Beamforming Optimization for Multiuser Wireless Systems using Meta-Heuristics

Pedro Bento*†, Carlos Henggeler Antunes†‡, Marco Gomes*†, Rui Dinis*§ and Vitor Silva*†
*Instituto de Telecomunicações (IT), Portugal
†Department of Electrical and Computer Engineering, University of Coimbra, 3030-290 Coimbra, Portugal
‡INESC Coimbra, DEEC, University of Coimbra, 3030-290, Coimbra, Portugal
§FCT-UNL, 2829-516 Caparica, Portugal

***Abstract*—In the next generation wireless systems, where m-MIMO and mmWave should be combined with beamforming techniques, array optimization at receiver ensuring a good SINR level can become a very large problem. Meta-heuristics can be an adequate approach to solve this problem with a mild computational effort.**

In this work, different meta-heuristics are implemented and applied to this problem, with Differential Evolution presenting promising connection setup time results that expectedly would enable its use in m-MIMO systems.

***Index Terms*—BeamForming (BF), Multiuser, Meta-heuristics, Differential Evolution (DE)**

I. Introduction

The increasing use of devices, such as smartphones and tablets, to access media content as well as to establish high speed connections between them, brings new challenges to the next generation wireless systems. In fact, the evolution from 4th Generation (4G) to 5th Generation (5G) wireless systems is driven by the expected significant growth in user bit rates (a 10 to 100 times increase) and overall system throughput (about 1000 times increase) [1]. To accomplish these trends, new transmission techniques are required, the most promising ones being the use of millimeter Wave (mmWave) bands and massive Multiple-Input and Multiple-Output (m-MIMO) schemes [2], [3].

The adoption of mmWave transmission is mainly due to the considerable bandwidth available. Moreover, the small wavelength means small antennas, allowing small-sized transmitter and receivers with very high number antenna elements and therefore enabling m-MIMO implementations. Also, m-MIMO can be used to explore Spatial Multiplexing (SM) and BeamForming (BF) enabling the service of multiple users with high bit-rates while reducing interference and/or increasing coverage [4]. In addition, mmWave frequencies present considerable challenges in terms of propagation (high propagation free-space path losses, small diffraction effects and almost total absorption losses due to obstacles) and they have high reflection indexes, thus building an environment rich in multipath components [5]. Therefore, in systems using m-MIMO at mmWave bands with BF like the ones presented in [6], [7], determining the phase of hundreds of antenna elements at receiver arrays ensuring the quality of communication, i.e. with a high Signal-to-Interference plus Noise Ratio (SINR), can become a very large problem, even when channel estimation and direction of arrival (DOA) are known.

Due to the combinatorial characteristics and the dimension of the search space of this problem, meta-heuristics can be an adequate algorithmic approach to obtain good quality solution with a mild computational effort. In this work, three meta-heuristic approaches are implemented to solve it and their performances are compared. Then, the one displaying the best performance can be selected to create a module to be used in the receivers of complex systems presented in [6], [7], using information given by channel and DOA estimation modules.

This work was supported in part by Fundação para a Ciência e Tecnologia (FCT) under the projects UID/EEA/50008/2013 (GLANCES, PURE-5GNET) and UID/MULTI/00308/2013 and the PhD grant SFRH/BD/108522/2015.

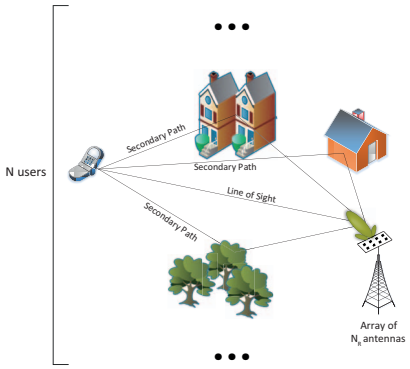


Fig. 1. Scheme of the multiuser m-MIMO system performing SM. In this system, the signal of each user suffers multipath propagation, resulting in several paths that are received by an array of N_R antennas.

Even with the use of meta-heuristic approaches, obtaining optimal (or near optimal) solutions to this problem can be time consuming, which do not fit real time requirements. Therefore, instead of pursuing the optimal solution, the algorithm can be tuned to find a solution that ensures a good quality communication, specified as a minimum SINR for a some given scenario. The solution can thus vary across different scenarios. As an example a scenario of a MIMO system using Quaternary Phase Shift Keying (QPSK) constellation and an Iterative Block Decision Feedback Equalization (IBDFE) receiver is considered, such as in [8], where the value for SINR is 6 dB to have a Bit Error Rate (BER) of 10^{-4} when SNR = 15 dB for uncoded situations and less than 3 dB for coded situations. As for most situations these values are much lower than the optimal one, which means that meta-heuristics can realistically obtain them, thus reducing the total runtime while ensuring good quality communication, this work follows this approach.

II. Problem Formulation

Let us consider a multiuser m-MIMO system performing SM, that can be represented by the scheme in Fig. 1.

The system have N users, where x_n, with $n = 1, \ldots, N$, is the time-domain signal transmitted by the n-th user, and each x_n signal is transmitted across a h_n channel that introduces multipath dispersion with L components (rays) expressed by

$$h_n(t) = \sum_{l=0}^{L-1} \alpha_l e^{j\varphi_l} \delta(t - \tau_l), \tag{1}$$

where α_l, φ_l and τ_l are the channel gain, the phase shift due to free space propagation and the delay of l-th ray, respectively.

At reception, a linear antenna array is used composed by N_R elements equally spaced to detect user u, resulting in an array factor that, according to [9], can be written as

$$F(\theta_{ul}) = \sum_{r=0}^{N_R-1} e^{(jR_r)} e^{\left(-j2\pi r \frac{d}{\lambda} sin\theta_{ul}\right)}, \tag{2}$$

978-1-5090-1701-0/16/$31.00 ©2016 IEEE

where θ_{ul} is the angle of arrival of l-th ray of the user u, R_r is the phase of r-th array element, d is the spacing between antennas and λ is the wavelength.

From the physical formulation of the problem, it can be seen that the phases of the elements of the array, R_r, are the decision variables of the optimization problem. They are responsible for the antenna orientation and, consequently, also for the reception power. Thus, the problem can be formulated as an optimization problem with the objective function

$$\max SINR_u = \max \frac{\sum_{l=0}^{L-1} \sigma_{ul}^2 |H_{ul} F(\theta_{ul})|^2}{\sum_{\substack{i=0 \\ i \neq u}}^{N-1} \sum_{l=0}^{L-1} \sigma_{il}^2 |H_{il} F(\theta_{il})|^2 + N_R \sigma_n^2}, \quad (3)$$

i.e. the maximization of the SINR for detection of user u. Note that in (3) the summation at the numerator represents the sum of the power of all the rays of the user u to be detected, and the double summation at the denominator represents the sum of the power of all the rays of all interfering users i, to which the noise power is added. The σ variables represent the variance of each component, with the indexes meaning: ul for rays of the user being detected, il for interfering users and their rays and n for noise. Variables H_{ul} and H_{il} are the DFT of the channel h_n (see Eq. (1)) for users u and i, respectively.

Once the number of antennas in m-MIMO systems reaches hundreds and their phases can assume any real value, the dimension of this problem escalates very quickly, which turns infeasible an exhaustive search of the solution space. Moreover, mmWaves have high reflection indexes, building an environment rich in multipath components. Thus, the use of meta-heuristics becomes an adequate approach to find good quality solutions with an acceptable computational effort coping with the combinatorial, nonlinear and irregular search space of this problem.

III. Meta-heuristics: A Brief Review

In the resolution of this problem, different population and non-population meta-heuristics were used, namely Genetic Algorithm (GA), Simulated Annealing (SA) and Differential Evolution (DE), in order to make a comparison of their performances. The solution encoding is the same in all implementations: a vector of real values with length N_R, where each value corresponds to the phase of the antenna with the same index. Also, the evaluation function is the objective function in all algorithms since there are no constraints. The algorithm parameters have been tuned through extensive experimentation.

A. Genetic Algorithm

The implementation of the GA [10] considered the choice of the following characteristics: solution encoding, evaluation function, initial population, genetic operators and selection mechanisms. In this subsection, characteristics that are specific of this implementation are described according to the pseudocode presented in algorithm 1 and the schemes of Figs. 2, 3 and 4.

As it is shown in the algorithm 1, the initial population $P(t)$ is random initialized with N_{parents} elements. Then, it origins an offspring population $C(t)$ of $N_{\text{offspring}}$ elements, with $N_{\text{offspring}} = 3 \times N_{\text{parents}}$, using the procedure illustrated in Fig. 2. In this procedure, the offspring population is generated by the genetic operators: crossover, mutation or a combination of both.

Crossover is presented in Fig. 3. It combines parent solutions according to a random binary mask. In one of the offspring, if the mask is 1 then the decision variable value of parent 1 is inserted in that position, else it is copied from parent 2. The opposite occurs in the other offspring. For example, in a problem with 4 antennas, $N_R = 4$, a solution may have antennas 1 and 2 of the array with a phase that can ensure a good SINR value, but the other two antennas have phases that make the array pointing in a direction with bad or not so good SINR values. On other hand, another solution may have antennas 3 and 4 with very promising phases, but the other two antennas make the SINR not so good. Therefore, by combining these two solutions using crossover a very good solution can be achieved. Moreover, weaker solutions can also be kept in the population helping to escape from local optima and avoid premature convergence.

However, if the solutions are only combined, they would be limited to the phases that are generated in the first population. Therefore, mutation enables to explore the search space. The mutation operator modifies one of the decision variables (phases of the antennas) of an existing solution as presented in Fig. 4. This mutation consists of the addition of a Gaussian variable with zero mean and variance that starts with a high value ($init\_var$) and decreases over time (dec_rate). This allows a wider search at the beginning, helping to escape from local optima and then narrowing down the search, improving the chance of discovering better solutions close to the best ones found. The combination of crossover and mutation operators contributes to further explore the search space.

After obtaining the offspring population, the selection of the parents for the next generation is done following the scheme illustrated

Algorithm 1 Genetic Algorithm

```
k ← 0;
while k ≤ N_attempts do
    t ← 0;
    Random initialization of P(t);
    Evaluate P(t);
    best(t) ← the best solution of P(t);
    while stop condition is not reached do
        t ← t + 1;
        Generate C(t) by crossover and mutation;        ▷ See Fig. 2
        Evaluate C(t);
        Select P(t);                                    ▷ See Fig. 2
        Insert a new random solution in P(t);
        best(t) ← the best solution of P(t);
    end while
    k ← k + 1;
end while
```

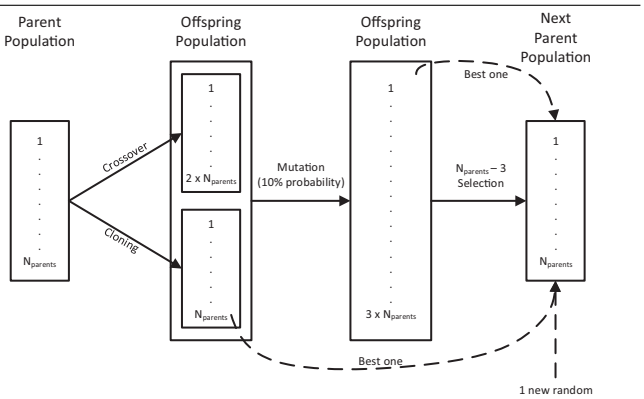


Fig. 2. Scheme of the evolution of a population using the genetic operators.

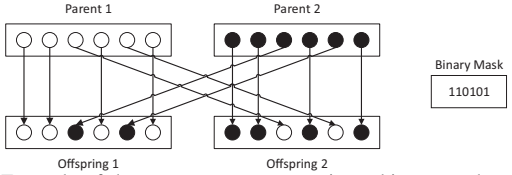


Fig. 3. Example of the crossover operator using a binary mask.

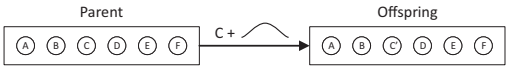


Fig. 4. Example of mutation of one decision variable.

in Fig. 2. Elitism is used to ensure that the best solution of the offspring population passes to the next generation as well as the best solution found until this iteration. In order to increase diversity, a new random solution is also inserted into the population. The remaining $N_{\text{parents}} - 3$ solutions are selected by binary tournament between two solutions randomly selected.

The process repeats until the stop condition is reached. This is a maximum number of iterations without improvement of the best solution, $iter\_no\_improv$, or a maximum number of generations, max_gen. The algorithm runs N_{attempts} independent attempts to be statistically characterized.

The parameter values used in this implementation are:

- $N_{\text{parents}} = 30$;
- $N_{\text{offspring}}$=3×N_{parents};
- $init\_var = 3$;
- $dec\_rate = 0.2$ every 100 iterations;
- $iter\_no\_improv$=200;
- $max\_gen = 2000$.

B. Simulated Annealing

Simulated Annealing [11] uses, in general, only one initial solution and searches the solution neighborhood to find better solutions, allowing the acceptance of worse solutions in order to avoid to be stalled in local optima.

In this implementation, the initial solution is randomly chosen as in GA. The neighborhood encompasses solutions obtained by changing a single variable. This change can be of $q/2$ or q, with q being a phase shift in degrees of one of the antennas. A small example, with only two decision variables, is presented in Fig. 5.

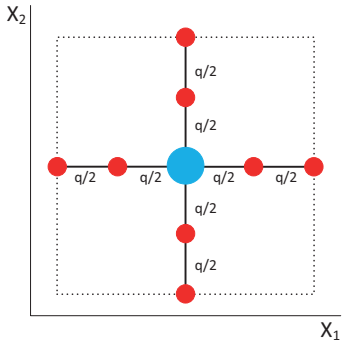


Fig. 5. Example of a neighborhood defined by a hypercube of edge $2q$.

The search of the neighborhood compares the current solution, v_c, with the best solution within the neighborhood, v_n. If v_c is better then it is selected, otherwise it can be selected according the following acceptance probability function:

$$p(v_c, v_n, T) = \frac{1}{1 + e^{\frac{eval(v_c) - eval(v_n)}{T}}}, \tag{4}$$

where $eval(i)$ is the value of the evaluation function of solution i and T is a control parameter that varies according the cooling rate, r. After searching the neighborhood, the algorithm checks whether the best solution within the neighborhood is better than the best one found until that iteration.

The algorithm is run N_{attempts} independent attempts. The following parameters have been tuned through experimentation:

- $q = 2^{\circ}$;
- $T_{max} = 10000$;
- $T_{min} = 0.01$;
- $r = 0.001$;

C. Differential Evolution

Differential Evolution [12] is a populational algorithm that uses a differential mutation operator. This operator creates mutant solutions adding the difference of two existing solutions to another solution.

In DE the parameters to be defined are: the choice of initial population, population size (Pop_size) scaling factor (F) the parameter (C) and the stop condition. As in GA, the stop condition is $iter\_no\_improv$ or max_gen. The scaling factor, F, controls the weight that the difference of two solutions has in the mutant solution; the parameter C controls the fraction of values in the new solution that are copied from the mutant one. According to C values, the behavior of DE can be controlled. When C is close to 1, solutions have a higher diversity with a wider search. When C is close to 0, search is narrowed down.

The initial population is built as in GA, since both use a population. The remaining parameters have been tuned through experimentation:

- $Pop\_size = 40$;
- $F = 0.8$;
- $C_{max} = 1$;
- $C_{min} = 0$;
- $C_{dec} = 0.1$ every 20 iterations;
- $iter\_no\_improv$=100;
- $max\_gen = 500$.

As in the previous algorithms, the best solution found is kept in memory and DE runs N_{attempts} independent attempts.

IV. Results

In this section, results obtained with the meta-heuristics developed are presented. It is divided in three subsections. In the first one, a simple system with only two users is considered, each one with two multipath components, with angles and power of each component known, and arrays of 4 antennas, $N_R = 4$, at receiver. In the second one, the same system is used, with only two users, each one with two multipath components, and $N_R = 4$, but 50 instances with angles and power randomly generated are tested. In the last one, a more realistic problem with 16 users is simulated, each one with four multipath components and $N_R = 64$. With the exception of these characteristics, the remaining ones are the same in all cases. Also note that, besides the specific stop condition of each algorithm, in all cases the minimum SINR level required is 6 dB in order to meet the requirements presented in [8]

Since the channel estimation is given by a separate module, for the sake of simplicity, the magnitude of each $\mathbf{H}_k$ signal is considered unitary and there are no phase shifts. As this approach has been developed for systems using m-MIMO at mmWave bands with BF, arrays have elements spaced by $\lambda/2$, whit $\lambda = 1mm$. Also, it is considered that base stations use sectoring, with antenna arrays covering 180° of the space. For calculation of noise power, Signal-to-Noise Ratio (SNR) of 15 dB is considered and the signal power at reception is considered to be the sum of the power of all components. Only if all components have the same angle, the real signal power is lower. However, this ensures that the worst scenario for noise power is considered.

In order to assess the results, polar diagrams with the normalized radiation pattern of the antenna array are used, such as in Fig. 7. The blue solid curved line represents the radiation pattern of the array, the red solid line represent the rays of the user to be detected and the dashed black lines represent the interfering rays. The length of each line is proportional to the component power.

All simulations have been done in the same machine (Intel i7-4710HQ @ 2.50Ghz) using MATLAB, guaranteeing a fair comparison, and results have been obtained through 100 independent attempts, $N_{\text{attempts}} = 100$.

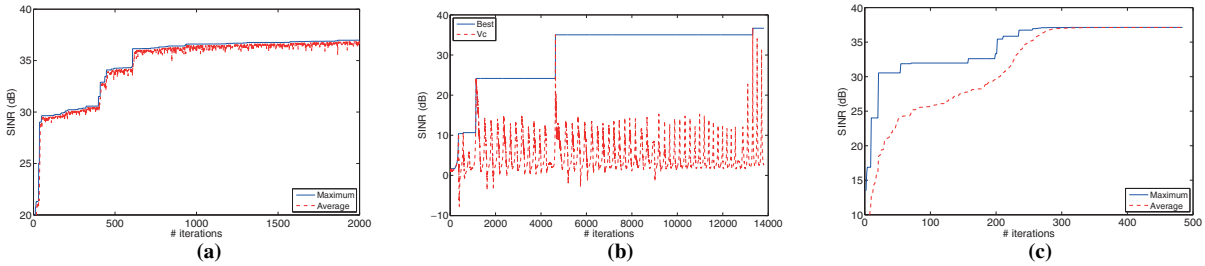


Fig. 6. Evolution of the evaluation function values during one independent attempt of (a) GA; (b) SA and (c) DE.

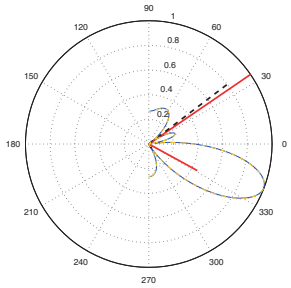


Fig. 7. Normalized radiation pattern of best solutions found by the algorithms (solid straight red: user to detect; dashed straight black: interference; solid curved blue: DE; dashed curved yellow: GA; dotted curved pink: SA).

A. A simple example

A simple system is considered with only two users, each one with two multipath components. This example aims a comparison of the algorithms in order to chose the one with the best performance to be used in next simulations, while presenting a simple problem that gives a better understanding of the problem.

For the sake of simplicity, the rays of the user to be detected are located at 34.38° and -28.65° with variance equal to 1 and 0.4483, respectively. For the interfering user, the rays are located at -57.87° and 37.24° with variance 0.0621 and 0.8276, respectively. These choices have been made in order to have the strongest ray to be detected affected by a strong interference, while the weakest ray almost does not have interference. In this way, the solution becomes almost intuitive and it is easier to validate the results. Note that the analysis of Fig. 7 makes expectable that the radiation pattern evolves in direction of the red ray located at -28.65°. Although this ray has less power than the other red ray, it does not have strong interferences close to it, becoming more suitable for detection.

Figs. 6a - 6c help to understand the behavior of the algorithms. In GA and DE, it is expected that the best value increases over iterations as well as the average value, once the population should converge to the optimal solution. Although this is seen, in GA the average value sometimes decreases, which means that the population has high diversity, allowing the search of a wider space and reducing the probability of being stalled in local optima. In SA, the value of the current solution is often far from the best value found so far due to the nature of the algorithm that admits solutions with worse values, possibly having great variation in its value. Moreover, although the solutions found by each algorithm present similar SINR values, the number of iterations to reach them have a great variation.

Fig. 7 shows the radiation pattern of the best solution found by each algorithm. All solutions point in the same direction and they are very similar. Still, it is possible to recognize some differences mainly in the side lobes, which result in the differences shown in table I.

Although these differences are very small, they result in significant differences in the evaluation function values. The best values of the evaluation function are displayed in table I, in which small differences in radiation patterns lead to a difference of about 0.5 dB

TABLE I. Results attained by the algorithms in 100 independent attempts.

		Worst	Best	Avg	Std	Median	Avg. Calls
GA	*Eval. Func. (dB)*	30.0339	37.1228	35.5353	1.4927	35.8862	85225.46
	Runtime (s)	0.7339	5.3445	3.5181	1.5153	3.5195	
SA	*Eval. Func. (dB)*	15.0863	36.6617	26.3436	7.2359	28.0752	221073.00
	Runtime (s)	7.0920	7.3713	7.1431	0.0371	7.1383	
DE	*Eval. Func. (dB)*	27.9808	37.1319	36.5954	1.6128	37.1314	14265.20
	Runtime (s)	0.2135	1.0039	0.7435	0.2107	0.8097	

TABLE II. Evaluation function values, in dB, attained by DE in 100 independent attempts for 50 random instances.

	Worst	Best	Avg	Std	Median	Avg. Calls
Min	6.2671	7.3920	7.3092	0.0224	7.3197	12311.60
Max	42.1749	45.5199	44.1938	7.8652	45.3821	16405.60
Avg.	27.2390	38.3170	37.3006	2.4658	38.2836	14198.49

in SINR value. This explains the big variations in the evaluation function values even in different attempts of the same meta-heuristic. Notwithstanding, all the values fit the minimum requirement of SINR = 6 dB.

It is possible to see that DE achieves the best performance considering the trade-off between runtime and the value of the objective function. It reaches the best values in average, with a median very close to the maximum value found with much less function calls, resulting in a lower runtime. As DE has a consistently better performance compared with the other meta-heuristics, it is selected to solve the problem presented in this article. These results show that DE is suitable for this specific problem. In next subsections, it is tested in more demanding instances.

B. Testing different instances

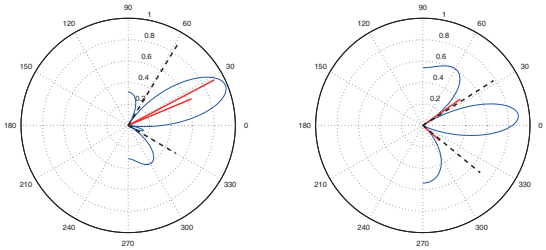
The goal of this subsection is to show that the DE algorithm has a good performance in a diversified range of instances, being suitable to attain SINR requirements with a low computational effort. For this purpose, 50 random instances were generated. As in the previous case, two users are considered, each one with two multipath components, and $N_R = 4$.

Results are summarized in tables II and III. Once more, there are big differences between instances and within the same instance. However, the minimum SINR value of 6 dB is always attained and the number of evaluation function calls, in average, remains almost the same. This results in a similar runtime even for less favorable instances, reinforcing that DE is suitable for this problem.

Two different instances are analyzed in more detail:: one of the most favorable ones, with strong rays close to each other, without near interferences (Fig. 8a), and one of the less favorable ones, with disperse weak rays, with strong interferences near them (Fig. 8b). In both instances arrays are oriented in order to have zeros in the angles of interference. However, in the less favorable instance the radiation pattern is disperse over the plane without presenting directivity. As this is a small problem and the main goal is to maximize SINR, DE

TABLE III. Runtime values, in seconds, of DE in 100 independent attempts for 50 random instances.

	Min	Max	Avg	Std	Median
Min	0.2083	1.0405	0.6571	0.1163	0.6768
Max	0.5571	1.5509	0.9103	0.2905	0.9080
Avg.	0.2460	1.1278	0.7627	0.2170	0.8134



(a) **(b)**

Fig. 8. Normalized radiation pattern of best solution found for one of the (a) most favorable instances; (b) less favorable instances.

only tries a fast solution that solves the problem without concerning directivity, and the easiest way to accomplish this is to use the space where there are no interferences. Nevertheless, in a real problem there are much more interference, and DE can only achieve good results if it takes advantage of BF directivity. Thus, in the next subsection a realistic problem is studied.

C. A realistic problem

Since DE presents a good performance results in simple problems, its behavior should be also assessed in a realistic system. Hence, a problem with 16 random users has been created, each user with four multipath components and $N_R = 64$. Now, there are more antennas which can improve directivity, but there is much more interference.

Furthermore, in order to keep low runtime values (few seconds), the stop condition is achieving SINR of 6 dB. This becomes mandatory due to the dimension of the search space. DE is run 100 independent times for each user, recording the runtime values to achieve SINR values of 3 dB and 6 dB, the bounds previously mentioned for uncoded and coded situations, respectively. Also, population is initialized with 20 elements only, which reduces runtime. When the population loses diversity, half of the population is replaced by new random solutions. However, this mechanism is very rarely used.

The results in table IV show that DE does not always reach SINR = 6 dB and even SINR = 3 dB, seldom leading to very low values. However, this occurs for a very small number of cases. In fact, in 96.38% of the attempts, the algorithm reaches SINR = 6 dB, with most of unsuccessful attempts being related with only two users that are hard to detect, such as the one illustrated in Fig. 9. If the SINR bound is 3 dB, the average success rate per user increases to 99.63%, which represents about 98∼99% for the users that are hard to detect. Moreover, these results are attained with less evaluation function calls than in previous subsections. However, as the problem dimension grows, the complexity of the evaluation function also grows.

TABLE IV. Evaluation function values, in dB, attained by DE in 100 independent attempts for a realistic problem.

	Worst	**Best**	**Avg**	**Std**	**Median**	**Avg. Calls**
Min	3.5180	6.1733	5.8848	0.1319	6.0332	4984.80
Max	6.0235	7.5288	6.2964	0.4914	6.2176	23906.40
Avg.	5.1838	6.9539	6.1647	0.2679	6.1386	10345.00

Compared with the previous subsections, the results in table V show that runtime significantly increases, which is expected due to the increased complexity of the problem. Although these values are not yet suitable for real time applications, DE still presents interesting features that can be explored. It should be noted that these results are obtained in a framework that is not suitable for this kind of application. With a more powerful CPU using all cores (MATLAB only uses one core by default) or parallel computing platforms such as FPGAs, runtime can become much lower.

Moreover, the array optimization is made, in general, at the setup time, which lasts over than a second. Also, the system can start a connection with the first solutions found, while DE keeps improving solutions. This feature allows the system to be adapted in real time.

TABLE V. Runtime values, in seconds, of DE in 100 independent attempts for a realistic problem.

		Min	**Max**	**Avg**	**Std**	**Median**
SINR≥3 dB	*Min*	0.7630	3.5410	1.9279	0.5645	1.9121
	Max	6.1966	22.4146	13.4234	4.9479	12.2533
	Avg.	2.0629	15.2813	4.9107	2.1646	4.5549
SINR≥6 dB	*Min*	1.4585	5.8643	3.5188	1.0604	3.4132
	Max	9.7473	22.4146	16.7721	3.8761	16.9585
	Avg.	3.4524	16.3327	7.2162	2.2510	6.9962

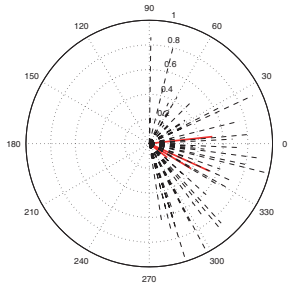


Fig. 9. Normalized power and location of users in a situation in which is very difficult to optimize the array considering the user to be detected.

V. Conclusion and Future Work

In this work, three meta-heuristics to perform array optimization in a m-MIMO system using BeamForming in mmWave bands were presented. Results show that the use of meta-heuristics is adequate to cope with the combinatorial and irregular nature of the search space. In our experiment, population meta-heuristics obtained better results. However, more research is needed in a real time context. DE revealed the most promising algorithm achieving good SINR values while keeping runtime lower than few seconds for all situations tested.

In future works, the DE approach should be improved and further tailored for the characteristics of this problem to decrease runtime and obtaining good performance solutions. Also, it must be tested in other situations without slow shift of mobile terminals and/or multipath components with few changes over time. Moreover, situations where a ray disappear should be assessed, in order to prevent the fact that mmWaves do not suffer diffraction.

References

[1] S. Rangan *et al.*, "Millimeter-Wave Cellular Wireless Networks: Potentials and Challenges," *Proceedings of the IEEE*, vol. 102, no. 3, pp. 366–385, Mar 2014.

[2] F. Boccardi *et al.*, "Five disruptive technology directions for 5G," *Commun. Mag., IEEE*, vol. 52, no. 2, pp. 74–80, Feb 2014.

[3] S. Sun *et al.*, "MIMO for millimeter-wave wireless communications: beamforming, spatial multiplexing, or both?" *Commun. Mag., IEEE*, vol. 52, no. 12, pp. 110–121, Dec 2014.

[4] E. Biglieri *et al.*, *MIMO Wireless Communications*. New York, NY, USA: Cambridge University Press, 2007.

[5] T. Rappaport *et al.*, *Millimeter Wave Wireless Communications*. Pearson Education, 2014.

[6] R. Dinis *et al.*, "A Massive MIMO Architecture for Highly Efficient mm-Wave Communications with Saturated Amplifiers," in *International Conf. on Electronics, Information, and Communication*, Jan 2015.

[7] R. Dinis *et al.*, "A multi-antenna technique for mm-wave communications with large constellations and strongly nonlinear amplifiers," in *Microwave Conf. (GeMiC), 2015 German*, Mar 2015, pp. 284–287.

[8] J. Gante *et al.*, "Towards an enhanced frequency reuse: Base station cooperation with turbo frequency domain receivers," in *Commun. (ICC), 2015 IEEE International Conf. on*, Jun 2015, pp. 2786–2790.

[9] P. Silva *et al.*, "Adaptive transceiver - antenna techniques for mobile broadband communications at millimetre - wave frequencies," in *Intelligent Signal Processing and Communication Systems*, 1996.

[10] Z. Michalewicz and D. B. Fogel, *How to solve it : modern heuristics*. New York: Springer, 2002.

[11] S. Kirkpatrick *et al.*, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[12] R. Storn and K. Price, "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *J. of Global Optimization*, vol. 11, no. 4, pp. 341–359, Dec. 1997.