A Bioinspired Scheduling Strategy for Dense Wireless Networks Under the SINR Model

Vinicius Fulber-Garcia[1](✉), Fábio Engel[2], and Elias P. Duarte Jr.[1]

[1] Federal University of Paraná, Curitiba, Brazil
{vfgarcia,elias}@inf.ufpr.br
[2] Technological Federal University of Paraná, Toledo, Paraná, Brazil
fabioe@utfpr.edu.br

Abstract. Novel wireless networking technologies such as massive Internet-of-Things and 5G-and-beyond cellular networks are becoming increasingly denser. The SINR model can improve the performance of dense wireless networks by taking into consideration the effects of interference to allow multiple simultaneous transmissions in the same coverage area. However, transmission scheduling under the SINR model is an NP-hard problem. This work presents a bioinspired solution based on a genetic heuristic. The Genetic-based Transmission Scheduler (GeTS) produces efficient transmission schedules, increasing the number of simultaneous transmissions (*i.e.*, spatial reuse). Simulation results are presented, including a convergence test and a comparison with the optimal algorithm and another heuristic.

1 Introduction

There is a noticeable trend in networking technologies towards high-density. Examples include cellular networks [3] and massive Internet-of-Things [8]. As a wireless channel is a shared medium, interference among multiple transmissions must be taken into consideration to allow the multiple devices to communicate efficiently. One of the most common ways to deal with interference is to define a schedule for the transmissions, separating transmitting devices in space or time [19]. Scheduling has also to take into consideration other characteristics of the communication channel, such as the fact that the power of the transmitted signal decreases with the distance. Efficient transmission schedules are essential to guarantee the latency and throughput requirements of ultra-dense wireless networks.

The SINR (Signal-to-Interference-and-Noise-Ratio) model has been used to represent the effect of cumulative interference on signal reception in wireless networks [9]. In the SINR model, it is necessary to schedule transmissions into slots. This model allows spatial reuse: multiple simultaneous transmissions by devices in the same coverage area can be scheduled to the same time slot. Those trans-

A. Abraham et al. (Eds.): ISDA 2022, LNNS 715, pp. 228–238, 2023.
https://doi.org/10.1007/978-3-031-35507-3_23

missions do interfere with one another but are possible as long as the power level of an interfering signal transmitted from one source is low enough not to prevent the proper reception of signals transmitted from the other sources. Spatial reuse improves the efficiency of ultra-dense networks, as the amount of simultaneous transmissions increases, and each device waits less time to communicate. Thus, the objective is to obtain the minimum schedule that allows all devices to communicate.

The SINR scheduling problem has been proven to be NP-hard [6]. For this reason, much of the work in this area explores approximation algorithms [1,2,9,15]. Although those algorithms are important from the theoretical point of view, they have little application in practice [18]. The fact that there is still a need for practical and efficient scheduling algorithms for SINR wireless networks has been the main motivation for us to propose a novel strategy to solve the problem.

This work presents a bioinspired solution to the problem of scheduling in wireless networks under the SINR model, which is called the Genetic-based Transmission Scheduler (GeTS). The proposed strategy consists of a scheduling algorithm for dense wireless networks that is based on the TDMA access mechanism (*Time Division Multiple Access*). The strategy is based on the so-called "down-to-earth heuristic" designed to improve spatial reuse: each device only communicates with its closest device [5]. GeTS employs a population of individuals – each representing a candidate schedule to solve the problem – that evolves over generations. To do so, we designed crossover and mutation mechanisms that allow the efficient exploration and exploitation of the search space. The objective is to find the schedules of minimum size, *i.e.*, with the minimum number time of slots possible. Besides presenting a convergence test to show the feasibility of the proposed algorithm, we evaluated the ability of GeTS to achieve near-optimal schedules for different numbers of devices, including comparisons with two other alternative algorithms.

The remainder of this work is organized as follows. Section 2 gives a brief overview of the SINR model and the SINR scheduling problem. Section 3 introduces GeTS, the Genetic-based Transmission Scheduler. Section 4 shows the results of two experiments conducted to evaluate GeTS. Finally, conclusions are presented in Sect. 5.

2 The SINR Model and Scheduling

The SINR (Signal-to-Interference-plus-Noise Ratio) model, also known as the physical interference model or physical model, is a wireless network model that considers the effects of cumulative interference on signal reception and the effects of path loss on the transmitted signal power. This model has been shown empirically [13,20] to provide a good approximation of real wireless communication environments.

The SINR model employs the signal-to-noise interference ratio metric to determine the quality of wireless communication links. This metric defines a criterion to determine whether a given transmission can take place. Equation 1 shows how the so-called SINR threshold γ is employed to determine whether a signal sent by a device i can be correctly received by a device j. The SINR threshold is computed based on three properties of wireless communications. The first is path loss, a property related to signal propagation according to which the power level of the transmitted signal P_{Ti} fades as it travels through free space. We assume that the devices are on the Euclidean plane and that the power of the transmitted signal decreases according to the inverse of the distance between the transmitter i and the receiver j, represented by $d(i,j)$ raised to an exponent that represents path loss, α. This particular signal propagation model has been called by some authors the geometric SINR model [7,11].

$$\frac{\frac{P_{Ti}}{d(i,j)^{\alpha}}}{N_0 + \sum_{\substack{k=1\\k\neq i}}^{\tau} \frac{P_{Tk}}{d(kj)^{\alpha}}} \geq \gamma \tag{1}$$

The two other properties refer to noise and interference between signals. Noise corresponds to spurious signals that cannot be avoided and are usually present in any communication. Antennas and even the receiving circuit can be sources of noise. Noise interferes with the transmissions. The SINR model represents all noise from different sources as a single constant N_0 called background noise.

Interference occurs when multiple transmissions take place simultaneously in the same area. Each device receives a composition of signals. Typically, a receiver is only responsible for decoding a single signal, which is the one with the highest power level, so that all others are considered interfering signals [12].

The so-called Down-to-Earth heuristic [5] was designed to improve spatial reuse: each device i only makes a transmission to the closest device j. Next, we describe how a device determines the power level of its transmissions. Consider that i is the only device making a transmission. In this case, the minimum power level P_{Ti}, required by device i to communicate successfully with j is shown as Eq. 2 below.

$$\frac{\frac{P_{Ti}}{d(i,j)^{\alpha}}}{N_0} = \gamma \quad \therefore \quad P_{Ti} = \gamma + \cdot N_0 \cdot d(i,j)^{\alpha} \tag{2}$$

However, in order to allow simultaneous transmissions, the transmission power level must be set so that the SINR condition at each receiver is above the minimum limit. We call this extra power the spare SINR level, γ_{spare}. Thus the transmission power P_{Ti} adopted by device i is such that the resulting SINR at the receiver j is $\gamma + \gamma_{spare}$, as shown in Eq. 3.

$$\frac{\frac{P_{Ti}}{d(i,j)^{\alpha}}}{N_0} = \gamma + \gamma_{spare} \quad \therefore \quad P_{Ti} = (\gamma + \gamma_{spare}) \cdot N_0 \cdot d(i,j)^{\alpha} \tag{3}$$

The spare power makes spatial reuse possible, under certain conditions. The amount of interference power P_{Φ} supported by the receiver j, when considering the reception of the signal transmitted by the device i, is given by the following equation, where P_{Rj} is the power level at the receiver j, given by $\frac{P_{Ti}}{d(i,j)^{\alpha}}$:

$$P_{\Phi} \leq \frac{P_{Rj}}{\gamma} - N_0 \tag{4}$$

A wireless channel is a shared communication medium that requires a protocol to allow multiple devices to communicate with each other. TDMA (Time Division Multiple Access) is a widely adopted strategy that schedules devices to transmit at specific time intervals, called time slots. The SINR model allows spatial reuse, *i.e.*, multiple transmissions can be scheduled to the same *slot*, thus improving the performance of the system as a whole. Figure 1 illustrates an example of an SINR schedule with spatial reuse.

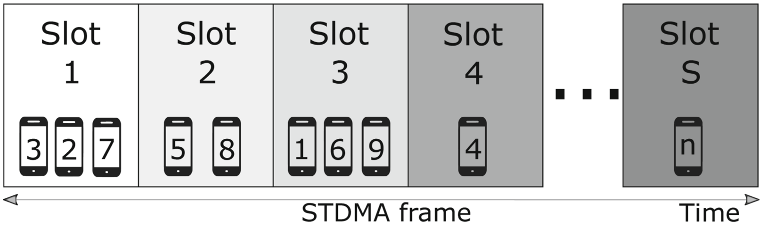

Fig. 1. An SINR scheduling example.

The most common approach to the SINR problem is to schedule links, not devices [15]. A link corresponds to a transmission from a source to a destination device. Thus, link $\ell = (i, j)$ represents a transmission, where i is the source and j is the destination. The length of the link is defined as the Euclidean distance $d(i, j)$ between the sender and the receiver. A scheduling algorithm assigns links to time slots, establishing an order for all the transmissions. The greater the number of simultaneous links in a given time slot, the greater the spatial reuse.

In the next section, we present a bioinspired strategy as a feasible solution to the Multi-Slot Scheduling Problem (MSSP), *i.e.*, the objective is to obtain a schedule with the minimum number of time slots. Another bioinspired approach has also been proposed recently [18], but in that work, the authors propose a genetic algorithm to solve the One-Slot Scheduling Problem (OSSP), with the objective of maximizing the number of transmissions on a time slot.

3 The Proposed Bioinspired Scheduling Strategy

In this section, we present GeTS[1] – the Genetic Transmission Scheduler, a genetic solution for the STDMA (TDMA model that allows spatial reuse [17]) scheduling problem. Genetic algorithms are stochastic heuristics based on Darwinian principles [10]. Those algorithms have been successfully used to solve a variety of problems in networking, such as resource allocation [14], fault diagnosis [4,16],

[1] Available at https://github.com/ViniGarcia/GeTS.

among many others. Genetic heuristics are well known to achieve optimal global solutions in acceptable computation time.

In our context, the genetic algorithm is used to find an SINR schedule, in which every link is assigned to a time slot. The source device has the opportunity to make a transmission in that time slot. The schedule has to assign a slot to each of the links. After the last time slot, the schedule is executed again and this goes on indefinitely. The main objective is to find a schedule with the smallest possible number of time slots. To reduce the size, it is necessary to maximize global spatial reuse, *i.e.* assign as many links as possible to communicate simultaneously. In other words: minimizing the schedule size implies maximizing spatial reuse across all time slots. Note that reducing schedule size has advantages not only in terms of raising the efficiency of the network as a whole but also in reducing the time it takes for a device to communicate.

A major goal of GeTS is to provide a configurable solution, in the sense that the user can tune a set of parameters that affect scheduling. Parameters are either related to the SINR model or to the genetic algorithm itself. The parameters related to the SINR model are path loss, background noise, interference limit, and maximum time slot size. In order to improve the chance that multiple simultaneous transmissions can be scheduled for the same time slots thus improving spatial reuse, the strategy employs the down-to-earth heuristics: each device only communicates with its closest device. Thus the positions of links in the Euclidean plane are known. The transmission power level is individually adjusted by each device according to the distance to the closest device, as described in Sect. 2. Every device is assumed to be able to detect the start of each time slot.

Genetic algorithms evaluate generations of individuals, each individual representing a possible solution to the problem at hand. Each individual is represented by a chromosome that carries information about the solution to the problem. The chromosome consists of multiple genes carrying alleles that represent solutions to specific parts of the problem. The set of individuals executing a genetic algorithm is called its population. In the case of GeTS, each individual carries a valid schedule. The chromosome corresponds to a schedule, *i.e.* a vector of time slots, each particular time slot (*i.e.*, a sub-vector) is a gene, and each device is scheduled into a time slot (gene) is an allele. Figure 2 illustrates the representation of the problem modeled with GeTS.

A genetic algorithm starts with an initial population, that must be generated beforehand, consisting of a predefined number of valid individuals (called the population size). A valid individual is a candidate for solving the problem. In the proposed solution, an initial population of schedules of the maximum size (number of links) is generated with unique and random genes, each of which corresponds to a time slot. Next, each gene is tested to check if all its assigned devices (alleles) can perform simultaneous transmissions. If that is the case, the gene is validated and becomes part of the chromosome of a given individual. If not, one of the devices is removed from the time slot and the validity of the corresponding gene is checked again. As mentioned above, in the worst case (no

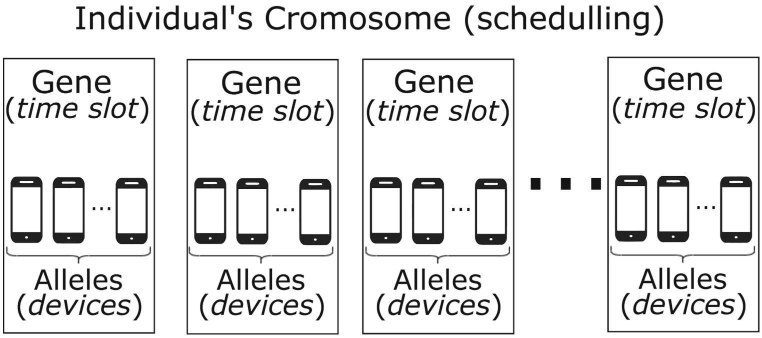

Fig. 2. GeTS: problem representation.

spatial reuse) a chromosome has the number of genes equal to the number of links in the network.

Genetic algorithms promote evolution by crossing and mutating individuals. Individuals present in a given generation are used to create new individuals for the next generation. A crossover between individuals occurs according to a given rate, which is the probability of two or more selected individuals producing new individuals for the next generation. The proposed solution uses a binary crossover mechanism in which two new individuals are always created from two individuals that already exist. The crossover strategy consists of mixing half of the genes of each individual, thus generating two new individuals: each with half of the alleles of the parents. As a given device (allele) cannot appear twice in the genes of an individual, occasionally it is necessary to replace some alleles of the resulting genes in the new chromosomes.

The selection of an individual for the crossover process is based on a binary tournament. Binary tournaments receive as input two individuals randomly selected from the current generation, the best of which is selected. The selection is based on an objective function: GeTS chooses the individual with the chromosome with the smallest number of genes. GeTS requires the execution of two binary tournaments to determine the pair of individuals to crossover.

Mutations are also employed by genetic algorithms to promote the evolution of the population in the following generations. In this process, some genes of specific individuals' chromosomes are modified according to some particular strategy. Mutations also occur according to a particular rate, similar to crossover. GeTS adopts the following mutation strategy: two genes are randomly chosen, merged, and checked to be valid, *i.e.* whether the devices (alleles) can make transmissions in the same time slot. If the mutation is valid, the resulting individual has a smaller chromosome – one gene less than the original one. If the mutation is invalid, it is discarded, and the original individual is returned intact.

As the proposed model relies both on external optimization (by rearranging chromosomes during crossover) and internal optimization (by integrating alleles during mutation), the crossover and mutation rates are typically high. This behavior is not usual in most genetic algorithms, which typically have a high crossover rate and a low mutation rate. However, the problem at hand – SINR scheduling – has properties that do require these particular rates to guarantee the proper exploration and exploitation of the search space. Thus, the optimization can both converge to local optima and approximate the global optimum.

Finally, the proposed genetic heuristic can be classified as elitist: only the top 10% of individuals are kept from one generation to the next. This process allows for safe explorations since it guarantees that the individuals of a generation with the best fitness are not lost in the next one.

4 Experimental Evaluation

This section describes the experiments executed to evaluate GeTS. The first experiment consists of a convergence test of the genetic algorithm. The second experiment evaluates the algorithm's ability to approximate the optimal (*i.e.*, minimum size) schedule.

The first experiment is the convergence test, which was executed to check the feasibility of the proposed genetic algorithm by determining whether it can evolve and eventually converge to a result after a certain number of generations (even if it is to a local optimum). For that convergence test, there is no predefined limit to the number of generations the genetic algorithm can create. To determine convergence, the following criterion was defined. In the convergence test, GeTS was employed to produce schedules for a highly dense network of 50 devices randomly placed on a $50\,\mathrm{m} \times 50\,\mathrm{m}$ area. The SINR parameters were set as follows (for all experiments): the path loss was set $\alpha = 4$; the SINR threshold $\gamma = 20$ decibels, while $\gamma_{spare} = 50$ decibels and the background noise $N_0 = -90\,\mathrm{dBm}$. The genetic algorithm was configured to operate with a population of 30 individuals, a maximum slot size of 15, and a crossover and mutation rate of 0.7. The algorithm stops evolving after it reaches a generation for which the mean of the objective function (scheduling size) is the same as that of the previous 100 generations.

The results are shown in Fig. 3, where each dot indicates the average size of schedules created in a given generation, and the error bars indicate the size of the worst (largest) and best (smallest) schedules of that generation. As can be seen in the graph, the first set of 100 generations produced schedules with an average size of 28.93 time slots, after further evolution with the application of crossover and mutation they reached an average size of 13.33 time slots in generation number 3400. By generation 3500 the algorithm had converged according to our criteria. The size variation of schedules produced across generations reflects the exploration of the search space by genetic heuristics. These results confirm that GeTS represents a feasible solution to the SINR scheduling problem.

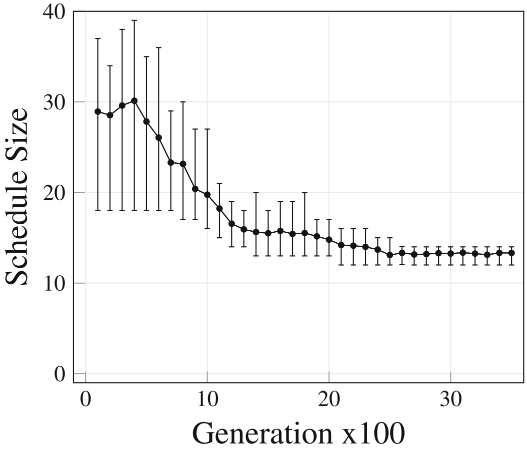

Fig. 3. Convergence test.

The next experiment was executed to evaluate the performance of the GeTS algorithm, in particular, its ability to produce schedules with sizes close to or equal to the optimum. For this reason, we compared GeTS with an optimal algorithm. Both algorithms were applied to find schedules for links determined by the down-to-earth heuristics, described in Sect. 2. The optimal algorithm is exhaustive: after checking all possible combinations to find the minimum sized schedule.

Figure 4 shows how close to the optimum GeTS gets, as well as the heuristic algorithm presented in [5]. That algorithm also computes the list of sets called δ that contains sets of devices that can transmit simultaneously in the same time slot. The algorithm models the problem as a graph $G_\delta = (V, E)$ in which the vertices are the δ sets, and there is an edge in E between every two vertices representing δ sets that have a device in common. Each device only needs to be in a single δ set (time slot). The final schedule is obtained by obtaining a Maximal Independent Set on G, which is done employing the efficient algorithm by Tsukiyama and others [21].

The scheduling strategies were compared for systems with 5 and 10 devices. The optimal algorithm could not be executed in a feasible time for larger system sizes. On the other hand, GeTS was executed for up to 50 devices in a few seconds. Areas of five different sizes were considered from 50m $\times$ 50 m to 200m $\times$ 200 m. Simulations were repeated 1,000 times, each for a different random distribution of the devices across the area. Figure 4 shows that GeTS surpasses the other heuristic algorithm and gets very close to the optimum, regardless of the number of devices considered, producing 90% or greater than that percentage of optimal schedules in all scenarios.

It is also possible to conclude from Fig. 4 that the percentage of optimal schedules decreases as the number of devices increases. To further investigate this fact, Fig. 5 shows the percentage of schedules with no spatial reuse. In other

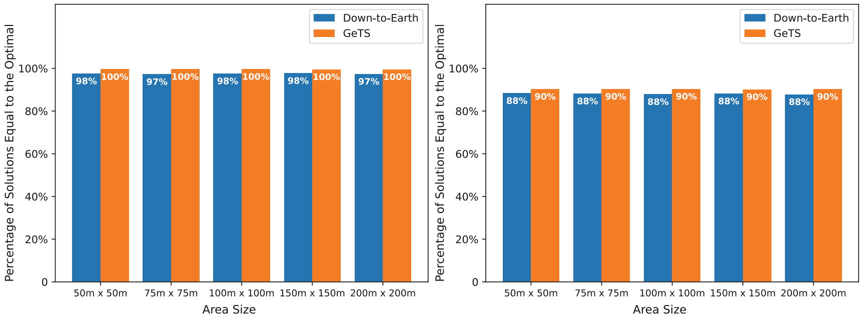

Fig. 4. Comparison with another heuristics: scenarios with 5 and 10 devices.

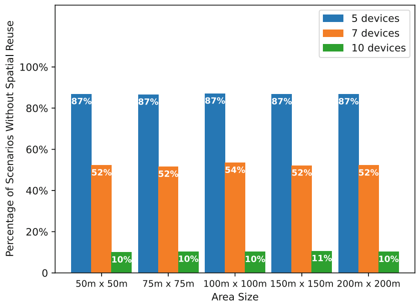

Fig. 5. Percentage of schedules with no spatial reuse.

words, the figure shows the percentage of schedules for which no pair of devices could be found to transmit simultaneously. In those cases, the schedule size is equal to the number of devices, since each device is assigned to transmit in a single time slot. From Fig. 5 it is also possible to conclude that the amount of spatial reuse potentially increases as the number of devices grows.

5 Conclusion

This work presented GeTS, a genetic solution to the scheduling problem in wireless networks under the SINR model. GeTS produces schedules for links generated by the so-called Down-to-Earth heuristics. The algorithm demanded high crossover and mutation rates and was evaluated in terms of the convergence and ability to produce good solutions to the problem. Simulations were executed for different scenarios, including comparisons with the optimal algorithm and another heuristic alternative. Results show that GeTS gets very close to the optimum and always surpasses the other heuristic.

References

1. Batta, M.S., et al.: A distributed TDMA scheduling algorithm for latency minimization in internet of things. In: International Conference on Electro Information Technology, pp. 108–113 (2019)
2. Blough, D.M., Resta, G., Santi, P.: Approximation algorithms for wireless link scheduling with SINR-based interference. Trans. Netw. **18**(6), 1701–1712 (2010)
3. Chowdhury, M.Z., et al.: 6G wireless communication systems: applications, requirements, technologies, challenges, and research directions. Open J. Commun. Soc. **1**, 957–975 (2020)
4. Duarte, E.P., Jr., et al.: Fault diagnosis of multiprocessor systems based on genetic and estimation of distribution algorithms: a performance evaluation. Int. J. Arti. Intell. Tools **19**(01), 1–18 (2010)
5. Engel, F., Duarte, E.: A down-to-earth scheduling strategy for dense SINR wireless networks. In: Latin-American Symposium on Dependable Computing, pp. 1–6 (2021)
6. Goussevskaia, O., et al.: Complexity in geometric SINR. In: International Symposium Mobile Ad Hoc Networking and Computing, pp. 100–109 (2007)
7. Goussevskaia, O., et al.: Efficiency of wireless networks: approximation algorithms for the physical interference model. Now Foundations and Trends (2010)
8. Guo, F., et al.: Enabling massive IoT toward 6G: a comprehensive survey. Internet Things J. **8**(15), 11891–11915 (2021)
9. Halldórsson, M.M., Wattenhofer, R.: Wireless network algorithmics. In: Steffen, B., Woeginger, G. (eds.) Computing and Software Science. LNCS, vol. 10000, pp. 141–160. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-91908-9_9
10. Kar, A.K.: Bio inspired computing - a review of algorithms and scope of applications. Expert Syst. Appl. **59**, 20–32 (2016)
11. Lam, N.X., et al.: A note on the complexity of minimum latency data aggregation scheduling with uniform power in physical interference model. Theoret. Comput. Sci. **569**, 70–73 (2015)
12. Lv, S., et al.: Scheduling under SINR model in ad hoc networks with successive interference cancellation. In: Global Telecommunications Conference, pp. 1–5 (2010)
13. Maheshwari, R., Jain, S., Das, S.R.: A measurement study of interference modeling and scheduling in low-power wireless networks. In: Conference on Embedded Network Sensor Systems, pp. 141–154 (2008)
14. Masdari, M., et al.: Bio-inspired virtual machine placement schemes in cloud computing environment: taxonomy, review, and future research directions. Clust. Comput. **23**(4), 2533–2563 (2020)
15. Moscibroda, T., Wattenhofer, R.: The complexity of connectivity in wireless networks. In: Computer and Communications Societies, pp. 1–13 (2006)
16. Nassu, B.T., et al.: A comparison of evolutionary algorithms for system-level diagnosis. In: The Genetic and Evolutionary Computation Conference, pp. 2053–2060 (2005)
17. Nelson, R., Kleinrock, L.: Spatial TDMA: a collision-free multihop channel access protocol. Trans. Commun. **33**(9), 934–944 (1985)
18. Saad, M., et al.: Wireless link scheduling via parallel genetic algorithm. Concurr. Comput. Pract. Exper. **34**(6), 6783 (2022)
19. Sgora, A., et al.: A survey of TDMA scheduling schemes in wireless multihop networks. ACM Comput. Surv. **47**(3), 1–39 (2015)

20. Son, D., Krishnamachari, B., Heidemann, J.: Experimental study of concurrent transmission in wireless sensor networks. In: Conference on Embedded Networked Sensor Systems, pp. 237–250 (2006)
21. Tsukiyama, S., et al.: A new algorithm for generating all the maximal independent sets. J. Comput. **6**(3), 505–517 (1977)