# A Genetic Scheduling Strategy with Spatial Reuse for Dense Wireless Networks

Vinicius Fulber-Garcia, Fábio Engel, and Elias P. Duarte Jr.

**Abstract** Novel networking technologies such as massive Internet-of-Things and 6G-and-beyond cellular networks are based on ultra-dense wireless communications. A wireless communication channel is a shared medium that demands access control, such as proper transmission scheduling. The SINR model can improve the performance of ultra-dense wireless networks by taking into consideration the effects of interference to allow multiple simultaneous transmissions in the same coverage area and using the same frequency band. However, scheduling in wireless networks under the SINR model is an NP-hard problem. This work presents a bioinspired solution based on a genetic heuristic to solve that problem. The proposed solution, called Genetic-based Transmission Scheduler (GeTS) produces a complete transmission schedule optimizing size, increasing the number of simultaneous transmissions (*i.e.*, spatial reuse) thus allowing devices to communicate as soon as possible. Simulation results are presented for GeTS, including a convergence test and comparisons with other alternatives. Results confirm the ability of the solution to produce near-optimal schedules.

## 1 Introduction

There is a noticeable trend in networking technologies towards ultra-density. Examples include cellular networks [8, 41], wireless sensor networks [7, 31] and massive Internet-of-Things [19, 25]. As a wireless channel is a shared medium [43], 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 transmit-

Vinicius Fulber-Garcia and Elias P. Duarte Jr.
Federal University of Paraná, Curitiba, Brazil e-mail: {vinicius, elias}@inf.ufpr.br

Fábio Engel
Technological Federal University of Paraná, Toledo, Paraná, Brazil e-mail: fabioe@utfpr.edu.br

ting devices in space or time [39]. 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. Efficiently scheduling communications over wireless links is essential to guarantee the latency and throughput requirements of ultra-dense 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 [20, 21]. This model allows multiple simultaneous transmissions that do interfere with one another, as long as the power level of an interfering signal transmitted from one source is low enough not to prevent the proper receptions of signals transmitted from the other sources. The so-called "spatial reuse" improves the efficiency of ultra-dense networks by allowing multiple signals to be transmitted simultaneously by devices that are within the coverage areas of each other and using the same frequency spectrum. Spatial reuse is only possible if mutual interference does not prevent the correct reception of the multiple simultaneous transmissions by their respective destinations. The greater the number of simultaneous transmissions, the greater the spatial reuse.

Thus, in the SINR model, it is necessary to schedule transmissions taking advantage of spatial reuse as much as possible. Time is divided into slots, and devices are scheduled to specific slots along with other devices with which mutual interference does not damage signal reception. Informally, the objective is to obtain the minimum schedule that allows all devices to communicate. This objective also implies having the largest feasible subsets of transmissions that can be scheduled simultaneously across all slots. Such a schedule that has (ideally) the minimum number of time slots needed for all devices to communicate, also reduces the time each device has to wait until it can communicate. The scheduling problem in SINR wireless networks has been proven to be NP-hard [17]. For this reason, much of the work in this area explores approximation algorithms [32, 18, 2, 4, 21, 5, 1]. Although those algorithms are important from the theoretical point of view, they have little application in practice [36]. 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) [13]. The proposed strategy consists of a scheduling algorithm for ultra-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 [11]. Genetic heuristics are inspired by Darwin's principle of natural selection [24]. They are probabilistic algorithms that have a parallel and adaptive search engine based on the principle of the survival of the fittest. This search technique does not require any prior knowledge. Therefore, they can solve complex problems and obtain solutions close to the optimum in a reasonable execution time.

GeTS introduces a new evolutionary model for solving the SINR scheduling problem. A population of individuals (each representing a candidate schedule to solve the problem) 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. Sections 2 and 3 describe the SINR model and the scheduling problem under the SINR model, respectively. Section 4 introduces GeTS, the Genetic-based Transmission Scheduler. Section 5 shows the results of two experiments conducted to evaluate GeTS. Finally, conclusions are presented in Section 6.

## 2 The SINR Model

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 [40, 29] 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 $\gamma$ is employed to determine whether a signal sent by a device $i$ can be correctly received by a device $j$, and is described in this section. The SINR threshold is computed based on three properties of wireless communications. The first is a property related to signal propagation, the path loss, 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, $\alpha$. This particular signal propagation model has been called by some authors the geometric SINR model [18, 23].

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

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 in the same frequency spectrum take place simultaneously. A device in the coverage area of multiple transmitters 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 [28]. Note that technologies such as CDMA (*Code Division Multiple Access*) and MIMO (*Multiple Input Multiple Output*) allow a single device to simultaneously receive signals from multiple transmitters, but they are not employed in this work.

Next, we describe how a device determines the power level of its transmissions. We assume each device $i$ only makes a transmission to the closest device $j$. 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 equation 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, $\gamma_{spare}$. 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 equation 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 $P_{\Phi}$ 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 $\frac{P_{Ti}}{d(i,j)^{\alpha}}$:

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

It is necessary to properly schedule transmissions in order to enforce that they are successful and also take advantage of spatial reuse and improve the performance of the system as a whole. The next section presents an overview of the problem of scheduling communications under the SINR model.

## 3 Scheduling in the SINR Model

Regardless of the particularities of specific models, all wireless networks rely on the same basic technology: data is transmitted with radio waves over the air, a shared communication medium. Any shared communication medium requires a protocol to manage access, which coordinates the way the multiple devices communicate with each other. Those protocols fall into two broad types: contention-based and contention-free. Devices running protocols of the first type employ a strategy based on tries, which can either succeed or fail, for instance, due to collisions. On the other hand, contention-free protocols try to prevent collisions by properly scheduling the transmissions. TDMA (Time Division Multiple Access) is a widely adopted contention-free strategy [12], described next.

TDMA allows multiple devices to share the same frequency channel by scheduling the transmissions on different time intervals, called time slots. Data is transmitted in packets, each of which requires a single time slot to reach the destination. A TDMA frame consists of a number of time slots, that number is called the frame length. Recall that 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. The so-called Spatial-TDMA, or simply STDMA, is a TDMA model that allows spatial reuse [35]. Figure 1 illustrates an example of an STDMA schedule where $n$ devices are scheduled over a frame of $S$ time slots, each device is allowed to transmit a single packet per frame.

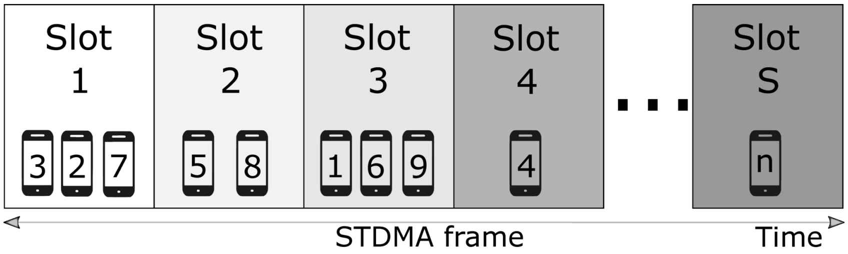

Fig. 1: A STDMA scheduling example.

An STDMA schedule must ensure that transmissions – including those that are simultaneous – can be correctly received. This problem has been called the “link scheduling problem” [32], which has the purpose to assign transmissions from a specific source to a specific destination to the time slots. In this context, 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. For this reason, the amount of spatial reuse achieved is often employed as an evaluation metric for scheduling algorithms [16]. Note that the resulting system is multi-hop, i.e. it is not fully connected – there are pairs of devices that are not connected by direct communication links. Thus in order to allow an arbitrary device to send a message to another arbitrary device, another step that consists of routing is required to allow an arbitrary device to communicate with another arbitrary device.

The scheduling problem has been considered under three main aspects [21]. The first aspect is gain, which corresponds to how much power a signal loses along the path until reaches some destination. The second aspect is power control, which corresponds to the strategy adopted to determine the power level to transmit a signal. Finally, the third aspect is the scheduling objective - the primary purpose to be achieved by the algorithm. The scheduling problem for the SINR model can be found in three main versions in the literature: *One-Slot Scheduling Problem* - OSSP, *Weighted One-Slot Scheduling Problem* - WOSSP, and *Multi-Scheduling Problem* slot - MSSP. OSSP takes as input a set of links, and the objective is to maximize the number of links that are scheduled simultaneously in a single slot such that all transmissions are correctly received. WOSSP is a version of OSSP in which each input link is assigned a weight. The weight can correspond for instance to the relative priorities of the different transmissions which the links represent. The objective of the problem is to select a subset of the weighted links such that the total weight or associated value is maximized and the SINR value on each receiver is at least equal to the SINR threshold. MSSP aims at scheduling all links in as few slots as possible while ensuring that all messages are correctly delivered.

The seminal work by Moscibroda and Wattenhofer [32] [3, 9] proves that the complexity of the number of time slots to schedule a set of links grows polylogarithmically with the number of devices. They also propose an algorithm that generates schedules of $O(log^4 n)$ slots for $n$ devices strongly connected and using a non-linear power assignment scheme. In [33] the same authors prove that the number of time slots required to schedule all links of an arbitrary topology is proportional to the squared logarithm of the number of devices times a predefined static interference measure.

As mentioned before, the scheduling problem for the SINR model has been proved to be NP-hard [17]. In the same work, the authors propose an approximation algorithm that starts by creating partitions of the input set of links that are scheduled separately. The authors show that the schedules produced by the algorithm have an approximation ratio that is proportional to the link length diversity, which can be informally explained as the ratio between the longest and shortest links. Chafekar and others [6] also consider end-to-end routing besides solving the link scheduling problem. The model adopted by [17] is a simplified version of the exact SINR, which is considered by Blough and others [4]. There have been other related efforts, mostly theoretical in nature [21].

A bioinspired approach has also been proposed recently [37]. First, the authors present a new genetic algorithm (GA) to select the maximum subset of the links among nodes of a wireless network under the SINR model, which may be scheduled to make transmissions at the same time slots. The genetic algorithm is based on a constraint-handling mechanism to find optimal link schedules. The authors give analytical proofs that the algorithm is guaranteed to produce link schedules that are optimal. Another genetic algorithm is also presented, this one an algorithm that can be executed in a parallel way. The parallel genetic algorithm relies on both data decomposition and exploratory decomposition and is shown empirically to perform (in terms of running time) significantly better than the serial algorithm, with significant speed-ups. The algorithm was implemented using OpenMP and is shown (empirically) to scale well as the size of the network grows. The implementation was based on multi-threads in the master-slave model, and the speed-ups are up to 10.4, confirming significant reductions in the execution times. Note that this work solves the OSSP problem with the objective of maximizing the number of transmissions in a (single) time slot. Thus it is not comparable with the solution proposed in the present paper. In the next section, we present a bioinspired strategy as a feasible solution to the MSSP problem, *i.e.* the objective is to obtain a schedule with the minimum number of time slots.

## 4 The Proposed Bioinspired Scheduling Strategy

In this section, we present GeTS – the Genetic Transmission Scheduler, a bioinspired solution for the STDMA scheduling problem[1]. Bioinspired computing draws inspiration from natural phenomena, such as the evolution of species and teamwork among individuals of particular species, to solve computational problems. Genetic algorithms are stochastic heuristics based on Darwinian principles [22, 14, 15]. Those algorithms have been successfully used to solve optimization problems in a variety of fields, such as network resource allocation [44, 30], fault diagnosis [10, 34], security [38, 26], among many other areas. Genetic heuristics are well known to achieve optimal global solutions in acceptable computation time.

In our context, the genetic algorithm is used to find a complete STDMA schedule, in which every device is assigned to a time slot. Each device has the opportunity to make a transmission in its assigned time slot. 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, thus being classified as an MSSP scheduling problem. The trivial solution to the complete schedule problem is to assign a single device to each time slot. In that case, the schedule size (total number of time slots) is equal to the number of devices. In order to reduce the size, it is necessary to maximize global spatial reuse, *i.e.* assign as many devices 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 limits, 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 heuristic: each device only communicates with its closest

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

device. Thus the problem becomes that of scheduling links connecting two devices, the positions of which 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 Section 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 has 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.

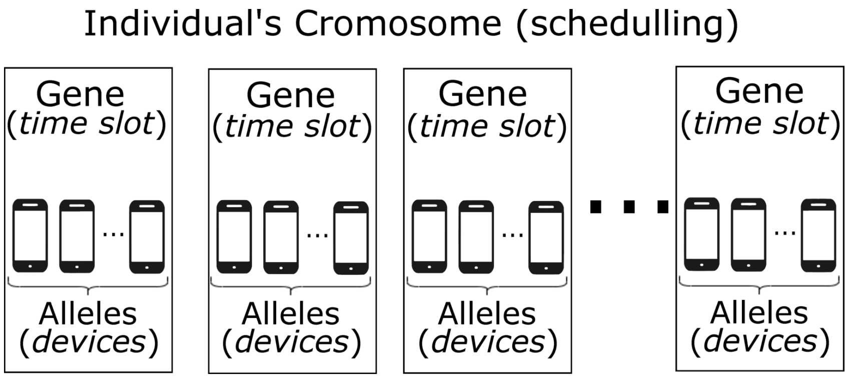

Fig. 2: GeTS: problem representation.

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 devices) 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 that has no spatial reuse, a chromosome has the number of genes equal to the number of devices in the network: a single device is scheduled to transmit in each time slot. This worst-case individual is always accepted as valid.

Evolution is accomplished in genetic algorithms by crossing and mutating individuals. Individuals present in a given generation are used to create new individuals that will belong to 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 employed 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, which in the case of GeTS leads to the choice of 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 in order to promote the evolution of the population in the following generations. In this process, the 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 – STDMA scheduling – has properties that do require this particular configuration of the rates to guarantee 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 individuals with the best fitness of a generation are not lost in the next one.

## 5 Experimental Evaluation

This section describes the experiments executed to evaluate the proposed bioinspired scheduler. 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 third experiment compares the algorithm to a greedy heuristic presented for the Down-to-Earth heuristics. Finally, a fourth experiment is presented comparing GeTS with the Stochastic k-Greedy algorithm.

## 5.1 Convergence Test

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. In order 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 50mX50m 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 $N_0 = -90$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 Figure 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) schedule 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 variation of the average size of schedules over the generations, as it increases and decreases reflects how the exploration of the search space was pursued by the genetic heuristic. The convergence and results were considered satisfactory.

## 5.2 Evaluation: GeTS Vs. The Optimal Algorithm

The second experiment was executed to evaluate the performance of the GeTS algorithm, in particular, its ability to produce a schedule with a size that is close to or equal to the optimum. For this reason, we compared GeTS with an optimal algorithm based on the down-to-earth heuristics [11], i.e. the only links considered are those from each device to its closest device. The algorithm computes the optimal, minimum-sized schedules. The power level of the transmissions adopted by each device is as described in Section 2, being set to a level that guarantees that the resulting SINR at the receiver is the minimum necessary to ensure that the transmission is

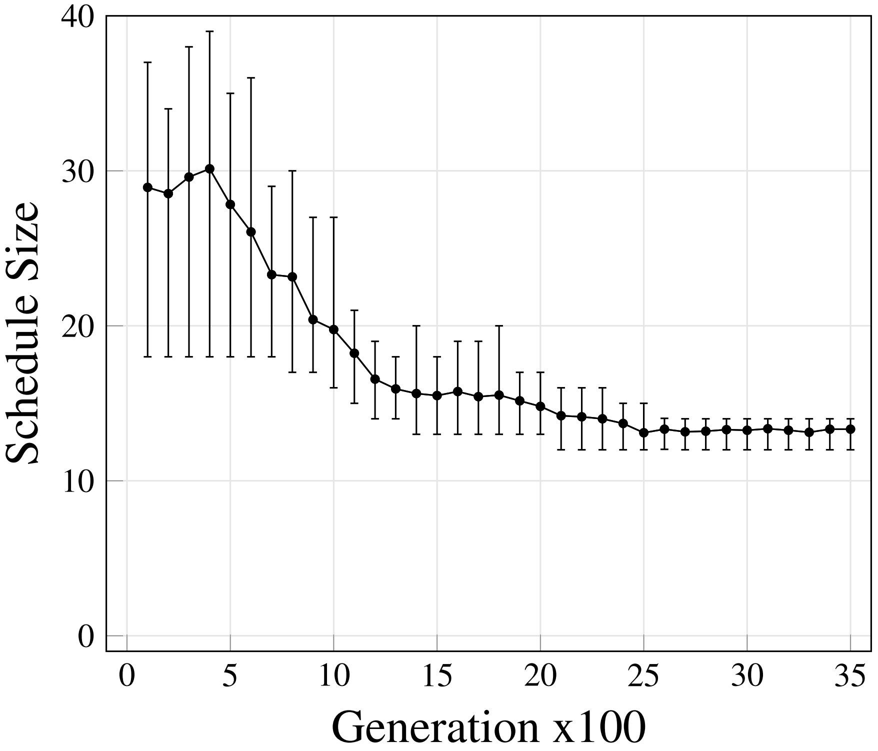

Fig. 3: Convergence test.

successful plus an extra margin that makes space reuse possible ($\gamma + \gamma_{spare}$). For each link $\ell = (i, j)$, the set of devices that can transmit simultaneously with $i$ without preventing $j$ from receiving $i$'s signal is computed. This computation is based on the SINR inequality (Equation 1). Let the sum of the interference power that allows the correct reception of the signal by $j$ be at most $P_\Phi$ for link $\ell = (i, j)$. Thus the acceptable interference power supported by $j$ must satisfy inequality $P_\Phi \leq \frac{\frac{P_{Ti}}{d(i,j)^\alpha}}{\gamma} - N_0$. For a transmitting device $k$ to transmit simultaneously with $\ell = (i, j)$, the power of the signal received by $j$ from $k$ must be less than or equal to $P_\Phi$.

The optimal algorithm employs the method described in the previous paragraph to determine the sets of transmissions (links) that can be scheduled to transmit simultaneously with each link $\ell = (i, j)$. After checking all possible combinations, the optimal algorithm generates a list of sets called $\delta$ that contains sets of devices that can transmit simultaneously in the same time slot. All combinations are checked to find the minimum-sized schedule.

### 5.3 Evaluation: GeTS Vs. Down-To-Earth Heuristic Algorithm

GeTS was also compared with the down-to-earth heuristic algorithm presented in [11]. The algorithm also computes the list of sets called $\delta$ that contains sets of devices that can transmit simultaneously in the same time slot. That algorithm models the problem as a graph $G_\delta = (V, E)$ in which the vertices are the $\delta$ sets, and there is an edge in $E$ between every two vertices representing $\delta$ sets that have a device in common. Each device only needs to be in a single $\delta$ 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 [42].

The scheduling strategies were compared for systems with 5, 7, and 10 devices. The reason we stopped at this system size is that the optimal algorithm took a very long time to execute for larger numbers of devices. Areas of five different sizes were considered: 50m X 50m, 75m X 75m, 100m X 100m, 150m X 150m, and 200m X 200m. Simulations were repeated 1,000 times, each for a different random distribution of the devices across the area. Comparisons of the GeTS and Down-to-Earth heuristic algorithm with the optimal algorithm for the different area sizes are shown in Figures 4 (5 devices), 5 (7 devices), and 6 (10 devices). The y axis shows the percentage of cases for which the algorithm in question produces exactly the same solution as the optimal algorithm.

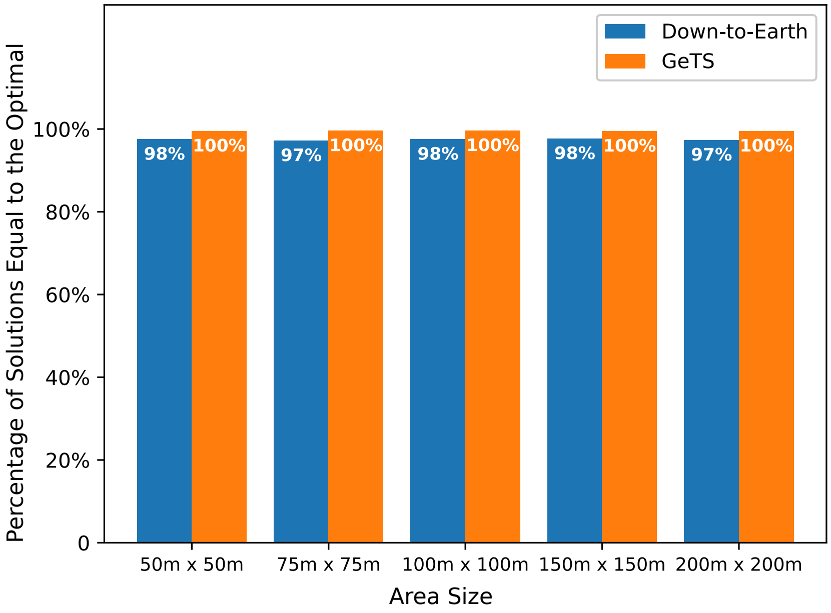

Fig. 4: Comparison with the optimal algorithm: scenarios with 5 devices.

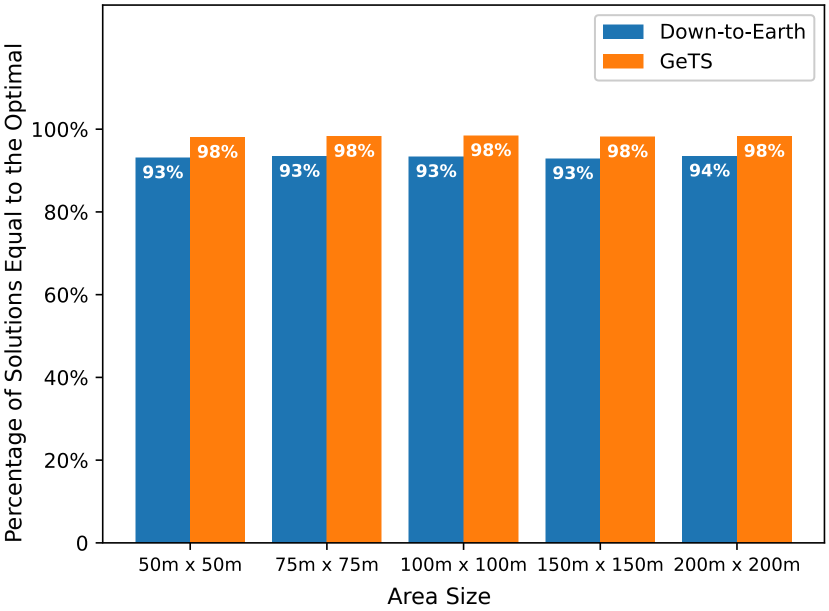

Fig. 5: Comparison with the optimal algorithm: scenarios with 7 devices.

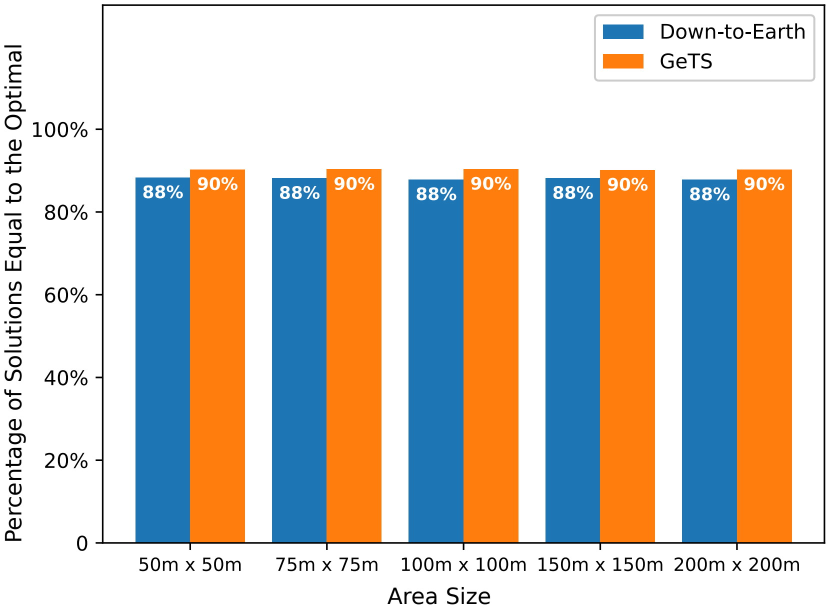

Fig. 6: Comparison with the optimal algorithm: scenarios with 10 devices.

Results show that GeTS 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. This means that in most cases, the algorithm can efficiently find the best solution from the vast space of possible solutions. It can be observed that the GeTS algorithm is more efficient than the *Down-to-Earth* algorithm in terms of obtaining the smallest schedule.

Figures 4, 5, and 6 also show that the percentage of optimal schedules decreases as the number of devices increases. To further investigate this fact, Figure 7 shows the percentage of schedules with no spatial reuse. In other 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 Figure 7 it is also possible to conclude that spatial reuse potentially increases as the number of devices grows.

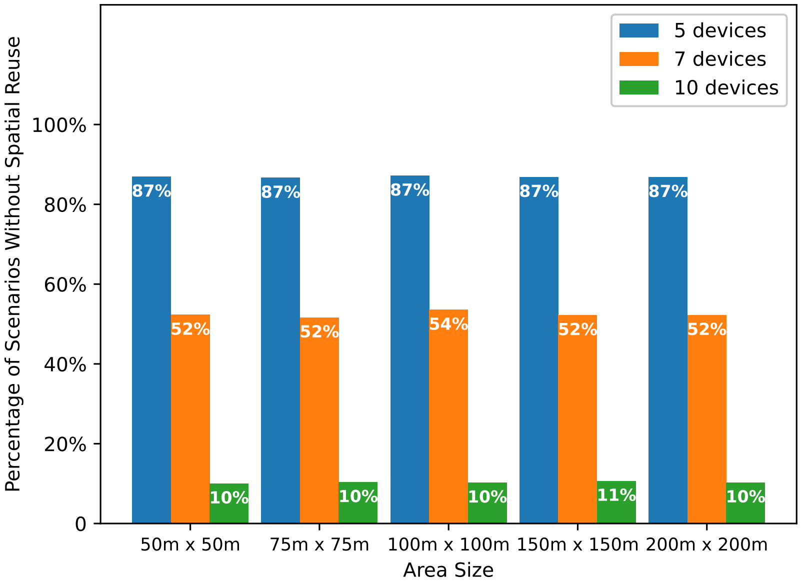

Fig. 7: Percentage of schedules with no spatial reuse.

On the other hand, the growth of the number of devices implies the growth of the number of possible combinations of transmitting devices that could be scheduled for the same time slot. For the optimal algorithm, the number of alternatives to check grows very fast as the algorithm is exponential. Notwithstanding, Figure 8 confirms that GeTS is able to obtain schedules for systems with from 5 to 50 devices in short time frames. The figure shows the results of the mean execution times of 10,000 GeTS executions for different placements of the devices. Note that GeTS

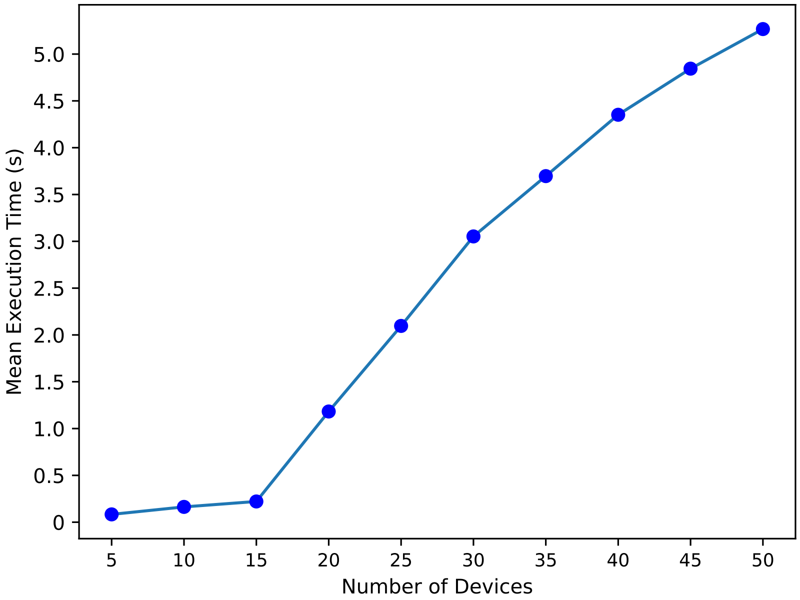

Fig. 8: Mean of the execution time of the GeTS algorithm.

was implemented in Python, even better results could certainly be obtained for a compiled language.

### 5.4 Evaluation: GeTS Vs. SK-Greedy

Finally, we compare the results of GeTS with those of the *Stochastic K-Greedy* (SK-Greedy) greedy heuristic, which was adapted to generate STDMA SINR schedules.

Sk-Greedy is a simple and efficient heuristic that schedules links produced by the Down-To-Earth strategy problem in wireless networks under the SINR model. The heuristic selects $k$ random candidates among all possible links, thus evaluating only those candidates to find a local solution to the problem instead of going through every possible link in each step. $k$ is an input parameter defined by the user. In the next phase, either the best or the first viable candidate from the set of $k$ candidates is selected as the solution. The global problem state is then updated, and the process is repeated until Sk-Greedy finds a complete solution, *i.e.*, a schedule in which every device is assigned to a time slot.

Sk-greedy is a stochastic heuristic. As Sk-Greedy selects random candidates at each step of the problem, the global search space is explored efficiently. Furthermore, it also exploits unexpected, specific portions of the search space.

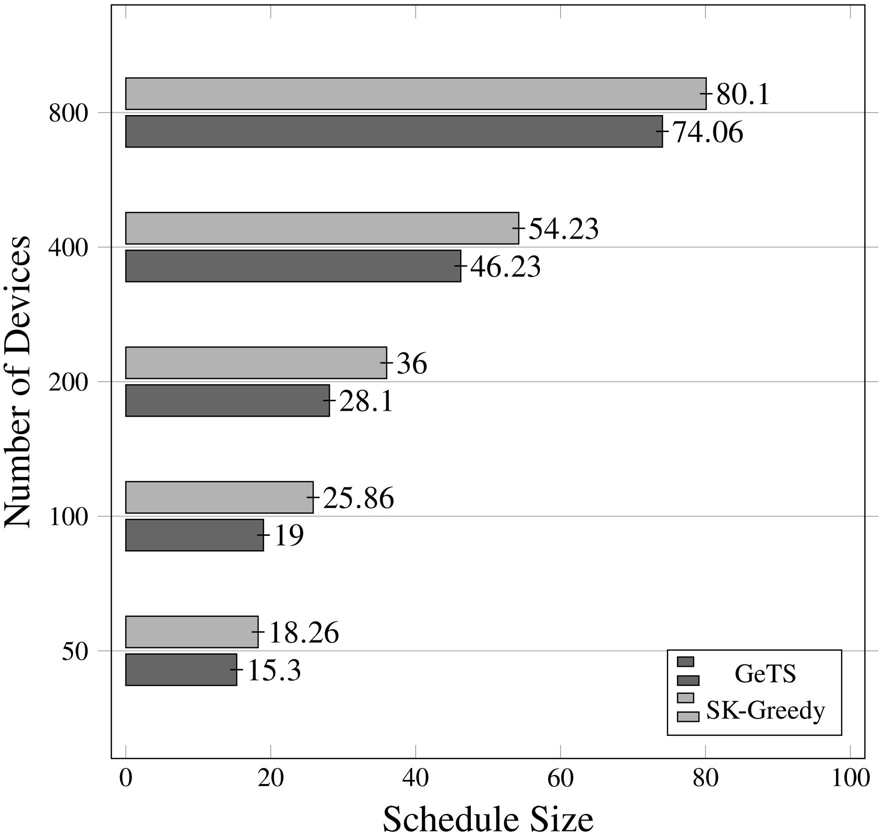

Fig. 9: Comparison between GeTS and SK-Greedy (area size = 400m x400m)

To increase the probability that a global optimum solution is found, the SINR scheduler can run Sk-Greedy multiple times, thus giving rise to a set of diverse but complete alternatives for the scheduling problem. After computing that diverse set of solutions, they are compared so that it is possible to select the best schedule, *i.e.*, the one with the highest amount of spatial reuse, which corresponds to the least number of time slots. That schedule is produced as the final output. To compare the multiple solutions, it is actually possible to employ the evaluation result values directly, in the case of mono-objective optimization, as the problem at hand. If multiple criteria were used (multi-objective optimization), it would be possible to employ strategies such as Pareto frontiers [27].

We call “Sk-Greedy Scheduler” the proposed scheduling algorithm based on the Sk-Greedy heuristic. The Sk-Greedy Scheduler receives as input the position of all devices on the Euclidean plane ($x$ and $y$ coordinates). The scheduler consists of the following three phases: (i) in the first phase, the communication graph is built given the locations of all devices and the rules for the existence of a link between two devices defined by the Down-To-Earth heuristics; (ii) as the Sk-Greedy Scheduler is a link scheduler, in the second phase of the algorithm, links are selected for allocation to each time slot, creating candidate schedules; (iii) the candidate schedules are evaluated in the final phase, and a decision is taken. The Sk-Greedy Scheduler produces slots iteratively, one by one until it finally has a complete schedule candidate (second step of the algorithm described above). The strategy to generate each is described next. Initially, there are no links assigned to the slot, and the algorithm selects a random link in the set of links available to schedule. The Sk-Greedy heuristic then proceeds to select $k - 1$ more links to be assigned to the slot.

Overall the process consists of three steps: (1) the heuristic selects $k$ available links randomly; (2) for each selected set of links, there is a verification of whether the transmission corresponding to a link can be simultaneous with the other links that have already been allocated to the slot; (3) the heuristic adopts the first-viable approach, *i.e.*, it simply selects the first link which can be simultaneous with the others already in the slot. Now, all the other links become available again (were selected but not used), and the algorithm goes back to step (1). The algorithm stops after none of the $k$ randomly selected available links can be assigned to the slot. The slot is then closed, and included in the current candidate schedule. The algorithm proceeds to the next slot, and the whole process is then repeated. The criterion employed by the algorithm to stop trying to generate slots is an empty set of available links, *i.e.*, all links are assigned to some slot, and there is a complete schedule candidate.

Finally, the Sk-Greedy scheduler reaches the third and final step in which all scheduled candidates are evaluated. As mentioned above, it suffices to employ a mono-objective criterion: the selection of the candidate with the highest spatial reuse, *i.e.*, that has the minimum number of times slots among all candidates. The mono-objective function employed as the criterion to select the best schedule among all candidates is shown in Equation 5. In case of a tie, *i.e.*, multiple candidates have the minimum number of time slots, and the algorithm returns all those candidates as solutions to the problem.

$$min \sum_{Slots}^{Scheduling} 1 \tag{5}$$

Note that the algorithm always returns some schedule that respects the constraints established by the SINR model. The best possible schedule would be one that consists of a single slot in which all devices transmit simultaneously, actually an impractical scenario. In the worst case, the scheduler will produce as output a schedule in which the number of slots is exactly the number of links, *i.e.*, no spatial reuse.

We verified through empirical tests that setting $K$=5 provided a good cost-benefit ratio between the results and the execution time of the SK-Greedy algorithm, and this configuration was adopted for the whole experiment. The experiment consisted of simulating an increasing number of devices in a 400m x 400m area: 50, 100, 200, 400, and 800 devices. We ran these simulations 30 times for both algorithms (GeTS and SK-Greedy) and checked the results produced after 20 seconds of execution time.

The results are shown in Figure 9, including the mean and standard deviation of the smallest schedule sizes returned by each of the 30 executions. It is possible to see that the sizes of the schedules generated by the two algorithms grow as the number of devices increases. Note that the search spaces also increase with the number of devices, and the same criterion to stop the algorithms was employed in all scenarios. Thus both algorithms performed a smaller number of generations/rounds and took longer to generate a schedule as the number of devices increased.

It is possible to see however that GeTS generates significantly smaller (better) schedules than those generated by the SK-Greedy heuristic in all scenarios. Although it produces competitive results, SK-Greedy is heavily dependent on a completely random factor: the selection of devices to be tested in an iteration. Thus, SK-Greedy disregards information from previous schedules when generating new schedules (it focuses on exploration). GeTS, on the other hand, benefits from the crossover strategy to merge and adapt previously evaluated schedules to improve them for future generations. This strategy, together with mutation, enables a balance between exploration and exploitation, which facilitates the generation of better schedules and the eventual convergence of the algorithm.

## 6 Conclusion

This work presented GeTS, a bioinspired solution to the scheduling problem in wireless networks under the SINR model. The trend towards ultra-dense wireless networks connecting very large numbers of devices raises concerns about mutual interference, which can represent a formidable obstacle to achieving the required performance levels. In this context, the SINR model represents a solution by allowing spatial reuse, i.e. multiple simultaneous transmissions, by taking into account the effects of cumulative interference on signal reception and the effects of path loss on the transmitted signal power, as well as background noise. However, it is necessary to schedule transmissions, so that the mutual interference of devices that communicate simultaneously does not damage signal reception. The objective is to obtain the minimum schedule that consists of the minimum number of time slots needed for all devices to communicate, also reducing the time it takes for a device to communicate. The scheduling problem in SINR wireless networks has been proven to be NP-hard [17].

The proposed bioinspired solution solves the time scheduling problem for this model in a feasible time. GeTS consists of a genetic algorithm that assumes the so-called down-to-earth heuristics, according to which each device only communicates with its closest device. The algorithm demanded high crossover and mutation rates and was evaluated in terms of convergence and ability to produce good solutions to the problem. Simulations were executed for different scenarios, including ultra-dense networks with as many as 50 and 800 devices per 50mX50m and 400mX400m areas, respectively. The solution was also compared with other alternatives, and results show that GeTS gets very close to the optimum and always produced results that are better than those of two other heuristic algorithms.

## References

1. Al-dhelaan, F., Wan, P.J., Yuan, H.: A new paradigm for shortest link scheduling in wireless networks: Theory and applications. In: Wireless Algorithms, Systems, and Applications, pp. 24–36. China (2016)
2. Batta, M.S., Harous, S., Louail, L., Aliouat, Z.: A distributed tdma scheduling algorithm for latency minimization in internet of things. In: International Conference on Electro Information Technology, pp. 108–113 (2019)
3. Belke, L., Kesselheim, T., Koster, A.M., Vöcking, B.: Comparative study of approximation algorithms and heuristics for sinr scheduling with power control. Theoretical Computer Science **553**, 64–73 (2014)
4. Blough, D.M., Resta, G., Santi, P.: Approximation algorithms for wireless link scheduling with sinr-based interference. Transactions on Networking **18**(6), 1701–1712 (2010)
5. Brar, G., Blough, D., Santi, P.: Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks. In: Annual International Conference on Mobile Computing and Networking, p. 2–13 (2006)
6. Chafekar, D., Kumar, A., Marathe, M., Parthasarathy, S., Srinivasan, A.: Cross-layer latency minimization in wireless networks with sinr constraints. In: International Symposium on Mobile Ad Hoc Networking and Computing, pp. 110–119 (2007)
7. Chang, Y., Yuan, X., Li, B., Niyato, D., Al-Dhahir, N.: A joint unsupervised learning and genetic algorithm approach for topology control in energy-efficient ultra-dense wireless sensor networks. Communications Letters **22**(11), 2370–2373 (2018)
8. Chowdhury, M.Z., Shahjalal, M., Ahmed, S., Jang, Y.M.: 6g wireless communication systems: Applications, requirements, technologies, challenges, and research directions. Open Journal of the Communications Society **1**, 957–975 (2020)
9. Dams, J., Hoefer, M., Kesselheim, T.: Scheduling in wireless networks with rayleigh-fading interference. Transactions on Mobile Computing **14**(7), 1503–1514 (2015)
10. Duarte Jr., E.P., Pozo, A., Nassu, B.: Fault diagnosis of multiprocessor systems based on genetic and estimation of distribution algorithms: a performance evaluation. International Journal on Artificial Intelligence Tools **19**(01), 1–18 (2010)
11. Engel, F., Duarte Jr, E.P.: A down-to-earth scheduling strategy for dense sinr wireless networks. In: Latin-American Symposium on Dependable Computing, pp. 1–6 (2021)
12. Ergen, S.C., Varaiya, P.: Tdma scheduling algorithms for wireless sensor networks. Wireless Networks **16**(4), 985–997 (2010)
13. Fulber-Garcia, V., Camargo, F.E.d., Duarte Jr., E.P.: A bioinspired scheduling strategy for dense wireless networks under the sinr model. Intelligent Systems Design and Applications (2022)
14. Fulber-Garcia, V., Camargo, F.E.d., Duarte Jr., E.P.: Sk-greedy: A heuristic scheduling algorithm for wireless networks under the sinr model. In: Latin-American Symposium on Dependable Computing - Workshop on Security, Privacy and Reliability on Wireless Sensing Networks, p. 143–148 (2022)
15. Fulber-Garcia, V., Luizelli, M.C., Santos, C.R.P.d., Spinosa, E.J., Duarte Jr., E.P.: Intelligent mapping of virtualized services on multi-domain networks. In: Intelligent Systems Design and Applications (2022)
16. Gore, A.D., Karandikar, A., Jagabathula, S.: On high spatial reuse link scheduling in stdma wireless ad hoc networks. In: Global Telecommunications Conference, pp. 736–741 (2007)
17. Goussevskaia, O., Oswald, Y.A., Wattenhofer, R.: Complexity in geometric sinr. In: International Symposium Mobile Ad Hoc Networking and Computing, pp. 100–109 (2007)

18. Goussevskaia, O., Pignolet, Y.A., Wattenhofer, R., et al.: Efficiency of Wireless Networks: Approximation Algorithms for the Physical Interference Model. Now Foundations and Trends (2010)
19. Guo, F., Yu, F.R., Zhang, H., Li, X., Ji, H., Leung, V.: Enabling massive iot toward 6g: A comprehensive survey. Internet of Things Journal **8**(15), 11891–11915 (2021)
20. Halldórsson, M., Mitra, P.: Wireless capacity with oblivious power in general metrics. In: Symposium on Discrete Algorithms, pp. 1538–1548. SIAM (2011)
21. Halldórsson, M.M., Wattenhofer, R.: Wireless Network Algorithmics, pp. 141–160. Springer International Publishing (2019)
22. Kar, A.K.: Bio inspired computing – a review of algorithms and scope of applications. Expert Systems with Applications **59**, 20–32 (2016)
23. Lam, N., Tran, T., An, M.K., Huynh, D.: A note on the complexity of minimum latency data aggregation scheduling with uniform power in physical interference model. Theoretical Computer Science **569**, 70–73 (2015)
24. Lambora, A., Gupta, K., Chopra, K.: Genetic algorithm - a literature review. In: International Conference on Machine Learning, Big Data, Cloud and Parallel Computing, pp. 380–384 (2019)
25. Lee, G., Youn, J.: Group-based transmission scheduling scheme for building lora-based massive iot. In: International Conference on Artificial Intelligence in Information and Communication, pp. 583–586 (2020)
26. de Lima, S., Silva, H., Luz, J., Lima, H., Silva, S., de Andrade, A., da Silva, A.: Artificial intelligence-based antivirus in order to detect malware preventively. Progress in Artificial Intelligence **10**(1), 1–22 (2021)
27. Lotov, A.V., Miettinen, K.: Visualizing the pareto frontier. In: Multiobjective optimization, pp. 213–243. Springer (2008)
28. Lv, S., Wang, X., Zhou, X.: Scheduling under sinr model in ad hoc networks with successive interference cancellation. In: Global Telecommunications Conf., pp. 1–5 (2010)
29. Maheshwari, R., Jain, S., Das, S.: A measurement study of interference modeling and scheduling in low-power wireless networks. In: Conference on Embedded Network Sensor Systems, pp. 141–154 (2008)
30. Masdari, M., Gharehpasha, S., Ghobaei-Arani, M., Ghasemi, V.: Bio-inspired virtual machine placement schemes in cloud computing environment: taxonomy, review, and future research directions. Cluster Computing **23**(4), 2533–2563 (2020)
31. Moraes, R., dos Reis, W., Rocha, H., Coura, D.: Power-efficient and interference-free link scheduling algorithms for connected wireless sensor networks. Wireless Networks **26**(5), 3099–3118 (2020)
32. Moscibroda, T., Wattenhofer, R.: The complexity of connectivity in wireless networks. In: Computer and Communications Societies, pp. 1–13 (2006)
33. Moscibroda, T., Wattenhofer, R., Zollinger, A.: Topology control meets sinr: The scheduling complexity of arbitrary topologies. In: International Symposium on Mobile Ad Hoc Networking and Computing, pp. 310–321 (2006)
34. Nassu, B.T., Duarte Jr, E.P., Ramirez Pozo, A.: A comparison of evolutionary algorithms for system-level diagnosis. In: The Genetic and Evolutionary Computation Conf., pp. 2053–2060 (2005)
35. Nelson, R., Kleinrock, L.: Spatial tdma: A collision-free multihop channel access protocol. Transactions on Communications **33**(9), 934–944 (1985)
36. Saad, M.: Wireless capacity maximization: A constrained genetic approach. In: International Conference on Communications, pp. 3855–3860 (2015)
37. Saad, M., El-Moursy, A., Alfawaz, O., Alnajjar, K., Abdallah, S.: Wireless link scheduling via parallel genetic algorithm. Concurrency and Computation: Practice and Experience **34**(6), e6783 (2022)
38. Saied, A., Overill, R.E., Radzik, T.: Detection of known and unknown ddos attacks using artificial neural networks. Neurocomputing **172**, 385–393 (2016)
39. Sgora, A., Vergados, D., Vergados, D.: A survey of tdma scheduling schemes in wireless multihop networks. ACM Computing Surveys **47**(3) (2015)

40. 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)
41. Teng, Y., Liu, M., Yu, F.R., Leung, V., Song, M., Zhang, Y.: Resource allocation for ultra-dense networks: A survey, some research issues and challenges. Communications Surveys & Tutorials **21**(3), 2134–2168 (2018)
42. Tsukiyama, S., Ide, M., Ariyoshi, H., Shirakawa, I.: A new algorithm for generating all the maximal independent sets. Journal on Computing **6**(3), 505–517 (1977)
43. White, C.: Data Communications and Computer Networks: A Business User's Approach, 8 edn. Cengage (2015)
44. Wu, D., Zhang, Z., Wu, S., Yang, J., Wang, R.: Biologically inspired resource allocation for network slices in 5g-enabled internet of things. Internet of Things Journal **6**(6), 9266–9279 (2018)