sensors

MDPI

*Article*

# Resource Allocation in the Cognitive Radio Network-Aided Internet of Things for the Cyber-Physical-Social System: An Efficient Jaya Algorithm †

**Xiong Luo [1,2,*], Zhijie He [1,2], Zhigang Zhao [1,2], Long Wang [1,2], Weiping Wang [1,2], Huansheng Ning [1], Jenq-Haur Wang [3], Wenbing Zhao [4] and Jun Zhang [5]**

[1] School of Computer and Communication Engineering, University of Science and Technology Beijing (USTB), Beijing 100083, China; G20178705@xs.ustb.edu.cn (Z.H.); B20180325@xs.ustb.edu.cn (Z.Z.); lwang@ustb.edu.cn (L.W.); weipingwangjt@ustb.edu.cn (W.W.); ninghuansheng@ustb.edu.cn (H.N.)

[2] Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing 100083, China

[3] Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei 10608, Taiwan; jhwang@csie.ntut.edu.tw

[4] Department of Electrical Engineering and Computer Science, Cleveland State University, Cleveland, OH 44115, USA; w.zhao1@csuohio.edu

[5] Science and Technology Division, North China Institute of Science and Technology, Beijing 101601, China; zhangjunnm@ncist.edu.cn

* Correspondence: xluo@ustb.edu.cn; Tel.: +86-10-6233-2526

† This paper is an extended version of "An Efficient Jaya Algorithm for Resource Allocation in the Cognitive-Radio-Networks-Aided Internet of Things" published in Proceedings of the IEEE Conferences on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, Congress on Cybermatics, Halifax, NS, Canada, 30 July–3 August 2018.

Received: 24 September 2018; Accepted: 17 October 2018; Published: 27 October 2018

**Abstract:** Currently, there is a growing demand for the use of communication network bandwidth for the Internet of Things (IoT) within the cyber-physical-social system (CPSS), while needing progressively more powerful technologies for using scarce spectrum resources. Then, cognitive radio networks (CRNs) as one of those important solutions mentioned above, are used to achieve IoT effectively. Generally, dynamic resource allocation plays a crucial role in the design of CRN-aided IoT systems. Aiming at this issue, orthogonal frequency division multiplexing (OFDM) has been identified as one of the successful technologies, which works with a multi-carrier parallel radio transmission strategy. In this article, through the use of swarm intelligence paradigm, a solution approach is accordingly proposed by employing an efficient Jaya algorithm, called PA-Jaya, to deal with the power allocation problem in cognitive OFDM radio networks for IoT. Because of the algorithm-specific parameter-free feature in the proposed PA-Jaya algorithm, a satisfactory computational performance could be achieved in the handling of this problem. For this optimization problem with some constraints, the simulation results show that compared with some popular algorithms, the efficiency of spectrum utilization could be further improved by using PA-Jaya algorithm with faster convergence speed, while maximizing the total transmission rate.

**Keywords:** cognitive radio networks (CRNs); orthogonal frequency division multiplexing (OFDM); resource allocation; Internet of Things (IoT); Jaya algorithm

## 1. Introduction

With the rapid development of the Internet of Things (IoT) within the cyber-physical-social system (CPSS), there is an ever-increasing demand for communication network bandwidth for achieving high-performance applications which enrich and broaden the interactions and connections in the cyber-physical-social world. Thus, some communication technologies have been developed using new frequency bands. Moreover, various improved modulation and coding technologies have also been proposed to utilize the existing frequency spectrum more efficiently [1–3]. However, spectrum resources are limited. They are a kind of extremely valuable natural resource and are becoming increasingly restricted or even depleted. In response to this limitation, it is necessary to improve the current spectrum management methods. Fortunately, the recent advances of big data and artificial intelligence technologies make it possible to address the issue.

Cognitive radio network (CRN), as an intelligent wireless communication system, can dynamically share and allocate the spectrum in accordance with the current state of the channel, and it has been seen as an effective solution to the current low utilization of spectrum resources [4,5]. Furthermore, CRNs have been employed to develop the high-performance IoT [6–13]. For example, CRN technology applied to wireless sensor network (WSN) systems in IoT can well sense the surrounding environment and adapt to environmental changes by changing the transmission parameters in real time through information acquired from the environment. Such characteristics make it especially suitable for practical production scenarios. Generally, in the CRN-aided IoT, the secondary user (i.e., cognitive user) may selectively access the idle frequency band of the primary user (i.e., authorized user) to improve the utilization of the licensed spectrum. Meanwhile, some limit requirements should be met by the interference of the secondary user to the primary user for the frequency band. These limits include the constraints of total power, the bit error rate (BER), and the acceptable interferences of the primary user [14].

Furthermore, orthogonal frequency division multiplexing (OFDM), as a multi-carrier parallel wireless transmission technology, has been identified as one suitable alternative to deal with the allocation and transmission of spectrum resources in CRNs [15,16]. Hence, the issue of how to adaptively allocate downlink resources in a cognitive multi-user OFDM system for IoT has drawn much attention from researchers.

Generally, there exist three types of CRNs, including interweave, underlay, and overlay CRNs. In an interweave scenario, the secondary users adaptively utilize the spectrum band whenever it is not used by the primary users [17]. In the underlay case, the secondary users are allowed to coexist with the primary users, provided that the interference caused to the primary users is below a predefined threshold [17]. In the latter category, the secondary system has the knowledge of the primary system codebook, channel gains, and transmitted information. A part of the secondary users' power can be used to retransmit the primary users' message so that the secondary users are able to maintain or improve the primary rate while achieving their own communication goals [18]. Generally speaking, the existing solutions mostly use the traditional mathematical optimization method or some greedy searching algorithms, which may suffer from a quite high computational complexity during the implementation process [18,19]. Some evolutionary algorithms, including simulated annealing (SA) [20], genetic algorithm (GA) [21], particle swarm optimization (PSO) [22,23], differential evolution (DE) [24], and immune clonal optimization (ICO) [25], are employed to deal with this issue, with the help of their effective computational features in the swarm intelligence paradigm. Through the use of those algorithms, a satisfactory solution effect was achieved during the resource allocation in CRNs [20–25]. However, there is still room to further improve the optimization effect due to the inherent disadvantages of those algorithms in the practical engineering applications of IoT, such as the possible ease of falling into the local optimal solutions in some cases, the difficulty adjusting many key parameters during the implementation process of the algorithms, and many drawbacks [26–32].

The Jaya algorithm has been widely used in engineering optimization fields due to its excellent searching ability with an algorithm-specific parameterless feature [33]. In consideration of the satisfactory optimization performance achieved by the Jaya algorithm, there is a general idea to utilize the Jaya algorithm to better serve CRN-aided IoT. In this article, motivated by our previous work in ref. [34], the power resource allocation in the downlink of a CRN-aided IoT system is modeled as a constrained optimization problem, and then a solution method is accordingly presented by incorporating the proposed Jaya algorithm into the cognitive OFDM radio network model to achieve a satisfactory power allocation result. Specifically, compared with the traditional Jaya, the proposed Jaya algorithm, named PA-Jaya, achieved faster convergence speed by using a parallel structure and an asynchronous iteration strategy. Actually, the simulation results show that compared with other popular optimization algorithms, our PA-Jaya algorithm can achieve a better power allocation with faster converge speed, which means that it can improve the efficiency of spectrum utilization more effectively in IoT.

This article is organized as follows. Section 2 presents the mathematical model used in the optimization problem we discuss. In Section 3, in addition to analysis for the basic idea and implementation of the traditional Jaya algorithm, the solution method using the proposed PA-Jaya is also presented. Section 4 shows the simulation results to verify the advantages of using the PA-Jaya algorithm, with a comparison with other algorithms. Finally, we provide a conclusion of this article in Section 5.

## 2. The Problem Formulation

### 2.1. The Basic Model of OFDM Power Allocation in CRNs

In this article, we consider the following application situation for IoT within a CRN. The service range of a base station covers one primary user and $K$ secondary users. The primary user and the secondary users use adjacent frequency bands, and the secondary users use OFDM transmission technology. Moreover, we assume that the channel is slowly fading within one OFDM symbol period, and the base station has full channel state information. The total number of subcarriers is $N$, and the bandwidth of each subcarrier is $W_c$. Let $b_{k,n}$ be the transmission rate of user $k$ on the subcarrier $n$. Thus, the maximum number of bits in a symbol transmitted on this subcarrier is set to [35–37]

$$b_{k,n} = \left\lfloor \log_2 \left\lfloor 1 + \frac{p_{k,n} g_{k,n}^2}{\delta(N_0 W_c + S_{k,n})} \right\rfloor \right\rfloor, \tag{1}$$

where $\lfloor \cdot \rfloor$ denotes the floor function; $p_{k,n}$ represents the power of user $k$ on subcarrier $n$; $g_{k,n}$ is the channel gain of user $k$ on subcarrier $n$; and $N_0$ represents the noise spectral density power, which is the same for all users and subcarriers, and it is a constant. In addition, $S_{k,n}$ indicates the primary user's interference to the secondary user. The variable $\delta$ indicates the BER of the transmission in the case of a physical layer using multiple quadrature amplitude modulation (MQAM) technology, and it can be expressed as

$$\delta = -\frac{2}{3} \ln(5p_e), \tag{2}$$

where $p_e$ denotes the BER.

Generally, in a CRN, the optimization goal of power resource allocation is to maximize the total transmission rate of the system for subusers under the restriction of the authorized users' interference threshold, total transmission power, and BER, so as to improve the spectrum utilization. Therefore, the problem can be modeled as

$$\max \quad W_c \sum_{n=1}^{N} \sum_{k=1}^{K} \lambda_{k,n} b_{k,n} = W_c \sum_{n=1}^{N} \sum_{k=1}^{K} \lambda_{k,n} \left\lfloor \log_2 \left\lfloor 1 + \frac{p_{k,n} g_{k,n}^2}{\delta(N_0 W_c + S_{k,n})} \right\rfloor \right\rfloor, \tag{3}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} \lambda_{k,n} = 1, \quad \lambda_{k,n} = \begin{cases} 0, \ b_{k,n} = 0 \\ 1, \ b_{k,n} \neq 0 \end{cases} \tag{4}$$

$$\sum_{n=1}^{N} \sum_{k=1}^{K} p_{k,n} \leqslant p_{\text{total}} \ , \tag{5}$$

$$\sum_{n=1}^{N} \sum_{k=1}^{K} \lambda_{k,n} p_{k,n} F_n \leqslant I_{\text{th}} \ , \tag{6}$$

$$p_e \leqslant p_u . \tag{7}$$

Here, Equation (4) shows that a subcarrier can only be occupied by one user. In addition, $\lambda_{k,n}$ is the subcarrier allocation state variable. It is equal to 1 only when the subcarrier $n$ is occupied by user $k$; otherwise, it is equal to 0. Equation (5) shows that the sum of the power $p_{k,n}$ transmitted by all secondary users cannot exceed the total system upper power limit $p_{\text{total}}$. In Equation (6), $F_n$ represents the interference factor of the secondary users to the primary user on subcarrier $n$, and $I_{\text{th}}$ is the highest user-acceptable maximum interference limit. It means that all subusers' interference to the primary user should not exceed its tolerable upper limit $I_{\text{th}}$. Moreover, Equation (7) indicates that the BER must be less than the maximum BER requirement $p_u$ [38].

### 2.2. The Complex Model with User Rate Proportionality Constraints

Based on the practical scenario discussed above for the basic OFDM power allocation model in CRNs, instead of considering the primary interference constraints, we considered the fairness of channel resource allocation among secondary users. Then, the capacity of different secondary users needs to meet the certain rate proportional constraint, as follows [39].

$$R_1 : R_2 : \cdots : R_K = \alpha_1 : \alpha_2 : \cdots : \alpha_K , \tag{8}$$

$$R_k = \sum_{n=1}^{N} \lambda_{k,n} b_{k,n} , \quad k = 1, 2, \cdots, K \tag{9}$$

$$R_k > 0, \quad k = 1, 2, \cdots, K \tag{10}$$

In Equation (8), $\alpha_k$ $(k = 1, 2, \cdots, K)$ is a predefined constant, representing the rate proportional constraint that needs to be met by secondary users. Different users' capacity is defined in Equation (9), and it can be further calculated with Equation (3). In addition, Equation (10) is a non-negative constraint on users' capacity, which is implicitly indicated by Equation (8). In this model, we still strive for the maximum value of the defined function in Equation (3). The interference constraint, Equation (6), in the previous model is ignored, and we add constraints from Equations (8)–(10) to further limit the capacity of different secondary users.

## 3. The Proposed Solution Method Using the PA-Jaya Algorithm

The Jaya algorithm is a variant of swarm intelligence, and it achieves the optimal solution by constantly performing an iterative search of the same principle [33]. It has been verified that, in some cases, the Jaya algorithm is more flexible and more advantageous because of its parameterless feature. In an effort to avoid the computational limitations and further improve the performance, we propose an efficient solution method for the optimization problem in CRN-aided IoT, through the use of the PA-Jaya algorithm. To this end, in order to achieve faster convergence speed, PA-Jaya was developed by using a parallel structure and an asynchronous iteration strategy.

### *3.1. The General Idea of Jaya Algorithm*

The Jaya algorithm was developed based on the thought that, during each iteration, the solution for a given problem should move toward the best solution and avoid the worst solution. Considering a general optimization problem, it is assumed that at any iteration $i$, the number of design variables is $N$ and the population size is $M$. If $p_{m,n}^{(i)} \triangleq p_{i,m,n}$ is the value of the $n$-th $(n = 1, 2, \cdots, 2N)$ variable for the $m$-th $(m = 1, 2, \cdots, M)$ candidate during the $i$-th iteration, then this value is updated according to the following formula.

$$p'_{i,m,n} = p_{i,m,n} + r_1(p_{i,\text{best},n} - |p_{i,m,n}|) - r_2(p_{i,\text{worst},n} - |p_{i,m,n}|), \tag{11}$$

where $p_{i,\text{best},n}$ is the value of the $n$-th parameter for the best solution and $p_{i,\text{worst},n}$ is the value of the $n$-th parameter for the worst solution in the population. Moreover, $p'_{i,m,n}$ is the new value of $p_{i,m,n}$, and $r_1$ and $r_2$ are two random numbers whose values are in the range of $[0, 1]$, respectively. The term "$r_1(p_{i,\text{best},n} - |p_{i,m,n}|)$" indicates that the solution tries to move towards the best solution, and the term "$-r_2(p_{i,\text{worst},n} - |p_{i,m,n}|)$" shows that the solution tries to avoid the worst solution. The two random numbers $r_1$ and $r_2$ provide the Jaya algorithm with the ability to perform a random search within the search space, theoretically ensuring that the algorithm can converge to the global optimal solution. In addition, the exploration capability of the Jaya algorithm is further improved by utilizing the absolute value of the candidate solution. $p'_{i,m,n}$ is accepted if the function value it produces is better. Let $P_{i,m} = \{p_{i,m,n}\}_{n=1}^{2N}$, $P'_{i,m} = \{p'_{i,m,n}\}_{n=1}^{2N}$, and $J(\cdot)$ be the fitness evaluation function; then, it can be expressed as

$$P_{i+1,m} = \begin{cases} P'_{i,m}, & \text{if } J(P'_{i,m}) > J(P_{i,m}) \\ P_{i,m}, & \text{if } J(P'_{i,m}) \leqslant J(P_{i,m}) \end{cases} \tag{12}$$

The Jaya algorithm updates the candidate solution by comparing the modified solution and its corresponding old solution. A given candidate solution is updated only when the fitness of its modified solution is better than that of the old one, in which case it is updated to the value of the modified solution. Otherwise, it keeps the value of the old one [33].

Afterward, the same procedure is followed for all the candidates in the population. Obviously, this means that only the best solutions are forwarded as input to the next iteration. In other words, the Jaya algorithm always tries to get closer to success (i.e., reaching the best solution) and tries to avoid failure (i.e., moving away from the worst solution). The algorithm struggles to become victorious by reaching the best solution: it is from this aspect that Jaya draws its name. The update procedure of the Jaya algorithm is shown in Algorithm 1. Compared with some traditional evolutionary algorithms, e.g., GA and PSO, there is no requirement for algorithm-specific parameters in Jaya, which is the biggest advantage of this algorithm [33,40–43].

**Algorithm 1** Jaya update procedure.

```
Input: population matrix $P$, population size $M$, the number of variables $2N$, fitness vector $f$, the best and worst solution vectors: $p_{\text{best}}$ and $p_{\text{worst}}$
Output: updated population matrix $P$, updated fitness vector $f$
for $m := 1$ to $M$ do
    for $n := 1$ to $2N$ do
        Choose a random number $r_{m,n,1}$ from [0,1];
        Choose a random number $r_{m,n,2}$ from [0,1];
        $p'_{m,n} = p_{m,n} + r_{m,n,1}(p_{\text{best},n} - |p_{m,n}|) - r_{m,n,2}(p_{\text{worst},n} - |p_{m,n}|)$;
    end for
    $f'_m = J(P'_m)$;
    if $f'_m > f_m$ then
        for $n := 1$ to $2N$ do
            $p_{m,n} = p'_{m,n}$;
        end for
        $f_m = f'_m$;
    end if
end for
```

### 3.2. PA-Jaya for the Fundamental Issue in the Cognitive OFDM Radio Network

With the analysis mentioned in Section 2, the basic spectrum resource allocation problem discussed in this article can be modeled as a constrained optimization problem. Hence, the problem can be transformed into finding a user's corresponding power allocation scheme for $p_{k,n}$ under the premise of satisfying the constraints in order to maximize the total transmission rate of all secondary users.

In the process of using evolutionary algorithms to solve problems, the coding representation of the individual population and the setting of the fitness function are crucial components. Therefore, here, we first specify these two parts.

According to the requirements mentioned above, the power allocation scheme can be expressed as a matrix of $K \times N$ dimensions. It can be expressed as

$$P = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,N-1} & p_{1,N} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,N-1} & p_{2,N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{K,1} & p_{K,2} & \cdots & p_{K,N-1} & p_{K,N} \end{bmatrix}, \tag{13}$$

where the row of the matrix represents the user $k$ $(k = 1, 2, \cdots, K)$, the column represents the subcarrier $n$ $(n = 1, 2, \cdots, N)$, and each element of the matrix $p_{k,n}$ represents the power obtained by the user $k$ on the subcarrier $n$. According to the constraint described by Equation (4), one subcarrier can be occupied by only one user; that is to say, there can be only one nonzero element per column of this matrix. Hence, we only considered those nonzero elements in the model.

Here, for convenience, we represent each solution individual in the population with a matrix of size $1 \times 2N$, in which the first $N$ elements are in the range of $[1, K]$, indicating how the $N$ subcarriers are assigned to the $K$ users. The next $N$ elements represent the specific power values allocated to each subcarrier. According to Equations (5) and (6), these variables should be in the range of $[0, p_{\text{total}}]$ and $[0, \frac{I_{\text{th}}}{F_n}]$. Then, the structure of the population can be depicted as in Figure 1.

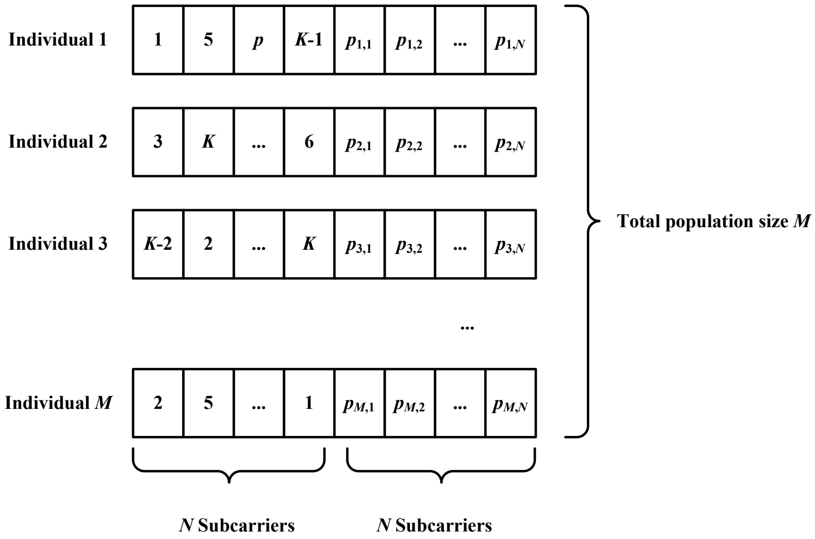

**Figure 1.** Structure of population and individuals designed for the power allocation problem.

Since the basic spectrum resource allocation problem discussed in this article can be clearly expressed as a constrained maximization problem, we chose Equation (3) as the fitness evaluation function of the proposed PA-Jaya algorithm. Considering the specificity of the individual coding methods in this article, it is necessary to perform special processing when using the PA-Jaya algorithm for a population update. That is, the first $N$ elements need to be rounded after each update, and they are bounded within $[1, K]$. The next $N$ elements are only bounded within $[0, p_{\text{total}}]$. The remaining constraints are judged after the whole population is generated, and the fitness value is set to 0 for the individual solutions that do not meet the requirements. All individual solutions need to be compared with the corresponding individual solutions before the population update, and those with larger fitness values enter the next iteration. Meanwhile, the best individuals and the worst individuals in the population are also updated.

The proposed PA-Jaya algorithm, as an improvement to the traditional Jaya algorithm, aims to speed up the convergence of the algorithm by updating the values asynchronously with a parallelized structure and by reducing infeasible solutions through an inner loop iteration strategy.

Motivated by the successful work of improving the convergence speed using asynchronous iteration in PSO [44,45], in the proposed PA-Jaya algorithm, the update of the best individual solutions is not performed only after the whole population update is completed, but is performed after each individual solution updates. Immediately after each individual solution update is completed, it is compared with all individual solutions of the previous generation, and the best individual is selected to update the subsequent individual solutions.

Another important improvement of the proposed Jaya algorithm is that it sets an inner loop to reduce the generation of infeasible solutions. Considering that the basic resource allocation problem in this article is a constrained optimization problem, all the candidate solutions obtained need to meet the requirements of Equations (5) and (6). In this case, if the traditional Jaya algorithm is directly applied, a large number of infeasible solutions will be generated in the population after each iteration, and these solutions will be discarded, resulting in unsatisfactory efficiency of the iterative solution. Therefore, an intuitive idea is to consider how to reduce these infeasible solutions during the update process of each iteration. The PA-Jaya algorithm addresses this problem through an inner loop in each iteration to repeat the update of a single candidate solution to satisfy the constraints as much as possible; thus, most of the solutions in the population after each iteration will be feasible solutions. Specifically, for the basic spectrum resource allocation problem discussed in this article, when we obtain the first candidate solution during the first iteration, we will immediately observe whether it meets the requirements of Equations (5) and (6). If these requirements are met, we continue to update the subsequent solutions in the population while also paying attention to the constraints, until all the candidate solutions are updated during this iteration. Otherwise, if the first candidate solution does not meet these constraints, the candidate solution will be regenerated until it meets all the requirements or the maximum number of inner loops is reached, and then the subsequent operations are consistent with the above. Thus, based on the above two tricks, the framework of the improved asynchronous Jaya algorithm can be described by Algorithm 2.

Furthermore, motivated by some successful work [46–49], we introduce the parallel structure into the proposed algorithm to make full use of computing resources and reduce the running time. Generally, parallel models of the common GA can be divided into four categories, including the master-slave model, coarse-grained model, fine-grained model, and hybrid model [47].

Here, we use the coarse-grained model, also called the land-based model, which is the most adaptable and widely used GA parallel model. The initial population is divided into several independent subpopulations, and the improved asynchronous Jaya iterative algorithm is used to perform evolution operations independently in each subpopulation. After a certain number of iterations, information exchange is conducted between different subpopulations, and the convergence speed is further accelerated by introducing the optimal individuals to the entire population. By adopting a parallelization structure, the independence between each subpopulation can be ensured, and the individual diversity of each subpopulation can be enriched. This not only ensures the exchange of excellent antibodies within the population, but also prevents the rapid spread of local optimal solutions and the occurrence of immature convergence. The update procedure of the improved PA-Jaya algorithm is shown in Algorithm 3.

**Algorithm 2** The improved asynchronous Jaya update procedure.

```
Input: population matrix $P$, population size $M$, the number of variables $2N$, the number of inner loops $T$, fitness vector $f$, the best and worst solution vectors: $p_{\text{best}}$ and $p_{\text{worst}}$
Output: updated population matrix $P$, updated fitness vector $f$
for $m := 1$ to $M$ do
    for $t := 1$ to $T$ do
        for $n := 1$ to $2N$ do
            Choose a random number $r_{m,n,1}$ from [0,1];
            Choose a random number $r_{m,n,2}$ from [0,1];
            $p'_{m,n} = p_{m,n} + r_{m,n,1}(p_{\text{best},n} - |p_{m,n}|) - r_{m,n,2}(p_{\text{worst},n} - |p_{m,n}|)$;
        end for
        if $P'_m$ meets constraints (5) and (6) then
            break;
        end if
    end for
    $f'_m = J(P'_m)$;
    if $f'_m > f_m$ then
        for $n := 1$ to $2N$ do
            $p_{m,n} = p'_{m,n}$;
        end for
        $f_m = f'_m$;
    end if
end for
```

**Algorithm 3** The improved PA-Jaya update procedure.

```
Input: population matrix $P$, the number of subpopulations $S$, the number of population iterations $D$, the number of iteration intervals of subpopulation information exchange $Q$
Output: updated population matrix $P$
for $i := 1$ to $D$ do
    for $s := 1$ to $S$ do
        Update $P_s$ with the improved asynchronous Jaya algorithm;
    end for
    if $i$ mod $Q = 0$ then
        for $s := 1$ to $S$ do
            Replace the worst individual solution in $P_s$ with the global best individual solution;
        end for
    end if
end for
```

Figure 2 presents a schematic diagram for using the PA-Jaya algorithm to achieve optimization for the cognitive OFDM radio network power allocation model. The basic steps of the solution method are as follows.

**Step 1: Parameter setting.** It starts with the setting of initial parameters, including the number of secondary users $(K)$, the number of variables $(2N)$, population size $(M)$, the number of population divisions $(S)$, the number of inner loops $(T)$, the number of iteration intervals of subpopulation information exchange $(Q)$, and the algorithm termination criterion. For the current situation, the termination criterion is set as the maximum number of iterations $(D)$, which means that the algorithm is terminated when it iterates more than this value.

**Step 2: PA-Jaya initialization.** Initial values of individual solutions can be randomly generated under the constraints in Equations (5) and (6). After finishing the same operation for all the variables in the whole population, the solutions matrix is well initialized.

**Step 3: Fitness evaluation.** Once all the candidate solutions are initialized, every individual solution is evaluated with the fitness function. Considering the practical problem in this article, the fitness function's purpose is to get the maximum object value according to Equation (3).

**Step 4: Solution update.** In each parallel subpopulation, by comparing the fitness function value of each candidate solution, we can easily select the best and the worst solutions. Hence, we are able to modify the old solution with the proposed asynchronous Jaya iteration strategy. Let $P_{i,m} = \{p_{i,m,n}\}_{n=1}^{2N}$, $P'_{i,m} = \{p'_{i,m,n}\}_{n=1}^{2N}$, and let $J(\cdot)$ be the mathematical operation in Equation (3), respectively. Then, the update of the candidate solution is checked by assessing

$$J(P'_{i,m}) > J(P_{i,m}). \tag{14}$$

If Equation (14) is met, $p'_{i,m}$ will be applied to the basic power allocation model; that is to say, the candidate solution will be updated through $p_{i+1,m} = p'_{i,m}$. Otherwise, it retains the original value. Generally, the traditional Jaya updates its best and worst individuals after the entire population has been updated. Compared with this, PA-Jaya updates the best value in every single individual calculation, and it can be seen as an asynchronous update strategy. In addition, a maximum number of loops in the inner layer is set to reduce infeasible solutions with the purpose of further accelerating the speed of convergence in PA-Jaya. Thus, the PA-Jaya algorithm guides the candidate solution update in each independent subpopulation, while all these subpopulations exchange information periodically. The whole process can be described by Algorithm 3.

**Step 5: Convergence criterion.** The stopping condition is checked. Once the searching process reaches the maximum number of iterations, the loop is terminated and the optimum solution is obtained.

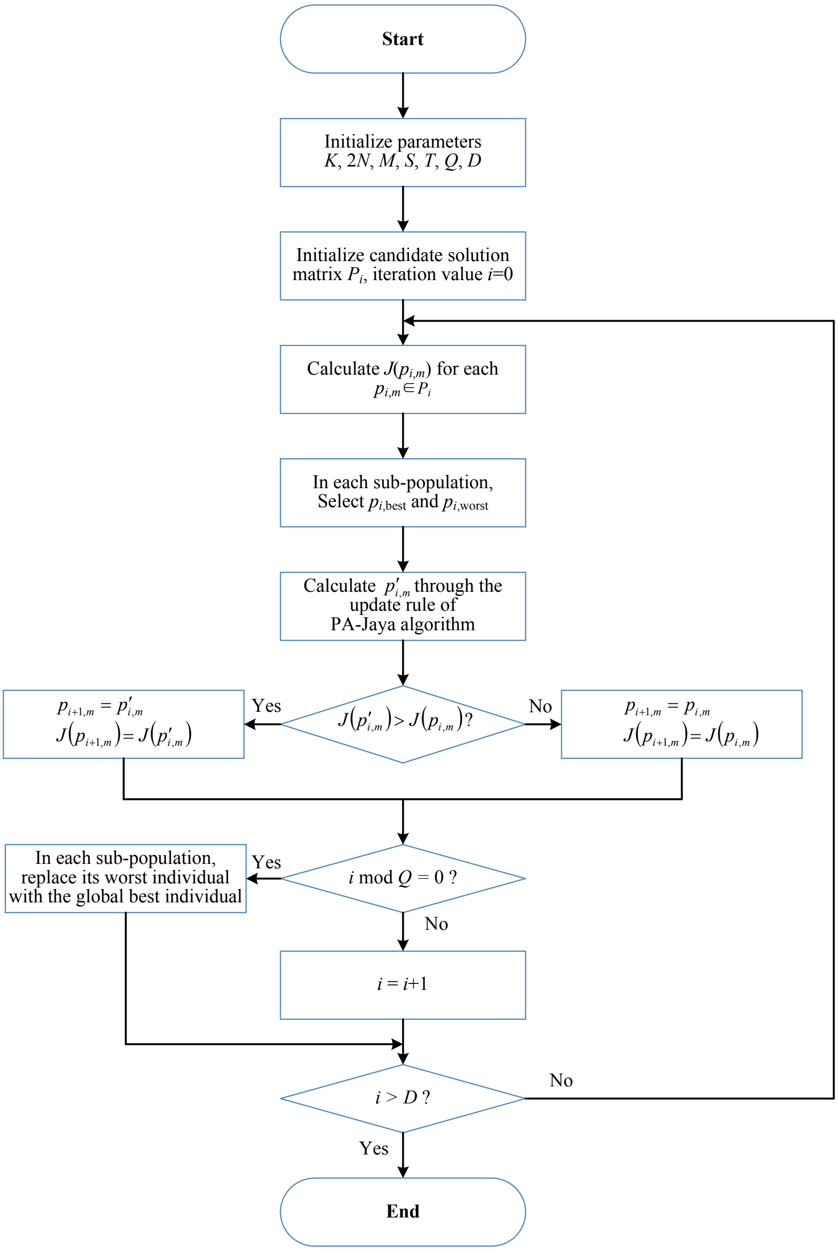
Start
Initialize parameters $K$, $2N$, $M$, $S$, $T$, $Q$, $D$
Initialize candidate solution matrix $P_i$, iteration value $i$=0
Calculate $J(p_{i,m})$ for each $p_{i,m} \in P_i$
In each sub-population, Select $p_{i,\mathrm{best}}$ and $p_{i,\mathrm{worst}}$
Calculate $p'_{i,m}$ through the update rule of PA-Jaya algorithm
$J(p'_{i,m}) > J(p_{i,m})$?
Yes
$p_{i+1,m} = p'_{i,m}$ $J(p_{i+1,m}) = J(p'_{i,m})$
No
$p_{i+1,m} = p_{i,m}$ $J(p_{i+1,m}) = J(p_{i,m})$
$i$ mod $Q$ = 0 ?
Yes
In each sub-population, replace its worst individual with the global best individual
No
$i = i+1$
$i > D$ ?
No
Yes
End

**Figure 2.** Flowchart of the PA-Jaya algorithm for the cognitive orthogonal frequency division multiplexing (OFDM) radio network power allocation model.

### *3.3. PA-Jaya for the User Fairness Issue in Cognitive OFDM Radio Network*

This problem becomes more complicated when the secondary user rate proportional constraint is considered. Obviously, in this problem, the subchannel and power should be jointly allocated to achieve the optimal capacity under the user rate proportional constraints. However, this joint optimization results in high computational costs. In order to address this complex model, the traditional methods are usually based on three steps. The first step is to allocate channels for each secondary user based on the interference factors of each channel. The second step roughly allocates the total transmit power to each user based on the number of channels obtained by each secondary user. The third step is implemented on the basis of the previous two allocations while using the traditional water-filling algorithm to choose the best allocation method for every secondary user.

In order to better handle this problem, the following fairness indicator is introduced [50].

$$\mathbf{I} = \left(\sum_{i=1}^{K} \frac{R_k}{\alpha_k}\right)^2 \left(K \sum_{i=1}^{K} \left(\frac{R_k}{\alpha_k}\right)^2\right)^{-1}. \tag{15}$$

The maximum value of Equation (15) is 1, and when it reaches this value, the fairness of the allocation model is optimal.

The signal-to-noise ratio (SNR) in this model can be written as

$$\mathrm{SNR} = \frac{g_{m,n}^2}{\delta(N_0 W_c + S_{m,n})}. \tag{16}$$

Theoretically, allocating more power to the channels with larger SNRs can maximize the total channel capacity demanded in this model, but doing so may not satisfy the user fairness constraints in Equation (10).

Then, considering the user fairness issue, an optimization approach, called the ROOT-FINDING algorithm, was proposed on the basis of proportional rate constraints [39]. For the case of high channel gain, the ROOT-FINDING algorithm can solve the user rate proportional constraint problem well and achieve spectrum efficiency improvement. However, in order to satisfy the proportional constraint, many subcarriers are often not correctly allocated to the optimal user, resulting in a low overall utilization efficiency of the subcarrier. In Ref. [51], an improved algorithm, called LINEAR, was proposed in an effort to effectively address this problem. This algorithm achieves an increase in spectral efficiency by relaxing the fairness constraint while improving the computational time. Since the user fairness and spectral efficiency may not be well weighed, the algorithm still has room for improvement.

In response to these limitations, through the use of the two-stage allocation scheme described in Ref. [51], here we present a novel method on the basis of the PA-Jaya algorithm. In the first stage, we fully consider the state of each channel. Assuming that transmit power is evenly distributed across all subchannels, then we can reasonably suppose that the proportion of subcarriers assigned to different users is approximately the same as their capacity rates after power allocation. Then, we use the proposed PA-Jaya algorithm to allocate each channel to the user with the best SNR as much as possible while satisfying the total power constraints and user fairness constraints. This procedure can be described as follows.

First, we assign each secondary user the unallocated subcarrier that has the maximum SNR for that user. Then, each user is assigned a number of subcarriers according to his proportion constant, and the priority of the secondary users is determined by the ratio of his obtained capacity to the proportion constant, which is $\frac{R_k}{\alpha_k}$ $(k = 1, 2, \cdots, K)$. Finally, all unoccupied subcarriers are allocated to users who can obtain the maximum SNR; in this case, each secondary user can get one subcarrier at most.

After the subcarrier allocation procedure, we use the proposed PA-Jaya algorithm to further distribute power. Motivated by the work in Ref. [50], in order to better apply our proposed PA-Jaya algorithm, we designed a special fitness function for this issue, which is

$$\mathbf{F} = \begin{cases} W_c \sum\limits_{n=1}^{N} \sum\limits_{k=1}^{K} \lambda_{k,n} b_{k,n} & \text{if} \quad \mathbf{I} \geqslant \epsilon \\ \mathbf{I} - \epsilon & \text{otherwise} \end{cases} \tag{17}$$

Here, $\epsilon$ is a predefined constant representing the minimum requirement for the fairness indicator. In this function, it is obvious that for those solutions which do not meet the user fairness constraint, the algorithm selects them through the fitness function so that they can gradually move in the direction of better satisfying the user fairness requirement. For those solutions satisfying the requirement, the algorithm helps them acquire more channel capacity, eventually converging to the optimal solution. The introduction of $\epsilon$ makes it possible to maximize the channel capacity while satisfying the minimum user fairness requirement. Moreover, the specific algorithm structure is similar to the previous one; it is not described in detail here.

### *3.4. Computational Complexity Analysis for PA-Jaya*

As described above, the population size is set as $M$, the number of variables in each candidate solution is $2N$, and the maximum number of inner iterations is $K$.

According to the flow shown in Algorithm 2, it is obvious that our proposed PA-Jaya algorithm needs to use a triple loop for each iteration update. The number of the outermost loop is set as the population size $M$, which indicates that each individual solution in the population is traversed to ensure that all solutions in the population are updated. The number of the middle loop is set as a predefined constant $K$. The function of this loop is to minimize the generation of infeasible solutions in the updated population, so as to accelerate the overall convergence speed of the algorithm. The number of the innermost loop is set as the number of variables $2N$ in each candidate solution. This loop is essential because our PA-Jaya algorithm needs to generate two random numbers, $r_1$ and $r_2$, for each of these variables when updating each candidate solution.

Then, in the worst case during each iteration, the time complexity required for our proposed PA-Jaya algorithm is $\mathcal{O}(M \times N \times K)$, which seems to be $K$ times the time required by the traditional Jaya algorithm. However, considering that $K$ is a relatively small constant, such an increase in time is acceptable. In addition, PA-Jaya requires fewer iterations to converge, which also makes up for the increase in time required for each iteration to some extent. Moreover, by using a parallelized structure, our algorithm makes full use of existing computing resources while greatly reducing the runtime of the algorithm.

## 4. Simulation Results and Analyses

### *4.1. Simulation Setup*

The simulation was conducted in the environment of MATLAB/Win 7. Assuming that the CRN-aided IoT system exists with a multipath frequency selective fading channel, and that the channel gain of each subcarrier is subject to Rayleigh fading with an average channel gain of 1, then the parameters of the simulation are set as shown in Table 1.

**Table 1.** Model parameters.

| Description | Parameter | Value |
|---|---|---|
| BER | $p_e$ | $10^{-5}$–$10^{-1}$ |
| Transmit BER | $\delta$ | 5 dB |
| Noise spectral density power | $N_0$ | $10^{-7}$ W/Hz |
| Interference factor | $S_{k,n}$ | $10^{-6}$ W |
| Subcarrier bandwidth | $W_c$ | 0.315 |
| Total system upper power limit | $p_{\text{total}}$ | 1–30 W |
| User-acceptable maximum interference limit | $\frac{I_{\text{th}}}{F_n}$ | $10^{-3}$–$10^{-2}$ W |
| The number of secondary users | $K$ | 8 |
| Population size | $M$ | 30 |
| Total number of subcarriers | $N$ | 64 |
| The number of subpopulations | $S$ | 3 |
| The iteration interval of subpopulation communicate | $Q$ | 10 |
| The number of inner loop | $T$ | 10 |
| Total number of iterations | $D$ | 200 |

In order to verify the performance of our proposed solution method using the PA-Jaya algorithm, the simulation model was run 10 times under the same parameter settings, and the average value was obtained. Moreover, the computational performance was also compared between our solution method and other popular algorithms, including SA [20], GA [21], PSO [22,23], DE [24], ICO [25], and traditional Jaya [33]. The detailed parameter settings for each algorithm can be found in Table 2. It is noted that for our proposed PA-Jaya algorithm, the initial population is randomly generated, and each subpopulation is updated by asynchronous Jaya iteration, as shown in Algorithm 2. Then, information exchange between different subpopulations takes place at regular intervals, and the overall process can be seen in Algorithm 3.

Moreover, for the user fairness issue discussed in this article, the user fairness proportion factor is randomly generated. Then, we compare our proposed PA-Jaya algorithm with the algorithms ROOT-FINDING [39] and LINEAR [51]. Our solution to this issue can be divided into two parts. The first part is the allocation of subchannels, which is the same as that in Ref. [50]. In the second part, we use the PA-Jaya algorithm to allocate the appropriate power to the channels of different users while meeting the required constraints. As the fitness function is set as Equation (17), and the minimum user rate proportionality is set to 0.99, the process of conducting PA-Jaya and setting the parameters are consistent with the former issue.

**Table 2.** Controlled parameters of PA-Jaya and other comparative algorithms.

| Parameter | SA | GA | PSO | DE | ICO | Jaya | PA-Jaya |
|---|---|---|---|---|---|---|---|
| Initial temperature | 100 | – | – | – | – | – | – |
| Reannealing interval | 100 | – | – | – | – | – | – |
| Population size | – | 30 | 30 | 30 | 30 | 30 | 30 |
| Scaling factor | – | – | – | 0.3 | – | – | – |
| Crossover factor | – | 0.3 | – | 0.1 | 0.3 | – | – |
| Mutation factor | – | 0.1 | – | – | 0.1 | – | – |
| Initial inertia weight | – | – | 0.9 | – | – | – | – |
| Convergence inertia weight | – | – | 0.4 | – | – | – | – |
| Local acceleration constant | – | – | 2 | – | – | – | – |
| Global acceleration constant | – | – | 2 | – | – | – | – |
| Cloning proportion | – | – | – | – | 0.2 | – | – |

## 4.2. Results and Discussion

We first compare the results of our proposed PA-Jaya algorithm in different subpopulation size cases with the asynchronous Jaya and traditional Jaya algorithms. When the total transmit power ($p_{\text{total}} = 1$ W) and the BER ($\delta = 5$ dB) are limited (which means that the model constraints are satisfied), and the maximum number of iterations $D$ is reached, the results of different algorithms are as shown in Figure 3.

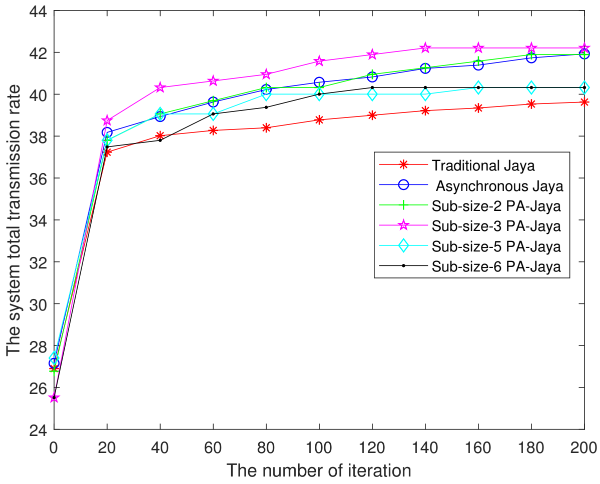

**Figure 3.** Performance comparison of different Jaya algorithms.

In Figure 3, the PA-Jaya algorithm can achieve better results compared with the traditional Jaya algorithm and the asynchronous Jaya algorithm without a parallel structure. Meanwhile, compared with the traditional Jaya algorithm, several other improved Jaya algorithms can get better results, which could be due to the way that asynchronous iteration improves the convergence speed. In addition, comparing the asynchronous Jaya algorithm and the PA-Jaya algorithm with different subpopulation sizes, it can be seen that when the number of subpopulations is 2, the result of PA-Jaya is similar to that of the asynchronous Jaya algorithm. When the number of subpopulations is 3, the result of PA-Jaya is better. However, as the number continues to grow, the result gets worse. On the one hand, this phenomenon shows that the use of a parallel structure can increase the diversity of the whole population to prevent premature local convergence, thereby further improving the performance of the Jaya algorithm. On the other hand, it also shows that when the subpopulation size is too small, the performance of the algorithm is limited. Taking into account the above situation, we set the number of subpopulations to 3 for the simulations hereafter reported in this article.

Meanwhile, Figure 4 shows the total transmission rate of the secondary users obtained by those algorithms under the same conditions. Then, when the number of secondary users is 8 and the maximum number of iterations $D$ is reached, the total transmission rate of the system under different BERs is as shown in Figure 5. Furthermore, the optimal solutions of the total system transmission rate are shown in Table 3, and they are recorded in every 20 generations. While setting the BER of the transmission ($\delta$) and the maximum number of iterations ($D$) to 5 dB and 200, respectively, the total transmission power of the secondary users changes as shown in Figure 6 with the interference tolerance of different primary users.

From Figure 4, it can be seen that, in each iteration, the solution performance of the total system transmission power obtained by our proposed PA-Jaya algorithm is obviously better than that of SA, GA, PSO, DE, ICO, and traditional Jaya algorithms while also achieving faster convergence speed. On the one hand, this is because of the iterative method and parameterless feature of the Jaya algorithm, which make it superior to other algorithms and allow it to converge faster. Then, the asynchronous iteration method we use further accelerates the process. In addition, it can be seen from this figure that several other algorithms fall into local convergence prematurely, and our PA-Jaya algorithm can avoid it well by adopting a multiple-subpopulation structure. This parallel structure helps our PA-Jaya algorithm keep searching for the optimal solution during the iteration.

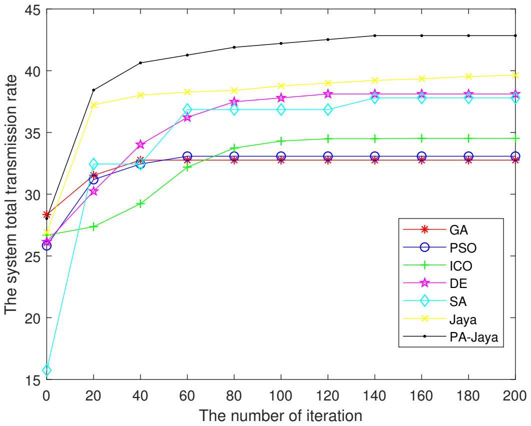

**Figure 4.** Relationship between the number of iterations and the total system transmission rate.

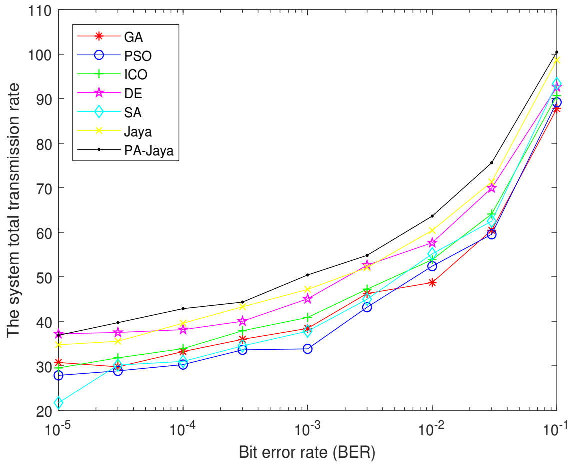

**Figure 5.** Relationship between the bit error rate (BER) and the total system transmission rate.

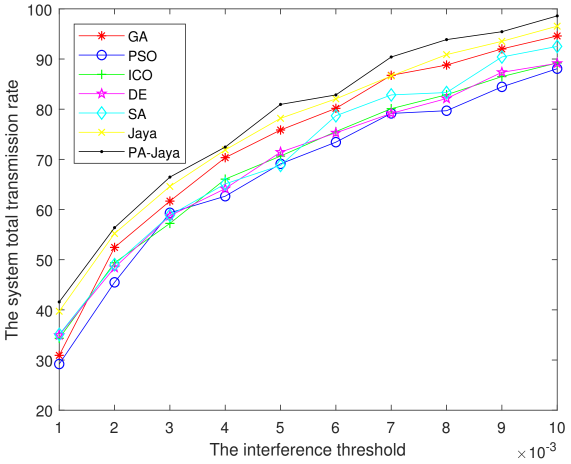

**Figure 6.** Relationship between the interference threshold and the total system transmission rate.

Moreover, as shown in Figure 5, with the lower BER required by the system, the constraint requirements decrease, so the total system transmission rate increases. It also indicates that the system can effectively adapt to different BER constraints in the power allocation model. For the same reason, the solution performance of our PA-Jaya algorithm is also better than those of SA, GA, PSO, DE, ICO, and the traditional Jaya algorithms.

**Table 3.** Optimal solutions of the total system transmission rate during iterations.

| Generation | GA | PSO | ICO | SA | DE | Jaya | PA-Jaya |
|---|---|---|---|---|---|---|---|
| 0 | 28.35 | 25.83 | 26.68 | 15.75 | 26.15 | 26.90 | 27.04 |
| 20 | 31.53 | 31.19 | 27.37 | 32.45 | 30.24 | 37.23 | 38.43 |
| 40 | 32.76 | 32.45 | 29.23 | 32.45 | 34.02 | 38.02 | 40.01 |
| 60 | 32.76 | 33.08 | 32.19 | 36.86 | 36.23 | 38.27 | 41.27 |
| 80 | 32.76 | 33.08 | 33.74 | 36.86 | 37.49 | 38.40 | 41.90 |
| 100 | 32.76 | 33.08 | 34.30 | 36.86 | 37.80 | 38.78 | 42.21 |
| 120 | 32.76 | 33.08 | 34.50 | 36.86 | 38.12 | 39.00 | 42.53 |
| 140 | 32.76 | 33.08 | 34.50 | 37.80 | 38.12 | 39.22 | 42.84 |
| 160 | 32.76 | 33.08 | 34.52 | 37.80 | 38.12 | 39.34 | 42.84 |
| 180 | 32.76 | 33.08 | 34.52 | 37.80 | 38.12 | 39.53 | 42.84 |
| 200 | 32.76 | 33.08 | 34.52 | 37.80 | 38.12 | 39.63 | 42.84 |

Figure 6 shows that with the increase of the tolerable interference threshold, the transmission power allowed to be used by the secondary users increases, and, therefore, the total transmission power of the system increases. Meanwhile, with the increase of the main user's tolerable interference threshold, our algorithm achieves better performance for the same reason mentioned above.

Subsequently, with the same settings for the simulation parameters, we used the proposed PA-Jaya algorithm to solve the multi-user OFDM CRN problem with the user proportional rate constraints mentioned in Section 2.2. Here, the interference generated by the secondary users to the primary users is ignored, and the proportional rate between different users is randomly generated. We set the number of secondary users as 16, with the purpose of better comparing the result of our PA-Jaya algorithm and others in the case of a large number of secondary users, which is slightly different from the former simulation setup. In Figure 7, the comparison is clearly shown, where "Alpha" indicates the user proportionality factor required in this model. Figures 8 and 9 show the comparison of these different algorithms in the user fairness indicator (**I**) and channel capacity with different user numbers, respectively.

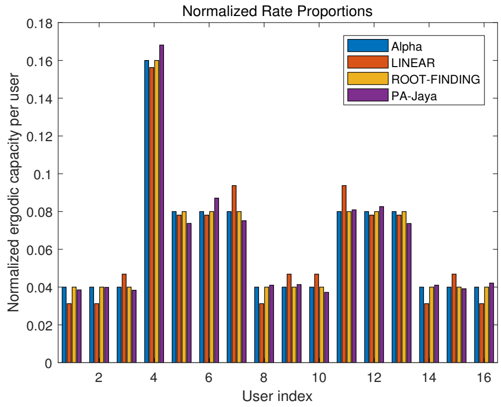

**Figure 7.** Comparison of different algorithm effects under normalized rate proportional constraint.

From the above simulation results, it is obvious that as the number of users increases, it becomes more difficult for the ROOT-FINDING algorithm to make full use of channel resources, so the channel capacity cannot achieve its optimal value. Meanwhile, the LINEAR algorithm has difficulty meeting the user fairness requirement. Our proposed PA-Jaya algorithm makes a compromise between these two aspects by adopting the fitness function and minimum user fairness constraint set. PA-Jaya achieves a similar result to LINEAR in terms of capacity. It is also shown that with the PA-Jaya algorithm, we can better weigh the capacity and fairness indicator, thus obtaining better results than the other algorithms.

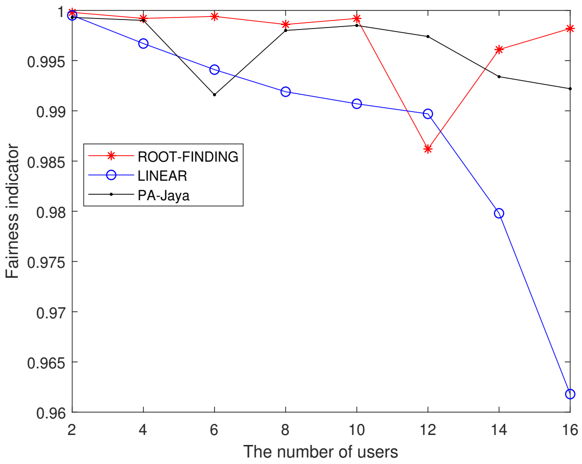

**Figure 8.** Comparison of different algorithms for the fairness indicator with different user numbers.

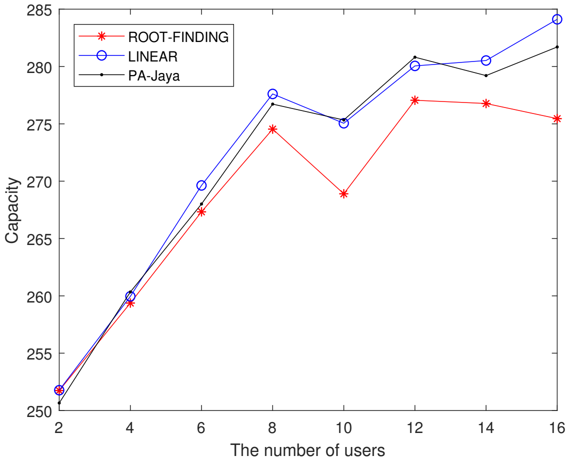

**Figure 9.** Comparison of different algorithms for capacity with different user numbers.

## 5. Conclusions

In this article, with the aim of applying the CRN-aided IoT to the CPSS, a solution method for multi-user cognitive OFDM radio network power allocation using our PA-Jaya algorithm is proposed. In consideration of the special advantages of the swarm intelligence-based PA-Jaya algorithm for tackling the optimization problem with faster convergence and higher efficiency, the solution method using PA-Jaya may be a competitive choice for addressing the power allocation issue in CRN-aided IoT with high-quality requirements for complex parameter optimization. Simulation results show that the proposed algorithm is an efficient scheme for CRNs and can maximize the total transmission rate of this system model while satisfying the requirements of tolerable interference, total power limitation, and BER of the primary user. For user fairness issues, our proposed PA-Jaya algorithm can maximize the required channel capacity while satisfying certain user fairness constraints. By employing our PA-Jaya algorithm, the CRN technology is further enhanced to sense environmental changes and adaptively adjust its transmission parameters so that it works well in WSN systems, which has important practical value for production scenarios.

**Author Contributions:** In this article, the initial idea was provided by X.L. and Z.H., who also revised the whole article; Z.H. and Z.Z. designed and implemented the experiments; the data and results were analyzed by L.W., W.W., H.N., J.-H.W., W.Z., and J.Z. All the participants have read and agreed to the final manuscript.

**Funding:** This research was funded by the National Key Research and Development Program of China grant number 2018YFC0808306, the National Natural Science Foundation of China grant number 61603032, the University of Science and Technology Beijing–National Taipei University of Technology Joint Research Program grant number TW201705, and the NTUT-USTB Joint Research Program.

**Acknowledgments:** The authors would like to thank editorial board and reviewers for the improvement of this article.

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

## References

1. Cohen, E.; Malka, D.; Shemer, A.; Shahmoon, A.; Zalevsky, Z.; London, M. Neural networks within multi-core optic fibers. *Sci. Rep.* **2016**, *6*, 29080. [CrossRef] [PubMed]
2. Malka, D.; Berkovic, G.; Tischler, Y.; Zalevsky, Z. Super-resolved raman spectra of toluene and toluene-chlorobenzene mixture. *Spectrosc. Lett.* **2015**, *48*, 431–435. [CrossRef]
3. Malka, D.; Berkovic, G.; Hammer, Y.; Zalevsky, Z. Super-resolved raman spectroscopy. *Spectrosc. Lett.* **2013**, *46*, 307–313. [CrossRef]
4. Dhanasekaran, S.; Reshma, T. Full-rate cooperative spectrum sharing scheme for cognitive radio communications. *IEEE Commun. Lett.* **2018**, *22*, 97–100. [CrossRef]

5. Kakalou, I.; Psannis, K.E.; Krawiec, P.; Badea, R. Cognitive radio network and network service chaining toward 5G: Challenges and requirements. *IEEE Commun. Mag.* **2017**, *55*, 145–151. [CrossRef]
6. Zhang, D.; Yang, L.T.; Chen, M.; Zhao, S.; Guo, M.; Zhang, Y. Real-time locating systems using active RFID for internet of things. *IEEE Syst. J.* **2016**, *10*, 1226–1235. [CrossRef]
7. Khan, A.A.; Rehmani, M.H.; Rachedi, A. Cognitive-radio-based internet of things: Applications, architectures, spectrum related functionalities, and future research directions. *IEEE Wirel. Commun.* **2017**, *24*, 17–25. [CrossRef]
8. Yi, L.; Deng, X.; Wang, M.; Ding, D.; Wang, Y. Localized confident information coverage hole detection in internet of things for radioactive pollution monitoring. *IEEE Access* **2017**, *5*, 18665–18674. [CrossRef]
9. Deng, X.; Tang, Z.; Yi, L.; Yang, L.T. Healing multi-modal confident information coverage holes in NB-IoT-enabled networks. *IEEE Internet Things J.* **2018**, *5*, 1463–1473. [CrossRef]
10. Hu, P.; Ning, H.; Qiu, T.; Zhang, Y.; Luo, X. Fog computing based face identification and resolution scheme in internet of things. *IEEE Trans. Ind. Inform.* **2017**, *13*, 1910–1920. [CrossRef]
11. D'Orazio, C.J.; Choo, K.K.R.; Yang, L.T. Data exfiltration from internet of things devices: iOS devices as case studies. *IEEE Internet Things J.* **2017**, *4*, 524–535. [CrossRef]
12. Luo, X.; Zhang, D.; Yang, L.T.; Liu, J.; Chang, X.; Ning, H. A kernel machine-based secure data sensing and fusion scheme in wireless sensor networks for the cyber-physical systems. *Future Gener. Comput. Syst.* **2016**, *61*, 85–96. [CrossRef]
13. Zhang, Q.; Yang, L.T.; Chen, Z.; Li, P.; Bu, F. An adaptive dropout deep computation model for industrial IoT big data learning with crowdsourcing to cloud computing. *IEEE Trans. Ind. Inform.* **2018**. [CrossRef]
14. Martinovic, J.; Jorswieck, E.; Scheithauer, G.; Fischer, A. Integer linear programming formulations for cognitive radio resource allocation. *IEEE Wirel. Commun. Lett.* **2017**, *6*, 494–497. [CrossRef]
15. Hwang, T.; Yang, C.; Wu, G.; Li, S.; Li, G.Y. OFDM and its wireless applications: A survey. *IEEE Trans. Veh. Technol.* **2009**, *58*, 1673–1694. [CrossRef]
16. Basar, E. On multiple-input multiple-output OFDM with index modulation for next generation wireless networks. *IEEE Trans. Signal Process.* **2016**, *64*, 3868–3878. [CrossRef]
17. Goldsmith, A.; Jafar, S.A.; Maric, I.; Srinivasa, S. Breaking spectrum gridlock with cognitive radios: An information theoretic perspective. *Proc. IEEE* **2009**, *97*, 894–914. [CrossRef]
18. Cheng, P.; Zhang, Z.; Chen, H.H.; Qiu, P. Optimal distributed joint frequency, rate, and power allocation in cognitive OFDMA systems. *IET Commun.* **2008**, *2*, 815–826. [CrossRef]
19. Kazemi, J.; Omidi, M.J.; Navaie, K. A novel low complexity energy-efficient resource allocation for OFDM systems. *Trans. Emerg. Telecommun. Technol.* **2017**, *28*, e2932. [CrossRef]
20. Wang, Q.; Liu, H.L.; Li, J.; Li, Y. A resource allocation evolutionary algorithm for OFDM system. *Int. J. Comput. Sci. Eng.* **2017**, *14*, 55–63. [CrossRef]
21. Khan, H.; Yoo, S.J. Active interference restriction in OFDM-based cognitive radio network using genetic algorithm. In Proceedings of the International Conference on Information and Communication Technology Convergence, Jeju Island, Korea, 28–30 October 2015; pp. 840–842.
22. Xu, L.; Wang, J.; Li, Y.; Li, Q.; Zhang, X. Resource allocation algorithm based on hybrid particle swarm optimization for multiuser cognitive OFDM network. *Expert Syst. Appl.* **2015**, *42*, 7186–7194. [CrossRef]
23. Tang, M.; Xin, Y.; Long, C.; Wei, X.; Liu, X. Optimizing power and rate in cognitive radio networks using improved particle swarm optimization with mutation strategy. *Wirel. Pers. Commun.* **2016**, *89*, 1027–1043. [CrossRef]
24. Zhang, X.; Zhang, X. Population-adaptive differential evolution-based power allocation algorithm for cognitive radio networks. *EURASIP J. Wirel. Commun. Netw.* **2016**, *2016*, 219. [CrossRef]
25. Shang, R.; Zhang, K.; Jiao, L.; Wu, J. Discrete polynary coding immune clonal selection-based joint subcarrier and power allocation in uplink cognitive OFDM network. *Int. J. Commun. Syst.* **2016**, *29*, 64–83. [CrossRef]
26. Luo, X.; Jiang, C.; Wang, W.; Xu, Y.; Wang, J.H.; Zhao W. User behavior prediction in social networks using weighted extreme learning machine with distribution optimization. *Future Gener. Comput. Syst.* **2018**. [CrossRef]
27. Zhang, Q.; Yang, L.T.; Yan, Z.; Chen, Z.; Li, P. An efficient deep learning model to predict cloud workload for industry informatics. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3170–3178. [CrossRef]
28. Luo, X.; Deng, J.; Liu, J.; Wang, W.; Ban, X.; Wang, J.H. A quantized kernel least mean square scheme with entropy-guided learning for intelligent data analysis. *China Commun.* **2017**, *14*, 127–136. [CrossRef]

29. Dong, M.; Li, H.; Ota, K.; Yang, L.T.; Zhu, H. Multicloud-based evacuation services for emergency management. *IEEE Cloud Comput.* **2014**, *1*, 50–59. [CrossRef]
30. Luo, X.; Xu, Y.; Wang, W.; Yuan, M.; Ban, X.; Zhu, Y.; Zhao, W. Towards enhancing stacked extreme learning machine with sparse autoencoder by correntropy. *J. Frankl. Inst.* **2018**, *355*, 1945–1966. [CrossRef]
31. Luo, X.; Sun, J.; Wang, L.; Wang, W.; Zhao, W.; Wu, J.; Wang, J.H.; Zhang, Z. Short-term wind speed forecasting via stacked extreme learning machine with generalized correntropy. *IEEE Trans. Ind. Inform.* **2018**. [CrossRef]
32. Luo, X.; Yang, X.; Jiang, C.; Ban, X. Timeliness online regularized extreme learning machine. *Intl. J. Mach. Learn. Cybern.* **2018**, *9*, 465–476. [CrossRef]
33. Rao, R.V. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *Int. J. Ind. Eng. Comput.* **2016**, *7*, 19–34.
34. Luo, X.; He, Z.; Wang, L.; Wang, W.; Ning, H.; Wang, J.H.; Zhao, W. An efficient Jaya algorithm for resource allocation in the cognitive-radio-networks-aided internet of things. In Proceedings of the IEEE Conferences on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, Congress on Cybermatics, Halifax, NS, Canada, 30 July–3 August 2018; pp. 118–125.
35. Jang, J.; Lee, K.B. Transmit power adaptation for multiuser OFDM systems. *IEEE J. Sel. Areas Commun.* **2003**, *21*, 171–178. [CrossRef]
36. Zhang, R.; Cui, S.; Liang, Y.C. On ergodic sum capacity of fading cognitive multiple-access and broadcast channels. *IEEE Trans. Inf. Theory* **2009**, *55*, 5161–5178. [CrossRef]
37. Zhang, R.; Liu, Y.; Lin, X. OFDMA based resources allocations for downlink cognitive radio. *Acta Phys. Sin.* **2010**, *38*, 632–637.
38. Kang, X.; Liang, Y.; Nallanathan, A.; Garg, H.K.; Zhang, R. Optimal power allocation for fading channels in cognitive radio networks: Ergodic capacity and outage capacity. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 940–950. [CrossRef]
39. Shen, Z.; Andrews, J.G.; Evans, B.L. Adaptive resource allocation in multiuser OFDM systems with proportional rate constraints. *IEEE Trans. Wirel. Commun.* **2005**, *4*, 2726–2737. [CrossRef]
40. Huang, C.; Wang, L.; Yeung, R.S.C.; Zhang, Z.; Chung, H.S.H.; Bensoussan, A. A prediction model guided Jaya algorithm for the PV system maximum power point tracking. *IEEE Trans. Sustain. Energy* **2018**, *9*, 45–55. [CrossRef]
41. Chen, M.; Li, Y.; Luo, X.; Wang, W.; Wang, L.; Zhao, W. A novel human activity recognition scheme for smart health using multilayer extreme learning machine. *IEEE Internet Things J.* **2018**. [CrossRef]
42. Yu, K.; Liang, J.J.; Qu, B.Y.; Chen, X.; Wang, H. Parameters identification of photovoltaic models using an improved JAYA optimization algorithm. *Energy Convers. Manag.* **2017**, *150*, 742–753. [CrossRef]
43. Kaur, A.; Sharma, S.; Mishra, A. Sensing period adaptation for multiobjective optimisation in cognitive radio using Jaya algorithm. *Electron. Lett.* **2017**, *53*, 1335–1336. [CrossRef]
44. Aziz, N.A.A.; Ibrahim, Z.; Mubin, M.; Nawawi, S.W.; Mohamad, M.S. Improving particle swarm optimization via adaptive switching asynchronous-synchronous update. *Appl. Soft Comput.* **2018**, *72*, 298–311. [CrossRef]
45. Koh, B.I.; George, A.D.; Haftka, R.T.; Fregly, B.J. Parallel asynchronous particle swarm optimization. *Int. J. Numer. Methods Eng.* **2006**, *67*, 578–595. [CrossRef] [PubMed]
46. Venter, G.; Sobieszczanskisobieski, J. Parallel particle swarm optimization algorithm accelerated by asynchronous evaluations. *J. Aerosp. Comput. Inf. Commun.* **2005**, *3*, 123–137. [CrossRef]
47. Cavuoti, S.; Garofalo, M.; Brescia, M.; Pescape, A.; Longo, G.; Ventre, G. Genetic algorithm modeling with GPU parallel computing technology. *Neural Nets and Surroundings*; Smart Innovation, Systems and Technologies; Springer: Berlin/Heidelberg, Germany, 2013; pp. 29–39.
48. Wang, L.; Zhang, Z.; Huang, C.; Tsui, K.L. A GPU-accelerated parallel Jaya algorithm for efficiently estimating Li-ion battery model parameters. *Appl. Soft Comput.* **2018**, *65*, 12–20. [CrossRef]
49. Sun, J.; Wang, Z.; Luo, X.; Shi, P.; Wang, W.; Wang, L.; Wang, J.H.; Zhao, W. A parallel recommender system using a collaborative filtering algorithm with correntropy for social networks. *IEEE Trans. Netw. Sci. Eng.* **2018**. [CrossRef]

50. Zhang, C.F.; Zhao, X.H. Adaptive resource allocation in multiuser OFDM system based on fairness threshold. *J. Commun.* **2011**, *32*, 65–71.
51. Wong, I.C.; Shen, Z.; Evans, B.L.; Andrews, J.G. A low complexity algorithm for proportional resource allocation in OFDMA systems. In Proceedings of the IEEE Workshop on Signal Processing Systems Design and Implementation, Austin, TX, USA, 13–15 October 2004; pp. 1–6.

© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).