

# Wireless Mesh Network Planning and Optimization through Genetic Algorithms

Rastin Pries, Dirk Staehle, Marieta Stoykova, Barbara Staehle, Phuoc Tran-Gia  
University of Wuerzburg, Institute of Computer Science, Wuerzburg, Germany  
Email: {pries,dstaehle,stoykova,bstaehle,trangia}@informatik.uni-wuerzburg.de

## Abstract

*Wireless Mesh Networks (WMNs) are a promising technology for providing broadband wireless access to the end user. They offer a higher degree of flexibility compared to traditional networks but on the expense of a more complex structure. Thus, planning and optimization of WMNs is a challenge. In this paper, we address this challenge using genetic algorithms. Genetic algorithms are able to evaluate and optimize large-scale WMNs in relatively small computation time. The results prove the effectiveness of the genetic operators to optimize the routing and channel assignment in WMNs.*

## 1. Introduction

The wireless mesh networking paradigm is a very promising extension to common wireless technologies. It is a new concept which helps to satisfy the user demands for more coverage, bandwidth, and mobility. Different from traditional wireless networks, a data flow in a WMN traverses 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. Hence, optimization approaches like linear programming or simulated annealing soon reach their limits when considering large WMNs. In contrast, genetic algorithms are able to solve this optimization approach because of their simplicity and ability to optimize even large WMN scenarios.

Genetic algorithms (GAs) 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 quality and to decide whether to keep it in the new population. However, the selection without any other operation will lead to local optima. Therefore, two operators, crossover and mutation, are used to create new individuals. These new individuals are called progenies.

In this paper we apply genetic algorithms on routing and channel allocation in order to optimize the throughput of WMNs. Our goal is to achieve a max min fair share resource allocation. Therefore, we encoded the WMN, created new

crossover variants, and investigate their impact as well as the impact of the mutation operator on the evolution. This way, we explore the suitability 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 impact of the crossover and mutation operator on the resulting solution is shown. Finally, Section 5 concludes this paper.

## 2. Related work

Wireless mesh networks have attracted 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. 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 works [2]–[4] deal with creating a wireless mesh network model, planning its parameters, and evaluating the solutions via linear programming. He et al. [2] propose 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 linear programming. 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.

Ghosh et al. [5] are the first to use genetic algorithms for wireless multi-hop optimization. They try to minimize the costs and to maximize the link availability of a UMTS network with optical wireless links to the radio network controllers. Besides Gosh et al., Badia et al. [6] use genetic algorithms for a joint routing and link scheduling for WMNs. The packet delivery ratio is optimized in dependency of the frame length. They examine that genetic algorithms solve the studied problems reasonably well, and also scale, whereas exact optimization techniques are unable to find solutions for larger topologies. The performance of the genetic algorithm is shown for a single-rate, single-channel, single-radio WMN.

Vanhatupa et al. [7], [8] apply a genetic algorithm for the WMN channel assignment. Capacity, AP fairness, and coverage metrics are used with equal significance to optimize the network. The routing is fixed, using either shortest path routing or expected transmission times. They show an enormous capacity increase with the channel assignment optimization. Compared to manual tuning, their algorithm is able to create a network plan with 133 % capacity, 98 % coverage, and 93 % costs and the algorithm needs 15 minutes for the optimization whereas the manual network planning takes hours.

In contrast to the papers from Badia [6] and Vanhatupa [8], we are evaluating the performance of a multi-channel, multi-radio, multi-rate WMN using both channel and route assignment. Our genetic algorithm optimizes the throughput while still maintaining a max-min fair throughput allocation between the nodes.

### 3. Wireless Mesh Network Planning and Optimization via Genetic Algorithms

In this section we show the parameters which we have to consider and to evaluate in order to optimize a WMN. Optimization in our case means that the routes and channels are optimized in order to provide a max-min fair share resource allocation. A solution is max-min fair if no rate can be increased without decreasing another rate to a smaller value [9].

Fig. 1 clarifies the complete procedure of the genetic algorithm for the planning and optimization of WMNs. First, a random population is created with a predefined number of individuals. The fitness of each individual is evaluated using the fitness function and the individuals are ordered according to the fitness value. The best individuals, the elite set, are kept for the new population. Afterwards, the crossover and mutation operator are used to create the remaining number of individuals for the new population. The procedure is recursively repeated until a sufficient solution is achieved.

In the next subsection we explain the steps of our WMN optimization approach in more detail.

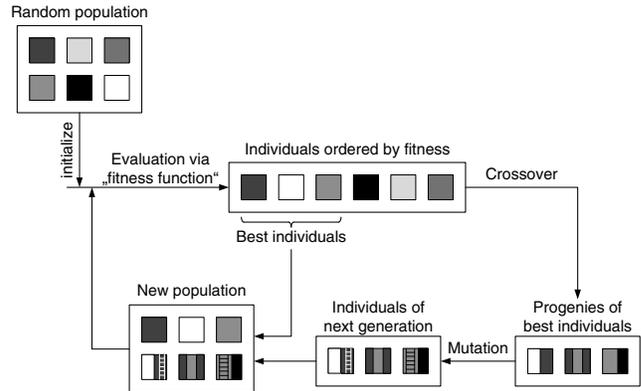


Figure 1. Functionality of genetic algorithms.

#### 3.1. Problem Formulation

We define a mesh network as a set of  $N$  nodes  $n_1, \dots, n_N$  and a set of links  $\mathcal{L}$  connecting the nodes. A subset  $G \subseteq \mathcal{N}$  contains the gateway nodes which are connected to the Internet. Each node  $n_i \in \mathcal{N} \setminus G$  has a fixed path and gateway to the Internet. The path is denoted as  $\mathcal{P}_i$  and consists of a set of links,  $\mathcal{P}_i \subseteq \mathcal{L}$ .

A link  $(i, j)$  between nodes  $i$  and  $j$  exists, if a communication between these nodes is possible. Let  $r_{i,j}$  be the data rate of the link  $(i, j)$ . The goal is now to optimize the paths from each node  $n_i \in \mathcal{N} \setminus G$  to the gateway so that the throughput in the WMN is maximized. Therefore, we first need to encode the WMN.

#### 3.2. Encoding

Our WMN representation includes only one link per user and is easy to handle and evaluate. This link denotes the next hop which the traffic of the considered node has to take in order to reach the gateway. Thus, we always imply only one possible path towards a gateway for the packets of each 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. Fig. 2 illustrates an example for the routing and channel encoding.

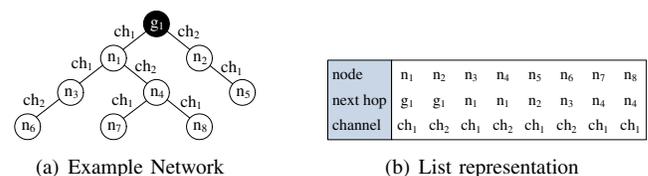


Figure 2. Example network and its list representation.

### 3.3. Fitness Function

After having created the initial population using the encoding scheme, the fitness of the population has to be determined. Based on the fitness value, the GA decides which individuals should be kept in the new population. Hence, it rates the performance of the individuals and allows only the best to be replicated.

The fitness of the WMN is estimated by using the allocated throughputs obtained from the max-min fair share algorithm. Several different fitness functions  $f(\mathcal{N})$  can be used to evaluate the individuals. Due to the fact that we try to maximize the minimal throughput, we set the fitness function to  $f(\mathcal{N}) = \min(\mathcal{R}_{\mathcal{N}})$  meaning that the fitness of an individual is determined by the minimal throughput of all end-to-end flows in the routing scheme  $\mathcal{R}_{\mathcal{N}}$ .

In order to determine the throughput of an end-to-end traffic flow in the WMN, we first have to define the collision domain of each link  $(i, j)$ . The collision domain  $\mathcal{D}_{i,j}$  of a link  $(i, j)$  corresponds to the set of all links  $(s, t)$  which can not be used in parallel to link  $(i, j)$  because of a too strong interference [10]. The nominal load of such a collision domain is the number of transmissions taking place in it. A transmission  $t_{k,i,j}$  corresponds to the hop from node  $i$  to node  $j$  taken by the flow towards node  $k$ , i.e.  $(i, j) \in \mathcal{P}_k$ . The number of transmission  $\lambda_{i,j}$  of link  $(i, j)$  corresponds to the number of end-to-end flows crossing it:

$$\lambda_{i,j} = \left| \{k | (i, j) \in \mathcal{P}_k\} \right|. \quad (1)$$

Fig. 3 shows an example for determining the link loads. Each node on the way to the gateway produces traffic resulting in a traffic load of 5 on the link between  $n_2$  and  $n_1$ .

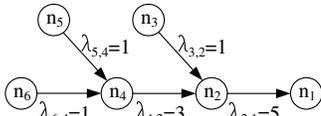


Figure 3. Link load calculation depending on the carried number of flows.

Correspondingly, the number of transmissions in collision domain  $\mathcal{D}_{i,j}$  is

$$m_{i,j} = \sum_{(s,t) \in \mathcal{D}_{i,j}} \lambda_{s,t}. \quad (2)$$

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 rates  $r_{i,j}$  and the loads  $\lambda_{i,j}$  into account. The throughput  $t_{i,j}$  of link  $\ell_{i,j}$  is then defined as:

$$t_{i,j} = \frac{1}{\sum_{(s,t) \in \mathcal{D}_{i,j}} \frac{\lambda_{s,t}}{r_{s,t}}}. \quad (3)$$

Now, we follow the principle of max-min fairness and fix the resources for the link with the smallest throughput. We call it the bottleneck of the network and denote it with  $\ell_{u,v}$ . The time resources occupied by  $\ell_{u,v}$  for supplying its  $\lambda_{u,v}$  flows can now be calculated as

$$\rho_{u,v}(\ell_{i,j}) = \lambda_{u,v} \cdot \frac{t_{u,v}}{r_{i,j}}. \quad (4)$$

They differ depending on the link for which they are calculated. Such links can be by  $\ell_{u,v}$  bottlenecked connections or parent-links on the path towards the gateway.

Having computed the occupied resources and having fixed 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 link throughputs. Moreover, we need to update  $\lambda_{i,j}$  by subtracting the flows supplied through the bottleneck. When all network resources are refreshed, we fix the next link with the smallest throughput. This way, we calculate the throughput of each end-to-end flow. The minimal throughput stands then for the fitness of the individual. More information about this algorithm can be found in [11].

### 3.4. Selection Principle

After the evaluation of a population, we select a set of solutions, which have the highest fitness of all and keep them in the new generation. This set is called the elite set. In addition to the elite set, the rest of the population is created by crossing and mutating the genes. Thereby, the number of progenies per individual is proportional to its fitness value. 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,  $m$  be the number of best ancestors to be kept in the next generation, and  $s(x)$  the selection probability for the individual  $x$ . The number of progenies of  $x$  is then given by

$$g(x) = (n - m) \cdot s(x). \quad (5)$$

The selection probability of the individual  $x$  is described by the relation between the fitness of this solution and the sum of the fitness values of all individuals from its population:

$$s(x) = \frac{f(x)}{\sum_{j=1}^n f(j)}. \quad (6)$$

This fitness dependent selection results in higher reproduction of genes from solutions with better performance.

### 3.5. Crossover Types

The crossover operator as well as the mutation operator are now applied to the selected number of individuals. For the cross of genes, we use the standard 2-Point Crossover [12] and two other variants which we especially created for the planning of WMNs, the Cell and the Subtree Crossover.

**3.5.1. 2-Point Crossover.** The 2-Point Crossover is the simplest realization of the genetic cross. It is an exchange of gene subsets which are randomly chosen sublists of the individuals list representation, the genotype. The start and end intersection points denoting the range of the sublist are chosen each time when the operator is applied.

Fig. 4 shows an example of the 2-Point Crossover between two network solutions. The intersection points are at the second and fifth position in the individuals code and enclose the sublist of genes for the nodes  $n_2$  to  $n_5$ . These denote the area which will be exchanged during the crossover. The resulting progenies of the individuals show one characteristic of this reproduction approach. It created 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. In Fig. 4 we can observe how the cross of two genotypes containing subgraphs with no gateway connection results in a reasonable solution (progeny of individual 1) or in an unconnected solution (progeny of individual 2).

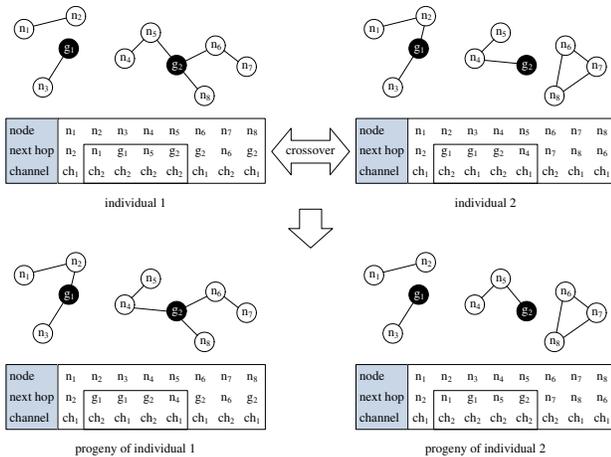


Figure 4. 2-Point Crossover between two individuals.

Since the 2-Point Crossover may lead to unconnected solutions, we have to be careful when evaluating the fitness of the resulting solutions. Thus, we adapt the fitness function to

$$\tilde{f}(\mathcal{N}) = f(\mathcal{N}) - \text{conless}(\mathcal{R}_{\mathcal{N}}). \quad (7)$$

which includes now the  $\text{conless}(\mathcal{R}_{\mathcal{N}})$  term denoting the number of nodes with no connection to any gateway. Hence, the 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.

**3.5.2. Cell Crossover.** The Cell Crossover represents a connectivity dependent exchange of genes. The crossover operator randomly chooses a gateway and exchanges the entire cell meaning that the routing information as well

as the channel allocation is exchanged. Fig. 5 shows an example for the crossover of two solutions. Black nodes denote the network gateways and the gray areas mark the chosen cell which is exchanged. The nodes which links have changed are marked gray in the resulting progenies. It is obvious that the number of nodes belonging to the cell differ between the individuals. Therefore, we have to attach unconnected nodes after the cell crossover (see progeny of individual 1).

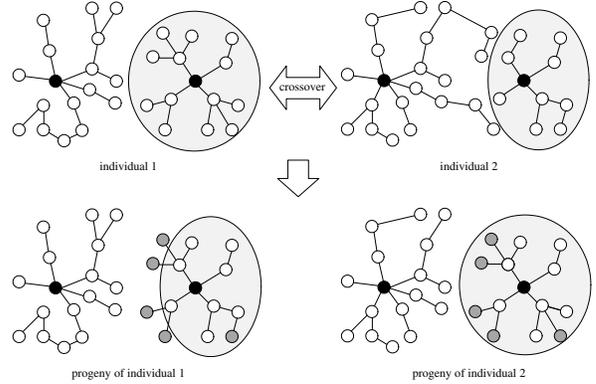


Figure 5. Cell Crossover between two individuals.

**3.5.3. Subtree Crossover.** The Subtree Crossover exchanges connectivity components with respect to the network structure. The crossover operator chooses a random position of the individuals code and exchanges the entire subtree with the node at the chosen position as root. Thereby, the channel allocation is exchanged together with the routing information.

Fig. 6 shows an example for the Subtree Crossover with two positions whose corresponding subtrees are meant to be exchanged. The gray nodes denote the subtrees which are going to be crossed. In the example, the crossover results on the one hand in a solution with good routing performance (progeny of individual 1) and on the other hand in a progeny with medium routing performance and still existent potential for further reproduction (progeny of individual 2).

### 3.6. Mutation

The mutation, i.e. the arbitrary modification of genes, is a very important part of the evolution process. The number of mutations is chosen based on the scenario size. For our WMN optimization, we use two mutation operators; the mutation of the routing and the mutation of the channel allocation. Both mutation operators are applied independently from each other.

For the routing scheme, the mutation operator substitutes some randomly chosen positions of the routing code with new information taken from a set of potential neighbors which would not cause the creation of cycles and would

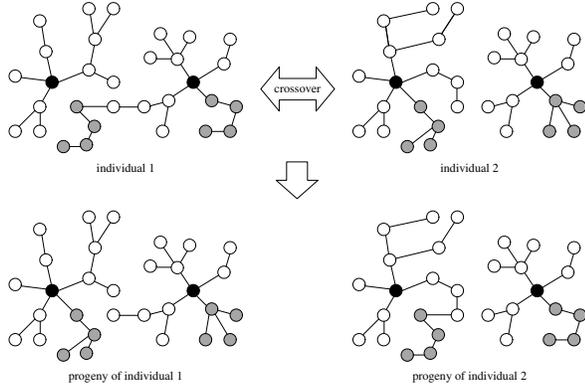


Figure 6. Subtree Crossover between two individuals.

not harm the tree structure of the solution. An example for the mutation of the routing scheme from three nodes is shown in Fig. 7. Here, the link towards the gateway of the three gray nodes is mutated. For the channel allocation, the mutation operator randomly chooses a channel from a list of possible channels and substitutes randomly chosen links from the WMN.

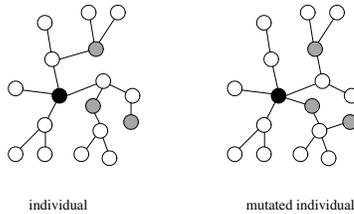


Figure 7. Mutation of the routing parameters.

## 4. Performance Evaluation

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 the two scenarios introduced in Table 1. Although we evaluated a large number of different scenarios, we highlight only the two most different ones here. 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 fixed level suitable for the investigation of the genetic operators. The fitness function  $f(\mathcal{N}) = \min(\mathcal{R}_{\mathcal{N}})$  calculates the minimal throughput of all end-to-end flows from the routing scheme  $\mathcal{R}_{\mathcal{N}}$ .

Table 1. Simulation Scenarios.

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

The parameters of the transmission technology 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
Transmission technology	WiMAX
carrier frequency	3500 MHz
channel bandwidth	20 MHz
maximum throughput	67.2 Mbps
available channels	3500 MHz; 3510 MHz
antenna power	25 dBm
pathloss model	WiMAX urban macrocell model

### 4.2. Population Evolution

Examining the evolution of the population 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

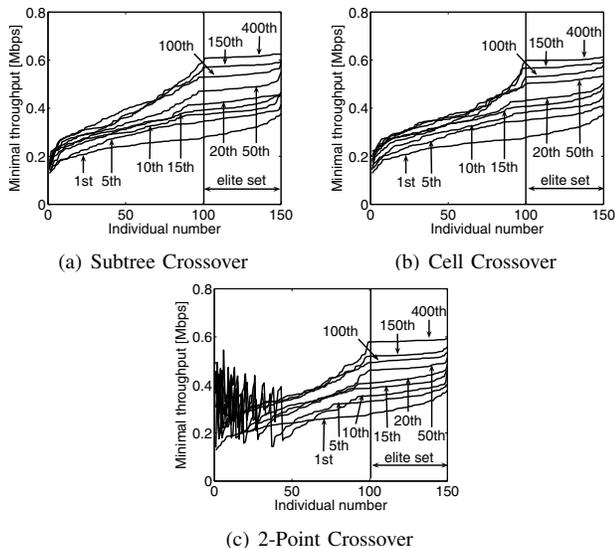


Figure 8. Generations progress by Subtree, Cell, and 2-Point Crossover tested on the G2U71 topology.

can affect the evolution by more or less stimulating it but none of them should disturb it.

Fig. 8 shows the minimal throughput growth during the evolution for different crossover types. For generating the results of Fig. 8, we used Scenario S1 from Table 1.

The x-axes show the individuals sorted by fitness while the y-axes present the minimal flow throughput of the solutions. The different crossovers show the evolution of the generations 1 to 20 in steps of 5 generations, and 50 to 400 in steps of 50 generations.

We have to mention that the fitness values are not comparable, due to the penalty costs used for the 2-Point Crossover. Hence, we consider only the minimal throughputs which only present the positive costs. This is also the reason for the strongly varying curves on the left side of Fig. 8(c). The individuals have a large minimal throughput but there are a lot of unconnected nodes which results in a lot of penalty costs and thus in lower fitness.

In all subfigures we can observe how the minimal throughput of the elite set grows with every generation. This is due to the selection principle which keeps the ancestors of the prior generation in the next one. The elite selection approach creates new populations with an elite set that is definitely better than the previous one.

The higher the generation number, the smaller the fitness growth. 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 improve 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. However, after about 400

generations, all crossover types show only a small fitness growth with every new generation.

### 4.3. Effectiveness of the Crossover Operator

In this section we compare the performance of the three crossover operators depending on the number of users and gateways in the network. Furthermore, we want to find out if there is an interaction between the efficiency of the crossover types depending on the topology.

The results for both scenarios from Table 1 are presented in Fig. 9. Fig. 9(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 applying a 95% confidence interval.

This scenario includes a high number of user nodes which are distributed in the coverage cells of only two gateways. This results in deep tree structures with long ways over multiple hops towards 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. 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 outperforms the intelligent Cell Crossover approach. This results from the small number of gateways which causes the cross of only one cell per new progeny and quickly leads to similar individuals.

The results from Scenario S2 are shown in Fig. 9(b). In contrast to the previous scenario, the higher number of available gateways cause a better efficiency of the Cell Crossover. Moreover, the small number of nodes belonging to one gateway allows a larger variety of 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 topology. However, the Cell and Subtree Crossover which exchange only connectivity components show a better performance than the 2-Point Crossover.

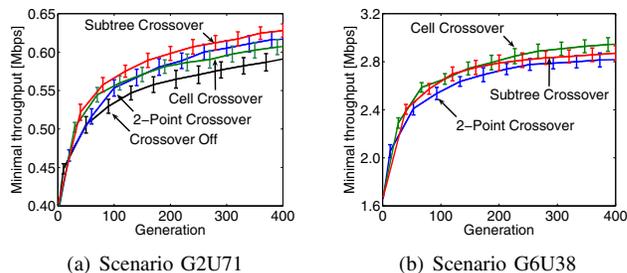


Figure 9. Effectiveness of the crossover operator.

The comparison of the crossover types shows that the crossover operator should be selected based on the considered topology to achieve the best solutions. In the next section we take a look at the influence of the mutation operator on the evolution of the population.

#### 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.

From Fig. 10 it is easy to see how crucial the usage of the mutation operator is for the success of the evolution. Without using mutation, the solution reaches a local optimum after about 20 generations. In contrast, when activating the mutation operator, the fitness of the solution grows even after 400 generations and there is still potential for further evolution. The reason is that the mutation only slightly changes the routing scheme and channel allocation but thereby creates new unexplored genes and fosters the evolution. In contrast, abandoning 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.

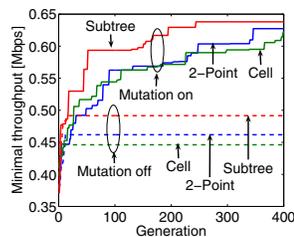


Figure 10. Mutation ON/OFF in combination with three crossover types tested on the G2U71 topology.

## 5. Conclusion

In this paper we present an approach for planning and optimizing WMNs using genetic algorithms. Different genetic operators are introduced which we especially designed for WMNs. The performance of these operators is evaluated in different scenarios and the two most different scenarios are shown in this paper. The results illustrate that our WMN-specific Cell and Subtree Crossover lead to better solutions compared to the well-known 2-Point Crossover. However, they have to be applied according to the network topology. The Subtree Crossover shows the best performance in scenarios with a small number of gateways whereas the Cell Crossover leads to the best solutions in scenarios with a large number of gateways. Finally, 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, 2007, pp. 893–902.
- [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, no. 11, 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, May 2007, pp. 275–286.
- [5] S. Ghosh, P. Ghosh, K. Basu, and S. K. Das, "GaMa: An Evolutionary Algorithmic Approach for the Design of Mesh-Based Radio Access Networks," in *LCN '05: Proceedings of the The IEEE Conference on Local Computer Networks 30th Anniversary*, Washington, DC, USA, November 2005, pp. 374–381.
- [6] L. Badia, A. Botta, and L. Lenzi, "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.
- [7] T. Vanhatupa, M. Hännikäinen, and T. D. Hämäläinen, "Genetic Algorithm to Optimize Node Placement and Configuration for WLAN Planning," in *4th International Symposium on Wireless Communication Systems, 2007. ISWCS 2007*, Trondheim, Norway, October 2007, pp. 612–616.
- [8] —, "Performance Model for IEEE 802.11s Wireless Mesh Network Deployment Design," *Journal of Parallel and Distributed Computing*, vol. 68, no. 3, pp. 291–305, March 2008.
- [9] D. P. Bertsekas and R. G. Gallager, *Data networks*. Prentice-Hall, Inc., 1987.
- [10] J. Jun and M. L. Sichitiu, "The Nominal Capacity of Wireless Mesh Networks," *IEEE Communications Magazine*, vol. 10, no. 5, pp. 8–14, October 2003.
- [11] D. Staehle, B. Staehle, and R. Pries, "Max-Min Fair Throughput in Multi-Gateway Multi-Rate Mesh Networks," University of Würzburg, Tech. Rep. 454, January 2009.
- [12] J. H. Holland, *Adaptation in natural and artificial systems*. Cambridge, MA, USA: University of Michigan Press, 1975.