

University of Würzburg
Institute of Computer Science
Research Report Series

**On the Usability of Genetic Algorithms for
Wireless Mesh Network Planning and
Optimization**

Rastin Pries, Dirk Staehle, Marieta Stoykova

Report No. 451

December 2008

University of Würzburg
Department of Computer Science
Am Hubland, D-97074 Würzburg, Germany
pries@informatik.uni-wuerzburg.de

On the Usability of Genetic Algorithms for Wireless Mesh Network Planning and Optimization

Rastin Pries, Dirk Staehle, Marieta Stoykova

University of Würzburg
Department of Computer Science
Am Hubland, D-97074 Würzburg, Germany
pries@informatik.uni-wuerzburg.de

Abstract

Wireless Mesh Networks (WMNs) are a promising technology to provide broadband wireless connectivity for the end user. They offer more flexibility compared to traditional networks but on the expense of a complex structure. Thus, planning and optimization of WMNs is a challenge. In this paper we focus on routing and channel assignment in WMNs for throughput maximization using genetic algorithms. Genetic algorithms provide a good solution of large-scale WMNs in relatively small computation time. We propose different genetic operators which are adapted for the planning and optimization of WMNs. The results prove the effectiveness of the genetic operators and show the advantages of a genetic optimization.

1 Introduction

The wireless mesh networking paradigm is a very promising extension of the common wireless technologies. It is a new concept which could supply the demand for more coverage, bandwidth, and mobility. Different from traditional wireless networks, a node in a WMN uses several hops to connect to the Internet.

The complex multi-hop structure of WMNs induces the need to investigate a large number of possible network configurations. Thereby, two main approaches offer different ways for planning and optimization of wireless networks. These are *Linear Programming* (LP) and *Genetic Algorithms* (GA). In this paper we use genetic algorithms.

Genetic algorithms are based on the idea of natural evolution and are used to solve optimization problems by simulating the biological cross of genes. A randomly created population of individuals represents the set of candidate solutions for a specific problem. The genetic algorithm applies a so called "fitness function" on each individual to evaluate its goodness and to decide whether to keep it in the new population. However, only the selection without any other operation will lead to a local optima. Therefore, two operators, crossover and mutation, are used to create new individuals.

In this paper we consider the planning issues routing and channel allocation. Based on the gene information, the fitness function of our GA evaluates the max-min fair throughput allocation in every individual. The individuals estimated as best solutions of the current population are replicated in order to create new unexplored gene combinations. This genotype reproduction is done by the genetic operators crossover and mutation. We present different implementations of the genetic operators and investigate their impact on the evolution. Thereby, we consider their effectiveness depending on various network topologies and explain the interactions between crossover and mutation in particular cases. In this way, we explore the attitude of genetic algorithms for planning and optimization of wireless mesh networks.

The remainder of this work is organized as follows. In Section 2 the work related to wireless network planning is reviewed. This is followed by Section 3 presenting genetic algorithms in general and our modifications in particular. In Section 4 the performance of the different genetic operators are shown. Finally, Section 5 concludes this paper.

2 Related work

Wireless mesh networks have awakened the interest of various researchers and Internet providers. Hence, a number of papers have been published on the problem of planning WMNs and estimating their performance.

Sen and Raman [1] introduce a variety of design considerations and a solution approach which breaks down the WMN-planning problem into four tractable parts: topology search, tower height assignment, antenna assignment, and power assignment. These sub-problems are inter-dependent and are solved by heuristics in a definite, significant order. The evaluations of the presented algorithms show that they are able to generate long-distance WiFi deployments of up to 31 nodes in practical settings.

Other related work [2–4] deal with creating a wireless mesh network model, planning its parameters, and evaluating the solutions via Linear Programming. He et al. [2] proposed mechanisms for optimizing the placement of integration points between the wireless and wired network. They developed algorithms to provide best coverage by making informed placement decisions based on neighborhood layouts, user demands, and wireless link characteristics. Amaldi et al. [3] propose other planning and optimization models based on LP. Their aim is to minimize the network installation costs by providing full coverage for wireless mesh clients. Thereby, traffic routing, interference, rate adaption, and channel assignment are taken into account. Another cost minimizing, topology planning approach is presented by So and Liang [4]. They propose an optimization framework which combines a heuristic with Benders decomposition to calculate the minimum deployment and maintenance cost of a given heterogeneous wireless mesh network. Furthermore, an analytical model is presented to investigate whether a particular relay station placement and channel assignment can satisfy the user demands and interference constraints.

Badia et al. [5] use genetic algorithms for a joint routing and link scheduling for WMNs. They found out that genetic algorithms solve their scenarios reasonably well, and also scale, whereas exact optimization techniques are unable to solve larger topologies. Vanhatupa et al. [6] use a genetic algorithm for the WMN channel assignment. Capacity, AP fairness, and coverage metrics were used with equal significance to optimize the network. The routing was fixed, using either shortest path routing or expected transmission times. They have shown a capacity increase of 131 % with the channel assignment optimization. In contrast to Vanhatupa et al. and Badia et al., we are maximizing the throughput in a multi-channel, multi-radio WMN by both channel and route assignment.

3 Wireless Mesh Network Planning and Optimization via Genetic Algorithms

3.1 Problem Formulation

To explain our planning and optimization approach, we define the considered network parameters and constraints. We start with a simple definition of a wireless mesh network, its nodes and connections.

A mesh network \mathcal{N} consists of a set Θ of N nodes n_1, \dots, n_N and a set \mathcal{L} of links, which are responsible for the routing possibilities in this network. Every node can either be a user locations $u_i \in \mathcal{U}$ or a gateway $g_j \in \mathcal{G}$. This means that

$$\Theta = \mathcal{U} \cup \mathcal{G} \text{ and } \mathcal{U} \cap \mathcal{G} = \emptyset. \quad (1)$$

The links $\ell_{ij} = (n_i, n_j)$ in the network are meant to work on definite frequency channels depending on the used technology. Let γ_{ij} be the function, which maps the Signal to Interference Ratio (SIR) received at node n_i for transmissions by node n_j , to the maximum throughput physically

achievable between them. The nodes n_i and n_j can be connected or simply "hear" each others transmissions if γ_{ij} is greater than or equal to a certain threshold γ_{conn} . Accordingly, we define a conflict graph $\mathcal{C}_{\mathcal{N}}$, which consist of all possible connections in the network \mathcal{N} and is equal to

$$\mathcal{C}_{\mathcal{N}} = \{(n_i, n_j) | \gamma_{ij} \geq \gamma_{conn}\}. \quad (2)$$

The conflict graph is obviously symmetrical and is the basis for later computation of the collisions between sending or receiving nodes. In our work we consider mesh networks with a predefined static routing scheme which we determine in a routing graph $\mathcal{R}_{\mathcal{N}}$. This routing graph consists of the same nodes like the conflict graph $\mathcal{C}_{\mathcal{N}}$. Though, the set of links contained in $\mathcal{R}_{\mathcal{N}}$ is definitely a subset of the connections in the conflict graph. The routing scheme of $\mathcal{R}_{\mathcal{N}}$ defines for every user u_i a static path p_i assigning the way for packets to and from the gateway g_i , which the node u_i is associated to. The path p_i is an ordered set containing edges of the connectivity/routing graph.

The goal of our evaluating algorithm is to find a max-min fair allocation of the throughput resources within the mesh network. Thereby we consider the channel allocation over the routing scheme and the consequential interferences.

3.2 Encoding

Our simple representation includes only one possible connection per user and is easy to handle and evaluate. This connection denotes the next hop which the traffic of the considered node has to go in order to reach the gateway. Thus, we always imply only one possible path toward a gateway for the packets of each user node. The routing information is coded in the individuals structure and does not need extra verification. Besides the routing information, the channel allocation is also included in the list representation. Figure 1 illustrates an example for the routing and channel encoding.

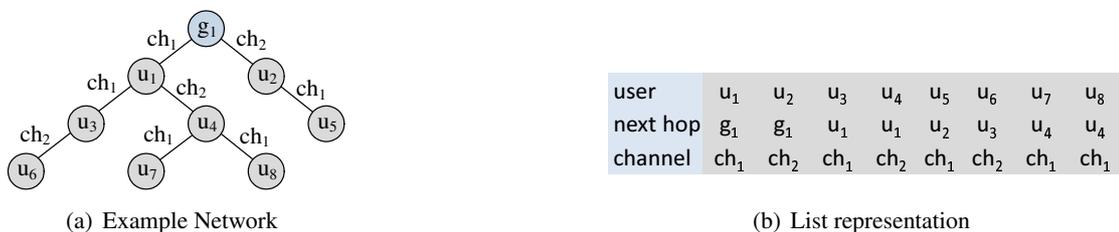


Figure 1: Example network and its list representation for the routing genes and the channel allocation

3.3 Crossover Types

3.3.1 2-Point Crossover

The 2-Point Crossover is the simplest realization of the genetic cross. It is a common exchange of gene subsets, which are random chosen sublists of the individuals genotype. The start and end intersection points denoting the range of the sublist are chosen every time when the operator is applied.

Figure 2 shows an example of the 2-Point Crossover between two network solutions. The intersection points lay here at the second and fifth position in the individuals code and border the sublist of genes for the nodes u_2, u_3, u_4 , and u_5 . These denote the area which will be exchanged during the crossover.

An important characteristic of this reproduction approach is the induction of solutions, which contain user locations with no connection to any gateway. This happens due to the unregulated and absolutely arbitrary selection of the gene subset which is meant to be exchanged.

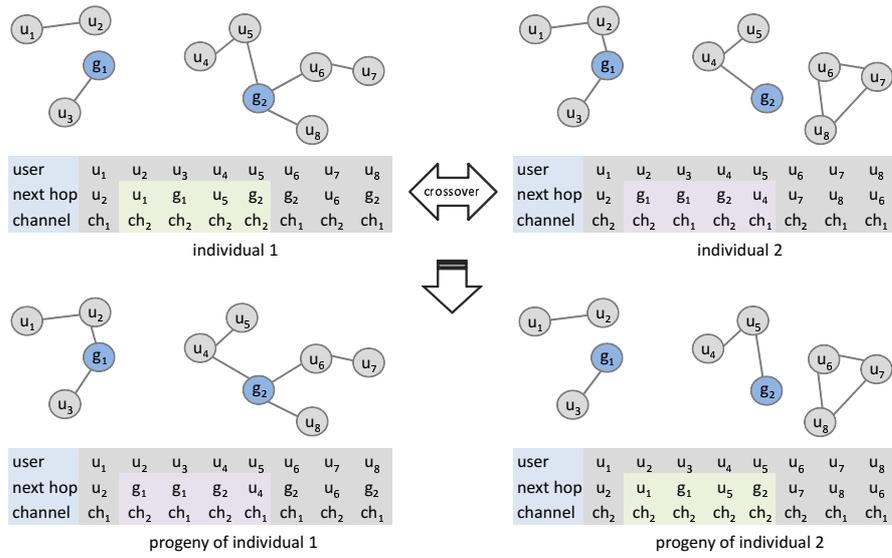


Figure 2: 2-Point Crossover between two network solutions

3.3.2 Cell Crossover

The Cell Crossover represents a connectivity dependent exchange of genes. The code information is crossed by considering only the positions of nodes belonging to the same subgraph component. The first step of this approach is to select a random value depending on the number of gateways in the solution. Its value denotes the number of connectivity components exchanged by the crossover. Thereby, only subgraphs with a gateway as root are allowed to be chosen. These are the cells \mathcal{S}_{g_i} of the network. Figure 3 shows an example for the crossover of two solutions which blue nodes are meant to be the network gateways. For the sake of readability, we do not denote the used channels over the network links in the picture. Though, when we talk about the gene cross of the routing information we actually always exchange also the corresponding channels. Furthermore, the colored areas denote the cell which is being exchanged and it is easy to see that its connections and number of nodes differ in the two individuals.

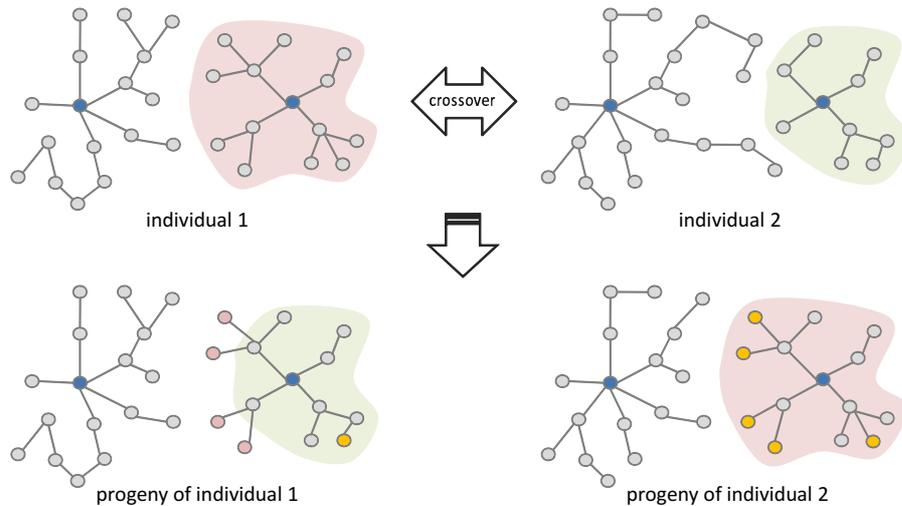


Figure 3: Cell Crossover between two network solutions

In order to create the progeny of individual 1, we consider the connections of the nodes from the green area of individual 2. During the crossover, their next neighbors are copied in the code scheme of the first progeny. The remaining network nodes keep the next hops from the genotype of individual 1. This creates the progeny of individual 1 where the left cell is the same as in its forefather. Though, the right exchanged cell includes an orange node, which connection is modified due to the cell copying and four pink nodes which are not affected by the crossover and keep their next hops. The same procedure is executed in order to create the progeny of individual 2. We consider the connections from the pink area in individual 1 and copy their connections in the scheme of the offspring. This emerges five orange nodes with modified next hop and changes also the structure of the left cell. Hence, the progeny of individual 2 contains the exact cell connections from the right subgraph of individual 1 and keeps the remaining genes of its forefather.

The Cell Crossover is a regulated exchange of genes which we developed in order to emerge more reasonable solutions during the evolution. With this approach we increase the possibility of creating an individual with high performance through the cross of two others containing good sub-solutions. Hence, in this case individuals with medium fitness can also be perfect forefathers.

3.3.3 Subtree Crossover

The Subtree Crossover is similar to the Cell Crossover. It also exchanges connectivity components considering the network structure. Though, these are particular subtrees and not whole gateway cells. The crossover operator chooses a random position of the individuals code and fixes the subtree with the node at the chosen position as root. The code of all locations contained in the subtree is then copied and kept. Thereby, the channel allocation is exchanged together with the routing information. This procedure can be executed more than ones between two solutions in order to create one progeny.

Figure 4 shows an example for the Subtree Crossover with two chosen positions which corresponding subtrees are meant to be exchanged. The orange colored nodes denote the edges of the connectivity components. They determine the positions which information is going to be modified in the code of the other solution in order to create its progeny. If we observe the individual 1 and its progeny, we see that the left set of nodes belongs in the resulting solution to a different gateway cell as in the original one. This is due to the copied information from individual 1, where the next hop for one of the orange nodes is the left gateway and not the right gray colored neighbor. The same procedure is executed in order to create the offspring of individual 2. Thereby, the fixed information from the first solution is copied in the genotype and consequently in the phenotype of the second progeny. Hence, the crossover of both individuals results on the one hand in a solution with good routing genes (progeny of individual 1) and on the other hand in an offspring with medium routing performance and still existent potential for further reproduction (progeny of individual 2).

This type of crossover gives us the opportunity to create new solutions consisting of reasonable connections and to make fine changes by keeping the tree structure of the network cells. It offers the evolution more chances than the Cell Crossover. Moreover, the smaller number of exchanged positions allows a better approximation of solutions.

3.4 Mutation

The mutation is a very important part of the evolution process. It denotes the arbitrarily modification of genes and generates a variety of solutions. The mutation operator changes some randomly chosen positions of the routing and channel allocation code by writing new information taken from a set of legal entries. As already mentioned, the genetic algorithm creates the first population by making connections depending on the potential neighbors relation. This is also the set of nodes,

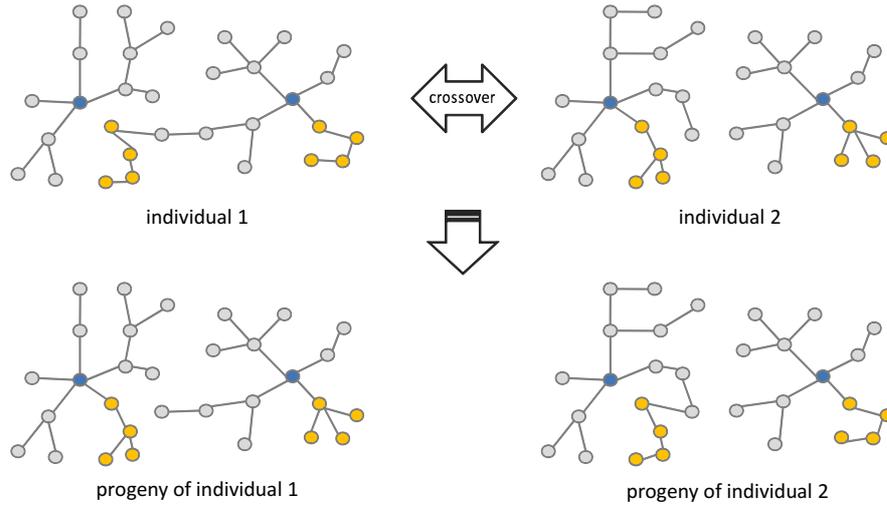


Figure 4: Subtree Crossover between two network solutions

which the operator uses to choose a new next hop for the user at the mutated position. Thereby, it selects only the potential neighbors which would not cause the creation of cycles and would not harm the tree structure of the solution if they become the next hop of the mutated node. Hence, we developed a mutation operator which still makes random modifications but also considers the network connectivity.

Besides the routing mutation, we also apply the mutation operator for the channel allocation. The modified positions are chosen independent from the routing mutation. In a list of available channels the operator chooses an arbitrary one, which differs from the current allocation. This procedure is executed several times per individual and aims to achieve a better channel allocation caused by coincidence.

Figure 5 shows the mutation of the routing information at three network nodes. The modified users are denoted through the orange color. The lists of potential neighbors of this nodes are:

$$\begin{aligned}\mathcal{P}(u_4) &= \{u_2, u_3, u_5, u_6\} \\ \mathcal{P}(u_8) &= \{g_1, u_6, u_{10}, u_{11}, u_{17}\} \\ \mathcal{P}(u_9) &= \{u_7, u_{10}\}\end{aligned}$$

The mutation operator chooses a new next hop for each orange node from a list of possible neighbors. Thereby, there are some constraints which need to be considered:

- the new next hop of u_4 toward the gateway can be neither u_2 nor u_3 because this would create a connectivity component consisting of the three nodes without connection to the gateway.
- for the same reason as at node u_4 , the user u_8 can not be connected to u_{10} or u_{11} . The new next hop must be either u_{17} or g_1 .

The new connections can be observed in the right part of Figure 5, where the mutated solution offers considerable differences in the routing structure.

3.5 Fitness Function

The evaluation part of the optimization called "fitness function" is the heart of the genetic algorithm. Based on the fitness value, the GA decides which individuals should be kept in the new population. Hence, it rates the performance of the genes and allows only the best to be replicated.

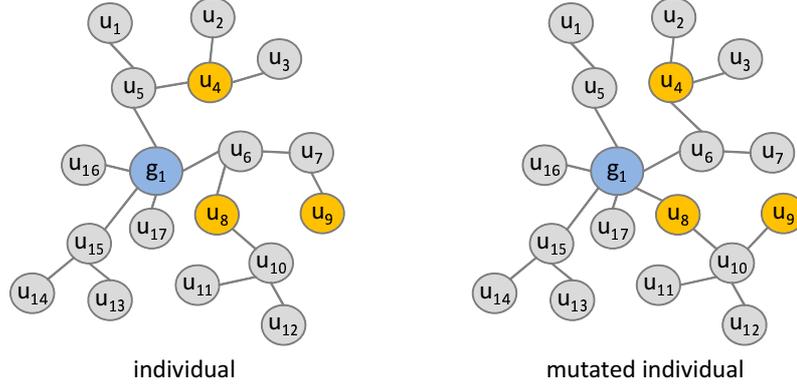


Figure 5: Mutation of the routing information

3.5.1 Max-Min Fair Share Throughput Allocation

The evaluation algorithm fairly distributes the network resources between the users and depending on this allocation calculates the fitness of the solution.

In order to reach a max-min fairness, we consider the collision domain of each network link and the number of transmissions contained in it. For this purpose, we define the number of transmissions going over the connection between two nodes n_i and n_j and denote it with λ_{ij} . Thereby, we consider only one direction – toward the corresponding gateway. We call λ_{ij} also the load of l_{ij} because it specifies the number of flows carried by this link. Figure 6 shows an example for the determination of the link loads. The λ_{ij} value yields from the number of nodes in the opposite flow direction which are the flow sources.

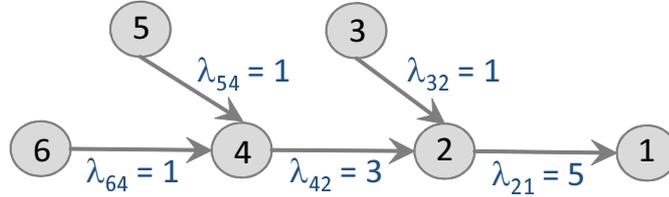


Figure 6: Link load calculation depending on the carried number of flows

The load of the collision domain \mathcal{D}_{ij} is then denoted by Λ_{ij} and contains the loads of all links from \mathcal{D}_{ij} as described in Equation 3.

$$\Lambda_{ij} = \sum_{l_{zk} \in \mathcal{D}_{ij}} \lambda_{zk} \quad (3)$$

In order to fairly supply all network users, we share the time resources among all transmissions taking place within the collision domains of the corresponding links. Thereby, we take the capacities c_{ij} and the loads λ_{ij} of the participating connections into account. The throughput t_{ij} of the link l_{ij} is then defined as follows:

$$t_{ij} = \frac{1}{\sum_{l_{zk} \in \mathcal{D}_{ij}} \frac{\lambda_{zk}}{c_{zk}}} \quad (4)$$

Now, we follow the principle of the max-min fair share algorithm and fix the resources for the link with minimum throughput. We call it the bottleneck of the network and denote it with l_{uv} .

The time resources occupied by ℓ_{uv} for supplying its λ_{uv} flows can now be calculated as shown in Equation 5. They differ depending on the link for which they are calculated. Such links can be by ℓ_{uv} bottlenecked connections or parent-links on the path toward the gateway.

$$\rho_{uv}(\ell_{ij}) = \lambda_{uv} \cdot \frac{t_{uv}}{c_{ij}} \quad (5)$$

Having computed the occupied resources and fixing the bottlenecked connections, we have to consider that a part of the time is now reserved. Hence, we must take this into account in a new calculation of the links throughputs. Moreover, we need to update λ_{ij} by subtracting the flows supplied through the bottleneck. When all network resources are refreshed, we fix the next link with the smallest throughput. In this way we flood the network over and over with new information until there are no more unfixed connections to be set as bottlenecks.

After the resources of the evaluated network are distributed, we estimate the fitness of \mathcal{N} by using the allocated throughputs. The fitness function of the evaluation yields to a value, which stands for the satisfaction of the users and the fairness of the resource allocation. We use the minimal throughput. However, other objective function are also possible.

The function $\min(\mathcal{R}_{\mathcal{N}})$ calculates the minimum throughput of all links from the routing scheme $\mathcal{R}_{\mathcal{N}}$. The value yielded by the fitness function is crucial for the further evolution of the evaluated network solution. It is deciding for the individuals replication and its gene diffusion in the next population.

3.6 Selection Principle

After the evaluation of every population, we select a set of solutions, which have the highest fitness of all and keep them in the new generation. This procedure accounts for the elitist type of selection. Furthermore, we use the set of the best solutions to build new ones by exchanging and mutating their genes. Thereby, the number of progenies per individual is proportional to its fitness. It is a function of the selection probability of this solution and the number of needed new individuals. Let n be the size of the population, m the number of best forefathers to be kept in the next generation, and $s(i)$ the selection probability for the individual i . The number of progenies of i is then defined as

$$\eta(i) = (n - m) \cdot s(i). \quad (6)$$

The selection probability of the individual i is described by the relation between the fitness of this solution and the sum of the fitness values of all individuals from its population. Equation 7 shows the definition of its function.

$$s(i) = \frac{f(i)}{\sum_j^n f(j)} \quad (7)$$

This fitness dependent determination of the offspring number accounts for the deterministic selection. It motivates the higher reproduction of genes from solutions with better performance. Thereby, for the next generations we create individuals by making new gene combinations on the basis of more and more reasonable forefathers.

4 Performance Evaluation

The performance of a genetic algorithm is an umbrella term for the efficiency of the genetic operators, the effectiveness of the fitness function, and the collaboration of some other parameters affecting the evolution. In this section we evaluate the influence of the three different crossover types and the mutation operator on the minimal throughput.

4.1 Simulation Settings

For the creation of the results presented in this section, we use two scenarios introduced in Table 1. The first one consists of 2 gateways and 71 users distributed over an area of 2 km to 1.2 km. Thereby, the minimal distance between users is 60 m and between the two gateways 700 m. For the sake of readability we call this topology **G2U71**. The second city contains a smaller number of users and a larger number of gateways. We choose this clearly different topology in order to show the influence of the crossover operators depending on the number of nodes. The 38 subscribers and 6 gateways of the second city are allocated in an area of 1.5 km to 1 km. The minimal distance between users is 60 m and between gateways 450 m. We call this topology **G6U38**.

The differences in the settings of the two configurations depend on the used topology of the corresponding scenario. Due to the larger number of nodes contained in G2U71, we configure Scenario S1 with more mutations and more exchanged subtrees than Scenario S2. Thereby, we keep the relation between crossover and mutation at a definite level suitable for the investigation of the genetic operators.

Table 1: Simulation Scenarios

Parameter	Scenario S1	Scenario S2
city topology	G2U71	G6U38
population size	150	150
elite set size	50	50
number of generations	400	400
crossover type	Subtree CO Cell CO 2-Point CO	Subtree CO Cell CO 2-Point CO
number of crossed subtrees	7	5
number of mutations	20	10
fitness function	$f(\mathcal{N}) = \min(\mathcal{R}_{\mathcal{N}})$	$f(\mathcal{N}) = \min(\mathcal{R}_{\mathcal{N}})$

For the crossover investigation we use the minimal flow throughput of the network as the optimization value which decides the fitness of the solution. This is accomplished using the objective function $f(\mathcal{N})$. However, by testing the 2-Point Crossover, we use an extension of $f(\mathcal{N})$. This is the function $\tilde{f}(\mathcal{N})$ presented in Equation 8:

$$\tilde{f}(\mathcal{N}) = f(\mathcal{N}) - \text{conless}(\mathcal{R}_{\mathcal{N}}) \quad (8)$$

With $\tilde{f}(\mathcal{N})$ we optimize the minimal throughput and penalize the solution for bad connectivity. The $\text{conless}(\mathcal{R}_{\mathcal{N}})$ term contains the number of users without connection to any gateway. This is essential for the 2-Point Crossover which exchanges random parts of the solutions code without considering the network connectivity. Hence, the minimal throughput contained in $f(\mathcal{N})$ presents the positive costs of the network while $\text{conless}(\mathcal{R}_{\mathcal{N}})$ stands for the penalty costs. As already mentioned, for the investigation of the Cell and Subtree Crossover we simply use $f(\mathcal{N})$ as objective function. In order to make their results comparable with this of the 2-Point Crossover, we present only the positive costs of the solutions achieved with this operator in the following figures.

The fixed parameters affect only the characteristics of the network connections. They are listed in Table 2 and denote the used carrier frequency, the channel bandwidth, and the available channels. They decide to some extent the performance of the user connections in a network solution but they do not affect the effectiveness of the genetic algorithm which we investigate in this section. Therefore, we do not consider their impact on the resulting solutions.

Table 2: General Parameter Settings

parameter	value
carrier frequency	3500 MHz
channel bandwidth	7 MHz
available channels	3500 MHz; 3510 MHz
antenna power	25 dBm
pathloss model	WiMAX urban macrocell model

4.2 Population Evolution

The investigation of the population evolution is an important consideration needed to demonstrate the effectiveness of the genetic algorithm. The growth of the fitness of the new generations must be observable in order to prove the correct functionality of the GA. Thereby, different genetic operators can affect the evolution by more or less stimulating it but none of them should disturb it.

Figure 7 shows the populations fitness growth during the evolution for different crossover types. For generating the results of Figure 7, we used Scenario S1 from Table 1. The x-axes show the individuals sorted by fitness population while the y-axes present the minimal flow throughput of the solutions.

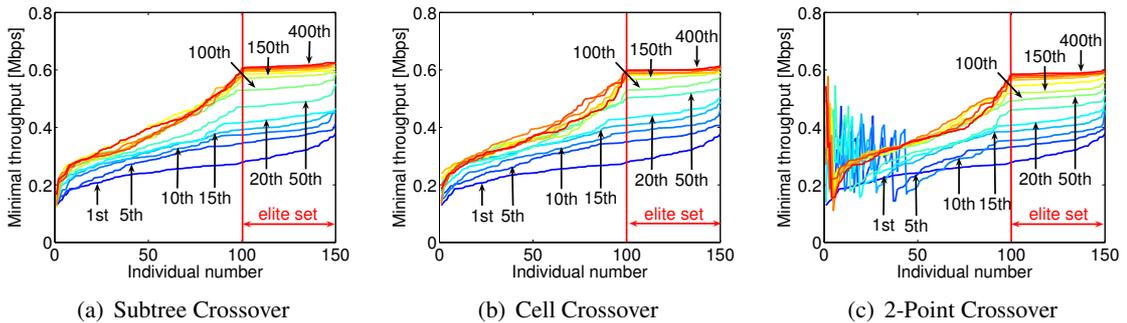


Figure 7: Generations progress by Subtree, Cell, and 2-Point Crossover tested on the G2U71 topology

As already mentioned, the fitness values are in this investigation not comparable, due to the penalty costs used for the 2-Point Crossover. Hence, we consider only the minimal throughput of the solutions which presents in all scenarios their positive costs.

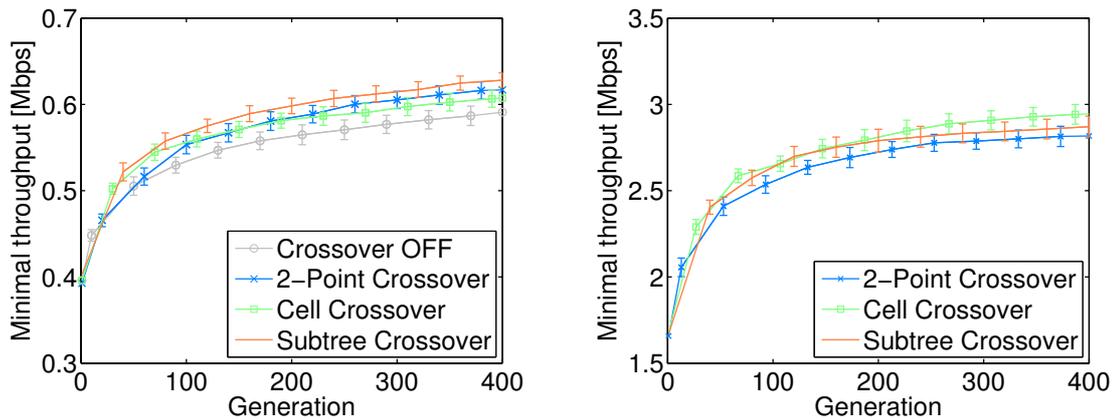
The particular subfigures of Figure 7 show the evolution of the generations 1 to 20 in steps of 5 generations, and 50 to 400 in steps of 50 generations. In all subfigures we can observe how the minimal throughput of the elite set, which size is set to 50, grows with every next generation. Therewith grows the fitness of the solutions. This is due to the selection principle which keeps the forefathers of the prior generation in the next one. The elite selection approach emerges new populations which elite set is definitely better than the previous one. Furthermore, the three evolution simulations start with the same population. This can be observed at the curve of the first generation, which is the same in all subfigures. Then, the curves of the low generations lie clearly over each other besides in Figure 7(c). However, using the 2-Point Crossover emerges the creation of networks containing not connected user nodes. This leads to high penalty costs and estimates bad fitness for a solution with high minimal flow throughput. Hence, the complexity of the curves at the left hand side in Figure 7(c) is comprehensible.

The higher the generation number, the smaller the fitness growth. This behavior is observable for all crossover types. The slowdown of the evolution is caused by two reasons. The first one is the similarity of the individuals due to the reproduction of similar or equal genes leading to better fitness. The second reason is the need of small and selective changes to prove the solutions fitness, which is hard to achieve accidentally. Though, the speed of the evolution depends also on the topology structure in combination with a suitable crossover principle. In the next section, we compare the performance of the three crossover operators depending on the number of users and gateways in the network.

4.3 Effectiveness of the Crossover Operator

In this section, we observe the behavior of the best individual of every generation. We do this in order to see if the evolution proceeds absolutely randomly or if there exists a coherency in the performance of a scenario applied over different seeds. We also want to find out if there is an interaction between the efficiency of the crossover types depending on the city topology.

The results for both scenarios from Table 1 and are presented in Figure 8. Figure 8(a) shows the evolution of the best individual during 400 generations with different crossover types and for not using the crossover operator at all for the G2U71 topology. Thereby, it illustrates the results of 20 seeds with 95% confidence intervals.



(a) Comparison of the crossover types on the topology G2U71

(b) Comparison of the crossover types on the topology G6U38

Figure 8: Effectiveness of the crossover operator

The figure includes a high number of user nodes which are distributed in the coverage cells of only two gateways. This emerges deep tree structures in the solutions and the users have long ways over multiple hops toward the corresponding gateway. Such network structures seem to be crucial for the effectiveness of the crossover types. We can observe that the subtree crossover leads to a better solution than the other two crossover types and its curve stays over the others during the whole evolution. The better performance of the subtree approach is a result of the exchange of small connectivity components which causes reasonable gene variations without disturbing the tree structure. The other two crossover types show a lower performance whereby the unregulated 2-Point Crossover even overtakes the intelligent cell exchanging approach. This happens due to the small number of gateways which causes the cross of only one cell per new progeny and quickly makes the individuals of a population very similar. The overlapping of the confidence intervals is caused by the seed dependency. The initial population contains randomly created individuals

which can hide much or less potential. Depending on this initial potential all the curves achieve higher or lower results.

The same statements can be made for the confidence intervals of Figure 8(b). These results are generated using Scenario S2. In contrast to the previous scenario, the higher number of available gateways causes a better efficiency of the Cell Crossover. The cell exchanging approach crosses during the evolution more and more reasonable gene combinations of the particular cell connections. Moreover, the small number of nodes belonging to one gateway allows more manifoldness between the individuals. This is due to the fact that small changes in the routing structure cause higher changes in the network performance than in the G2U71 city. However, the Cell and Subtree Crossover which exchange only connectivity components show generally better effectiveness than the 2-Point approach.

The comparison of the crossover types shows that the choice of a suitable crossover operator depending on the considered topology achieves better results than a random one.

4.4 Effectiveness of the Mutation Operator

The mutation is an important part of the natural evolution. In the organic world as well as in genetic algorithms, it accomplishes the gene diversity and helps the evolution to grow. In the following, we investigate the influence of the mutation operator considering Scenario S1.

Figure 9 shows the results. The dotted lines present the results without using mutation and the solid lines show the results where mutation is activated.

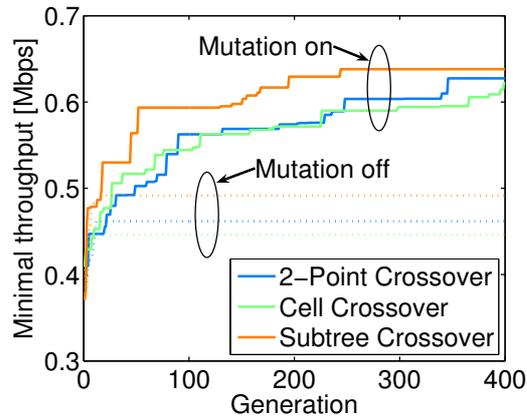


Figure 9: Mutation ON/OFF in combination with three crossover types tested on the G2U71 topology

It is easy to see how crucial the usage of the mutation operator is for the success of the evolution. The dotted lines get rapidly to a local maximum and stop the growth of the best individuals fitness. In contrast, the solid lines grow slowly most of the time. By using the mutation operator even at the 400th generation there is still potential for further evolution. The mutation operator creates new unexplored genes and gets the evolution out of the local maximum. In contrast, abstaining from the mutation operator quickly leads to very similar individuals containing the same gene combinations evaluated at the beginning of the evolution as the best ones.

Figure 10 shows an example of a network solution achieved with our genetic algorithm after 400 generations. To generate this solution, we used the G2U71 topology, an elite set size of 50, the mutation operator, and the Subtree Crossover with 7 exchanged subtrees per progeny. The numbers at the network links show the used frequency channel in MHz. The tree structure of the network is easy recognizable. The large number of users distributed between only two gateways requires long simulation time to achieve a near perfect solution. The complexity of the network and the

multiple possible paths over several hops toward a gateway offer numerous alternative solutions. This shows again how important it is to use an appropriate genetic algorithm configuration in order to test reasonable gene combinations and to achieve good solutions in reasonable time.

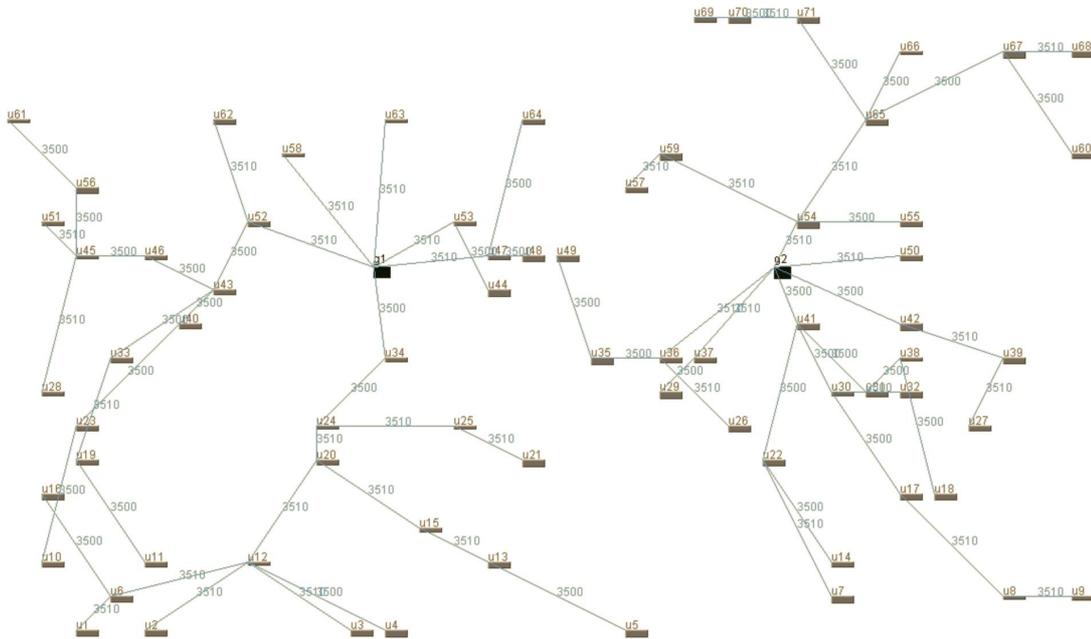


Figure 10: Solution of the G2U71 topology achieved with our genetic algorithm after 400 generations using the Subtree Crossover and mutation

5 Conclusion

In this paper we presented an approach for planning and optimization of WMNs using genetic algorithms. We introduced different genetic operators especially designed for the optimization of WMNs. We observed the effectiveness of the particular genetic operators. Thereby, it was shown that our WMN-specific Cell and Subtree Crossover achieve better solutions than the common 2-Point Crossover. Though, after observing different network topologies, we proved that the performance of the crossover types is topology dependent. The Subtree Crossover suits more the optimization of networks with a small number of gateways and large number of users. In contrast, the Cell Crossover achieves better solutions of networks disposing of multiple gateways. Furthermore, we have shown that a reasonable network optimization is only possible by using mutation. We tested the influence of this operator in combination with all crossover types and proved that in all cases it strongly forces the evolution. The results show that the best genetic algorithm configuration uses crossover and mutation in combination.

References

- [1] S. Sen and B. Raman, "Long distance wireless mesh network planning: problem formulation and solution," in *WWW '07: Proceedings of the 16th international conference on World Wide Web*, (New York, NY, USA), pp. 893–902, 2007.
- [2] B. He, B. Xie, and D. P. Agrawal, "Optimizing deployment of Internet gateway in Wireless Mesh Networks," *Computer Communications*, vol. 31, no. 7, pp. 1259–1275, 2008.

- [3] E. Amaldi, A. Capone, M. Cesana, I. Filippini, and F. Malucelli, "Optimization models and methods for planning Wireless Mesh Networks," *Computer Networks*, vol. 52, pp. 2159–2171, August 2008.
- [4] A. So and B. Liang, "Minimum Cost Configuration of Relay and Channel Infrastructure in Heterogeneous Wireless Mesh Networks," in *Networking*, (Atlanta, GA, USA), pp. 275–286, May 2007.
- [5] L. Badia, A. Botta, and L. Lenzini, "A genetic approach to joint routing and link scheduling for wireless mesh networks," *Elsevier Ad Hoc Networks Journal*, vol. Special issue on Bio-Inspired Computing, p. 11, April 2008.
- [6] T. Vanhatupa, M. Hännikäinen, and T. D. Hämäläinen, "Performance Model for IEEE 802.11s Wireless Mesh Network Deployment Design," *Journal of Parallel and Distributed Computing*, vol. 68, pp. 291–305, March 2008.