An Efficient Greedy-Based Autonomous Resource Block Assignment Scheme for 5G Cellular Networks with Self-Organizing Relaying Terminals

Yaser M. M. Fouad, Ramy H. Gohary, and Halim Yanikomeroglu

***Abstract*—In future cellular networks, self-organizing relaying terminals (RTs) are expected to play a crucial role in assisting the communication between base stations and wireless terminals (WTs), which include, not only active user terminals, but also machine-type communication devices. In the absence of channel quality indicators, the effective utilization of RTs requires a mechanism by which these RTs can assign available resource blocks (RBs) to a potentially large number of WTs with minimal conflicts. This requires optimizing RB assignments over a large set of lengthy sequences, which is computationally prohibitive for networks with large numbers of RTs. To alleviate the difficulty in designing such sequences, we develop a greedy algorithm, whereby pairs of RB assignment sequences are selected in an efficient sequential manner. The performance of the sequences generated by this algorithm is comparable to that of the sequences generated by exhaustive search, but with a significantly less computational cost.**

I. INTRODUCTION

The prospect of utilizing self-organizing relaying terminals (RTs) in cellular networks forebodes a significant increase in the number of wireless terminals (WTs) that can be supported by the network [1]. These WTs may include, not only user terminals, but also machine-type communication devices that are actively engaged in communication with the base station (BS). In particular, using idle user terminals as RTs, will enable replacing weak direct links between the WTs and the BS with cascades of multiple strong ones. This approach is referred to as terminal relaying [1]. A fundamental impediment in applying this approach in practice is that it requires efficient mechanisms for the RTs to allocate the available (time-frequency) resource blocks (RBs) to their assisted WTs.

When instantaneous channel quality indicators (CQIs) are available at the BS, such allocations can be optimally determined. However, in practice, acquiring this information incurs a significant overhead and can seriously infringe on the resources available for communication. This is especially true when the RTs are mobile and their number is large. Hence, to extract the potential gains of terminal relaying, it is desirable for the RBs to be allocated blindly, i.e., without CQIs, and autonomously, i.e., without centralized coordination.

Autonomous RB allocation schemes have been developed for networks in which the CQIs on each RB are assumed available, e.g., [2], [3]. However, the mobility of both WTs and RTs in terminal relaying networks renders this assumption rather difficult to realize in practice. In contrast, in the absence of CQIs, blind allocation schemes that use random RB assignments are available, e.g., [4]. These schemes offer a significant reduction in the overhead necessary to acquire the CQIs, but the lack of coherence in their structure results in, otherwise avoidable, RB allocation conflicts. This drawback is alleviated by the scheme proposed in [5] wherein the RB assignment sequences are endowed with a multiplicative cyclic group (MCG) structure. In [5], this structure was used to obtain cyclically-generated RB assignment sequences and an exhaustive search was performed to select the sequences that minimize the number of RB allocation conflicts between RTs. Despite its advantages, when the number of RTs increases, the cost of exhaustive search over cyclically-generated sequences becomes impractical.

In this paper, we consider a distributed relay-assisted cellular system in which the CQIs are neither available at the RTs nor at the BS. Each RT locally assigns RBs to its incoming WTs according to a prescribed RB assignment sequence; i.e., without centralized coordination by the BS. To facilitate the sequences selection, we develop a greedy algorithm for efficient cyclic generation of RB assignment sequences for networks with large numbers of RTs. In particular, the assignment sequences generated by the proposed algorithm are obtained from distinct group generators and cyclic shifts of the MCG structure. This reduces the quest for designing sequences to the quest for finding group generators and cyclic shifts that result in minimal assignment conflicts. The proposed algorithm sequentially selects the group generator pairs and their associated cyclic shifts. In particular, in each iteration, 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 number of assignment conflicts. Numerical results suggest that the performance of the sequences generated by this algorithm is comparable to that of the sequences generated by exhaustive search, but with a significantly less computational cost. Furthermore, the set of sequences generated by the greedy algorithm does not depend on the number of RTs. In particular, for a system with M RTs, only the first M group generators and their associated cyclic shifts in the look-up table are

The authors are with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada.

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.

utilized for obtaining the assignment sequences. This implies that it suffices for the greedy algorithm to be run once off-line, independently of the number of RTs to be used in the system. This is in contrast with the sequences designed with exhaustive search [5], which is re-run each time the number of RTs in the network changes. To summarize, the proposed greedy algorithm possesses the following features:

- The greedy algorithm yields RB assignment sequences with performance comparable to that of the sequences generated by exhaustive search.
- The computational complexity of the greedy algorithm is polynomial in the number of RBs, and does not depend on the number of RTs, whereas the computational complexity of exhaustive search is exponential in the number of RTs.
- The RB assignment sequences do not need to be updated when the number of RTs in the network changes.

II. Autonomous RB assignments using cyclic sequences

We consider an autonomous RB assignment scheme for the downlink of cellular networks with self-organizing RTs. An RT is selected by an active WT according to the observed signal strength of the RT-WT link. The RT does not have access to the CQIs and, once selected, it allocates one or multiple RBs to the WT, depending on the data rate requirement thereof.

To maintain autonomy, RB assignments can be performed randomly [4] or according to prescribed assignment sequences [5] with entries corresponding to available RBs. In either approach, no coordination with the BS or other RTs is required. This autonomy might result in instances at which one RB is assigned to multiple WTs simultaneously. Occurrence of such an instance, referred to as a 'hit', is likely to result in high interference levels and significantly deteriorates the quality of the received signal of the WTs. Unlike the random approach, which does not offer performance guarantees, using prescribed sequences provides an opportunity for improving performance. This can be done by properly selecting the RB assignment sequences to minimize hit occurrences in the absence of CQIs. Unfortunately, optimizing RB assignments over all possible sequences constitutes a formidable task that, for a network with N RBs, requires exhaustive search over $N!$ sequences. A more practical approach is the one in which the exhaustive search is restricted to cyclically-generated sequences [5], that is, sequences generated from a single element in the sequence.

A subset of such sequences are the pseudonoise (PN) sequences that are commonly used in Frequency Hopping (FH) systems [6]. However, the selection of PN sequences that are 'good' for FH systems is based on the number of collisions in a communication scenario that differs significantly from the RB assignment one. In particular, in FH systems, a WT occupies a particular frequency slot during one dwell interval and relinquishes it after that, whereas in assigning RBs, a WT is assumed to have a full buffer and to retain its assigned RBs during the entire transmission interval.

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

Definition 1: 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 [7]. □

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

Definition 2: 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 [7]. □

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: Given a positive integer N, the Euler Totient function, $\phi(N)$, is defined as the cardinality of the set of integers smaller than N that are coprime with N [8]. □

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 a cyclic sequence with N entries that spans all the elements of $\mathbb{G}$. Hence, to study a cyclically-generated sequence for a network with N RBs, it suffices to consider the order-N cyclic group that generates it. 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 [5], cyclically-generated 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 [5], 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 sequences that are cyclically-generated by the MCG structure incurs a computational complexity that prohibits its utilization in systems with practical numbers of RTs. To overcome this difficulty, we will develop an efficient greedy algorithm in the following section.

III. A greedy-based approach

To facilitate the assignment of RBs in practical terminal relaying scenarios, in this section, we will propose an efficient greedy algorithm in which cyclically-generated sequences are selected sequentially. The performance of this algorithm is comparable to that of exhaustive search, but its computational complexity is significantly less. This property renders the greedy algorithm attractive for practical application in systems in which the number of RTs is too large for exhaustive search to be used.

The implementation of the greedy algorithm relies on a technique for pairing group generators and cyclic shifts that yield cyclically-generated sequences, which proceed in reversed orders. In each iteration of the greedy algorithm, one pair of group generators and the associated cyclic shifts are chosen. This implies that the algorithm exhausts all possible group generator pairs in $\frac{\phi(N)}{2}$ iterations.

After pairing the group generators, the algorithm, in its first iteration, arbitrarily selects a pair, along with a cyclic shift $s = 0$, to be the initial basis of the algorithm. In each of the remaining $\frac{\phi(N)}{2} - 1$ iterations, the algorithm augments its basis by an additional pair and its associated cyclic shifts. After the last iteration, the selected group generator pairs, along with their cyclic shifts, are tabulated in a look-up table according to the order with which they were selected. For a system with M RTs, the M sequences that are cyclically-generated by the first $M/2$ group generator pairs and cyclic shifts in the table are utilized for assigning RBs. In the following sections, we will provide details on the pairing technique as well as the generation and selection 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 optimal cyclically-generated sequences are any two sequences that proceed in reversed orders. In particular, when these two sequences are utilized for autonomous RB assignment, no hits will occur as long as the number of WTs is less than the available RBs. Since each of these sequences is generated by a group generator and a cyclic shift, it is required to find two group generators and their associated cyclic shifts that would generate reversed sequences. For any group generator g and cyclic shift s, there exists a unique inverse (g^{-1}, s^{-1}) that generates the same sequence but in the reversed order, where g^{-1} is the modular arithmetic inverse of g satisfying

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

where e is the group identity element and N is the order of the group. It can be readily shown that the sequences generated by g and g^{-1} proceed in reversed orders. Hence, the cyclic shift assigned to g^{-1}, denoted by s^{-1}, must be in the direction opposite to the one assigned to g. This cyclic shift can be expressed as $s^{-1} = N - s - 1$. For example, if we consider the MCG structure proposed in [5], the cyclically-generated sequences can be obtained by

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

where g_i is the i^{th} generator of the MCG of order N, and s_i is the associated cyclic shift.

B. Sequential Selection of Assignment Sequences

Let $\mathcal{S}$ be the set containing the already selected algorithm basis and $\mathcal{U}$ be the set containing the group generator pairs that have yet to be selected. At first, $\mathcal{S}$ is empty and $\mathcal{U}$ contains the $\frac{\phi(N)}{2}$ group generator pairs. In the first iteration, the greedy algorithm arbitrarily selects a pair of group generators (g, g^{-1}) from $\mathcal{U}$ and a cyclic shift s and moves (g, g^{-1}, s, s^{-1}) to $\mathcal{S}$. Without loss of generality, s can be set equal to 0 and s^{-1} to $N-1$. In each of the subsequent $\frac{\phi(N)}{2} - 1$ iterations, the algorithm examines each of the remaining group generator pairs, i.e., the ones not already in $\mathcal{S}$, for all possible cyclic shifts $s \in \{0, \dots, N-1\}$. In particular, the algorithm evaluates the number of pairwise hits between the sequences generated by each of the remaining group generator pairs and the individual group generator pairs in $\mathcal{S}$ over all possible cyclic shifts. We note that for the algorithm to be able to adapt to both, uniform and non-uniform WT distributions, the metric for choosing the group generator pairs and cyclic shifts must not depend on the instantaneous loads of RTs. Hence, when evaluating the number of pairwise hits for a group generator pair, the algorithm considers all possible distributions of the system load over the RTs.

Since the two sequences generated by (g, g^{-1}, s, s^{-1}) proceed in reversed orders, it can be shown that both sequences incur the same number of pairwise hits with those generated by each of the group generator pairs and cyclic shifts in $\mathcal{S}$. Hence, it suffices to evaluate the number of pairwise hits between the sequences obtained by only one of the group generators of the pair being examined and the ones obtained by the individual pairs in $\mathcal{S}$. This resembles a 3 RT setup, wherein 2 RTs are assigned a pair from $\mathcal{S}$, and the third RT is assigned a group generator from the examined pair.

To evaluate the number of pairwise hits, we associate an $N \times N$ load matrix, X_i, with the i-th RT, which is assumed to have a group generator g_i and a cyclic shift s_i. In particular, the row and column indices of X_i represent the RB and the load level of RT i, respectively. In particular, the (ℓ_1, ℓ_2)-th entry of X_i represents the binary state of RB ℓ_1 when the load level of RT i, is ℓ_2. This entry will be set equal to 1 if RB ℓ_1 is assigned by RT i, when it is loaded with ℓ_2 WTs. Similarly, this entry will be set equal to 0 if RB ℓ_1 is not assigned by RT i, when it is loaded with ℓ_2 WTs. Using this notation, the (ℓ_1, ℓ_2)-th entry of X_i can be written as

$$\left[X_i\right]_{\ell_1,\ell_2} = \begin{cases} 1 \quad \text{if} \quad \ell_1 \in S(\ell_2), \quad \text{and} \\ 0 \quad \text{otherwise}, \end{cases}$$

where $S(\ell_2) = \left\{g_i^{(k+s_i)} \pmod{N+1}\right\}_{k=1}^{\ell_2}$. To elaborate, suppose that the group generator and cyclic shift pairs (g_i, s_i) and (g_j, s_j) yield the assignment sequences $(1,3,2,4)$ and $(3,2,4,1)$ for the i-th and j-th RTs, respectively. The corresponding load matrices, X_i and X_j, are

$$X_i = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \text{and} \quad X_j = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \tag{3}$$

The inner product of the ℓ_1-th column of X_i and the ℓ_2-th column of X_j yields the number of hits when RTs i and j are loaded by ℓ_1 and ℓ_2 WTs, respectively. This product is given by the (ℓ_1, ℓ_2)-th entry of the pairwise hit matrix $H_{i,j}$, where

$$H_{i,j} = X_i^T X_j. \tag{4}$$

We will now show how the $\{H_{i,j}\}$ matrices can be used to evaluate the total number of pairwise hits over all possible load combinations. To do this, we consider a fully loaded system; i.e., a system loaded with N WTs. Let the group generator being examined and its associated cyclic shift be denoted by (g_i, s_i), and let the corresponding load matrix be denoted by X_i. Similarly, let the group generator pair in $\mathcal{S}$ and its associated cyclic shifts be denoted by $(g_j, g_j^{-1}, s_j, s_j^{-1})$ and let the corresponding load matrices be denoted by X_j and $\bar{X}_j$. In this notation, we have used $\bar{X}_j$ to denote the load matrix corresponding to (g_j^{-1}, s_j^{-1}). Utilizing these load matrices, the total number of pairwise hits over all possible load combinations between the sequences of (g_i, s_i) and $(g_j, g_j^{-1}, s_j, s_j^{-1}) \in \mathcal{S}$ can be expressed as

$$V_{i,j} = \sum_{(k_1,k_2,k_3)\in\mathcal{Y}} H_{i,j}(k_1, k_2) + \bar{H}_{i,j}(k_1, k_3), \quad (5)$$

where k_m is the load of RT m, $\mathcal{Y} \triangleq \{(k_1, k_2, k_3)| \sum_{m=1}^{3} k_m = N,\ k_m \in \mathbb{N}\}$, $H_{i,j}$ is defined in (4) and $\bar{H}_{i,j} \triangleq X_i^T \bar{X}_j$. Using (5), the total number of pairwise hits between the sequence generated by (g_i, s_i) and the ones generated by each of the group generator pairs and the cyclic shifts in $\mathcal{S}$ can be expressed as

$$I_i = \sum_{(g_j,g_j^{-1},s_j,s_j^{-1})\in\mathcal{S}} V_{i,j}. \quad (6)$$

After evaluating $\{I_i\}$, the algorithm basis, $\mathcal{S}$, is augmented by the group generator pair and the associated cyclic shifts that yield the minimum total number of pairwise hits I_i. The algorithm continues to iterate until $\mathcal{S}$ contains all the $\frac{\phi(N)}{2}$ group generator pairs and $|\mathcal{U}| = 0$. The selected group generator pairs and their cyclic shifts are then tabulated in a look-up table according to the order with which they were included in $\mathcal{S}$. We note that that order depends solely on the number of pairwise hits, but not on the number of RTs in the system. This implies that it suffices for the greedy algorithm to be run one time only to obtain a look-up table of cyclically-generated sequences for systems with less than $\phi(N)$ RTs. Thus, in contrast with the exhaustive search scheme developed in [5], when a WT switches to the relaying mode, the currently active RTs do not need to update their assignment sequences.

C. *Computational Complexity of The Greedy Algorithm*

We note that only addition operations contribute to the computational cost of evaluating $V_{i,j}$ in (5) and I_i in (6). The number of additions required to evaluate each of the pairwise hit matrices $H_{i,j}$ in (5) is bounded by N^3, and since each iteration of the greedy algorithm considers a setup with 3 RTs, it follows that the cardinality of $\mathcal{Y}$ is equal to the number of ways the 100% load (i.e. $K = N$ WTs) can be partitioned into 3 ordered parts, where K is the total load of the 3 RTs. This number is given by$\binom{N-1}{2}$ [9]. Hence, the number of addition operations required for computing the pairwise number of hits between the selected group generator i and each of the group generator pairs in $\mathcal{S}$ over all possible cyclic shifts $s \in \{0, \ldots, N-1\}$ can be bounded by

$$2N^4+2N\binom{N-1}{2} = 2N^4+N^3-3N^2+2N = \mathcal{O}(N^4). \quad (7)$$

Since the number of group generator pairs is given by $\frac{\phi(N)}{2}$, in each of the $\frac{\phi(N)}{2} - 1$ iterations, the number of group generator pairs in each of the two sets, $\mathcal{U}$ and $\mathcal{S}$, can be bounded by $\frac{\phi(N)}{2} - 1$. Consequently, the number of addition operations required for obtaining the cyclic sequences by the greedy algorithm can be bounded by

$$(2N^4 + N^3 - 3N^2 + 2N)\Big(\frac{\phi(N)}{2} - 1\Big)^3. \quad (8)$$

From (8), it can be seen that the computational complexity of the proposed greedy algorithm is polynomial in N. In particular, since $\phi(N) < N$, the computational complexity of the greedy algorithm is bounded by $\mathcal{O}(N^7)$. This is in contrast with the complexity of the exhaustive search proposed in [5], which was shown to be bounded by

$$\frac{1}{2}M(M-1)N^M\left(N^2(N-1) + \binom{N-1}{M-1}\right). \quad (9)$$

Comparing (8) and (9), it can be seen that the computational complexity of exhaustive search is exponential in the number of RTs, M, whereas the complexity of the greedy algorithm is independent of M.

IV. Simulation Results

In this section, we compare the performance of the uniformly-distributed random assignment sequences in [4], with that of the cyclic sequences generated by the greedy algorithm, and the exhaustive search in [5] at different relative loads $\frac{K}{N}$; i.e., the ratio of the currently assigned RBs to the total number of RBs in the system. For the scheme in [5] and the one proposed herein, the cyclic sequences are generated by using the MCG structure. To compare the greedy algorithm performance with that of exhaustive search, in the forthcoming simulations we will restrict our attention to the case of $M = 3$ RTs; exhaustive search is computationally prohibitive for $M > 3$. To invoke the MCG structure, the number of RBs will be chosen to be $N = P - 1$, where P is a prime number [5].

In the following examples, the performance is measured by the average number of hits over all possible load combinations; i.e., the average over all ordered triples $(k_1, k_2, k_3) \in \{(k_1, k_2, k_3)| \sum_{i=1}^{3} k_i = K, k_i \geq 0\}$ where k_1, k_2, k_3 are the load levels of RTs 1, 2 and 3, respectively and K is the total load of the network.

Example 1: In this example, we compare the performance of the sequences generated by the greedy algorithm and those generated by the exhaustive search in [5] when $N = 16$, and $N = 40$ RBs. We will consider the greedy algorithm with the metric in Section III-B. The performance of the algorithm is depicted in Figure 1.

From Figure 1, it can be seen that the cyclically-generated sequences obtained by both, the exhaustive search and the greedy algorithm, outperform the random assignment sequences. For example, for a system with $N = 40$ RBs at a relative load, $\frac{K}{N} = 90\%$, the average number of hits of the random assignment scheme in [4] and the cyclically-generated sequences obtained by exhaustive search, and the greedy

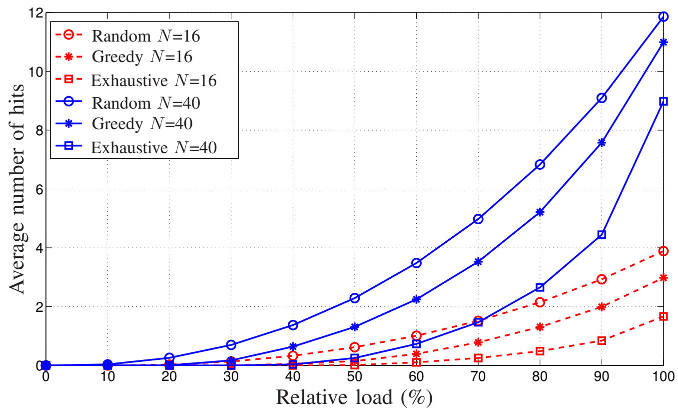

Fig. 1. Comparison between random assignments in [4] and MCG structured assignment with the exhaustive search [5], and the proposed greedy algorithm for $N = 16$ and $N = 40$ RBs.

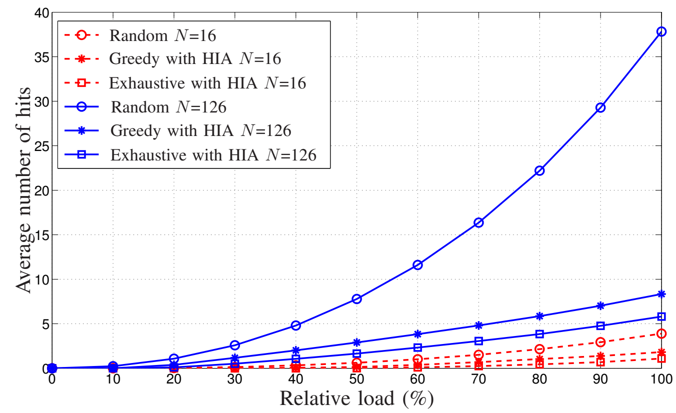

Fig. 2. Comparison between random assignments in [4] and MCG structured assignment utilizing HIA with the exhaustive search [5], and the proposed greedy algorithm for $N = 16$ and $N = 126$ RBs.

algorithm proposed herein are 9.1, 4.4, and 7.6, respectively. Furthermore, it can be seen that the performance of the sequences obtained by the greedy algorithm is comparable to that of those obtained by exhaustive search. For example, in the case of $N = 16$ RBs, and $\frac{K}{N} = 100\%$, the average number of hits observed by the greedy algorithm and the exhaustive search are 2.9 and 1.7, respectively. Despite its potential suboptimality, the greedy algorithm possesses a significantly lower computational complexity when compared with exhaustive search. This renders it suitable for systems with practical numbers of RTs for which exhaustive search is computationally prohibitive. □

Example 2: In this example, we consider a setup similar to the one in Example 1, but with $N = 16$ and $N = 126$. Furthermore, each RT is assumed to know the assignment sequences of the neighbouring RTs and to be able to identify the RT with which it collided once a hit is detected. This is possible, for instance, if RTs use different modulation schemes. Utilizing this information, the RTs identify the RBs already utilized by their neighbouring RTs. Subsequently, when a hit occurs each RT updates its assignment sequence to avoid the RBs already assigned by other RTs. This avoidance technique was proposed in [5] and was referred to therein as the hit identification and avoidance (HIA) algorithm. We note that the HIA algorithm cannot be applied to the randomly generated assignment sequences because those sequences do not possess a specific structure. The performance of the MCG structured sequences generated by exhaustive search, and the proposed greedy algorithm with HIA, along with the random assignments of [4] are depicted in Figure 2.

From this figure, it can be seen that cyclically-generated sequences exhibit a significant performance improvement in the average number of hits when the HIA algorithm is utilized. For example, for the case of $N = 16$ RBs, the average number of hits yielded by the greedy algorithm with HIA when $\frac{K}{N} = 100\%$ is 1.6 in contrast with 2.9 when HIA was not utilized. Simulation results suggest that this performance advantage increases with the number of resource blocks, N. In addition, this figure shows that the HIA algorithm reduces the performance gap between sequences generated by the greedy algorithm, and the exhaustive search. □

V. Conclusion

In this paper, we proposed a greedy algorithm that enables terminal relays to autonomously and efficiently assign RBs to incoming WTs. The proposed algorithm yields a performance comparable to the optimal one yielded by exhaustive search, but with a significant reduction in the computational cost. This renders the proposed algorithm attractive in systems with practical numbers of RTs and WTs. Furthermore, the set of sequences cyclically-generated by the greedy algorithm can accommodate any number of RTs, provided that this number is less than the number of cyclic sequence generators. A key feature of this algorithm is that it automatically accommodates the temporal variations in the number of available RTs in practical terminal relaying scenarios.

Acknowledgment

The authors would like to thank Dr. Gamini Senarath and Dr. Petar Djukic of Huawei Technologies Canada Co., Ltd.

References

[1] T. C.-Y. Ng and W. Yu, “Joint optimization of relay strategies and resource allocations in cooperative cellular networks,” *IEEE J. Select. Areas Commun.*, vol. 25, pp. 328–339, Feb. 2007.

[2] T. Wang, G. B. Giannakis, and R. Wang, “Smart regenerative relays for link-adaptive cooperative communications,” *IEEE Trans. Commun.*, vol. 56, Nov. 2008.

[3] W. Rhee and J. Cioffi, “Increase in capacity of multiuser OFDM system using dynamic subchannel allocation,” in *Proc. IEEE Vehic. Tech. Conf. (VTC)*, (Tokyo), May 2000.

[4] C. Koutsimanis and G. Fodor, “A dynamic resource allocation scheme for guaranteed bit rate services in OFDMA networks,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, (Beijing), 2008.

[5] 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, pp. 637–647, Feb. 2012.

[6] Z. Kostic and N. Sollenberger, “Performance and implementation of dynamic frequency hopping in limited-bandwidth cellular systems,” *IEEE Trans. Wireless Commun.*, vol. 1, pp. 28–36, Jan. 2002.

[7] W. W. Adams and L. J. Goldstein, *Introduction to Number Theory*. New Jersey: Prentice-Hall, Inc., 1976.

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

[9] G. E. Andrews, *The Theory of Partitions*. Cambridge: Cambridge University Press, 1976.