ARTICLE IN PRESS

Ad Hoc Networks xxx (2008) xxx–xxx

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

A genetic approach to joint routing and link scheduling for wireless mesh networks ☆

Leonardo Badia [a,*], Alessio Botta [a], Luciano Lenzini [b]

[a] IMT Lucca Institute for Advanced Studies, Piazza S. Ponziano 6, 55100 Lucca, Italy
[b] Department of Information Engineering, University of Pisa, via Diotisalvi 2, 56122 Pisa, Italy

ARTICLE INFO

Article history:
Received 28 February 2007
Received in revised form 18 April 2008
Accepted 21 April 2008
Available online xxxx

Keywords:
Wireless mesh networks
Routing
Link scheduling
Integer linear programming
Genetic algorithms

ABSTRACT

Wireless mesh networks are an attractive technology for providing broadband connectivity to mobile clients who are just on the edge of wired networks, and also for building self-organized networks in places where wired infrastructures are not available or not deemed to be worth deploying. This paper investigates the joint link scheduling and routing issues involved in the delivery of a given backlog from any node of a wireless mesh network towards a specific node (which acts as a gateway), within a given deadline. Scheduling and routing are assumed to be aware of the physical interference among nodes, which is modeled in the paper by means of a signal-to-interference ratio. Firstly, we present a theoretical model which allows us to formulate the task of deriving joint routing and scheduling as an integer linear programming problem. Secondly, since the problem cannot be dealt with using exact methods, we propose and use a technique based on genetic algorithms. To the best of our knowledge, these algorithms have never been used before for working out these kinds of optimization problems in a wireless mesh environment. We show that our technique is suitable for this purpose as it provides a good trade-off between fast computation and the overall goodness of the solution found. Our experience has in fact shown that genetic algorithms would seem to be quite promising for solving more complex models than the one dealt with in this paper, such as those including multiple flows and multi-radio multi-channels.

1. Introduction

Wireless mesh networks (WMNs) are an emerging class of networks, usually built on fixed nodes that are interconnected via wireless links to form a multi-hop network [1]. Their main goal is to provide broadband access to mobile clients who are just on the edge of wired networks. WMNs can be used where cable deployment is not feasible or is too expensive, such as in remote valleys or rural areas, but also in home environments and offices. End-users are served by nodes called *mesh routers*, which are generally assumed to be stationary. Mesh routers (MRs) are in turn wirelessly inter-connected so as to form a network *backhaul*, where radio resource management challenges come into play. Moreover, some mesh routers are generally provided with access (e.g. through wires) to the Internet and therefore can act as gateways for the entire WMN. Communication from any router to gateways is multi-hop.

Several WMN issues are thus common to those of multi-hop wireless networks, such as determining link scheduling in order to obtain high throughput efficiency [2,3] or selecting appropriate routes between source and destination [4,5]. However, the fact that mesh routers are fixed makes the WMN backhaul inherently different from distributed wireless networks (e.g. ad hoc or sensor

☆ Part of this work has appeared in the proceedings of *IEEE WiOpt 2007* [25].

* Corresponding author. Tel.: +39 0583 4326716; fax: +39 0583 4326565.
E-mail addresses: l.badia@imtlucca.it (L. Badia), a.botta@imtlucca.it (A. Botta), l.lenzini@iet.unipi.it (L. Lenzini).

1570-8705/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2008.04.005

networks), where the nodes may be portable devices. For example, problems such as energy consumption are no longer an issue. Also, the uncertainties about location of the terminals, due to mobility or difficulty to communicate, as well as their computational capability, are mitigated. This makes it sensible to opt for a centralized network management, as opposed to the distributed approaches used for ad hoc and sensor networks. In this case, MR nodes act in a coordinated fashion under the supervision of a network entity which determines the management based on global knowledge of the network topology and additional conditions.

Cross-layer approaches where the routing and link scheduling functionalities are jointly addressed has been extensively studied in multi-hop wireless networks [4–7]. For WMNs, one of the most relevant problems is to determine the shortest deadline within which a specified backlog vector can be jointly routed and scheduled between MRs and a gateway. This is the primary objective of the paper, which focuses on two major innovations.

Firstly, we formulate our problem through an ILP framework [8] by capturing the characteristics both of the WMN topology and of the radio channel, which allows us to determine the feasibility conditions for our problem. In the design of our framework we give particular emphasis to wireless interference and related aspects. In particular, to check whether simultaneous transmissions can be activated in an interference-free manner, we employ the so-called *physical interference model*, which computes the signal-to-interference ratio (SIR) at each active node and compares it with an appropriate threshold [9]. ILP formulations generally use another approach, named *protocol interference model*, which is simpler to apply but in our case may actually lead to oversimplifications. To the best of our knowledge, our ILP formulation is the only one available that explicitly addresses the physical interference model in its original version with linear constraints and binary variables. This is a novelty item of our analysis which, independently of the GA approach, distinguishes our investigation from other related work. The choice of this interference model is far from marginal. Actually, the physical interference model was shown to be highly superior to the protocol model for problems of this kind, as shown for example in [10,11]. Also, we believe that one more merit of our ILP framework is to leave room for possible extensions to specific cases of interest, in which a given objective function is proposed.

Secondly, we use genetic algorithms (GAs) to solve the cross-layer problem, and this technique copes reasonably well with our framework. It is known from the literature [2] that finding link activation patterns that satisfy traffic requirements and keep interference under control typically causes *NP*-complete problems. This means that the problem cannot be guaranteed to be solved in polynomial time. Exact approaches fail to find a solution in a reasonable time, even with not very large topologies, e.g. with 8–10 nodes.

GAs are an optimization technique which imitates evolutionary processes existing in nature [12]. They do not guarantee to find the best possible solution within a given amount of time: even if they are customized appropriately, they only solve the problem optimally with unlimited computational time at their disposal. However, GAs often work in practical cases as they provide a “good enough” solution in a reasonable time. Moreover, they appear to be ideal for handling discrete values, multiple constraints and also multiple objectives, as happens in problems of network planning [13,14], as well as with the problem discussed in this paper. We would like to stress that although GAs are often seen as a standard technique that can be used within any optimization framework, our problem requires many original issues to be implemented in the GA, which will be examined in detail in the paper.

The rest of this paper is organized as follows: in Section 2 we discuss the related literature. In Section 3 we outline the basic assumptions of the model and describe the variables and the notation utilized. In Section 4, we analyze the technical issues and formulate the ILP model. In Section 5 we describe GAs and discuss their application to our case study. Finally, in Section 6 we present numerical results, and we draw conclusions in Section 7.

2. Related work

Bio-inspired techniques, and more specifically GAs, are used in many studies of wireless networks for a preliminary network planning [13]. This also applies to WMN: in [14], an overview on how to use GAs to help the deployment of inter-urban mesh networks is reported. Another very recent paper [15] employs GAs for sensor networks. Even though the kind of network is different, the authors present some physical layer considerations about channel activation and the mutual interference of nodes.

However, in all these investigations dealing with network deployment, the usage of GAs is mostly motivated by the high complexity of the problem, which prevents it from being solved with exact techniques. On the other hand, the capability of GA of quickly giving good solutions to the problem is not exploited, since the time for the optimization process to converge is not as relevant as in shorter timescale problems such as routing or scheduling. Indeed, we believe that in our problem we are able to show this additional advantage offered by the computational efficiency of GAs. To our knowledge, this fact is not very frequently explored in the literature, thus making our work innovative.

For what concerns more specific aspects of WMN management, there are many papers in the literature [2–7,16,17] that can be related to the present work, as they investigate routing, scheduling, or both, through what can be seen as a linear programming framework. for example, [4] discusses routing optimization for wireless networks, but the main focus is on sensor networks, and, as commonly done for such systems, energy efficiency is considered as the objective. Also, there is no consideration about mutual interference of the nodes, which is instead very important for WMNs.

The analysis of [16] is, on the other hand, more applicable to our scenario, since it deals with throughput maximization and focuses on interference relationships. The authors discuss the need for joint routing and scheduling as significantly improve the performance with respect to separate optimization of these layers. Moreover, they review and compare different interference models, in particular protocol and physical interference model. The latter one is rarely used in the literature. A notable exception to this is [2]. In this paper, the evaluation of the SINR relationships is used to find feasible schedules in a WMN, and computationally efficient solutions are proposed to this end. Unlike our investigation, a link activation pattern is sought in order to meet pre-determined link weights which can correspond to the routes.

Our analysis, instead, jointly addresses routing and scheduling. This places it in the field of cross-layer solutions, whose investigation includes also channel assignment, as in [17], where a joint channel assignment and routing problem is approached, but is solved through heuristics. Exact solutions are discussed in [5], where a joint channel assignment and routing is proposed through an ILP framework. The scheduling issue is considered as the solution to a preliminary optimization, where only links that can be scheduled together are used. However, this paper only accounts for the protocol interference model, and in the routing phase non-integer link activation values are utilized. Similar considerations hold for [3], where a two-phase algorithm is introduced. First, a routing LP is solved that takes the protocol interference model into account but does not do any scheduling. This solution is then scheduled over time using a different algorithm.

In [6], an optimization approach is proposed to jointly solve link scheduling and routing, as we do in the present paper. However, the meaning of link scheduling is different, as in [6] the feasibility of a vector of rates is simply sought, whereas our aim is to determine a link activation pattern, which delivers the backlog of every node to the gateways within an assigned deadline. This framework is further extended in [7] to include channel assignment as well. However, again the fact that the binary variables of link activation are relaxed to rational values is shown in [7] to be a limiting assumption, which may lead to inaccuracies in the solution. Also, all previous contributions presenting a cross-layer approach take into account the protocol interference model, whereas we use the more suitable physical interference model.

3. Basic assumptions of the model

We represent the backhaul of a WMN as a directed graph $\mathbf{G} = (\mathbf{N},\mathbf{E})$, which consists of $N = |\mathbf{N}|$ nodes representing the mesh routers of the WMN, connected by *directed* edges of the set $\mathbf{E}$ corresponding to possible links between terminals. Notation $e = (i,j) \in \mathbf{E}$ means that $i \in \mathbf{N}$ is the transmitter node of link e and $j \in \mathbf{N}$ is the receiver. We denote with $\mathbf{Y} \subset \mathbf{N}$ the set of the gateways, which are anycast end destinations for the mesh routers. In general, not all pairs of nodes are connected through an edge. We denote the one-hop input and output neighbors set of a node i as $\mathbf{S}_i$ and $\mathbf{R}_i$. In other words, $\mathbf{S}_i$ and $\mathbf{R}_i$ are the set of nodes for which an edge exists in $\mathbf{E}$ to and from node i, respectively, i.e. $\mathbf{S}_i = \{j \in \mathbf{N} | (j, i) \in \mathbf{E}\}$ and $\mathbf{R}_i = \{j \in \mathbf{N} | (i, j) \in \mathbf{E}\}$.

Hereafter for the sake of simplicity we also assume that nodes can use a single power level. This is not a limiting assumption, as multiple power levels can be taken into account by considering multiple edges for the same pair of nodes, without changing the rationale of the analysis. Similarly, we assume that all nodes own a single radio interface and are enabled to transmit on a unique narrow frequency band. Indeed, the extension of WMN management to the multiple channel case looks promising and several standards are envisioned to explicitly include support for such a case. All these extensions (multiple channels, multiple power levels, etc.) can be seen as improvements of the basic framework discussed here and are left for further research on this topic.

We assume that the WMN system operates in synchronous time slotted mode where time slots are labeled via integer numbers $0, 1, \ldots, t$. This means that we consider a discrete time axis, where the sampling rate is equal to the slot length, whose choice depends on physical layer aspects only and is therefore out of the scope of the present paper.

Every edge $(i,j) \in \mathbf{E}$ is also associated with a transmission rate r_{ij} and a path gain g_{ij}. The former describes the number of packets, assumed to be constant, that can be sent during a time slot over the edge (i,j), whereas the latter is the ratio between received and transmitted power when node i transmits to j, which will be used in the following when modeling interference between transmission links. Both r_{ij} and g_{ij} variables can be collected into matrices $\mathbf{R} = (r_{ij})$ and $\mathbf{G} = (g_{ij})$. Another assumption made for analytical tractability is that it is not possible to underutilize an edge below the available rate r_{ij}, unless the transmitter does not have enough packets to send. This generally prevents the sender from splitting the data into parts smaller than the whole rate of an edge. However, this would be really beneficial in a negligible number of cases; thus, in practical cases this assumption is not restrictive.

To solve the joint link scheduling and routing problem, we define a 0-1 scheduling variable $x_{ij}(t)$ for every $(i,j) \in \mathbf{E}$, as

$$x_{ij}(t) = \begin{cases} 1 & \text{if } i \to j \text{ is active on time slot } t \\ 0 & \text{otherwise.} \end{cases}$$

In other words, $x_{ij}(t)$ denotes whether or not there is a data transmission (i.e. the link is activated) on time t. These variables are bound to be integer, varying over a discrete (slotted) time, so as to determine a time-division scheduling pattern for the WMN backhaul [2]. Similarly to the analysis presented by [6], we remark that the derivation of a scheduling pattern of links implicitly determines the routing as well. However, rather than working on a per-flow basis, we derive the routes by looking at the dynamics of the link activation over time. Unlike other papers [5,7], we impose the $x_{ij}(t)$ variables to be strictly binary and varying over discrete time t. In other words, we explicitly avoid relaxing constraints about variables to be integer, which is an approximation that can lead to strongly sub-optimal results.

For the sake of analytical tractability, we will consider a periodic scheduling, i.e. we focus on a frame of duration T slots, which is assumed to set the cycle of link activations. This means that links are activated according to the solution found for t between 0 and $T-1$, and this pattern can be repeated identically every T slots. We assume that each node supports a single flow towards a gateway. For the sake of simplicity, the amount of traffic per node is known in advance and is already available at the beginning of the frame at the non-gateway nodes. Any extension about packet arrivals delayed throughout the whole frame is left for future work. The goal within a single frame is to deliver the traffic to the gateways. Depending on the status of WMN backhaul links, this can be done by sending it directly to a gateway or by relaying to one or more nodes before reaching the destination gateway. In the latter case flow traffic is aggregated at some intermediate node towards a gateway.

The progress status of the transmission to the gateways is modeled through variables $q_i(t)$, which describe the queue length at each node i at time slot t. In reality, these are more like *auxiliary* variables, since, as will be shown in the following, they can be put in relationship through flow constraints with the binary variables $x_{ij}(t)$. For every t in $0, 1, \ldots, T-1$, we assume that $q_i(t)$ represents the amount of traffic in queue at node i, that needs to be delivered to one of the gateways before the end of the frame. The connection between $q_i(t)$ and $x_{ij}(t)$ is such that $q_i(t)$ represents the amount of data *before* the application of the transmissions identified by $x_{ij}(t)$, whereas $q_i(t+1)$ describes the outcome of these transmissions. For this reason, $q_i(t)$ varies in such a way that at the beginning of the frame $q_i(0)$ represents the overall amount of data (i.e. the aggregated demand from its associated end users) to deliver for node i, and $q_i(T)$ describes the residual backlog at node i after the application of the joint routing and scheduling pattern.

4. Problem formulation and main constraints

The problem of assigning meaningful 0-1 values to $x_{ij}(t)$ can be seen as a flow optimization problem subject to three different kinds of constraints. The constraints of the first kind describe the flow conservation and delivery of all traffic to the gateways. Also, two other types of conditions are needed to check the feasibility of the link activation pattern. Both of them are related to the feasibility of simultaneous activations of links, which is generically beneficial as it improves the transmission parallelism. Only *compatible* transmissions can be scheduled in the same time slot, where "compatibility" means "possibility to be used simultaneously". Modeling this property among *wireless* link transmission is challenging, and several models have been proposed [9]. To check whether two transmissions can coexist, two conditions must be met:

- the radio equipment of a single node is limited and can not be used for too many tasks (i.e., transmission/reception). This means that a node can either receive from a single source or transmit to a single destination. Wireless transceivers are usually *half duplex* [7], which means that they can not be used for reception and transmission at the same time.
- interference issues also need to be checked. Several models can be used, and we will refer to the physical interference model [9].

We classify three kinds of constraints: flow constraints, direct compatibility constraints, and interference constraints. These are discussed in their respective subsections.

4.1. Flow constraints

The flow constraints include flow conservation for every time slot t at each node:

$$\begin{aligned} q_i(t+1) = \max\left(0, q_i(t) - \sum_{j\in\mathbf{R}_i} x_{ij}(t) r_{ij}\right) \\ + \sum_{j\in\mathbf{S}_i}(\min(q_j(t), x_{ji}(t) r_{ji})), \\ \forall i \in \mathbf{N},\ \forall t = 0, \ldots, T-1. \end{aligned} \tag{1}$$

In the formulation of this constraint in a linear version simultaneous transmission and reception are allowed. In fact, the right-hand terms account for both incoming and exiting packets. However, the fact that the active outgoing links (in the first term) and the active incoming links (in the second term) can be at most one on aggregate is accounted for in the half duplex constraint included in the following. Additionally, at time T everything has to be delivered to the gateways:

$$\sum_{i\in\mathbf{N}} q_i(0) = \sum_{i\in\mathbf{Y}} q_i(T) \tag{2}$$

We also assume that the gateways do not generate traffic. The formulation of a related constraint is not strictly necessary, but it is useful to simplify the resulting algorithm. Thus, we require

$$q_i(0) = 0, \quad \forall i \in \mathbf{Y}, \tag{3}$$

$$\textstyle\sum_{j\in\mathbf{R}_i} x_{ij}(t) \leqslant 0, \quad \forall i \in \mathbf{Y}, \quad \forall t \in 0, 1, \ldots, T-1. \tag{4}$$

4.2. Direct compatibility constraints

The constraints that we call *direct compatibility constraints* relate to the impossibility of utilizing a transceiver equipment of a node for more purposes than is designed for. The limitations preventing nodes from multiple transmissions and receptions can be written as

$$\textstyle\sum_{j\in\mathbf{R}_i} x_{ij}(t) \leqslant 1, \quad \forall i \in \mathbf{N},\ \forall t = 0, \ldots, T-1, \tag{5}$$

$$\textstyle\sum_{j\in\mathbf{S}_i} x_{ji}(t) \leqslant 1, \quad \forall i \in \mathbf{N},\ \forall t = 0, \ldots, T-1. \tag{6}$$

However, wireless links are intrinsically half duplex, unless special techniques are employed, which implement full duplexing, such as directional antennas [18] or multiple channels [7]. If there is no frequency or spatial separation between transmitter and receiver, a transmission would destroy any simultaneous reception due to the self-interfering transmitted power. Thus, to account for a half duplex channel, the constraints above are simply merged so as to form:

$$\sum_{j\in\mathbf{R}_i} x_{ij}(t) + \sum_{j\in\mathbf{S}_i} x_{ji}(t) \leqslant 1\ \forall i \in \mathbf{N}, \quad \forall t = 0, \ldots, T-1. \tag{7}$$

4.3. Interference compatibility constraints

The physical interference model evaluates the signal-to-interference ratio (SIR) of every transmission and assumes that, in order to be successful, the received power at every active receiver i has to overcome a SIR threshold called γ_i. Even though γ_i can be a different value for every node i, if the traffic flows are homogeneous and the modulation techniques are the same, it is sensible to use the same threshold γ for all the nodes. Also, for the sake of simplicity and without loss of generality, we omit ambient noise terms, which could be included by considering the signal-to-interference-plus-noise ratio (SINR) instead of the SIR. This does not lead to any significant changes in the mathematical formulation. The interference compatibility constraint can be written as

$$\gamma x_{ij}(t) \leqslant \frac{g_{ij}x_{ij}(t)}{\sum_{k\in\mathbf{S}_j\setminus\{i\}} g_{kj}\sum_{\ell\in\mathbf{R}_k\setminus\{j\}} x_{k\ell}(t)}, \quad \forall(i,j)\in\mathbf{E}, \quad \forall t=0,\ldots,T-1, \tag{8}$$

which is in accordance with the most commonly used definition of SIR [9]. The physical meaning of this expression is as follows. Assuming all links use the same power, the activation of a link from i to j at time t, corresponding to having $x_{ij}(t)$ equal to 1, is subject to having an SIR on this link greater than or equal to γ, which is obtained by checking whether the ratio between the useful power (numerator term) over the interfering power plus noise (denominator) is greater than or equal to γ. Note that, to be meaningful, the interference constraint must be applied to active links only. This is the reason behind the mathematical formulation of Eq. (8), where if $x_{ij}(t) = 1$, the above inequality holds and the term $x_{ij}(t)$ can be removed from both left-hand and right-hand terms, whereas if $x_{ij}(t) = 0$, the above inequality is trivially always verified.

In order to have a linear constraint rather than a quadratic one, the following artifice is employed. Rearrange Eq. (8) as:

$$g_{ij}x_{ij}(t) \geqslant \gamma x_{ij}(t)\sum_{k\in\mathbf{S}_j\setminus\{i\}} g_{kj}\sum_{\ell\in\mathbf{R}_k\setminus\{j\}} x_{k\ell}(t), \quad \forall(i,j)\in\mathbf{E}, \quad \forall t=0,\ldots,T-1. \tag{9}$$

This is still a quadratic constraint, which, however, is equivalent to the following linear relationship:

$$g_{ij} \geqslant \gamma\sum_{k\in\mathbf{S}_j\setminus\{i\}} g_{kj}\left(\left(\sum_{\ell\in\mathbf{R}_k\setminus\{j\}} x_{k\ell(t)}\right) + x_{ij}(t) - 1\right), \quad \forall(i,j)\in\mathbf{E}, \quad \forall t=0,\ldots,T-1. \tag{10}$$

The equivalence between Eqs. (9) and (10) can be proven as follows. Observe that, due to constraint Eq. (7), the inner-most sum ($\sum_{\ell\in\mathbf{R}_k\setminus\{j\}} x_{k\ell}(t)$) is always less than or equal to 1. Thus, if $x_{ij}(t) = 0$ we have that Eq. (10) is trivially verified, as Eq. (9) was. Otherwise, i.e., if $x_{ij}(t) = 1$ the formulation coincides with the one of Eq. (9).

Even though an ILP formulation is possible, the solution is hard to find with exact methods. This happens since the problem can be shown to be *NP*-complete [2]. As it will be shown in Section 6, the joint routing and scheduling problem becomes untreatable even with a limited number of mesh routers, i.e. more than five nodes including a gateway. The computational complexity is also strongly dependent on T. Heuristic solutions [17] might work in certain cases, but they fail to adapt to different network scenarios.

For these reasons, we propose in this paper a self-configurable and efficient solution technique based on Genetic algorithms, which will be explained in detail in the next section.

5. A genetic approach for joint routing and scheduling in WMNS

Genetic algorithms are a meta-heuristic technique employed to solve optimization problems, which imitate *Natural Selection*, i.e. the process of adaptation to the environment performed by living beings [12,19]. GAs are an appealing approach to solve complex problems, such as the one stated in the previous sections. Among their most interesting features, GAs

- are able to find "good solutions" to an unconstrained problem in a reasonable time, and they *always* find at least one "good" suboptimal solution,
- does not require a differentiable objective function and can be tailored to handle any sort of constraint,
- can easily handle discrete problems by choosing a numerable alphabet of symbols (e.g., integer numbers) for the chromosome,
- can be adapted in certain parameters (e.g., number of individuals in the population), so as to scale well as the problem size increases,
- can be customized to include some heuristics and experts' knowledge in the generation of the initial population and in the design of the genetic operators.

For these reasons, we approach the problem formulated in Section 4 with GAs. We remark that this allows us to avoid any relaxation of the constraints, including the integer constraint of variables $x_{ij}(t)$.

5.1. Genetic algorithms: background

A GA determines, rather than a single solution, a whole *population* consisting of *individuals*, which are all candidate solutions to the problem. The distinctive features of each individual are mapped into a structure called *chromosome*. The chromosome is a string of *genes*, whose values can be chosen in a set of symbols. An application-dependant process generates the individual by decoding its chromosome. The symbols used as values of the genes can be binary, integer or real numbers, depending on the nature of the problem. Once an individual is generated, a *fitness function* is employed to evaluate its goodness as a solution to the problem. Usually, low fitness values are given to the best individuals (minimization problem). For the sake of simplicity, in the following we will blur the definitions of individual and chromosome.

A GA starts at time $t = 0$ with an initial population generated either randomly, or with some heuristic approach that exploits the knowledge of an expert in the problem domain. The algorithm then proceeds in steps called *generations*. At each generation t, a new population $P(t+1)$ is evolved from $P(t)$. As generations pass, the population should improve globally thanks to the application of *genetic operators* that mimic the natural evolution mechanisms. To this aim, the best individuals are chosen from $P(t)$ (*selection*) to be mated (*crossover*) and slightly modified (*mutation*), so as to create the new population $P(t+1)$.

The selection operator is used to decide which individuals in $P(t)$ should be chosen to generate $P(t+1)$. Optionally, an *elite* of the selected individuals (i.e. a small subset of the best performing individuals) survives and is moved from $P(t)$ to $P(t+1)$ without any change. The rest of the population is obtained through a crossover operator which chooses some of the individuals and *mates* them, that is, substitutes them with their *children*, which are newly generated individuals obtained by mixing the genetic material in the parents' chromosomes. The actual implementation of a crossover operation very much depends on the coding schema of the chromosome. Finally, the mutation operator is invoked to introduce some new genetic material in the population by randomly modifying the values of some genes. Again, different kinds of mutation operators can be defined to handle different sets of symbols. The population continues to evolve until a stopping criterion is fulfilled, the simplest being a maximum number of generations. Fig. 1 reports an overall pseudo-code description of the basic GA algorithm.

If crossover and mutation are sufficiently general, GAs can be shown to allow the exploration of the whole solution space. If an optimization goal is set, they are bound to find the optimal solution, even though there is no guarantee that it will be the optimal one, nor can the time to find it be predicted. However, since the execution time is generally rapid, GAs are also interesting for practical purposes as they can be employed as fast procedures to find a "good enough" solution to the problem. This gives them an advantage with respect to exact techniques such as Branch and Cut used in commercial solvers, since any solution produced by a GA is directly applicable and simply improves as long as the number of generation increases. Therefore, GA could be used to operate online WMN management, where the solution may be iteratively updated. This could also be an interesting development of the present analysis for future work.

Note that, while some classes of problems can be solved by directly applying a basic version of a GA, more often the development of such an algorithm for a specific problem requires an elaborated engineering process involving a good amount of design and tailoring. Indeed, the design of a GA includes finding suitable representation schemata, coding strategies, genetic operators, values of parameters, etc. Furthermore, for constrained problems like the one under investigation in this paper, this additional processing is mandatory as we are forced to select and adapt appropriate constraint-handling methods from the ones available in the literature [20,21].

```
initialize P(0)
repeat
    evaluate P(t) via fitness_function;
    apply selection to choose parents;
    apply crossover to generate offspring
    apply mutation to offspring
    generate P(t+1)
    increase t by 1
until a termination condition is verified
```

Fig. 1. The pseudo-code of a basic GA.

5.2. A GA-based approach for the ILP problem

The first step when designing a GA is to identify how to mathematically represent a solution as an individual, in order to create a population. Given the natural binary formulation of the problem, a trivial solution is to use a link-based schema [22], where the chromosome is composed by $N \cdot (N-1) \cdot T$ binary genes, each directly mapping a $x_{ij}(t)$. The link-based coding schema consists of concatenating the linearization of the adjacency matrices for each time slot. Since we assume that the gateways are not producing traffic, and thus do not need the activation of any output link, we can remove the rows of the adjacency matrix with index $i \in \mathbf{Y}$, reducing the number of bits in the chromosome to $N \cdot (N-1-|\mathbf{Y}|) \cdot T$. We observe that, given the half duplex constraint in Eq. (7), a maximum number of $N/2$ links per slot can be activated concurrently. This observation justifies the search for a more compact coding schema, which can be based on nodes rather than links. In the node-based coding schema, each time slot is coded into a string of $|\mathbf{N}\backslash\mathbf{Y}|$ integers, each one ranging in $[0,N]$. Thus, the chromosomes are coded as sequences of $(N-|\mathbf{Y}|)\cdot T$ integers, sorted first internally to each frame by any ordering of the nodes, then frame-by-frame in an increasing order. Formally, the genetic map of any individual is: $(y_1(0), y_2(0), \ldots, y_{N-|\mathbf{Y}|}(0), \ldots y_n(t) \ldots, y_1(T-1), y_2(T-1), \ldots, y_{N-|\mathbf{Y}|}(T-1))$. Given a generic $y_n(t)$, the decoding schema is:

$$\begin{aligned} &y_n(t) = 0 \Rightarrow x_{n,y_n(t)}(t) = 0, \\ &y_n(t) > 0 \Rightarrow x_{n,y_n(t)}(t) = 1. \end{aligned} \tag{11}$$

In other words, if $y_n(t) = 0$, node n is not transmitting at time t (that is, $x_{nj}(t) = 0\ \forall j \in \mathbf{N}$), whereas if $y_n(t) = k$, $k > 0$, node n is transmitting to node k at time t, and thus $x_{nk}(t) = 1$ and $x_{nj}(t) = 0\ \forall j \in \mathbf{N}$, $j \neq k$. This coding schema can be implemented using just $\lceil \log_2(N+1) \rceil \cdot (N-|\mathbf{Y}|) \cdot T$ bits, so as to scale much better than the link-based one of [22] when N increases.

To generate an initial population, composed of 500 individuals, we randomly assign a value in $[0,N]$ to each gene of each individual in the population. The GA proceeds by iteratively modifying the population, that is, by cyclically applying the selection, the crossover, and the mutation operators as described in Section 5.1. As the selection operator, we use the robust and well-known *stochastic universal sampling* [19]. As regards the other operators (crossover and mutation), we developed our customized versions. Our coding schema has two granularity levels: the *node level*, represented by a single gene that codes the activation of a link between two nodes, and the *slot level*, that is, the overall configuration of the network for one time slot. We designed our operators so as to work on both levels of granularity.

The crossover operator is the *0.5-uniform crossover* [19]. The standard version of this operator chooses the value of each gene in the chromosome of a child as the value of either the first or the second parent, with a uniform probability. This is a node level granularity. However, it can be useful to adopt a similar approach also on the slot level. In fact, in this way a good concurrent allocation of links in a given slot may be spread over the population, if the evolution process recognizes it as beneficial. Thus, our modified uniform crossover may act, with a uniform probability, on one of the two different granularities, mixing single integers or whole time slots from the two parents to generate the child.

A similar approach was used to develop the mutation operator. We recall that the aim of this operator is to introduce some local modifications of the individuals in the current population in order to explore new possible solutions. Thus, our mutation operator can perform, with uniform probability, one of the following operations:

- mutate the chromosome on a node level granularity by a uniform random mutation [19], i.e., each gene can be randomly changed to a different value in $[0, N]$, with a mutation probability of 0.1,
- scramble some of the time slot of the chromosome (e.g. switch the transmission patterns of slot number 1 and number 5),
- replace some time slots with a duplicate of other slots of the same chromosome (e.g., replace the transmission pattern of slot number 5 by copying that of slot number 1),
- replace some time slots with all-zero slots (i.e. all transmissions in these slots are turned off).

With the aforementioned specifications, it is not guaranteed that the individual obtained from the crossover and mutation processes will be feasible. However, note that there are constraints that must necessarily be satisfied by each individual generated during the algorithm, and constraints that can be unsatisfied by some individuals. The first class includes the direct compatibility constraints of Eq. (7). The second class includes the flow constraints of Eqs. (1) and (2), and the interference compatibility constraints of Eq. (10). The reason of this classification is that the constraints of the first class are the basis to coherently derive the ILP formulation in Section 4. Instead, even though the constraints of the second class also determine feasibility conditions for the final solution, fulfilling all of them is not strictly necessary to have a "good" individual in the evolution process. For example, observe that the half duplex constraint Eq. (7) is also implicit in the formulation of both the flow constraint Eq. (1) and the linear version of the interference constraint Eq. (10). For this reason, we used two different techniques to approach these classes of constraints.

First class constraints are always satisfied by means of a *repair* process, which is performed after the application of any genetic operator that might produce an infeasible individual. For instance, suppose that the mutation operator generated an individual in which, at some time, a node has two input links activated in the same slot. In this case, the repair randomly deactivates one of the links, fixing the corresponding gene and setting its value to 0. Another case handled by the repair procedure is the activation of an output link by a node that has no more packets to send. Also this transmission is "turned off" since it does not correspond to a physical transmission and may erroneously be recognized as causing interference.

No action is taken instead to repair violations to the second class of constraints. Since repair is performed each time a genetic operator is applied, it must be designed to be an extremely fast and efficient routine. Thus, we decided to repair only the constraints of the first class. Moreover, we only consider repair through deactivations of links selected in a random fashion among the ones which are violating a constraint of the first class. Nevertheless, further research could lead to a more effective repair process based on a pre-evaluation of all the possible fixed individuals generated by an infeasible one. The repair algorithm acts by visiting all the genes in a time slot in a sequential fashion according to a random order, and by deactivating links when a transmitting node is conflicting with another already visited one.

Even though constraints in the second class are handled by allowing infeasible individuals to survive in the population, we give lower values of the fitness function to those individuals. This means that even though these individuals are kept in the population, their constraint violation is recognized by means of a fitness decrease, realized by means of a *penalty function*. The penalty is computed in the following way:

$$\rho = \sum_{i \in \mathbf{N}} q_i(0) - \sum_{i \in \mathbf{Y}} q_i(T) + \sum_{t=0}^{T-1} \sum_{(i,j) \in \mathrm{E}} p_{ij}(t), \tag{12}$$

where $p_{ij}(t)$ describes the interferences violations at frame t, that is

$$p_{ij}(t) = \begin{cases} 1 & \text{if } g_{ij} - \gamma \sum_{k \in \mathbf{S}_j \setminus \{i\}} g_{kj} \left(\left(\sum_{\ell \in \mathbf{R}_k \setminus \{j\}} x_{k\ell}(t) \right) \right. \\ & \left. \quad + x_{ij}(t) - 1 \right) < 0, \\ 0 & \text{otherwise}. \end{cases} \tag{13}$$

The fitness function can also incorporate, by a linear combination, some metrics of the network that we want to optimize. Interestingly, the best results, both in terms of convergence speed and goodness of the solution found, were given by also including a penalty computed according to the number of activated links per slot. The rationale behind this approach is that the delivery of packets to a gateway should be performed with as few active links as possible, since, the higher the number of active links, the higher the interference and the lower the number of alternative routes which can be discovered. More formally, let $t^* = \max_t(t| \ \forall j \in \mathbf{Y}, \exists i \in \mathbf{N} \setminus \mathbf{Y} \text{ s.t. } x_{ij}(t) = 1)$ be the last time slot in which a link to a gateway is active. The proposed metric is defined as:

$$\mathrm{obj} = \frac{\sum_{t=0}^{t^*} x_{ij}(t) - \sum_{t=t^*+1}^{T-1} x_{ij}(t)}{T \cdot (N+1)}. \tag{14}$$

The denominator coefficient $T(N+1)$ is introduced just as a normalization constant in order to keep the value of obj always less than 1, since it should not be stronger than the violation of a constraint. This metric tries to combine two contrasting objectives. On the one hand, it is preferable to avoid unnecessary link activations, in order to decrease interference. So, if two solutions deliver the same amount of traffic, the ones with fewer active links is preferred. However, if there is still traffic left to be delivered to the gateway, activating links is encouraged in order to create more routes to the gateway. The final fitness function F is taken as $F = W \cdot \text{obj} - \rho$, where the weight W was empirically set to 0.5. A feasible function, i.e., which delivers all the traffic without violating interference constraints, satisfies $F \geqslant 0$.

We remark that F only aims at obtaining feasible solutions without any specific technical goal than testing the convergence speed and overall goodness of GAs in dealing with such WMN problems. However, this vanilla approach can be easily modified to incorporate any other network measure, such as global interference, throughput, minimum number of time slots, etc.

We used a hybrid stopping condition which still stops the GA after a maximum of 200 generations and tries to perform an early stop in two cases:

- a good feasible solution is found quickly, or
- the problem seems to be infeasible.

The idea is that, if we already are in the feasible region, we are not interested in optimizing the network metric much more, and that if the problem seems infeasible, we should give up early with the best solution found. Thus, in the former case, we perform an early stop if, after a first feasible solution is found (that is, a solution with $F \geqslant 0$), we do not find any other better solution in 5 generations. In the latter case, we perform an early stop if we have not found any feasible solution and we have noticed no improvements in the last 50 generations; in such a case, we still return the best solution found (though its F is lower than 0).

6. Numerical evaluations

To evaluate the performance of our GA, we focus on a grid topology with 12 nodes placed at the vertices of 30 m $\times$ 30 m squares, as reported in Fig. 2. We assume that the 12 grid intersections are occupied by a single node and one of them, which can occupy positions A, B, or C, acts as a gateway.

We believe that this network size is large enough to be representative of the management achieved for larger networks also. This is justified by the following observations. First, notice that the real important aspect in such a multi-hop scenario is not the number of nodes per se, but the depth of multi-hop. The topology shown in Fig. 2 implies that nodes opposite to the gateway have to send their packets through at least 3 or 4 hops. This makes our multi-hop analysis sensible, and such deep multi-hop is rarely addressed in the literature. We also remark that, thanks to the use of the physical interference model, we are able to raise the limit of multi-hop from two, which is the practical limitation for the protocol model [22] to three–four hops. However, a number of traversed hops larger than 4 would still imply a poor performance, since the network parallelism would be strongly decreased due to bottleneck effects. More in general, not only deeper multi-hop is difficult to manage, but would also require a larger number of gateways [22]. Networks with more than 12 nodes and more than 1 gateway can be easily framed into our analysis as well, if we think of grouping nodes into clusters. In practice, this means that we can easily extend the application of the proposed GA from the scenario represented in Fig. 2 to clustered cases where clusters comprise up to 12 nodes.

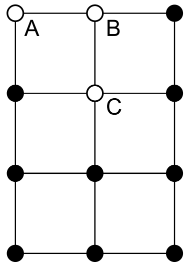

Fig. 2. Grid topology with 12 nodes, including three possible gateway positions (white dots).

The scheme of Fig. 2 is actually employed to derive multiple network topologies. Even though the node placement is identical for any instance of the same scenario, the network topology can be different since we evaluate channel gains and rates according to a time-varying channel. For each scenario we generated 10 different topology instances by varying the channel gain. The channel gain of an edge having length equal to d consists of a path loss term proportional to $d^{-3.5}$ and a shadowing term. This last part is obtained by evaluating a log-normal random variable with standard deviation equal to 5 dB, but we also consider a correlation model which is a two-dimensional extension of the Gudmundson's model [23], which gives a correlation at 100 meters equal to 0.6.

The rate of the communication link (i,j) is a discrete value function of the gain g_{ij}. Table 1 reports the rate values assigned according to the attenuation with respect to the average path loss at 1 meter. The table is to be read as follows: if the gain g_{ij} falls within the range reported in the left-hand column, the rate r_{ij} is equal to the value in the right-hand column, expressed in packets/slot. Finally, we assume a SIR target γ equal to 3.5 dB for all the receivers.

We implemented the GA algorithm as discussed in Section 5, using the procedures contained in the genetic algorithm toolbox of MATLAB Release 2006a [24] as a basis. We executed the GA five times for each topology instance, in order to avoid particularly unfortunate cases where the GA terminates in a dead end of the state space. Note that the solutions found in this manner are still average values, over 10 different topology scenarios.

As performance metrics, we considered both the fraction of cases (i.e. topology instances) in which the GA finds

Table 1
Rate assignment as a function of the channel gain

Channel gain	Rate (pkt/slot)
$g_{ij} \geqslant -53$ dB	11
-53 dB $> g_{ij} \geqslant -60$ dB	5
-60 dB $> g_{ij} \geqslant -65$ dB	2
-65 dB $> g_{ij} \geqslant -70$ dB	1
-70 dB $> g_{ij}$	0

a feasible solution (i.e., a solution which allows the delivery of the backlog from any node to the gateway within the frame duration) within the above termination conditions, and the delivery ratio (i.e., the ratio of delivered packets over the total traffic of each node) of the best solution found.

In the following, we show detailed results considering GA performance. In Figs. 3–5, we show the performance of the GA in the 12-node scenario, for the case where the load to deliver to the gateway is fixed for each node to six packets, and we vary the frame length T. Different positions of the gateway are considered. The number of nodes is too high to allow a detailed comparison with exact methods, which are theoretically possible within our ILP framework, as performed in [25]. The interested reader may refer to this paper for a detailed comparison. Here, we simply report that sample executions of exact search exhibit good agreement with the GA, on the same line of [25]. We remark that, with respect to exact solution techniques, not only is the genetic algorithm more computationally efficient, but it also has the considerable advantage of being more scalable.

As expected, the fraction of feasible solutions found is an increasing function of the frame length, since a larger T offers a higher degree of freedom in accommodating the packets over the schedule. When $T \geqslant 20$, a solution is always found even in the worst scenarios (gateway in positions A and B). The scenario with the gateway in position C performs better in this sense, since all packets are delivered in 17 time slots.

However, it is worth noting that, even when the GA fails to find an exact solution, either because the optimization

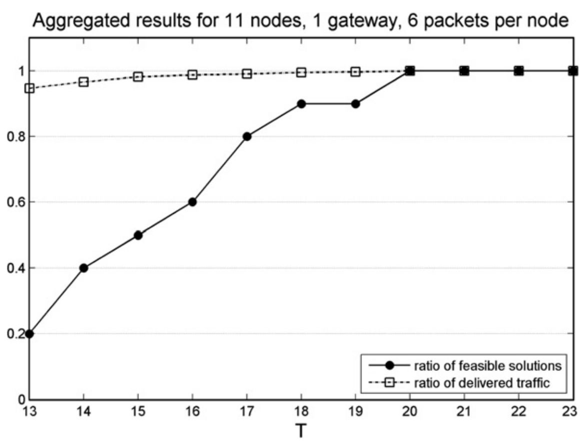

Fig. 3. Twelve node topology, gateway in position A. GA performance as a function of the frame length.

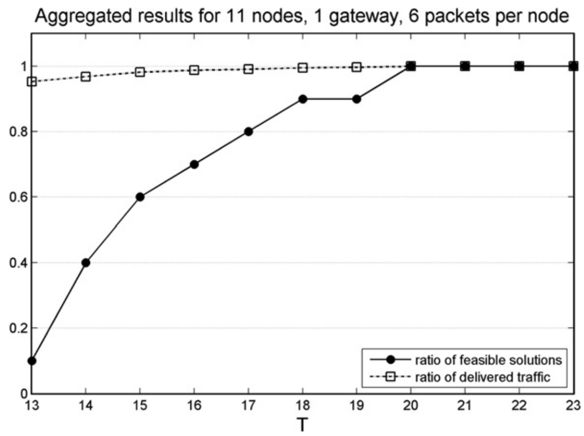

Fig. 4. Twelve node topology, gateway in position B. GA performance as a function of the frame length.

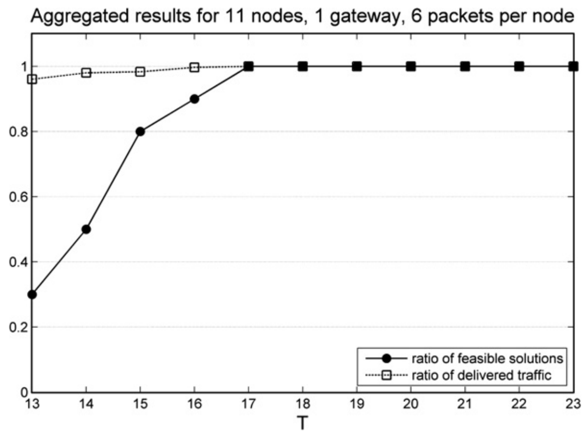

Fig. 5. Twelve node topology, gateway in position C. GA performance as a function of the frame length.

stops to a suboptimal value or since it does not actually exist, the delivery ratio achieved by GA is still fairly high. The GA is always able to deliver more than 90% of the traffic, even for small values of T. This represents a very important advantage of GA in practical implementation, as it gives a solution in any case, and, when this is not the optimal one, it is still very close to it.

For what concerns the position of the sink, we notice again that the scenario with the gateway in position C achieves the best performance. Also, observe that, even though having the gateway in position B does not decrease the overall length of the schedule with respect to position A, the delivery ratio is slightly higher for intermediate values of T.

Fig. 6 shows the result of another similar investigation, where T is kept constantly equal to 20 and instead the load per node is changed. The scenario B is considered, even though the other cases obtain qualitatively similar results to the former case (i.e., having the gateway in position A or C is slightly worst or better, respectively). With respect to Figs. 3–5, the trend is reverted, since the higher the load the more difficult it is to have a solution and also to find it

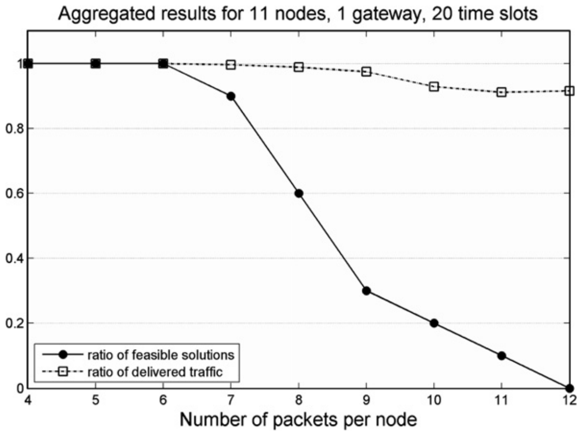

Fig. 6. Twelve node topology, gateway in position B. GA performance as a function of the load per node.

through the GA. The ratio of feasible solutions found by the GA decreases rapidly; however, also for this graph we still see a very high ratio of delivered traffic. For example, if the load per node equals 12 packets, none of the topologies are solved by the GA. However, the delivery ratio is larger than 90%. From an information theory point of view [9], these curves may also be used to discuss network capacity. In this case, the "critical" load of the network, i.e., the value around which the fraction of feasible allocations drops significantly, is 8/9 packets per node. This corresponds with an average utilization of about 80% of the available data rates.

In [25], topologies with up to nine nodes were considered. The results presented here for a 12-node topology suggest that the GA scales sufficiently well as the size of the topology increases. Next, we investigate in more detail the computational complexity of the GA, in order to have comparison results with the exact techniques. To this end, Fig. 7 reports the result of a complexity analysis for a small 5-node scenario, in order to have results for an exact technique also. To this end, we also implemented an exact ILP solution technique using the LPSOLVE model solver [26]. In such a small scenario, a feasible solution was found by both algorithms, and we measure the complexity through the following performance indices: (a) number of evaluations of the fitness function made by GA; (b) simplex iterations performed by LPSOLVE. This gives a rough idea

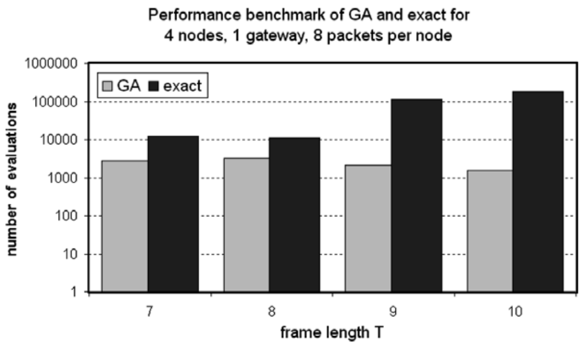

Fig. 7. Computational complexity benchmark.

of how the algorithms scale when the size of the problem increases. Moreover, we vary the frame size T, since the complexity of the problem strongly depends on it.

As shown in Fig. 7, whereas the exact technique explodes already when T is changed from 7 to 10, the complexity of the GA stays more or less constant. Indeed, it even slightly decreases when the frame length is very high, since in these cases a solution is found very rapidly, as is reasonable to expect. This proves how good the GA is in finding a quick valid solution to easy problems. In practical cases, it is possible that the network resources are not fully utilized, as, for example, the traffic per node may be significantly lower than what can be allocated over an entire frame. However, it can also happen that exact techniques fail to quickly solve the problem, due to its large size. In this case, GAs can be seen as a very good alternative to heuristics, since by modifying their meta-parameters they are able to adapt themselves to different problem instances.

7. Conclusions

In this paper we have investigated joint link scheduling and routing strategies for wireless mesh networks. We have proposed an optimization framework making use of an entirely ILP formulation, where we particularly aimed at keeping the integer constraint of link activation variables and adopting the more realistic physical interference model. This led us to the formulation of an ILP problem whose solution captures both levels of link scheduling and routing in a cross-layer fashion, by supplying a periodic link activation pattern which is able to deliver a given amount of traffic to the network gateways.

The main findings are that the physical interference model is still treatable within the ILP framework. The hard part of the problem is due to the integer constraint, which causes the computational complexity to grow exponentially, both in the number of nodes and in the length of the time frame. Due to the inherent complexity of solving such a problem, we also proposed a fast and efficient solution technique, namely genetic algorithms. After having discussed theoretical principles of GAs, we introduced several original implementation parts in order to obtain efficient GAs for the problem under investigation.

Finally, the proposed GA has been tested in sample wireless mesh network scenarios. The numerical evaluations show that the GA is able to solve both scenarios reasonably well, and also scales well, whereas exact optimization techniques are unable to solve the larger topologies. The solution found by GA is not always optimal. However, it is always very close to the optimum. Moreover, the GA is a very good approach for realistic cases where feasible solutions are easy to find, since in these cases they converge very rapidly, compared to other techniques, to a solution which is good in practice. For these reasons, we believe that GAs could be very useful tools for a centralized management of WMNs due to their good level of efficiency in a reasonable computational time.

Future research could be devoted to further optimizing the proposed GA, for example to enable it to deal with nonbinary structure in order to better manage larger networks and/or decrease the computational complexity even more. Also, we envision that GAs could be used in more complex problems characterized by multiple flows and multi-radio multi-channels, due to their ability to cope with multi-dimensional constraints and objectives.

References

[1] I.F. Akyildiz, X. Wang, W. Wang, Wireless mesh networks: a survey, Comput. Networks 47 (2005).
[2] G. Brar, D. Blough, P. Santi, Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks, Proc. ACM Mobicom (2006) 2–13.
[3] V.S. Anil Kumar, M.V. Marathe, S. Parthasarathy, A. Srinivasan, Algorithmic aspects of capacity in wireless networks, Proc. ACM SIGMETRICS (2005) 133–144.
[4] J.-H. Chang, L. Tassiulas, Maximum lifetime routing in wireless sensor networks, IEEE/ACM Trans. Network 12 (4) (2004) 609–619.
[5] M. Alicherry, R. Bhatia, L. Li, Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks, Proc. ACM Mobicom (2005) 58–72.
[6] M. Kodialam, T. Nandagopal, Characterizing achievable rates in multi-hop wireless networks: the joint routing and scheduling problem, Proc. ACM Mobicom (2003) 42–54.
[7] M. Kodialam, T. Nandagopal, Characterizing the capacity region in multi-radio multi-channel wireless mesh networks, Proc. ACM Mobicom (2003) 73–87.
[8] A. Wolsey, Integer Programming, John Wiley and Sons, New York, 1998.
[9] P. Gupta, P.R. Kumar, The capacity of wireless networks, IEEE Trans. Inf. Theory 46 (2) (2000).
[10] M. Burkhart, P. von Rickenbach, R. Wattenhofer, A. Zollinger, Does topology control reduce interference?, Proc ACM Mobihoc (2004) 9–19.
[11] L. Badia, A. Erta, L. Lenzini, M. Zorzi, A general interference-aware framework for joint routing and link scheduling in wireless mesh networks, IEEE Network 22 (1) (2008) 32–38.
[12] K.A. De Jong, Evolutionary Computation: A Unified Approach, MIT Press, 2006.
[13] B. Al-Bassam, A. Alheraish, S.H. Bakry, A tutorial on using genetic algorithms for the design of network topology, Int. J. Network Manage. 16 (4) (2006) 253–262.
[14] K.-T. Ko, K.-S. Tang, C.-Y. Chan, K.-F. Man, S. Kwong, Using genetic algorithms to design mesh networks, IEEE Comput. Mag. 30 (8) (1997) 56–60.
[15] K. Ferentinos, T. Tsiligiridis, Adaptive design optimization of wireless sensor networks using genetic algorithms, Comput. Networks 51 (2007) 1031–1051.
[16] K. Jain, J. Padhye, V.N. Padmanabhan, L. Qiu, Impact of interference on multi-hop wireless network performance, Proc. ACM Mobicom (2003) 66–80.
[17] A. Raniwala, K. Gopalan, T.-C. Chiueh, Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks, ACM Mobile Comput. Commun. Rev. 18 (2) (2004) 50–65.
[18] J.A. Stine, Exploiting smart antennas in wireless mesh networks using contention access, IEEE Wireless Commun. 13 (2) (2006) 38–49.
[19] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Program, third ed., Springer-Verlag, Berlin, 1996.
[20] Z. Michalewicz, M. Schoenauer, Evolutionary algorithms for constrained parameter optimization problems, Evolution. Comput. 4 (1) (1996) 1–32.
[21] C.A. Coello Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Comput. Methods Appl. Mech. Engrg. 191 (11–12) (2002) 1245–1287.
[22] J. Jun, M.L. Sichitiu, The nominal capacity of wireless mesh networks, IEEE Wireless Commun. 10 (5) (2003) 8–14.
[23] M. Gudmundson, Correlation model for shadow fading in mobile radio systems, Electron. Lett. 27 (23) (1991) 2145–2146.
[24] The Mathworks, MATLAB Release 2006a, <www.mathworks.com>.
[25] L. Badia, A. Botta, L. Lenzini, Joint routing and link scheduling for wireless mesh networks through genetic algorithms, in: Proc. WiOpt, 2007.
[26] M. Berkelaar, K. Eikland, P. Notebaert, lp_solve: Open source (Mixed-Integer) Linear Programming system, available at (website): <lpsolve.sourceforge.net/5.5/>.

Leonardo Badia was born in Ferrara, Italy, in 1977. He received the Laurea Degree (with honors) in Electrical Engineering and the Ph.D. in Information Engineering from the University of Ferrara, Italy, in 2000 and 2004, respectively. During 2002 and 2003, he was on leave at the Radio System Technology Labs (now Wireless@KTH), Royal Institute of Technology of Stockholm, Sweden. After having been with the Engineering Department of the Università di Ferrara, Italy, he joined in 2006 the Institute for Advanced Studies "IMT Lucca", in Lucca, Italy, where he is currently an assistant professor. He also collaborates with DEI, University of Padova, Italy. His research interests include energy efficient Ad Hoc Networks, transmission protocol modeling, Admission Control and economic modeling of Radio Resource Management for Wireless Networks. He serves also as reviewer for several periodicals in the communication area.

Alessio Botta was born in Livorno, Italy, in 1980. He received the M.Sc. (with honors) in Information Technology Engineering for Business Management in 2004 from the University of Pisa, Italy. Since 2005 he has been a PhD Student in Computer Science and Engineering at the Institute for Advanced Studies "IMT Lucca", in Lucca, Italy. During 2006, he was on leave at the Department of Maths and Computer Science, Suffolk University, Boston MA, USA. His research interests are in the field of adaptation of Fuzzy Systems, Constrained Genetic Programming, and Computational Intelligence techniques for Global Computing.

Luciano Lenzini holds a degree in physics from the University of Pisa, Italy. He joined CNUCE, an institute of the Italian National Research Council (CNR), in 1970. Starting in 1973, he spent a year and a half at the IBM Scientific Centre in Cambridge, Massachusetts, working on computer networks. He has directed and been involved in several national and international projects and also European projects of the 5th and 6th framework program. In 1994, he joined the Department of Information Engineering of the University of Pisa as a Full Professor. His current research interests include the design and performance evaluation of: MAC protocols for wireless networks, architectures and protocols for mesh networks, and the Quality of Service provision in integrated and differentiated services networks. He is currently on the Editorial Boards of Computer Networks and the Journal of Communications and Networks as area editor for wireless networks. He served as general chairman for the 1992 IEEE Workshop on Metropolitan Area Networks, for the 2002 European Wireless Networks and for the First International Conference on Performance Evaluation Methodologies and Tools (ValueTools) which was held in Pisa in October 2006, of which he is a co-founder.