On Intelligent Reflecting Surfaces Element Allocation Using Genetic Algorithms

Shashi Prabh
School of Engineering and Applied Science
Ahmedabad University
Ahmedabad, India
Email: shashi.prabh@ahduni.edu.in

***Abstract*—Intelligent reflecting surfaces (IRS) are surfaces consisting of an array of a large number of passive but individually controlled reflecting elements. This paper explores the use of genetic algorithms for optimal allocation of the reflecting elements in a multi-user setting. In particular, it considers maximizing the sum-rate and equalizing the throughput in a wireless environment consisting of multiple co-located IRSes. A real-coded genetic algorithm for solving this problem is presented. Experimental results suggest that the method is effective.**

***Index Terms*—Intelligent reflecting surfaces, IRS element allocation, Genetic algorithms, Sum-rate maximization, Throughput equalization.**

I. Introduction

Intelligent Reflecting Surfaces (IRS) consist of an array of passive but individually configurable scattering elements. The electromagnetic properties of the individual elements can be controlled using software for creating smart wireless environments. In comparison with the conventional approach of improving the transmitter and the receiver, IRS adds the capability of controlling and optimizing the wireless environment which can be used for overcoming blind areas, increasing data rate at selected locations, backscatter communications and enabling many other novel applications by coating objects with sensing IRS [1], [2]. IRSes are particularly suited for millimeter wave communications [3] and are envisioned to be the enabler of low latency applications in 6G networks.

There have been extensive studies on the physical layer aspects of IRSes and IRS-aided networks. Several results exist where power, phase shift, subcarrier and reflection coefficient are optimized for sum-rate maximization and energy minimization [4]–[7]. However, medium access control (MAC) related issues in IRS-assisted networks are largely open [8]. This paper addresses allocation of IRS elements to the users for sum-rate (or throughput) maximization and throughput equalization in a setting involving multiple co-located IRSes where multiple users are served simultaneously. Given channel gains, transmit power and reflection coefficient as input, this work solves the problem of assigning a particular IRS and determining the number of elements to a user. From the results in [9], it is straightforward to see that this problem is NP-hard. In the MAC framework presented in [10], the number of elements of an IRS is divided equally among orthogonal subchannels forming "RIS groups," and users are associated with an RIS group. In [8], the design of MAC architectures are presented where TDMA is used within orthogonal subchannels. Time, frequency, transmit power and IRS phase configuration is computed for each user. The use of AI/ML is proposed to overcome the excessive computational complexity associated with search in high-dimensional space.

Genetic algorithms are metaheuristics that have been found to perform very well on several intractable problems, such as, Traveling Salesman Problem [11], multiprocessor scheduling [12], project scheduling with resource limits [13] and solving Sudoku puzzles [14]. More recently, genetic algorithms have been used to design and assist neural networks [15], [16]. Optimization of phase shifts of IRS elements using genetic algorithm has been reported in [17], [18]. A study of the use of genetic algorithms for the allocation of IRS elements is presented in this paper. The algorithm essentially takes channel gains for all paths as input and produces IRS assignments and element allocation for each user. Its performance is evaluated and compared with `BlackBoxOptim` optimizer [19] and direct source to destination line-of-sight sum-rate as the baseline. Experimental results suggest that the genetic algorithm is effective in solving this problem.

The rest of this paper is organized as follows. We present the problem formulation in Section II followed by a description of the genetic algorithm in Section III. We present the experimental details and results in Section IV, and conclusions in Section V.

II. Problem formulation

We consider IRS-aided multi-user communication systems where M IRSes assist the transmissions between N users and a base station. Let each IRS consist of L reflective elements. Given the channel characteristics between the users and the base station, the users and the IRSes and the IRSes and the base station as the input, an optimal assignment of an IRS combined with the allocation of IRS elements to all the users is computed. The base station, assumed to be able to control the IRS controllers, then notifies the respective controllers. This can be a model for scheduling a subset of N users in a TDMA slot (the subset of users can change from one slot to another). This can also be a model for the saturated node scenario where the transmit buffers of the N users is never empty.

Consider deterministic line-of-sight flat fading channel between the base station and the user, between the base station and IRS and between the IRS and the user. Let $\beta_{sd} \in [0,1]$ and $\beta_{\text{IRS}} \in [0,1]$ denote the channel gains between the base station and the user on the direct path and on the path via IRS, respectively. The channel gain between a user and a base station through an IRS varies for different elements of the IRS. In the following, we assume that the physical dimensions of the IRS are small enough so that the deviations of the element-wise channel gain from the IRS average, β_{IRS}, are negligible. Let $\mu \in (0,1]$ be the amplitude reflection coefficient of the IRS. We denote the variance of the additive white Gaussian noise at the receiver by σ^2. If l reflective elements are allocated to a user then the SNR at the receiver is

$$\text{SNR}_{\text{IRS}} = \frac{P(\sqrt{\beta_{sd}} + l\mu\sqrt{\beta_{IRS}})^2}{\sigma^2}, \tag{1}$$

where P is the transmit power [20]. Let the users be assigned orthogonal subchannels where the vector $\boldsymbol{B} \in \mathbb{R}^N$ denotes the bandwidth allocated to each user. In the following, we use notation of the form $l_{i,j}$ to denote the $(i,j)^{\text{th}}$ element of the matrix $\boldsymbol{l}$. The data rate for user i via direct path and IRS j is given by

$$\rho_{i,j}(l_{i,j}) = B_i \log_2(1 + \frac{P_i(\sqrt{H_i} + l_{i,j}\mu\sqrt{G_{i,j}})^2}{B_i\sigma^2}), \tag{2}$$

where $\boldsymbol{P} \in \mathbb{R}^N$, $\boldsymbol{l} \in \mathbb{N}^{N\times M}$, $\boldsymbol{H} \in \mathbb{R}^N$ denotes the vector containing β_{sd} and $\boldsymbol{G} \in \mathbb{R}^{N\times M}$ denotes the matrix containing β_{IRS}. We define the sum-rate maximization problem with the constraint that each user gets non-zero rate as:

$$\max_{\boldsymbol{l}} \quad \sum_{i=1}^{N} B_i \log_2(1 + \frac{P_i(\sqrt{H_i} + \sum_{j=1}^{M} l_{i,j}\mu\sqrt{G_{i,j}})^2}{B_i\sigma^2}) \tag{P1}$$

$$\text{s.t.} \quad \sum_i l_{i,j} \le L \ \forall j, \tag{3a}$$

$$l_{i,j} \in \{0,1,\ldots,L\} \ \forall i,\ \forall j, \tag{3b}$$

$$\sum_j \mathbf{1}(l_{i,j} > 0) = 1 \ \forall i, \tag{3c}$$

$$\rho_i(\boldsymbol{l}) > 0 \ \forall i \tag{3d}$$

$\mathbf{1}()$ is the indicator function. Constraint (3c) ensures that all the elements allocated to a given user are assigned from only one IRS in order to avoid interference caused by the signals reflected from multiple IRSes.

The sum-rate maximization problem (P1) usually leads to large variations in the user rates, which was also observed in our evaluations. Consequently, we define the throughput equalization problem where the objective is the minimization of the variance of data rates:

$$\min_{\boldsymbol{l}} \quad \text{var}(B_i \log_2(1 + \frac{P_i(\sqrt{H_i} + \sum_{j=1}^{M} l_{i,j}\mu\sqrt{G_{i,j}})^2}{B_i\sigma^2})) \tag{P2}$$

$$\text{s.t.} \quad (3a) - (3d)$$

355.67	2.45	255.56	2.22	367.85	1.5	600.31	3.66

Fig. 1: Example of chromosome encoding for 5 users

As stated earlier, these discrete optimization problems are NP-hard. The next section presents a genetic algorithm designed to solve these problems.

III. Genetic Algorithm

A. Preliminaries

Genetic algorithms are metaheuristic and belong to the class of so-called population methods. Unlike gradient descent based methods and simulated annealing where a single design point is used to guide the search, in genetic algorithms the search begins with a set of design points, called initial *population,* that are supposed to be distributed throughout the search space. The search is then guided by evolving a set of design points that are closer to the optimal in an iterative manner. The iterative evolution of the design points, called *chromosomes*, is made to resemble natural selection and consists of three steps, namely, *selection, crossover* and *mutation*. The selection process determines the good chromosomes using a given fitness function and replicates them while discarding the bad ones. The crossover process creates new chromosomes from the population left after the selection process by splicing two solutions at one or more crossover points. Lastly, mutation alters the chromosomes at random locations to maintain diversity and to avoid getting stuck in local minima. After each iteration, the evolved population is evaluated and the algorithm stops when the terminating condition is satisfied. Please refer to [21] for more details.

B. The Algorithm

The following algorithm is a so-called *real-coded genetic algorithm.* These genetic algorithms are known to be robust and suit the optimization problems being solved here. In the following we describe chromosome encoding, initial population generation and the fitness function. As described in the preliminaries, after the initial population generation the algorithm carries out selection, crossover and mutation, in order, iteratively. The methods used in these steps are specified below. At the end of each iteration the fitness of the population is evaluated. The algorithm terminates if the termination condition is satisfied.

Encoding: The chromosomes are encoded as vectors of floating point numbers in $\mathbb{R}^{2N}$ where odd indices contain elements allocation $l_{i,j} \in [1, L+1)$ and even indices identify the IRS $j \in [1, M+1)$ (Fig. 1). The integral solution is recovered by *rounding down* each of the $2N$ elements of the fittest chromosome.

Initial population: The initial population is generated by drawing element allocations and IRS assignments uniformly randomly in $[1, M * L/N]$ and $[1, M+1)$, respectively. Drawing element allocations from $[1, L]$ causes extremely

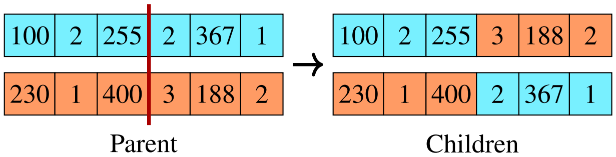


Fig. 2: Single point crossover

large convergence time when N is large and M/N is small (0.1 or smaller).

Fitness function: The data rate for each user is calculated according to (2). The fitness function returns the value of the objective function when all the constraints are satisfied, and returns $-\infty$ when any constraint is violated or any IRS identifier gets out of range.

Selection: Tournament selection method selects the fittest chromosome from a set of s randomly chosen chromosomes where s is an input parameter. Truncation selection method selects a random chromosome from the fittest s chromosomes in the population. Tournament and truncation selection methods work better than other selection methods on the instances that were tested. Out of these tournament selection performed better on more instances.

Elitism. Elitism where the fittest solution is always duplicated in the next generation is used. In the experiments, the algorithm without elitism was usually unable to regenerate the best solution once it got destroyed.

Crossover: Single Point crossover method for crossover is used. A random index in the chromosome vector is chosen and the chromosomes of the parents are spliced at this point. An example is shown in Fig. 2.

Mutation: For mutation, we use the bit string representation of the real-coded chromosomes. The bit string is traversed and the individual bits of the string is are flipped with a given (small) probability.

Termination: Termination is determined by either convergence or external measures like maximum execution time or maximum number of iterations.

IV. Experimental Results

This section presents the details of the simulation scenario and the results. The set-up consisted of 3 IRSes, a base station and a set of users shown in Figure 3. The users were located in a room of size $60\,\mathrm{m}$ X $20\,\mathrm{m}$. We simulated two cases: 5 users and 100 users. In the first case, the users were located at {(0,0), (15,0), (30,0), (45,0) and (60,0)} and in the second case 100 users were placed uniformly randomly across the room. The location along the vertical axis was restricted to upto $18\,\mathrm{m}$ to avoid too much disparity in SNR. Three IRSes were located at (0,20), (30, 20) and (60, 20). The base station was located at (100, 10). We refer to this setting as *Scenario A*. In *Scenario B*, an obstruction is placed at the right wall of the room which *totally blocks* the direct base station to user communication. The algorithm and the simulator were implemented in `Julia`. The network and algorithm parameters are listed in Table I and II. We calculated the input matrix $\boldsymbol{G}$ using the 3GPP pathloss

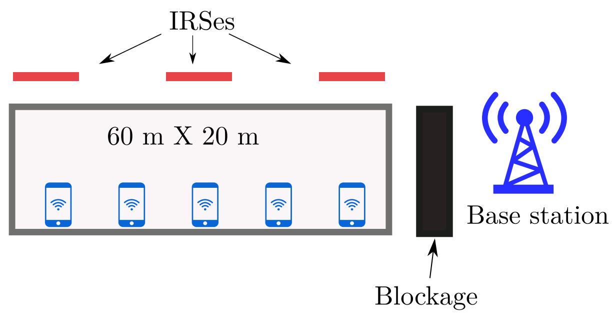


Fig. 3: Simulation scenarios. Scenario A excludes the blockage shown here.

TABLE I: Network parameters

Parameter	Value
Transmit power	10 mW
Carrier frequency	2.4 GHz
Bandwidth	10 MHz
Noise figure	10 dB
No. of IRS	3
No. of IRS elements	10000

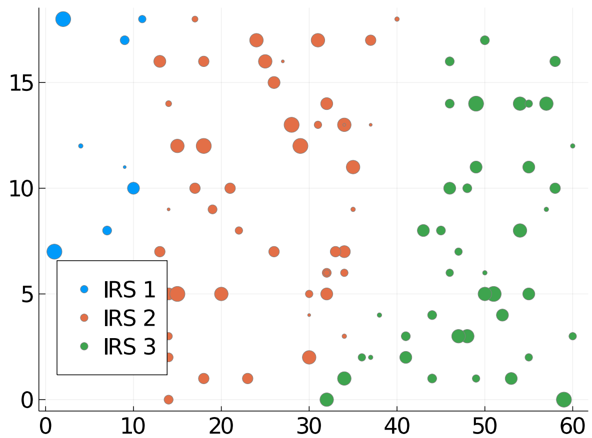
model for the Urban Scenario [22]. Further, we ensured that the reflected power from an IRS did not exceed $P_{tx}/2$ [23].

Differential evolution is another population method. We used its implementation in `BlackBoxOptim` optimizer [19] for performance comparison. In particular, we used the `adaptive_de_rand_1_bin` method available in the package. We also report the sum-rate obtained from direct line-of-sight paths between users and base station (had there been no obstruction and no IRSes). For Scenario A, the sum-rates obtained were $514\,\mathrm{Mbps}$ and $534\,\mathrm{Mbps}$ for the 5 users case by the genetic algorithm and `BlackBoxOptim`, respectively. The corresponding sum-rates for Scenario B were $503\,\mathrm{Mbps}$ and $524\,\mathrm{Mbps}$. Direct line-of-sight sum-rate was $151\,\mathrm{Mbps}$. We see that the genetic algorithm and differential evolution implementation of `BlackBoxOptimof` achieved very similar sum-rates.

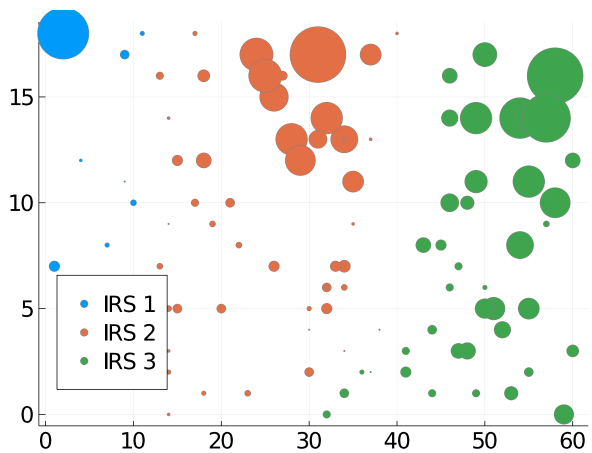
For the 100 users case, `BlackBoxOptim` failed and did not produce a feasible solution. The genetic algorithm obtained $2.909\,\mathrm{Gbps}$ for Scenario A and $1.056\,\mathrm{Gbps}$ for Scenario B while the direct line-of-sight sum-rate was $1.734\,\mathrm{Gbps}$. Unlike the 5 users case where using IRSes improved the sum-rate by over a factor of 3, in this 100 users the improvement was about 1.5 times, and in Scenario B the IRSes were the bottleneck.

TABLE II: Genetic algorithm parameters

Parameter	Value
Population size	100
Selection method	Tournament
Selection size	10
Crossover method	Single point
Mutation method	Random bitwise
Mutation probability	0.25 for 5 users and 0.05 for 100 users

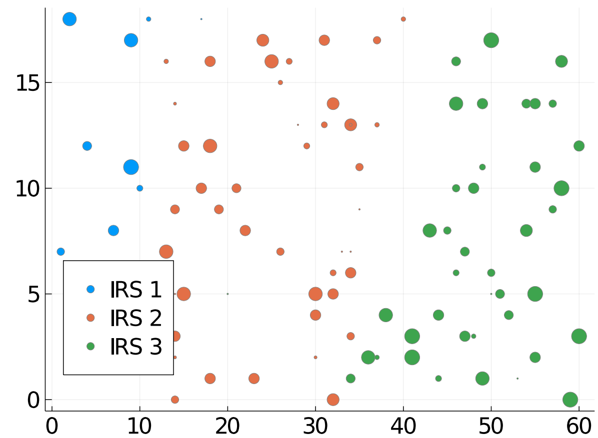


(a) Elements allocation

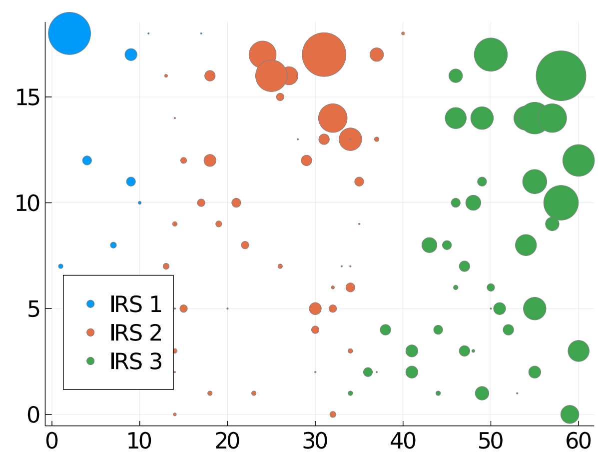


(b) Data rate

Fig. 4: 100 users, Scenario A



(a) Elements allocation



(b) Data rate

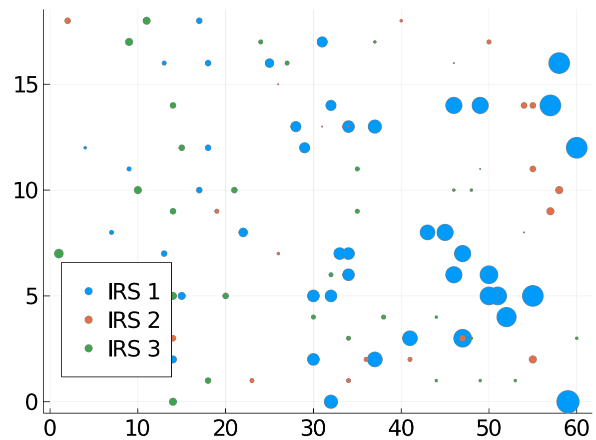
Fig. 5: 100 users, Scenario B

TABLE III: Performance comparison. GA: Genetic Algorithm, DE: Differential Evolution, Direct: Non-IRS-aided sum-rate if direct line-of-sight paths between users and base station were available

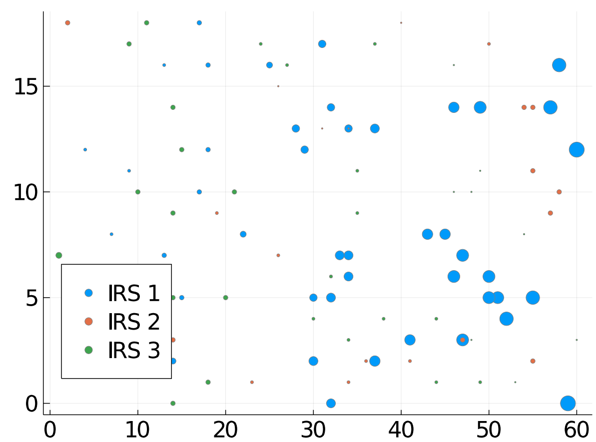
	5 users			100 users		
Method	GA	DE	Direct	GA	DE	Direct
Scenario A sum-rate (Mbps)	514	534	151	2909	—	1734
Scenario B sum-rate (Mbps)	503	524	151	1056	—	1734
Execution time (s)	1.400	0.06	NA	24	—	NA

There were not enough elements per user to compensate for the additional power loss due to longer path lengths from the users to the base station. The execution times on Intel Core i5-8250U CPU with 7.5 GB RAM and Ubuntu 18.04 distribution of Linux OS are summarized in Table III. Figs. 4 and 5 visualize the results. Figs. 4a and 5a show the user locations and the number of elements as well as the IRS allocated to each user. Figs. 4b and 5b show the resulting data rate for each user. The size of the markers is proportional to the allocation and the rate in the respective figures. IRS 1, 2 and 3 refer to the IRSes located at (0,20), (30, 20) and (60, 20), respectively. Not surprisingly, the optimal solution divides the area into 3 distinct sections where the nearest IRS is assigned to the users. Further, nodes near IRSes get much larger data rates and being the closest to the base station, very many nodes associated with IRS 3 receive large data rates.

Throughput equalization: The large data rate disparity as a result of sum-rate maximization may not be desirable for human users. To equalize the throughput, the fitness function was modified to return negative of the variance of data rates (P2). The rest of the algorithm and the settings were kept the same as that for the 100 users sum-rate maximization. Figs. 6a and 6b show the elements allocation. Fig. 7 shows the resulting data rates for each user for Scenario B. The figures for the data rate for the two scenarios were similar. As compared to the sum rate maximization, convergence for equalization took significantly longer (15 s). The resulting data rates were more equitable and the standard deviation of the data rates decreased from 2.9×10^8 to 1.9×10^8 for Scenario A and from 1.2×10^8 to 4.36×10^6 for Scenario B. The sum rates after equalization were $1.945\,\mathrm{Gbps}$ and $39.05\,\mathrm{Mbps}$ for Scenarios A and B, respectively. The equalization is less effective in Scenario A because the data rates on direct source-destination paths remain the same.



(a) Scenario A. Elements allocation



(b) Scenario B. Elements allocation

Fig. 6: 100 users. Throughput equalization

V. Conclusions and Future Work

This work presented a genetic algorithm for optimizing IRS element allocation in IRS-aided wireless environments. It showed that using genetic algorithms is a promising approach for solving this NP-hard problem. It can help in designing fast heuristics for IRS element allocation for a given deployment set-up. Simulations on a set-up consisting of 3 IRSes of 10000 elements each showed that when the ratio of IRS element count to number of users is large, using IRS can lead to significant increase in the sum-rate from the baseline of direct user to base station channel. However, when the number of users get large, the sum-rate increase using IRS gets smaller because of decrease in the available number of elements per user. The algorithm can also be used to equalize throughput though the convergence was slower in the experiments. Interesting future directions are nonorthogonal medium access and speeding-up the convergence.

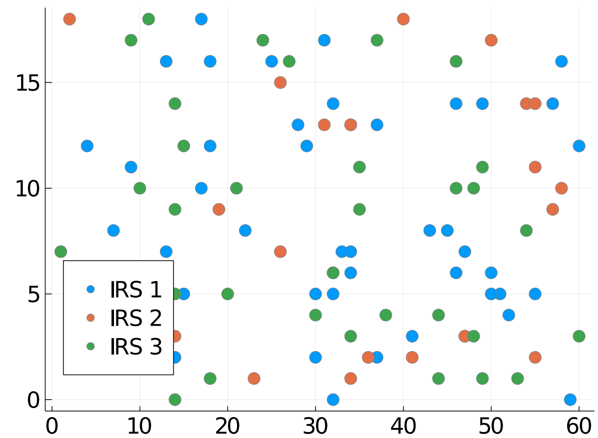


Fig. 7: Rates after throughput equalization.

Acknowledgment

The comments of Prof. Emil Björnsson and anonymous reviewers helped in improving the quality of the paper. The author is grateful for their time. This work was supported by Ahmedabad University grant number URBSEASE20A6/SUG/20-21/03-SP-08.23.

References

[1] M. Di Renzo, M. Debbah, D.-T. Phan-Huy, A. Zappone, M.-S. Alouini, C. Yuen, V. Sciancalepore, G. C. Alexandropoulos, J. Hoydis, H. Gacanin *et al.*, "Smart radio environments empowered by reconfigurable AI meta-surfaces: An idea whose time has come," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, pp. 1–20, 2019.

[2] F. Liu, Y. Cui, C. Masouros, J. Xu, T. X. Han, Y. C. Eldar, and S. Buzzi, "Integrated sensing and communications: Towards dual-functional wireless networks for 6G and beyond," *arXiv preprint 2108.07165*, 2021.

[3] D. Pozar, S. Targonski, and H. Syrigos, "Design of millimeter wave microstrip reflectarrays," *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 2, pp. 287–296, 1997.

[4] F. Fang, J. Cheng, and Z. Ding, "Joint energy efficient subchannel and power optimization for a downlink NOMA heterogeneous network," *IEEE Trans. on Vehicular Technology*, vol. 68, no. 2, pp. 1351–1364, 2019.

[5] J. Zuo, Y. Liu, Z. Qin, and N. Al-Dhahir, "Resource allocation in intelligent reflecting surface assisted NOMA systems," *IEEE Transactions on Communications*, vol. 68, no. 11, pp. 7170–7183, 2020.

[6] H. Guo, Y.-C. Liang, J. Chen, and E. G. Larsson, "Weighted sum-rate maximization for reconfigurable intelligent surface aided wireless networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3064–3076, 2020.

[7] T. Bai, C. Pan, H. Ren, Y. Deng, M. Elkashlan, and A. Nallanathan, "Resource allocation for intelligent reflecting surface aided wireless powered mobile edge computing in OFDM systems," *IEEE Transactions on Wireless Communications*, vol. 20, no. 8, pp. 5389–5407, 2021.

[8] X. Cao, B. Yang, C. Huang, C. Yuen, M. Di Renzo, Z. Han, D. Niyato, H. V. Poor, and L. Hanzo, "AI-Assisted MAC for Reconfigurable Intelligent-Surface-Aided Wireless Networks: Challenges and Opportunities," *IEEE Communications Magazine*, vol. 59, no. 6, pp. 21–27, 2021.

[9] Z.-Q. Luo and S. Zhang, "Dynamic spectrum management: Complexity and duality," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 1, pp. 57–73, 2008.

[10] X. Cao, B. Yang, H. Zhang, C. Huang, C. Yuen, and Z. Han, "Reconfigurable-intelligent-surface-assisted MAC for wireless networks: Protocol design, analysis, and optimization," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 14 171–14 186, 2021.

[11] J. Grefenstette, R. Gopal, B. Rosmaita, and D. Van Gucht, "Genetic algorithms for the traveling salesman problem," in *Proceedings of the first International Conference on Genetic Algorithms and their Applications*, vol. 160, no. 168. Lawrence Erlbaum, 1985, pp. 160–168.

[12] E. S. Hou, N. Ansari, and H. Ren, "A genetic algorithm for multiprocessor scheduling," *IEEE Transactions on Parallel and Distributed systems*, vol. 5, no. 2, pp. 113–120, 1994.

[13] S. Hartmann, "A competitive genetic algorithm for resource-constrained project scheduling," *Naval Research Logistics*, vol. 45, no. 7, pp. 733–750, 1998.

[14] T. Mantere and J. Koljonen, "Solving and rating sudoku puzzles with genetic algorithms," in *New Developments in Artificial Intelligence and the Semantic Web, Proceedings of the 12th Finnish Artificial Intelligence Conference STeP*, 2006, pp. 86–92.

[15] T. Shinozaki and S. Watanabe, "Structure discovery of deep neural network based on evolutionary algorithms," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 4979–4983.

[16] E. Bochinski, T. Senst, and T. Sikora, "Hyper-parameter optimization for convolutional neural network committees based on evolutionary algorithms," in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 3924–3928.

[17] Z. Peng, T. Li, C. Pan, H. Ren, W. Xu, and M. D. Renzo, "Analysis and optimization for RIS-aided multi-pair communications relying on statistical CSI," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 4, pp. 3897–3901, 2021.

[18] V. D. P. Souto, R. D. Souza, B. F. UchoBa-Filho, and Y. Li, "Intelligent reflecting surfaces beamforming optimization with statistical channel knowledge," *Sensors*, vol. 22, no. 6, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/6/2390

[19] Robert Feldt, "BlackBoxOptim." [Online]. Available: https://github.com/robertfeldt/BlackBoxOptim.jl

[20] E. Björnson, Özgecan. Özdogan, and E. G. Larsson, "Intelligent reflecting surface versus decode-and-forward: How large surfaces are needed to beat relaying?" *IEEE Wireless Communications Letters*, vol. 9, no. 2, pp. 244–248, 2020.

[21] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998.

[22] "Evolved Universal Terrestrial Radio Access (E-UTRA). Further advancements for E-UTRA physical layer aspects (Release 9)," 3rd Generation Partnership Project (3GPP), Tech. Rep. 36.814 V9.2.0, Apr 2017.

[23] E. Björnson and L. Sanguinetti, "Demystifying the power scaling law of intelligent reflecting surfaces and metasurfaces," in *2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2019, pp. 549–553.