

Research Article

Optimized Group Channel Assignment Using Computational Geometry over Wireless Mesh Networks

Anitha Manikandan and Yogesh Palanichamy

Department of Information Science and Technology, College of Engineering, Anna University, Guindy, Chennai, Tamilnadu 600 025, India

Correspondence should be addressed to Anitha Manikandan; animani.kpm@gmail.com

Received 31 March 2015; Accepted 14 June 2015

Academic Editor: Peter Jung

Copyright © 2015 A. Manikandan and Y. Palanichamy. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Wireless Mesh Networks (WMNs) are an evolving division in the field of wireless networks due to their ease of deployment and assured last mile connectivity. It sets out a favorable situation to guarantee the Internet connectivity to all the mobile and static nodes. A wireless environment is dynamic, heterogeneous, and unpredictable as the nodes communicate through the unguided links called channels. The number of nonoverlapping channels available is less than the number of mesh nodes; hence, the same channel will be shared among many nodes. This scarcity of the channels causes interference and degrades the performance of the network. In this paper, we have presented a group based channel assignment method to minimize the interference. We have formulated a mathematical model using Nonlinear Programming (NLP). The objective function defines the channel assignment strategy which eventually reduces the interference. We have adapted the cognitive model of Discrete Particle Swarm Optimization (DPSO), for solving the optimization function. The channel assignment problem is an NP hard problem; hence, we have taken the benefits of a stochastic approach to find a solution that is optimal or near optimal. Finally, we have performed simulations to investigate the efficiency of our proposed work.

1. Introduction

Wireless Mesh Network [1–4] is an infrastructure based multihop network that can be deployed even in underprivileged areas like terrains where providing Internet connectivity is a laborious and expensive task. Deployment of WMNs is undergoing a rapid advancement in the area of networking due to the inherent features of WMNs like self-healing and self-organizing. The WMN infrastructure contains different components in three layers. End user devices like laptops, PCs, and so forth are known as mesh clients (MCs) and the MCs form the bottom most layer. Mesh routers (MRs) that forward the data in multiple hops between the source and the destination form the middle layer. MRs with higher configuration are called Mesh Gateways (MGs). These MGs that render the connectivity to the Internet and other networks for the end users form the top layer. Due to the falling-off price of WLAN equipment (e.g., mesh routers), each device is manufactured with multiple radios to support simultaneous

transmission and reception of data using multiple orthogonal channels. The IEEE 802.11b/g and IEEE 802.11a standards define 3 and 8 nonoverlapping frequency channels available in the legal frequency spectrum. This composes the entire network to act as multiradio multichannel WMN. This characteristic enhances the network throughput [5, 6] by using the increased parallelism in the network and also decreases the transmission delay. However, these advantages also introduce other problems in the network like network partitioning, timing mismatch, distributed operation, routing issues [7], link scheduling, and channel assignment (i.e., mapping of channels to radios at each node) [8].

The key design issue in multiradio multichannel WMN is the assignment of the orthogonal channels, that is, binding the radios with the channels. Channel assignment schemes should be aware of the trade-off between the network connectivity and the interference. Two nodes in the transmission range can communicate with each other if they share one common channel in any of its radios. However, if the same

channel is used in different nodes for communication, it leads to interference. “Interference” is a major setback that occurs in the wireless network where the signals collide and results in packet loss and the end result of this is the performance degradation of the network. On that point there are several solutions suggested in the literature to address these problems [6, 9–12].

The nonoverlapping channels are less in number and hence it becomes necessary to assign the same channel to different radios. Assigning the same channel to different radios results in interference. But this is a fact which cannot be avoided since the number of orthogonal channels cannot be increased beyond a limit that is posed by telecommunication departments of various countries. Hence, it is not possible to avoid interference altogether and it is possible only to minimize the interference. This paper concentrates on how to assign channels in order to minimize the interference. We are invigorated by the job scheduling problem which is an operation research problem that optimally assigns jobs on the shop floor to the machines available to finish the jobs with minimal time consumption [13–15]. We have modelled the channel assignment problem as an optimization problem with the aim of optimally assigning channels to all the links in order to minimize the interference. We have used the nonlinear programming (NLP) technique to formulate our mathematical model. Our channel assignment scheme is a traffic independent algorithm [12, 16, 17]; that is, we do not regard the traffic flow in the network for the channel designation.

The strategy used in this paper has two major steps. The first step clusters the WMN nodes based on their locations using Voronoi diagram [18–20]. We apply the novel idea of membership to select the leader for each cluster or group. Membership of a node refers to the number of groups in which that particular node is present. The node with the maximum value of membership is elected as the leader of that group. The selection of the leaders is responsible for computing the Interference Matrix of that particular group. We have used the protocol model described in [21, 22] to represent the interference in the network. Generally, Conflict Graphs [8, 10, 12, 16] are used to represent the interference present in wireless networks. This research work forms the interference matrices using an enhanced version of the Conflict Graph called Enhanced Multiradio Multichannel Conflict Graph (E-MMCG) [23]. E-MMCG considers the effect of colocated radios assigned with the same channel in the estimation of interference. Besides the novel idea of membership, integrating the protocol model described in [21, 22] and E-MMCG [23] is a novel contribution of this research work.

The second step is a swarm intelligence based approach known as Discrete Particle Swarm Optimization (DPSO) [24] for solving the channel assignment problem in the group. DPSO is a variation of Particle Swarm Optimization (PSO) [25, 26] which is the original version of the approach. The original version was designed to solve continuous problems by Kennedy and Eberhart [27] and was later modified to support discrete problems. We have used the cognitive model of DPSO with the inertia weight to make the particle converge

faster. Applying nonlinear programming (NLP) to model the channel assignment problem as an optimization problem with the objective of minimizing the interference is another novel contribution of this research work.

We make the following contributions to achieve our objective:

- (i) We have employed the Voronoi diagram for clustering the nodes into small subgraphs in order to define the optimization problem with a smaller number of nodes rather than formulating it for the entire network.
- (ii) We have designed a simple algorithm to identify the leaders for each group which has control over the small graph and also it takes the responsibility of assigning the channels to the available links in the small group.
- (iii) We have derived a mathematical model to define the objective of our proposed system and also adapted an evolutionary algorithm approach to solve the optimization problem.
- (iv) The DPSO algorithm is not applied in its original form but rather we have defined the velocity clamping parameter to suit our channel assignment algorithm and also the DPSO employed in this paper does not follow any social parameter since the graph is already divided into small subgraphs where the required information is exchanged and controlled by the leader of that group.

The remainder of the paper is formed as follows. We have discussed the related work in Section 2 and the network architecture in Section 3 and the assumptions made. In Section 4, we have formulated the problem. In Section 5, we have presented our algorithm and we have discussed the performance and comparison in Section 6. Finally, in Section 7 we have concluded the paper.

2. Related Work

There are many solutions proposed for the channel assignment problem by various researchers. But nevertheless it is a significant issue which requires further refinement to the solutions proposed, as it is an NP hard problem [28]. The authors of papers [22, 29] have done an extensive survey of the solutions provided by eminent researchers. We present a brief study done on the channel assignment problem in multiradio multichannel WMNs.

Subramanian et al. proposed a solution for quasistatic channel assignment problem [10]. Channel assignment is performed using a variation of the graph coloring problem. They have developed centralized and distributed algorithms which are based on heuristic search algorithm and greedy approximation algorithm, respectively. They have used a tabu search and Max k cut problems modelled as semidefinite programming and integer linear programming. They have enforced a constraint known as the interface constraint where the number of channels assigned to each node cannot exceed the number of interfaces it has. This approach considers the

traffic load for solving the optimization problem and also preserves the topology. They have obtained a tight lower bound for optimal network interference. But the method proposed in the paper fails to consider the multiple-link case [22].

Avallone and Akyildiz proposed Maxflow-based Channel Assignment and Routing (MCAR) [11], a centralized algorithm that jointly considers the channel assignment and routing. Their proposed heuristic solution aims to maximize the throughput which is modelled as integer linear programming (ILP) as the problem is defined as an NP complete problem. The algorithm is split into link-group binding and group channel assignment. In the first step, links are grouped based on the flows they carry by enforcing a condition that a number of different groups assigned to its links do not exceed the number of radio interfaces. In the second step, the assignment is made to all the links in the group. They have performed extensive simulations and have indicated that their algorithm performs better when compared with Load Aware Channel Assignment (LACA) [5] and Balanced Static Channel Assignment (BSCA) [9] regardless of the traffic demands.

Kyasanur and Vaidya proposed a protocol named Probabilistic Channel Usage Based Channel Assignment (PCUCA) [30] that defines each node to have both fixed and switchable interfaces. The fixed interfaces are employed to obtain the data transmitted by other nodes through fixed channels while the switchable interfaces change over to the fixed channels to transmit the information. Due to the handling of two interfaces, the channel assignment is done in two stages, namely, assigning channels to fixed interfaces and switchable interfaces. This algorithm has been devised to alleviate the channel oscillation problem and the authors also have proved it experimentally. But the algorithm does not provide an optimal solution for the network performance.

Raniwala et al. [5] proposed a joint routing and channel assignment algorithm named Hyacinth. The routing protocol establishes a tree topology with gateway nodes and mesh nodes, and then the channel assignment is carried out based on the topology. Multiple Network Interface Cards (NICs) are equipped in the 802.11 hardware. A set of NICs operates as UP-NICs and a set of NICs work as DOWN-NICs where the former connects to its parent and the latter connects to its children. Then, these interfaces are bound with their neighbors and then the channels are bound to these interfaces. Each node is responsible for binding only its DOWN-NIC with the channel since its UP-NICs are restricted to following its parent. They have used two metrics, namely, the number of links using this channel within the interference range and the aggregate traffic load on this channel from all links using this channel within the interference range.

Marina et al. [16] proposed a traffic independent approach for channel assignment named Connected Low Interference Channel Assignment (CLICA). The authors have proved that the problem is NP complete. They have formulated an ILP for finding the optimum lower bound. A greedy approach is used to assign channels based on priorities assigned to the nodes based on the Depth First Search (DFS) algorithm. Then, the nodes are traversed in the decreasing order to assign

channels. They have compared their algorithm with simple channel assignment scheme and showed that their algorithm is completed in polynomial time. But there are drawbacks in the algorithm as there is some sacrifice in the fairness and computation of unassigned radios which is added overhead.

In recent years, researchers have introduced game theoretical concept for solving the channel assignment problem. We have expounded some papers where authors have applied game theory in this section.

Bezzina et al. proposed an algorithm named Interference-Aware Game Based Channel Assignment Algorithm (IGCA) [31]. They have formulated the problem based on the game theoretic approach. The proposed algorithm aims to minimize the network interference and improves the number of simultaneous connections in the WMN backhaul network. They have modelled the problem as a common interest game where the mesh routers are termed as players. Duarte et al. proposed a near optimal channel assignment by utilizing partially overlapping channels rather depending on the orthogonal channels [32]. Near Optimal Partially Overlapping Channel Assignment (NPOCA) is modelled as a cooperative channel assignment game. Each player formalizes its own utility function based on the strategy it follows. Then, each player negotiates with other players on their nonindependent strategy. The channels are allocated to the radios as per the strategy. The drawback of their system is that the nodes with low probability during the negotiation phase are not identified. This low probability reduces the average performance of their system.

Nezhad and Cerdà-Alabern proposed an algorithm known as Semidynamic and Distributed Channel Assignment (SICA) [33]. This algorithm does not rely on the central node and assumes that nodes do not possess complete information. This algorithm is designed based on the online learner algorithm that assigns best channels to each radio. The nodes gather information during the channel sensing period and utilize this data for strategy formation. The authors have compared SICA and Urban-X [34] and concluded that their algorithm performs better. Yen et al. [35] proposed a two-stage channel allocation algorithm. Their algorithm assigns channels to radios using the game theoretic approach and then assigns the radio-channel pair to a link using a greedy method. Here, the interfaces are modelled rather than the mesh routers themselves. The proposed algorithm aims to minimize the interference. The authors have considered physical model to represent the interference. They have compared their approach with cooperative channel assignment game (CoCAG) [32] and concluded that their algorithm outperforms the latter.

Game theory has become a conjugal method for finding solutions to many problems. But not all of game theory's success stories are like that [36]. The players need to exchange their strategies and come to terms in order to find the solution for the problem. The dynamic nature of WMN makes the application of game theory a complex process. Each time when there is a change in the communication among the nodes the strategy has to be set and discussed. The next problem with game theory approach is that it is a greedy approach, where each player has his own maximization

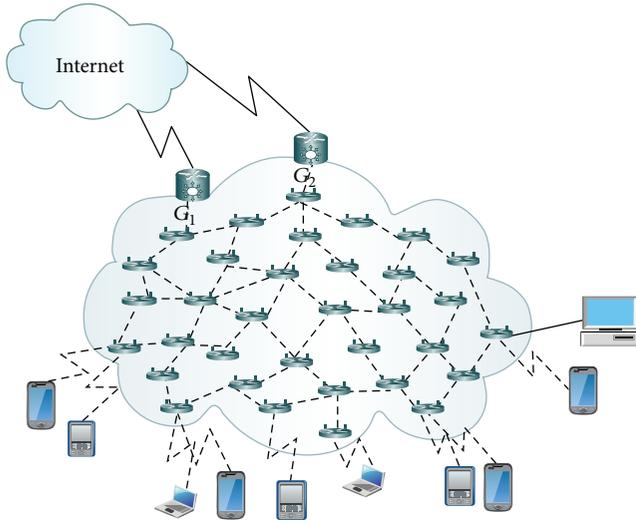


FIGURE 1: Wireless Mesh Network architecture.

goal [37] and tries to achieve his goal irrespective of other participants. The next major issue is providing security. Wireless networks are vulnerable to attacks, and hence game theory may have attackers disguised as players and negotiate their strategy with other participants, where the defender may not recognize that it is conferring with a malicious player. So, game theory requires more security.

Our algorithm considers a group assignment rather than the individual assignment unlike the existing channel assignment algorithms discussed above. We propose a distributed algorithm to avoid burden or bottleneck at one point. We also form groups based on the location, which will not change since the WMN nodes are usually static in nature. We have employed the Hybrid Wireless Mesh Protocol (HWMP) as the routing protocol (default protocol of IEEE 802.11s). Our algorithm is traffic independent; that is, channels are not assigned based on the traffic flow. In the following section, we formally describe our channel assignment method.

3. Network Architecture and Assumptions

This paper considers the WMN architecture shown in Figure 1. In this architecture, it is considered that all the backhaul nodes (middle layer nodes) and gateway nodes are equipped with multiple radios and each radio is capable of supporting multiple orthogonal channels. The number of orthogonal channels is outnumbered by the number of links. This clearly states that interference cannot be avoided in the wireless environment but can be minimized. In the proposed work, the links are grouped (identified by the Link Matrix M_L) so that the number of groups does not exceed the number of available channels of that particular node. The Enhanced Multiradio Multichannel Conflict Graph (E-MMCG) is applied to model the interference, according to the protocol model discussed in [22, 23]. This graph also considers the cochannel interference created by the radios. We have derived a mathematical model based on the job scheduling problem which is an operation research problem

[13]. The proposed optimization model is performed to minimize the interference by proper channel assignment among the nodes in the network. Generally, the channel assignment problem is an NP hard problem. Finding an exact solution to an NP hard problem is a time consuming procedure and sometimes it is not possible. Hence, we propose a heuristic approach for resolving the problem. The DPSO, a variation of the evolutionary algorithm PSO, is employed to solve the optimization problem. The DPSO provides an optimal solution or nearly optimal solution for assigning channels to the communication links.

We have made the following assumptions to define the problem:

- (i) All the nodes in the network have the same transmission range T_r , detection range D_r , and interference range I_r . These three follow the inequality relation $T_r < D_r < I_r$. The Euclidean distance between nodes say u and v should be less than or equal to T_r ; that is, $d(u, v) \leq T_r$.
- (ii) All the nodes are in general position [38, 39] so that no three nodes are collinear and no four nodes fall on the same circle.
- (iii) A node u can communicate with node v only if it shares the same channel for communication and if they are in each other's T_r .
- (iv) There is more than one gateway node (MG) in the network. Each MG is capable of more storage capacity than the nodes in the backhaul network.
- (v) The interference model we use in our work is the two-hop interference model. This model describes that two links interfere if they are sharing a common router [12, 40–42]. When a node is shared by two other nodes, the two links connecting them interfere if the same channel is assigned.
- (vi) All the nodes need not have the same number of radios and support the same number of channels.

4. Problem Definition

The network architecture is depicted as an undirected graph $G(V, E)$ where V is the set of vertices of the graph defining the nodes and E is the set of edges defining the link between the nodes of the network architecture. The number of links in each node is greater than the number of available orthogonal channels; hence, interference is present. The interference in the network is represented as a protocol model according to [21]. According to the protocol model, two links say $a \leftrightarrow b$ and $c \leftrightarrow d$ interfere if the Euclidean distances $(d(a, c), d(a, d), d(b, c), d(b, d))$ are less than the transmission range T_r and $a, b, c, d \in V$ and $a \leftrightarrow b$ and $c \leftrightarrow d \in E$. We apply the Conflict Graph to model the interference. The graph is plotted in a Euclidean space \mathbb{R} where each node in the graph knows its 2D coordinate position (x, y) . With these coordinates positions each pair of nodes is capable of finding its Euclidean distance d . The Voronoi neighbors are derived based on this Euclidean distance d and the Voronoi

diagram $V_D(G)$. The Delaunay Triangulation $D_T(G)$ gives the Connectivity Graph C_G among the neighbours which is a subgraph of $G(V, E)$. Then two matrices are constructed: one is the Link Matrix M_L which describes the interfering links and the second is the Interference Matrix I based on each node's assignment of channels to each link with respect to its radio. Then, the NLP model is formulated as shown in

$$\min \sum_{j=1}^C \sum_{i=1}^L (M_L * I)_{ij} C_{ij} \quad (1)$$

$$\text{s.t. } \sum_{i=1}^L C_{ij} \leq m_j, \quad j = 1, 2, \dots, C \quad (2)$$

$$\sum_{i=1}^L C_{ij} = 1 \quad j = 1, 2, \dots, C \quad (3)$$

$$M_L = \begin{cases} 1, & \text{for incident links} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

$$I_{lc} \geq 0 \quad \forall 0 < l \leq L, 0 < c \leq C, \quad (5)$$

where $M_L \Leftrightarrow$ the incidence Boolean square matrix of size $L \times L$, represented using Conflict Graph. $L \Leftrightarrow$ set of interfering links present in Conflict Graph C_C . $C \Leftrightarrow$ set of orthogonal channels. $I \Leftrightarrow$ Interference Matrix of size $L \times C$, computed using triangulation property (Thales theorem). $m_j \Leftrightarrow$ the maximum number of k -orthogonal channels available depending on the standard. $C_{ij} \Leftrightarrow$ the channel assignment matrix of size $L \times C$; $I_{lc} \Leftrightarrow$ the value in Interference Matrix I in row l and column c .

This problem is similar to the optimization problem in standard job assignment problem or a weapon target assignment problem in operation research. The assignment problem is usually an NP hard problem for which we provide a swarm intelligence approach to obtain an optimal or near optimal solution.

The mathematical model formulated consists of an objective function as in (1). The value of this function defines the optimal allocation of channels to links. The goal is to reduce the number of interfering links which subsequently reduces the interference in the network and hence the objective function is defined as a minimization function. The function consists of a binary matrix M_L that describes the interfering links for any $l_i \in L$, where L stands for the set of vertices (which we later term as "link vertices") in the Conflict Graph C_C . The information of interfering links can be obtained from the Conflict Graph C_C as shown in Figure 3. The Interference Matrix I is computed based on the *Thales* theorem. The leader consolidates the number of times each channel $c_j \in C$ is assigned to a link while applying the triangulation property. Here, C represents the set of orthogonal channels available in the network. Multiplication of the Link Matrix and the Interference Matrix gives the matrix of size $L \times C$. This matrix demonstrates the number of channels assigned to a particular link and the number of times the same channel is assigned to that link. The matrix C_{ij} is the channel assignment solution matrix obtained from the particle's position matrix

$(X_k^{t+1}(i, j))$ with binary values, constructed by the DPSO algorithm for each iteration. Initially the values of C_{ij} represented as $(X_k^t(i, j))$ are randomly assigned and then updated for each iteration based on the velocity matrix $V_k^{(t+1)}(i, j)$. For each channel $c_j \in C$ assigned for each link $l_i \in L$, the value of the objective function is computed. If the current value is less than the previous value then the current solution is considered as the optimal solution.

Equations from (2)–(4) are the constraints that must be satisfied for a solution to be a feasible solution. The channels assigned for the links should not exceed the maximum number of orthogonal channels represented as m_j as shown in (2). The next condition in (3) states that each link must be assigned with a single channel. The matrix M_L is assigned with binary values based on the Conflict Graph C_C as shown in (4).

The construction algorithms of matrices and the solution are elaborately explained in the following sections.

5. Computational Geometric Based Channel Assignment

This section formally describes the proposed channel assignment scheme known as Computational Geometric Based Channel Assignment (CGCA). This method has been devised for the channel assignment problem in order to minimize the interference and hence increase the throughput of the network. The goal is to assign channels to each radio of the nodes in the backhaul network that will participate in the data communication. Hybrid Wireless Mesh Protocol (HWMP) has been used in proactive mode as the routing protocol [4]. Minimization problems can be viewed as a two-entry ordered tuple [43], where one entry provides a set of feasible solution S to the problem and the other entry is the objective function f of the problem. Then, for each solution say $s \in S$ the objective function value $f(s)$ is computed. The value $f(s)$ is compared with the previous value say $f(s')$ where s and $s' \in S$. If $f(s) < f(s')$ then the feasible solution $s \in S$ becomes the current optimal solution for the problem. The best solution for the objective function is obtained when the value of f does not change for further iterations. We propose an algorithm that works in two phases:

- (i) The first phase identifies the neighboring nodes using computational geometric method. The Voronoi diagram has been applied to identify the neighboring nodes in the transmission range.
- (ii) The second phase assigns channels to the radios of the nodes participating in data communication. To solve the channel assignment problem, the problem is represented as a nonlinear programming (NLP) model and the solution is obtained using the stochastic approach called the Discrete Particle Swarm Optimization (DPSO).

In the first phase, each node identifies its neighbors with the help of Voronoi diagram. Then, each node sends its group information like the number of members in the group and the nodes in the group to Mesh Gateway (MG). A node

may be present in more than one group which is used as its membership number. A node having maximum number of memberships (i.e., a node present in a number of groups) is considered to be the leader of that group. Each leader is responsible for assigning and updating the channel information to the gateway. These small groups are considered for the channel assignment. A distributed channel assignment strategy is proposed as an NLP model where the objective function is defined as the minimization function. The goal of our programming model is to minimize the number of interfering links which subsequently reduces the interference. The swarm intelligence approach has been employed to solve the nonlinear equations. A detailed explanation is given in the subsequent sections.

5.1. Neighborhood Discovery and Leader Selection. The network is split into small subgraphs to think locally and act locally. To split the network into small subgraphs, Voronoi diagram is used. The Voronoi diagram is applied to the graph to find the neighbor nodes of any node $n \in N \wedge n \in G$. To compute the Voronoi diagram, the mesh routers are viewed and plotted as points in a planar graph. Voronoi diagram partitions the plane and the neighboring nodes are identified by the Voronoi edges which are shared with the neighboring nodes. This clustering is used for making the channel assignment distributed and localized. As a result, the convergence is much faster than assigning channels to the entire network.

We assume that each node knows its location in the space using GPS [44] and broadcasts its position to its neighbors. If the nodes know their position, then the distance between two nodes can be computed. The nodes are assumed to be in general positions so that no three nodes are collinear and no four nodes fall on the same circle [38, 39]. Each mesh router (MR) computes its own Voronoi cell and the entire Voronoi diagram delivers a graph with the neighbors close to each node.

Each node registers its neighborhood group with the gateway node MG. The gateway consolidates these clusters to identify some landmark nodes called leaders based on the occurrence of the node in the groups. That is, a node occurring in maximum number of groups is polled as the leader. These leaders take up the responsibility of assigning channels and transmitting the channel information to gateway MG. MG acts as the central server for storing the information so as to avoid packet loss and to improve the throughput of the network as it is the point which encounters maximum amount of traffic throughout the communication. This safeguards the data from signal degradation due to the interference present in the network.

Figure 2 explains the leader selection process for more clarity. The circles in the diagram represent the routers of the network. The node in red color is the special node called the gateway. The groups created by the node are named G_i where $i = 1, 2, \dots, N$ and N is the number of nodes. The dotted lines around the nodes denote the group of each node created. For example, we have shown two groups created for node 1 (G_1) and node 2 (G_2). The nodes in groups G_1

and G_2 are computed as $G_1 = \{n_0, n_1, n_2, n_6\}$ and $G_2 = \{n_1, n_2, n_3, n_5\}$ by applying the algorithm in [18]. From this, the membership for nodes n_0, n_1, n_2 , and n_6 is computed as 2, 1, 2, and 1, respectively, by the gateway node. All the nodes within the transmission range are considered to be neighbors of a particular node. Here, for simplicity, we have grouped only the one hop neighbors of nodes 1 and 2. The Voronoi diagram of group G_2 is shown in Figure 2(b). The thick lines are the bisectors between the points and the thin lines are the Delaunay Triangulation edges.

Each node $n_j \in N$, where $j = 0, 1, 2, \dots, N$, is responsible for computing its group G_i , where $0 \leq i \leq |G|$. $G = \{G_0, G_1, \dots, G_N\}$ is the set of groups in which the subscripts represent the nodes for which the group is formed; for example, G_0 is the group for the gateway node. The groups present in G are not disjoint; that is, a node may be present in more than one group. After the groups are registered, the gateway looks for the prospective leaders by checking the membership of the nodes present in the group. We determine the membership of each node by counting the number of times it is present in G . The maximum occurrence of a node n_j is computed using (6). Each time when a node n_j appears in any group $G_i \in G$ the membership $M(n_j)$ is incremented by 1. Then, MG sends the leader information to all the groups formed in the network:

$$M(n_j) = \sum_{G_i \in G \wedge n_j \in G_i} 1. \quad (6)$$

For example, we have taken group G_2 and assumed its leader to be node 2 as the node belongs to G_1 and G_2 . In this leader selection process, some groups may not have leaders since all the nodes $n_j \in G_i$; they may be the leader of some other group. Thus, this group does not require any leader for itself.

Figure 2(b) shows the connectivity among the nodes that fall in the group formed by node 2. The Delaunay Triangulation (DT) edges show the nodes which share the same Voronoi edge and they are termed to be neighbors. The DT edges provide the connectivity between the nodes and the graph formed is called the Connectivity Graph C_G . Then, the channels are assigned to the radios of the nodes in these small groups $\{G_1, G_2, \dots, G_N\}$ independently.

Algorithm 1 explains the selection of a leader in each group which is carried out by the MG node after the nodes register their groups with it.

The next step of our channel assignment procedure is to construct the link and interference matrices based on the protocol model to define the interference. Our process of modelling the interference is inspired by the E-MMCG algorithm [23] which models the radios as nodes instead of the routers themselves and also considers the cochannel interference. An elaborate description is given in the next section.

5.2. Channel Assignment and Matrices Construction. The Link Matrix M_L and Interference Matrix I are constructed based on the Connectivity Graph C_G , which is created by

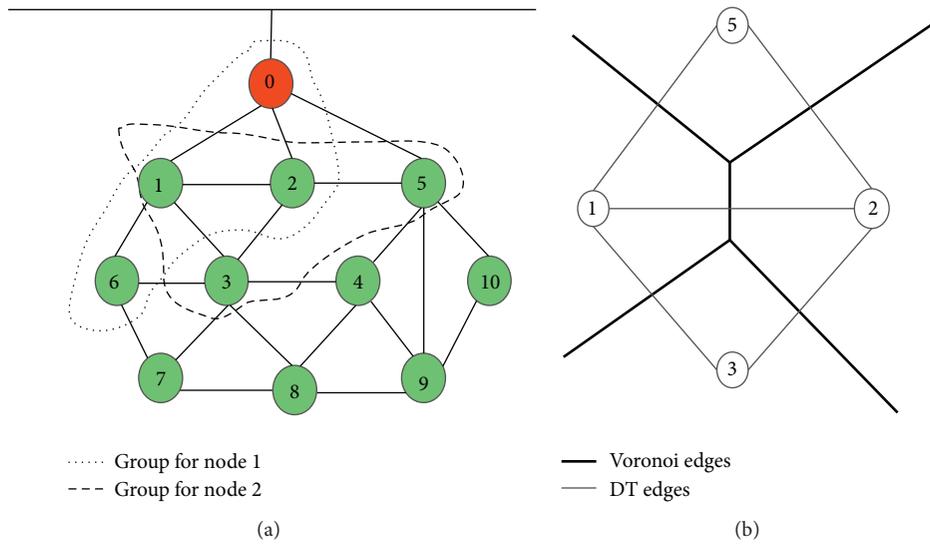


FIGURE 2: (a) Leader selection. (b) Voronoi diagram for node 2 (G_2).

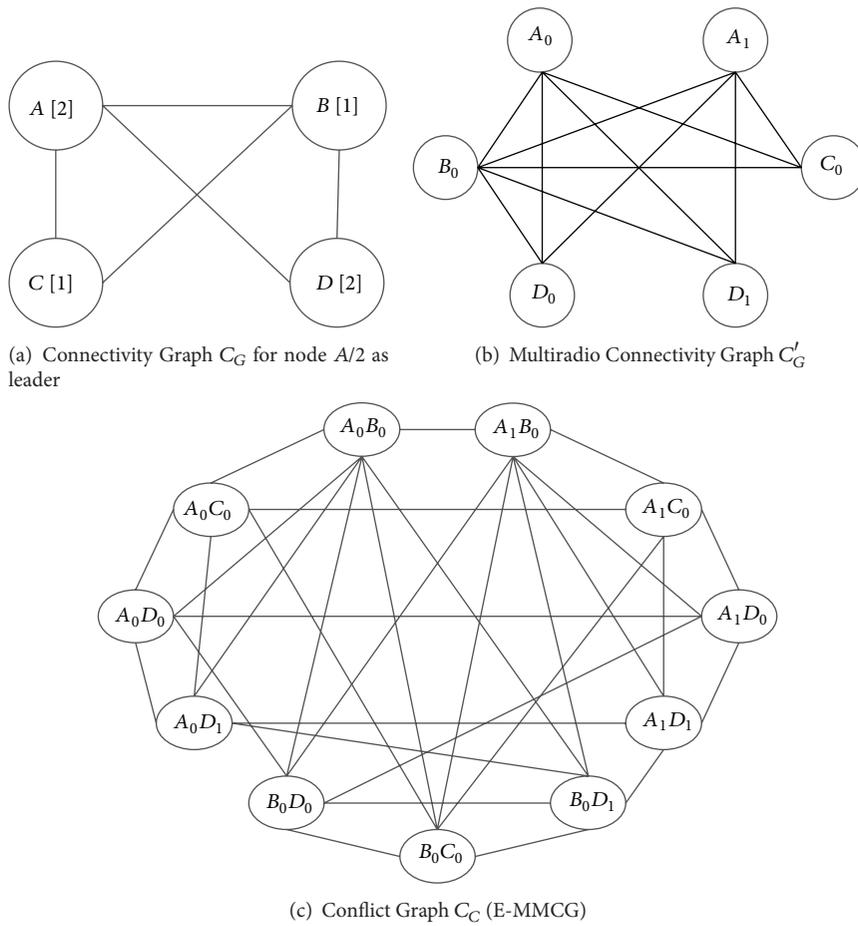


FIGURE 3: Interference model.

```

Let  $G$  be the groups created by  $N$  nodes and registered with gateway MG  $G = \{G_1, G_2, \dots, G_N\}$ 
For each node  $n_j \in N$ 
  For each Group  $G_i = \{n_1, n_2, \dots, n_j\}$ 
     $k = 0$ 
     $L = \phi$ 
    if  $(\exists n_j \in N: n_j \text{ in } G_i)$ 
       $k = k + 1$ 
       $M(n_j) = \max(k, 1)$ 
    End for
     $L(G_i) = \max(M(n_1), M(n_2), \dots, M(n_j)), \forall j \leq i$ 
     $L = L \cup L(G_i)$ 
  End for

```

ALGORITHM 1: Leader selection.

applying the Delaunay Triangulation (DT) in the Voronoi diagram. The DT connectivity gives the possible neighbors of any node to form a group. Sections 5.2.1, 5.2.2, and 5.2.3 in this section explain the interference model and the steps involved in constructing the link M_L and interference I matrices.

5.2.1. Interference Model. We have employed E-MMCG [23] to model the interference effect. We have redrawn the graph in Figure 2(b), the Connectivity Graph C_G with the alphabets, to distinguish it from the number of radios that each node contains (2-A, 1-B, 3-C, and 5-D) as in Figure 3(a). Figure 3(a) shows the subtopology generated after identifying the neighbors and the leader of that particular group. The circles represent the node and the number in the square brackets inside each circle represents the number of radios incorporated in that particular node. We represent the topology for group G_2 as graph $C_G(V_G, E_G)$, where V_G and E_G are the vertices and edges of the Connectivity Graph, respectively, for node 2 (A). Each node is incorporated with different number of radios which is represented in the square brackets.

An intermediate graph known as the Multiradio Connectivity Graph (MCG) $C'_G(V'_G, E'_G)$ is built, where the nodes in C_G are replaced with their radios and the links are established between the radios as the edges as shown in Figure 3(b). Graph C'_G provides the information of interfering radios and is used in constructing the MMCG graph C_C . Algorithm 2 shown below explains the procedure for converting $C_G \rightarrow C'_G$.

Each node in C_G is replaced with its radios; for example, in Figure 3, node A is replaced by its radios A_0 and A_1 and the edges are established between the radios. The Multiradio Connectivity Graph is used to create MMCG graph to model the interference and we have used the E-MMCG algorithm [23] to obtain the Conflict Graph named $C_C(V_C, E_C)$ as shown in Figure 3(c). V_C is the set of vertices named “link vertices” to differentiate it from the set of vertices in V'_G . It represents the link present between two radios/nodes in C'_G . E_C is the set of edges named “link edges” which is established between two link vertices if the interference subsists. The E-MMCG shown in Figure 3(c) includes the cochannel interference; that is, the radios present in a node may use the same channel which

```

Input: Connectivity Graph
Output: Multi-radio Connectivity Graph
Let  $V_G = \{u_1, u_2, \dots, u_g\}$ 
Each node  $u_g \in V_G$  consists of radios  $R_i \mid i = \{1, 2, \dots, r\}$ 
 $\forall V_G \in C_G$ 
   $\forall R_i \in u_g$ 
    Create nodes  $V'_G$ 
  End for
End for
 $\forall (u_i, u_j) \in V_G \wedge d(u_i, u_j) \leq T_r \wedge i \neq j // 1 \leq i, j \leq g$ 
  if  $(R_i \in u_i \wedge R_j \in u_j)$ 
    Create links  $\forall R_i$  and  $R_j$ 
  End if
End for

```

ALGORITHM 2: Multiradio Connectivity Graph ($C'_G(V'_G, E'_G)$) \vdash Connectivity Graph ($C_G(V_G, E_G)$).

results in interference. Using the two graphs C'_G and C_C , the link and interference matrices are constructed.

5.2.2. Link Matrix. Link Matrix M_L is a $L \times L$ matrix where $L \subseteq V_C \wedge V_C \in C_C$. The Link Matrix explains the number of links that interfere with a particular link. Construction of M_L is done using Algorithm 3 as described below. The degree of each link vertex belonging to C_C is computed using

$$D(v_C) = \sum_{k=2}^L 1 \mid \exists 1 \text{ in } M_L(j, k), \quad (7)$$

$$\text{where } j = 1, 2, \dots, L \bigwedge \forall v_c \in V_C.$$

The degree of each “link vertex” is incremented by 1 for every link incident on that particular “link vertex” $v_c \in V_C$ present in the Conflict Graph C_C . The parameter L is the number of “link vertices” in the Conflict Graph.

5.2.3. Channel Assignment and Interference Matrix. Interference matrix I is constructed by consolidating the number of times the same channel is used by the radios. First, we present

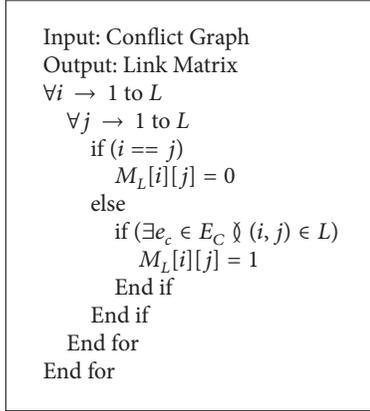
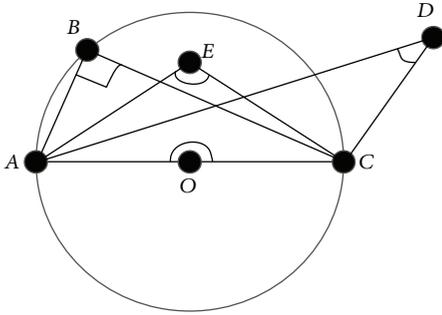

 ALGORITHM 3: Link Matrix M_L .


FIGURE 4: Thales theorem.

the mathematical background used to construct the matrix. Then, using this mathematical background, the steps involved in constructing Interference Matrix are described.

Mathematical Background. We have used Thales theorem and associated triangulation property for our derivation of the Interference Matrix I .

Theorem 1. *The triangulation property says that a circumcircle drawn with respect to the longer side of a triangle enforces three cases.*

Case 1. $\angle AOC = 2 \times \angle ABC$ | Points A , B , and C lie on the perimeter of the circle with O as centre.

Case 2. $\angle AEC > 90^\circ$ | Point E lies inside the circle.

Case 3. $\angle ADC < 90^\circ$ | Point D lies outside the circle.

Figure 4 is used to explain the theorem and all the three cases. The points on a circle with one chord \overline{AC} as its diameter provide the relationship with angles and the point location. The theorem states that the angle created by the points on the perimeter of the circle is always a 90° regardless of the point location; that is, $\angle ABC = 90^\circ$.

The other property related to triangulation states that the largest angle is always opposite to the largest side of the triangle. With reference to Figure 4, in $\triangle AEC$ $\angle AEC > 90^\circ$,

TABLE 1: Skewness factor.

Value of skewness	Quality of the \triangle
$(S = \pm 1)$	Degenerate
$0 < S < \pm 1$	Bad
$S = 0$	Excellent

and hence the largest side is \overline{AC} . This property implicitly states that the sides holding the largest angle \overline{AE} and \overline{CE} come close to the largest side of the triangle \overline{AC} . This property is used to identify the edges that are close to each other. To determine and quantify this closeness factor of the edges we compute skewness S for the triangles using

$$\text{Skewness } (S) = \frac{\text{Optimal triangle size} - \text{Actual triangle size}}{\text{Optimal triangle size}} \quad (8)$$

The optimal triangle is the equilateral/equiangular triangle [45]. Both the optimal triangle and the actual triangle are computed as in

$$\Delta \text{ size} = \frac{1}{2} \times \det \begin{vmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{vmatrix} \quad (9)$$

Using these properties, we group the links and the channels are assigned to the links by the radios based on the skewness value. Table 1 shows the quality of the triangle based on the skewness [45]. For simplicity, we consider only three cases. First case ($S = \pm 1$) happens when the triangle almost flattens and appears to be in parallel with the diameter of the circle, which can never be used (general position avoiding collinear coordinates). The interpretation of the negative skewness factor S is slightly different. In $+S$, the sides move close to the largest side of the \triangle (Figure 4; for $\triangle AEC$ line segments \overline{AE} and \overline{CE} are coming close to \overline{AC}), but in $-S$, line segment \overline{AD} moves closer to line segments \overline{AC} and \overline{CD} for $\triangle ADC$. When $S = 0.9$ to 0.1 , we consider it to be bad as this case leads to sides coming closer to the diameter. The last case where $S = 0$ represents the optimal size of the triangle.

The triangulation property is applied to the Multiradio Connectivity Graph C'_G . These chords explained in Figure 4 are analogous to the links of the graph C'_G . Steps involved in grouping the links and assigning channels to the links are listed as follows.

Step 1. Identify the link vertex $v_c \in C_C$ for each radio $v'_G \in V'_G$ of the leader node.

Step 2. Compute the degree of v_c using (7).

Step 3. Identify the corresponding radios in C'_G for the incident links equal to $D(v_c)$.

Step 4. Check whether triangles are formed by the identified radios in C'_G .

```

Leaders in the graph  $LDR_m$ , where  $m = \{1, 2, \dots, m\}$ ,  $m < N$ 
Number of radios per node  $R_r$ , where  $r = \{1, 2, \dots, r\}$  //  $r$  depends on the product
Number of  $k$ -orthogonal channels  $C$ , where  $C = \{1, 2, \dots, k\}$  // depends on the frequency spectrum of the radios
 $R_{INL} = \emptyset$  // associated radios of incident links in  $C'_G$ 
 $I[|V_C|][Channels] = 0$  // rows = link vertices in  $C_C$  and columns = available orthogonal channels
Input: Multi-radio Connectivity Graph ( $C'_G$ ) and E-MMCG ( $C_C$ )
Output: Interference Matrix  $I$ 
 $\forall LDR_m$ 
   $\forall R_{ldr}$ 
    Identify its corresponding link vertex  $v_c \in V_C$  in  $C_C$  // arbitrary selection
     $d = D(v_c)$  // using (7)
    INL = set of link vertices incident to  $v_c$ 
     $\forall INL \in C'_G$ 
       $R_{INL} =$  Identify the associated radios in  $C'_G$ 
       $\bigcup_1^d R_{INL}$ 
    End for
    Identify the triangles formed with other radios  $r \in R_{INL}$  in  $C'_G$  with reference to  $R_{ldr}$ 
     $\forall \Delta$ 
      Check the necessary condition // using Thales theorem
      Compute skewness  $S$  // using (8)
       $\Delta_{sides} =$  sides of the triangle
       $\forall R_{INL}$ 
        if ( $S == 0$ ) // Assign same channel to all the links
           $\forall i \rightarrow 1$  to  $|V_C|$ 
             $\forall side \in \Delta_{sides} \wedge side \in V_C$ 
               $I[i][j] = I[i][j] + 1$  //  $j$  may be assigned any arbitrary value between 1 to  $|C|$ 
            End for
          End for
        elseif ( $0 < S < 1$ )
           $\forall i \rightarrow 1$  to  $|V_C|$ 
             $\forall side \in \Delta_{sides} \wedge side \in V_C$ 
               $j = k$ 
               $I[i][j] = I[i][j] + 1$ 
               $k = k + 1$ 
            End for
          End for
        End if
      End for
    End for
  End for
End for

```

ALGORITHM 4: Interference Matrix I construction.

Step 5. Compute the skewness of the triangles.

Step 6. Based on the skewness value, assign channels to the links.

Step 7. If the skewness $S = 0$, then the leader assigns same channel to all the links as the triangle formed is an equilateral triangle.

Step 8. If the skewness $0.1 > S > 0.9$, then each radio assigns channel independently to all its links that form the triangle and sends the information to its leader.

Step 9. Leader consolidates the number of times the same channel is used for that particular link as the Interference Matrix.

Algorithm 4 explains the detailed procedure for constructing the Interference Matrix based on the steps described above.

Notations section explains the notations used in our algorithm.

5.3. Discrete Particle Swarm Optimization. Particle Swarm Optimization (PSO) is a stochastic approach based on search algorithms to solve NP hard problem. Generally heuristic algorithms are employed to solve problems, where finding an exact solution is time consuming and sometimes not possible. Originally, the PSO was developed to solve a continuous problem, and then the discrete version was introduced by Kennedy and Eberhart [46]. PSO is used to solve the multidimensional problem and the convergence

time taken is much less and the number of parameters to be tuned is minimal when compared with other evolutionary algorithms. The fundamental notion of PSO is the creation of a particle swarm and each particle provides a feasible solution for the problem. The solutions are evaluated based on some fitness function. There are some standard fitness functions, but in our proposed work, the fitness function is the objective function defined by the NLP model as in (1). During successive iterations, the algorithm calculates the objective function value for the feasible solution and assesses the fitness of the solution. After assessing the fitness and based on the optimization problem, the velocity and the position of a particle that are represented in (10) and (11) are updated. The velocity parameter reflects the experimental knowledge and socially exchanged information among the neighborhood:

$$V_{id}(t+1) = V_{id}(t) + \varphi_1 r_1 (P_{id} - X_{id}) + \varphi_2 r_2 (P_{gd} - X_{id}), \quad (10)$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1). \quad (11)$$

In (10), there are three terms; the first term defines the momentum of the particle to retain its current direction itself. Equation (10) does not have the inertia weight constant in the original velocity parameter definition. The later version consists of an explicit inertia weight parameter (ω) which ranges from 0.8 to 1.2 defined in previous studies [47]. This inertia weight ω reduces the trade-off between the exploitation and exploration of the particle. The second term and the third term are called self-knowledge and social knowledge, respectively. The self-knowledge term is used to know how much the current position is close to its own best position and the social knowledge is used to know how much the current solution is close to its best neighborhood solution l_b or the global solution g_b . φ_1 and φ_2 are called the cognitive coefficient and social coefficient, respectively. The technical study done on the parameter of the PSO provides information on selecting the value for the positive constants $[\varphi_1, \varphi_2]$. Generally, the value of these two constants is 2 and their range is between 0 and 4. Utilizing the inertia weight and the two constants φ_1 and φ_2 together is a straight forward method in selecting the values for the parameters. Inequality (12) yields the relation between the three parameters as in [48]

$$\omega > \frac{1}{2}(\varphi_1 + \varphi_2) - 1. \quad (12)$$

This condition is used to avoid the cyclic behavior of the particles in the search space. There are different velocity models with the aim of providing variation in PSO. The velocity model we have employed in our work is the cognitive-only velocity model [48] with inertia weight. The social component is not considered in this model. The behavior of the particles tends to search locally rather than exploring globally. This behavior is applied to our local graph to assign channels locally. Since this model is slightly slower, to make it converge faster we have introduced the inertia weight to the velocity equation which is statically initialized and checked for the condition given in (12).

The topology of the swarm in PSO has an impact on the performance of the evolution process. There are different ways of communication done among the neighborhood inside a swarm. We have considered a star topology for the communication among the neighbors, since we have employed a leader based clustering to obtain the neighborhood. In our star topology, the central is selected based on the membership. In the proposed work, each radio in the leader suggests some feasible solution for assigning channels to the links in Algorithm 3. The leader then combines all the suggested solutions and the total impact of the assignment is fortified as the Interference Matrix I . Thus, there is no further requirement of the neighborhood particles n_b . Hence, the current particle position is initialized by the leader node which influences all the neighbor nodes. We use this model, since our search area is within the neighbors with a leader identified by the Voronoi diagram. The velocity equation of the cognitive model used in this work is represented in

$$V_k^{(t+1)}(i, j) = \omega V_k^t(i, j) + \varphi_1 r_1 (Pb_k^t(i, j) - X_k^t(i, j)). \quad (13)$$

In (13), $V_k^t(i, j)$ is the element in i th row (links) and j th column (channels) of the k th velocity matrix at time $t + 1$. The particle is of D dimension and moves around the D dimension search space with a velocity parameter and each particle gives a feasible solution. The velocity clamping is used to restrict the particles movement. The velocity parameter is defined with D dimension in the range of $V_{\max} \rightarrow -V_{\max}$ to $+V_{\max}$ in order to avoid the particle moving too far away from search space and the value of V_{\max} is preset based on the problem under consideration. Normally, V_{\max} is in the range of $[-X_{\max}, +X_{\max}]$. But for our problem the position matrix is a binary matrix and hence we determine V_{\max} by

$$V_{\max} = \left\lceil \frac{NN \times NR}{NC} \right\rceil, \quad (14)$$

where NN is the number of nodes in the cluster, NR is the number of radios in the cluster, and NC is the number of k -orthogonal channels available in the radio spectrum which is supported by the radios of each router. In the example graph shown in Figure 3(a), $NN = 4$, $NR = 6$, and $NC = 3$. The V_{\max} value is $\lceil (4 \times 6)/3 \rceil = 8$ and this fixes up the upper and lower bound of the velocity value and hence the range of velocity matrix is $[-8, 8]$ for the problem. The velocity can take values up to 16 [13]. Initially, the velocity matrix $V_k(i, j)$ is generated randomly at time t where i is the number of links in C_G^t and j is the number of orthogonal channels. This matrix provides the information with respect to the particles movement. If $V_k^{t+1}(i, j) > V_k^t(i, j) \forall i, j$ it indicates that the particle's position $X_k^{t+1}(i, j)$ is close to the particle's best position. Otherwise,

```

Initialize the following parameters:
Particle index  $k$ 
Particles best position matrix  $Pb_k^t(i, j)$ 
Particles current position  $X_k^t(i, j)$ 
Velocity clamping range  $V_{\max}$ 
Velocity matrix  $V_k^t(i, j)$ 
Inertia weight  $\omega$ 
Cognitive constant  $\varphi_1$ 
Stochastic random number  $r_1$ 
Initialize the  $D$  dimensional swarm with  $P$  particles
repeat
   $\forall$ particle  $k \rightarrow 1$  to  $P$ 
    Calculate  $f(X_k^t(i, j))$  &  $f(Pb_k^t(i, j))$  //  $f(\cdot)$  is the fitness function
    if ( $f(X_k^t(i, j)) < f(Pb_k^t(i, j))$ )
       $\forall i, j \leftrightarrow Pb_k^t(i, j) = X_k^t(i, j)$  // current assignment is best
    elseif ( $f(X_k^t(i, j)) > f(Pb_k^t(i, j))$ )
       $\forall i \in L \rightarrow 1$  to  $L$ 
         $\forall j \in C \rightarrow 1$  to  $C$ 
          if ( $Pb_k^t(i, j) > X_k^t(i, j)$ )
             $V_k^{(t+1)}(i, j) = \omega \times V_k^t(i, j) + \varphi_1 \times r_1$ 
          else
             $V_k^{(t+1)}(i, j) = \omega \times V_k^t(i, j) - \varphi_1 \times r_1$ 
            if ( $\max < V_k^{(t+1)}(i, j)$ )
               $\max = V_k^{(t+1)}(i, j)$ 
             $X_k^{t+1}(i, j) = 1$ 
          End if
        End if
      End for
    End for
  End for
End for
Until

```

ALGORITHM 5: Discrete Particle Swarm Optimization for CGCA.

$X_k^{t+1}(i, j)$ is moving away from the best position and if there is no change it is already in the best position:

$$X_k^t(i, j) = \begin{cases} 1, & \text{if } j\text{th channel is assigned to } i\text{th link} \\ 0, & \text{if there is no channel assignment,} \end{cases} \quad (15)$$

$$Pb_k^t(i, j) = \begin{cases} 1, & \text{if } j\text{th channel is assigned to } i\text{th link} \\ 0, & \text{if there is no channel assignment.} \end{cases} \quad (16)$$

Equations (15) and (16) are used to initialize the particle position and its best position, respectively. In consecutive iterations, $V_k^{t+1}(i, j)$ is updated based on (13). Then, particle's position $X_k^{t+1}(i, j)$ is updated based on the updated velocity $V_k^{t+1}(i, j)$ using (17). In each iteration, if $f(X_k^t(i, j)) < f(Pb_k^t(i, j))$, where f is the fitness function, then the best position matrix is replaced by the current position. Otherwise, the velocity and position matrices are updated and the position is evaluated. This particle tends to move towards the

best particle Pb to satisfy the fitness function. At any particular iteration, the particle reaches Pb , which becomes the termination condition of the algorithm. We have employed PSO for optimally allocating the k -orthogonal channels to the links in order to reduce the interference in the network, which will further increase the throughput of the data transmission. The particle position is updated by

$$X_k^{t+1}(i, j) = \begin{cases} 1 & \text{if } (V_k^{(t+1)}(i, j) = \max \{V_k^{(t+1)}(i, j)\}), \quad \forall j \in \{1, 2, \dots, c\} \\ 0 & \text{otherwise.} \end{cases} \quad (17)$$

To update the position matrix, we identify the maximum value in each row and 1 is placed in that index of $X_k^{t+1}(i, j)$. Algorithm 5 shows the steps involved in optimizing our objective function. The solution to this optimization problem states the possible combination of channels that can be assigned to the links in order to minimize the interference.

6. Experimental Evaluation

Computational Geometry Based Channel Assignment (CGCA) is a traffic independent channel assignment

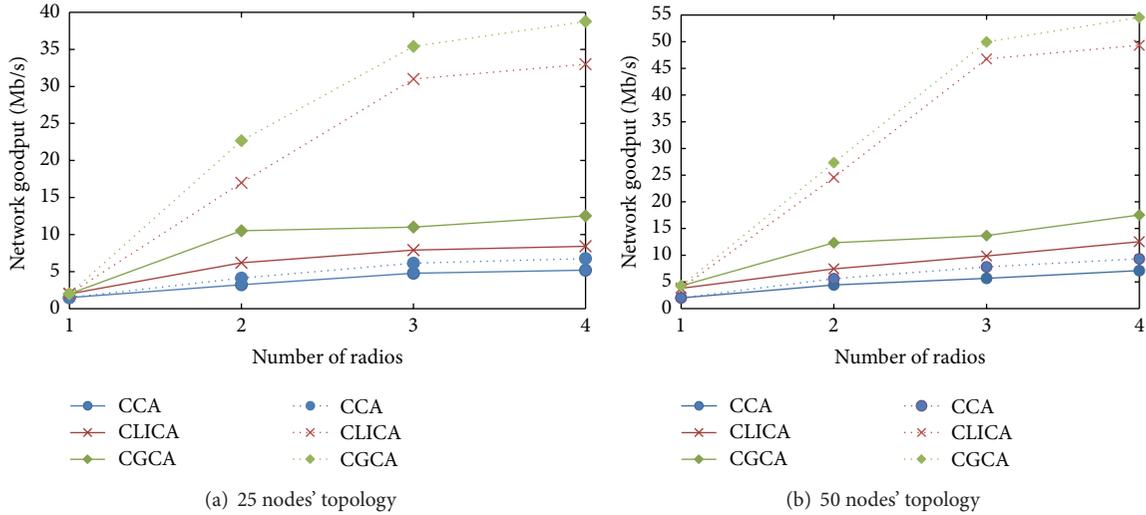


FIGURE 5: Radios versus goodput.

solution that attempts to achieve maximum throughput by minimizing the interference in the network. We have used the default routing protocol (HWMP) for routing and measured the performance of our system with Connected Low Interference Channel Assignment (CLICA) and Common Channel Assignment (CCA) [49]. CLICA uses greedy approach to assign channels. So, the algorithm has to compute the unassigned radios for further assignment. As the algorithm uses greedy approach, fairness is sacrificed [22]. CLICA methodology uses the nodes to construct the Conflict Graph for interference model and hence they do not consider interference caused by colocated radios. The proposed algorithm CGCA addresses the problems and hence performs better than CLICA. We performed simulations to evaluate the performance of CGCA and the results suggest that the distributed channel assignment performs well.

6.1. Simulation Environment. Our scheme is evaluated by simulation using the simulator tool Network Simulator [50] with an extension Hyacinth [51] to support multiple interfaces. We have used the HWMP, the default routing protocol of IEEE 802.11s. Table 2 shows the parameters used in our simulation environment.

6.2. Performance Analysis. CGCA presents an optimal or near optimal solution for each feasible solution. The performance shows that it minimizes the interference by optimally utilizing the available orthogonal channels in the network. In this subsection, we have used various parameters to evaluate our algorithm and have compared it with CLICA and CCA. The graphs show the comparison among all the three traffic independent channel assignment algorithms and show that our algorithm performs well compared to the other two algorithms.

Graphs in Figure 5 show the change of goodput of the network with increasing number of radios in 25 nodes' and 50

TABLE 2: Parameters used in simulation.

Parameter (s)	Value (s)
Simulation time	250 s
Node Distribution Model	Uniform 0, 100
Transmission range (Tr)	250 m
Interference range (Ir)	550 m
Network area size	$1000 \times 1000 \text{ m}^2$
Network	25 nodes' and 50 nodes' scenarios
MAC standard used	IEEE 802.11 n
Routing protocol	HWMP
Packet size	1500 bytes
Propagation Model	Two-rays
Data stream type	CBR
Mobility	Static nodes

nodes' topologies. Goodput is defined to be the ratio between the number of packets sent and the number of packets received per second. The performance is tested for both the 25 nodes' and the 50 nodes' scenarios eventually increasing the number of orthogonal channels from 3 to 12. From the results shown in graphs, it is observed that even by increasing the number of radios for each node it does not improve the goodput much, but increasing the number of orthogonal channels shows high improvement in the goodput. This is because the number of orthogonal channels C available in the network is limited [5]. This simulation study shows that CGCA and CLICA adjust themselves based on the number of available channels.

The results in Figure 6 show the measurement of average throughput over 25 nodes' and 50 nodes' topologies for different number of available channels. CGCA shows up to 50% and 15% increase in the throughput when compared with CCA and CLICA, respectively. The average throughput of CCA is less because the algorithm requires as many radios as the number of channels available. The throughput generally

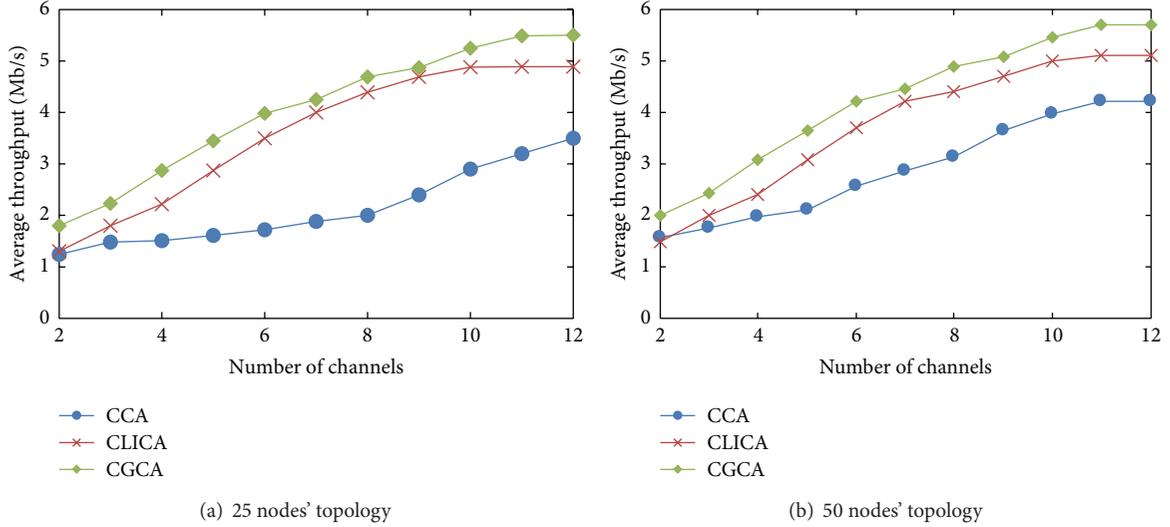


FIGURE 6: Orthogonal channels versus average throughput.

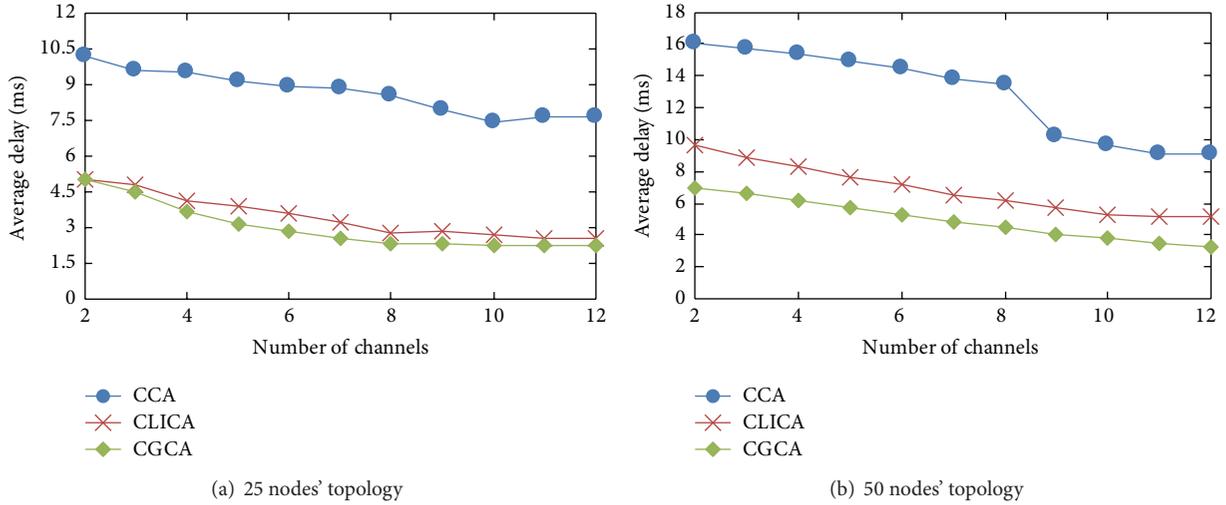


FIGURE 7: Channels versus average delay.

increases when the number of radios eventually increases. The throughput of CGCA has increased because the channels are assigned locally and each group has more number of channels. CGCA is not a greedy algorithm and in CGCA the links are assigned with channels by applying the triangulation property. The nodes forming a triangle independently choose any channel $c_j \in C$ if the triangle is not equilateral. The selection of the channel is random from the available set of orthogonal channels.

Figure 7 shows the average delay concerning the number of radios per node in the network which participate in forwarding the data. We have evaluated our algorithm for different traffic flows and the graph plotted against the number of radios and average delay illustrate that CGCA delay is reduced to 2.27 ms for 25 nodes' topology and 3.24 ms for 50 nodes' topology, respectively, when compared to CCA and CLICA. Delay is reduced in CGCA since the channel

switching delay is reduced due to the fact that there are more channels available for assignment and the CGCA is designed based on the static channel assignment strategy. As a result of this strategy, the channels for the links in the path for the data to flow through the network are predetermined which reduces the time taken for selecting and binding the channel with the links in the path.

The throughput of the proposed work is high because of the increase in the packet delivery ratio, which is evident from the results shown in Figure 8. The result of this experiment also indicates that the packet loss during the transmission of data is less when compared to the other two systems. We have also shown the result produced in both the scenarios. As the number of nodes increases, there is more assurance provided by the network in forwarding and receiving packets in WMN. But the problem here is that as the number of nodes increases the network incurs more interference. But

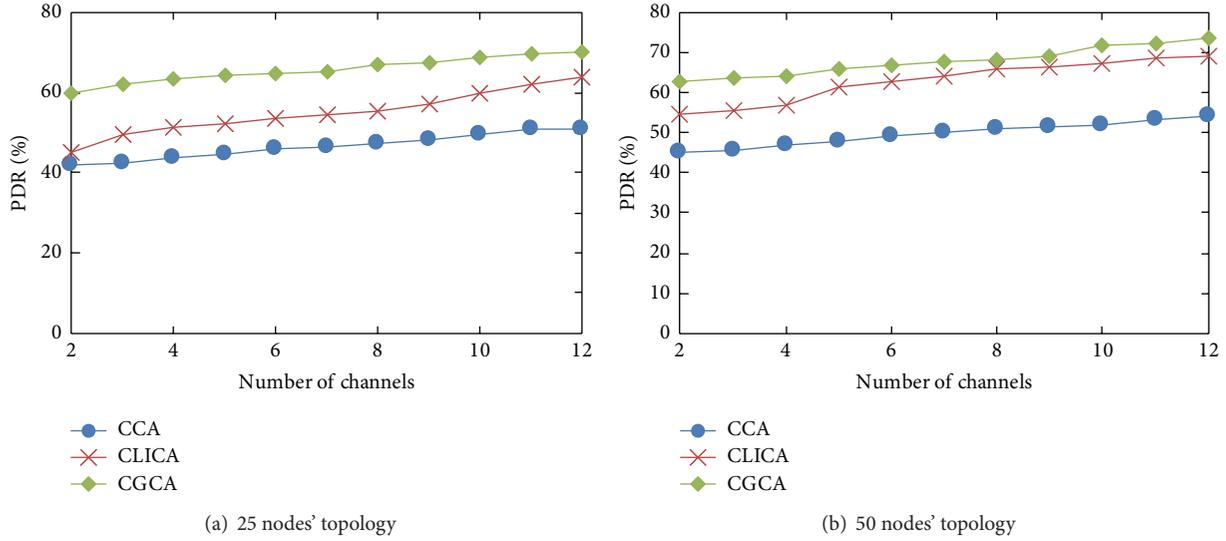


FIGURE 8: Channels versus packet delivery ratio.

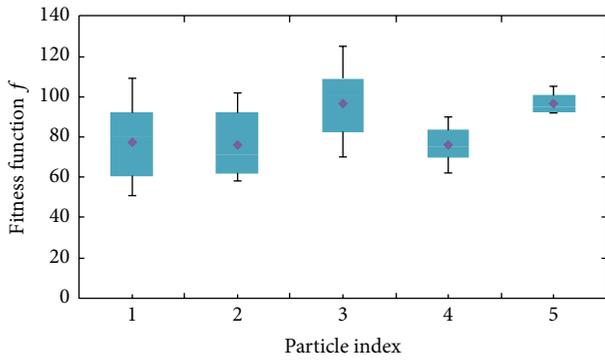


FIGURE 9: Particle indices versus fitness function.

the proposed algorithm minimizes the number of interfering links which leads to less interference in the network. Due to this, reduced interference PDR increases.

The box and whisker chart shown in Figure 9 explain how the fitness function varies based on particle P_k where $k = 1, 2, \dots, 5$. We have evaluated the fitness function f (objective function) for 5 particle indices in 15 iterations. The stopping criterion for this experiment is the number of iterations. The mean value of the fitness function is plotted inside every box. The mean value lies between 76 and 96 approximately and does not exceed the range. This is because the particle position is fixed by the maximum velocity. The range for the velocity to vary is fixed by velocity clamping parameter. Velocity clamping ensures that the velocity does not fluctuate beyond the range.

Graph in Figure 10 shows the number of times a channel is used in each group. The number of groups considered for our experiment ranges from 4 to 10 formed using the Voronoi diagram as explained in Section 5.1. We tested our algorithm in both spectrums, namely, 2 GHz and 5 GHz, where the former has only 3 nonoverlapping channels and

the latter consists of 8 nonoverlapping channels. Both results in Figure 10 describe that the channels assigned to links are independent and do not depend on the traffic flows; that is, each radio independently chooses the channels irrespective of other radios. There is no control imposed on the radios for choosing the channels. The selection of channels need not follow any constraints unless the channel is supported by the standard and belongs to C . This is shown by the mean value (number of times the channel is used) of each channel that moves to and fro from high to low and low to high. The mean value of the same channel allocated in 5 GHz is less when compared to 2 GHz radio since the number of orthogonal channels is increased. This implicitly states that the increase in the orthogonal channels improves the performance of the network by reducing the interference and increasing the throughput. The best channel allocation is determined by the fitness function shown in (1).

The interference range is always greater than the transmission range for each node; that is, $I_r > T_r$. This helps the wireless nodes to transmit the data and sense the neighboring nodes; that is, say u and v are the nodes and the distance between them should be less than or equal to the transmission range $d(u, v) \leq T_r$. But there is always a trade-off between the connectivity and the interference between the nodes in the network. Communication between nodes happens only if it falls in the transmission range of each other. If the connectivity is better, then the interference is more due to the signal overlap. This signal overlapping can be reduced to some extent by employing the nonoverlapping channels but cannot be completely avoided as the number of orthogonal channels is less. Figure 11 shows the probability of packet loss with respect to the interference range. The packet loss probability increases for increased interference range because when the interference range increases the links between the nodes interfere and the signal collides if links use the same orthogonal channel (which cannot be completely avoided). The packet loss probability also increases even if the

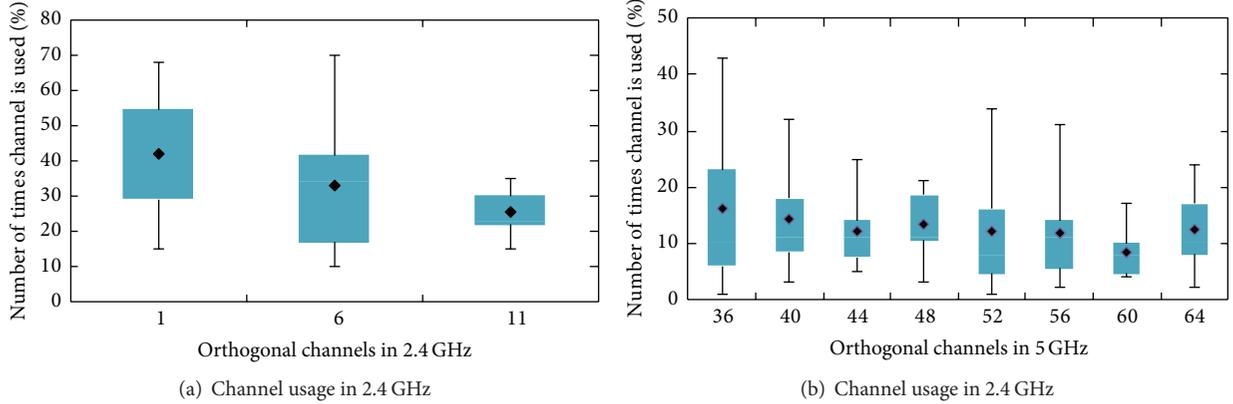


FIGURE 10: Orthogonal channels versus channel usage.

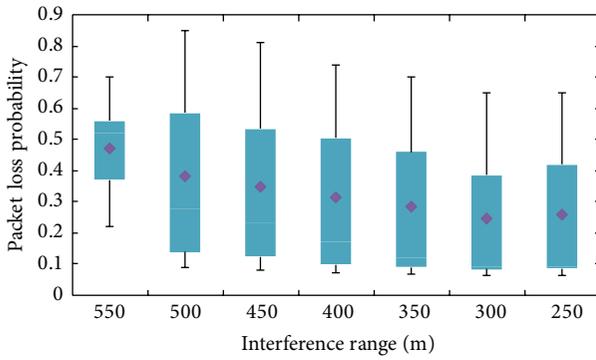


FIGURE 11: Interference versus packet loss.

interference range decreases since the connectivity between the nodes is reduced and hence the nodes are not able to transmit the packets completely.

The interference range is varied with respect to the transmission range. We have fixed the transmission range to be 250 m. The first box shows that the transmission range and interference range are same and the last box shows that the interference range is equal to transmission range. The mean value in the graph states that the packet loss is high when the interference range is too high and too low with respect to the transmission range of the node.

7. Conclusion

In this paper, we have formulated a programming model for the channel assignment problem and we have employed the stochastic approach to solve the problem. We have done the channel assignment for a small group rather than assigning for the entire network. The groups are created using the Voronoi diagram and then we use DPSO, a bioinspired algorithm which produces optimal or near optimal solutions, to solve our channel assignment problem. The channel assignment problem has been formulated as a nonlinear programming model. We have done an exhaustive experimental study using NS2 simulation and observed that our algorithm

achieves good throughput and less packet loss by optimally assigning the channels to the links. The control overhead is more in our algorithm during the leader selection process and also there is a lack of binding among the groups. Our future direction is to reduce the control overhead and achieve binding among the groups.

Notations

N :	Number of nodes in the network
M_L :	Link Matrix derived from the graph $C_C(V_C, E_C)$ of size $V_C \times V_C$
I :	Interference Matrix derived from the graphs $C'_G(V'_G, E'_G)$ and $C_C(V_C, E_C)$
$C_G(V_G, E_G)$:	Connectivity Graph with V_G vertices and E_G edges
$C'_G(V'_G, E'_G)$:	Multiradio Connectivity Graph with V'_G vertices and E'_G edges
$C_C(V_C, E_C)$:	Conflict Graph with V_C link vertices and E_C link edges
R_i :	Number of radios in each node
LDR_m :	Leader of m th group where $m = 1, 2, \dots, m$ and $m \leq N$
R_{ldr} :	Radios in the leader node
R_{INL} :	Radios associated with the incident link vertices V_C
L :	Number of links $L \subseteq V_C$
INL:	Incident links for the leader
$D(v_c)$:	Degree of the node $v_c \in R_{ldr}$
Δ :	Triangle
Δ_{sides} :	Sides of the triangle
$\ddot{\}$:	Between
\vdash :	Derived from
d :	Euclidean distance between nodes u and v
\angle :	Angle.

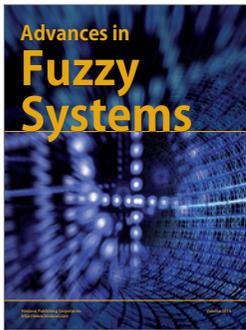
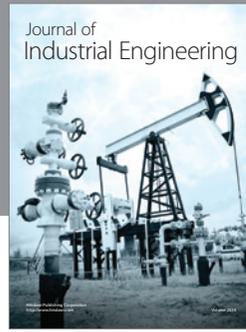
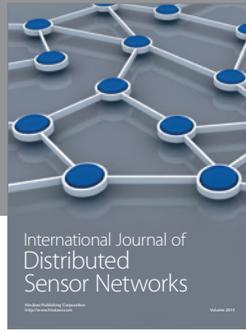
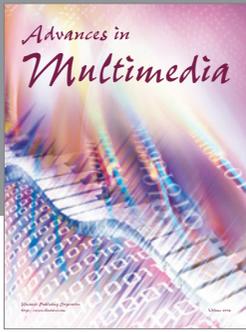
Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks*, vol. 47, no. 4, pp. 445–487, 2005.
- [2] M. Seyedzadegan, M. Othman, B. Mohd Ali, and S. Subramaniam, "Wireless mesh networks: WMN overview, WMN architecture," in *Proceedings of the International Conference on Communication Engineering and Networks (IPCSIT '11)*, vol. 19, pp. 12–18, 2011.
- [3] D. Benyamina, A. Hafid, and M. Gendreau, "Wireless mesh networks design—a survey," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 299–310, 2012.
- [4] IEEE Standard Association, *Part II: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 10: Mesh Networking*, IEEE Standard Association, New York, NY, USA, 2011.
- [5] A. Raniwala, K. Gopalan, and T. Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, no. 2, pp. 50–65, 2004.
- [6] A. Raniwala and T.-C. Chiueh, "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, vol. 3, pp. 2223–2234, Miami, Fla, USA, March 2005.
- [7] Z. Zaichen and Y. Xutao, "A simple single radio multi-channel protocol for wireless mesh networks," in *Proceedings of the 2nd International Conference on Future Computer and Communication (ICFCC '10)*, vol. 3, pp. V3441–V3445, Wuhan, China, May 2010.
- [8] S. Chiochan and E. Hossain, "Channel assignment for throughput optimization in multichannel multiradio wireless mesh networks using network coding," *IEEE Transactions on Mobile Computing*, vol. 12, no. 1, pp. 118–135, 2013.
- [9] M. Kodialam and T. Nandagopal, "Characterizing the capacity region in multi-radio multi-channel wireless mesh networks," in *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking (MobiCom '05)*, pp. 73–87, Cologne, Germany, September 2005.
- [10] A. P. Subramanian, H. Gupta, S. R. Das, and J. Cao, "Minimum interference channel assignment in multiradio wireless mesh networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 12, pp. 1459–1473, 2008.
- [11] S. Avallone and I. F. Akyildiz, "A channel assignment algorithm for multi-radio wireless mesh networks," *Computer Communications*, vol. 31, no. 7, pp. 1343–1353, 2008.
- [12] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot, "Interference-aware channel assignment in multi-radio wireless mesh networks," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, vol. 6, Barcelona, Spain, April 2006.
- [13] H. Izakian, B. T. Ladani, A. Abraham, and V. Snášel, "A discrete particle swarm optimization approach for grid job scheduling," *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 9, pp. 4219–4234, 2010.
- [14] E. G. Coffman, *Computer and Job-Shop Scheduling Theory*, John Wiley & Sons, New York, NY, USA, 1976.
- [15] R. Zhang and C. Wu, "Bottleneck machine identification based on optimization for the job shopscheduling problem," *ICIC Express Letters*, vol. 2, no. 2, pp. 175–180, 2008.
- [16] M. K. Marina, S. R. Das, and A. P. Subramanian, "A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks," *Computer Networks*, vol. 54, no. 2, pp. 241–256, 2010.
- [17] K. Bong-Jun, V. Misra, J. Padhye, and D. Rubenstein, "Distributed channel assignment in multi-radio 802.11 mesh networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '07)*, pp. 3978–3983, Kowloon, Hong Kong, March 2007.
- [18] W. Alsalih, K. Islam, Y. Núñez-Rodríguez, and H. Xiao, "Distributed Voronoi diagram computation in wireless sensor networks," in *Proceedings of the 20th ACM Annual Symposium on Parallelism in Algorithms and Architectures (SPAA '08)*, p. 364, ACM, Munich, Germany, June 2008.
- [19] B. A. Bash and P. J. Desnoyers, "Exact distributed Voronoi cell computation in sensor networks," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 236–243, Cambridge, Mass, USA, April 2007.
- [20] H. Chin Jang, "Applications of Geometric Algorithms to Reduce Interference in Wireless Mesh Network," *International Journal on Applications of Graph Theory in wireless Ad Hoc Networks and Sensor Networks*, vol. 2, no. 1, pp. 62–85, 2010.
- [21] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.
- [22] W. Si, S. Selvakennedy, and A. Y. Zomaya, "An overview of Channel Assignment methods for multi-radio multi-channel wireless mesh networks," *Journal of Parallel and Distributed Computing*, vol. 70, no. 5, pp. 505–524, 2010.
- [23] S. M. Kala, R. Musham, M. P. Kumar Reddy, and B. Reddy Tamma, "Interference mitigation in wireless mesh networks through radio co-location aware conflict graphs," *Wireless Networks*, 2015.
- [24] X. Zhuang, H. Cheng, N. Xiong, and L. T. Yang, "Channel assignment in multi-radio wireless networks based on PSO algorithm," in *Proceedings of the 5th International Conference on Future Information Technology (FutureTech '10)*, pp. 1–6, Busan, Republic of Korea, May 2010.
- [25] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science (MHS '05)*, vol. 43, pp. 39–43, Nagoya, Japan, October 1995.
- [26] R. C. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the Congress on Evolutionary Computation*, vol. 1, pp. 81–86, IEEE, Seoul, Republic of Korea, May 2001.
- [27] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, Perth, Wash, USA, December 1995.
- [28] A. Raniwala, K. Gopalan, and T. Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, no. 2, 2004.
- [29] J. Crichigno, M.-Y. Wu, and W. Shu, "Protocols and architectures for channel assignment in wireless mesh networks," *Ad Hoc Networks*, vol. 6, no. 7, pp. 1051–1077, 2008.
- [30] P. Kyasanur and N. H. Vaidya, "Routing and interface assignment in multi-channel multi-interface wireless networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '05)*, vol. 4, pp. 2051–2056, IEEE, March 2005.

- [31] A. Bezzina, M. Ayari, R. Langar, G. Pujolle, and L. Saidane, "Interference-aware game-based channel assignment algorithm for MR-MC WMNs," in *Proceedings of the IFIP Wireless Days (WD '14)*, pp. 1–6, IEEE, Rio de Janeiro, Brazil, November 2014.
- [32] P. B. F. Duarte, Z. M. Fadlullah, A. V. Vasilakos, and N. Kato, "On the partially overlapped channel assignment on wireless mesh network backbone: a game theoretic approach," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 1, pp. 119–127, 2012.
- [33] M. A. Nezhad and L. Cerdà-Alabern, "Adaptive channel assignment for wireless mesh networks using game theory," in *Proceedings of the 8th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS '11)*, pp. 746–751, Valencia, Spain, October 2011.
- [34] W. Kim, A. J. Kessler, M. Di Felice, and M. Gerla, "Urban-x: towards distributed channel assignment in cognitive multi-radio mesh networks," in *Proceedings of the IFIP Wireless Days (WD '10)*, Venice, Italy, October 2010.
- [35] L.-H. Yen, Y.-K. Dai, and K.-H. Chi, "Resource allocation for multi-channel multi-radio wireless backhaul networks: a game-theoretic approach," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC' 13)*, pp. 481–486, Shanghai, China, April 2013.
- [36] <http://www.symposium-magazine.com/game-theory-is-useful-except-when-it-is-not-ariel-d-procaccia>.
- [37] N. M. Dhande and V. S. Kapse, "Interference aware channel assignment in WMN using a game theoretic approach," *International Journal of Engineering Research & Technology*, vol. 2, no. 9, 2013.
- [38] S. C. Nandy, *Advanced Computing and Microelectronics*, Lecture notes on Voronoi Diagrams, Unit Indian Statistical Institute, Kolkata, India, <http://www.tcs.tifr.res.in/~ghosh/subhas-lecture.pdf>.
- [39] M. De Berg, O. Cheong, M. van Kreveld, and M. Overmars, "Voronoi diagrams," in *Computational Geometry Algorithms and Applications*, Springer, 3rd edition, 2008.
- [40] Y. Yang, J. Wang, and R. Kravets, "Interference-aware load balancing for multihop wireless networks," Tech. Rep. UIUCDCS-R-2005-2526, Department of Computer Science, University of Illinois at Urbana-Champaign, 2005.
- [41] J. Padhye, S. Agarwal, V. N. Padmanabhan, L. Qiu, A. Rao, and B. Zill, "Estimation of link interference in static multi-hop wireless networks," in *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement (IMC '05)*, p. 28, Berkeley, Calif, USA, October 2005.
- [42] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom '04)*, pp. 114–128, ACM, Philadelphia, Pa, USA, October 2004.
- [43] R. Mendes, *Population topologies and their influence in particle swarm performance [Ph.D. thesis]*, University of Minho, Braga, Portugal, 2004.
- [44] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global Positioning System: Theory and Practice*, Springer, 4th edition, 1997.
- [45] <https://www.sharcnet.ca/Software/TGrid/html/ug/node380.htm>.
- [46] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, vol. 5, pp. 4104–4108, IEEE, Orlando, Fla, USA, October 1997.
- [47] A.-Q. Mu, D.-X. Cao, and X.-H. Wang, "A modified particle swarm optimization algorithm," *Natural Science*, vol. 1, no. 2, pp. 151–155, 2009.
- [48] A. P. Engelbrecht, *Computational Intelligence—An Introduction*, Wiley, 2nd edition, 2007.
- [49] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, "A multi-radio unification protocol for IEEE 802.11 wireless networks," in *Proceedings of the IEEE 1st International Conference on Broadband Networks (BroadNets '04)*, pp. 344–354, October 2004.
- [50] <http://www.isi.edu/nsnam/ns/>.
- [51] <http://personales.unican.es/aguero/>.




Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

