terminal relaying is expected to offer an effective means for realizing machine-type communications (MTC) in wireless cellular networks. In the absence of channel quality indicators, the effective utilization of relaying terminals (RTs) requires a mechanism by which RTs can autonomously assign available resource blocks (RBs) to potentially large numbers of uncoordinated MTC devices with minimal conflicts. Unlike random RB assignments, which do not offer performance guarantees, using prescribed RB assignment sequences provides an opportunity for obtaining performance gains. However, realizing these gains requires optimizing RB assignments over a large set of lengthy sequences. One technique for selecting assignment sequences is based on an exhaustive search of exponential complexity over sequences generated by multiplicative cyclic groups. This technique restricts the number of RBs to be prime minus one and does not consider sequences generated using other group operations. In this paper, we use group isomorphism to eliminate the constraint on the number of RBs and to show that the optimal assignment sequences generated by a specific cyclic group are globally optimal over the set of all cyclically generated sequences. We develop a greedy algorithm with polynomial complexity for the sequential selection of RB assignment sequences in systems with large numbers of RTs and arbitrary device distributions. This algorithm is further simplified by invoking the graphical representation of cyclic groups. The resulting algorithm is more efficient and thus suitable for generating assignment sequences for relay-assisted massive multiple access Internet-of-Things systems. Numerical results show that the performance of the sequences generated by the greedy algorithms is comparable to that of those generated by exhaustive search, but with much less computational cost.**

***Index Terms*—Cooperative D2D Relaying, distributed resource allocation, machine-type communications, sequence-based resource allocation, terminal relaying.**

I. Introduction

MACHINE-TYPE communications (MTC) are prospected to be the engine that will underlie a wide range of Internet-of-Things-based (IoT-based) applications, including smart grids and intelligent transportation systems [1], [2]. Such applications will involve communication between large numbers of low-power geographically dispersed MTC devices. The communication between these devices can be facilitated by exploiting the currently available cellular infrastructure. However, this structure has two major drawbacks: first, the transmission power of the MTC devices may not be sufficient to communicate directly with the base station (BS); and second, the prospective large number of MTC devices communicating simultaneously with the BS may result in heavy interference [3], [4]. These drawbacks can be mitigated by utilizing a terminal relaying scheme, wherein a wireless user terminal assists the communication of its neighboring MTC devices by relaying their signals to the BS.

Using terminal relaying, the MTC devices in each cell are grouped into clusters, where each cluster is served with a relaying terminal (RT) that acts as the cluster head [5], [6]. Each of these RTs aggregates data from the MTC devices within its cluster and relays it to the BS. This relaying-based approach has been extensively studied by the 3rd Generation Partnership Project 3GPP for the Long Term Evolution (LTE) system. In particular, this approach was first initiated in 3GPP for Release 12 of LTE under proximity-based services [7]. In this release it was only limited to providing coverage extensions to terminals with unfavorable channel conditions. This approach was further extended in Release 14 to provide support for vehicle to infrastructure applications [8]. In these applications one vehicle will act as a cluster head and accordingly relay the signals of its neighboring vehicles to the BS. Another 3GPP work stream; i.e., the 3GPP 5G New Radio (NR), envisions that wireless terminals can act as relays and accordingly collect data from their neighboring MTC devices over an operator-controlled sidleink in the licensed cellular spectrum. Since each MTC device communicates only with its neighboring RT, terminal relaying offers the following favorable features:

1) The MTC devices can utilize low power communications and low complexity transmitters, thereby reducing their cost and power requirements.
2) The data collected by the wireless terminals can be locally processed thus reducing the required resources on the expensive long range link; i.e., the link between the wireless terminal and the BS.

Manuscript received August 23, 2016; revised February 25, 2017; accepted April 28, 2017. Date of publication May 17, 2017; date of current version October 13, 2017. This work is supported in part by Huawei Canada Co., Ltd., and in part by the Ontario Ministry of Economic Development and Innovation's ORF-RE (Ontario Research Fund - Research Excellence) program. The review of this paper was coordinated by Prof. C. Assi. (*Corresponding author: Yaser M. M. Fouad*).

The authors are with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S5B6, Canada (e-mail: yfouad@sce.carleton.ca; gohary@sce.carleton.ca; halim@sce.carleton.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2017.2705191

3) The massive number of MTC devices attempting simultaneous communication with the BS can be significantly reduced, thereby reducing the interference at the BS and the probability of collision on the random access channel.

Despite its potential advantages, the efficient implementation of terminal relaying in an MTC framework requires a mechanism by which the RTs can allocate the available (time-frequency) resource blocks (RBs) to their assisted devices.

In the presence of global instantaneous channel quality indicators (CQIs), the RBs can be centrally allocated by the BS to both the BS-RT and the RT-MTC links; e.g., [9]–[13]. To avoid the overhead associated with centralized coordination, semi-distributed RB assignment schemes were discussed in [14], [15]. In these schemes, the role of the BS is limited to propagating the estimated interference (price) of each link and the devices independently compete for the available resources using game theoretic approaches. These schemes were further extended in [16] by relaxing the perfect CQIs; i.e., by incorporating channel uncertainties. Alternatively, when only local instantaneous CQIs are available, the RBs can be independently allocated by the RTs to their MTC devices in a distributed manner; e.g., [17]–[19]. The availability of CQIs also enables the implementation of another class of medium access schemes, which does not require explicit RB allocations. One such scheme is the coded random one, wherein the devices would choose their RBs randomly and the RTs use the available CQIs to cancel the effect of interference [20]–[22]. The performance of these schemes is generally good. However, acquiring the CQIs necessary for their operation in practice incurs a significant overhead and can seriously infringe on the resources available for communication. This drawback was partially mitigated in another class of medium access schemes by relaxing the requirement of the instantaneous CQIs. In particular, this class uses a statistical model of channel uncertainties to perform the RB assignments based on imperfect CQIs; e.g., [16], [23]. Although this approach reduces the signalling overhead in fixed relaying systems, the prospected high mobility of both the RTs and the MTC devices as well as their large numbers might result in highly suboptimal RB assignments. Hence, to extract the potential gains of terminal relaying in IoT systems, it is desirable to design a computationally efficient scheme by which the RBs can be allocated blindly, i.e., without CQIs, and autonomously, i.e., without centralized coordination.

In the absence of CQIs, autonomous RB allocation schemes that use random RB assignments are available, e.g., [24]. Other examples of random access schemes include ALOHA and Slotted-ALOHA in which wireless devices transmit their data over randomly selected RBs [25]. These schemes dispense with the overhead necessary for centralized coordination and CQI acquisition, but the lack of coherence in their structure results in, otherwise avoidable, RB allocation conflicts. This drawback of the random schemes is alleviated by the one proposed in [26], wherein the RB assignment sequences are endowed with a multiplicative cyclic group (MCG) structure. In this work, the proposed structure was used to obtain cyclically-generated (CG) RB assignment sequences and an exhaustive search was performed to select the sequences that minimize the number of RB allocation conflicts between the RTs. Despite its advantages, when the number of RTs increases, the computational cost of exhaustive search over CG sequences becomes practically infeasible. As such, the exhaustive search cannot be used for admission control in the IoT paradigm, which is expected to comprise tens of thousands of RBs and potentially hundreds of RTs.

In this paper, we consider a distributed relay-assisted cellular system that supports MTC. In this system, the CQIs are assumed to be neither available at the RTs nor at the BS. Each RT locally assigns RBs to its incoming MTC devices according to a prescribed CG RB assignment sequence; i.e., without centralized coordination by the BS. These cyclic sequences are generated at the RTs by using the properties of cyclic groups with a particular group operation. However, some group operations impose constraints on the number of RBs that would prevent the direct application of cyclic-structured RB assignments in IoT applications; e.g., the multiplicative group operation constrains the number of system RBs to be a prime minus one. To alleviate this constraint, we use a property of cyclic groups, namely, group isomorphism, to show that the performance of the CG assignment sequences is independent of the group operation, thereby waiving the requirement on the number of RBs to be a prime minus one as in [26]. A direct consequence of this observation is that the optimal assignment sequences obtained by searching over the set of sequences generated by a specific cyclic group are globally optimal over the set of all CG sequences of the same number of elements. Furthermore, we provide a computationally efficient technique that, in contrast with the scheme in [26], enables the design of RB assignment sequences for networks with large numbers of RTs and RBs. In particular, we develop a greedy algorithm, whereby the RB assignment sequences are CG in an efficient manner that relies on the structure of cyclic groups and the sequential assessment of the number of assignment conflicts between the sequences. Since the assignment sequences generated by the proposed algorithm are obtained from distinct group generators and cyclic shifts, the quest for designing sequences is reduced to the quest for finding group generators and cyclic shifts that result in minimal assignment conflicts. In each iteration of the proposed algorithm, a pair of group generators is chosen such that the number of assignment conflicts between the sequences that they generate and the ones generated by the previously selected group generators is minimized. The selected group generators are then ranked in a look-up table in an ascending order of their observed number of assignment conflicts. To further simplify the design, we invoke a graphical representation of cyclic groups to gain insight into their structure, and to subsequently develop a graphical greedy algorithm for selecting the group generators and cyclic shifts.

Numerical results suggest that the performance of the sequences generated by the proposed greedy algorithms is comparable to that of the sequences generated by exhaustive search, but with a significantly less computational cost. In particular, the complexity of the proposed greedy algorithms is polynomial, whereas that of the proposed exhaustive search in [26] is exponential. As such, the greedy algorithms enable the application of cyclic-structured RB assignments in IoT systems, which are expected to comprise tens of thousands of RBs and

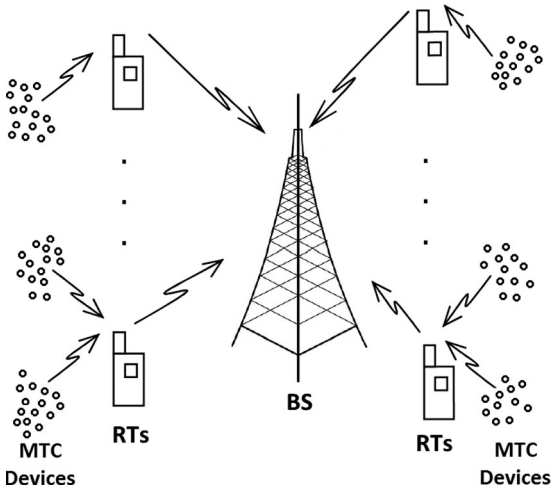

Fig. 1. A cellular MTC system assisted by RTs.

potentially hundreds of RTs. Cyclic RB assignment sequences for systems with that many RTs and RBs are computationally prohibitive when designed by the exhaustive search technique of [26]. Furthermore, the proposed greedy algorithms do not require active RTs to update their RB assignment sequences when new RTs access the system. In particular, for a network with a given number of RBs, the look-up table provided by the algorithms contains all the generators of the corresponding cyclic group. When a new RT enters the system at any given instant, it uses the first unutilized group generator for obtaining its RB assignment sequence. This is in contrast with the exhaustive search of [26], which necessitates that all active RTs update their assignment sequences when new RTs enter the system. To summarize, the contributions of this work are highlighted below. In particular, this paper

1) develops a flexible greedy algorithm that offers a tradeoff between the performance of the CG sequences and the computational complexity of their selection process;
2) invokes graphical representation of cyclic groups to develop an efficient greedy algorithm that enables the application of cyclic-structured RB assignments in IoT systems with hundreds of RTs and thousands of RBs;
3) reduces the signalling overhead in the network by allowing active RTs to maintain their RB assignment sequences when new RTs enter the system; and
4) invokes group isomorphism to show that exhaustive search over sequences CG by a given group operation suffices to guarantee optimality over all CG sequences.

The latter contribution results in: 1) exposing the independence between the performance of the cyclic-structured RB assignment sequences and the group operation underlying their generation, thereby reducing the computational complexity of the sequences selection process; and 2) eliminating the constraint imposed by the MCG structure on number of RBs.

II. Autonomous Assignments Using CG Sequences

We consider an autonomous uncoordinated RB assignment scheme for the uplink of a relay-assisted cellular network that supports MTC as depicted in Fig. 1. In this scheme, wireless terminals act as RTs and assist the communication between their neighboring MTC devices and the BS. An RT is selected by an active MTC device according to the observed signal strength. The RT does not have access to the CQIs and, once selected, it allocates one or multiple RBs to the MTC device, depending on the data rate requirement thereof. RTs relay the data of their neighboring MTC devices to their respective BSs over a different set of RBs. Hence, an RT can always aggregate its own data and communicate it along with the relayed MTC data to the BS. RTs communicate the RB allocations to individual MTC devices over a dedicated control channel. This channel is used by the RT to notify its assisted MTC devices that it is no longer available for relaying. Once notified of the RT updated status, the MTC devices will select one of the remaining RTs depending on the received signal strength. The selected RT will then assign its available RBs to fulfil the data rate requirements of these MTC devices. To avoid excessive signalling between neighboring RTs for coordinating their RB assignments, each RT autonomously assigns resources to its assisted devices.

To maintain autonomy, two methodologies can be considered for RB assignment. The first methodology uses unstructured random RB allocations [24], whereas the second one performs the RB assignments based on prescribed sequences [26]. In both methodologies, the absence of coordination between the RTs might result in instances at which one RB is assigned to multiple MTC devices simultaneously. Occurrences of such an instance, referred to as a 'hit', are likely to result in high interference levels, thereby significantly deteriorating the quality of the signals observed by the MTC devices. Hence, to improve the quality of these signals, it is desirable for the RTs to reduce the number of hit occurrences. Unfortunately, achieving this goal is not guaranteed if the RBs were to be assigned randomly. This is because a random assignment does not possess a known structure that would enable controlling the number of hit occurrences. In contrast, RB assignments that are based on prescribed sequences provide an opportunity for optimizing the number of hits, thereby reducing the interference levels observed by the MTC devices. Our focus in the remainder of this paper is on the latter approach. In particular, we consider a structured RB assignment scheme, whereby each RT performs the RB assignments to its MTC devices according to prescribed assignment sequences with entries corresponding to available RBs. Unfortunately, optimizing the assignments of these RBs over all possible sequences constitutes a formidable task that, for a network with N RBs and M RTs, requires exhaustive search over $\binom{N!}{M}$ combinations of sequences.

When the number of RTs is small, e.g., $M = 4$ RTs, a practical approach is to restrict the exhaustive search to CG sequences [26], that is, sequences generated from a single element in the sequence. This is because such sequences posses the following favorable features:

1) The only available sequences that can be generated from a single generator.
2) Can be easily generated locally by the RTs by repeated applications of a simple group operation, e.g., addition, on a single seed.
3) Offer a relatively small number of sequences compared to the total number of available sequences of the same length $N!$.

4) In contrast with random sequences, the cyclic structure of these sequences can be utilized to further reduce the number of assignment conflicts between neighboring RTs; e.g., by using the hit identification and avoidance algorithm as discussed in Section VI.

Instances of CG sequences are the pseudonoise (PN) and low density sequences (LDS), which are commonly used in Frequency Hopping (FH) systems [27]–[29] and the Zadoff-Chu (ZC) sequences, which are implemented in Long Term Evolution (LTE) systems [30]. The design of these sequences is based on communication applications that differ significantly from the RB assignment one considered herein. For instance, the design of PN sequences that are 'good' for FH systems is based on minimizing the number of collisions for an application in which each wireless device occupies a particular frequency slot during one dwell interval, and relinquishes it after that. In contrast, the design of 'good' ZC sequences is based on communication applications that require certain orthogonality properties, e.g., random access preambles and synchronization in LTE systems [31]. Being generated to meet other design criteria, these sequences do not necessarily perform well when used for assigning RBs in the MTC relay-assisted communication systems considered herein. To see that, we note that in MTC no spreading is needed, in contrast with applications for which ZC sequences are preferable, and that each MTC device is entitled to retain its assigned RBs during the entire transmission interval, in contrast with applications for which PN sequences are preferable. Finally, we note that despite the underlying differences in the design, the PN, ZC, and LDS sequences are generated from a single seed. Hence, by construction, these sequences constitute a proper subset of the cyclically-generated sequences.

We now review preliminaries pertaining to the generation of cyclic RB assignment sequences. We begin by recalling the following definitions.

Definition 1 ([32]): A group $\mathbb{G}$ is a set on which a group operation (denoted by juxtaposition) is defined such that: $\mathbb{G}$ is associative; for all $(x, y) \in \mathbb{G} \times \mathbb{G}$, $xy \in \mathbb{G}$; and the inverse for each element is in $\mathbb{G}$. The order of $\mathbb{G}$ is the number of its elements. ■

Cyclic RB assignment sequences are generated from a particular class of groups known as cyclic groups, which are defined next.

Definition 2 ([32]): A group $\mathbb{G}$ is cyclic if all its elements can be generated by repeated application of the group operation on one element in $\mathbb{G}$. Such an element is called a group generator. ■

For a cyclic group $\mathbb{G}$ of order N, the number of group generators is given by the Euler Totient function, $\phi(N)$.

Definition 3 ([33]): Given a positive integer N, the Euler Totient function, $\phi(N)$, is the cardinality of the set of integers that are coprime with N; that is, $\phi(N)$ is the cardinality of the set $\{q < N | \gcd(q, N) = 1\}$, where $\gcd(q, N)$ is the greatest common divisor of q and N. ■

Immediate from Definition 2 is that repeatedly applying the group operation on one of the generators of a group $\mathbb{G}$ with order N yields an N-entry cyclic sequence that spans all the elements of $\mathbb{G}$. This implies that there are $\phi(N)$ distinct cyclic sequences, each corresponding to a distinct group generator. Hence, when designing CG RB assignment sequences for a network with N RBs, it suffices to consider the order-N cyclic group that generates them. In particular, given a distinct group generator, each RT can generate a cyclic sequence that spans all the RBs available to the network, and the quest for optimal cyclic sequences is distilled to the quest for optimal generators and group operation. In [26], CG sequences were obtained using the MCG structure. These sequences were enriched by considering cyclically shifted versions thereof. In particular, applying a cyclic shift, s, to a sequence, cyclically rotates its entries by s slots, where $s \in \{0, \ldots, N-1\}$. In [26], the group generators and cyclic shifts were chosen by exhaustive search to ensure a minimal number of hit occurrences. Despite its advantages in minimizing hit occurrences, performing an exhaustive search over all CG sequences by the MCG structure incurs a computational cost that prohibits its utilization in systems with practical numbers of RTs; e.g., exhaustive search cannot be implemented in systems with more than 4 RTs. Moreover, restricting RB assignments in [26] to sequences generated by the MCG structure imposes a constraint on the number of RBs and guarantees the optimality of the selected sequences only with respect to MCG structure. These drawbacks will be alleviated by the techniques proposed hereinafter.

III. Global Optimality Over the Set of CG Sequences

The design of optimal RB assignment sequences for systems with large numbers of RBs and RTs involves a computationally infeasible exhaustive search over a significantly large set of sequence combinations. To address this drawback, a possible approach is to restrict this search to the set of sequences that can be easily generated from a single generator; i.e., CG sequences. This reduces the number of candidate RB assignment sequences for a system with N RBs from $N!$ to $\phi(N)$. However, the entries of the $\phi(N)$ cyclic sequences depend on the group operation used for their generation. Based on this observation, one might be tempted to search exhaustively over sequences generated by all possible group operations; restricting oneself to a specific group operation would guarantee optimality over that particular group.

Fortunately, optimizing the sequences over all group operations is not necessary. Indeed, in this section, we show that the set of CG sequences that are optimal with respect to a given group operation are globally optimal over the set of all CG sequences of the same order. This follows from the observations that the performance of an RB assignment sequence depends solely on the sequence structure, but not on its actual entities, and that this structure is independent of the group operation. The latter observation follows from a fundamental property of cyclic-groups known as group isomorphism, which is described next.

A. Application of Group Isomorphism in RB Assignment

For any two cyclic groups of the same order, there exists a structure-preserving function that maps their entities. This function is referred to as group isomorphism, which is formally defined as follows.

Definition 4 (Isopmorphism of Cyclic Groups [32]): Let $\mathbb{G}$ and $\bar{\mathbb{G}}$ be two cyclic groups of the same order. An isomorphism $F : \mathbb{G} \rightarrow \bar{\mathbb{G}}$ is a one-to-one and onto map that respects the group operation, that is, if $x, y \in \mathbb{G}$, then $F(x), F(y) \in \bar{\mathbb{G}}$ and $F(xy) = F(x)F(y)$. ■

Although juxtaposition is used to denote the group operations in both $\mathbb{G}$ and $\bar{\mathbb{G}}$, these operations are generally different.

Definition 4 implies that the isomorphism F can be considered as a relabelling function that maps every element in $\mathbb{G}$ to an element in $\bar{\mathbb{G}}$. In addition, since both groups, $\mathbb{G}$ and $\bar{\mathbb{G}}$, have the same order, N, they also have the same number of group generators, $\phi(N)$, irrespective of their underlying group operations. Using these implications, we will show that, when designing cyclic sequences for RB assignment, the selection of the underlying group operation is immaterial. The key step to show that follows from the following lemma.

Lemma 1 (Isomorphism With Additive Groups [33], [34]): Any cyclic group, $\mathbb{G}_N$, of order N is isomorphic to the additive cyclic group (ACG), $(\mathbb{Z}_N, +) = \{0, 1, 2, \ldots, N-1\}$, irrespective of the group operation that underlies $\mathbb{G}_N$. ■

Although this lemma asserts the existence of an isomorphism between any two cyclic groups of order N, $\mathbb{G}_N$ and $\bar{\mathbb{G}}_N$, it does not provide information on the sequences generated by these groups. In the following theorem, we will show that not only $\mathbb{G}$ and $\bar{\mathbb{G}}$ are isomorphic, but also their CG sequences.

Theorem 1: For any cyclic group $\mathbb{G}_N$ of order N, there exists an isomorphism F such that the set of sequences that are CG by the group operation underlying $\mathbb{G}_N$ is isomorphic to the set of sequences that are CG by the additive group operation underlying $\mathbb{Z}_N$.

Proof: See Appendix. ■

We note that the isomorphism, F, in this theorem is not unique. In other words, there are multiple isomorphisms that map the sequences of $\mathbb{G}_N$ to those of $\mathbb{Z}_N$. What is common between these isomorphisms is that they all map the set of sequences generated by $\mathbb{G}_N$ to that of those generated by $\mathbb{Z}_N$.

Before discussing the implications of Theorem 1 on the proposed technique, we verify it with an illustrative example.

Example 1: Let $\mathbb{G}_{10}$ and $\mathbb{Z}_{10}$ be the multiplicative and additive cyclic groups of order $N = 10$, respectively. The number of CG sequences by both groups is given by $\phi(10) = 4$; cf. Definition 3. For the MCG structure, the set of CG sequences is obtained by

$$\left\{ g_i^{(k+s_i)} \pmod{N+1} \right\}_{k=1}^{N} \tag{1}$$

where g_i is the ith group generator of $\mathbb{G}_N$, s_i is the cyclic shift and $\pmod{N+1}$ is the modular operation that yields the integer remainder of dividing $g_i^{(k+s_i)}$ by $(N+1)$. For ease of exposition, in this example we set all cyclic shifts to 0. Doing so, yields the following sequences for $\mathbb{G}_{10}$:

$(g_1 = 2)$	2	4	8	5	10	9	7	3	6	1
$(g_2 = 6)$	6	3	7	9	10	5	8	4	2	1
$(g_3 = 7)$	7	5	2	3	10	4	6	9	8	1
$(g_4 = 8)$	8	9	6	4	10	3	2	5	7	1

Similarly, the set of CG sequences by the ACG structure can be obtained by

$$\left\{ g_i(k + s_i) \pmod{N} \right\}_{k=1}^{N} \tag{2}$$

where g_i is the ith group generator of $\mathbb{Z}_N$ and s_i is the cyclic shift. Applying (2) yields the following sequences for $\mathbb{Z}_{10}$:

$(g_1 = 1)$	1	2	3	4	5	6	7	8	9	0
$(g_2 = 3)$	3	6	9	2	5	8	1	4	7	0
$(g_3 = 7)$	7	4	1	8	5	2	9	6	3	0
$(g_4 = 9)$	9	8	7	6	5	4	3	2	1	0

Considering the isomorphism mapping function $\{F : \mathbb{G}_{10} \rightarrow \mathbb{Z}_{10} : F((g_1 = 2)^k \pmod{N+1}) = k \pmod{N}\}$, the elements of the two group structures can be mapped as:

MCG	2	4	8	5	10	9	7	3	6	1
ACG	1	2	3	4	5	6	7	8	9	0

Applying this mapping to the set of CG sequences by the MCG structure yields the one generated by the ACG structure. We note that the sequence-to-sequence mapping is not unique. For instance, another mapping function is $\{F : \mathbb{G}_{10} \rightarrow \mathbb{Z}_{10} : F((g_2 = 3)^k \pmod{N+1}) = k \pmod{N}\}$. However, under any isomorphism, the complete set of CG sequences by $\mathbb{Z}_{10}$ will be obtained from the one generated by $\mathbb{G}_{10}$. ■

To illustrate the implications of Theorem 1 on RB assignment, we note that this theorem asserts that all cyclic group operations will provide sequences with the same structure, but with different labels. In other words, the relative locations between any two elements of a CG sequence will be preserved under an isomorphism between any two cyclic groups of the same order. Now, consider the process of RB assignment in a relay-assisted MTC system. A hit will occur between any two RTs when they concurrently assign the same RB irrespective of its label; i.e., when they both use the same element in their respective RB assignment sequences. Hence, only the relative locations of elements in the RB assignment sequences, i.e., the sequence structures, will affect the number of hit occurrences. Since isomorphism preserves the sequence structures across all group operations, performing an exhaustive search over the set of cyclic sequences generated by a group of order N guarantees optimality over the set of all CG sequences of the same order. This implies that, without loss of generality, the CG sequences by the ACG structure can be utilized for RB assignments in any IoT system with an arbitrary number of RBs, N. More specifically, this observation enables us to eliminate the restriction imposed by the MCG structure of [26] on N to be a prime minus one.

IV. A Greedy-Based Approach

The goal of this section is to develop an efficient technique for designing assignment sequences that reduce assignment conflicts in relay-assisted IoT systems. In these systems, the number of RBs and RTs is expected to be large, which renders

RB assignment schemes that rely on exhaustive search rather impractical. To address this difficulty, the assignment scheme developed herein will rely on a greedy algorithm that sequentially selects the RB assignment sequences from the set of CG ones. The performance of this algorithm will be shown to be comparable to that of exhaustive search, but its computational complexity is significantly less.

Consider a group of order N, corresponding to a system with N RBs. For such a system, the greedy algorithm begins by pairing the $\phi(N)$ group generators, so that the two entries of each pair yield reversed cyclic sequences. The algorithm will later associate with each pair a cyclic shift that determines the starting point of each sequence. The optimization of these shifts and their association with the group generator pairs will be clarified below. In the first iteration of the greedy algorithm, a pair is arbitrarily selected, along with a cyclic shift $s = 0$, to be the initial basis of the algorithm. In each of the subsequent iterations, the algorithm augments its basis with one group generator pair and its associated cyclic shift, thereby exhausting all group generator pairs in $\frac{\phi(N)}{2} - 1$ iterations. The group generator pair and the cyclic shift selected in each iteration are tabulated in a look-up table, which is constructed once offline for every value of N supported by the system. For example, a system in which N_1 and N_2 RBs are used in different coverage areas would require two look-up tables. Once constructed for a coverage area with N RBs, the table can be used to generate assignment sequences for any number of RTs less than $\phi(N)$; for M RTs, only the first $M/2$ entries of the look-up table are used to generate the M RB assignment sequences. Note that when an RT becomes unable to assist its neighboring MTC devices, it immediately notifies the BS of its updated status and releases the group generator and cyclic shift pair that it used to generate its RB assignment sequence. The released pair can then be acquired by any new RT that joins the system. However, the procedure through which this pair is acquired depends on the admission protocol between the RTs and the BSs. In particular, for systems in which the RTs notify BSs of their joining, the look-up table is administrated by BSs and accordingly they can assign the released pair to the new RT. In contrast, for completely autonomous systems, the RTs are self-admitted once they are capable of relaying and thus will need to locally select their pair. In such systems, a new RT will sequentially attempt to utilize the pairs in the same order of its local look-up table until it reaches a released one.

In the remainder of this section, we will elaborate on the details of the pairing technique and the selection methodology of the RB assignment sequences.

A. Pairing of Group Generators

Our pairing methodology is based on finding the optimal generators for the two-relay case. In that case, the quest for the optimal CG sequences can be distilled to the quest for two assignment sequences that proceed in reversed orders. In particular, using these sequences for RB assignment results in no hits as long as the number of assigned RBs is less than the number of available ones. To obtain sequences that proceed in reversed orders, we begin by noting that for a group generator g and cyclic shift s, there exists a unique inverse (g^{-1}, s^{-1}), where g^{-1} is the modular arithmetic inverse of g satisfying

$$gg^{-1} \equiv e \pmod{N+1}, \tag{3}$$

where in writing (3) we used e to denote the group identity element. To determine s^{-1}, we note that the sequences generated by (g, s) and (g^{-1}, s^{-1}) should proceed in reversed orders, which implies that

$$s^{-1} = N - s - 1. \tag{4}$$

For example, consider the pair $(g = 2, s = 0)$ of the MCG of order $N = 10$. The inverse of this pair is $(g^{-1} = 6, s^{-1} = 9)$ and from (1), its CG sequence becomes $(1, 6, 3, 7, 9, 10, 5, 8, 4, 2)$, which is the reversed version of the sequence generated by $(g = 2, s = 0)$, cf. Example 1.

B. Sequential Selection of Assignment Sequences

Let $\mathcal{T}$ be the look-up table to be constructed by the algorithm and let $\mathcal{U}$ be the set containing the group generator pairs that have yet to be selected. At first, $\mathcal{T}$ is empty and $\mathcal{U}$ contains the $\frac{\phi(N)}{2}$ group generator pairs. The algorithm arbitrarily selects a group generator pair (g, g^{-1}) from $\mathcal{U}$ and, without loss of generality, sets its cyclic shift s to 0, which, using (4), yields $s^{-1} = N - 1$. The quadruple (g, g^{-1}, s, s^{-1}) is taken to be the initial basis of the algorithm and the pair (g, g^{-1}) is removed from $\mathcal{U}$. In each of the subsequent $\frac{\phi(N)}{2} - 1$ iterations, the algorithm examines the group generator pairs in $\mathcal{U}$ against a particular performance metric for all possible cyclic shifts $s \in \{0, \ldots, N-1\}$. The quadruple that yields the best performance is then used to augment $\mathcal{T}$ and its group generators are removed from $\mathcal{U}$. The algorithm continues to iterate until the exhaustion of all the pairs in $\mathcal{U}$.

When determining an appropriate performance metric for choosing the group generator pairs and cyclic shifts, it is desirable that this metric enables the algorithm to adapt to both uniform and non-uniform distributions of MTC devices. As such, this metric should not depend on the instantaneous loads of the RTs. One such metric is the average number of pairwise hits over all possible system load distributions. Another metric is based on the graphical representation of cyclic groups, which will be discussed in the following section.

We note that considering the performance metric to be the average number of pairwise hits over all possible system load distributions facilitates the sequence-selection process. To see this, let (g, g^{-1}, s, s^{-1}) be the quadruple examined at the Q-th iteration of the greedy algorithm. Now, the sequences generated by (g, s) and (g^{-1}, s^{-1}) proceed in reversed orders and thus these sequences result in identical average number of pairwise hits with the sequences generated by the $Q - 1$ quadruples previously selected, $(g_q, g_q^{-1}, s_q, s_q^{-1}) \in \mathcal{T}$, $q = 1, \ldots, Q-1$. Hence, when evaluating the performance of a quadruple (g, g^{-1}, s, s^{-1}), it suffices to evaluate the metric for the sequence generated by either (g, s) or (g^{-1}, s^{-1}).

We conclude this section by pointing out a key advantage of using the look-up table yielded by the greedy algorithm. In particular, the order of the quadruples in this table depends

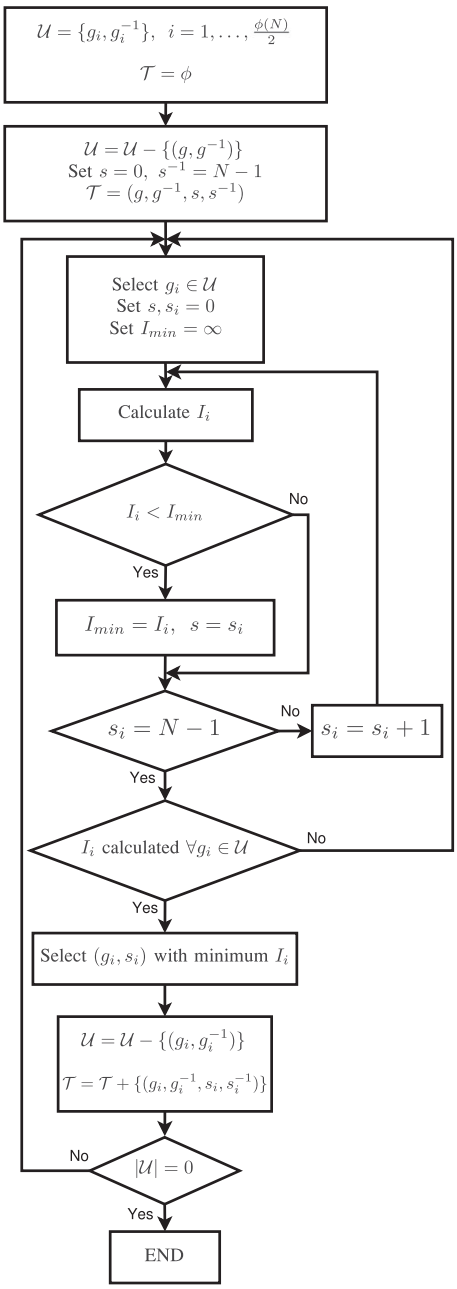

Fig. 2. The greedy algorithm.

solely on the number of pairwise hits, but not on the number of RTs in the system. Thus, it suffices for the system to have only one look-up table to support any number of RTs less than $\phi(N)$. Subsequently, by using the greedy algorithm, active RTs do not need to update their sequences when new ones join the system. In contrast, the exhaustive search scheme in [26], requires having one look-up table for each possible number of RTs. Thus, in the exhaustive search scheme, when a new RT joins the system, the ones already active must switch to the look-up table that corresponds to the new number of RTs.

The greedy algorithm is summarized in the flowchart in Fig. 2. In this flowchart, we will use I_i to denote the performance metric for a candidate group generator cyclic shift pair (g_i, s_i). This metric will be discussed in detail in the following section.

C. Performance Evaluation of Cyclic Assignment Sequences

In the greedy algorithm, the sequence-selection mechanism is based on evaluating the performance of a candidate sequence with respect to the ones previously selected in the look-up table $\mathcal{T}$ only. This table is then augmented by the quadruple corresponding to the sequence yielding the best performance; i.e., in each iteration $|\mathcal{T}|$ increases by 1. In contrast, in the exhaustive search in [26], the sequence-selection mechanism is based on the computationally intensive task of the joint performance evaluation of all CG sequences and the look-up table is constructed all at once. In this section, we adapt the average number of pairwise hits metric in [26] to the selection methodology of the greedy algorithm developed herein.

1) The Performance Metric: Let $X_i \in \{0,1\}^{N\times N}$ be what we refer to as the load matrix of the i-th RT, $i = 1, \ldots, M$. The row and column indices of X_i represent the RB index and the load level of the i-th RT, respectively. The (r_1, r_2)-th entry of X_i represents the binary state of RB r_1 when the load level of the i-th RT, is r_2. This entry is 1 if RB r_1 is assigned by the i-th RT when loaded with r_2 devices, and is 0 otherwise, that is,

$$X_i(r_1, r_2) = \begin{cases} 1 & \text{if} \quad r_1 \in S_i(r_2), \quad \text{and} \\ 0 & \text{otherwise,} \end{cases}$$

where $S_i(r_2) = \left\{g_i^{(q+s_i)} \pmod{N+1}\right\}_{q=1}^{r_2}$ is the assignment sequence of the i-th RT and (g_i, s_i) is its group generator and cyclic shift pair. As an illustrative example, suppose that the pairs (g_i, s_i) and (g_j, s_j) yield the assignment sequences $(1, 3, 2, 4)$ and $(3, 2, 4, 1)$, respectively. The load matrices, X_i and X_j, corresponding to these sequences are

$$X_i = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \text{and} \quad X_j = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

By using these matrices, we can evaluate the performance of two sequences, say S_i and S_j, that are assigned to the i-th and j-th RTs, respectively. To see this, let ℓ_i and ℓ_j denote the loads, and k_i and k_j denote the indices of the last RBs assigned by these RTs. Note that, in general $k_i \geq \ell_i$ because when the i-th RT detects a hit at an assigned RB, it relinquishes this RB and assigns the following one in its RB assignment sequence. Unfortunately, there is no direct relation that maps ℓ_i and ℓ_j to k_i and k_j. However, an iterative method for evaluating k_i and k_j for given RT loads will be provided in the following section. For now, we will assume that the values of k_i and k_j are given. To evaluate the performance of the sequences S_i and S_j for these values, we observe that the number of pairwise hits detected by the RTs is given by the inner product of the k_i-th column of X_i and the k_j-th column of X_j. This product is given by the (k_i, k_j)-th entry of the pairwise hit matrix

$$H_{i,j} = X_i^T X_j. \tag{5}$$

These $\{H_{i,j}\}$ matrices will enable us to evaluate the number of pairwise hits between the sequences generated by a candidate group generator cyclic shift pair, (g_i, s_i), and any of

the $\{(g_j, s_j, g_j^{-1}, s_j^{-1})\}$ quadruples in the look-up table, $\mathcal{T}$. To do so, for each $H_{i,j}$ matrix, we define a complement matrix $\bar{H}_{i,j} \triangleq X_i^T \bar{X}_j$, where $\bar{X}_j$ is the load matrix corresponding to the sequence generated by (g_j^{-1}, s_j^{-1}). In essence, $\bar{H}_{i,j}$ characterizes the number of pairwise hits between the sequence generated by (g_i, s_i) and that generated by (g_j^{-1}, s_j^{-1}). By denoting the index of the last assigned RB in the sequence generated by (g_j^{-1}, s_j^{-1}) as $\bar{k}_j$, the number of pairwise hits between the sequences corresponding to the pairs (g_i, s_i), (g_j, s_j) and (g_j^{-1}, s_j^{-1}) can be expressed as

$$H_{i,j}(k_i, k_j) + \bar{H}_{i,j}(k_i, \bar{k}_j). \tag{6}$$

Averaging over all possible load combinations of a system with N MTC devices, the total number of pairwise hits between the sequence corresponding to (g_i, s_i) and those corresponding to each of the quadruples in $\mathcal{T}$ can be expressed as

$$I_i = \frac{1}{2|\mathcal{T}|} \sum_{j=1, j\neq i}^{|\mathcal{T}|} \sum_{(k_i, k_j, \bar{k}_j)\in\mathcal{Y}} H_{i,j}(k_i, k_j) + \bar{H}_{i,j}(k_i, \bar{k}_j),$$

$$\text{where} \quad i = 1, \ldots, |\mathcal{U}|. \tag{7}$$

where $\mathcal{Y}$ is the set of all the indices $(k_i, k_j, \bar{k}_j)$ such that the sum of their corresponding loads is equal to N.

2) Evaluating the Index of the Last Assigned RB in a Sequence: The use of the $\{H_{i,j}\}$ matrices in the evaluation of the number of pairwise hits between two RTs requires invoking the dynamics of the RB assignment process. In this process, the index of the last assigned RB in the assignment sequences of the two RTs might be higher than their actual number of assigned RBs. This is because when an RT detects a hit on assigning a particular RB, it relinquishes this RB and assigns the following one in its sequence. This process is illustrated by the following example.

Example 2: Let the RB assignment sequences of the i-th and j-th RTs be $(1, 3, 5, 4, 6, 2, 7)$ and $(3, 5, 4, 7, 6, 1, 2)$, respectively. Let the load of the j-th RT be two MTC devices, which are assigned RBs 3 and 5. Now, suppose that the i-th RT has two incoming devices each requesting an RB. Following its sequence, the i-th RT will assign RBs 1 and 3 to the incoming devices, which will result in a hit at RB 3. Subsequently, it will relinquish this RB and assign the following one in its sequence; i.e., RB 5. Since this RB has also been assigned by the j-th RT, the i-th RT again relinquishes this RB and assigns RB 4 to the second device. ■

Careful inspection of this example yields two observations: first, k_i is in general greater than ℓ_i; e.g., for the i-th RT, $k_i = 4$, whereas $\ell_i = 2$; and second, the relinquishment of a particular RB and the assignment of a subsequent one can result in additional hits and subsequently an increase in k_i.

Unfortunately, there is no direct relation that maps loads to sequence indices. However, these indices can be iteratively evaluated using the pairwise hit matrices in (5). In particular, consider the number of pairwise hits in (6) and let k_j and $\bar{k}_j$ be fixed. To determine k_i, we provide a function f_i that, at the μ-th iteration, evaluates a tentative index $k^{(\mu)}$ based on $k^{(\mu-1)}$ with the initial index $k^{(0)}$ being equal to the load of the i-th RT, ℓ_i. In particular, at the μ-th iteration, f_i evaluates the number of hit occurrences between the sequences corresponding to the three pairs (g_i, s_i), (g_j, s_j) and (g_j^{-1}, s_j^{-1}) when $k = k^{(\mu-1)}$. If no hits are detected, then $k_i = k^{(\mu-1)}$; none of the assigned RBs will be relinquished and thus k_i will not increase. On the other hand, if hits are detected, f_i obtains a tentative index, $k^{(\mu)}$, by adding the number of hit occurrences when $k_i = k^{(\mu-1)}$ to the value of $k^{(0)}$; i.e.,

$$\begin{aligned} k^{(\mu)} &= f_i(k^{(\mu-1)}) \\ &= k^{(0)} + H_{i,j}(k^{(\mu-1)}, k_j) + \bar{H}_{i,j}(k^{(\mu-1)}, k_j), \\ \text{where } k^{(0)} &= \ell_i. \end{aligned} \tag{8}$$

Since the assumption in this work is that the maximum number of MTC devices does not exceed N, there exists at least one index, k, at which no additional hits are detected; i.e., $f_i(k) = k$. This index will be equal to k_i, and using it with k_j and $\bar{k}_j$ in (6) yields the number of pairwise hits between the sequences corresponding to the pairs (g_i, s_i), (g_j, s_j) and (g_j^{-1}, s_j^{-1}). Note that k_i depends on the RT loads, and therefore must be revaluated for each load distribution.

We conclude this section by noting that the evaluation of k_i in (8) is bounded by N iterations, each involving 3 additions. Hence, the computational complexity of evaluating k_i is upper bounded by $3N$ additions.

D. Computational Complexity of the Greedy Algorithm

To assess the computational complexity of the greedy algorithm, we note that this algorithm relies on evaluating I_i in (7) for each of the group generator pairs in $\mathcal{U}$; i.e., $\forall (g_i, g_i^{-1}) \in \mathcal{U}$, $i = 1, \ldots, \frac{\phi(N)}{2}$. This evaluation involves only addition operations, but no multiplications. In particular, computing I_i for the pair (g_i, g_i^{-1}) involves: 1) the evaluation of the $H_{i,j}$ and $\bar{H}_{i,j}$ matrices; 2) the index k_i; and 3) the cardinality of the set $|\mathcal{Y}|$. The number of additions required to evaluate each of the pairwise hit matrices in (7) is bounded by N^3, and the complexity of evaluating k_i for a given load distribution is bounded by $3N$ additions; see previous section for details. Finally, by construction, the cardinality of $\mathcal{Y}$ in (7) is equal to the number of ordered partitions of the integer N. This number is given by $|\mathcal{Y}| = \binom{N-1}{2}$ [35]. Using these elements, we can now provide a bound on the complexity of the proposed greedy algorithm.

In essence, the algorithm iteratively selects the group generator in $\mathcal{U}$ that yields the minimum number of pairwise hits with the quadruples in $\mathcal{T}$. We note that when computing the number of hits between a group generator g_i and a given quadruple in $\mathcal{T}$ over all possible load combinations, the pairwise hit matrices need to be evaluated once; these matrices do not depend on the RT loads. In addition, the number of possible load combinations for which k_i must be evaluated is given by $|\mathcal{Y}|$. Hence, the complexity of evaluating the pairwise number of hits between a group generator g_i and a given quadruple in $\mathcal{T}$ over all possible

load combinations is bounded by

$$2N^3 + 3N\binom{N-1}{2} + \binom{N-1}{2} = 3.5N^3 - 4N^2 + 1.5N + 1. \quad (9)$$

In the left hand side of (9), the first term represents the number of additions required for evaluating the pairwise matrices, $H_{i,j}$ and $\bar{H}_{i,j}$. The second term represents the number of additions required to evaluate the index k_i for the $\binom{N-1}{2}$ possible load distributions. Finally, the last term represents the complexity involved in the summation of the number of pairwise hits for the $\binom{N-1}{2}$ load distributions.

Since s_i can take values from 0 to $N-1$, it follows that the complexity of evaluating the performance of a candidate group generator, g_i, for all possible cyclic shifts with respect to a quadruple in $\mathcal{T}$ is bounded by $3.5N^4 - 4N^3 + 1.5N^2 + N$. Hence, the number of additions required for evaluating the performance of g_i against all the quadruples in $\mathcal{T}$ can be bounded by the product of this number and the number of quadruples in $\mathcal{T}$; i.e., $(3.5N^4 - 4N^3 + 1.5N^2 + N)(\frac{\phi(N)}{2} - 1)$. Finally, to determine the overall complexity of the greedy algorithm, we note that this algorithm exhausts the group generator pairs in $\mathcal{U}$ in $\frac{\phi(N)}{2} - 1$ iterations, each of which involves all the entries in $\mathcal{U}$. Since $|\mathcal{U}| \leq \frac{\phi(N)}{2} - 1$, it follows that the overall computational complexity of the algorithm is bounded by

$$(3.5N^4 - 4N^3 + 1.5N^2 + N)\left(\frac{\phi(N)}{2} - 1\right)^3. \quad (10)$$

We note that by definition, for any given integer N, $\phi(N) < N$. Hence, the computational complexity of the greedy algorithm in (10) can be bounded by $\mathcal{O}(N^7)$. This is in sharp contrast with the complexity of the exhaustive search in [26], which was shown to be bounded by $\frac{1}{2}M(M-1)N^M(N^2(N-1) + \binom{N-1}{M-1})$; i.e., exponential in the number of RTs, M. Hence, in contrast with exhaustive search proposed in [26], the greedy algorithm is readily applicable in IoT systems with potentially large numbers of RTs.

V. The Graphical Greedy Algorithm

In the previous section, we developed an efficient greedy algorithm that yields a look-up table of RB assignment sequences. The computational complexity of this algorithm was shown to be polynomial and to depend solely on N, but not on M. For example, using (10), the computational complexity of constructing the look-up table for a system with any number of RTs, M, and $N = 100$ RBs can be bounded by 2.5×10^{12} additions, which is manageable by widely available processors. However, for large IoT systems that serve thousands of devices concurrently, the greedy algorithm is not computationally feasible. For instance, for an MTC system with $N = 1000$ RBs, the computational complexity of the greedy algorithm is bounded by 10^{21} additions! For such systems, more efficient sequence selection mechanisms are required.

In this section, we propose an alternate greedy algorithm that significantly reduces the computational complexity required for selecting the assignment sequences. The philosophy of this algorithm is similar to that of the one proposed in the previous section, but it depends on a metric that is inspired by the graphical representation of cyclic groups. In particular, this metric is based on the observation that the RBs located at the beginning of each sequence, referred to as the dominating RBs, are the most significant contributors to the number of hit occurrences. This is because these RBs will always be assigned by the RTs to incoming MTC devices even when the system is lightly loaded. Hence, to reduce the number of hit occurrences, the CG sequences should be selected such that their dominating RBs do not overlap.

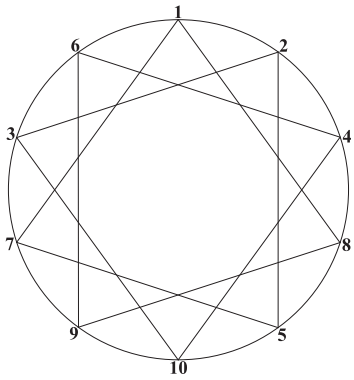

Fig. 3. A graphical representation of a cyclic group of order 10. This cyclic group has 2 group generator pairs that are represented by the outer circle and the inner pattern, respectively.

Following analogous steps to those used in the greedy algorithm discussed hereinabove, we let $\mathcal{T}_g$ be the look-up table to be constructed by the graphical greedy algorithm and $\mathcal{U}_g$ be the set containing the group generator pairs that have yet to be selected. In each iteration, the algorithm uses a metric based on the graphical representation of cyclic groups to evaluate the number of pairwise hits between the dominating RBs of the sequences CG by each of the group generator pairs in $\mathcal{U}_g$ and those CG by the quadruples already selected in $\mathcal{T}_g$. This is in contrast with the greedy algorithm discussed in Section IV, which considers all the N RBs when evaluating the number of pairwise hits. The look-up table $\mathcal{T}_g$ is then augmented by the group generator pair and the cyclic shifts that correspond to the sequence resulting in the minimum number of hits between dominating RBs. The graphical representation of cyclic groups and the metric inspired thereby are discussed in more details in the following sections.

1) The Graphical Representation of Cyclic Groups [36]: A cyclic group with finite order can be represented graphically by connecting its elements in the same order as they appear in its CG sequences. To do so, the group elements are first positioned on the circumference of a circle in the same order of one cyclic sequence and then connected according to the remaining ones. This results in a particular pattern that connects all the group elements for each cyclic sequence. An illustrative example of the graphical representation of the ACG of order 10 is presented in Fig. 3. In this figure, the group has 2 pairs of group generators, one pair corresponds to the outer circle while the other corresponds to the inner pattern.

Since shifting a sequence cyclically rotates its entries, this shifting will rotate the pattern corresponding to the sequence. This rotation is clockwise when the cyclic shift is negative and anticlockwise when it is positive.

We conclude this section by the following remarks:

Remark 1: Using the isomorphism property, it can be readily shown that the graphical representation of a cyclic group depends solely on the group order N; i.e., it is independent of the group operation; cf. Definition 4. Hence, it is sufficient to study the properties of the graphical representation of only the ACG. ■

Remark 2: It was shown in [36] that the graphical representation of a cyclic group is independent of the CG sequence that is selected to construct the outer circle. Hence, the inner patterns of this representation are unique and depend only on the group order. ■

Using the graphical representation of cyclic groups, in the following section, we provide an alternate greedy algorithm that selects the CG sequences based on their patterns.

2) The Graphical Metric: Let $\mathcal{T}_g$ be the look-up table to be constructed by the algorithm and let $\mathcal{U}_g$ be the set containing the group generator pairs that have yet to be selected. At first, $\mathcal{T}_g$ is empty and $\mathcal{U}_g$ contains the $\frac{\phi(N)}{2}$ group generator pairs. In the first iteration, a pair (g, g^{-1}) is arbitrarily selected and removed from $\mathcal{U}_g$. By associating a cyclic shift $s = 0$ with this pair, we obtain the first quadruple (g, g^{-1}, s, s^{-1}) of the look-up table $\mathcal{T}_g$. This quadruple is then used to generate the first pattern in the graphical representation of the cyclic group; i.e., the outer circle. The dominating RBs in these two sequences are then marked. The number of these dominating RBs ranges from 1 to N and depends on the system load. However, since the contribution of an RB to the average number of hits decreases with its position in the sequence, we will consider only the few RBs at the beginning of each sequence to be the dominating RBs.

Following analogous steps to the greedy algorithm discussed hereinabove, in each of the subsequent $\frac{\phi(N)}{2} - 1$ iterations, the algorithm examines the group generator pairs in $\mathcal{U}_g$ for all possible cyclic shifts. However, unlike the greedy algorithm which considers the hits between the N RBs over all possible load combinations, the graphical algorithm considers only the hits between dominant RBs in the patterns corresponding to the pairs in $\mathcal{U}_g$ and the quadruples in $\mathcal{T}_g$. The look-up table $\mathcal{T}_g$ is then augmented by the quadruple (g, g^{-1}, s, s^{-1}) that yields the minimum hits between the dominant RBs. The group generators of this quadruple are then removed from $\mathcal{U}_g$ and the algorithm iterates until exhaustion of all pairs in $\mathcal{U}_g$. As an example of the graphical metric, consider the ACG, $\mathbb{Z}_{12}$, and consider the dominant RBs to be the first two in each sequence. This cyclic group has two pairs of group generators and its CG sequences are as follows:

$(g_1, s_1) = (1, 0)$	1	2	3	4	5	6	7	8	9	10	11	0
$(g_1^{-1}, s_1^{-1}) = (11, 11)$	0	11	10	9	8	7	6	5	4	3	2	1
$(g_2, s_2) = (5, 0)$	5	10	3	8	1	6	11	4	9	2	7	0
$(g_2^{-1}, s_2^{-1}) = (7, 11)$	0	7	2	9	4	11	6	1	8	3	10	5

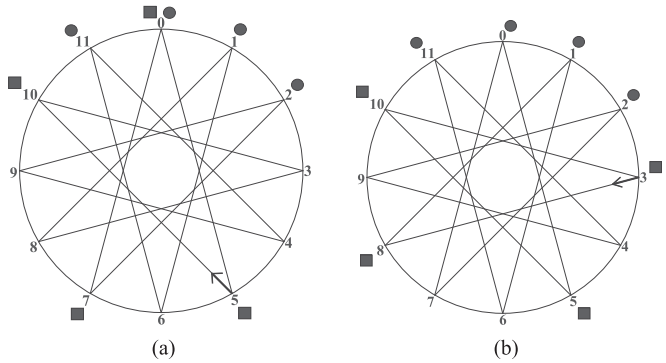

Fig. 4. Rotation of the inner pattern of $\mathbb{Z}_{12}$ to minimize the number of hits between the dominant RBs (the first element of the inner pattern is marked by an arrow for ease of exposition). (a) Unrotated pattern. (b) Rotated pattern.

The corresponding patterns of these two pairs and their dominant RBs are depicted in Fig. 4(a). The outer circle corresponds to the group generator pair in $\mathcal{T}_g$, and the inner pattern corresponds to the examined pair in $\mathcal{U}_g$. The dominant RBs of the pair corresponding to the outer circle are marked with circles and the ones of the pair corresponding to the inner pattern are marked with squares. To avoid the hit at RB 0, the inner pattern is rotated anticlockwise by a cyclic shift $s = 2$. The rotated pattern is depicted in Fig. 4(b). The look-up table is then augmented by the quadruple $(g, g^{-1}, s, s^{-1}) = (5, 7, 2, 9)$, which corresponds to the rotated inner pattern.

Finally, we note that the quadruples in the look-up table yielded by the graphical greedy algorithm depends solely on the number of pairwise hits, but not on the number of RTs in the system. Hence, active RTs do not need to update their sequences when new ones join the system. This is in contrast with the exhaustive search scheme, which requires each active RT to update its RB assignment sequence when a new RT enters the system thus resulting in a significant signalling overhead that might not be feasible for IoT systems with large numbers of RTs and MTC devices.

A. Computational Complexity of the Graphical Greedy Algorithm

The computational cost of the graphical greedy algorithm is mainly due to the evaluation of the number of hits between dominant RBs. This cost involves only addition operations and, similar to the algorithm in Section IV, is polynomial in N. However, the order of the computational complexity of the graphical greedy algorithm is significantly lower than that of the greedy algorithm. To see this, in this section we provide a bound on the complexity of the graphical greedy algorithm.

Denoting the number of dominant RBs by D, it can be seen that the number of required additions for evaluating the number of hits between dominant RBs of a pattern corresponding to an examined pair from $\mathcal{U}_g$ and all quadruples in $\mathcal{T}_g$ is bounded by $4D$. Since the cyclic shift can take values from 0 to $N - 1$, it follows that the complexity of evaluating the number of hits

TABLE I
A COMPARATIVE NUMERICAL EXAMPLE OF THE COMPLEXITIES OF THE GREEDY ALGORITHMS AND THE EXHAUSTIVE SEARCH

Search method	Exhaustive search	Greedy algorithm	Graphical greedy
$(M, N) = (6, 100)$	10^{21}	10^{14}	10^{8}
$(M, N) = (10, 100)$	10^{33}	10^{14}	10^{8}
$(M, N) = (10, 1000)$	10^{53}	10^{21}	10^{12}

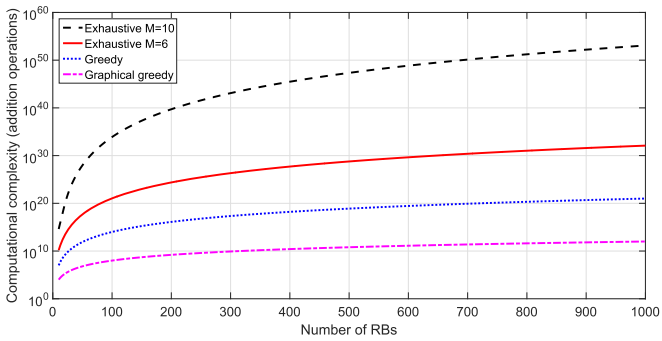

Fig. 5. Comparison between the computational complexities of the proposed greedy algorithms and exhaustive search for systems with $M = 6$ and $M = 10$ RTs.

between the dominant RBs of the examined pattern over all possible cyclic shifts can be bounded by $4ND$. Finally, since $D \leq N$ and in each iteration $|\mathcal{U}_g| \leq \frac{\phi(N)}{2} - 1$, it follows that the computational complexity of the graphical greedy algorithm is bounded by

$$4ND\left(\frac{\phi(N)}{2} - 1\right)^2 \leq N^4, \tag{11}$$

which shows that the graphical greedy algorithm offers a computational saving of N^3 over the greedy algorithm of Section IV; cf. Section IV-D. This is because, when selecting the RB assignment sequences, the graphical greedy algorithm considers only the number of hits between dominant RBs, which is independent of the system load. A numerical comparison between the complexity bounds on the greedy algorithms and the exhaustive search is provided in Table I.

From Table I, it can be seen that the computational complexity of exhaustive search is exponential in the number of RTs; cf. Section IV-D. Hence, it is computationally prohibitive even for IoT systems with small numbers of RTs; e.g., exhaustive search for a system with $M = 6$ and $N = 1000$ requires 10^{53} additions. In contrast with exhaustive search, the computational complexities of both greedy algorithms are polynomial and independent of the number of RTs. Hence, they are applicable in IoT systems, which comprise hundreds of RTs. The computational complexities of both greedy algorithms and exhaustive search are presented in Fig. 5.

VI. SIMULATION RESULTS

In this section, we compare, in the first four examples, the performance of the RB assignment sequences obtained by the two greedy algorithms developed herein, the uniformly-distributed random assignment scheme developed in [24], and the exhaustive search developed in [26] for different relative loads $\frac{L}{N}$; i.e., different ratios of the currently assigned RBs, L, to the total number of RBs in the system, N. For the scheme in [26] and the ones proposed herein, the cyclic sequences are generated using the MCG structure of Section IV-A with $N = P - 1$, where P is a prime number; cf. [26]. In all numerical examples, the MTC devices will select their assisting RTs from the pool of RTs that are capable of and willing to assist communication irrespective of their status; i.e., whether having its own data to transmit or not.

To be able to compare the performance of the greedy algorithms developed herein with that of the exhaustive search developed in [26], in the first two examples, we will restrict attention to the case of $M = 3$ RTs; exhaustive search is computationally prohibitive for $M > 3$. In these examples the performance is measured by the average number of hits over all possible load combinations; i.e., over all ordered triples $(\ell_1, \ell_2, \ell_3) \in \mathcal{L} \triangleq \{(\ell_1, \ell_2, \ell_3) | \sum_{i=1}^{3} \ell_i = L, \ell_i \geq 0\}$, where ℓ_1, ℓ_2 and ℓ_3 are the load levels of RTs 1, 2 and 3, respectively. In these examples, the index of the last assigned RB by each of the 3 RTs, (k_1, k_2, k_3), is evaluated for each load distribution $(\ell_1, \ell_2, \ell_3) \in \mathcal{L}$; cf. Section IV-C.2.

In the following two examples of this section, we will increase the number of RTs and RBs and compare the performance of the graphical greedy algorithm with the random assignment of [24]. In both examples, we will consider only 1000 MTC load distributions that are uniformly distributed over the M RTs. This is because the number of load combinations increases significantly with N and M; e.g., for $N = 500$ and $M = 25$, this number is equal to $\binom{499}{24}$.

Example 3: In this example, we compare the performance of the CG sequences obtained by the greedy algorithm developed in Section IV, the graphical greedy algorithm of Section V, and the exhaustive search in [26] when the number of RBs $N = 16$ and $N = 40$. For both cases, the number of dominant RBs of the graphical greedy algorithm is set to $D = 3$. The performance of these algorithms is depicted in Fig. 6.

From Fig. 6, it can be seen that the CG sequences obtained by exhaustive search and the greedy algorithms outperform randomly-generated ones. For example, for a system with $N = 40$ RBs at a relative load, $\frac{L}{N} = 90\%$, the average number of hits resulting from RB assignments based on exhaustive search [26], greedy algorithm of Section IV, graphical greedy algorithm of Section V, and random sequences [24] are 4.4, 6.6, 7.4 and 9.1, respectively. This figure also shows that the greedy-based sequences provide comparable performance to the upper bound yielded by exhaustive search. The reason that exhaustive search constitutes an upper bound on the performance follows from the fact that it jointly optimizes the performance of the CG sequences of all the M RTs over all possible group generators and cyclic shifts. In contrast, the greedy algorithms sequentially

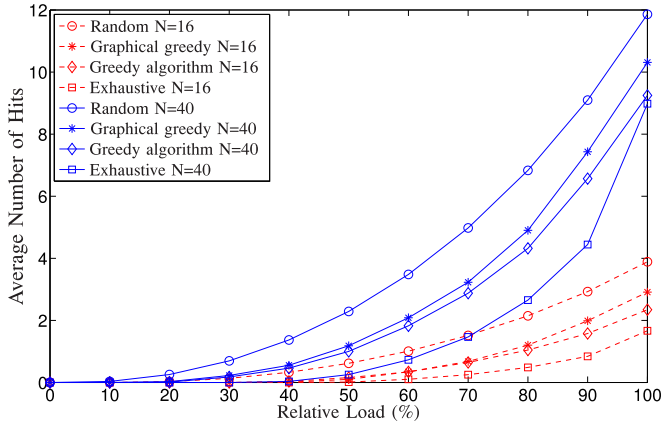

Fig. 6. Comparison between randomly-generated assignment sequences [24] and the CG sequences obtained by the exhaustive search in [26] and the greedy algorithms.

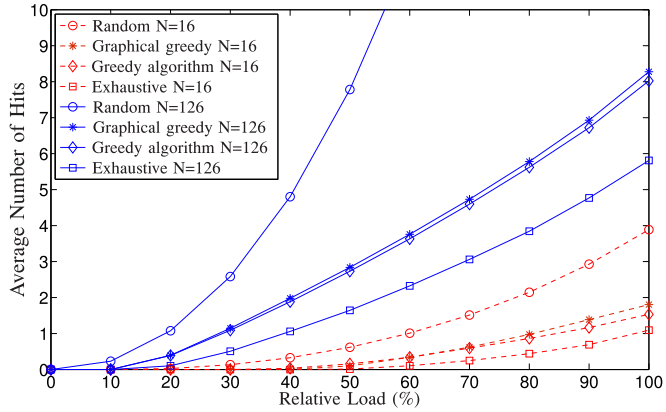

Fig. 7. Comparison between randomly-generated assignment sequences of [24] and the CG sequences obtained by exhaustive search in [26] and the greedy algorithms with HIA.

optimize the CG sequences. In particular, these algorithms consider only the previously selected sequences when optimizing current ones. Despite their potential suboptimality, the greedy algorithms possess significantly lower computational complexities when compared with exhaustive search. This renders them suitable for systems with practical numbers of RTs and MTC devices; for such systems exhaustive search is computationally prohibitive, cf. Table I. ■

Example 4: In this example we consider a setup similar to the one in the previous example, but with $N = 16$ and $N = 126$. For both cases, the number of dominant RBs of the graphical greedy algorithm is set to $D = 3$. In this example, the RTs apply the hit identification and avoidance (HIA) technique developed in [26]. In this technique, each RT is assumed to know the assignment sequences of the neighboring RTs and to be able to identify the RT with which it collided once a hit occurs. This is possible, for instance, if neighboring RTs use different modulation schemes. Using this information, the RTs update their assignment sequences to avoid the RBs that are already assigned by neighboring RTs. Note that the HIA technique cannot be applied to the sequences that are randomly-generated because such sequences do not possess a specific structure. In Fig. 7, we depict the performance improvement when the HIA technique is used with the MCG structured sequences generated by exhaustive search, the greedy algorithm in Section IV and the graphical greedy algorithm in Section V. For comparison, the performance of the randomly-generated sequences is also shown.

From this figure, it can be seen that using the HIA algorithm significantly reduces the average number of hit occurrences. For example, for the case of $N = 16$ RBs, the usage of the HIA reduces the average number of hits of the greedy algorithm from 2.3 to 1.5 when $\frac{L}{N} = 100\%$. The reason behind this reduction is that HIA enables the RTs to avoid future hits. In particular, when the HIA is implemented, the RTs are able to estimate the load levels of their neighbors based on their previous hits and subsequently avoid RB assignments that would result in future hits. Note that from this figure, it can be seen that the HIA also reduces the performance gap between sequences generated by exhaustive search and those generated by the greedy algorithms. ■

In the previous examples, we provided performance results for IoT systems with relatively small numbers of RTs and RBs. In the following two examples, systems with larger numbers of RTs and RBs will be considered. In these examples, the performance of the CG sequences obtained by the graphical greedy algorithm will be compared only with the randomly-generated sequences [24] since the exhaustive search scheme in [26] and the greedy algorithm of Section IV are computationally expensive in this case.

Example 5: In this example, we compare the performance of the RB assignment sequences generated by the graphical greedy algorithm of Section V and the randomly-generated assignment sequences [24], when the number of RBs $N = 520$ and $N = 750$. In both cases, the number of RTs is $M = 15$ and the number of dominant RBs is set to $D = 25$. The performance of the graphical greedy algorithm with and without the implementation of the HIA technique is compared with that of the random assignment scheme over 1000 load distributions of MTC devices for each relative load $\frac{L}{N}$. Note that the exhaustive search of [26] and the greedy algorithm of Section IV are not considered in this example because of their large computational complexity. For example, the computational complexity of exhaustive search when $N = 520$ and $M = 15$ is in the order of 10^{69}, whereas that of the greedy algorithm is in the order of 10^{17}.

From Fig. 8, it can be seen that for both the $N = 520$ and $N = 750$ scenarios, the sequences that are obtained by the graphical greedy algorithm outperform those that are randomly-generated. This performance advantage increases significantly with the implementation of the HIA technique. For example, for a system with $N = 520$ RBs at a relative load, $\frac{L}{N} = 90\%$, the average number of hits resulting from using the randomly-generated sequences and that resulting from using the sequences obtained by the graphical greedy algorithm without HIA are 623 and 504, respectively. In contrast, the

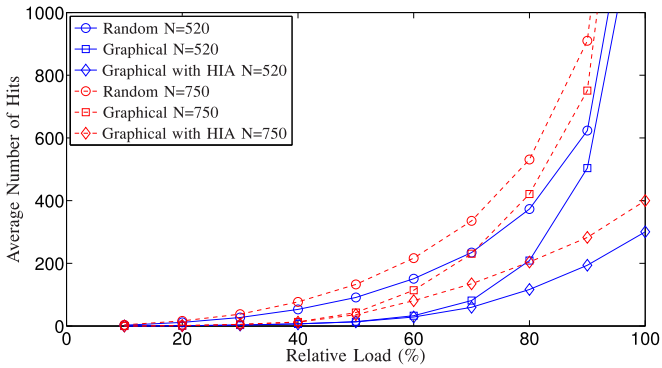

Fig. 8. Comparison between the random [24] and the graphical greedy-based RB assignment sequences with and without the HIA technique for $N = 520$ and $N = 750$ RBs.

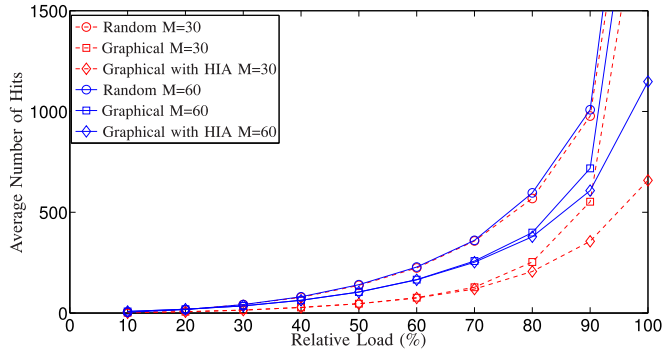

Fig. 9. Comparison between the random [24] and the graphical greedy-based RB assignment sequences with and without the HIA for $M = 30$ and $M = 60$ RTs.

corresponding number of hits resulting from using the ones obtained by the graphical greedy algorithm with HIA is only 194. In addition, it can be seen from the figure that the performance advantage of using the HIA algorithm increases with N at high loading scenarios. For example, the performance of the system with $N = 750$ and HIA was found to be superior to that of the system with $N = 520$, but without HIA, when $\frac{L}{N} \geq 80\%$. Similar to the instance examined in Example 4, this phenomenon results from the implementation of the HIA algorithm by the RTs, which enables them to estimate the load levels of their neighbors and accordingly avoid future hits. This performance advantage is more pronounced at highly loaded scenarios with large numbers of RBs since the accuracy of estimating the load levels of the neighbors of each RT improves with the increasing number of hits. In other words, although high loads and larger numbers of RBs usually result in more hits, they can have a positive impact on the performance when the HIA is implemented. Finally, we note that the increase in the average number of hits with respect to the previous examples is mainly due to the larger number of RTs and RBs and the consideration of only 1000 load distributions of MTC devices. ■

Example 6: In this example, we compare the performance of the CG RB assignment sequences obtained by the graphical greedy algorithm and the randomly-generated assignment sequences [24] when the number of RTs $M = 30$ and $M = 60$. In both cases, the number of RBs is $N = 750$ and the number of dominant RBs is set to $D = 25$. Similar to the previous example, the performance of the graphical greedy algorithm is evaluated with and without the implementation of HIA.

From Fig. 9, it can be seen that the sequences obtained by the graphical greedy algorithm outperforms those that are randomly-generated for $M = 30$ and $M = 60$. For example, for a system with $M = 30$ RTs and a relative load of $\frac{L}{N} = 90\%$ the average number of hits resulting from using the randomly-generated sequences and that resulting from using the sequences obtained by the graphical greedy algorithm without HIA are 977 and 552, respectively. In contrast, the corresponding number of hits resulting from using the sequences obtained by the graphical greedy algorithm with HIA is 356. In other words, the greedy algorithm without HIA has a 40% gain over the randomly-generated assignment sequences. However, this gain decreases as the number of RTs increases. For example, this gain reduces to 30% at the same relative load when $M = 60$. This is because the number of utilized assignment sequences increases with M, which results in the selection of assignment sequences with lower performance. Finally, it can be seen from the figure that at a relative load of $\frac{L}{N} = 100\%$, the system with $M = 60$ and HIA is superior to that with $M = 30$, but without HIA. The reason behind this phenomenon is the hit avoidance feature provided by the HIA as described in the previous example. ■

Example 7: In this example, we evaluate the aggregate bit rate when the proposed sequences are used in a system with $M = 3$ RTs and $N = 40$ RBs. Each RT assigns one RB to each incoming MTC device during a scheduling interval of 1 ms. In other words, in each scheduling interval, 3 RBs corresponding to $N = 40$ RTs are assigned. Once RT i assigns an RB to an incoming MTC device, it increases the number of its assigned RBs, N_i, by one. At this point, two possibilities can occur, either successful assignment or collision. In case of a successful assignment, the assigned RB is retained and the RT proceeds during the next scheduling interval to assign the following RB in its sequence to the incoming MTC device. In case of collision, the RT which was first to assign the RB retains it, whereas the other conflicting RT releases the RB and reduces its N_i by one at the end of its scheduling interval. The impact of a collision will be reflected on the interference observed by the MTC devices. In particular, a collision on a given RB will result in an interference between the transmissions of their respective MTC devices for the duration of one scheduling interval. In this example, the system is initially unloaded and the number of MTC devices is increased in a round Robin fashion until two RTs are loaded with 13 MTC devices and the third RT is loaded with 14 MTC devices. In that case, the system is fully loaded.

The channel gains between the RTs and the MTC devices are generated using the line-of-sight Hotspot pathloss model [37] with frequency-flat Rayleigh fading and standard Gaussian noise spectral density of -174 dBm/Hz. The RB bandwidth, and the receiver noise figure are set to 180 KHz and 5 dB, respectively [37]. The transmit power, P_t, of the MTC devices is set

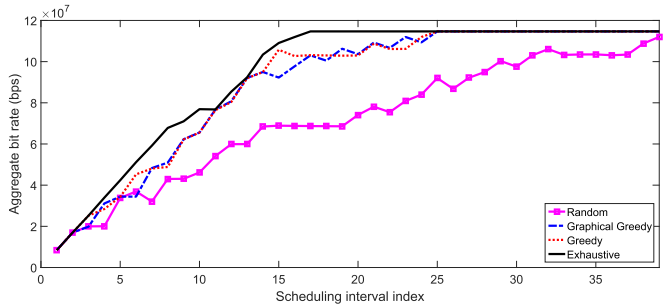

Fig. 10. Aggregate bit rates of random [24], greedy-based and exhaustive search-based RB assignment sequences.

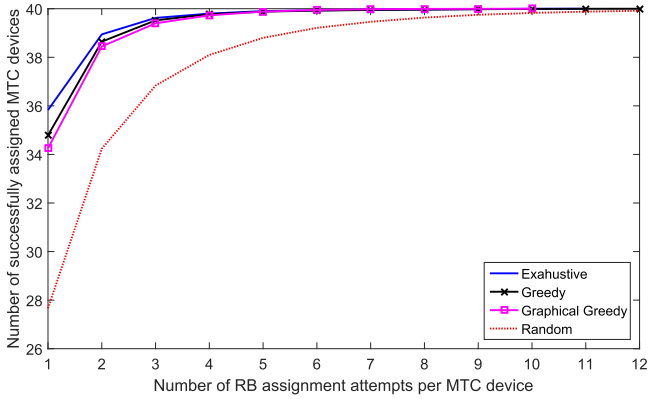

Fig. 11. CDF of the number of RB assignment attempts needed for successful assignment.

to 0 dBm and the locations of the MTC devices and the RTs are uniformly distributed within the cell boundaries. For a practical dense deployment scenario, the MTC devices are assumed to be less than 300 m away from their assisting RTs and the maximum inter-RT distance is set to 100 m. The RTs implement the HIA algorithm of [26] and the performance is averaged over 10000 channel realizations. Using the aforementioned parameters, the aggregate data rate, R, at a given scheduling interval can be calculated using:

$$R = \sum_{i=1}^{M} \sum_{j=1}^{N_i} \log_2 \left(1 + \frac{|H_{q,i,B_j}|^2 P_t}{P_n + \sum_{m=1, m \neq q}^{Q} |H_{m,i,B_j}|^2 P_t} \right), \tag{12}$$

where P_n is the noise power, B_j is the j-th RB assigned by RT i, Q is the total number of MTC devices in the system, and H_{q,i,B_j} is the channel coefficient between RT i and MTC device q which is assigned B_j and is set to zero if q is not assigned B_j. Note that, the selected RB, B_j, by the i-th RT will depend on its sequence.

As shown in Fig. 10, the sequences obtained by the greedy algorithms significantly outperform the random RB assignments and provide comparable performance to those obtained by exhaustive search. For example, at the 17^{th} scheduling interval, the greedy-based sequences achieve an aggregate bit rate of 10.3×10^7 bps, which is only 10% lower than that of its exhaustive search counterpart. In contrast, random RB assignments achieve a rate of only 6.86×10^7 bps. Fig. 10 also illustrates the effect of hit occurrences on the aggregate bit rate. This effect is reflected in the jagged behavior of the aggregate bit rate curves. In particular, during a scheduling interval, the three RTs perform RB assignments to their respective MTC devices. At this point, three possibilities can occur: 1) successful RB assignments with no collisions; 2) collisions between concurrent RB assignments; or 3) collisions between concurrent RB assignments and previous ones. In case one, the aggregate rate increases by the sum of the rates of the new transmissions. In case two, the aggregate rate will remain flat when all the concurrent RB assignments collide since all transmissions are unsuccessful, whereas when only two assignments are colliding, the rate will slightly increase since a successful transmission will be achieved on one RB. In case three, the aggregate rate can either: 1) significantly decrease when all the concurrent assignments collide; 2) slightly decrease when two assignments collide; or 3) slightly increase when one assignment collide. Hence, from these observations, it can be concluded that the number of hits plays a key role in constraining the aggregate bit rate. ■

Example 8: In this example, we evaluate the number of MTC devices with successful RB assignments vs the number of RB assignment attempts. The setup of this simulation is similar to that of the previous one. However, unlike the previous example, the total system load is not uniformly distributed across the 3 RTs and the performance results are averaged over 10000 load distributions across the 3 RTs.

Consistent with our observations in the previous example, Fig. 11 shows that the greedy-based sequences significantly outperform their random counterparts and provide comparable performance to those obtained by exhaustive search. For example, for two RB assignment attempts, the average numbers of MTC devices with successful assignments are 38.94, 38.64, 38.44, and 34.23 for RB assignments based on exhaustive search, greedy, graphical greedy, and random sequences, respectively. From Fig. 11, it can be seen that using the sequences selected by exhaustive search and greedy algorithms, the 99-th percentile of all MTC devices are successfully assigned with at most three RB assignment attempts, whereas in the case of random assignments the 99-th percentile is reached with eight attempts. Using the results reported in Fig. 11, one can evaluate the power wastage incurred by collisions and the data transmission delay experienced by the MTC devices; i.e., the duration between the first RB assignment and the successful one. This can be done by invoking the scheduling interval and P_t.

Denoting the transmission delay by d and the number of RB assignment attempts by n, it can be seen that $d{=}s(n-1)$, where s is the scheduling interval duration. For example, a device with 3 assignment attempts will incur a delay of 2 ms.

To evaluate the power wastage due to collisions, denoted by P_w, one can use $P_w = P_t(n-1)$. Using Fig. 11, it can be concluded that 96% and 85.5% of the MTC devices incurred

less than 1 mW power wastage and 1 ms delay when using greedy-based and random sequences, respectively. ■

We conclude this section by noting that the proposed work offers a flexible tradeoff between sequence generation complexity and performance by focusing attention on CG sequences due to their favorable features; cf. Section II. However, the underlying cyclic structure limits the number of sequences of length N that are available for RB assignments to $N\phi(N)$, which is much less than the total number of sequences, $N!$. Such a limitation might have an impact on performance. To overcome this limitation and potentially improve performance, it may be possible to enrich the set of available sequences by seeking number theoretic techniques for combining multiple cyclic ones.

VII. Conclusion

Relaying terminals are prospected to facilitate MTC in future cellular networks. However, their effective utilization is contingent upon the development of efficient RB assignment schemes that do not depend on CQIs. In this paper, we considered CG sequences for RB assignment. We showed that all CG sequences are isomorphic to the ones obtained by the ACG structure. This property enables us to eliminate the prime minus one restriction of the MCG on the number of RBs and ensures that the optimal CG sequences by the additive group operation are globally optimal over the set of all CG sequences. To facilitate the selection of these sequences, we developed a greedy algorithm with polynomial complexity for systems with large numbers of RTs and RBs. To further reduce the computational cost, we invoked the graphical representation of cyclic groups to develop a more efficient greedy algorithm. Numerical results show that the greedy-based sequences provide comparable performance to that of the sequences obtained by exhaustive search and significantly outperform randomly-generated ones.

Appendix
Proof of Theorem 1

Let $\mathbb{Z}_N$ be an ACG and $\mathbb{G}_N$ be an arbitrary cyclic group, both of order N. From Lemma 1, there is at least one isomorphism function that maps the elements of $\mathbb{G}_N$ to those of $\mathbb{Z}_N$. Furthermore, Theorem 3.26 in [38], asserts that such an isomorphism maps every group generator in $\mathbb{G}_N$ to a distinct group generator in $\mathbb{Z}_N$. Using these facts, we will develop a constructive proof to show that the sequences obtained by the generators of $\mathbb{G}_N$ are isomorphic to those obtained by the generators of $\mathbb{Z}_N$.

Let g_1 and g_2 be two distinct generators of $\mathbb{G}_N$ and let F be the isomorphism that maps these generators to two group generators of $\mathbb{Z}_N$, say z_1 and z_2, i.e., $g_1 \overset{F}{\mapsto} z_1$ and $g_2 \overset{F}{\mapsto} z_2$. To establish the desired isomorphism, we will generate the sequences in $\mathbb{Z}_N$ that are isomorphic to the ones generated by g_1 and g_2 in two different ways. In the first way, the sequences in $\mathbb{Z}_N$ are generated by applying F to every element of the sequences generated by g_1 and g_2. However, at this point, there is no guarantee that the sequences obtained in this way can be CG by the group generators of $\mathbb{Z}_N$. To prove that this is indeed the case, in the second way, we will make use of Theorem 3.26 in [38] to determine z_1 and z_2, and to obtain their corresponding sequences. The isomorphism between the cyclic sequences obtained by $\mathbb{G}_N$ and $\mathbb{Z}_N$ is established by showing that the sequences of the two ways are identical.

For the first way, we note that the cyclic sequences of g_1 and g_2 can be expressed as

$$S_1 = (g_1^1 (\mathrm{mod}\, N+1), \ldots, g_1^N (\mathrm{mod}\, N+1), \text{ and} \tag{13}$$

$$S_2 = (g_2^1 (\mathrm{mod}\, N+1), \ldots, g_2^N (\mathrm{mod}\, N+1)). \tag{14}$$

Since the sequence generated by any of the group generators of $\mathbb{G}_N$ spans the complete sequence of integers from 1 to N, then there exists an integer $1 \leq q \leq N$ such that $g_2 = g_1^q \pmod{N+1}$. Subsequently, S_2 can be written as

$$S_2 = (g_1^q (\mathrm{mod}\, N+1), \ldots, g_1^{Nq} (\mathrm{mod}\, N+1)), \tag{15}$$

where in writing (15), we have used the fact that $(g_1^q \pmod{N+1})^m \pmod{N+1} = g_1^{mq} \pmod{N+1} \quad \forall m \in \{1, \ldots, N\}$.

Next, we define an isomorphism function, F that maps the element $g_1^k \pmod{N+1} \in \mathbb{G}_N$ to the element $k \pmod{N} \in \mathbb{Z}_N, k = 0, \ldots, N-1\}$. Applying this isomorphism on the sequences S_1 and S_2 yields

$$S_1' = (1, 2, \ldots, N-1, 0), \text{ and} \tag{16}$$

$$S_2' = (q (\mathrm{mod}\, N),\ 2q (\mathrm{mod}\, N), \ldots,\ Nq (\mathrm{mod}\, N)). \tag{17}$$

We will now provide another way for generating these sequences. Applying F on g_1 and g_2 yields z_1 and z_2, which are group generators of $\mathbb{Z}_N$, cf. [38, Theorem 3.26], where

$$z_1 = F(\, g_1 (\mathrm{mod}\, N+1)) = 1, \qquad \text{and} \tag{18}$$

$$\begin{aligned} z_2 &= F(\, g_2 (\mathrm{mod}\, N+1)) \\ &= F(\, g_1^q (\mathrm{mod}\, N+1)) = q (\mathrm{mod}\, N). \end{aligned} \tag{19}$$

Since z_1 and z_2 are group generators of $\mathbb{Z}_N$, applying (2) on them yields the cyclic sequences

$$\tilde{S_1} = (z_1 (\mathrm{mod}\, N),\ 2z_1 (\mathrm{mod}\, N), \ldots,\ Nz_1 (\mathrm{mod}\, N)), \tag{20}$$

$$\tilde{S_2} = (z_2 (\mathrm{mod}\, N),\ 2z_2 (\mathrm{mod}\, N), \ldots,\ Nz_2 (\mathrm{mod}\, N)). \tag{21}$$

Substituting (18) and (19) in (20) and (21) yields

$$\tilde{S_1} = (1, 2, \ldots, 0) = S_1', \qquad \text{and} \tag{22}$$

$$\tilde{S_2} = (q (\mathrm{mod}\, N), 2q (\mathrm{mod}\, N), \ldots, Nq (\mathrm{mod}\, N)) = S_2'. \tag{23}$$

The equalities in (22) and (23) yield that the set of sequences generated by an arbitrary cyclic group $\mathbb{G}_N$ are isomorphic to the set of sequences generated by the ACG $\mathbb{Z}_N$. ■

REFERENCES

[1] J. Kim, J. Lee, J. Kim, and J. Yun, "M2M service platforms: Survey, issues, and enabling technologies," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 61–76, Jan.–Mar. 2014.

[2] S. Lien and K. Chen, "Massive access management for QoS guarantees in 3GPP machine-to-machine communications," *IEEE Commun. Lett.*, vol. 15, no. 3, pp. 311–313, Mar. 2011.

[3] M. Hasan, E. Hossain, and D. Niyato, "Random access for machine-to-machine communication in LTE-advanced networks: Issues and approaches," *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 86–93, Jun. 2013.

[4] S.-Y. Lien, K.-C. Chen, and Y. Lin, "Toward ubiquitous massive accesses in 3GPP machine-to-machine communications," *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 66–74, Apr. 2011.

[5] G. Wu, S. Talwar, K. Johnsson, N. Himayat, and K. Johnson, "M2M: From mobile to embedded internet," *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 36–43, Apr. 2011.

[6] S. Wang, H. Su, H. Hsieh, S. Yeh, and M. Ho, "Random access design for clustered wireless machine to machine networks," in *Proc. IEEE Int. Black Sea Conf. Commun. Netw.*, Jul. 2013, pp. 107–111.

[7] 3GPP, "Study on LTE device to device proximity services, radio aspects (Release 12)," 3GPP, Sophia Antipolis, France, TR 36.843 version 12.0.1, Mar. 2014. [Online]. Available: http://www.3gpp.org/ftp/specs/archive/36_series/36.843.

[8] 3GPP, "Study on scenarios and requirements for next generation access technologies (Release 14)," 3GPP, Sophia Antipolis, France, TR 38.913 version 14.0.0, Oct. 2016. [Online]. Available: http://www.3gpp.org/ftp/specs/archive/38_series/38.913.

[9] W. Dang, M. Tao, H. Mu, and J. Huang, "Subcarrier-pair based resource allocation for cooperative multi-relay OFDM systems," *IEEE Trans. Wireless Commun.*, vol. 9, no. 5, pp. 1640–1649, May 2010.

[10] Q. Song, Z. Ning, Y. Huang, and L. Guo, "A tabu-based optimal sub-carrier allocation algorithm in multi-hop OFDMA relaying network," in *Proc. IEEE Int. Conf. Inf. Commun. Technol.*, Apr. 2013, pp. 381–386.

[11] Q. Song, Y. Huang, Z. Ning, and F. Wang, "Subcarrier allocation in multi-hop orthogonal frequency division multiple access wireless networks," *Elsevier J. Comput. Elect. Eng.*, vol. 40, pp. 599–611, Feb. 2014.

[12] Y. Cao, T. Jiang, and C. Wang, "Cooperative device-to-device communications in cellular networks," *IEEE Wireless Commun. Mag.*, vol. 22, no. 3, pp. 124–129, Jun. 2015.

[13] P. Phunchongharn, E. Hossain, and D. I. Kim, "Resource allocation for device-to-device communications underlaying LTE-advanced networks," *IEEE Wireless Commun. Mag.*, vol. 20, no. 4, pp. 91–100, Aug. 2013.

[14] Q. Ye, M. Al-Shalash, C. Caramanis, and J. G. Andrews, "Distributed resource allocation in device-to-device enhanced cellular networks," *IEEE Trans. Commun.*, vol. 63, no. 2, pp. 441–454, Feb. 2015.

[15] R. Yin, G. Yu, C. Zhong, and Z. Zhang, "Distributed resource allocation for D2D communication underlaying cellular networks," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2013, pp. 138–143.

[16] M. Hasan and E. Hossain, "Distributed resource allocation for relay-aided device-to-device communication under channel uncertainties: A stable matching approach," *IEEE Trans. Commun.*, vol. 63, no. 10, pp. 3882–3897, Oct. 2015.

[17] T. Wang, G. B. Giannakis, and R. Wang, "Smart regenerative relays for link-adaptive cooperative communications," *IEEE Trans. Commun.*, vol. 56, no. 11, pp. 1950–1960, Nov. 2008.

[18] W. Rhee and J. Cioffi, "Increase in capacity of multiuser OFDM system using dynamic subchannel allocation," in *Proc. IEEE Veh. Tech. Conf.*, Tokyo, Japan, May 2000, pp. 1085–1089.

[19] A. Behnad, X. Gao, and X. Wang, "Distributed resource allocation for multihop decode-and-forward relay systems," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4821–4826, Oct. 2015.

[20] E. Paolini, G. Liva, and M. Chiani, "Coded slotted ALOHA: A graph-based method for uncoordinated multiple access," *IEEE Trans. Inf. Theory*, vol. 61, no. 12, pp. 6815–6832, Dec. 2015.

[21] J. Goseling, C. Stefanovic, and P. Popovski, "Sign-compute-resolve for random access," in *Proc. 52nd Annu. Allerton Conf. Commun., Control, Comput.*, Sep. 2014, pp. 675–682.

[22] G. Liva, "Graph-based analysis and optimization of contention resolution diversity slotted ALOHA," *IEEE Trans. Commun.*, vol. 59, no. 2, pp. 477–487, Feb. 2011.

[23] Z. Chang, T. Ristaniemi, and Z. Niu, "Radio resource allocation for collaborative OFDMA relay networks with imperfect channel state information," *IEEE Trans. Wireless Commun.*, vol. 13, no. 5, pp. 2824–2835, May 2014.

[24] C. Koutsimanis and G. Fodor, "A dynamic resource allocation scheme for guaranteed bit rate services in OFDMA networks," in *Proc. IEEE Int. Conf. Commun.*, Beijing, China, May 2008, pp. 2524–2530.

[25] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. Chichester, U.K.: Wiley, 2007.

[26] Y. M. Fouad, R. H. Gohary, and H. Yanikomeroglu, "An autonomous resource block assignment scheme for OFDMA-based relay-assisted cellular networks," *IEEE Trans. Wireless Commun.*, vol. 11, no. 2, pp. 637–647, Feb. 2012.

[27] Z. Kostic and N. Sollenberger, "Performance and implementation of dynamic frequency hopping in limited-bandwidth cellular systems," *IEEE Trans. Wireless Commun.*, vol. 1, no. 1, pp. 28–36, Jan. 2002.

[28] P. Popovski, H. Yomo, S. Aprili, and R. Prasad, "Frequency rolling: A cooperative frequency hopping for mutually interfering WPANS," in *Proc. ACM Int. Symp. Mobile Ad Hoc Ntwk. Comput.*, May 2004, pp. 199–209.

[29] A. A. Shaar and L. Fukshansky, "A new family of one-coincidence sets of sequences with dispersed elements for frequency hopping cdma systems," Jan. 2017. [Online]. Available https://arxiv.org/abs/1701.05209.

[30] M. Mansour, "Optimized architecture for computing Zadoff-Chu sequences with application to LTE," in *Proc. 28th IEEE Conf. Global Telecommun.*, Nov. 2009, pp. 6113–6118.

[31] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation (Release 12)," 3GPP, Sophia Antipolis, France, ETSI TS 36.211 version 12.3.0, Oct. 2014. [Online]. Available: http://www.itu.int/pub/R-REP-M.2135-1-2009.

[32] W. W. Adams and L. J. Goldstein, *Introduction to Number Theory*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1976.

[33] J. A. Gallian, *Contemporary Abstract Algebra*. Boston, MA, USA: Houghton Mifflin, 1998.

[34] S. Roman, *Fundamentals of Group Theory: An Advanced Approach*. Boston, MA, USA: Springer, 2011.

[35] G. E. Andrews, *The Theory of Partitions*. Cambridge, U.K.: Cambridge Univ. Press, 1976.

[36] G. Hahn and G. Sabidussi, *Graph Symmetry: Algebraic Methods and Applications*. Amsterdam, the Netherlands: Kluwer, 1997.

[37] 3GPP, "Further advancements for E-UTRA physical layer aspects (Release 9)," 3GPP, Sophia Antipolis, France, TR 36.814 version 9.0.0, Mar. 2010. [Online]. Available: http://www.3gpp.org/ftp/specs/archive/36_series/36.814.

[38] L. Gilbert and J. Gilbert, *Elements of Modern Algebra*. Boston, MA, USA: Brooks/Cole, 2005.

Yaser M. M. Fouad received the B.Sc. degree in electronics and communications engineering from the Arab Academy for Science and Technology and Maritime Transport, Cairo, Egypt, in 2009, and the M.A.Sc. degree in electrical and computer engineering in 2011 from Carleton University, Ottawa, ON, Canada, where he is currently working toward the Ph.D. degree in electrical and computer engineering.

He is currently working as a Wireless Research Scientist at Intel Labs. His research interests include efficient resource allocation in relay-assisted cellular networks, applications of graph theory in the design of medium access control schemes for multihop networks, and power allocation techniques in multiple antenna and multiuser systems.

Ramy H. Gohary (S'02–M'06) received the B.Eng.(Hons.) degree from Assiut University, Asyut, Egypt, in 1996, the M.Sc. degree from Cairo University, Giza, Egypt, in 2000, and the Ph.D. degree from McMaster University, Hamilton, ON, Canada, in 2006, all in electronics and communications engineering. From 2006 to 2007, he was a Postdoctoral Fellow with McMaster University. He is an Assistant Professor in the Department of Systems and Computer Engineering. In 2008, he was a Visiting Scholar in the Electrical and Computer Engineering Department, of the University of Minnesota, Minneapolis, MN, USA. From 2008 to 2010, he was a Visiting Scientist with the Terrestrial Wireless Systems Branch, Communications Research Centre, Canada. From 2010 to 2012, he was the Project Manager of the Carleton-BlackBerry (formerly Research In Motion (RIM)) research project, and he is currently the Project Manager of the Carleton-Huawei collaborative research project. He has co-authored 80+ IEEE journal and conference papers, and co-supervised 10+ Ph.D. and Master's students. He is the co-inventor of 5 U.S. patents. His research interests include design of embedded systems, applications of embedded systems in machine-to-machine communications, Internet-of-Things, cross-layer design of wireless networks, analysis and design of MIMO wireless communication systems, applications of optimization and geometry in signal processing and communications, information theoretic aspects of multiuser communication systems, and applications of iterative detection and decoding techniques in multiple antenna and multiuser systems. He is also interested in the analysis, design and hardware implementation of computer networks.

He has been the Volunteer Team Co-Chair of the IEEE Vehicular Technology Conference (VTC), Fall 2010, and the Local Arrangement Co-Chair of the IEEE Signal Processing Advances for Wireless Communications (SPAWC), June 2014. He was also a Technical Program Committee Member for: the 2013, 2014, 2015, 2016 IEEE Wireless Communications and Networking Conference (WCNC), the 2014 and 2015 IEEE Conference on Communications (ICC), the 2014 IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), the 2014 IEEE Global Telecommunications Conference (Globecom), the 2015 Spring and Fall IEEE Vehicular Technology Conferences (VTC), and the 2014 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT). He is a member of the College of Reviewers and the Panel Review Board of the Ontario Centre of Excellence (OCE). He received the Natural Sciences and Engineering Research Council visiting fellowship award in 2007. He has been named the "Best Professor" by the Carleton Student Engineering Society for 2015. He is a registered Limited Engineering Licensee (LEL) in the province of Ontario, Canada.

Halim Yanikomeroglu (F'17) was born in Giresun, Turkey, in 1968. He received the B.Sc. degree in electrical and electronics engineering from the Middle East Technical University, Ankara, Turkey, in 1990, the M.A.Sc. degree in electrical engineering (now ECE), and the Ph.D. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 1992 and 1998, respectively.

During 1993–1994, he was with the R&D Group of Marconi Kominikasyon A.S., Ankara, Turkey. Since 1998 he has been with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, where he is now a Full Professor. He spent the 2011–2012 academic year at TOBB University of Economics and Technology, Ankara, Turkey, as a Visiting Professor. His research interests include wireless technologies with a special emphasis on cellular networks. In recent years, his research has been funded by Huawei, Telus, Allen Vanguard, Blackberry, Samsung, Industry Canada, Communications Research Centre of Canada (CRC), DragonWave, Mapsted, and Nortel. This collaborative research resulted in about 25 patents (granted and applied).

He is a Distinguished Lecturer for the IEEE Communications Society (2015–2018) and a Distinguished Speaker for the IEEE Vehicular Technology Society in 5G wireless technologies. He has been involved in the organization of the IEEE Wireless Communications and Networking Conference (WCNC) from its inception in 1998 in various capacities including serving as a Steering Committee member, Executive Committee member and the Technical Program Chair or Co-Chair of WCNC 2004 (Atlanta), WCNC 2008 (Las Vegas), and WCNC 2014 (Istanbul). He was the General Co-Chair of the IEEE 72nd Vehicular Technology Conference (VTC 2010-Fall) held in Ottawa, and is currently serving as the IEEE 86th Vehicular Technology Conference (VTC 2017-Fall) to be held in Toronto. He has served in the editorial boards of the IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and IEEE COMMUNICATIONS SURVEYS & TUTORIALS. He was the Chair of the IEEE's Technical Committee on Personal Communications (now called Wireless Technical Committee). He received the IEEE Ottawa Section Outstanding Educator Award in 2014, Carleton University Faculty Graduate Mentoring Award in 2010, the Carleton University Graduate Students Association Excellence Award in Graduate Teaching in 2010, and the Carleton University Research Achievement Award in 2009. He is a registered Professional Engineer in the province of Ontario, Canada.