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

Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada

22-25 June 2014 | IEEE International Symposium on Signal Processing Advances in Wireless Communications (SPAWC) 2014 | Toronto, ON, Canada

Introduction

- Signal relaying enhances the wireless link quality between wireless terminals (WTs) and the base station (BS).
- In practical systems, only a small portion of the connected WTs are active.
- Potential gains can be achieved if inactive wireless terminals are allowed to relay the signals of other users and nearby machine type communication (MTC) devices to the BS.

Figure : Terminal relaying example

Previous Work

- A distributed RB assignment scheme that does not depend on channel quality indicators (CQIs) was proposed.
- Each RT utilizes a locally generated RB assignment sequence when allocating its RBs.
- To facilitate their generation, the RB assignment sequences are endowed with a multiplicative cyclic group (MCG) structure.
- The scheme selects the optimal cyclically-generated sequences by performing an exhaustive search over all MCG-structured cyclically-generated RB assignment sequences.

Challenges

- The computational complexity associated with the exhaustive search is exponential in the number of RTs.
- Cannot be implemented in systems with large numbers of WTs; e.g., systems supporting MTC devices.
- When new RTs enter the system, each RT must update its RB assignment sequence.

Preliminaries

- **Cyclic groups**: 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.
- Each group generator yields a cyclically-generated sequence.
- The set of cyclically-generated sequences is enriched by considering cyclically shifted versions thereof.
- Each cyclic sequence can be generated by a group generator and a cyclic shift:

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

where g_i is the i^{th} generator of the MCG of N elements, and s_i is the associated cyclic shift.
- There is no cyclic shift for which the sequences generated by two distinct group generators coincide.

The Greedy Algorithm: Objective

- **Objective:** Generate an ordered look-up table of group generators and cyclic shifts that can be utilized for sequences generation by RTs.
- The order in the look-up table depends on the interference observed by the WTs when the cyclically-generated sequences are used for RB assignment.
- Due to the absence of CQIs, the interference is measured in number of hit occurrences over all possible load combinations, whereby a hit is defined as the event in which an RB is assigned to multiple WTs concurrently.

The Greedy Algorithm: Overview

- The implementation of the greedy algorithm relies on pairing group generators and cyclic shifts such that their cyclically-generated sequences proceed in reversed orders.
- 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; initially $\mathcal{S}$ is empty and $\mathcal{U}$ contains all the group generator pairs.
- In each iteration, the pair that results in the minimum number of hit occurrences with respect to the ones previously selected is moved from $\mathcal{U}$ to $\mathcal{S}$.
- The procedure is repeated until the exhaustion of all pairs in $\mathcal{U}$.
- The pairs are then tabulated in a look-up table according to the order of selection.

The Simplified Greedy Algorithm: Observations

- **Observation:** The RBs located at the beginning of each sequence (the significant RBs) are the most significant contributors to the average number of hit occurrences.
- **Observation:** The graphical representation of cyclic groups is unique and depends only on the number of elements in the cyclic group, whereby each cyclically-generated sequence contributes by a unique pattern that connects all the elements of the group.

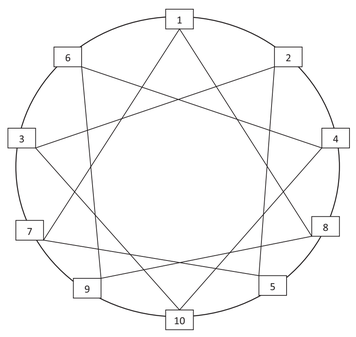


Figure : Graphical representation of a cyclic group of 10 elements

The Simplified Greedy Algorithm: Overview

- Using the same $\mathcal{S}$ and $\mathcal{U}$, in each iteration, the algorithm selects a group generator pair and a cyclic shift.
- This pair is selected such that its pattern results in the minimum number of collisions with the significant RBs of the pairs previously added to $\mathcal{S}$.

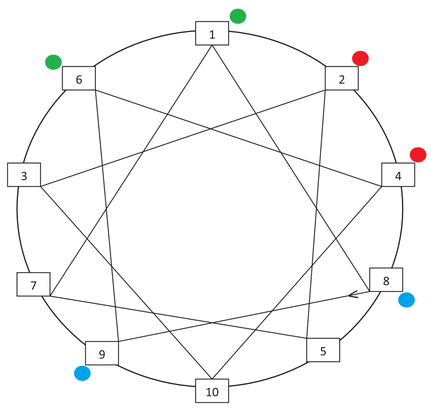
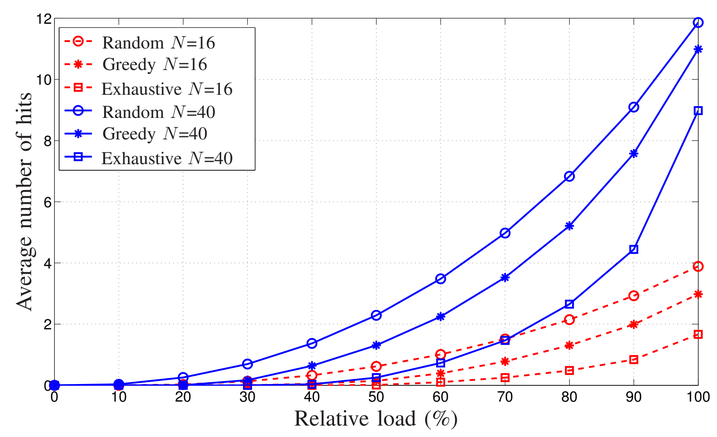


Figure : Significant RBs of the cyclic group of 10 elements

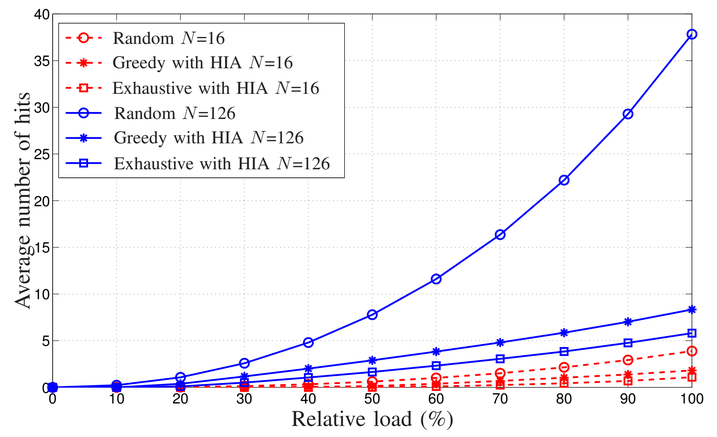
Simulation Setup

- A setup of 3 RTs is considered.
- We compare the performance of the uniformly-distributed random assignment sequences, with that of the cyclic sequences generated by the greedy algorithm, and the exhaustive search 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.
- The performance is measured by the average number of hit occurrences over all possible load combinations.
- In the second simulation, each RT is assumed to know the assignment sequences of its neighbouring RTs and to be able to identify the RT with which it collided once a hit occurs; e.g, RTs use different modulation schemes.
- Subsequently, the RTs identify the RBs already utilized by their neighbouring RTs and accordingly update their assignment sequences to avoid future collisions.
- This avoidance technique was previously proposed in literature and referred to as the hit identification and avoidance (HIA) algorithm.

Simulation Results



Simulation Results



Conclusion

We proposed an efficient greedy algorithm for autonomous RB assignment with the following features:

- Yields RB assignment sequences with performance comparable to that of the sequences generated by exhaustive search.
- Automatically accommodates the temporal variations in the number of available RTs in practical scenarios.
- 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.