We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Chapter 1

Evolutionary Algorithms for Wireless Communications — A Review of the State-of-the art

Sotirios K. Goudos

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/59147

1. Introduction

Several evolutionary algorithms (EAs) have emerged in the past decade that mimic biological entities behavior and evolution. Darwin's theory of evolution is the major inspiration source for EAs. The foundation of Darwin's theory of evolution is natural selection. The study of evolutionary algorithms began in the 1960s. Several researchers independently developed three mainstream evolutionary algorithms, namely, genetic algorithms [1, 2], evolutionary programming [3], and evolution strategies [4]. EAs are widely used for the solution of single and multi-objective optimization problems. Swarm Intelligence (SI) algorithms are also a special type of EAs. SI can be defined as the collective behavior of decentralized and self-organized swarms. SI algorithms among others include Particle Swarm Optimization (PSO) [5], Ant Colony Optimization [6], and Artificial Bee Colony (ABC) [7].

PSO is an evolutionary algorithm that mimics the swarm behavior of bird flocking and fish schooling [5]. The most common PSO algorithms include the classical Inertia Weight PSO (IWPSO) and Constriction Factor PSO (CFPSO) [8]. PSO is an easy to implement algorithm with computational efficiency. The PSO algorithm is inherently used only for real-valued problems. An option to expand PSO for discrete valued problems also exists. Among others PSO algorithms include, the barebones (BB) and the exploiting barebones (BBExp). BBPSO has been successfully applied to the cell to switch assignment problem [9].

Artificial Bee Colony (ABC) [7] is a recently proposed SI algorithm, which has been applied to several real world engineering problems. The ABC algorithm models and simulates the honey bee behavior in food foraging. In the ABC algorithm, a potential solution to the optimization problem is represented by the position of a food source while the food source corresponds to the quality (objective function fitness) of the associated solution. The ABC algorithm has been

INTECH
open science | open minds

© 2014 The Author(s). Licensee InTech. This chapter is distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and eproduction in any medium, provided the original work is properly cited.

successfully applied to several problems in wireless communications [10]. ABC variants that improve the original algorithm have also been proposed [11].

Ant Colony Optimization (ACO) is a population-based metaheuristic introduced by Marco Dorigo [12]. This algorithm was inspired by the behaviour of real ants. The algorithm is based on the fact that ant colonies can find the shortest path between their nest and a food source just by depositing and reacting to pheromones while they are exploring their environment. ACO is suitable for solving combinatorial optimization problems, which are common in wireless communications.

Differential evolution (DE) [13, 14] is a population-based stochastic global optimization algorithm, which has been used in several real world engineering problems. Several DE variants or strategies exist. One of the DE advantages is that very few control parameters have to be adjusted in each algorithm run. However, the control parameters involved in DE are highly dependent on the optimization problem. Moreover, the selection of the appropriate strategy for trial vector generation requires additional computational time using a trial-and-error search procedure. Therefore, it is not always an easy task to fine-tune the control parameters and strategy. Since finding the suitable control parameter values and strategy in such a way is often very time-consuming, there has been an increasing interest among researchers in designing new adaptive and self-adaptive DE variants. Self adaptive DE (SaDE), is a DE algorithm that self-adapts both control parameters and strategy based on learning experiences from previous generations is presented in [15-17]. SaDE has been applied to microwave filter design, [18], and to antenna arrays synthesis [19].

The purpose of this chapter is to briefly describe the above algorithms and present their application to wireless communications optimization problems found in the literature. This chapter also presents results from different cases using PSO, ABC, ACO and DE. These include the cell to switch assignment problem in cellular networks using PSO algorithms, peak to average power ratio (PAPR) reduction of OFDM signals with the partial transmit sequences (PTS) approach using ABC and ACO algorithms [7, 11], and dual-band microwave filter design for wireless communications using SADE.

This chapter is subdivided into four sections. Section 2 presents the different evolutionary algorithms. Section 3 reviews the related work in wireless communications problems from the literature. Section 4 describes the design cases and presents the numerical results. Finally section 5 contains the discussion about the advantages of using a EA-based approach and the conclusions.

2. Methods

A population (or swarm) in PSO, ABC, ACO and DE consists of NP vectors (or particles) $\bar{x}_{G,i}$, i=1, 2, ..., NP, where G is the generation number. The population is initialized randomly from a uniform distribution. Each D-dimensional vector represents a possible solution, which is expressed as:

http://dx.doi.org/10.5772/59147

$$\overline{x}_{G,i} = (x_{G,1i}, x_{G,2i}, ...x_{G,ji}, ..., x_{G,Di}) \quad (1)$$

The population is initialized as follows:

$$x_{0,ji} = rand_{j[0,1)}\left(x_{j,U} - x_{j,L}\right) + x_{j,L} \quad j = 1,2,.....,D \quad (2)$$

where $x_{j,L}$ and $x_{j,U}$ are *D*-dimensional vectors of the lower and upper bounds respectively and $rand_{j[0,1)}$ is a uniformly distributed random number within [0,1). The stopping criterion for PSO, ABC and DE is usually the generation number or the number of objective-function evaluations.

2.1. Particle Swarm Optimization (PSO)

In PSO, the particles move in the search space, where each particle position is updated by two optimum values. The first one is the best solution (fitness) that has been achieved so far. This value is called *pbest*. The other one is the global best value obtained so far by any particle in the swarm. This best value is called *gbest*. After finding the *pbest* and *gbest*, the velocity update rule is an important factor in a PSO algorithm. The most commonly used algorithm defines that the velocity of each particle for every problem dimension is updated with the following equation:

$$u_{G+1,ni} = wu_{G,ni} + c_1 rand_{1(0,1)}(pbest_{G+1,ni} - x_{G,ni}) + c_2 rand_{2(0,1)}(gbest_{G+1,ni} - x_{G,ni}) \quad (3)$$

where $u_{G+1,ni}$ is the ith particle velocity in the nth dimension, *G+1* denotes the current iteration and *G* the previous, $x_{G,ni}$ is the particle position in the nth dimension, $rand_{1(0,1)}$, $rand_{2(0,1)}$ are uniformly distributed random numbers in (0,1), w is a parameter known as the inertia weight, and c_1 and c_2 are the learning factors.

The parameter w (inertia weight) is a constant between 0 and 1. This parameter represents the particle's fly without any external influence. The higher the value of w, or the closer it is to one, the more the particle stays unaffected from pbest and gbest. The inertia weight controls the impact of the previous velocity: a large inertia weight favors exploration, while a small inertia weight favors exploitation. The parameter c_1 represents the influence of the particle memory on its best position, while the parameter c_2 represents the influence of the swarm best position. Therefore, in the Inertia Weight PSO (IWPSO) algorithm the parameters to be determined are: the swarm size (or population size), usually 100 or less, the cognitive learning factor c_1 and the social learning factor c_2 (usually both are set to equal to 2.0), the inertia weight w, and the maximum number of iterations. It is common practice to linearly decrease the inertia weight starting from 0.9 or 0.95 to 0.4.

Clerc [8] suggested the use of a different velocity update rule, which introduced a parameter K called constriction factor. The role of the constriction factor is to ensure convergence when all the particles have stopped their movement. The velocity update rule is then given by:

$$u_{G+1,ni} = K\left[u_{G,ni} + c_1 rand_{1(0,1)}(pbest_{G+1,ni} - x_{G,ni}) + c_2 rand_{2(0,1)}(gbest_{G+1,ni} - x_{G,ni})\right] \tag{4}$$

$$K = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|} \tag{5}$$

where $\varphi = c_1 + c_2$ and $\varphi > 4$.This PSO algorithm variant is known as Constriction Factor PSO (CFPSO).

2.2. Barebones PSO

Kennedy [20] proposed a new PSO approach, the BB PSO, where the standard PSO velocity equation is replaced with samples from a normal distribution. In this method, the position update rule for the ith particle in the nth dimension becomes

$$x_{G+1,ni} = N\left(\frac{pbest_{G+1,ni} + gbest_{G+1,ni}}{2}, \left|pbest_{G+1,ni} - gbest_{G+1,ni}\right|\right) \tag{6}$$

$N(\ ,\ )$ denotes the normal distribution. The method allows particles with *pbest* significant different from *gbest* to make large step sizes towards it. When *pbest* is close to *gbest*, step size decreases and limits exploration in favor of exploitation.

In [20], a variation of BB PSO, the BBExp PSO, was also proposed. In this method, approximately half of the time velocity is based on samples from a normal distribution; for the rest of the time, velocity is derived from the particle's personal best position. The position update rule, (6), is modified into

$$x_{G+1,ni} = \begin{cases} N\left(\dfrac{pbest_{G+1,ni} + gbest_{G+1,ni}}{2}, \left|pbest_{G+1,ni} - gbest_{G+1,ni}\right|\right), & U(0,1) > 0.5 \\ pbest_{G+1,ni}, & \text{otherwise} \end{cases} \tag{7}$$

where $U(\ ,\ )$ denotes the uniform distribution. In BBExp PSO, position updates equal $pbest_n$ for half of the time resulting in the improved exploitation of $pbest_n$ compared to the BB PSO. One may notice

that the barebones PSO algorithms do not require parameter tuning. More details can be found in [20, 21].

2.3. Artificial bee colony optimization

The ABC algorithm models and simulates the honey bee behavior in food foraging. In ABC algorithm, a potential solution to the optimization problem is represented by the position of a food source while the nectar amount of a food source corresponds to the quality (objective function fitness) of the associated solution. In order to find the best solution the algorithm defines three classes of bees: employed bees, onlooker bees and scout bees. The employed bee searches for the food sources, the onlooker bee makes a decision to choose the food sources by sharing the information of employed bee, and the scout bee is used to determine a new food source if a food source is abandoned by the employed bee and onlooker bee. For each food source exists only one employed bee (i.e. the number of the employed bees is equal to the number of solutions). The employed bees search for new neighbor food source near of their hive. A new position of the $\bar{x}_i=(x_{i,1},\ ..,\ x_{i,j},\ ..,\ x_{i,D})$ solution, where D is the problem dimension, is generated using

$$u_{i,j} = x_{i,j} + \varphi_{i,j}\left(x_{i,j} - x_{k,j}\right) \tag{8}$$

where $k \in \{1, 2, .., SN\}$, $k \neq i$, $j \in \{1, 2, .., D\}$ are randomly chosen indices, where SN is the number of food sources, and $\varphi_{i,j}$ is a uniformly distributed random number within [-1,1]. ABC uses a greedy selection operator, which for minimization problems is defined by

$$\bar{x}_i' = \begin{cases} \bar{u}_i, & \text{if } f(\bar{u}_i) < f(\bar{x}_i) \\ \bar{x}_i, & \text{otherwise} \end{cases} \tag{9}$$

where $\bar{x}_i'$ is the new position of the food source.

An onlooker bee chooses a food source depending on the probability value associated with that food source, p_i, given by

$$p_i = \frac{fit_i}{\sum_{m=1}^{SN} fit_m} \tag{10}$$

where fit_i is the fitness value of the ith solution which is proportional to the nectar amount of the food source in the ith position. When a food source (solution) cannot be improved anymore then the scout bee helps the colony to randomly generate create new solutions

$$x_{i,j} = rand_{j(0,1)}\left(x_{j,U} - x_{j,L}\right) + x_{j,L} \quad j = 1, 2, ..., D \tag{11}$$

where $x_{j,L}$ and $x_{j,U}$ are the lower and upper bounds of the jth dimension respectively and $rand_{j(0,1)}$ is a uniformly distributed random number within (0,1).

2.4. Ant colony optimization

Ant colony optimization (ACO) [6, 12, 22] is a meta-heuristic inspired by the ants' foraging behavior. At the core of this behavior is the indirect communication between the ants by means of chemical pheromone trails, which enables them to find short paths between their nest and food sources. Ants can sense pheromone. When they decide path to follow a path, they tend to choose the ones with strong pheromone intensities way back to the nest or to the food source. Therefore, shorter paths would accumulate more pheromone than longer ones. This feature of real ant colonies is exploited in ACO algorithms in order to solve combinatorial optimization problems considered to be NP-Hard.

2.5. Differential Evolution (DE)

The initial population evolves in each generation with the use of three operators: mutation, crossover and selection. Depending on the form of these operators several DE variants or strategies exist in the literature [14, 23]. The choice of the best DE strategy depends on problem type [24]. In SaDE the following four strategies are used for trial vector generation. These include *DE/rand/1bin*, *DE/rand-to-best/2/bin*, *DE/rand/2/bin*, and *DE/current-to-rand/1* [25]. In these strategies a mutant vector $\bar{v}_{G+1,i}$ for each target vector $\bar{x}_{G,i}$ is computed by:

$$
\begin{aligned}
&DE/rand/1/bin\\
&\overline{v}_{G+1,i} = \overline{x}_{G,r_1} + F(\overline{x}_{G,r_2} - \overline{x}_{G,r_3}),\quad r_1 \neq r_2 \neq r_3\\
&DE/rand-to-best/2/bin\\
&\overline{v}_{G+1,i} = \overline{x}_{G,i} + F(\overline{x}_{G,best} - \overline{x}_{G,i}) + F(\overline{x}_{G,r_1} - \overline{x}_{G,r_2}) + F(\overline{x}_{G,r_3} - \overline{x}_{G,r_4}),\quad r_1 \neq r_2 \neq r_3 \neq r_4\\
&DE/rand/2/bin\\
&\overline{v}_{G+1,i} = \overline{x}_{G,r_1} + F(\overline{x}_{G,r_2} - \overline{x}_{G,r_3}) + F(\overline{x}_{G,r_4} - \overline{x}_{G,r_5}),\quad r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5\\
&DE/current-to-rand/1/bin\\
&\overline{v}_{G+1,i} = \overline{x}_{G,i} + K(\overline{x}_{G,r_1} - \overline{x}_{G,i}) + F(\overline{x}_{G,r_2} - \overline{x}_{G,r_3}),\quad r_1 \neq r_2 \neq r_3
\end{aligned}
\tag{12}
$$

where r_1, r_2, r_3, r_4, r_5 are randomly chosen indices from the population, which are different from index i, F is a mutation control parameter, K a coefficient responsible for the level of recombination that occurs between $\bar{x}_{G,i}$ and $\bar{x}_{G,r_1}$. After mutation, the crossover operator is applied to generate a trial vector $\bar{u}_{G+1,i}=(u_{G+1,1i}, u_{G+1,2i}, \ldots u_{G+1,ji}, \ldots\ldots, u_{G+1,Di})$ whose coordinates are given by:

$$
u_{G+1,ji} = \begin{cases} v_{G+1,ji}, & \text{if } rand_{j[0,1)} \leq CR \text{ or j=rn(i)} \\ x_{G+1,ji}, & \text{if } rand_{j[0,1)} > CR \text{ and j} \neq \text{rn(i)} \end{cases}
\tag{13}
$$

where $j=1, 2, \ldots\ldots, D$, $rand_{j[0,1)}$ is a number from a uniform random distribution from the interval [0,1), rn(i) a randomly chosen index from (1, 2, ..., D), and CR the crossover constant from the interval [0,1]. DE uses a greedy selection operator, which for minimization problems is defined by:

$$\bar{x}_{G+1,i} = \begin{cases} \bar{u}_{G+1,i}, & \text{if } f(\bar{u}_{G+1,i}) < f(\bar{x}_{G,i}) \\ \bar{x}_{G,i}, & \text{otherwise} \end{cases} \tag{14}$$

where $f(\bar{u}_{G+1,i})$, $f(\bar{x}_{G,i})$ are the fitness values of the trial and the old vector respectively. Therefore, the newly found trial vector $\bar{u}_{G+1,i}$ replaces the old vector $\bar{x}_{G,i}$ only when it produces a lower objective-function value than the old one. Otherwise, the old vector remains in the next generation. The stopping criterion for the DE is usually the generation number or the number of objective-function evaluations.

2.6. Self-Adaptive DE (SADE)

In the SaDE algorithm both the trial vector generation strategies and the control parameters are self-adapted according to previous learning experiences. SaDE maintains a strategy candidate pool, consisting of the four strategies given in (2). Each strategy is assigned a certain probability. The sum of all probabilities is equal to one. These probabilities are initialized with a value of 0.25 and gradually adapted during evolution. The probability of applying the *m-th* strategy is p_m, $m=1, 2, \ldots\ldots, M$, where M is the total number of strategies. At generation G, the number of successful trial vectors generated by the *m-th* strategy is denoted as $ns_{m,G}$, while the number of trial vectors that fail to replace the old vectors in the next generation is $nf_{m,G}$. An additional parameter called the learning period (LP) is introduced in [17]. This corresponds to the number of the previous generations that store the success and fail statistics. After LP generations, the probabilities of selecting different strategies are updated according to:

$$p_{m,G} = \frac{S_{m,G}}{\sum_{m=1}^{M} S_{m,G}}$$

where

$$S_{m,G} = \frac{\sum_{g=G-LP}^{G-1} ns_{m,g}}{\sum_{g=G-LP}^{G-1} ns_{m,g} + \sum_{g=G-LP}^{G-1} nf_{m,g}} + \varepsilon \tag{15}$$

where $S_{m,G}$ is the success rate of the trial vectors generated by the *m-th* strategy within the previous LP generation and ε is a constant set equal to 0.01 to avoid possible null success rates.

Therefore, according to (5) strategies with high success rates have higher probability to be applied at the current generation.

The control parameters are self-adapted in the following way. The mutation control parameter F is approximated by a normal distribution with mean value 0.5 and standard deviation 0.3, that is $N(0.5, 0.3)$. The parameter K is a random number in the interval [0, 1] generated by a uniform distribution. The crossover rate control parameter CR used by the *m-th* strategy is also approximated by a normal distribution with mean value CR_m and standard deviation 0.1, that is $N(CR_m, 0.1)$. The initial value of CR_m is 0.5 for all strategies. The values of crossover rates that have successfully generated trial vectors in the previous LP generations are stored in a crossover rate memory for each strategy ${CR_m}^{memory}$ that is an array of size LP. At each generation, the median value stored in memory for the *m-th* strategy ${CR_m}^{median}$ is calculated and the CR values generated are given by a normal distribution with mean value ${CR_m}^{median}$ and standard deviation 0.1. That way the crossover values are evolved at each generation to follow the successful values found. The authors in [17] suggest a value between 20 and 60 for the parameter LP. The sensitivity analysis performed in [17] for the LP parameter showed it had no significant impact on SaDE performance. More details about the SaDE algorithm can be found in [17].

3. Related work

This section presents a brief literature review of applications of evolutionary algorithms and their variants to wireless communications problems.

Genetic algorithms (GAs) are among the widely used optimization techniques for addressing design problems in wireless communications. In [26] a Smith prediction filter is proposed for power control design of direct-sequence code-division multiple-access cellular mobile radio systems. A fixed-order robust H∞ loop filter is developed using a genetic algorithm to minimize the worst-case variance of the received SINR from the minimax perspective. The authors in [27] present an antenna selection method for multiple-input multiple-output wireless systems based on a GA that seeks the best subset of antenna elements. The problem of receive antenna selection and symbol detection for multiple-input, multiple-output (MIMO) systems is solved in [28] by applying a genetic algorithm (GA) variant. The paper in [29] addresses the problem of joint transmit/receive antenna selection for MIMO systems using a real-valued genetic algorithm (RVGA). The optimization objective is to improve the channel capacity of multiple-input/multiple-output (MIMO) systems. The study in [30] presents a pattern discovery algorithm for multi-streams mining in wireless sensor networks. This algorithm adapts genetic operators with Elitism Strategy.The paper in [31] studies joint multiuser linear precoding design in the forward link of fixed multibeam satellite systems. The authors use a generic optimization framework for linear precoding design to handle any objective functions of data rate with general linear and nonlinear power constraints. In [32] an energy-efficient genetic algorithm mechanism

is presented to resolve quality of service (QoS) multicast routing problem: The proposed genetic algorithm depends on bounded end-to-end delay and minimum energy cost of the multicast tree.

SI algorithms are among the most commonly used algorithms for solving problems in wireless communications. PSO and several PSO variants have been used in the literature to solve different problems. In [33] optimal power scheduling for distributed detection in a Gaussian sensor network is addressed for both independent and correlated observations. A PSO based technique is developed to find the optimal power allocation for arbitrary correlations. The authors in [34] apply PSO to solve the constrained nonlinear optimisation problem for the minimum bit error rate (MBER) multiuser transmitter (MUT). The proposed PSO aided symbol-specific MBER-MUT and average MBER-MUT schemes provide improved performance in comparison to the conventional minimum mean-square error MUT scheme. Several issues in Wireless Sensor Networks (WSNs) can be formulated as multidimensional optimization problems, and addressed using bio-inspired algorithms. In [35] the authors present a brief survey of how PSO is used to address these issues. The authors in [36] propose a new approach to estimate the location of a sensor in a wireless sensor network based on a new PSO algorithm with a log-barrier approach. The paper in [37] presents a predistorter based on a cluster-based implementation particle swarm optimization technique with embedded model-size estimation capability and validates the proposed technique on a Doherty power amplifier prototype.

The ABC algorithm and its variants have been successfully applied to several optimization problems in wireless communications. Among others these include issues in WSN [38-41], WiMax network planning [42] and channel assignment [43].

The ACO algorithm has also been applied to several combinatorial problems in wireless systems which include problems in mobile ad hoc networks [44-46], problems in WSN [47-51], cognitive radio [52], resource allocation in multiuser OFDM systems [53] and MIMO problems [54, 55].

DE variants have also been applied to variety of optimization problems like multi-user detection in multi-carrier CDMA [56], WSNs issues [57, 58], urban area path loss prediction [59], spectrum sharing [60], optimization of interleave-division multiple-access communication systems [61]. Other papers use a number of different optimization algorithms and compare results. For example in [62] spectrum allocation methods for cognitive radio based on GA, quantum genetic algorithm (QGA), and PSO, are proposed.

4. Results and discussions

In this section we present numerical results from different optimization problems in wireless communications using different algorithms.

4.1. Cell to switch assignment in cellular networks using barebones PSO

Cell assignment is an important issue in the area of resource management in cellular networks. The problem is an NP-hard one and requires efficient search techniques for its solution in real-time. We briefly present an example case of solving this problem using the barebones PSO [9]. The effective assignment of cells to switches in order to minimize the cost of network deployment is a challenging issue in cellular networking.

The cell-to-switch assignment (CSA) problem consists of optimally assigning cells to network switches while respecting certain constraints such as the call volume of each cell and the switches capacity [63]. The objective of the optimization is the reduction of implementation and operational costs. Usually, the cost function considers the cost of linking cells to switches (cabling cost) and the cost of handoff between different cells (handoff cost). The problem is an NP-hard one with exponential complexity and cannot be solved analytically in real size networks. Other evolutionary techniques like tabu search [64], ACO [65] and GAs [66] have been used in the literature for solving this problem. The problem formulation is given below.

We consider n unique and distinct cells in a given service area and m switches with known location and traffic parameters. The objective is the optimum assignment of cells to switches in order to minimize the total cost that comprises the handoff and cabling costs.

In single-homing CSA, each cell belongs to one cluster and it is assigned to one switch at a time. In this case, the objective function to be minimized is [63]:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} c_{ik}x_{ik} + \sum_{i=1}^{n}\sum_{\substack{j=1 \\ j\neq i}}^{n} h_{ij}\left(1-y_{ij}\right), \quad k=1,...,m \tag{16}$$

where c_{ik} is the cabling cost per time unit between cell i and switch k, x_{ik} is a parameter that takes the value one when cell i is assigned to switch k (otherwise, x_{ik}=0) and h_{ij} is the cost per time unit for the handoffs that occur between cells i and j. The first term in (16) gives the total cabling cost and the second one is the total handoff cost per time unit among cells. Therefore, y_{ij} is defined as

$$y_{ij} = \sum_{k=1}^{m} x_{ik}x_{jk}, \quad i,j=1,...,n \tag{17}$$

where y_{ij} is one when cells i and j are connected to the same switch, otherwise it is zero. The product $x_{ik}x_{jk}$ in (17) defines the variable

$$z_{ijk}=x_{ik}x_{jk} \quad i,j=1,...,n \text{ and } k=1,...,m \tag{18}$$

that is zero unless cells i and j are connected to switch k. In this case, it is one.

http://dx.doi.org/10.5772/59147

Cell assignment is subject to further constraints. The call handling capacity of each switch should not be violated at any time, i.e.

$$\sum_{k=1}^{n} \lambda_i x_{ik} \le M_k, \quad i = 1,...,n \tag{19}$$

where λ_i is the number of calls that cell i handles per unit time and M_k is the call handling capacity of switch k. Also, each cell is assigned only to one switch, i.e.

$$\sum_{k=1}^{m} x_{ik} = 1, \quad i = 1,...,n \tag{20}$$

The optimization problem defined by (16) and subject to (17)-(20), can be converted [63] to an integer programming one by replacing (18) with the

$$\begin{aligned} &0 \le z_{ijk} \le x_{jk}, x_{ik} \\ &z_{ijk} \ge x_{ik} + x_{jk} - 1, \end{aligned} \quad i, j = 1,...,n \text{ and } k = 1,...,m \tag{21}$$

We compare BB and BBExp PSO with ACO [65] and binary PSO (BPSO) [67]. We run 100 independent trials for each algorithm. The statistical results are presented and compared In the proposed barebones PSO variants, the only parameter we set was the swarm size. In the examples presented here, this was set to 5 particles. The ACO parameters were the same as in [65]. For the BPSO, we set the learning factors c_1 and c_2 equal to two. Systems with varied number of cells and switches that range from 15 to 200 and from 2 to 7, respectively, were considered.

The percentage of successfully obtained solutions as a function of the number of cells and switches indicates the solution accuracy of the algorithms. Figure 1 shows the results of the application of BB PSO, BBExp PSO, BPSO and ACO in a single-homing system for swarm size equal to 5 particles. In the first case, BB PSO outperforms the other methods in systems with small complexity but as the complexity increases (n/m=150/6 and 200/7) BBExp gives the best results. As it was expected, solution accuracy decreases with system complexity. In general, BB PSO outperforms the other methods for small n and m. However, its performance degrades with system complexity; in this case, BBExp gives better results. In any case, at least one of BB and BBExp PSO is better than BPSO and ACO. We have also evaluated the different algorithms in terms of the computational time required for the derivation of the previous results. Figure 2 shows small differences in results between the four algorithms. In all cases, barebones PSO algorithms are slightly faster. BBExp outperforms BB as system complexity increases. The computational cost of the methods grows exponentially with n/m and increases with the swarm size. More details about this problem can be found in [9].

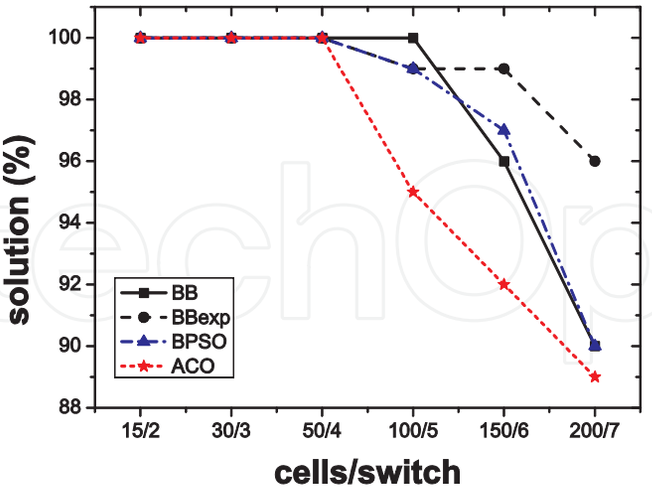


Figure 1. Successful solutions vs cells/switch

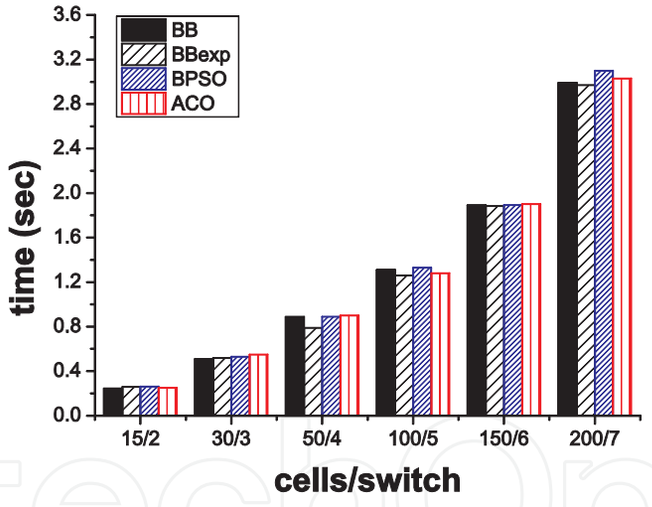


Figure 2. Computational time vs cells/switch chart

4.2. A PTS technique based on ACO and ABC for PAPR reduction of OFDM signals

A major drawback of OFDM signals is the high value of peak to average power ratio (PAPR). Partial transmit sequences (PTS) [68], is a popular PAPR reduction method with good PAPR reduction performance. However, PTS requires an exhaustive search in order to find the optimal phase factors. Thus, the search complexity is high. Several methods have been published in the literature for PAPR reduction using PTS with low search complexity [10, 69, 70]. The problem description is given below.

http://dx.doi.org/10.5772/59147

In an OFDM system, the high-rate data steam is split into N low-rate data streams that are simultaneously transmitted using N subcarriers. The discrete-time signal of such a system is given by

$$s_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} S_n e^{\frac{j2\pi nk}{LN}} \quad k = 0,1,....,LN-1 \tag{22}$$

where L is the oversampling factor, $\boldsymbol{S} = [S_0, S_1, ..., S_{N-1}]^T$ is the input signal block. Each symbol is modulated by either phase-shift keying (PSK) or quadrature amplitude modulation (QAM).

The PAPR of the signal in (22) is defined as the ratio of the maximum to average power and is expressed in dB as

$$PAPR(s) = 10\log_{10} \frac{\max_{0 \le k \le LN-1} \left|s_k\right|^2}{E\left[\left|s_k\right|^2\right]} \tag{23}$$

where $E[.]$ is the expected value operation.

In the PTS approach the $\boldsymbol{S}$ input data OFDM block is partitioned into M disjointed subblocks represented by the vector $\boldsymbol{S}_m$ $m = 1, 2, ..., M-1$ and oversampled by inserting $(L-1)N$ zeros. Then the PTS process is expressed as

$$\boldsymbol{S} = \sum_{m=1}^{M} \boldsymbol{S}_m \tag{24}$$

Next, the subblocks are converted to time domain using LN point inverse fast fourier transform (IFFT). The representation of the OFDM block in time domain is expressed by

$$\boldsymbol{s} = IFFT\left\{\sum_{m=1}^{M} \boldsymbol{S}_m\right\} = \sum_{m=1}^{M} IFFT\left\{\boldsymbol{S}_m\right\} = \sum_{m=1}^{M} \boldsymbol{s}_m \tag{25}$$

The PTS objective is to produce a weighted combination of the M subblocks using $\boldsymbol{b} = [b_1, b_2, ..., b_M]^T$ complex phase factors to minimize PAPR. The transmitted signal in time domain after this combination is given by

$$\boldsymbol{s}'(\boldsymbol{b}) = \sum_{m=1}^{M} b_m \boldsymbol{s}_m \tag{26}$$

In order to reduce the search complexity the phase factor possible values are limited to a finite set. The set of allowable phase factors is

$$A = \left\{ e^{\frac{j2\pi n}{W}} \middle| n = 0,1,...,W-1 \right\} \tag{27}$$

where W is the number of allowed phase factors. Therefore in case of M subblocks and W phase factors the total number of possible combinations is W^M. In order to reduce the search complexity we usually set fixed one phase factor.

The optimization goal of the PTS scheme is to find the optimum phase combination for minimum PAPR. Thus, the objective function can be expressed as

Minimize

$$F(\boldsymbol{b}) = 10\log_{10} \frac{\max_{0 \le k \le LN-1} \left| \boldsymbol{s}'(\boldsymbol{b}) \right|^2}{E\left[\left| \boldsymbol{s}'(\boldsymbol{b}) \right|^2 \right]} \tag{28}$$

subject to

$$\boldsymbol{b} \in \left\{ e^{j\phi_m} \right\}^M \text{ where } \phi_m \in \left\{ \frac{2\pi l}{W} \middle| l = 0,1,...,W-1 \right\} \tag{29}$$

We have evaluated objective function above using evolutionary algorithms and methods found in the literature. We have used two main measurement criteria namely the complementary cumulative distribution function (CCDF) and the computational complexity. In all our simulations, 10E5 random OFDM blocks are generated. The transmitted signal is oversampled by a factor L=4. We consider 16-QAM modulation with N=256 sub-carriers which are divided into M=16 random subblocks. The phase factors for W=2 are selected. We consider the first phase factor to be fixed so the total number of unknown phase factors is M-1.

The control parameters in all simulations are given below. In the PSO algorithm c_1 and c_2 are set equal to 2.05 while the inertia weight is linearly decreased starting from 0.9 to 0.4. For ACO the initial pheromone value τ_0 is set to 1.0e-6, the pheromonone update constant Q is set to 20, the exploration constant q_0 is set to 1, the global pheromone decay rate ρ_g is 0.9, the local pheromone decay rate ρ_l is 0.5, the pheromone sensitivity α is 1, and the visibility sensitivity is β is 5.

Figure 3 presents the comparison between the CCDF by different PTS reduction techniques. For $\Pr(PAPR > PAPR_0) = 10^{-3}$ the PAPR of the original OFDM transmitted signal is 10.84dB. For all evolutionary algorithms, the population size NP is set to 30 and the maximum number of

http://dx.doi.org/10.5772/59147

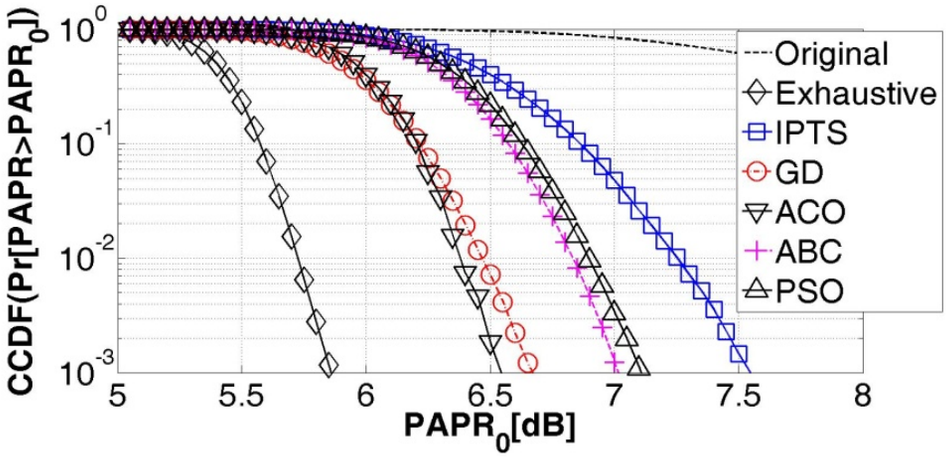


Figure 3. PARP reduction performance comparison of the BBO-PTS algorithms with other PTS schemes for NP=30, G=30.

generations G is set to 30. Thus, the computational complexity of this case is $NP \times G = 900$. The computational complexity of the exhaustive search is $W^{M-1} = 32768$ while the PAPR for this case is 5.86dB. The PAPR by the iterative flipping algorithm for PTS (IPTS) [69] is 7.55dB with search complexity (M-1)W=30. The PAPR by the gradient descent method (GD) [71] with search complexity $C_{M-1}^{r} W^{r} I = C_{15}^{2} 2^{2} 3 = 1260$ is 6.96dB. The PAPR by ABC [10], PSO [72], and ACO [6], is 7.01dB, 7.13dB, and 6.52dB, respectively. Table 1 holds the comparison of the search complexity among the different methods for $CCDF = 10^{-3}$, NP=30, and G=30. It is obvious that ACO presents the better performance among the other methods with the same search complexity.

Method	Computational Complexity	PAPR (dB)
Original	0	10.84
Exhaustive	$W^{M-1} = 32768$	5.86
IPTS	$(M-1)W = 30$	7.55
GD	$C_{M-1}^{r} W^{r} I = C_{15}^{2} 2^{2} 3 = 1260$	6.96
PSO	$NP \times G = 900$	7.13
ABC	$NP \times G = 900$	7.01
ACO	$NP \times G = 900$	6.52

Table 1. Comparison of computational complexity for CCDF=1e-3 among different PTS Schemes

4.3. Dual-band microwave filter design using SADE

Microwave filters are among the important components of a modern wireless communication system. Several papers exist in the literature that address the filter design problem [73]. Open

Loop Ring Resonator (OLRR) filters, which consist of two uniform microstrip lines and pairs of open loops between them, are widely used as the building block in several multiband bandpass filter design cases [74]. In [74], two pairs of folded OLRRs operating at two passbands are proposed to produce dual-band response.

A dual band OLRR filter is shown in Figure 4. The frequency response of such a filter depends on the filter dimensions and spacings between microstrip lines [74, 75]. The design parameters for this case are the ones shown in Figure 4, (W_1, W_2, L_1, L_2, L_3, L_4, L_5, S_1, S_2, S_3, G), all expressed in mm.

Such a filter design problem can be defined by the minimization of $|S_{11}|$ in the passband frequency range. This design problem is therefore defined by the minimization of the objective function:

$$F(\bar{x}) = 20\log\left\{\max\left|S_{11}(\bar{x}, f)\right|,\ f \in S_p\right\} \tag{30}$$

where $\bar{x}$ is the vector of filter geometry parameters and S_p is the set of distinct frequencies in the desired passband frequency ranges.

The filter is designed for operation in two WiMax (IEEE 802.16) frequency bands. These are the 3.5GHz and the 5.8GHz frequency bands. For this case, we set S_p={3.55, 3.6, 5.75, 5.8}. Figure 5 shows the simulated frequency response of this design. The simulated current distribution for the 3.6GHz and 5.8GHz frequencies is presented in Figure 6, where the resonating ring in each case is clearly seen. In the first passband between 3.508 and 3.809 GHz, the filter has a return loss less than 10dB and insertion loss greater than 0.5dB. In the second passband between 5.744 and 6.121 GHz the results also show a return loss less than 10dB and insertion loss greater than 0.5dB. The rejection band (between 4.236 and 5.367 GHz) has an insertion loss less than 20dB. In the first passband the return loss is less than 29dB at both 3.533 GHz and 3.759 GHz. In the second passband the return loss is less than 22dB at 5.794GHz and less than 28dB at 6.07GHz.

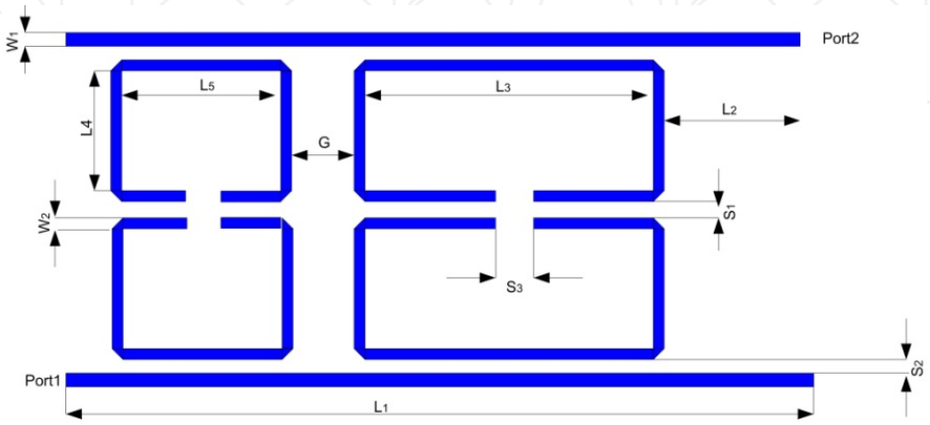


Figure 4. Dual-band filter geometry

http://dx.doi.org/10.5772/59147

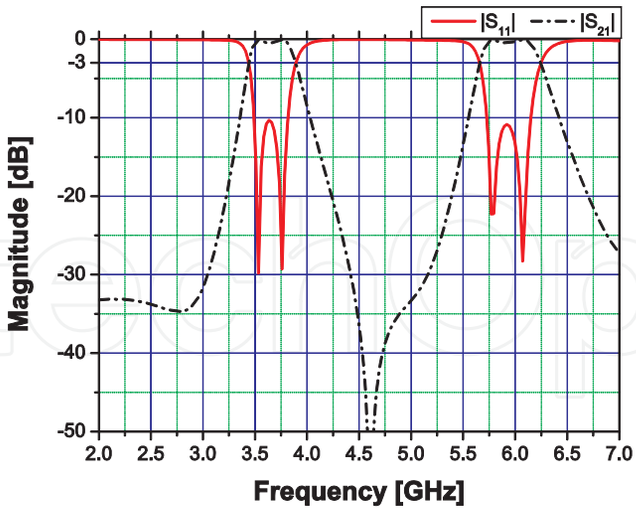
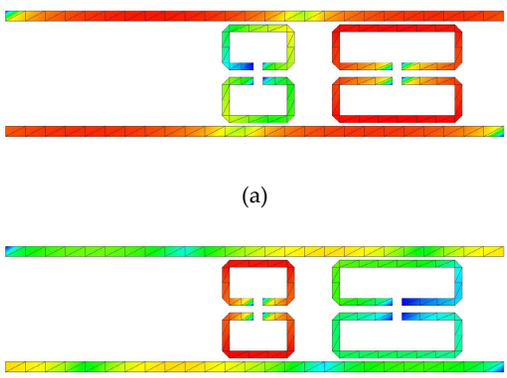


Figure 5. Simulated frequency response of the dual-band filter.



(a)

(b)

Figure 6. Current distribution simulations for dual-band filter at (a) 3.5GHz and (b) 5.8GHz

5. Conclusion

A brief survey of different evolutionary algorithms and their application to different problems in wireless communications has been presented. It must be pointed out that several evolutionary algorithms exist in the literature. GAs and SI algorithms are among those most commonly used. In order to select, the best algorithm for every problem one has to consider the problem characteristics. Another key issue is the selection of the algorithm control parameters, which is also in most cases problem-dependent. One may also use at first the control parameters for these algorithms that commonly perform well regardless of the characteristics of the problem to be solved. The example of the CSA problem in cellular

networks showed the better performance of BB and BBExp compared to BPSO and ACO in terms of successfully obtained solutions and execution time. In the PTS optimization problem ACO outperformed ABC and PSO.

The selection of the SADE technique for microwave filter design has lead to a successful filter design which exhibits low loss in the passbands and high isolation between the passbands. The DE algorithms are also robust optimizers. In classical DE algorithms, the selection of the appropriate strategy for trial vector generation and control parameters requires additional computational time using a trial-and-error search procedure. Therefore, it is not always an easy task to fine-tune the control parameters and strategy given also that commonly the appropriate control parameters and strategy selection are problem dependent. The SaDE advantage though, is the fact that no additional time for solving a given problem is required. SaDE requires only the adjustment of two parameters: the population size and the number of iterations.

The practical examples subject to several constraints presented in this chapter show the applicability and the efficiency of using such algorithms.

Author details

Sotirios K. Goudos*

Address all correspondence to: sgoudo@physics.auth.gr

Radiocommunications Laboratory, Section of Applied and Environmental Physics, Department of Physics, Aristotle University of Thessaloniki, Thessaloniki, Greece

References

[1] Goldberg D. E. Genetic algorithms in search, optimization and machine learning. New York:Addison Wesley;1989.

[2] Holland J. H. Adaptation in natural and artificial systems. Ann Arbor: The University of Michigan Press;1975.

[3] Fogel D. B. Evolutionary computation: Toward a new philosophy of machine intelligence. IEEE Press, Piscataway, NJ;1995.

[4] Beyer H. G. and Schwefel H. P. Evolution strategies: A comprehensive introduction. Natural Computing 2002;1 (1) 3-52.

[5] Kennedy J. and Eberhart R., "Particle swarm optimization", in IEEE International Conference on Neural Networks, pp. 1942-1948, Piscataway, NJ, 1995.

[6] Dorigo M. and Stutzle T. Ant colony optimization. Cambridge, MA:The MIT Press; 2004.

[7] Karaboga D. and Basturk B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (abc) algorithm. Journal of Global Optimization 2007;39 (3) 459-471.

[8] Clerc M., "The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization", in Proceedings of the 1999 Congress on Evolutionary Computation, 1999. CEC 99 pp. 1951-1957, Washington, DC, 1999.

[9] Goudos S. K., Baltzis K. B., Bachtsevanidis C. and Sahalos J. N. Cell-to-switch assignment in cellular networks using barebones particle swarm optimization. IEICE Electron. Express 2010;7 (4) 254-260.

[10] Wang Y., Chen W. and Tellambura C. A papr reduction method based on artificial bee colony algorithm for ofdm signals. IEEE Transactions on Wireless Communications 2010;9 (10) 2994-2999.

[11] Zhu G. and Kwong S. Gbest-guided artificial bee colony algorithm for numerical function optimization. Applied Mathematics and Computation 2010;217 (7) 3166-3173.

[12] Dorigo M., Maniezzo V. and Colorni A. Ant system: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 1996;26 (1) 29-41.

[13] Storn R. and Price K., "Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces", in Tech. Rep. TR-95-012, Berkeley, CA, 1995.

[14] Storn R. and Price K. Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 1997;11 (4) 341-359.

[15] Qin A. K. and Suganthan P. N., "Self-adaptive differential evolution algorithm for numerical optimization", in The 2005 IEEE Congress on Evolutionary Computation, 2005., ed. P. N. Suganthan, pp. 1785-1791, 2005.

[16] Huang V. L., Qin A. K. and Suganthan P. N., "Self-adaptive differential evolution algorithm for constrained real-parameter optimization", in IEEE Congress on Evolutionary Computation, 2006. CEC 2006., ed. A. K. Qin, pp. 17-24, 2006.

[17] Qin A. K., Huang V. L. and Suganthan P. N. Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Transactions on Evolutionary Computation 2009;13 (2) 398-417.

[18] Goudos S. K., Zaharis Z. D. and Yioultsis T. V. Application of a differential evolution algorithm with strategy adaptation to the design of multi-band microwave filters for wireless communications. Progress in Electromagnetics Research 2010;109 (123-137.

[19] Goudos S. K., Siakavara K., Samaras T., Vafiadis E. E. and Sahalos J. N. Sparse linear array synthesis with multiple constraints using differential evolution with strategy adaptation. IEEE Antennas and Wireless Propagation Letters 2011;10 (670-673.

[20] Kennedy J. Bare bones particle swarms. Proceedings of the IEEE Swarm Intelligence Symposium 2003 (SIS 2003) 2003;80-87.

[21] Pan F., Hu X., Eberhart R. and Chen Y., "An analysis of bare bones particle swarm", 2008.

[22] Dorigo M. and Gambardella L. M. Ant colonies for the travelling salesman problem. BioSystems 1997;43 (2) 73-81.

[23] Storn R. Differential evolution research-trends and open questions. Studies in Computational Intelligence 2008;143 (1-31.

[24] Mezura-Montes E., Velazquez-Reyes J. and Coello Coello C. A., "A comparative study of differential evolution variants for global optimization", in GECCO 2006-Genetic and Evolutionary Computation Conference, pp. 485-492, Seattle, WA, 2006.

[25] Iorio A. W. and Li X. Solving rotated multi-objective optimization problems using differential evolution. Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science) 2004;3339 (861-872.

[26] Lee B. K., Chen H. W. and Chen B. S. Power control of cellular radio systems via robust smith prediction filter. IEEE Transactions on Wireless Communications 2004;3 (5) 1822-1831.

[27] Karamalis P. D., Skentos N. D. and Kanatas A. G. Selecting array configurations for mimo systems: An evolutionary computation approach. IEEE Transactions on Wireless Communications 2004;3 (6) 1994-1998.

[28] Lu H. Y. and Fang W. H. Joint receive antenna selection and symbol detection for mimo systems: A heterogeneous genetic approach. IEEE Communications Letters 2009;13 (2) 97-99.

[29] Lain J. K. Joint transmit/receive antenna selection for mimo systems: A real-valued genetic approach. IEEE Communications Letters 2011;15 (1) 58-60.

[30] Cheng W., Shi H., Yin X. and Li D. An elitism strategy based genetic algorithm for streaming pattern discovery in wireless sensor networks. IEEE Communications Letters 2011;15 (4) 419-421.

[31] Zheng G., Chatzinotas S. and Ottersten B. Generic optimization of linear precoding in multibeam satellite systems. IEEE Transactions on Wireless Communications 2012;11 (6) 2308-2320.

[32] Lu T. and Zhu J. Genetic algorithm for energy-efficient qos multicast routing. IEEE Communications Letters 2013;17 (1) 31-34.

[33] Wimalajeewa T. and Jayaweera S. K. Optimal power scheduling for correlated data fusion in wireless sensor networks via constrained pso. IEEE Transactions on Wireless Communications 2008;7 (9) 3608-3618.

[34] Yao W., Chen S., Tan S. and Hanzo L. Minimum bit error rate multiuser transmission designs using particle swarm optimisation. IEEE Transactions on Wireless Communications 2009;8 (10) 5012-5017.

[35] Kulkarni R. V. and Venayagamoorthy G. K. Particle swarm optimization in wireless-sensor networks: A brief survey. IEEE Trans Syst Man Cybern Pt C Appl Rev 2011;41 (2) 262-267.

[36] Nguyen H. A., Guo H. and Low K. S. Real-time estimation of sensor node's position using particle swarm optimization with log-barrier constraint. IEEE Trans. Instrum. Meas. 2011;60 (11) 3619-3628.

[37] Abdelhafiz A. H., Hammi O., Zerguine A., Al-Awami A. T. and Ghannouchi F. M. A pso based memory polynomial predistorter with embedded dimension estimation. IEEE Transactions on Broadcasting 2013;59 (4) 665-673.

[38] Li M., Xiong W. and Liang Q. Wireless sensor networks node localization algorithm based on improved abc algorithm. Chin. J. Sens. Actuators 2013;26 (2) 241-245.

[39] Li M., Xiong W. and Liang Q., "An improved abc-based node localization algorithm for wireless sensor network", in 2012 8th International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2012, Shanghai, 2012.

[40] He P. Y. and Jiang M. Y., "Dynamic deployment of wireless sensor networks by an improved artificial bee colony algorithm", in 2013 International Conference on Sensors, Mechatronics and Automation, ICSMA 2013, pp. 862-866, Shenzhen, 2014.

[41] Gu Y., Xu X., Du J., Hou R. and Qian H. Anycast routing protocol for wireless sensor networks based on artificial bee colony. Chin. J. Sens. Actuators 2013;26 (4) 564-569.

[42] Berrocal-Plaza V., Vega-Rodríguez M. A., Gómez-Pulido J. A. and Sánchez-Pérez J. M., "Artificial bee colony algorithm applied to wimax network planning problem", in 2011 11th International Conference on Intelligent Systems Design and Applications, ISDA'11, pp. 504-509, Cordoba, 2011.

[43] Liu J., Jia Z., Qin X., Chang C., Xu G. and Xia X., "The applications in channel assignment based on cooperative hybrid artificial bee colony algorithm", in 2011 International Conference on Electrical Engineering and Automation, EEA 2011, pp. 401-406, Beijing, 2012.

[44] Deepalakshmi P. and Radhakrishnan S. An ant colony-based, receiver-initiated multicast mesh protocol for collaborative applications of mobile ad hoc networks. Eur Trans Telecommun 2014;25 (3) 354-369.

[45] Balaji V. and Duraisamy V. Ant optimized link quality for ad hoc on demand distance vector. Wireless Pers Commun 2014;

[46] Shirkande S. D. and Vatti R. A., "Aco based routing algorithms for ad-hoc network (wsn,manets):A survey", in 3rd International Conference on Communication Systems and Network Technologies, CSNT 2013, pp. 230-235, Gwalior, 2013.

[47] Liu X. and He D. Ant colony optimization with greedy migration mechanism for node deployment in wireless sensor networks. J Network Comput Appl 2014;39 (1) 310-318.

[48] Liu X. A transmission scheme for wireless sensor networks using ant colony optimization with unconventional characteristics. IEEE Communications Letters 2014;18 (7) 1214-1217.

[49] Liu W. and Wang L. Ant colony optimization routing algorithm based on wsn. ICIC Express Lett Part B Appl. 2013;4 (3) 541-547.

[50] Li Z. and Shi Q., "An qos algorithm based on aco for wireless sensor network", in 15th IEEE International Conference on High Performance Computing and Communications, HPCC 2013 and 11th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, EUC 2013, pp. 1671-1674, IEEE Computer Society, Zhangjiajie, Hunan, 2014.

[51] Liu X. Sensor deployment of wireless sensor networks based on ant colony optimization with three classes of ant transitions. IEEE Communications Letters 2012;16 (10) 1604-1607.

[52] He Q., Feng Z. and Zhang P. Reconfiguration decision making based on ant colony optimization in cognitive radio network. Wireless Pers Commun 2013;71 (2) 1247-1269.

[53] Zhao Y., Xu X., Hao Z., Tao X. and Zhang P., "Resource allocation in multiuser ofdm system based on ant colony optimization", in IEEE Wireless Communications and Networking Conference 2010, WCNC 2010, Sydney, NSW, 2010.

[54] Marinello J. C. and Abrao T., "Lattice reduction aided detector for dense mimo via ant colony optimization", in 2013 IEEE Wireless Communications and Networking Conference, WCNC 2013, pp. 2839-2844, Shanghai, 2013.

[55] Lain J. K. and Chen J. Y. Near-mld mimo detection based on a modified ant colony optimization. IEEE Communications Letters 2010;14 (8) 722-724.

[56] Das S., Mukherjee R., Kundu R. and Vasilakos T., "Multi-user detection in multi-carrier cdma wireless broadband system using a binary adaptive differential evolution

algorithm", in 2013 15th Genetic and Evolutionary Computation Conference, GECCO 2013, pp. 1245-1252, Amsterdam, 2013.

[57] Kundu S., Das S., Vasilakos A. V. and Biswas S. A modified differential evolution-based combined routing and sleep scheduling scheme for lifetime maximization of wireless sensor networks. Soft Computing 2014;

[58] Yin X., Ling Z., Guan L. and Liang F. Minimum distance clustering algorithm based on an improved differential evolution. Int. J. Sens. Netw. 2014;15 (1) 1-10.

[59] Wu C. Y., Yen H. S., Chiu C. C., Chiang J. S. and Chang C. W., "Urban area propagation path loss reduction by dynamic differential evolution algorithm", in 1st International Conference on Intelligent Green Building and Smart Grid, IGBSG 2014, IEEE Computer Society, Taipei, 2014.

[60] Wang Z. and Zhang W. Spectrum sharing with limited channel feedback. IEEE Transactions on Wireless Communications 2013;12 (5) 2524-2532.

[61] Li K., Wang X. and Ping L. Analysis and optimization of interleave-division multiple-access communication systems. IEEE Transactions on Wireless Communications 2007;6 (5) 1973-1982.

[62] Zhao Z., Peng Z., Zheng S. and Shang J. Cognitive radio spectrum allocation using evolutionary algorithms. IEEE Transactions on Wireless Communications 2009;8 (9) 4421-4425.

[63] Merchant A. and Sengupta B. Assignment of cells to switches in pcs networks. IEEE/ACM Transactions on Networking 1995;3 (5) 521-526.

[64] Pierre S. and Houéto F. A tabu search approach for assigning cells to switches in cellular mobile networks. Computer Communications 2002;25 (5) 464-477.

[65] Shyu S. J., Lin B. M. T. and Hsiao T. S. Ant colony optimization for the cell assignment problem in pcs networks. Computers and Operations Research 2006;33 (6) 1713-1740.

[66] Salcedo-Sanz S. and Yao X. Assignment of cells to switches in a cellular mobile network using a hybrid hopfield network-genetic algorithm approach. Appl. Soft Comput. J. 2008;8 (1) 216-224.

[67] Kennedy J. and Eberhart R. C., "Discrete binary version of the particle swarm algorithm", in Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, pp. 4104-4108, 1997.

[68] Müller S. H. and Huber J. B. Ofdm with reduced peak-to-average power ratio by optimum combination of partial transmit sequences. Electronics Letters 1997;33 (5) 368-369.

[69] Cimini Jr L. J. and Sollenberger N. R. Peak-to-average power ratio reduction of an ofdm signal using partial transmit sequences. IEEE Communications Letters 2000;4 (3) 86-88.

[70] Tellambura C. Improved phase factor computation for the par reduction of an ofdm signal using pts. IEEE Communications Letters 2001;5 (4) 135-137.

[71] Han S. H. and Lee J. H. Papr reduction of ofdm signals using a reduced complexity pts technique. IEEE Signal Processing Letters 2004;11 (11) 887-890.

[72] Wen J. H., Lee S. H., Huang Y. F. and Hung H. L. A suboptimal pts algorithm based on paticle swarm optimization for papr reduction in ofdm systems. EURASIP J. Wireless Commun. Networking 2008;2008 (14)

[73] Wang X. H., Wang B. Z. and Chen K. J. Compact broadband dual-band bandpass filters using slotted ground structures. Progress in Electromagnetics Research 2008;82 (151-166.

[74] Chen C. Y. and Hsu C. Y. A simple and effective method for microstrip dual-band filters design. IEEE Microwave and Wireless Components Letters 2006;16 (5) 246-248.

[75] Koziel S. and Bandler J. W. Space mapping with multiple coarse models for optimization of microwave components. IEEE Microwave and Wireless Components Letters 2008;18 (1) 1-3.

IntechOpen