Sk-Greedy: A Heuristic Scheduling Algorithm for Wireless Networks under the SINR Model

Vinicius Fulber-Garcia
Elias P. Duarte Jr.
{vfgarcia,elias}@inf.ufpr.br
Federal University of Paraná
Curitiba, Paraná, Brazil

Fábio Engel de Camargo
fabioe@utfpr.edu.br
Technological Federal University of Paraná
Toledo, Paraná, Brazil

ABSTRACT

Recent advances in networking technologies such as massive Internet-of-Things and 6G-and-beyond cellular networks indicate a trend towards increasingly 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. However, finding the shortest schedule in wireless networks under the SINR model is an NP-hard problem. In this work, we present a greedy heuristic algorithm to solve that problem. The proposed solution, called Sk-Greedy (Stochastic k Greedy) algorithm produces a complete transmission schedule optimizing size, with the purpose of increasing the number of simultaneous transmissions (*i.e.*, spatial reuse) thus allowing devices to communicate as frequently as possible. Simulation results are presented, including comparisons of Sk-Greedy with the optimal algorithm. Results confirm that the solution requires short execution times to produce near-optimal schedules.

CCS CONCEPTS

• **Networks → Wireless access networks**; *Network manageability*.

KEYWORDS

Networks; Wireless; Scheduling; Optimization

ACM Reference Format:
Vinicius Fulber-Garcia, Elias P. Duarte Jr., and Fábio Engel de Camargo. 2022. Sk-Greedy: A Heuristic Scheduling Algorithm for Wireless Networks under the SINR Model. In *11th Latin-American Symposium on Dependable Computing (LADC 2022), November 21–24, 2022, Fortaleza/CE, Brazil.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3569902.3570190

1 INTRODUCTION

Current wireless networking technologies have become increasingly denser. Examples include wireless sensor networks [5, 20], cellular networks [6, 25], and massive Internet-of-Things [11, 16]. It is necessary to schedule communications over a wireless channel, as it is a shared medium [27]. At a given time, it is important to allow only transmissions that do not interfere with each other at a level that turns reception impossible. Besides mutual interference, other factors affect scheduling, in particular the fact that the power of the transmitted signal decreases with the distance.

In this work, the time is divided into equal-length slots, and devices are scheduled to transmit in specific time slots [23]. The devices are capable of recognizing the beginning of a slot. All devices must be scheduled to a time slot. Thus, the schedule is executed from the first slot to the last, and then again from the first, and so on. The trivial approach to scheduling is to assign a single device to each slot. This is not efficient, as each device has to wait for all others to have a chance to communicate before it can make the next transmission. In the context of high-speed networks, it is important to employ efficient schedules to guarantee the latency and throughput requirements.

According to the traditional model for scheduling in wireless networks (usually called *graph* model [9]) no two devices can be scheduled to the same time slot if they are within the coverage areas of each other. In dense networks, this becomes a problem: devices can be placed very close to each other. The SINR (Signal-to-Interference-and-Noise-Ratio) model becomes a relevant alternative in this case. The SINR model allows multiple simultaneous transmissions that do interfere with one another. Informally, the model computes all the interference resulting from the simultaneous transmissions, as well as the power of the transmitted signals at the receivers. With this information, it becomes possible to determine which transmissions can be simultaneous. Allowing devices whose coverage areas intersect to communicate simultaneously is called "spatial reuse". The higher the spatial reuse, the higher the efficiency of ultra-dense networks. Informally, the objective is to obtain the minimum schedule in terms of time slots, this problem has been called Shortest Link Scheduling (SLS) [28]. The Smallest Link Schedule has the lowest possible number of time slots and thus the largest number of simultaneous transmissions that can be scheduled across all slots. Such a schedule reduces the time each device has to wait until it can communicate, allowing all devices to communicate more frequently.

The SLS problem in SINR wireless networks has been proven to be NP-hard [9]. For this reason, much of the work in this area explores approximation algorithms [1–4, 10, 12, 21]. Although those algorithms are important from the theoretical point of view, they have little application in practice [22]. The fact that there is still a clear 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.

LADC 2022, November 21–24, 2022, Fortaleza/CE, Brazil

ACM ISBN 978-1-4503-9737-7/22/11…$15.00
https://doi.org/10.1145/3569902.3570190

The proposed strategy is a stochastic greedy heuristic that schedules links defined by the Down-to-Earth (DTE) heuristics [8]. The DTE heuristics presents a practical solution to determine the set of links that allow all devices to make transmissions. According to the DTE strategy, each device only communicates with the closest device and adjusts the transmission power to the minimum required to reach the receiver, plus a spare amount of power that will guarantee that the communication will succeed even if there are other simultaneous transmissions.

Sk-Greedy is a stochastic greedy heuristic that aims at a balance of both exploring/exploiting the search space of the scheduling problem. At first, k links are selected randomly among the links to be scheduled, and those links are evaluated. The best candidate from the set is selected as a local solution, updating the problem state and repeating the process until the algorithm finds a complete solution. Randomizing the candidates for each problem state enables the exploration of the problem search space. And the execution of a greedy approach to process the set of k random candidates exploits specific unexpected portions of the search space. For this reason, the Sk-greedy heuristic is classified as a stochastic heuristic. To improve the probability of finding the global optimum solution, the Sk-Greedy algorithm is executed multiple times, creating a population of several complete solutions for the optimization problem. Finally, all those complete solutions are compared to decide which is the best; that one is returned as the final result.

Experiments were executed to evaluate the proposed Sk-Greedy heuristic. An experiment compares the scheduling results from the proposed heuristic with the results from the optimal algorithm for scheduling links defined with the DTE strategy. The optimal algorithm checks all scheduling alternatives. Another experiment evaluates the ability of the Sk-Greedy scheduler to find optimized scheduling results for networks with a large number of communication links, which are infeasible to be processed by the optimal algorithm.

The remainder of this work is organized as follows. Section 2 gives a brief description of the SINR model and the DTE heuristic for link selection and power assignment. Section 3 presents the SK-Greedy heuristic scheduling strategy. Section **??** shows the results of two experiments conducted to evaluate SK-Greedy. Finally, conclusions are presented in Section 5.

2 THE DOWN-TO-EARTH SINR SCHEDULING STRATEGY

The SINR (Signal-to-Interference-plus-Noise Ratio) model takes into account the effects of cumulative interference on transmitted signals as well as the effects of path loss to determine whether reception is possible or not. The SINR model has been shown [19, 24] to provide a good approximation of real wireless communication channels.

The SINR model employs the signal-to-noise interference ratio to determine whether a given transmission will succeed. Equation 1 shows how the SINR threshold γ is employed to determine whether a transmission from device i can be correctly received by a device j. The SINR threshold is computed based on several parameters. The first is path loss, *i.e.* the power level of the transmitted signal P_{Ti} fades as it propagates from the transmitter to the receiver. Devices are placed on the Euclidean plane, and the power of the transmitted signal decreases according to the inverse of the distance $d(i,j)$ between i and j raised to the path loss, α. This specific signal propagation model has been called the geometric SINR model [10, 15].

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

The other parameters are noise and signal interference. Background noise (N_0) corresponds to spurious signals that cannot be avoided and interfere with any communication. Antennas and even the receiving circuit can be sources of noise. Interference occurs among multiple simultaneous transmissions. A device in the coverage area of multiple transmitters receives all transmitted signals each with a different power level. 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 [18]. 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.

As mentioned in the Introduction, in the SINR model, it is necessary to schedule communications to guarantee that they will succeed. The basic unit that is scheduled is a link from a transmitting device i to a particular receiver j. Each device must be the transmitter of some link so that each device has the opportunity to communicate. So, starting from a set of devices positioned on the Euclidean plane, the first problem is to determine the links themselves, *i.e.*, which device will communicate with which other device. In this work, we employ the so-called Down-to-Earth (DTE) heuristic [8] to solve this problem, described next.

First, the DTE heuristic determines that each device i only makes a transmission to the closest device j. In other words, the set of links to schedule L consist of links (i,j), such that the distance $d(i,j) \leq d(i,k), \forall k \in V$, where V is the set of devices. Next, the DTE heuristic determines the power level to be employed by each transmitter i. The minimum power level P_{Ti} required by device i to communicate 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 spatial reuse (multiple simultaneous transmissions), the power level must be above the minimum, so that interference can be tolerated. A configurable spare SINR level – γ_{spare} – can be employed to compute the power level to use. Thus the transmission power P_{Ti} adopted by device i is computed in a way to guarantee 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 level allows spatial reuse under certain conditions. The amount of interference P_{Φ} supported by each 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 $\frac{P_{Ti}}{d(i,j)^{\alpha}}$:

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

After determining the set of links, the next problem is scheduling which is an NP-hard problem. In the next section, we present the SK-Greedy heuristic to solve this problem.

3 THE SK-GREEDY SCHEDULER

In this section, we present the Sk-Greedy Scheduler, a simple but efficient algorithm to solve the scheduling problem in wireless networks under the SINR model and with links determined by the DTE heuristic strategy.

3.1 The Stochastic k Greedy Heuristic

Greedy heuristics evaluate the search space of an optimization problem looking for the best partial solutions locally, iteratively composing a complete solution. There are many ways to define which is the best local solution. In generic problems, greedy heuristics employ an objective function to evaluate all the candidates for the partial solution given the current problem state. Based on this evaluation, the heuristics thus decides the best-fitted candidate, update the problem state, and repeat the process until finding a complete solution. We call this approach "best-fit".

Another possible approach that greedy heuristics use to decide the local candidate to adopt is as follows. Given a set of candidates for a partial solution of the problem state, evaluate the candidates until finding the first viable candidate. Once detected, decide on that candidate, update the problem state, and repeat the process until the problem is completely solved. This approach is usually employed to create efficient heuristics (in terms of execution time) for constrained optimization problems. We call this approach "first-viable".

Overall, greedy heuristics are, by definition, used in the context of deterministic algorithms. Thus, these heuristics always return the same result given the same inputs. Due to that reason, we can say that greedy heuristics focus on exploiting the search space, not exploring it. It can be a good strategy for problems with low complexity and small search spaces. But, considering sophisticated scenarios (*e.g.*, a very large number of constraints, multiple candidates to solve each problem state, and a large search space), such as the scheduling problem in wireless networks under the SINR model, exploiting-based heuristics may be stuck in a local optimum far away from the global best solution.

In this way, we propose a new stochastic greedy heuristic that aims at a balance both exploring/exploiting the search space of the optimization problem in question. This heuristic, which we call Stochastic k-Greedy (or simply Sk-Greedy), can work both with the "best-fit" and "first-viable" approaches, evaluating a set of candidates precisely in the same way as in a traditional greedy heuristic. The main difference between the Sk-greedy heuristic to a regular deterministic greedy heuristic is on how it determines the set of candidates to be evaluated for each problem state.

The SK-Greedy heuristic, instead of evaluating all the available candidates to find a local solution for a problem state, selects k random candidates among the ones available and evaluates each them. Next, the best fitted or the first viable candidate from the k-set is taken as the local solution, updating the problem state and repeating the process until the algorithm finds a complete solution.

Randomizing the evaluated candidates for each problem state enables the exploration of the problem search space. And the execution of a greedy approach for the random k-set exploits a particular and unexpected portion of the same search space. For this reason, the Sk-greedy heuristic is classified as a stochastic heuristic.

To improve the probability of finding the global optimum solution, algorithms based on the Sk-Greedy heuristic can execute several times, creating a population of different complete solutions for the optimization problem. Finally, all those complete solutions are compared to decide which is the best, and that one is returned as the final result. The comparison can be done by simply employing their evaluation result values (mono-objective optimization) or using other strategies (multi-objective optimization) – such as Pareto frontiers [17]).

3.2 SINR Scheduling with Sk-Greedy

We modeled the scheduling problem of DTE links in wireless networks under the SINR model using the Sk-Greedy heuristic. We call the proposed algorithm Sk-Greedy Scheduler.

The Sk-Greedy Scheduler receives as input the coordinates of each device on the Euclidean plane. The scheduler executes a three-step algorithm which consists of (1) the establishment of the communication graph, (2) the generation of schedule candidates, and (3) candidate evaluation and decision. The first step consists of applying the DTE strategy (presented in Section 2) to define the communication links for the provided devices. The links, in turn, are processed in the second step of the algorithm.

The Sk-Greedy Scheduler composes a complete schedule candidate (second step) by generating its slots iteratively. The slot generation method is as follows. First, the algorithm starts with an empty slot, allocating a link randomly selected from the set of links to schedule, called available links. Next, the Sk-Greedy heuristic is run to define the other links that will be included in the current slot. This process occurs in three stages: (I) k links are randomly selected from the available ones (the user defines k as an input parameter); (II) each link is checked, verifying if it can coexist with the other links already assigned to the slot (*i.e.*, the transmissions can be simultaneous); (III) the first edge that satisfies this coexistence criterion (first-viable approach) is included in the slot, the other edges are returned to the set of available edges, and the algorithm moves to Stage I. If among the k links none could be assigned to the current slot with the first one, the slot is closed, included in the set of schedule candidates, and a next slot is started, repeating the entire process. The algorithm goes on generating slots until including all the available links have been assigned to a slot, thus having a complete schedule candidate. The process is shown in Figure 1.

In the third step, the proposed scheduler evaluates all the schedule candidates with a mono-objective function: minimizing the number of slots of the final schedule. The objective function is

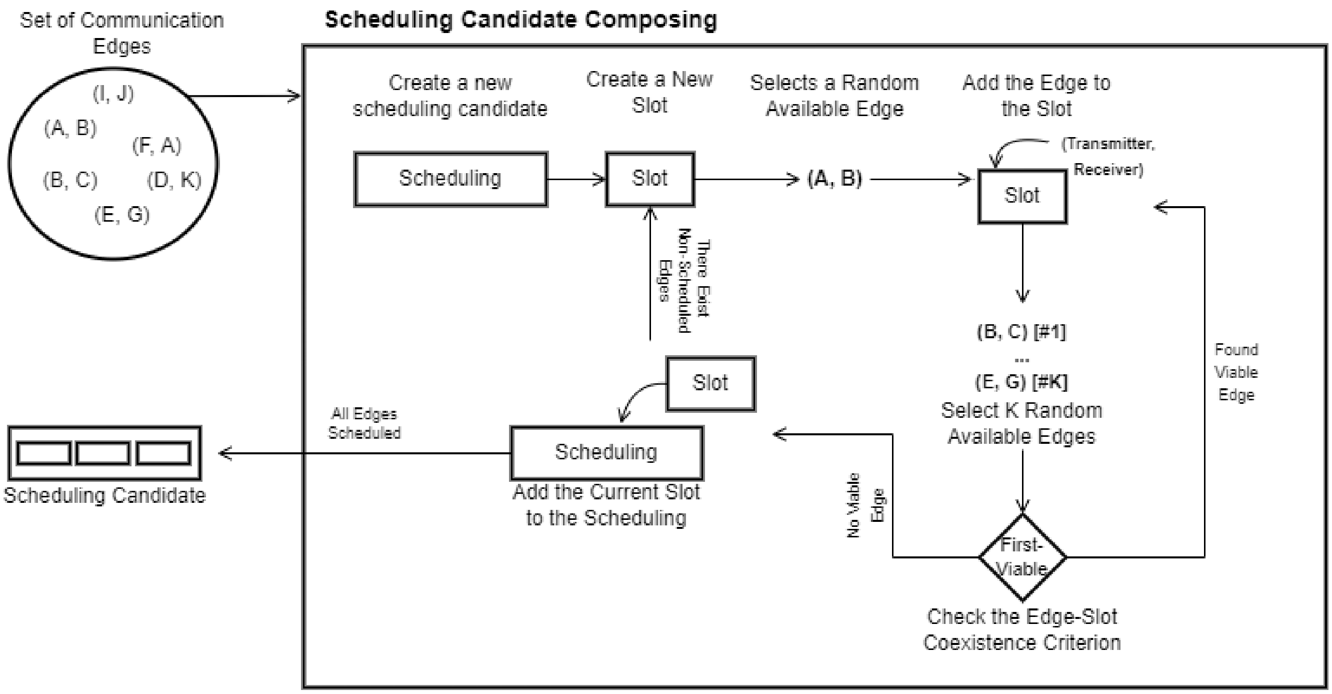

Figure 1: The Process of Composing a Schedule Candidate

mathematically expressed as Equation 5. Note that if different candidates present the same smallest number of slots, they are all returned as optimized solutions for the scheduling problem.

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

Finally, it is relevant to highlight that the algorithm always returns a viable schedule. In the best case, the returned schedule has a single slot (a rare and typically impractical scenario). In the worst case, however, the returned schedule has the number of slots equal to the number of links.

4 EXPERIMENTAL RESULTS

This section describes the experiments executed to evaluate the proposed Sk-Greedy Scheduler. The first experiment compares the scheduling results from the proposed heuristic with the results from the optimal algorithm for scheduling DTE links, which we call "Optimal DTE algorithm". The optimal algorithm checks all scheduling alternatives. The second experiment evaluates the Sk-Greedy Scheduler's ability to find optimized scheduling results for networks with a large number of communication links, which are infeasible to be processed by the optimal algorithm.

We employed the following SINR parameters for all experiments: the path loss was set to $\alpha = 4$; and the SINR threshold was set to $\gamma = 20$ decibels, $\gamma_{spare} = 50$ decibels and the background noise $N_0 = -90$dBm. The experiments were executed on a machine based on a Core I3-4010 processor, with 8GB RAM DDR3, running Ubuntu

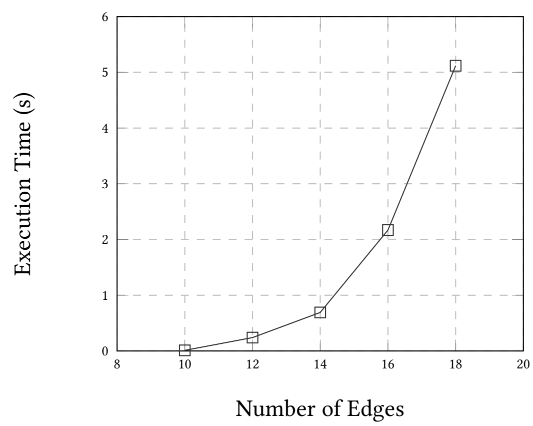

Figure 2: Execution Time (Optimal DTE)

16.04. Both the Sk-Greedy Scheduler and Optimal DTE were implemented using Python 3.9. Finally, all the experiments were repeated 30 times, achieving a confidence level of 95%.

For the first experiment, we simulated networks of 10, 12, 14, 16, and 18 devices randomly distributed on an area of 400 x 400 square meters. Thus, we executed the Optimal DTE algorithm to determine the mean execution time and the optimal scheduling size for each simulation scenario. Figure 2 presents the mean execution times, while the black bars in Figure 3 show the optimal scheduling sizes. We submitted the simulation files to the Sk-Greedy Scheduler (k=2), which was configured it to execute for the same time taken

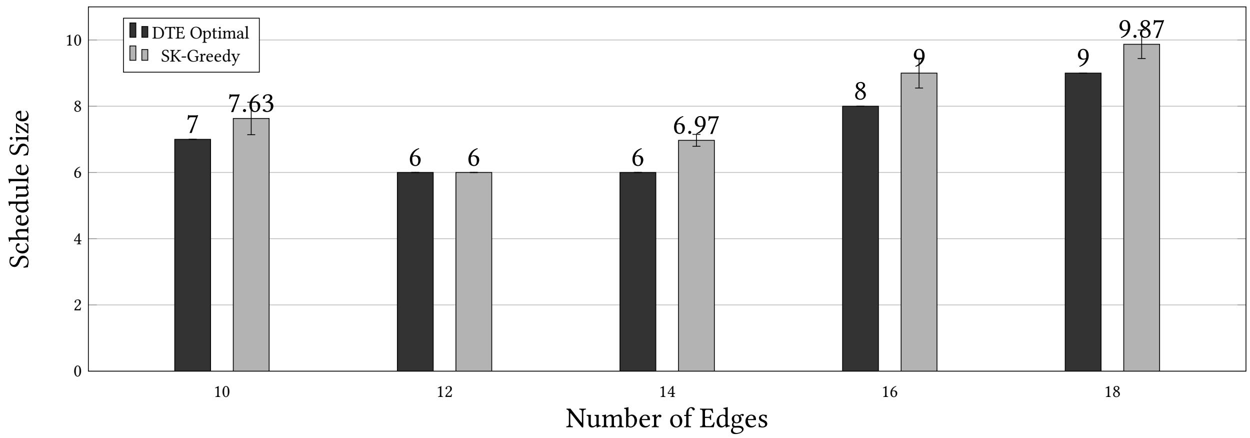

Figure 3: Comparison between DTE Optimal and SK-Greedy Scheduler (Area Size = 400x400m)

by the optimal algorithm. Figure 3 presents the mean size of the scheduling results returned by the Sk-Greedy Scheduler for each simulation with gray bars.

Figure 2 shows that the execution time of the Optimal DTE algorithm grows fast as the number of communication edges to be scheduled increases. In particular, increasing the number of communication edges from 10 to 18 (80%) made the execution time increase by 46527.28%. Furthermore, the optimal algorithm could not be executed (in our environment) for more than 18 communication links.

Figure 3 shows the schedule sizes returned by both the Optimal DTE (black bars) and Sk-Greedy (gray bars) algorithms. The results show that the proposed heuristic always produced the optimal or near-optimal results for this case study: the mean scheduling size did not exceed the optimal scheduling size plus one for all the scenarios. Furthermore, it is relevant to highlight that the k parameter of the Sk-Greedy Scheduler was set to 2 to increase the number of schedule candidates (exploration) instead of exploiting just a few candidates. This decision was motivated due to the small number of communication links to be scheduled and the limited execution times.

For the second experiment, we simulated the allocation of 50, 100, 250, and 500 devices randomly distributed in an area of 400 x 400 square meters. All those simulations are unfeasible to be processed by the Optimal DTE algorithm. In this way, we only executed the SK-Greedy Scheduler for these scenarios. We configured the scheduler to generate and evaluate 100 scheduling candidates with k=2 (fast execution). We measured the mean execution time to create candidates and decide on a scheduling result. Furthermore, we analyzed the mean size of the schedules resulting from 30 executions of the algorithm.

Figure 4 shows the mean execution time for each scenario of the second experiment. We can note that the execution time increases as the number of communication edges increases. It is the expected behavior, since increasing the number of communication edges means enlarging the search space to be considered by the scheduler. For this experiment, we observe that the execution time increases (close to) linearly. It occurs because with k=2, the scheduler does

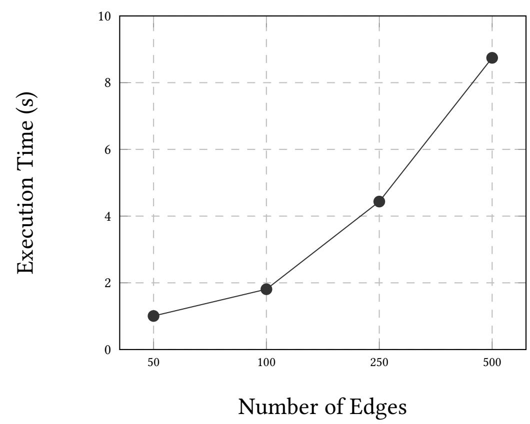

Figure 4: Execution Time For 100 Generations (SK-Greedy Scheduler)

not spend much time in frustrated attempts to include new links to a slot, keeping the mean number of evaluations per-link with a maximum difference of 38% for all the scenarios.

Finally, Figure 5 presents the mean schedule size returned by the Sk-Greedy Scheduler for each scenario of the second experiment. First, it is possible to conclude that the scheduler optimized the number of slots of resulting schedules, compared with the baseline (one link per slot). With the proposed strategy, the scheduler could reduce the number of slots from 2.22 (50 communication links) to 8.48 (500 communication links) times.

It is important to highlight that the schedule size reduction ratio increases as the number of communication edges increase in the same area. This occurs due to the characteristics of the DTE heuristic (each device creates a communication edge with the closest device). With a large number of devices, they are naturally closer to each other in comparison with a scenario with a few devices in an area with the same size. Thus, the power required to transmit is much lower, and the probability of finding multiple communication edges that can be scheduled to the same slot increases significantly.

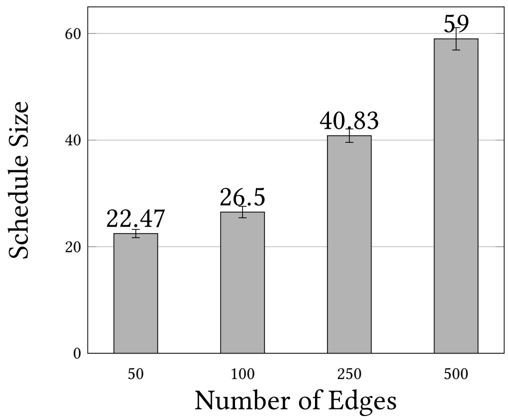

Figure 5: Scheduling Sizes From SK-Greedy Scheduler (Area Size = 400x400m)

5 CONCLUSIONS

Current wireless networks are increasingly denser and have very high requirements in terms of throughput and latency. The SINR model has the potential of improving the performance of very dense networks, as it allows spatial reuse, *i.e.*, multiple simultaneous transmissions within the same coverage area. As scheduling under the SINR model is NP-hard, it is necessary to develop efficient heuristics that are capable of producing solutions that are close to the optimal. Although several scheduling algorithms have been proposed in the literature, most have a purely theoretical focus. The DTE heuristics presents a practical solution to determine the set of links that allow all devices to make transmissions. According to that strategy, each device only communicates with the closest device and adjusts the transmission power to the minimum required to reach the receiver, plus a spare amount of power that will guarantee that the communication is possible even if there are other simultaneous transmissions. In this work, we presented Sk-Greedy, a greedy stochastic heuristic that solves the scheduling problem for DTE links efficiently, and produces results that are very close to those of the optimal algorithm. Future work includes investigating distributed scheduling strategies [7]. Providing the scheduler with a failure detection service [26] is the first step towards a dynamic scheduler that can support critical applications [14]. Multiple services can then be designed, such as reliable broadcast [13], among others.

REFERENCES

[1] Fahad Al-dhelaan, Pengjun Wan, and Huaqiang Yuan. 2016. A New Paradigm for Shortest Link Scheduling in Wireless Networks: Theory and Applications. In *Wireless Algorithms, Systems, and Applications*. Spring, China, 24–36.
[2] Mohamed Sofiane Batta, Saad Harous, Lemia Louail, and Zibouda Aliouat. 2019. A Distributed TDMA Scheduling Algorithm for Latency Minimization in Internet of Things. In *IEEE International Conference on Electro Information Technology*. IEEE, 108–113.
[3] Douglas M. Blough, G. Resta, and P. Santi. 2010. Approximation Algorithms for Wireless Link Scheduling With SINR-Based Interference. *IEEE/ACM Transactions on Networking* 18, 6 (2010), 1701–1712.
[4] Gurashish Brar, Douglas M. Blough, and Paolo Santi. 2006. Computationally Efficient Scheduling with the Physical Interference Model for Throughput Improvement in Wireless Mesh Networks. In *Annual International Conference on Mobile Computing and Networking*. ACM, 2–13.
[5] Yuchao Chang, Xiaobing Yuan, Baoqing Li, Dusit Niyato, and Naofal Al-Dhahir. 2018. A joint unsupervised learning and genetic algorithm approach for topology control in energy-efficient ultra-dense wireless sensor networks. *IEEE Communications Letters* 22, 11 (2018), 2370–2373.
[6] Mostafa Zaman Chowdhury, Md. Shahjalal, Shakil Ahmed, and Yeong Min Jang. 2020. 6G Wireless Communication Systems: Applications, Requirements, Technologies, Challenges, and Research Directions. *IEEE Open Journal of the Communications Society* 1 (2020), 957–975.
[7] José Carlos da Silva and Flávio Assis. 2019. A distributed algorithm to schedule TSCH links under the SINR model. *Design Automation for Embedded Systems* 23, 1 (2019), 21–39.
[8] Fábio Engel and Elias Duarte. 2021. A Down-to-Earth Scheduling Strategy for Dense SINR Wireless Networks. In *Latin-American Symposium on Dependable Computing*. SBC, 1–6.
[9] Olga Goussevskaia, Yvonne Anne Oswald, and Rogert Wattenhofer. 2007. Complexity in Geometric SINR. In *ACM Int. Symp. Mobile Ad Hoc Networking and Computing*. ACM, 100–109.
[10] Olga Goussevskaia, Yvonne-Anne Pignolet, and Roger Wattenhofer. 2010. *Efficiency of Wireless Networks: Approximation Algorithms for the Physical Interference Model*. Now Foundations and Trends.
[11] Fengxian Guo, F. Richard Yu, Heli Zhang, Xi Li, Hong Ji, and Victor C. M. Leung. 2021. Enabling Massive IoT Toward 6G: A Comprehensive Survey. *IEEE Internet of Things Journal* 8, 15 (2021), 11891–11915.
[12] Magnús M. Halldórsson and Roger Wattenhofer. 2019. *Wireless Network Algorithmics*. Springer International Publishing, 141–160.
[13] Denis Jeanneau, Luiz A Rodrigues, Luciana Arantes, and Elias P Duarte Jr. 2017. An autonomic hierarchical reliable broadcast protocol for asynchronous distributed systems with failure detection. *Journal of the Brazilian Computer Society* 23, 1 (2017), 1–14.
[14] Neeraj Kumar, Al-Sakib Khan Pathan, Elias P Duarte, and Riaz Ahmed Shaikh. 2015. Critical applications in vehicular ad hoc/sensor networks. *Telecommunication Systems* 58, 4 (2015), 275–277.
[15] Nhat X. Lam, Tien Tran, Min Kyung An, and Dung T. Huynh. 2015. A note on the complexity of minimum latency data aggregation scheduling with uniform power in physical interference model. *Theoretical Computer Science* 569 (2015), 70–73.
[16] Gyeongheon Lee and Joosang Youn. 2020. Group-based Transmission Scheduling Scheme for Building LoRa-based Massive IoT. In *International Conference on Artificial Intelligence in Information and Communication*. 583–586.
[17] A. V. Lotov and K. Miettinen. 2008. Visualizing the Pareto frontier. In *Multiobjective optimization*. Springer, 213–243.
[18] Shaohe Lv, Xiaodong Wang, and Xingming Zhou. 2010. Scheduling under SINR Model in Ad Hoc Networks with Successive Interference Cancellation. In *IEEE Globecom*. IEEE, 1–5.
[19] Ritesh Maheshwari, Shweta Jain, and Samir R. Das. 2008. A Measurement Study of Interference Modeling and Scheduling in Low-Power Wireless Networks. In *ACM Conference on Embedded Network Sensor Systems*. ACM, 141–154.
[20] Renato EN Moraes, Welber WF dos Reis, Helder RO Rocha, and Daniel JC Coura. 2020. Power-efficient and interference-free link scheduling algorithms for connected wireless sensor networks. *Wireless Networks* 26, 5 (2020), 3099–3118.
[21] Thomas Moscibroda and Roger Wattenhofer. 2006. The Complexity of Connectivity in Wireless Networks. In *IEEE Computer and Communications Societies, Barcelona, Spain*. IEEE, 1–13.
[22] Mohamed Saad. 2015. Wireless capacity maximization: A constrained genetic approach. In *IEEE International Conference on Communications*. 3855–3860.
[23] Aggeliki Sgora, Dimitrios J. Vergados, and Dimitrios D. Vergados. 2015. A Survey of TDMA Scheduling Schemes in Wireless Multihop Networks. *ACM Comput. Surv.* 47, 3 (2015).
[24] Dongjin Son, Bhaskar Krishnamachari, and John Heidemann. 2006. Experimental Study of Concurrent Transmission in Wireless Sensor Networks. In *The ACM Conference on Embedded Networked Sensor Systems*. ACM, 237–250.
[25] Yinglei Teng, Mengting Liu, F Richard Yu, Victor CM Leung, Mei Song, and Yong Zhang. 2018. Resource allocation for ultra-dense networks: A survey, some research issues and challenges. *IEEE Communications Surveys & Tutorials* 21, 3 (2018), 2134–2168.
[26] Rogerio C Turchetti and Elias P Duarte Jr. 2017. NFV-FD: Implementation of a failure detector using network virtualization technology. *International Journal of Network Management* 27, 6 (2017), e1988.
[27] Curt M. White. 2015. *Data Communications and Computer Networks: A Business User's Approach* (8 ed.). Cengage.
[28] Jiguo Yu, Baogui Huang, Xiuzhen Cheng, and Mohammed Atiquzzaman. 2016. Shortest link scheduling algorithms in wireless networks under the SINR model. *IEEE Transactions on Vehicular Technology* 66, 3 (2016), 2643–2657.