A Genetic Approach for Wireless Mesh Network Planning and Optimization

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

ABSTRACT

Wireless Mesh Networks (WMNs) are gaining an increasingly important role in next generation last mile access. 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 for large-scale WMNs in relatively small computation time. The results prove the effectiveness of the genetic operators and show the advantages of a genetic optimization. However, these operators have to be configured carefully to avoid local optima. We will show the influence of the selection principles as well as evaluation functions on the optimization.

Categories and Subject Descriptors

C.2.1 [**COMPUTER-COMMUNICATION NETWORKS**]: Network Architecture and Design—*Wireless communication*
; G.1.6 [**NUMERICAL ANALYSIS**]: General—*Optimization*

General Terms

Algorithms, Performance

Keywords

Wireless Mesh Networks, Planning, Optimization, Routing, Genetic Algorithms

1. INTRODUCTION

Wireless Mesh Networks (WMNs) are a promising technology to provide broadband wireless connectivity for the end user. Compared to traditional wireless networks, WMN planning and optimization is more challenging because several wireless hops are needed to connect a node to the Internet. A suitable solution for the planning and optimization of such WMNs are Genetic Algorithms (GAs) due to their ability to solve complex problems in relatively small computation time.

Genetic algorithms are based on the idea of natural evolution and are used to solve optimization problems by simulating the biological cross of genes. They refer to a theory investigated by John Holland and his students at the University of Michigan in the 1970s [6]. Multiple variations of genetic algorithms exist nowadays but most of them are based on the model introduced by Holland.

A randomly created population of individuals represents the set of candidate solutions for a specific problem. The genetic algorithm applies a fitness function on each individual to evaluate its performance and to decide whether to keep it in the following generation. Using only the selection without any other operation will result in a local optima. Therefore, two operators, crossover and mutation, are used to create new individuals, called progenies.

In this paper we evaluate the influence of different genetic operators for routing and channel allocation in WMNs. Optimizing these operators helps us to minimize the time for the evaluation. In contrast to applying exact optimization techniques on this problem, genetic algorithms scale well so that we are able to optimize even large WMNs. Thereby, we want to increase the throughput of the complete WMN while sharing the resources fairly among the nodes. This is achieved by applying a max-min fair share algorithm presented in [11] and by tuning the genetic parameters. A solution is max-min fair if no rate can be increased without decreasing another rate to a smaller value [2]. In contrast to our previous work [10] where we evaluated the influence of the genetic operators crossover and mutation on the solution, we are evaluating the parameters fitness function, number of generations, and elite set size.

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 our algorithm for WMN throughput maximization. In Section 4 the influence of the fitness function, the number of generations, and the elite set size are evaluated. Finally, Section 5 concludes this paper.

2. RELATED WORK

Genetic algorithms have been used for radio network planning for years [1, 3–5, 7, 9, 12]. Calégari et al. [3] applied a genetic algorithm for UMTS base station placement in order to obtain a maximum coverage. They claimed that the performance of the GA strongly depends on the fitness function. Another paper on UMTS optimization with genetic algorithms was published by Ghosh et al. [5] in 2005. Genetic algorithms are used to minimize the costs and to maximize the link availability of a UMTS network with optical wireless links to the radio network controllers.

Besides cellular network planning, genetic algorithms are used for the optimization of Mobile Ad hoc NETworks (MANETs) and Wireless Sensor Networks (WSNs). More problems occur in these networks due to a huge parameter space for which genetic algorithms may outperform manual tuning. Montana and Redi [9] studied how optimal parameters for a reactive routing protocol in an ad hoc network with multiple antennas can be found using a GA. The fitness of a given parameter set is evaluated by a weighted sum of packet drops and transmission delays obtained from simulation. The results demonstrate that the automated parameter tuning using a genetic algorithm leads to better results than manually chosen parameter combinations. In WSNs it is under some conditions beneficial to establish a cluster hierarchy, where selected nodes act as cluster heads and aggregate and transmit their children node's data to the base station. Hussain et al. [7] propose a GA for demonstrating that their solution is suitable for maximizing the network lifetime. Ferentinos et al. [4] developed a genetic algorithm to determine which sensor nodes should be active, to choose cluster-heads and radio ranges.

Badia et al. [1] use genetic algorithms for a joint routing and link scheduling for WMNs. The packet delivery ratio is optimized in dependency of the frame length. It was shown 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. [12] apply 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. Using a channel assignment optimization, a mean capacity increase of 131 % was shown.

In contrast to the papers from Badia [1] and Vanhatupa [12], 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 fairness between the nodes.

3. WMN PLANNING USING GENETIC ALGORITHMS

The objective of this paper is to support the WMN planning process by optimizing the performance of a WMN. With the help of genetic algorithms, near optimal solutions can be achieved in relatively small computation time. In this section, we show the parameters which we have to consider and to evaluate in order to achieve a near optimal WMN solution, meaning that the throughput in the WMN is max-min fair.

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, is 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 repeated until a sufficient solution is achieved. In the next subsection we explain the steps of our WMN optimization approach in more detail.

3.1 Problem Formulation

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

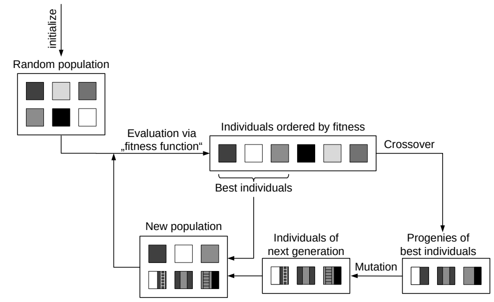


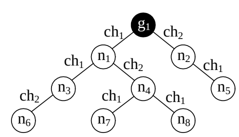
Figure 1: Functionality of genetic algorithms.

set of paths form a tree with a gateway as root and the complete network forms a forest.

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 $n_i \in \mathcal{N} \setminus \mathcal{G}$ to the gateway so that the throughput in the WMN is max-min fair.

3.2 Encoding

In order to optimize the throughput using genetic algorithms, the WMN trees have to be encoded. Our simple 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.



(a) Example network

node	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8
next hop	g_1	g_1	n_1	n_1	n_2	n_3	n_4	n_4
channel	ch_1	ch_2	ch_1	ch_2	ch_1	ch_2	ch_1	ch_1

(b) List representation

Figure 2: Example network and its list representation.

3.3 Fitness Function

The evaluation part of the optimization 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.

The fitness of the WMN is estimated using the allocated throughputs of each flow. The fitness function $f(\mathcal{N})$ of the evaluation represents the user satisfaction and the fairness of the resource allocation. In Section 4 we evaluate the influence of different fitness functions on the evolution of the population.

In order to determine the throughput of each link in the WMN, we 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 the interference from a transmission on link (s, t) alone is strong enough to disturb a parallel transmission on link (i, j) [8]. The nominal load of such a collision domain is the number of transmissions taking place in the collision domain. 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 transmissions $\lambda_{i,j}$ on link (i, j)

corresponds to the number of end-to-end flows crossing it:

$$\lambda_{i,j} = \Big|\{k|(i,j) \in \mathcal{P}_k\}\Big|. \tag{1}$$

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.

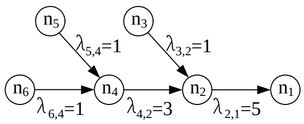


Figure 3: Link load calculation depending on the carried number of flows.

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

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

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:

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

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 on $l(i,j)$ by $\ell_{u,v}$ for supplying its $\lambda_{u,v}$ flows can now be calculated as

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

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 flow which is needed to evaluate the fitness of the WMN. 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 Section 4 we vary the size of the elite set in order to see the influence on the solution. 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. 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

$$g(x) = \|(n-m) \cdot s(x)\|. \tag{5}$$

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:

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

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

3.5 Crossover Types

The list representation helps us to apply the genetic operators. For the cross of genes, we use the 2-Point Crossover which is generally applied for genetic optimization [6] and another variant which we especially created for the planning of WMNs, the Subtree Crossover.

3.5.1 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 randomly 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.

Fig. 4 shows an example of a 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 creates 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, which is the progeny of individual 1 or in an unconnected solution shown in progeny of individual 2.

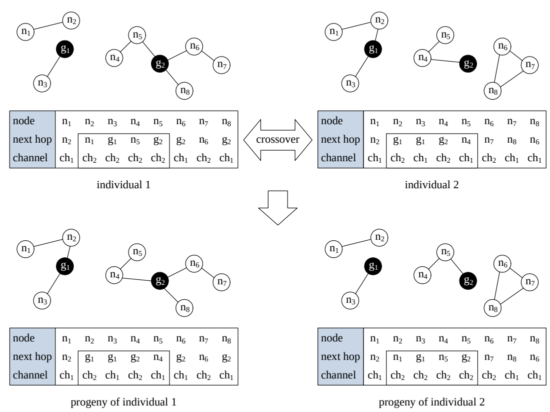


Figure 4: 2-Point Crossover between two individuals.

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

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

which includes now the $conless\,(\mathcal{R}_{\mathcal{N}})$ 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 $conless\,(\mathcal{R}_{\mathcal{N}})$ stands for the penalty costs.

3.5.2 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. 5 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 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).

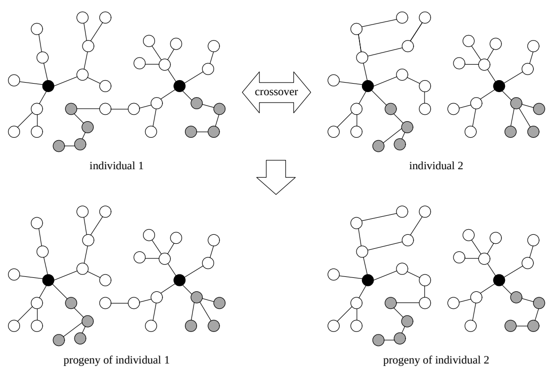


Figure 5: Subtree Crossover between two network solutions.

3.6 Mutation

The mutation, i.e. the arbitrary modification of genes, is a very important part of the evolution process. The mutation operator substitutes some randomly chosen positions of the routing and channel allocation code with new information taken from a set of legal entries. These entries are selected from a list of potential neighbors which would not cause the creation of cycles and would not harm the tree structure of the solution. 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. The number of mutations is chosen in dependence on the scenario size.

4. PERFORMANCE EVALUATION

The performance of a genetic algorithm for WMN planning reflects the quality of the genetic operators, the effectiveness of the fitness function, and the interaction of some other parameters affecting the evolution. In this section we evaluate the influence of the fitness function, the number of generations, as well as the elite set size.

4.1 Simulation Settings

To test the performance of the GA and its parameters, we use two scenarios introduced in Table 1. Both scenarios consist of 2 gateways and 71 users distributed over an area of 2 km to 1.2 km. Thereby, the minimal distance between the users is 60 m and between the two gateways 700 m.

Besides the parameters for the genetic algorithm, the table also lists the fixed parameters which affect only the characteristics of the network connections. Based on these parameters, the pathloss and the resulting Signal to Interference Ratio (SIR) are calculated for all links. An appropriate data rate is chosen according to the SIR. Thus, these parameters do not affect the effectiveness of the genetic algorithm itself. Therefore, we do not consider their impact on the resulting solutions.

Table 1: Parameters for the Genetic Algorithm.

GA Parameters	Scenario S1	Scenario S2
population size	150	
elite set size	50	5,...,125
number of generations	400	
crossover type	Subtree CO	2-Point CO Subtree CO
number of crossed subtrees	rand(0,7)	
number of mutations	rand(0,20)	
fitness function	$f_1(\mathcal{N}), ...f_8(\mathcal{N})$	$f_1(\mathcal{N})$
WMN Parameters		
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 Influence of the Fitness Function

The fitness function of our genetic algorithm is responsible for the max-min fairness of the resulting solution. Some fitness functions might lead to complete unfair resource distributions in the WMN. Therefore, we evaluate different fitness functions in this section. We use several combinations of the functions $min(\mathcal{R}_\mathcal{N})$, $median(\mathcal{R}_\mathcal{N})$, $mean(\mathcal{R}_\mathcal{N})$, and $var(\mathcal{R}_\mathcal{N})$, which we apply on all routing links of a network solution $\mathcal{N}$. The function $min\,(\mathcal{R}_\mathcal{N})$ calculates for example the minimum throughput of all links used in routing scheme $\mathcal{R}_\mathcal{N}$. We define the following eight different fitness functions:

$$
\begin{aligned}
f_1(\mathcal{N}) &= min(\mathcal{R}_\mathcal{N}) = minimum\ throughput(\mathcal{R}_\mathcal{N}) \\
f_2(\mathcal{N}) &= median(\mathcal{R}_\mathcal{N}) = median\ throughput(\mathcal{R}_\mathcal{N}) \\
f_3(\mathcal{N}) &= mean(\mathcal{R}_\mathcal{N}) = mean\ throughput(\mathcal{R}_\mathcal{N}) \\
f_4(\mathcal{N}) &= min(\mathcal{R}_\mathcal{N}) + \frac{median(\mathcal{R}_\mathcal{N})}{p} \\
f_5(\mathcal{N}) &= mean(\mathcal{R}_\mathcal{N}) - var(\mathcal{R}_\mathcal{N}) \\
f_6(\mathcal{N}) &= min(\mathcal{R}_\mathcal{N}) + \frac{median(\mathcal{R}_\mathcal{N})}{p} + \frac{mean(\mathcal{R}_\mathcal{N})}{|\mathcal{L}|} \\
f_7(\mathcal{N}) &= \sum_{i=0}^{|\tilde{T}|-1} (\left|\tilde{T}\right| - i) \cdot \tilde{T}(i) \\
f_8(\mathcal{N}) &= \sum_{i=0}^{|\tilde{T}|-1} k^{|\tilde{T}|-i} \cdot \tilde{T}(i)
\end{aligned}
$$

The last two functions weight the link throughputs with a factor depending on the corresponding throughput value. Therewith, we aim to achieve a kind of max-min fairness not only with the throughput allocation made by the evaluating algorithm but also with the fitness value from a reasonable fitness function. For this purpose, an ascendingly sorted list $\tilde{T}$ of the throughputs of all routing links in the solution $\mathcal{N}$ is used. Each throughput value from $\tilde{T}$ is weighted with a factor depending on its place in the list, giving more weight to lower positions. This should result in a fitness value with which mainly the smaller link throughputs are optimized at the expense of the higher ones. The parameter k of function $f_8(\mathcal{N})$ is a constant which we set to 1.5 and p is set to 8 for the experiments.

Fig. 6 shows a comparison of the throughput allocation of the best individual after 400 generations for different fitness functions. For the sake of readability, the curves of the eight functions are shown in two separate figures. The y-axes lists the throughput in Mbps of the flows, whose IDs are represented by normalized values on the x-axes.

The throughput values are ascendingly sorted for easier readability. A curve completely parallel to the x-axis would mean a perfect fairness between all flows and a curve whose minimum throughput is above $f_1(\mathcal{N})$ would mean that the solution is max-min fair. This allows to see that the unfairest resource distributions are achieved with the fitness functions $f_2(\mathcal{N})$ and $f_3(\mathcal{N})$.

Optimizing only the median with $f_2(\mathcal{N})$, we do not pay attention to the rest of the throughput allocation. This is why the left part of the $f_2(\mathcal{N})$ curve stays very low. The throughputs at the highest ID positions can even accidentally be much higher than the others. This is due to the fitness function which does not control their behavior. The $f_3(\mathcal{N})$ function optimizing only the mean throughput also results in a very unfair solution, as the number of hops towards the gateway are minimized in order to get some nodes with very high throughput which boost the mean value.

All other fitness functions result in a max-min fair resource distribution with a maximized minimum throughput. In the resulting solutions of $f_1(\mathcal{N})$, $f_6(\mathcal{N})$, and $f_8(\mathcal{N})$, some flows have a very high throughput but not on the costs of other flows. The fairest solutions is achieved with fitness function $f_7(\mathcal{N})$ where all flows have a similar throughput about 0.7 Mbps.

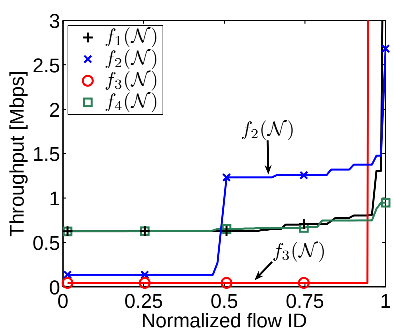
4.3 Population Evolution

The evolution speed is an important factor 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.

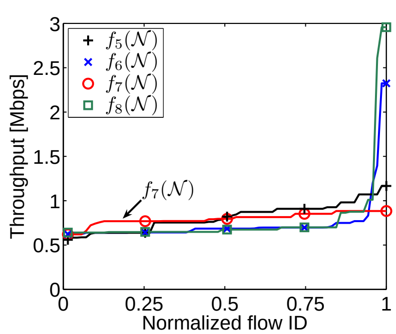
Fig. 7 shows the minimal throughput growth during the evolution. For generating the results of Fig. 7, we used Scenario S2 from Table 1 and set the elite set size to 50. The x-axes show the individuals sorted by fitness while the y-axes present the minimal flow throughput of the solutions. The figures illustrate the evolution of the generations 1 to 400. We have to mention that the fitness values in this investigation are not comparable, due to the penalty costs used for the 2-Point Crossover. Hence, we consider only the minimal throughputs which present the positive costs only. This is also the reason for the strongly varying curves on the left side of Fig. 7(b). 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.

From the figure we can observe how the minimal throughput of the elite set, grows with every next generation. This is due to the selection principle which keeps the best 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

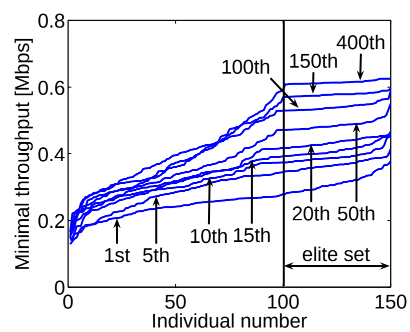


(a) $f_1(\mathcal{N})$ to $f_4(\mathcal{N})$

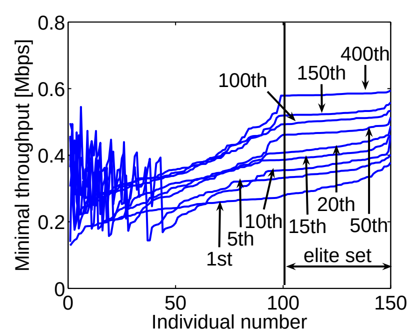


(b) $f_5(\mathcal{N})$ to $f_8(\mathcal{N})$

Figure 6: Throughput allocation of the best individual.



(a) Subtree Crossover



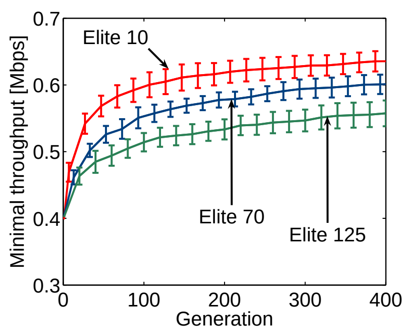
(b) 2-Point Crossover

Figure 7: Generations progress.

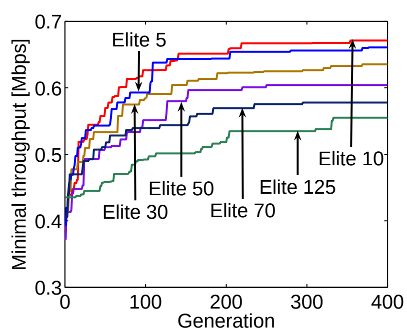
is the need of small and intelligent changes to improve the solutions fitness, which is hard to achieve accidentally. Though, the speed of the evolution depends also on the topology structure in combination with a suitable crossover principle. However, after about 400 generations, both crossover types show only a small fitness growth with every new generation.

4.4 Influence of the Elite Set Size

In this subsection, we examine the impact of the elite set size on the progress of the evolution using again Scenario S2. Fig. 8(a) shows the results averaged over 15 different initial populations. It presents the fitness progress of the best individual for three different elite set sizes using the Subtree Crossover. As we can observe, the best results are achieved with the smallest elite set size of 10 individuals. The bigger the elite set, the larger the number of individuals containing bad genes which are kept in the next generation. Furthermore, a size of 125 means that we create only 25 progenies (due to the population size of 150) and the probability of building new unexplored gene combinations decreases. This leads to a slowdown of the evolution and achieves solutions which definitely have lower fitness than the ones from the 10-curve. This means that scenarios with larger elite set sizes could also get to the best solution but they would need a larger number of generations. In Fig. 8(b) we consider the influence of six different elite set sizes with a single seed. We can see the same behavior as in Fig. 8(a). Smaller elite sets cause faster evolution and lead to better solutions. However, a too small elite set size is also bad as the figure shows for an elite set size of 5.



(a) 15 seeds



(b) One seed

Figure 8: Comparison of different elite set sizes.

5. CONCLUSION

In this paper we optimized the routing and channel assignment in WMNs using genetic algorithms. We use GAs because the number of nodes in the network are too high to apply linear programming.

We evaluated the influence of the fitness function, the number of generations, and the elite set size on the resulting network fitness. The results show that the fitness function should be chosen with care because some functions lead to an unfair share of the resources. Using a fitness value built on weighted throughputs of all network flows results in the best solutions.

In addition to choosing a good fitness function, we illustrated that the elite set size should be chosen according to the population size. A small population with a large elite set size often results in a local optimum. The elite set size is also responsible for the required number of generations to get a good solution. We showed that with an elite set size of one third of the population size, a near optimal solution is achieved in our scenario after 400 generations.

Our GA can be used for WMN network planning to optimize the routes and channel assignment to achieve a max-min fair throughput allocation. In future work, we will extend our genetic algorithm to optimize the number of gateways and their location.

6. REFERENCES

[1] 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*, Special issue on Bio-Inspired Computing:11, April 2008.

[2] D. P. Bertsekas and R. G. Gallager. *Data networks*. Prentice-Hall, 1987.

[3] P. Calégari, F. Guidec, P. Kuonen, and D. Wagner. Genetic Approach to Radio Network Optimization for Mobile Systems. In *47th Vehicular Technology Conference (IEEE-VTC 97)*, Phoenix, AZ, USA, May 1997.

[4] K. P. Ferentinos and T. A. Tsiligiridis. Adaptive design optimization of wireless sensor networks using genetic algorithms. *Comput. Netw.*, 51(4), 2007.

[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*, pages 374–381, Washington, DC, USA, November 2005.

[6] J. H. Holland. *Adaptation in natural and artificial systems*. University of Michigan Press, Cambridge, MA, USA, 1975.

[7] S. Hussain, A. Matin, and O. Islam. Genetic algorithm for hierarchical wireless sensor networks. *JOURNAL OF NETWORKS*, 2(5), 2007.

[8] J. Jun and M. L. Sichitiu. The Nominal Capacity of Wireless Mesh Networks. *IEEE Communications Magazine*, 10(5):8–14, October 2003.

[9] D. Montana and J. Redi. Optimizing parameters of a mobile ad hoc network protocol with a genetic algorithm. In *Proceedings of the 2005 conference on Genetic and evolutionary computation*, 2005.

[10] R. Pries, D. Staehle, and M. Stoykova. On the Usability of Genetic Algorithms for Wireless Mesh Network Planning and Optimization. Technical Report 451, Universtiy of Würzburg, Würzburg, Germany, November 2008.

[11] D. Staehle, B. Staehle, and R. Pries. Max-Min Fair Throughput in Multi-Gateway Multi-Rate Mesh Networks. Technical Report 454, University of Würzburg, January 2009.

[12] 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*, 68(3):291–305, March 2008.