Metaheuristic Optimisation for Radio Interface-Constrained Channel Assignment in a Hybrid Wi-Fi–Dynamic Spectrum Access Wireless Mesh Network

Natasha Zlobinsky[1(✉)], David Johnson[1], Amit Kumar Mishra[1], and Albert A. Lysko[1,2]

[1] University of Cape Town, Cape Town, South Africa
natzlob@gmail.com, akmishra@ieee.org
[2] Council for Scientific and Industrial Research, Pretoria, South Africa
alysko@csir.co.za

Abstract. Channel Assignment (CA) in wireless mesh networks (WMNs) has not been well studied in scenarios where the network uses Dynamic Spectrum Access (DSA). This work aims to fill some of this gap. We compare metaheuristic algorithms for optimising the CA in a WMN that has both Wi-Fi and DSA radios (where DSA could be Television White Spaces or 6 GHz). We also present a novel algorithm used alongside these metaheuristic algorithms to ensure that the CA solutions are feasible. Feasible solutions meet the interface constraint, i.e. only as many channels are allocated to a node as it has radios. The algorithm also allows the topology to be preserved by maintaining links. Many previous studies tried to ensure feasibility and/or topology preservation by using two separate steps. The first step optimised without checking feasibility and the second step fixed infeasible solutions. This second step often negated the benefits of the previous step and degraded performance. Other CA algorithms tend to use simple on/off interference models, instead of models that more realistically reflect the physical layer environment, such as the Signal to Interference plus Noise Ratio ($SINR$). We present our more realistic $SINR$-based model and optimisation objective. Simulated Annealing (SA) and Genetic Algorithm (GA) are applied to the problem. Performance is evaluated and verified through simulation. We find that GA outperforms SA, finding higher quality solutions faster, although both metaheuristics are better than random allocations. GA can be used daily to find good CAs in changing conditions.

Keywords: Channel Assignment · Dynamic Spectrum Access · DSA · Wireless Mesh Networks · WMN · CBRS · Wi-Fi 6E · TVWS · Genetic Algorithm · Simulated Annealing

H. Jin et al. (Eds.): CROWNCOM 2021/WiCON 2021, LNICST 427, pp. 56–76, 2022.
https://doi.org/10.1007/978-3-030-98002-3_4

1 Introduction

Recently, Dynamic Spectrum Access (DSA) has been gaining traction again. This is as regulatory bodies around the world have been opening up the spectrum bands that were formerly reserved for licensed users, for opportunistic use by other users. Examples are Citizens Broadband Radio Service (CBRS) Spectrum Access System (SAS) [1] and Automated Frequency Coordination in Wi-Fi 6E [2]. Most of the associated spectrum bands require (or will require) the use of databases to acquire access to channels within the bands. Another technology, Wireless Mesh Networks (WMNs), has proven its usefulness in extending Internet access from a gateway node to a wider area [3–6]. This is especially useful in rural areas or informal settlements where Internet connectivity infrastructure is not reliable. Bringing together DSA and WMN technologies can be very advantageous, especially in bringing connectivity to the unconnected, unreliably connected, or underserved.

This novel type of network comes with new challenges and avenues for research. Channel Assignment in such DSA WMNs is especially challenging. The limited channel availability, the fact that different nodes may have different allowed channels, interference within the network, as well as the possibility of other secondary users causing interference, all add to the complexity of an already NP-complete problem [7].

This work employs two metaheuristic algorithms (Simulated Annealing and Genetic Algorithm) to address the Channel Assignment problem in a WMN that uses DSA. We introduce a novel algorithm that ensures both the radio interface constraint and the connectivity or topology preservation constraint are met. This algorithm is used in conjunction with either Simulated Annealing or Genetic Algorithm. We optimise on Signal to Noise and Interference Ratio ($SINR$), rather than unrealistic binary interference models.

We continue this section with some brief background information on DSA technologies, as well as the metaheuristic optimisation techniques used in this work. An overview of related work is given in Sect. 2. Then, in Sect. 3, we present and formulate models for the problem. The methodology is detailed in Sect. 4. Simulation results are presented and discussed in Sect. 5, before concluding.

1.1 Dynamic Spectrum Access

Dynamic Spectrum Access has emerged as a way for the radio frequency spectrum to be used more efficiently. DSA became more important after it had been found that large parts of the radio spectrum remain unused while being licensed to certain users, creating an artificial spectrum scarcity. DSA refers to any of a number of techniques whereby wireless frequency bands can be shared opportunistically between the primary (licensed) users of the spectrum and secondary (unlicensed) users (SUs). It is enabled by cognitive radio, through spectrum sensing and/or the use of Geolocation Spectrum Databases (GLSDs). While practical spectrum sensing still remains in the research stage, GLSD based approaches have received wide acceptance and practical use. Using DSA methods, radios can adjust their spectrum use according to current environmental conditions while ensuring that Primary Users (PUs) or incumbents are protected from harmful interference.

Television White Spaces (TVWS) is one band in which DSA is used. TVWS refers to the unused portions of the spectrum in the 470–694 MHz range traditionally licensed to TV transmitters. SUs have been allowed to access this spectrum by a number of national regulatory bodies, including the FCC in the United States of America, Ofcom in the United Kingdom, and ICASA in South Africa. Most regulations require the use of a GLSD to ensure compliance and protection of TV broadcast services.

Citizens Broadband Radio Service (CBRS) is a band of spectrum in the 3.5 GHz range that was recently opened for sharing with incumbents for commercial use in the United States [1]. Service providers can deploy networks in this band without requiring spectrum licenses. Access is divided into three tiers: incumbent access, priority access, and general authorised access. CBRS uses a Spectrum Access System (SAS), which grants requests by SUs to access channels in the band, using a database of CBRS radio base stations, similar to the GLSD in TVWS.

To minimise interference with satellite links, Wi-Fi 6E is set to use Automated Frequency Selection (AFC), as the 6 GHz band has been opened up for unlicensed use by either low power indoor Access Points (APs), or standard power outdoor Wi-Fi APs [2]. This will also use a database to coordinate spectrum use among all users. To obtain available channels and request access, APs must consult an AFC provider before starting to transmit.

Our work can be extended to any and all of these DSA technologies and so we expect it to become increasingly useful in time.

1.2 Metaheuristic Algorithms for Optimisation

We give some brief background on the metaheuristic stochastic optimisation algorithms employed in this work. We have selected these algorithms because they are some of the most well-known and readily available algorithms, which are widely applied and verified. This means they would be easier to implement in a real network. It also means that our experiments can be replicated readily using the same algorithms, perhaps in other coding languages or with other simulation frameworks. For these reasons, we have also chosen to implement the most common "vanilla" versions of these algorithms. Variations are left as future work.

Simulated Annealing. Simulated Annealing (SA) is a probabilistic search heuristic used in optimisation problems with complex, often discrete, search spaces. It is based on, and analogous to, the physical process of annealing (of a metal, for example) in statistical mechanics, whereby atoms are cooled in a specific slow way until reaching the state of minimum energy [8]. The aim is always to find the lowest "energy" solution. That is the solution with the lowest cost. The algorithm starts with the system in a certain arbitrary configuration or state, i.e. a solution, and then it computes the "energy", which is the value of the objective function or cost of that solution at that iteration. From there, a new neighbour solution is generated by applying a slight alteration to the system state and its cost value computed and compared to the previous cost. The new solution is either accepted or rejected based on whether it has a lower "energy" value than the first solution and according to the temperature parameter. The new candidate solution is always accepted if the cost value has improved, and accepted probabilistically if the new solution is worse. The probability of acceptance is based on the difference in cost between the new candidate solution and the old solution, as well as on the current temperature value. The accepted solution is then the starting point for the next iteration.

The temperature parameter relates to how likely the algorithm is to choose a worse solution than the current one, which can prevent it from stagnating on a local minimum. The temperature must initially be set to a certain high value and decreased every iteration according to a defined cooling function, the choice of which is up to the implementer. Some examples are exponential multiplicative cooling, logarithmic multiplicative cooling, and linear multiplicative cooling [9]. The process of generating a new neighbour solution and accepting or rejecting the solution continues until the termination conditions are met. These could be a specified number of iterations or when an acceptable running time is reached, or an acceptably low solution has been settled on. Certain tests and rules-of-thumb can be followed to determine whether to stop or continue with the algorithm or estimate the convergence time, e.g., the Geweke test [10].

Genetic Algorithm. The Genetic Algorithm is a well-known metaheuristic algorithm based on the evolution of genes through generations, whereby the fittest individuals are selected as parents, they reproduce, and genes occasionally mutate. The components required are:

- a fitness function (optimisation objective function);
- a population of chromosomes, also called genomes (an encoding for solutions in the solution search space);
- a selection method by which parents for the next generation are selected;
- a crossover or reproduction method to produce the next generation; and
- a mutation method by which random changes are introduced to chromosomes, preventing convergence to local minima.

The algorithm aims to find the solution with the maximum fitness. It continues until a) the fitness value of the chromosome with the best value thus far stays the same for a certain number of iterations, or b) after an acceptable predetermined total number of generations is reached. One of several parent selection methods may be used. A popular method is Roulette Wheel selection, where each chromosome in the current generation is given a probability of being selected that is proportional to its fitness. This method is vulnerable to causing premature convergence. Linear Rank selection tries to prevent a single solution from dominating and causing premature convergence in Roulette Wheel selection by instead ranking individuals according to their inverse fitness and then basing the probability of selection on the rank rather than the actual fitness value. The highest fitness solutions are given the highest value rank. For example, out of ten solutions, the highest fitness will have rank position 10 (not 1).

2 Related Work

While the channel selection and assignment problems may appear to be well studied, there is no other work that presents an algorithm for a WMN using DSA methods, such as a GLSD, along with spectrum sensing. To the best of our knowledge, this paper is also one of the first works to use the $SINR$ perceived by the mesh nodes for CA in a WMN. It is common in the literature to use simplistic binary conflict-based objectives, using unrealistic interference and channel models, and neglecting the requirement to maintain connectivity in the network.

Simulated Annealing is evaluated by Chen and Chen [11] for CA in WMNs, while considering the interface constraint. The interface constraint states that the number of channels assigned to a node cannot exceed the number of interfaces or radios it has. In one method of Chen's work, the interface constraint is modelled with a penalty function for candidate solutions. In the other method, solutions that violate the interface constraint are not allowed, and infeasible solutions are converted to feasible solutions by merge operations. A weakness of this work is that this merge operation once again introduces the interference the first step aimed to minimise. Interference is considered binary, either present or not, and connectivity is ensured by assigning every link a channel.

Sridhar et al. present a CA methodology for multi-radio WMNs that use only Wi-Fi spectrum [12]. The optimisation goal is minimising interference. However, they also introduce a constraint to ensure that each link is assigned a channel for topology preservation. They weight the interference objective by the link traffic, which is predicted from previous averages. Lagrangian relaxation is used to find lower bounds. They also present a GA-based metaheuristic for solving the problem. A distributed algorithm is also presented, but this requires that all radios maintain a channel assignment matrix as well as a radio usage matrix for all nodes in the network, both of which are difficult to realise. Pal and Nasipuri also present a GA, but for joint routing and channel assignment [13]. They optimise on route quality. Balusu et al. combine GAs with learning automata to minimise interference in WMN CA for multicast tree topologies [14]. Multicast tree networks are also investigated by Cheng and Yang, who present GA, SA and Tabu search solutions for joint Quality of Service (QoS) routing and channel assignment in multi-radio multi-channel WMNs [15]. A GA is employed by Ding et al. for minimising total interference and maximum link interference in WMNs with partially overlapping channels [16]. They also model interference simplistically. All these works have differences from ours.

A number of works use Particle Swarm Optimisation (PSO) e.g., [17–19]. Subramanian et al. use Tabu search to minimise binary interference, first ignoring the radio constraint and then merging channel assignments to comply with the interface constraint [20]. This two-step method has the same weakness as [11], where the second step negates the first. Finally, the case of a multi-radio multi-channel network as SUs coexisting with PUs is addressed by Qin et al., using Lyapunov optimisation of throughput and average delay [21].

In view of the existing literature, we bring novelty to this field by tackling CA in WMNs in situations where the networks use the licensed spectrum opportunistically as SUs, in the presence of other SUs. Our approach uses Wi-Fi as an additional option, rather than using only Wi-Fi channels. We also take into account that different nodes may have different allowed channels, since the network is geographically spread out. Furthermore, we bring a realistic $SINR$ model instead of a simple on/off interference model. Ours is the first work to compare metaheuristic optimisation algorithms for such a network and scenario, giving consideration to all these factors. We also present a novel algorithm for ensuring that both the connectivity constraint and the interface constraint are met at once.

3 Problem Formulation

3.1 Network Model

The scenario we consider is a WMN consisting of nodes equipped with both Wi-Fi radios and radios capable of accessing alternative spectrum, such as TVWS or CBRS, as unlicensed or Secondary Users. These mesh nodes also act as APs to clients on another radio interface (this could be 2.4 GHz or 5 GHz Wi-Fi, for example). There are also Primary Users of the alternative spectrum band, which need to be protected from interference. Thus, it is required that devices use a GLSD to get a list of channels that are allowed at a device's location. This is the case for TVWS as well as Wi-Fi 6E 6 GHz AFC. A single node is the gateway to the Internet from the mesh network and also acts as the gateway to the GLSD. Mesh nodes may not all have direct access to the Internet and hence to the GLSD, but all nodes will have a connection path to the gateway node (and thus to the GLSD), which may not be optimal. The gateway node will gather the list of allowed channels and powers for all the nodes in the network.

Ensuring that all the nodes have an initial connection to the GLSD in a way that complies with regulation could be done using the method of Maliwatu [22]. In this method, nodes begin in passive scanning mode, listening for beacons, while one node (the gateway in our case) has Internet access. The node with Internet and GLSD access picks a channel and broadcasts beacon frames on this channel, along with an ordered list of alternative channels. One-hop neighbours receive this beacon frame, tune to that channel, and query the GLSD through the first node. The one-hop neighbour then selects a channel from the list of alternative channels. It can now join the network and start broadcasting beacons for the next-hop neighbour. This then allows second-hop neighbours to repeat the process and join the network, through the one-hop neighbour. This process continues until reaching the outermost set of nodes. We also assume that the gateway node will act as a controller, gathering the average $SINR$ readings from all the nodes and performing any channel assignment optimisation algorithm.

In addition, the network may be in the presence of devices external to the network, which are also making use of the alternate spectrum band and so may cause interference. An example of this scenario is shown in Fig. 1.

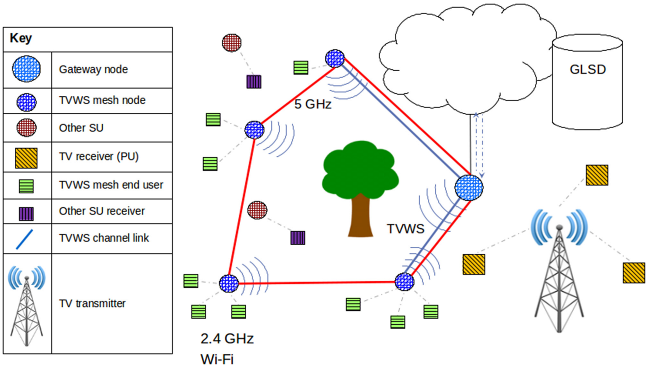


Fig. 1. An infrastructure WMN using both DSA alternative spectrum and Wi-Fi

3.2 Problem Statement and Motivation

Given this scenario, the question arises, "how to allocate channels to the mesh node radio interfaces optimally, according to certain metrics?". The main issues are minimising interference within the network and from external interference sources, while ensuring connectivity is guaranteed. Connectivity must at least be maintained along the most important paths, and between as many nodes as possible. Different channels may be allowed for use by different nodes in the network because they are placed in different geographic locations. In addition, different channels may experience different levels of external interference, loss, fading, and utilisation. Hence, the problem of assigning channels optimally is an important and difficult one in this scenario.

The CA problem is well known to be NP-hard since it is, in essence, a graph-colouring problem [7]. In the context of a WMN, it is even more difficult and goes beyond a basic graph colouring problem. Firstly, this is because the links are not the same, as mentioned, and would require a model of a weighted graph. Secondly, this is because, while interference must be avoided, it is also necessary to maintain connectivity and meet the interface constraint. We have determined that the problem is also not convex, by plotting the objective function for a scaled-down three-node (A, B, C) three-link (A-B, B-C, A-C) version of the problem, shown in Fig. 2. Each of the three axes represents the channels assigned to a link. The sawtooth shape in the one plane, and presence of higher values within the low-value regions (shown by purple, red and orange values inside the black region) make this problem non-convex, even in low dimensions. This justifies our use of metaheuristic optimisation algorithms and not convex optimisation algorithms.

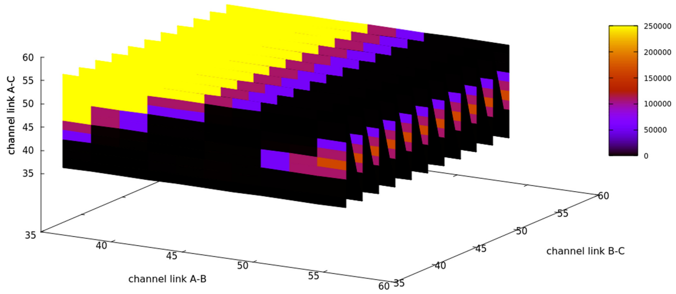


Fig. 2. Map of the objective function value of CA problem in a three-node WMN

3.3 Assumptions

The goal of the CA algorithm is to assign channels to a set of links.

Definition 1. *A link is defined as a pair of radio interfaces between which traffic could potentially flow directly if tuned to the same channel.*

In a network, over the course of a day, the routing algorithm will select and use various paths. Therefore, the set of links used for relaying traffic over the course of a day will vary. The selected paths are dependent on the capacity of the links, which is affected by the channel allocation. On the other hand, channel allocation should consider the links used, especially those with the highest traffic load. So there is a circular dependency between the two problems of routing and CA. While these two issues are very much interlinked, our channel assignment will be quasi-static (or semi-dynamic) and not change according to routing in near real-time.

This is a practical and advantageous decision, rather than a limitation. Suppose the CA attempts to keep up with the rapidly changing routes, and routing is, in turn, trying to keep up with changing channel allocations. This would cause network instability, which leads to a bad user experience, which is not desirable. Channel switching causes loss of network connectivity during the time the Network Interface Card (NIC) switches its channel and tries to re-establish connectivity, and this can be on the order of seconds in reality. Optimisation algorithms, such as those we present here, are time-consuming to run and resource-intensive. This is especially true on commodity mesh radios, which are resource-constrained, even if a dedicated controller node is used with more power. The distribution of the final channel assignment to the nodes in the network also requires time. These factors all point to the fact that we would not want the CA to change, or the optimisation algorithm to run, too often. A reasonable trade-off would thus be to run the optimisation once a day, for example. This could be run at a time when the network is not busy, such as in the middle of the night. A 24-h schedule such as this is already employed by other systems for resource management (e.g., Aruba Airmatch [23]) so it is practical and can be accepted in the industry.

Some other assumptions that apply are:

- Nodes are stationary, and the gateway node knows their locations. The mechanism for obtaining and distributing location information is out of the scope of this work.
- The nodes are mostly in the same geographical area. However, some nodes on the edges may be in different geographical areas, where the GLSD defines the boundaries. If they are not, the WMN can be partitioned into clusters with largely overlapping allowed channel lists. For this reason, we also do not present results for larger WMNs, as a large network would be partitionable into clusters. There are also practical limitations on performance in the case of large WMNs. We consider a network of 50 or more nodes as large.
- If the nodes at the cusp of two clusters do not share a sufficient number of overlapping allowed channels in the DSA band, they can be linked by a Wi-Fi channel.
- Channel widths are fixed to the same value for all channels at all nodes.
- We use average $SINR$ measurements per node in the optimisation. This is because, if the average $SINR$ over the network is large, a high throughput can be expected. $SINR$ is a direct measure of the result of changing channel assignments on the signal reception and interference experienced by nodes. These measurements will be gathered on all the channels by all nodes for different possible channel assignments. An average of all the samples for a particular CA will be used in the optimisation. Either the samples, or the overall averages will be sent to the controller/gateway node to perform the optimisation. The method by which nodes obtain $SINR$ samples could be using acknowledgement (ACK) frames, similarly to Cho et al. [24].
- All links are saturated with traffic, so the total $SINR$ across the network is also a fair objective, and no other fairness criteria is necessary.

3.4 Mathematical Model

In the usual way, we model the network as a graph $G = (V, E)$ where V is the set of nodes (vertices) and edges E are the links between nodes. Edges are potential links and not necessarily carrying traffic at this stage. Each edge $e \in E$ could be tuned to a particular channel at any time, i.e. $E \mapsto C$, where C is the full set of considered allowed channels for the whole network. C is the union of channels allowed in different locations of the WMN according to the GLSD. Each node v has a set $C(v)$ of channels it is allowed to use. For two nodes v_1 and v_2, $C(v_1) \neq C(v_2)$ in general, although they could be equal and should have channels in common $(C(v_1) \cap C(v_2) \neq \emptyset)$, especially if v_1 and v_2 are neighbours. A channel is specified by a channel number, a centre frequency and a channel bandwidth. There might also be other transmitting devices (other SUs) that can influence the reception of nodes in G if they are transmitting with power in the same channel that one of the links E is tuned to. These are added to the conflict graph. Connectivity graph G maps to a conflict graph G_c.

Definition 2. *Conflict graph $G_c = (V_c, E_c)$, where the vertices of the conflict graph are the edges in G i.e. $V_c = E$. An edge $e' \in E_c$ exists between two vertices in V_c if the two links could interfere if tuned to an overlapping channel. This could occur when the interfering signal power is above the receiver sensitivity.*

We add vertices and edges representing outside sources of interference to form $\hat{G}_c$, but note that these are fixed as their channels cannot be switched and their transmit power cannot be controlled.

An edge e' exists if a transmission in link 2 causes power to leak into, or be transmitted in, the channel on which link 1 is operating. This can occur if the two links are tuned to the same channel. This can also happen if the links are tuned to different channels while the spectrum mask of the transmitter node is wide or the receive filtering is poor, so that power leaks into the channel on which link 1 is operating. We can model this as a weighted conflict graph denoted $\langle G_c(V_c, E_c), w \rangle$, where the weight w represents the interference power per link.

Considering this conflict graph, we aim to minimise the conflict but maximise the wanted signal power received by each node and so maintain connectivity in G. We can satisfy both these requirements simply by considering $SINR$. This measure encapsulates the goals of having the highest desired received signal level throughout the network, while also minimising conflict (interference). The optimisation objective is thus to find the channel assignment A, which is a mapping of $E \mapsto C$ that maximises the average $SINR$, i.e.

$$\begin{aligned} \max_{A=E\mapsto C} \sum_{v \in V} \frac{P_{wanted,v}(A)}{\sum_{i \in I} P_i(A) + N} \implies \min_A \sum_{v \in V} \frac{\sum_{i \in I} P_i(A) + N}{P_{wanted,v}(A)} \\ = \min_A \sum_{v \in V} \frac{\sum_{x \in V \setminus u} P_{x,v}(A) + N}{P_{u,v}(A)} \\ = \min_A \frac{|V|}{\sum_{v \in V} \overline{SINR_v(A)}} \end{aligned} \tag{1}$$

over all possible channel assignments, subject to the radio interface constraint:

$$|A(v)| \leq R_v \quad \forall\, v \in V \tag{2}$$

where:
$A(v)$ is the channel assignment of node v and $|\cdot|$ indicates the size (number of channels assigned to the node);
R_v is the number of radios at node v;
$P_{u,v}$ is the power received at node v from transmitting node u;
P_i is interfering power received at node v from an interfering transmission i over the whole channel width of channel c to which node v is tuned.
N is the noise power, which in ns3 is modelled as the product of the thermal noise (N_t) and the noise figure (F_N), as shown in Eq. (3).

$$N = N_t \times F_N = kTB \times F_N \tag{3}$$

where k is Boltzmann's constant ($= 1.380649 \times 10^{-23} JK^{-1}$), T is the temperature in Kelvin and B is the channel width.

A transmitting node is considered interfering with v if it is in the set of nodes V minus the node u, the node transmitting the desired signal to v. We only consider there to be one wanted receive signal per time slot.

Each transmitted signal is subject to propagation loss as well as frequency-selective fading. As usual, the received signal power at node v from node u's transmitted power $P_{u,v}$ (in W) before receive filtering is related by the propagation loss L according to the chosen loss model. We apply the basic Friis transmission loss model in Eq. (4). This also implies that we assume an isotropic antenna model, but this can also be changed in the simulation for future work. We note, however, that our method is easily extensible to other propagation loss models and is not limited to work on any particular propagation loss model only. This model is used without loss of generality.

$$P_{u,v} = P_u \frac{G_v G_u \lambda^2}{(4\pi d)^2} = \frac{P_u}{L_{u,v}} \tag{4}$$

where
G_u is the transmission gain of node u's antenna (unitless)
G_v is the receive gain of node v's antenna (unitless)
λ is the wavelength (in m), inversely proportional to the frequency, so is affected by the channel assignment
d is the distance between the nodes (in m)
or, in dB,

$$P_{u,v}(dB) = P_u(dB) - L_{u,v}(dB) \tag{5}$$

where path loss $L(dB)$ is the absolute value of the loss in dB.

Before considering interference, a link only exists if the effective received signal power on that link is above the receive sensitivity s_v of the receiver node v. That is, the link will be pruned unless

$$\begin{aligned} P_{u,v}(dB) &\geq s_v \\ SNR_v \times N &\geq s_v \\ SNR &\geq s_v/N \end{aligned} \tag{6}$$

SNR can only be measured if it is above the receiver sensitivity/noise. This constraint reduces the number of links that require channel assignment and reduces the edges in the conflict graph that need to be considered. We also have to ensure that in the CA, constraint (6) is met, so that connectivity is maintained. Additionally, interference is only considered if the interference power at the receiver is above the energy detection threshold of the receiver.

In the simulation framework of Network Simulator 3 (ns3), frames are split into constant $SINR$ chunks and overlapping frame chunks are considered as additional contributions to the overall noise. Interfering signals are only considered as interference when the frame chunks actually overlap with those of the wanted frame at each considered receiving node in time. Preamble and payload parts of frames are treated separately because the payload might have a higher modulation and coding rate than the BPSK-encoded preamble.

4 Methodology

The optimisation methods all generate candidate solutions from the CA solution space and obtain $SINR$ measurements from all nodes based on that solution (CA), in order to optimise on that measurement. In a real implementation, over the course of a day, $SINR$ samples for some of these solutions will be taken. For those solutions with insufficient $SINR$ samples, such samples must be gathered during the running of the optimisation algorithm, possibly by generating traffic between nodes for this purpose. The algorithm will start with a randomly generated feasible candidate CA and iteratively improve on that solution. For the results presented here, we have used simulation in ns3 for evaluation purposes, because this provides a controlled environment for ease, efficiency, clarity and cost-effectiveness of experimentation.

4.1 Generating Feasible Candidate Solutions

While we have used the graph analogy for this problem, it is not a simple graph colouring problem. One of the added complexities that distinguishes this problem from normal graph colouring is the interface constraint in Eq. (2). Another is that connectivity must be maintained between links through ensuring Eq. (6) is true and having common channels assigned to link nodes, while collisions should be avoided. In all of the metaheuristic optimisation methods we need to generate a set of possible solutions, that is, the solution space. We can either generate each solution and check for feasibility afterwards, or ensure feasibility within the generation procedure. Our method does the latter. We have developed a simple novel algorithm to generate candidate solutions that are feasible. A feasible solution is one that satisfies the interface constraint while using only allowed channels at each node. In this algorithm, we attempt to allocate channels to all links in the network. However, this might not be possible. Therefore, we allocate DSA channels to as many links as possible out of the full set. To ensure connectivity on the remaining links, Wi-Fi is used. This algorithm is outlined in Algorithm 1.

4.2 Optimisation

The objective is to find a link→channel mapping (A) that maximises the total average $SINR$ in the network. For each considered solution, all nodes scan the environment for a period of time and obtain a large set of sample $SINR$ values for traffic flow through a particular link→channel mapping for a particular interference environment. We then use the average of these values in the cost. In Simulated Annealing, we desire that the objective function (so-called "energy" value E) incorporates these $SINR$ samples in a way that the desired result is the lowest cost. Hence, the selected cost is based on $1/SINR$. For Genetic Algorithm, where we use a fitness value, this is the normalised average of $SINR$. All results are shown as the scaled inverse $SINR$ for direct comparison between optimisation methods.

Simulated Annealing. The cost per iteration j is shown in Eq. (7), where V is the number of nodes, n is the number of $SINR$ samples per node, and $\overline{SINR}$ is the average of the $SINR$ measurements per node.

$$E_j = \frac{1}{V}\sum_{v=1}^{V}\left[\frac{1}{n}\sum_{i=1}^{n}\frac{1}{SINR_j(i)}(v)\right] = \frac{1}{V}\sum_{v=1}^{V}\frac{1}{\overline{SINR_j(v)}} \tag{7}$$

In SA, the change in cost every iteration is used to decide whether to accept or reject the particular CA solution. If the new solution is better than the previous solution, i.e., has a lower cost, the new solution is always accepted. However, if the new CA has a higher cost, this worse solution is accepted with a probability h given by Eq. (8). This is realised by selecting a random value a between 0 and 1 and evaluating if $a < h$.

$$h = \exp(-\frac{\Delta E}{kT}) = \exp(-\frac{E_j - E_{j-1}}{k \cdot T_j}) \tag{8}$$

where
E_j is the "energy" or cost at iteration j, given by Eq. (7)
k is Boltzmann's constant ($1.380649 \times 10^{-23} JK^{-1}$)
T_j is the temperature at iteration j

Algorithm 1: Initial channel allocation

```
Data: C = allowed channel set, c = single channel in C, n_i = node number i,
      L = set of links, l = link in L, A = channels assigned = ∅, r = number
      of interfaces per node=2
Result: complete A(l) ∀ l ∈ L
for l = (n_i, n_j) ∈ L do
    if A(n_i) < r and A(n_j) < r then
        c = random channel ∈ C(n_i) ∩ C(n_j);
        A(n_i) = c;
        A(n_j) = c;
    end
    else if A(n_i) == r and A(n_j) < r then
        {c} = A(n_i) ⋃ C(n_j);
        if {c} ≠ ∅ then
            c = {c} [0];
        end
        else
            c=choose one of A(n_i);
            A(n_j) = c;
        end
    end
    else if A(n_i) < r and A(n_j) == r then
        {c} = A(n_j) ⋃ C(n_i);
        if {c} ≠ ∅ then
            c = {c} [0];
        end
        else
            c=choose one of C(n_j);
            A(n_i) = c;
        end
    end
    else
        both interfaces already assigned channels;
        {c} = A(n_i) ⋃ A(n_j);
        if {c} ≠ ∅ then
            c = {c} [0];
        end
        else
            continue;
        end
    end
    A(l) = c;
end
∀ l unassigned, assign a 5 GHz Wi-Fi channel
```

If $a < h$, the solution is accepted. If not, the solution is rejected. If Eq. (8) always evaluates close to 1, higher cost solutions will always be accepted and the SA algorithm will not converge. Conversely, if Eq. (8) always evaluates to a value very close to 0, almost no "worse" solutions will be accepted and the algorithm will converge prematurely on a local minimum that may be much worse than the true optimum.

A careful balance of temperature ranges and ΔE ranges as well as k-value must be formulated to tune the algorithm appropriately. Boltzmann's constant k could be omitted from this relation (or set to 1) in practice if it makes the probability of accepting a point extremely low, leading to converging on a local minimum. Including or leaving this constant out, is part of the parameter tuning required to ensure the algorithm behaves well. We have omitted k but added another constant to scale the $1/SINR$ values appropriately.

The other parameter tuning required is the selection of the starting temperature and the temperature cooling function. A starting temperature that is too high or a cooling function that decreases too slowly will cause much slower convergence. On the other hand, starting with a temperature that is too low or a cooling function that reduces too quickly may result in converging prematurely. Starting temperature and the temperature cooling function must be adjusted in consideration of the number of iterations the algorithm is expected to run for, or that is considered acceptable. We ran experiments with various cooling functions (e.g., logarithmic and exponential functions) in this work before finding a suitable one: the linear temperature cooling function shown in Eq. (9).

$$T_j = T_{start} - \alpha \cdot j \tag{9}$$

where j is iteration count and α is a constant set to 0.02. We selected this value for α by reversing the calculation (9) for appropriate starting temperature (20) and final temperature (0.1) and the desired number of iterations (1000), and confirming by experimentation that it works well. We start with a lower temperature value of 20, selected by observation of the ΔE values for our problem, and scale the $1/SINR$ values appropriately. With these adjustments, the algorithm is able to converge sufficiently within 1000 iterations.

The neighbour generation procedure whereby a new solution is generated is to shuffle the links randomly and perform Algorithm 1.

Genetic Algorithm. For the GA, we encode a genome also as a link→channel mapping, where the links are all node pairs possible in the mesh and where the condition of Eq. (6) is met. To generate a genome, we randomly shuffle the set of links, randomly shuffle the set of allowed channels, and use Algorithm 1 to generate a feasible genome. We then generate a population by generating a number of genomes. We determined from experimentation that a population size of 20 functions well without excessive computational burden. This population is confirmed as a good choice by [25], who find that a population size of 20 presents less structural bias than populations of 5 or 100 individuals in general.

Both Roulette Wheel selection and Linear Rank selection were implemented. For the Roulette Wheel selection, we generated a piecewise constant probability distribution, where the intervals are 1 + the population size and the weights are the fitness values of the chromosomes in the population. For Linear Rank selection, we sort the chromosomes by their inverse fitness value so that the genome with the highest fitness has the lowest rank (highest number). We then create a piecewise constant probability distribution of the ranks and select two parent chromosomes randomly according to that distribution. It was found that Linear Rank selection outperforms Roulette Wheel selection, so only the results for Linear Rank selection are shown. We select as many parents as the current population and each pair of parents generates two children. The previous generation is eliminated once they reproduce, so the size of the population remains stable.

Once two parents have been selected, the next operator is crossover. The crossover operator randomly selects an index in the genome (a link) greater than the first and smaller than the last, as the crossover point. We then split both parents at this crossover point and generate two new children by joining the first section of the first parent with the second section of the second parent, and the first section of the second parent with the second section of the first parent. Mutation is done with a probability of 0.5, by randomly selecting one link and randomly selecting a new channel for that link, and replacing the currently assigned channel with the new one. The 0.5 probability was found to provide a suitable trade-off between exploration and exploitation for the population size and problem. This follows the findings of [26].

5 Results and Discussion

To evaluate the performance of the algorithms, we have simulated the network using ns3. We have built on top of the existing ns3 classes and created a module for the multi-radio multi-channel WMN simulation with interference, which models the spectrum sensing part of the DSA. Additionally, we created new ns3 modules for each of the optimisation techniques. This code can be reused by others wishing to build on this work or replicate these results [27].

Simulations were run on a T2 large Amazon Web Services EC2 instance with 8 GiB of memory and 2 virtual CPUs, both with Ubuntu 16.04 Operating System, and using ns3-dev version [27] forked from the ns3 GitHub [28].

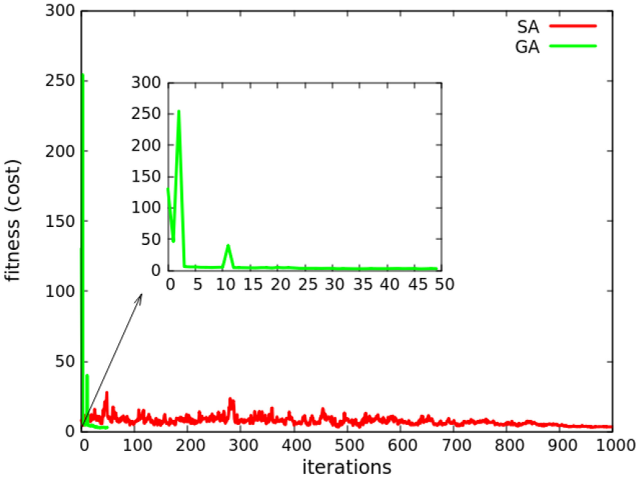
In each iteration of all the optimisation algorithms, the WMN simulation is run for a period of 5 s. This was found to yield sufficient $SINR$ samples for the average to be meaningful. In the mesh simulation, nodes are set up in an equally spaced grid. Each node has two DSA interfaces (representing the DSA band interface). Constant bitrate UDP traffic is generated at the transmit node for every possible link in the network so as to saturate the links. Packets will be received on the other side if there is a common channel between the two nodes and the received signal is above the receive sensitivity. The interference is included in the $SINR$ measurement using ns3's InterferenceHelper class, and interference is counted only if the overlapping packet chunk is above the sensitivity of the receiver. The simulation parameters are given in Table 1. Table 2 compares the mean and standard deviation of CA final costs for 10 runs of SA and GA, and random CAs. We can see that for all presented WMN sizes, SA is significantly better than random allocations (between 120% and 620% better).

Table 1. Parameters used in simulations

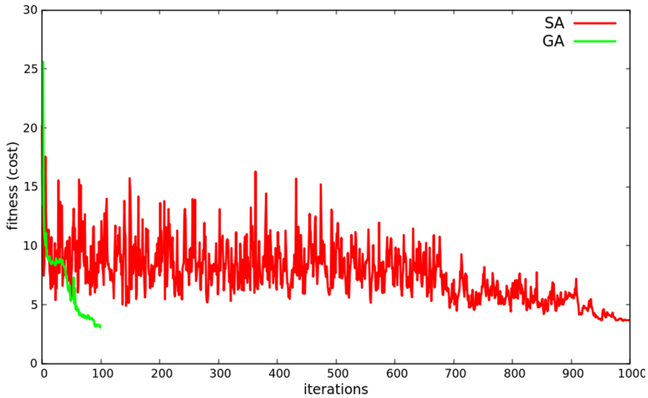
Parameter	**Value**
Mesh network size	9–49 nodes
Number of interfaces	2
Distance between grid nodes	100 m (vertical and horizontal)
Channel bandwidth	10 MHz
Propagation loss model	Friis
Propagation delay model	ConstantSpeed
Packet interval	0.01 s
Packet size	1024 bytes
Error rate model	NistErrorRateModel
Mesh routing algorithm	OLSR

In comparison, GA is significantly better than both random allocations (between 380% and 1268%) and SA (between 16% and 54% better). While the averages improve significantly, the standard deviation also reduces significantly so that the chances of SA or GA producing a substantially worse solution than those shown here are very low.

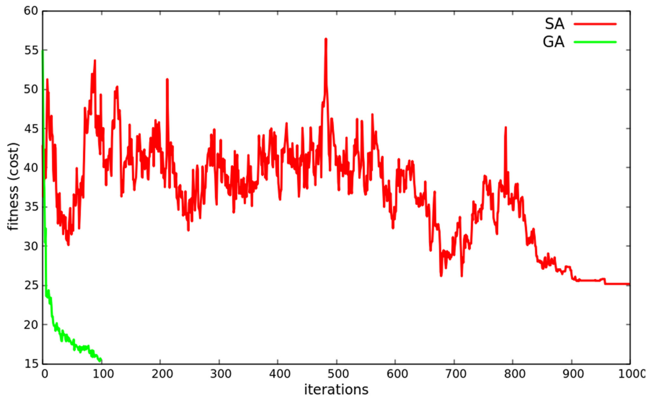
The deviations from the average cost values get smaller over time as the algorithms converge. Regardless of the starting point, different runs start to converge on similar values, especially in the GA case. Figure 3 shows the cost of the solutions found by both SA and GA at each iteration averaged over 10 different runs, for different network sizes. For GA the average population costs for the different runs are averaged. We can observe clearly that GA converges much quicker than SA. Different runs of GA also converge on solutions that are closer than SA (as seen by the smaller standard deviations in Table 2. We can obtain a reasonably good solution using GA within 25 iterations (or even less) for a 9-node WMN. Even for the larger 16 and 49-node mesh networks, the solutions within 50 iterations are better than SA after the same number of iterations. We note, however, that for one iteration of GA, we need to perform 20 runs of the WMN simulation (or perform sampling windows for 20 different CAs), since there are 20 individuals per population. This sampling window is the most time-intensive portion of the optimisation. Hence, 1 iteration of GA is roughly equivalent in time to 20 iterations of SA; so 50 iterations of GA are equivalent to 1000 iterations of SA in time. Still, within the same amount of time, we are able to achieve significantly better results with GA than with SA, although both achieve much better results than CAs.



(a) 9 nodes



(b) 16 nodes



(c) 49 nodes

Fig. 3. Average cost of all runs of SA and GA (average cost of population) per iteration compared

Table 2. Average and standard deviation of cost values for random channel allocations, SA and GA for 10 runs of each

	Random	**SA**	**GA**
Nodes	Mean (SD)	Mean (SD)	Mean (SD)
9	26.0 (±35.8)	3.6 (±1.6)	1.9 (±0.25)
16	28.1 (±35.7)	4.3 (±2.7)	3.6 (±1.0)
49	56.0 (±51.7)	25.4 (±4.9)	11.6 (±5.2)

6 Conclusion

We have presented new methods for channel assignment in wireless mesh networks using DSA. Our work is unique in that it considers the realistic measure of average $SINR$ over the mesh to represent the performance of the channel assignment in the network, and includes a new method for ensuring the feasibility of the solutions according to each node's allowed channels and number of radio interfaces. This method is necessary because different nodes may have different allowed channels. A comparison of different channel allocation algorithms, i.e., random allocations, Simulated Annealing and Genetic Algorithm, was done using simulations in ns3. It was found that the metaheuristic algorithms significantly improve results over random CAs. In our implementation of the problem, we observed that GA performs significantly better than SA, with both lower average cost values, and less variation among final solutions for different runs with the same parameters.

We plan on extending this study to include other metaheuristic optimisation methods. Future work will also include extending our study to include aspects specific to the chosen frequency bands, consider different channel bandwidths and include other propagation and antenna models.

References

1. CBRS, SAS and Spectrum Sharing: The Complete Guide. https://blinqnetworks.com/cbrs-sas-spectrum-sharing-guide/. Accessed 26 July 2021
2. Wi-Fi Alliance®furthers Automated Frequency Coordination specification and compliance development to accelerate Wi-Fi 6E. https://www.wi-fi.org/news-events/newsroom/wi-fi-alliance-furthers-automated-frequency-coordination-specification-and. Accessed 26 July 2021
3. iNethi Homepage. https://www.inethi.org.za/. Accessed 24 July 2021
4. Zenzeleni Community Networks Homepage. https://zenzeleni.net/. Accessed 24 July 2021
5. AlterMundi Homepage. https://altermundi.net/. Accessed 24 July 2021
6. Belli, L.: Community Networks: The Internet by the People, for the People, 1st edn. FGV Direito Rio, Internet Society, Rio de Janeiro (2017)
7. Hale, W.: Frequency assignment: theory and applications. Proc. IEEE **68**(12), 1497–1514 (1980)

8. Kirkpatrick, S.: Optimisation by simulated annealing. Science **220**(4598), 671–680 (1983)
9. Nourani, Y., Andresen, B.: A comparison of simulated annealing cooling strategies. J. Phys. A Math. Gen. **31**(41), 8373–8385 (1998)
10. Geweke, J., Tanizaki, H.: Bayesian estimation of state-space models using the Metropolis-Hastings algorithm within Gibbs sampling. Comput. Stat. Data Anal. **37**(2001), 151–170 (2001)
11. Chen, Y., Chen, C.: Simulated annealing for interface-constrained channel assignment in wireless mesh networks. Ad Hoc Netw. **29**(2015), 32–44 (2015)
12. Sridhar, S., Guo, J., Jha, S.: Channel assignment in multi-radio wireless mesh networks: a graph-theoretic approach, In: First International Communication Systems and Networks and Workshops, Bangalore, India, vol. 2009, pp. 1–10 (2009)
13. Pal, A., Nasipuri, A: JRCA: a joint routing and channel assignment scheme for wireless mesh networks. In: Conference Proceedings of the 30th IEEE International Performance, Computing, and Communications Conference, pp. 1–8 (2011)
14. Balusu, N., Pabboju, S., Narsimha, G.: An intelligent channel assignment approach for minimum interference in wireless mesh networks using learning automata and genetic algorithms. Wireless Pers. Commun. **106**(3), 1293–1307 (2019). https://doi.org/10.1007/s11277-019-06214-3
15. Cheng, H., Yang, S.: Joint QoS multicast routing and channel assignment in multiradio multichannel wireless mesh networks using intelligent computational methods. Appl. Soft Comput. J. **11**(2011), 1953–1964 (2011)
16. Ding, Y., Huang, Y., et al.: Channel assignment with partially overlapping channels in wireless mesh networks. In: Proceedings of the 4th Annual International Conference on Wireless Internet, pp. 1–9. ICST, Maui (2008)
17. Zhuang, X., Cheng, H., Xiong, N.: Channel assignment in multi-radio wireless networks based on PSO algorithm. In: 2010 5th International Conference on Future Information Technology, FutureTech 2010 - Proceedings, pp. 1–6 (2010)
18. Shawkat, A., et al.: On the use of particle swarm optimisation techniques for channel assignments in cognitive radio networks. In: Multidisciplinary Computational Intelligence Techniques: Applications in Business, Engineering, and Medicine, pp. 202–214, Information Science Reference, United States of America (2012)
19. Chakraborty, M., Chowdhury, R., Basu, J., Janarthanan, R. Konar, A.: A particle swarm optimisation-based approach towards the solution of the dynamic channel assignment problem in mobile cellular networks. In: IEEE Region 10 Annual International Conference, Proceedings/TENCON, pp. 1–6 (2008)
20. Subramanian, A.P., Gupta, H., Das, S.R.: Minimum interference channel assignment in multi-radio wireless mesh networks. In: 2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, pp. 481–490 (2007)
21. Qin, A., et al.: Opportunistic scheduling and channel allocation in MC-MR cognitive radio networks. IEEE Trans. Veh. Technol. **63**(7), 3351–3368 (2014)
22. Maliwatu, R.: A new connectivity strategy for wireless mesh networks using dynamic spectrum access. Ph.D. thesis, Cape Town (2020)
23. AirMatch. https://www.arubanetworks.com/techdocs/ArubaOS_81_Web_Help/Content/ArubaFrameStyles/ARM/mCell.htm. Accessed 31 July 2021
24. Cho, S.: $SINR$-based MCS level adaptation in CSMA/CA wireless networks to embrace IoT devices. Symmetry **9**(10), 236–253 (2017)
25. Kononova, A.V., et al.: Structural bias in population-based algorithms. Inf. Sci. **298**(2015), 468–490 (2015)

26. Deb, K., Agrawal, S., et al.: Understanding interactions among genetic algorithm parameters. Found. Genet. Algorithms **5**(5), 265–286 (1999)
27. ns-3-dev-git. https://github.com/natzlob/ns-3-dev-git/tree/mrmc-fresh. Accessed 31 July 2021
28. ns-3-dev-git. https://github.com/nsnam/ns-3-dev-git. Accessed 31 July 2021