mathematics

MDPI

Article

Optimal Power Allocation Based on Metaheuristic Algorithms in Wireless Network

Qiushi Sun *, Haitao Wu and Ovanes Petrosian

Faculty of Applied Mathematics and Control Processes, Saint Petersburg State University, 198504 Saint Petersburg, Russia
* Correspondence: st059656@student.spbu.ru

Abstract: An optimal power allocation is a fundamental challenge for massive multiple-input–multiple-output (MIMO) systems because the power allocation should be acclimated to time-varying channels and heavy traffic conditions throughout the communication network. Although massive model-driven algorithms have been employed to solve this issue, most of them require analytically tractable mathematical models and have a high computational complexity. This paper considers the metaheuristic algorithms for the power allocation issue. A series of state-of-the-art stochastic algorithms are compared with the benchmark algorithm on network scales. The simulation results demonstrate the superiority of the proposed algorithms against the conventional benchmark algorithms.

Keywords: power allocation; interference optimization; metaheuristic; MIMO; interference management

MSC: 94-10

check for updates

Citation: Sun, Q.; Wu, H.; Petrosian, O. Optimal Power Allocation Based on Metaheuristic Algorithms In Wireless Network. *Mathematics* **2022**, *10*, 3336. https://doi.org/10.3390/math10183336

Academic Editor: David Barilla

Received: 19 July 2022
Accepted: 8 September 2022
Published: 14 September 2022

1. Introduction

The rapid growth in the number of smart devices has led to an explosion in demand for mobile multimedia services, and traditional cellular networks are facing significant challenges. The shrinking communication distance between devices due to the dense population in large cities has created new communication opportunities. The research on multiple-input and multiple-output (MIMO) systems stands out as a critical technology candidate for 5G. With the exponentially increasing traffic rates and user density, wireless communication cellular networks need more system capacity and spectrum effectiveness. The density of access points (APs) increases with the massive number of terminals connecting to the network. Dense small-cell deployment is the most popular schema to accommodate the spectrum [1]. The communication network is filled with crowded signals because of dense APs and small cells. Hence, mitigating intra-cell and inter-cell interference is essential to enhance the overall capability of the cellular network system [2].

Consequently, the optimal power and wireless resource control in cellular communication networks have received considerable critical attention in recent decades. To understand how well the power allocation schema achieves in an industry communication environment with limited computational resources, it is also essential to study the computational complexity of this issue. It is proved that the utility maximization problem in the MIMO system is non-deterministic polynomial-time hardness (NP-hard) for a large class of objective functions [3]. In addition, the power allocation issue investigated by this paper is nonlinear, non-convex, and NP-hard.

Massive model-driven approaches have been proposed to solve optimal power control issues with interference management. For instance, fractional programming (FP)-proposed quadratic transform can significantly facilitate the optimization involving ratios by recasting the original non-convex problem as a sequence of convex problems [4]; the weighted minimum mean square error (WMMSE) uses the local channel knowledge and converges to a stationary point of the weighted sum-rate maximization problem [5]. A distributed iterative algorithm is proposed in [6] to maximize the weighted sum rate of a wireless cellular network based on coordinated scheduling and discrete power control. A heuristic joint proportionally power spectrum adaptation algorithm is proposed in [7] that coordinates multiple base stations to optimize the system utility. The power allocation mechanism is proposed based on the unique Nash equilibrium for cellular networks, and the optimal solution maximizes the utility function [8]. Meanwhile, there has been extensive development of machine learning (ML) algorithms for power control in the wireless communication system [9]. Through supervised learning, [10] use the neural network to approximate the given objective algorithms. The deep neural network is trained to learn the map between geographic locations and policies, and then used to predict the optimal power allocation [11]. ML methods are model free and data driven, the optimal solution is obtained by data learning rather than systems-oriented modeling and analysis, and the dataset's quality limits the performance of these methods.

Most of the existing algorithms revolved around partial optimal or heuristic algorithms. The excellent performance of these methods can be observed through simulation experiments, but it is still challenging to implement them in industrial scenarios. There are mainly two components:

- These algorithms highly rely on the analytical and tractable mathematical model. However, it is hard to construct a perfect mathematical model in practical implementation scenarios because of the specific user distribution and geographical environment.
- The computational complexities of these approaches are pretty high, especially in large-scale MIMO systems. Therefore, it becomes impractical to implement these algorithms with interference optimization.

In recent years, stochastic algorithms have been developed in wireless communications fields. The metaheuristics algorithms are the most widely used random search methods to solve intractable problems [12]. Metaheuristic algorithms do not require a perfect mathematical model and are compliant with the optimization in practical implementation. Several state-of-art approaches to solving optimal power allocation issues in a MIMO system are addressed. Adaptive particle swarm optimization (PSO) addressed the issue of resource allocation in a wireless sensor network [13]. A modified single-level artificial bee colony (ABC) was provided to handle the resource allocation problem for an underlay D2D communication network [14].

In this paper, we investigate the communication network in an interfering multiple-access channel (IMAC) scenario. We concentrate on the global optimization for the power allocation issues in the MIMO system to maximize the overall sum-rate SINR. This issue can be classified as a static optimization problem which is an ordinary multivariate function. Hence, metaheuristic algorithms are implemented to handle intractable issues. We review a wide range of metaheuristic algorithms and improved an open-source simulation environment [15] as the black-box simulator, which simulates the set of allocated power as a given input and the sum-rate SINR as output. Furthermore, we provide a statistical and significant comparison of the performance of the metaheuristic algorithms and conventional optimization techniques. The simulation results indicate which algorithm outperforms the other metaheuristic and benchmark algorithms.

To find an efficient approach for this optimization problem, we consider various competitions in global optimization to select the best algorithms from feasible techniques. Moreover, the competitive process generates novel ideas which can be developed into practical solutions. We consider the Special Session and Competition on Large-Scale Global Optimization in the past ten years and concentrate on different mathematical optimization methods with one objective [16]. We select nine metaheuristics algorithms from the winning rank, such as the artificial bee colony (ABC), self-adaptive differential evolution (jDE and iDE), particle swarm optimization generational (GPSO), extended ant colony optimization (EACO), differential evolution (DE), particle swarm optimization (PSO), simulated annealing (SA), monotonic basin hopping (MBH), and covariance matrix adaptation–evolution strategy (CMA-ES). In addition, the contribution of our research work is twofold:

- We present an optimal power allocation problem under interference management and propose model-free metaheuristic approaches.
- Through numerical experiments on simulated communication networks and wireless channels, we substantiate the effectiveness and flexibility of the proposed metaheuristic methods and search for the potentially best algorithms for this open research problem.

The remainder of the paper is organized as follows. Section 2 provides a detailed mathematical model of the optimal power allocation problem, and Section 3 describes the modified metaheuristic algorithms for solving the PA. Section 4 includes a numerical experiment with an illustrative result. Section 5 provides the conclusions and prospects for further research.

2. Mathematical Model

We investigate downlink transmission in a homogeneous wireless cellular network with IMAC. In the wireless cellular network, there is only one base station in the middle of the cell, and the organization of each cell is approximately the same. Thus, the communication model in the cell is essentially the same. Similar to [17], the homogeneous wireless network consists of N cells, and a base station (BS) simultaneously serves K users (UE) by sharing frequency bands at the center of each cell. A simple example of the communication network with 3 cells is shown in Figure 1. The notation $g_{n,j,k}$ represents the independent channel coefficient of the communication link from the n-th BS to k-th user in cell j at time slot t, which is expressed as follows:

$$g_{n,j,k} = |h^t_{n,j,k}|^2 \beta_{n,j,k}, \tag{1}$$

where $h^t_{n,j,k}$ denotes the complex small-scale fading component and $\beta_{n,j,k}$ denotes the large-scale fading element which consider both shadow fading and geometric attenuation. Therefore, the downlink signal-to-interference-plus-noise ratio (SINR) received by user k is formulated as

$$\gamma^t_{n,k} = \frac{g^t_{n,n,k} p^t_{n,k}}{\sum_{k' \neq k} g^t_{n,n,k} p^t_{n,k'} + \sum_{n' \in D_n} g^t_{n',n,k} \sum_j p^t_{n',j} + \sigma^2}, \tag{2}$$

where D_n is set of interference cells of the n-th cell, $p^t_{n,k}$ denotes the transmit power for the k-th user connected with the n-th BS at time slot t, and σ^2 presents the variance of additive white Gaussian noise. $\sum_{k' \neq k} g^t_{n,n,k} p^t_{n,k'}$ and $\sum_{n' \in D_n} g^t_{n',n,k} \sum_j p^t_{n',j}$ express the intra-cell and inter-cell interference power, respectively. Considering normalized bandwidth, the SINR of link $l_{n,k}$ between n-th cell and k-th user at time slot t is expressed as

$$C^t_{n,k} = \log_2(1 + \gamma^t_{n,k}) \tag{3}$$

The objective function is to maximize the sum-rate utility of the whole cellular network under the power constraint. The problem is formulated as

$$\max_{\boldsymbol{p}^t} C(\boldsymbol{g}^t, \boldsymbol{p}^t) \text{s.t.} \qquad 0 \leq p^t_{n,k} \leq P_{\max}, \quad \forall n, k, \tag{4}$$

where $P_{\max}$ is the maximum transmit power; $\boldsymbol{p}^t = \{p^t_{n,k} | \forall n, k\}$ is the set of allocated power; $\boldsymbol{g}^t = \{g^t_{n',n,k} | \forall n', n, k\}$ is the set of channel gain. The sum-rate utility is given as

$$C(\boldsymbol{g}^t, \boldsymbol{p}^t) = \sum_{n,k} C^t_{n,k} \tag{5}$$

For model-driven approaches, it is generally hard to evaluate the performance gap from the optimal solution, and practical implementation is limited because of the imperfect mathematical model. Furthermore, it is hard to adapt the model-based method to heterogeneous cellular networks because of the imperfect mathematical model in real communication scenarios. Thus, model-free metaheuristic algorithms are discussed in the following section.

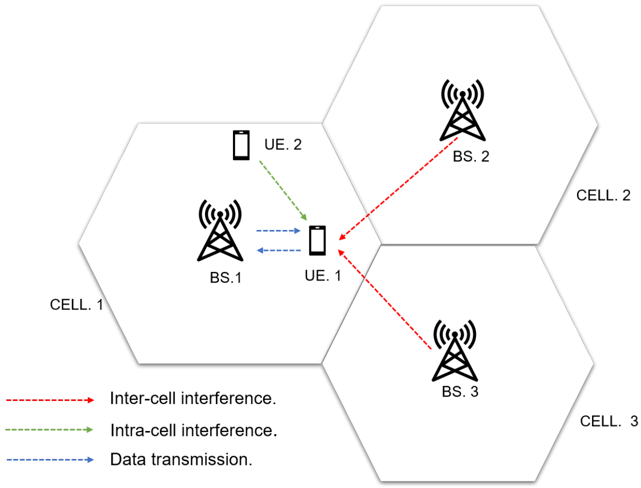


Figure 1. An illustrative example of a cellular network with 3 cells.

3. Metaheuristic Approach

In this framework, we consider nine metaheuristic algorithms from four primary types. A brief description of these algorithms is as follows:

3.1. Swarm Intelligence Algorithms

3.1.1. ABC

The artificial bee colony (ABC) algorithm is a stochastic search technique based on the intelligent foraging behavior of honey bee swarms [18]. In this algorithm, each candidate solution indicates the location of the food source in the search space, and the quality of the food source is employed as a fitness evaluator.

The model involves three essential elements: employed bees, onlookers, and food sources. The amount of employee bees is equal to the food sources. Employed bees depart from the hive to search for a food source and collect information about the quality of the other food sources in the neighborhoods of discovered location. Once back in the hive, they transmit information about the explored food source to the onlookers. Onlookers evaluate a new location from the information provided by the employed bees according to the selection probability of quality and prefer the food source with high fitness value. Onlooker becomes an employed bee when it selects a new food source to explore. The employed bee switches to the scout bee and randomly searches for new food resources in the search space when its explored food source is abandoned. This process is repeated until the optimal food source is found.

Advantages of ABC: It requires few parameters, performs robustly, converges fast, and is highly flexible. Disadvantages of ABC: It may converge prematurely in the phase of its search, and the classification accuracy of the best value it obtains may not meet the requirement [19].

3.1.2. PSO

Particle swarm optimization (PSO) is a swarm intelligence technique. The initial idea of PSO is inspired by the population behavior of bird flocking and fish schooling. PSO and evolutional strategic techniques have many standard features. This algorithm simulates the behavior of members' information interaction and collaboration. The difference with the genetic algorithm is that PSO does not require evolution operators, such as crossover and mutation. In the model, there is a population of candidate solutions called particles. These particles move around in the search space over their position and velocity. Each particle's movement is guided toward the best-explored positions in the search space, updated as other particles find better positions. This is expected to move the swarm toward the best solutions. The PSO process can be formulated as:

$$v_{id}^{t+1} = v_{id}^{t} + c_1 \times rand(0,1) \times (p_{id}^{t} - x_{id}^{t}) + c_2 \times rand(0,1) \times (p_{gd}^{t} - x_{id}^{t}) \quad (6)$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t} \quad (7)$$

where x_{id}^{t} and x_{id}^{t} represent the position and velocity of each particle, the parameter d is the population size, i is the index of the each particle, and t is the number of iterations. c_1 and c_2 are learning factors. p_i represents value explored by ith particle, p_g represents value explored by neighbors of the ith particle. PSO can be implemented as Algorithm 1:

Algorithm 1 Particle Swarm Optimization [20]

Require: Generate initial individual
Ensure: The best vector
  while Termination condition not met do **do**
    for Each particle x with position p_i **do**
      Calculate fitness value
      if fitness value is greater than the current best value p_{best} **then**
        Set current best value as p_{best}
      end if
    end for
    Select the particle with the overall best fitness value and set it as g_{best}
    for Each particle **do**
      Calculate particle velocity
      Update position of particle
    end for
  end while

Advantages of PSO: It has a simple calculation without overlapping and mutation. Disadvantages of PSO: It may fall into local optimum in high-dimensional space and has a low convergence rate in the iteration [21].

3.1.3. GPSO

Particle swarm optimization generational (GPSO) is a variant of the standard PSO algorithm. In the PSO algorithm, velocity is one of the most significant parameters; if particles' velocity in the swarm is updated effectively, no search effort will be wasted by searching in the wrong directions. The procedure for PSO is to move the particle to search for the positions of optimal solutions. The velocity at which the particles change positions is usually adjusted by multiplying the velocity by a factor. Unlike the standard algorithm, the velocity is first calculated for all particles; then, the position is updated.

GPSO can handle stochastic optimization problems according to iterative random seed schema. However, it is not suitable for multi-objective problems [22].

3.1.4. EACO

Ant colony optimization (ACO) is a classical bio-inspired technique based on foraging behavior of natural ants. The ACO algorithm simulates the process of a colony of ants seeking for the shortest path from nest to the food source. In the model, a group of simulation agents imitate the foraging behavior of natural ants to search for the minimum value of function. Each agent departs from the nest in search of food source and arrives at the nest as the end of the trial. Each agent leaves a marker called the pheromone on the path they take in search of food source. The pheromone concentration on each path is used to evaluate the distance of the path and the quality of the food source. The information implied by the pheromone on the path plays an important role in the subsequent agent's selection to the path. The higher the fitness value of the path evaluation, the higher the probability that the path be accepted. Extended ACO improves the original algorithm by using the multi-kernel Gaussian distribution based on three parameters which are computed depending on the quality of each previous solution [17]. The objective function values are ranked through an oracle penalty method. Advantages of EACO: Its parallel process can search solutions independently and simultaneously [23]. Disadvantages of EACO: Its probability distribution iteration changes and the convergence time is not stable.

3.2. Differential Evolution Algorithms

3.2.1. DE

The differential evolution (DE) algorithm is one of the most popular techniques for continuous optimization problems. DE is based on the evolution strategy but not inspired by the natural paradigm like common ones. It is proposed to search for the minimum value of non-differentiable and nonlinear continuous functions. Classical DE has two significant features to be adjusted: the learning strategy and the control parameters. The learning strategy comprises the primary type of operators in genetic algorithms, such as mutation, crossover, and selection. A basic variant of the DE algorithm works by having a population of candidate solutions. These agents are moved around in the search space to combine the positions of existing agents from the population. If the value of an agent's new position is improved, it is accepted and forms part of the population. It is excepted but not guaranteed that a global optimal solution will eventually be found.

In the mutation, a mutant vector is generated as formula:

$$v_{i,G+1} = x_{r,G} + F(x_{r,G} - x_{r,G}) \tag{8}$$

where F represent the scaling factor, G is the number of iterations. x_{r1}, x_{r2}, and x_{r3} are random searched vectors in current iteration. In the crossover, a trail vector is produced by combining the parent vector with a mutated vector.

$$u_{i,G+1} = \{ \begin{array}{c} v_{i,G+1}\ if\ rand_j \leqslant Cr \\ x_{r,G}\ if\ rand_j > Cr \end{array} . \tag{9}$$

where Cr represents the crossover rate. j is random number in the resulting array. v_i is current best value, x_i is best searched value. DE can be implemented as Algorithm 2:

Algorithm 2 Differential Evolution [24]

Require: Generate initial population;
Ensure: The best vector;
  while Termination condition not met do **do**
    for Each solution x_i in population **do**
      Generate new solution s_i;
      if fitness(s_i) $\geq$ fitness(x_i) **then**
        Retain s_i in population;
      else
        Retain x_i in population;
      end if
    end for
    Evaluate fitness of the new population
    Update the best solution
  end while
  Return best solution

Advantages of DE: It can handle optimization problems with high computational complexity. Disadvantages of DE: It requires parameter tuning and its convergence is not stable [25].

3.2.2. jDE&iDE

These are two different variants of the DE algorithm based on the mechanism of self-adaptation. The learning strategies and control parameters involved in the standard DE algorithm highly rely on the specific optimization problem. This process may cost amount of time to select the strategy and adjust the parameters to make the model have a good performance. Many different proposals have been made to self-adapt both the CR and the F parameters of the original differential evolution algorithm. There are many different proposals that have been proved to adapt the CR and F parameters. The first variant (jDE) does not use the DE operators to update parameters F and CR; the procedure is more like parameter control rather than self-adaptation [26]. The second variant (iDE) uses a variation of the selected DE operator to update CR and F parameters for each individual [27].

3.3. Random Search Algorithms

3.3.1. SA

Simulated annealing is a stochastic global search optimization technique to search. This algorithm emulates the statistical annealing procedure of the crystals growing to reach the global optimal internal energy configuration [28]. The annealing process works by first exciting the atoms in the material at a high temperature. This step can push atoms to heat up and accelerate their motion. The next step is to slowly cool down the temperature to reduce their excitability, making atoms convert into a more stable configuration. The essential component to implementing this simulated annealing process is to initialize a random solution in the neighborhood of the current optimal solution and evaluate the objective functions. Once the fitness value of the cost function is smaller than its current best value, the solution is accepted, and the new best fitness value is updated. Once the fitness value is higher than the current best value, the point is accepted or rejected with probability. A parameter temperature is introduced to calculate the probability. In the cooling schedule, the temperature is reduced with the acceptance probability converging to zero. The whole annealing process is terminated after a large number of trials. This strategy avoids being trapped in the local optimal solution.

Advantages of SA: It can handle the problem with arbitrary systems and cost functions. Disadvantages of SA: It requires parameter tuning and is possible to be trapped into local minima.

3.3.2. MBH

Monotonic basin hopping (MBH) is a stochastic global optimization technique. This algorithm is a two-phase approach that combines the global stepping algorithm with the local minimization procedure at each iteration [29]. The algorithm model uses random perturbations to jump basins and a local search algorithm to optimize each basin. The model iterates as follows: The first phase uses random perturbation to jump basins of coordinates. The second phase uses a local optimization procedure to evaluate the new coordinates and decide to accept or reject the coordinates based on the minimized function value. This algorithm's original idea is to map the objective function into searching the local minima from the initial point. This mechanism can significantly improve the efficiency of problem solving. Main idea of MBH is mapping the objective function $f(x_0)$ into the local minima found starting from x_0; MBH can be implemented as Algorithm 3:

Algorithm 3 Monotonic basin hopping

Require: $x_0 \longleftarrow$ Generate initial solution
Ensure: optimal x, $f(x)$
  $x_0 \longleftarrow$ generate initial solution $g_0 = 0$ and $d_0 = 0$
  $x_0 \longleftarrow$ minimize (f, x_0)
  repeat
    $y \longleftarrow$ perturb(x)
    $y \longleftarrow$ minimize (f, y)
    $x \longleftarrow$ acceptance (x, z)
  until termination condition met

In this paper, we combine this concept's original generalization, resulting in a meta-algorithm that operates on any population using a suitable algorithm. The actual method is recovered when a population containing a single individual is used and coupled with a local optimizer.

3.4. Evolution Strategy Algorithms

CMA-ES

Covariance matrix adaptation–evolution strategy (CMA-ES) is a stochastic technique for involuting nonlinear, non-convex, continuous black-box optimization problems [30]. It is based on the idea of self-adaptation in evolution strategies. The mechanism of this algorithm is to construct parametric distribution on the searching space in which feature functions are defined in advance. A population of solution candidates is selected from this parametric searching distribution. Then, these candidates are evaluated by a black-box function. Given the function values at the sampled points, updating and storing the covariance matrix dominates the time and space complexity in each iteration of the algorithm. The covariance matrix where time and space complexity dominate is updated and stored at each iteration of the algorithm.

Advantages of CMA-ES: It is suitable for small-scale non-separable optimization problems Disadvantages of CMA-ES: It has high complexity and premature stagnation.

4. Simulation Environment

Algorithm deployment has a high requirement for low computational complexity, and it is considered here. The configuration of the simulation platform is expressed as: CPU Intel i7 10750H and RAM 16 GB. A series of simulation experiments are executed to compare the performance of these metaheuristic algorithms and to find the best algorithm for the power allocation issue.

We consider wireless cellular networks of different scales, with cell populations of 2×2, 3×3, and 4×4. In each cell, users distributed randomly and uniformly in range $r \in [R_{\min}, R_{\max}]$. The small-scale fading follows Rayleigh distribution, and the Jakes model is adopted with f_d. The large-scale fading is formula as $\beta = -120.9 - 37.6 \log_{10}(d) + 10 \log_{10}(z)$ dB, according to the Long-Term Evolution (LTE) standard, where z is shadow effect element and d is the transmitter-to-receiver distance (km). Table 1 collects the primary parameters of the network. The maximum number of iterations is determined as 1000, based on the simulation results. In general, metaheuristic algorithms use randomized search techniques, in which optimization performance highly relies on the initial value and fine parameter tuning. Hence, reproducible optimization results obtained under the same conditions cannot be guaranteed. Therefore, we conducted 20 repeated trials and performed a statistical analysis of the results to compare the performance of the proposed algorithms. The performance of the compared methods is evaluated by averaging 20 trial runs.

Table 1. Simulation parameters of cellular network.

Notation	Simulation Parameter	Value
N	Number of BS	4, 9, 16
M	Average users per cell	2, 4, 8
K	Total number of user	MN
f_d	The Doppler frequency	10 HZ
$P_{\min}$	Minimum allocated power	5 dBm
$P_{\max}$	Maximum allocated power	38 dBm
$R_{\min}$	Inner space distance	0.01 km
$R_{\max}$	Half cell-to-cell distance	1 km
T	Time period	20 ms
σ^2	Noise power spectral density	−114 dBm/Hz
z	Shadowing	8 dB standard deviation

5. Simulation Results

In this section, we present the simulation results to indicate the performance of the metaheuristic algorithms. We use a simple generation–evaluation method for the metaheuristic algorithm for tuning the parameters. A set of a priori candidate configurations is generated. Then, each of these configurations is evaluated to find its optimal configuration. Furthermore, we explore the computation capability of the above algorithms on different network scales. The length of the search process is 50,000 evaluations. Meanwhile, three benchmark algorithms, which are the FP, WMMSE, and random strategy (RAND) stated before, are tested as the comparisons.

Figures 2–4 indicate the searching process for metaheuristic algorithms with different N values. The network's average sum rate is expressed as the value of the objective function based on the number of fitness evaluations. According to the rate of the rising fitness value, we intercept two intervals from the searching range, which are [8000, 15,000] and [37,000, 43,000]. We can observe that the differential evolution algorithms always have good performance when it is rapidly rising, and the swarm intelligence algorithms always have a good performance when it is slowly rising. The results statistically indicate that the performances of the proposed algorithms are similar after fixed generations.

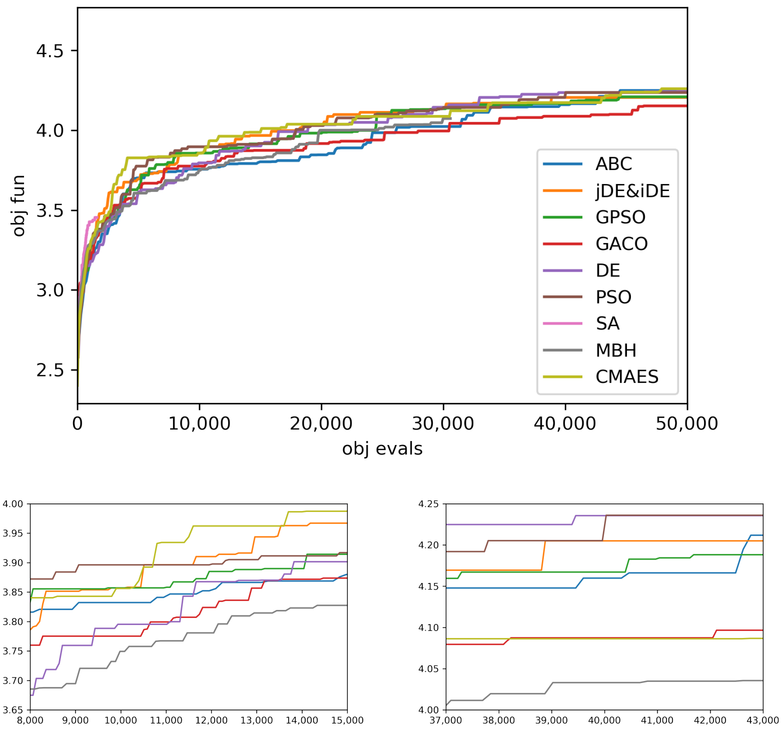


Figure 2. Average rate during fitness evaluation ($N = 4$).

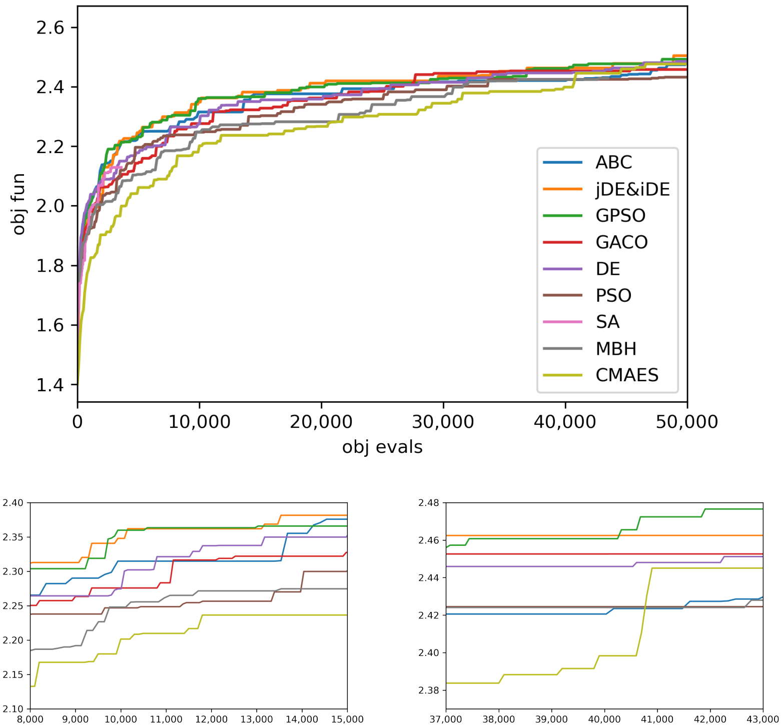


Figure 3. Average rate during fitness evaluation ($N = 9$).

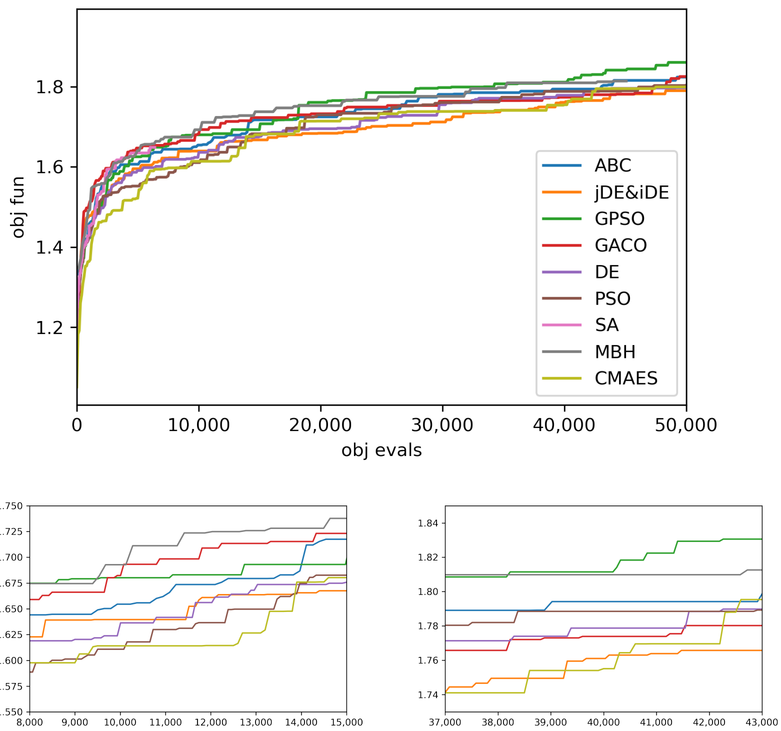


Figure 4. Average rate during fitness evaluation ($N = 16$).

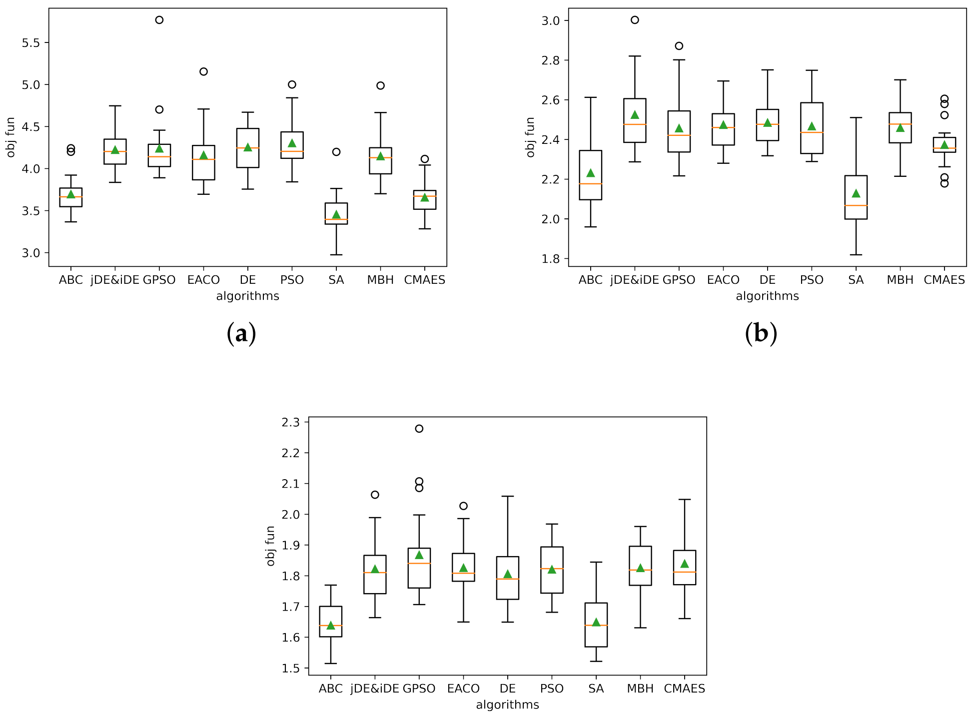
Table 2 shows the obtained solution of the numerical experiments. We focus on the average performance of the above algorithms for 20 trials. Based on the previous results, the swarm intelligence algorithms PSO and GPSO perform the best when $N = 4$ and

$N = 16$, respectively. The differential evolution algorithms jDE&iDE perform the best when $N = 9$.

Table 2. Obtained solution (bps/Hz) of the numerical experiments.

Algorithms			$N = 4$			$N = 9$			$N = 16$
	Max	**Mean**	**Std**	**Max**	**Mean**	**Std**	**Max**	**Mean**	**Std**
ABC	4.241	3.694	0.224	2.612	2.231	0.197	1.769	1.638	0.067
jDE&iDE	4.747	4.221	0.222	3.002	2.524	0.181	2.063	1.821	0.093
GPSO	4.768	4.238	0.402	2.871	2.456	0.169	2.278	1.867	0.143
EACO	5.153	4.160	0.345	2.695	2.474	0.118	2.026	1.825	0.083
DE	4.670	4.252	0.275	2.750	2.484	0.122	2.058	1.805	0.100
PSO	4.999	4.303	0.313	2.748	2.467	0.151	1.968	1.820	0.089
SA	4.198	3.454	0.268	2.510	2.128	0.177	1.844	1.648	0.090
MBH	4.987	4.147	0.292	2.700	2.458	0.131	1.960	1.825	0.083
CMAES	4.114	3.655	0.216	2.605	2.373	0.103	2.048	1.839	0.086

Figure 5 shows the corresponding distribution of the best fitness for the metaheuristic algorithms. CMA-ES is the most robust technique based on the average most minor standard deviation of the best values.



(**a**) (**b**) (**c**)

Figure 5. Distribution of best fitness over 20 trials. (**a**) $N = 4$; (**b**) $N = 9$; (**c**) $N = 16$.

We also obtain a numerical example result of the experiment trials with different user densities and scales of the network. Compared with the values of the averaged sum rate in Figures 6 and 7, the performance of the metaheuristic algorithms is not stable, especially depending on the specific solution-scale effects. Additionally, the best fitness value of the metaheuristic algorithms decreases significantly with the increase in the solution's computation scale compared to the result of the conventional algorithm. In large-scale scenarios, this type of approach costs much more than the other algorithms over time, which means that there is still potential to improve the performance of the metaheuristic algorithms.

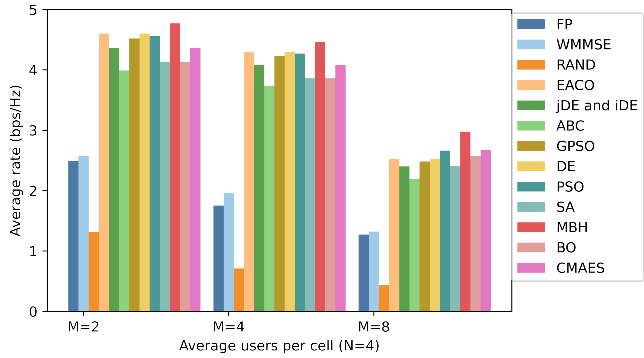


Figure 6. The average rate versus user number per cell.

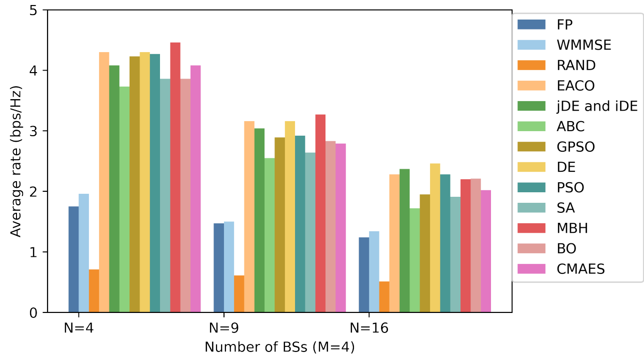


Figure 7. The average rate versus number of cells.

6. Conclusions

The optimal power allocation problem in the cellular network with an IMAC has been investigated, and the model-free metaheuristic approaches have been implemented to handle this problem. To be consistent with the optimization objectives of the PA problem, the network's sum-rate SINR is used as the objective function. Then, a range of metaheuristic algorithms are proposed, and these algorithms work as a black-box solver to search for the optimal power allocation under constraints with a specific CSI.

The simulation results show that the proposed metaheuristic algorithms outperform the conventional benchmark algorithms in different scenarios. We can observe that metaheuristic algorithms have good generalization abilities with simulated communication networks. The experiment results statistically demonstrate that it is hard to determine the overall winner of the algorithms. The metaheuristic methods perform well generally, and the actual performance gap is related to the solution scales. The covariance matrix adaptation–evolution strategy (CMA-ES) is the most robust technique. Differential evolution algorithms (DE and jDE&iDE) and swarm intelligence algorithms (GPSO and PSO) excel in general scenarios.

In our future research, the heterogeneous network will be further studied to accommodate the industry scenarios with specific user distributions and geographical environments.

Author Contributions: Conceptualization, Q.S.; methodology, Q.S.; validation, Q.S. and O.P.; formal analysis, H.W.; investigation, Q.S.; data curation, H.W.; writing—original draft preparation, Q.S.; writing—review and editing, O.P.; visualization, Q.S.; supervision, O.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Saint Petersburg State University, project ID: 93024916, and a grant of the President of the Russian Federation for the state support of young Russian scientists—candidates of science (Project number MK-4674.2021.1.1).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hu, R.Q.; Qian, Y. An energy efficient and spectrum efficient wireless heterogeneous network framework for 5G systems. *IEEE Commun. Mag.* **2014**, *52*, 94–101. [CrossRef]
2. Zhang, H.; Chu, X.; Guo, W.; Wang, S. Coexistence of Wi-Fi and heterogeneous small cell networks sharing unlicensed spectrum. *IEEE Commun. Mag.* **2015**, *53*, 158–164. [CrossRef]
3. Lei, L.; Yuan, D.; Ho, C.K.; Sun, S. Power and channel allocation for non-orthogonal multiple access in 5G systems: Tractability and computation. *IEEE Trans. Wirel. Commun.* **2016**, *15*, 8580–8594. [CrossRef]
4. Shen, K.; Yu, W. Fractional programming for communication systems—Part I: Power control and beamforming. *IEEE Trans. Signal Process.* **2018**, *66*, 2616–2630. [CrossRef]
5. Shi, Q.; Razaviyayn, M.; Luo, Z.Q.; He, C. An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel. *IEEE Trans. Signal Process.* **2011**, *59*, 4331–4340. [CrossRef]
6. Zhang, H.; Venturino, L.; Prasad, N.; Li, P.; Rangarajan, S.; Wang, X. Weighted sum-rate maximization in multi-cell networks via coordinated scheduling and discrete power control. *IEEE J. Sel. Areas Commun.* **2011**, *29*, 1214–1224. [CrossRef]
7. Yu, W.; Kwon, T.; Shin, C. Multicell coordination via joint scheduling, beamforming, and power spectrum adaptation. *IEEE Trans. Wirel. Commun.* **2013**, *12*, 1–14. [CrossRef]
8. Shi, C.; Wang, F.; Sellathurai, M.; Zhou, J. Non-cooperative game theoretic power allocation strategy for distributed multiple-radar architecture in a spectrum sharing environment. *IEEE Access* **2018**, *6*, 17787–17800. [CrossRef]
9. Wang, T.; Wen, C.K.; Wang, H.; Gao, F.; Jiang, T.; Jin, S. Deep learning for wireless physical layer: Opportunities and challenges. *China Commun.* **2017**, *14*, 92–111. [CrossRef]
10. Vu, T.X.; Lei, L.; Chatzinotas, S.; Ottersten, B. Machine learning based antenna selection and power allocation in multi-user MISO systems. In Proceedings of the 2019 International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT), Avignon, France, 3–7 June 2019; pp. 1–6.
11. Sanguinetti, L.; Zappone, A.; Debbah, M. Deep learning power allocation in massive MIMO. In Proceedings of the 2018 52nd Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 28–31 October 2018; pp. 1257–1261.
12. Reddy, Y.B. Genetic algorithm approach for adaptive subcarrier, bit, and power allocation. In Proceedings of the 2007 IEEE International Conference on Networking, Sensing and Control, London, UK, 15–17 April 2007; pp. 14–19.
13. Mukherjee, A.; Goswami, P.; Yan, Z.; Yang, L.; Rodrigues, J.J. ADAI and adaptive PSO-based resource allocation for wireless sensor networks. *IEEE Access* **2019**, *7*, 131163–131171. [CrossRef]
14. Khanolkar, S.; Sharma, N.; Anpalagan, A. Energy-Efficient Resource Allocation in Underlay D2D Communication using ABC Algorithm. *Wirel. Pers. Commun.* **2022**, *125*, 1443–1468. [CrossRef]
15. Meng, F.; Chen, P.; Wu, L.; Cheng, J. Power allocation in multi-user cellular networks: Deep reinforcement learning approaches. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 6255–6267.
16. Zholobova, A.; Zholobov, Y.; Polyakov, I.; Petrosian, O.; Vlasova, T. An Industry Maintenance Planning Optimization Problem Using CMA-VNS and Its Variations. In Proceedings of the International Conference on Mathematical Optimization Theory and Operations Research, Irkutsk, Russia, 5–10 July 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 429–443.
17. Schlüter, M.; Egea, J.A.; Banga, J.R. Extended ant colony optimization for non-convex mixed integer nonlinear programming. *Comput. Oper. Res.* **2009**, *36*, 2217–2229. [CrossRef]
18. Karaboga, D.; Basturk, B. On the performance of artificial bee colony (ABC) algorithm. *Appl. Soft Comput.* **2008**, *8*, 687–697. [CrossRef]
19. Yan, G.; Li, C. An effective refinement artificial bee colony optimization algorithm based on chaotic search and application for pid control tuning. *J. Comput. Inf. Syst.* **2011**, *7*, 3309–3316.
20. Eberhart, R.; Kennedy, J. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
21. Bai, Q. Analysis of particle swarm optimization algorithm. *Comput. Inf. Sci.* **2010**, *3*, 180. [CrossRef]
22. Biscani, F.; Izzo, D. A parallel global multiobjective framework for optimization: Pagmo. *J. Open Source Softw.* **2020**, *5*, 2338. [CrossRef]
23. Selvi, V.; Umarani, R. Comparative analysis of ant colony and particle swarm optimization techniques. *Int. J. Comput. Appl.* **2010**, *5*, 1–6. [CrossRef]

24. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
25. Wu, Y.C.; Lee, W.P.; Chien, C.W. Modified the performance of differential evolution algorithm with dual evolution strategy. In Proceedings of the International Conference on Machine Learning and Computing, Singapore, 26–28 February 2011; Volume 3, pp. 57–63.
26. Elsayed, S.M.; Sarker, R.A.; Essam, D.L. Differential evolution with multiple strategies for solving CEC2011 real-world numerical optimization problems. In Proceedings of the 2011 IEEE Congress of Evolutionary Computation (CEC), New Orleans, LA, USA, 5–8 June 2011; pp. 1041–1048.
27. Brest, J.; Greiner, S.; Boskovic, B.; Mernik, M.; Zumer, V. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Trans. Evol. Comput.* **2006**, *10*, 646–657. [CrossRef]
28. Corana, A.; Marchesi, M.; Martini, C.; Ridella, S. Minimizing multimodal functions of continuous variables with the "simulated annealing" algorithm—Corrigenda for this article is available here. *ACM Trans. Math. Softw. (TOMS)* **1987**, *13*, 262–280. [CrossRef]
29. Wales, D.J.; Doye, J.P. Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms. *J. Phys. Chem. A* **1997**, *101*, 5111–5116. [CrossRef]
30. Hansen, N. The CMA evolution strategy: A comparing review. In *Towards a New Evolutionary Computation*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 192.