Energy Conversion and Management 50 (2009) 2020–2028

Contents lists available at ScienceDirect

Energy Conversion and Management

journal homepage: www.elsevier.com/locate/enconman

Comparison of metaheuristic techniques to determine optimal placement of biomass power plants

P. Reche-López [a], N. Ruiz-Reyes [a], S. García Galán [a], F. Jurado [b,*]

[a] Telecommunication Engineering Department, University of Jaén Polytechnic School, C/ Alfonso X el Sabio 28, 23700 Linares, Jaén, Spain
[b] Electrical Engineering Department, University of Jaén Polytechnic School, C/ Alfonso X el Sabio 28, 23700 Linares, Jaén, Spain

ARTICLE INFO

Article history:
Received 4 September 2008
Accepted 6 April 2009
Available online 2 May 2009

Keywords:
Biomass
Distributed generation
Metaheuristics
Simulated Annealing
Tabu search
Genetic Algorithms
Particle Swarm Optimization
Profitability index

ABSTRACT

This paper deals with the application and comparison of several metaheuristic techniques to optimize the placement and supply area of biomass-fueled power plants. Both, trajectory and population-based methods are applied for our goal. In particular, two well-known trajectory method, such as Simulated Annealing (SA) and Tabu Search (TS), and two commonly used population-based methods, such as Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) are hereby considered. In addition, a new binary PSO algorithm has been proposed, which incorporates an inertia weight factor, like the classical continuous approach. The fitness function for the metaheuristics is the profitability index, defined as the ratio between the net present value and the initial investment. In this work, forest residues are considered as biomass source, and the problem constraints are: the generation system must be located inside the supply area, and its maximum electric power is 5 MW. The comparative results obtained by all considered metaheuristics are discussed. Random walk has also been assessed for the problem we deal with.

1. Introduction

Biomass is the most common form of renewable energy, widely used in the third world, but until recently less so in the Western world. Nowadays, electric energy from biomass-fueled plants represents nine Exa Joules (EJ)/year. The best agricultural crop yields of biomass provide 10–15 dry ton/ha per year so that some 11,000 ha can give rise biomass for a 30 MW power station, enough to supply electricity to 30,000 houses. Global biomass power capacity added in 2005 amounted to 2–3 GW, bringing total capacity to about 44 GW [1]. Biomass currently caters about 10% of the world's primary energy supplies. Latterly, much attention has been focused on identifying suitable biomass species, which can provide high-energy outputs, to replace conventional fossil fuel energy sources. As with any energy resource, there are limitations on the use and applicability of biomass, and it must compete not only with fossil fuels but with other renewable energy sources, such as wind, solar and wave power.

Therefore, more research is required to prove that power generation from biomass is both technically and economically viable. In such sense, some interesting results can be found in [2–5]. An evaluation of the facility of a large-scale biomass scheme for production of electricity in Spain is presented in [3]. The biomass power plant is based on woody biomass, and the location selection process is carried out by a Geographical Information System (GIS). A GIS-based method is developed in [4] to assess the biomass potential for power production. The method offers the tools to distinguish the geographic distribution of the biomass potential. The main factors that influence the location and number of energy conversion facilities are plant capacity and distribution of the available biomass. In [5], a method for the optimal location and sizing of biomass fueled gas turbine power plants is presented. Both, profitability in using biomass and power loss are considered in the objective fitness function to obtain the solution. The method consists of two steps. The first step aims to achieve the plant size that maximizes the profitability of the project. The second step tries to determine the optimal location of the gas turbines in the electric system to minimize the power loss of the system. However, in the above cited papers, the specifications (size and selected sites) for the power plant are predefined parameters. In this paper, the problem to be solved consists on determining the optimum placement and size of the power plant. So, problem complexity increases significantly and justifies the use of modern heuristic techniques.

Biomass use for power generation is firmly expanding in Europe, where bioelectricity is mostly produced from wood residues [6]. The biomass power industry is also active in the United States, where some 85% of total wood process wastes (excluding forest residues) are used for power generation. Therefore, we will focus our attention on woods residues (in particular, forest residues) as

* Corresponding author. Tel.: +34 953 648518; fax: +34 953 648508.
E-mail address: fjurado@ujaen.es (F. Jurado).

0196-8904/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.enconman.2009.04.008

biomass source. Amongst the different process options for getting electricity from biomass, gasification is the most likely cost-effective process. Many studies have exposed the advantages of gasification over combustion for power production [7]. Gasification opens the possibility of going from the traditional, small-scale, low-efficiency steam cycle to combined steam and gas turbine with higher efficiency.

The conversion of biomass by gasification into a fuel suitable for direct use in gas turbines or gas engines increases greatly the potential usefulness of biomass as a renewable resource. Gas turbines can offer solutions to today's energy situation as a supplement or support function to the conventional central generation [8]. In this paper, the biomass-fueled system consists of gas turbines.

A biomass-based power system presents the problem of determining the optimal placement and the supply area for the biomass plant in order to provide a given electric power. The installation of biomass power plants at non-optimal places can result in an increase of costs and system losses, making the power plant economically unfeasible. Therefore, the use of an optimization method capable of determining the best solution regarding the placement and supply area of the electric power plant can be very useful. In last decades, a new kind of approximate algorithms has emerged, which basically tries to combine basic heuristic methods in higher level frameworks aimed at efficiently and effectively exploring a search space. These algorithms are commonly called metaheuristics [9]. Although metaheuristics do not always guarantee the globally optimal solution, they provide suboptimal solutions in short CPU times. These algorithms can be interpreted as introducing a bias such that high quality solutions are produced quickly.

When a realistic problem formulation with the just mentioned considerations is to be solved, classical analytical, numerical programming or heuristic methods are usually either high time-consuming or do not provide good results. It is shown in [10] that traditional optimization techniques (like interior-point method and gradient-based method) perform better than metaheuristics for problems where the response surface is strongly convex. However, this assumption is not true in most of optimization problems, where multimodal functions are considered. Empirical comparisons were performed in [11], showing that metaheuristic methods provide a substantial advantage over classical methods when the fitness function is multimodal. Therefore, metaheuristics outperform classical analytical, numerical programming and heuristic methods in most real-world optimization problems, as the one addressed in this work.

Different metaheuristics with promising results have been applied to Distributed Generation (DG) and renewable energies. Haghifam et al. propose in [12] a strategy for the placement of DG units in a changeable environment. Uncertainties in the system are modeled using logic fuzzy. The true Pareto-optimal solutions are established with a multiobjective genetic algorithm and the final solution is determined using a max–min approach. In [13] a multiobjective formulation for the siting and sizing of DG resources into existing distribution networks is proposed, the implemented technique is based on a genetic algorithm and an ϵ-constrained method. Borges et al. propose in [14] a parallel genetic algorithm-based methodology for network reconfiguration in a dispersed generation framework. In [15], a new method based on genetic algorithms is employed to determine the optimal distributed generation location on a distribution network, considering the vulnerability of the system to voltage sags. The work in [16] describes heuristic and probabilistic procedures to estimate wind power availability. The wind velocity data are employed for technology and site selection. In [17], a hybrid optimization approach based on genetic algorithms is applied to design a photovoltaic-diesel system. The PV-diesel system, optimized by the hybrid approach, is compared with a stand-alone PV system that has been calculated using a classical design method.

In this work, four metaheuristic approaches are applied to the problem of determining the optimal placement and supply area of biomass-fueled power plants. In particular, two well-known trajectory methods, such as Simulated Annealing (SA) and Tabu Search (TS), and two commonly used population-based methods, such as Genetic Algorithms (GA) and Particle Swarm Optimization (PSO), are hereby considered. These algorithms are descent biased, the profitability index being the objective fitness function.

The main original contributions of this work are referred to: (1) a new Binary Particle Swarm Optimization (BPSO) algorithm, incorporating inertia weight factor, as in the classical continuous approach of Kennedy and Eberhart [18,19]; (2) application and comparison of metaheuristic techniques, including the proposed BPSO algorithm, to determine the optimal placement and the supply area of biomass power plants. The values of the parameters for the different tested algorithms have been optimized. The proposed BPSO algorithm has been assessed by comparison with other discrete PSO algorithms, including the original discrete one from Kennedy and Eberhart [20]. A classical approach of Genetic Algorithms (GA) has also been applied to validate the results obtained by the proposed BPSO algorithm.

This paper is structured as follows. Section 1 introduces concepts about biomass-fueled power systems, briefly reviews previously published literature on using metaheuristic methods applied to DG and renewable resources, and outlines the main contributions of the paper. Section 2 overviews the principles of metaheuristics and the algorithms compared in this work, describing in detail the proposed BPSO algorithm. Section 3 is devoted to the problem description. Experimental results are shown in Section 4, which allow to assess the performance of all tested algorithms. Finally, Section 5 outlines some meaningful conclusions.

2. Metaheuristics

2.1. Basics and approaches

Many optimization problems of practical as well as theoretical importance consist of the search for a "best" configuration of a set of variables to achieve some goals. They can be classified into two categories: those where solutions are encoded with real-valued variables, and those where solutions are encoded with discrete variables. Among the latter ones, we find the so-called Combinatorial Optimization (CO) problems, where looking for an object from a finite set is intended [9]. Examples for CO problems are the Traveling Salesman problem (TSP), the Quadratic Assignment problem (QAP), Timetabling and Scheduling problems. The problem addressed in this work belongs to this category (CO problem).

Due to the practical importance of CO problems, many algorithms to tackle them have been developed. They can be classified as either complete or approximate algorithms. Complete algorithms are guaranteed to find an optimal solution at the expense of too high computation times for practical purposes. Thus, the use of approximate methods to solve CO problems has received increasingly attention in the time. In approximate methods we sacrifice the guarantee of finding optimal solutions for the sake of getting good solutions in a significantly reduced amount of time. In the last 20 years, a new kind of approximate algorithms has emerged, which guide a subordinate heuristic for exploring and exploiting the search space in order to find efficiently near-optimal solutions. This class of algorithms, commonly called metaheuristics, includes but is not restricted to Ant Colony Optimization (ACO), Evolutionary Computation (EC) including Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Iterated Local Search (ILS), Simulated Annealing (SA), and Tabu Search (TS).

Two very important concepts in metaheuristics are intensification and diversification. They are in some way contrary, but also complementary to each other. The balance between diversification and intensification is an important issue in real-world problems. There are several approaches to classify metaheuristics, according to their properties. We briefly summarize the most important ways of classifying metaheuristics:

- *Nature-inspired vs. non-nature-inspired*. Perhaps, the most intuitive way of classifying metaheuristics is based on the origin of the algorithm. Therefore, we can find nature-inspired algorithms, like GA and ACO, and non-nature-inspired ones, such as TS and ILS.
- *Population-based vs. trajectory methods*. Algorithms working on a single solution at any time are called *trajectory methods*, and encompass single point search-based metaheuristics, like TS and ILS. They all give rise to a trajectory in the search space during the search process. On the contrary, population-based metaheuristics, like GA and PSO, perform search processes which describe the evolution of a set of points in the search space.

2.2. Survey of metaheuristic techniques

In the following, we briefly describe the metaheuristic techniques considered in this work. Here, we have used two trajectory methods (SA and TS) and two population-based methods (GA and PSO).

2.2.1. Simulated annealing

The underlying principle of SA is in its analogy with the thermodynamics, especially with the way that liquids freeze and crystallize or metals cool and anneal. This algorithm was first presented by Kirkpatrick in [21]. The performance of SA is mainly based on the so-called temperature parameter T, which is decreased during the search process. Thus, at the beginning of the search, the probability of accepting uphill moves (to escape from local maxima) is high, and it gradually decreases, converging to a simple iterative improvement algorithm. This process is analogous to the annealing process of metals and glass, which assume a low energy configuration when cooled with an appropriate cooling schedule.

Regarding the search process, the algorithm is the result of two combined strategies: random walk and iterative improvement. In the first phase of the search, the bias toward improvements is low and it permits the exploration of the search space; this erratic component is slowly decreased, thus leading the search to converge to a (local) minimum. The probability of accepting uphill moves is controlled by two factors: the difference of the objective functions and the temperature.

The choice of an appropriate cooling schedule is crucial for the performance of the algorithm. The cooling schedule and the initial temperature should be adapted to the particular problem instance, since the cost of escaping from local minima depends on the structure of the search landscape. SA has been applied to several CO problems (QAP and scheduling).

2.2.2. Tabu search

TS was first introduced in Glover [22]. TS explicitly uses the history of the search, both to escape from local maxima and to implement an explorative strategy.

The classic TS algorithm uses a short term memory to escape from local maxima and to avoid cycles. The short term memory is implemented as a tabu list that keeps track of the most recently visited solutions and forbids moves toward them. The neighborhood of the current solution is thus restricted to the solutions that do not belong to the tabu list. This set of solutions is commonly referred to allowed set. At each iteration, the best solution from the allowed set is chosen as the new solution. Additionally, this solution is added to the tabu list and one of the solutions that already were in the tabu list is removed (usually in a FIFO order). The algorithm stops when a halt condition is met (in our case, the maximum number of iterations). It might also terminate if the allowed set was empty (all the solutions in the neighborhood of the current solution were forbidden by the tabu list).

The use of a tabu list prevents from returning to recently visited solutions. Therefore, it prevents from endless cycling, and forces the search to accept even uphill moves. The length of the tabu list (the tabu tenure) controls the memory of the search process. With small tabu tenures, the search will concentrate on small areas of the search space. On the opposite, a large tabu tenure forces the search process to explore larger regions, because it forbids revisiting a higher number of solutions.

2.2.3. Genetic algorithms

They are general purpose search algorithms that use principles inspired by natural genetics to evolve solutions to problems. A GA starts off with a population of randomly generated chromosomes, and advances toward better chromosomes by applying genetic operators. During successive iterations, called generations, chromosomes in the population are rated for their adaptation as solutions. On the basis of these evaluations, a new population of chromosomes is formed using a selection mechanism and specific genetic operators, such as crossover and mutation. An evaluation or fitness function must be devised for each problem to be solved. Given a particular chromosome (a possible solution), the fitness function returns a single numerical value, which is supposed to be proportional to the utility or adaptation of the solution represented by that chromosome.

Although there are many possible variants GA, the underlying mechanism operates on a population of chromosomes or individual, and consists of three operations:

- Evaluation of individual fitness. For each problem to be solved, a suitable fitness function is required.
- Formation of a gene pool through selection mechanisms. Here, the so-called *elitist strategy* has been used in order to include into the gene pool the best found solutions.
- Recombination through crossover and mutation operators. In this work, single point crossover is performed, and an exponentially decreasing function is used for the mutation probability.

GA are especially well-fitted to difficult environments where the space is usually large, discontinuous, complex and poorly understood. The basic principles of GA were first laid down by Holland [23], and are well described in many books, such as [24, 25]. It is generally accepted that application of GA must take into account the following components:

- A genetic representation of solutions to the problem.
- A way to create an initial population of solutions.
- An evaluation function, which gives the fitness of each chromosome.
- Genetic operators, which modify the genetic composition of offspring during reproduction.
- Values for the parameters of the GA (population size, probabilities of applying genetic operators, etc.).

2.2.4. Particle Swarm Optimization

The classical PSO algorithm [18,19] is initialized with a swarm of particles randomly placed on the search space. At the tth iteration, position of the ith particle is updated by adding to its previous position the new velocity vector, according to the following equation:

$$x_{i,j}^t = x_{i,j}^{t-1} + v_{i,j}^t, \quad i = 1, \ldots, P \quad j = 1, \ldots, N \tag{1}$$

where $\mathbf{x}_i^t = [x_{i,1}^t, \ldots, x_{i,N}^t]$ denotes the position vector of the ith particle at the tth iteration, and $\mathbf{v}_i^t = [v_{i,1}^t, \ldots, v_{i,N}^t]$ represents the velocity vector of the ith particle at the tth iteration, N being the number of variables of the function to be optimized and P the number of particles in the swarm.

The velocity vector $\mathbf{v}_i^t$ is updated according to the following equation:

$$v_{i,j}^t = \omega \cdot v_{i,j}^{t-1} + c_1 \cdot rand_{1_i} \cdot (pbest_{i,j}^{t-1} - x_{i,j}^{t-1}) + c_2 \cdot rand_{2_i} \cdot (gbest^{t-1} - x_{i,j}^{t-1}) \tag{2}$$

where $\mathbf{pbest}_i^{t-1} = [pbest_{i,1}^{t-1}, \ldots, pbest_{i,N}^{t-1}]$ is the best solution achieved for the ith particle at the $(t-1)$th iteration, and $\mathbf{gbest}^{t-1} = [gbest_1^{t-1}, \ldots, gbest_N^{t-1}]$ is the best position found for all particles in the swarm at the $(t-1)$th iteration. c_1 and c_2 are positive real numbers, called learning factors or acceleration constants, that are used to weight the particle individual knowledge and the swarm social knowledge, respectively. $rand_{1_i}$ and $rand_{2_i}$ are real random numbers uniformly distributed between 0 and 1, that cause stochastic changes in the ith particle trajectory. Finally, ω is the inertia factor, which represents the weight applied to the previous velocity of the ith particle. A suitable selection of inertia factor in Eq. (2) provides a balance between global and local explorations.

The classical version of the PSO algorithm [18,19] operates in a continuous search space. In order to solve optimization problems in discrete search spaces, several binary discrete PSO algorithms have been proposed. In a binary discrete space the position of a particle is represented by a N-length bit string and the movement of the particle consists of flipping some of these bits. Kennedy and Eberhart proposed in [20] the first binary version of PSO. Since then, other binary approaches for PSO have appeared in the literature [26–29]. Among them, the approach in [27] closely resemble the classical continuous PSO approach.

2.3. The proposed binary PSO algorithm

In this work, we have developed and applied a improved version of the binary PSO algorithm proposed in [27], which incorporates a inertia weight factor, like the classical continuous approach [18,19]. Now, particle position ($\mathbf{x}_i$) and particle velocity ($\mathbf{v}_i$) are N-length binary vectors. The algorithm uses the Hamming distance, and the logical AND '·', OR ('+') and XOR '$\oplus$' operators.

Particle position is updated by using the XOR operator instead of the sum-operator, as in [27]:

$$x_{i,j}^t = x_{i,j}^{t-1} \oplus v_{i,j}^t, \quad i = 1, \ldots, P \quad j = 1, \ldots, N \tag{3}$$

In our approach, the velocity vector can be interpreted as a change vector. Thus, if $v_{i,j}^t = $ '1', then $x_{i,j}^t = \bar{x}_{i,j}^{t-1}$, $\bar{x}_{i,j}^{t-1}$ being the logical negation of $x_{i,j}^{t-1}$. However, if $v_{i,j} = $ '0', then $x_{i,j}^t = x_{i,j}^{t-1}$ (no change happens). The velocity vector (change vector) is updated by applying the following equation:

$$v_{i,j}^t = \bar{\omega}_{i,j} + \omega_{i,j} \cdot (c_{1_{i,j}} \cdot (pbest_{i,j}^{t-1} \oplus x_{i,j}^{t-1}) + c_{2_{i,j}} \cdot (gbest_j^{t-1} \oplus x_{i,j}^{t-1})) \tag{4}$$

where:

- $\mathbf{c}_{1i} = [c_{1_{i,1}}, \ldots, c_{1_{i,N}}]$, $\mathbf{c}_{2i} = [c_{2_{i,1}}, \ldots, c_{2_{i,N}}]$ are random N-length binary strings, whose components have the same probability.
- $\mathbf{pbest}_i^{t-1} = [pbest_{i,1}^{t-1}, \ldots, pbest_{i,N}^{t-1}]$, $\mathbf{gbest}^{t-1} = [gbest_1^{t-1}, \ldots, gbest_N^{t-1}]$ are also N-length binary strings.
- $\boldsymbol{\omega}_i = [\omega_{i,1}, \ldots, \omega_{i,N}]$ is the inertial vector of the ith particle. It is a random N-length binary vector, whose components are '0' with probability $\mathscr{P}_\omega$.
- $\bar{\boldsymbol{\omega}}_i = [\bar{\omega}_{i,1}, \ldots, \bar{\omega}_{i,N}]$ is the one's complement of inertial vector $\boldsymbol{\omega}_i$.

In our improved binary PSO approach, a very important parameter is probability $\mathscr{P}_\omega$, here called *inertial probability*. As just stated, bits in $\boldsymbol{\omega}_i$ are '0' with probability $\mathscr{P}_\omega$. Inertial probability decreases with the number of iterations, in such a way that at the initial iterations (high $\mathscr{P}_\omega$ values) the algorithm *explores* the search space and at the last iterations (low $\mathscr{P}_\omega$ values) the algorithm *exploit* the search space.

It must be noted that if $\omega_{i,j} = $ '0', then $v_{i,j}^t = $ '1', and so position of the ith particle is changed. However, if $\omega_{i,j} = $ '0', the movement of the ith particle at the tth iteration is conducted by $\mathbf{pbest}_i^{t-1}$ and $\mathbf{gbest}^{t-1}$ solutions, with a partially stochastic behavior due to the random learning vectors $\mathbf{c}_{1i}$ and $\mathbf{c}_{2i}$. The idea is to allow particle swarm to perform a random exploration over the space search at the initial iterations. Later, when the swarm has acquired enough knowledge about the problem, the movement of each particle is mainly conducted by $\mathbf{pbest}_i$ and $\mathbf{gbest}$ solutions. In this work, an exponentially decreasing function is used for probability $\mathscr{P}_\omega$.

3. Problem formulation, solution representation and fitness function

3.1. Problem description

The problem to be solved consists on determining the optimum placement and supply area of a biomass-fueled power plant based on forest residues. For such goal, four metaheuristic techniques are applied and compared. Here, we have employed two trajectory methods (SA and TS) and two population-based methods (GA and BPSO). The size of the generation system depends on: (1) biomass quantity that can be collected, (2) selection of parcels where to collect the biomass. Placement of power plant (parcel p) mainly depends on the characteristics of the parcels. In this work, K parcels of constant area have been considered, all of them characterized by a predominant biomass type (forest residues in this work). These parcels also present other relevant characteristics, such as accessibility [30].

The values of the variables involved in the problem are obtained from databases or Geographic Information Systems (GIS). These are the following:

- S_i: Area of parcel i (km^2).
- U_i: Usability coefficient of parcel i. It is applied to only take the usable surface into account.
- D_i: Net density of dry biomass yield from parcel i (ton/(km^2 year)).
- LHV_i: Lower heat value of biomass in parcel i (MW h/ton).
- L_p: Length of the electric line that connects the power plant to the grid (km).
- $dist(p, i)$: Distance between parcel i and the power plant, which is located in parcel p(km).
- C_{cu_i}: Biomass collection unit cost in parcel i (€/ton).

Therefore, given the total mean efficiency of the electric generation system, η, the electricity produced, E_g (MW h/yr), is equal to:

$$E_g = \eta \cdot \sum_{i=1}^{K} (S_i \cdot U_i \cdot D_i \cdot LHV_i) \tag{5}$$

Assuming a plant running time of T(h/yr), the electric power, P_e (MW), is:

$$P_e = \frac{E_g}{T} \tag{6}$$

3.2. Coding of the solution

Before using a given metaheuristic to determine the optimum placement and supply area of the biomass power plant, the representation of a feasible solution must be defined. A solution consists of three parts: (1) X component of location plant; (2) Y component of location plant; (3) Size of supply area for the power plant. These components are binary Gray coded in order to exploit some useful properties of Gray code related with the Hamming distance.

We have considered a rectangular search space with $x \in [1, L_X]$ and $y \in [1, L_Y]$, L_X and L_Y being sizes in X-dimension and Y-dimension, respectively. Supply area is a square shaped region which has the plant at the centroid. In order to obtain not only the placement of the power plant but also the size of the supply area, a prefixed number of supply region sizes has been assumed (i.e. size number 0 corresponds to a 1×1 region, size number 1 corresponds to a 3×3 region, and maximum size number S corresponds to a $(2 \cdot S + 1) \times (2 \cdot S + 1)$ region). Thus, the total number of bits to code the solution is:

$$N = \log_2 L_X + \log_2 L_Y + \log_2 S \tag{7}$$

3.3. Objective fitness function: profitability index

The objective fitness function takes costs and benefits into consideration. Specifically, initial investment and collection, transportation, maintenance and operation costs are considered, together with benefits from the sale of electrical energy. Therefore, the profitability index is chosen as the objective function.

In this section some interesting parameters to evaluate the profitability index of the project are reviewed. The initial investment, the present value of cash inflows (benefits) and cash outflows (costs) and the net present value are studied and adapted to the particularities of this work.

3.3.1. Initial investment

The initial investment (*INV*) for the design, construction of the generation plant and required equipment is expressed as:

$$INV = INV_f + I_s \cdot P_e + C_L \cdot L_p \tag{8}$$

where INV_f is the fixed investment (€), I_s is the specific investment (€/MW) and C_L the electric line cost (€/km).

3.3.2. Cash inflows

The present value of cash inflows (PV_{IN}) is obtained from the sold electric energy during the useful lifetime, V_u. It can be written as:

$$PV_{IN} = p_g \cdot E_g \cdot \frac{K_g \cdot (1 - K_g^{V_u})}{1 - K_g} \tag{9}$$

where p_g is the selling price of the electric energy injected to the network (€/MW h), E_g the sold and produced electric energy (MW h/yr) and $K_g = \frac{1+r_g}{1+d}$, r_g being the annual increase rate of the sold energy price and d the nominal discount rate.

3.3.3. Cash outflows

The present value of cash outflows (PV_{OUT}) is the sum of the following costs during the useful lifetime of the plant: annual collection cost, C_c, annual transport cost, C_t, and annual maintenance and operation costs, C_{mo}.

The annual cost of biomass collection is $C_c = \sum_{i=1}^{K}(C_{cu_i} \cdot U_i \cdot S_i \cdot D_i)$.

The annual cost of biomass transport is $C_t = \sum_{i=1}^{K}(C_{tu_i} \cdot U_i \cdot S_i \cdot D_i \cdot dist(p,i))$, where C_{tu_i} is the biomass transport unit cost in parcel i (€/(ton km)).

The annual maintenance and operation costs are $C_{mo} = C_{mof} + m \cdot E_g$, where C_{mof} is the fixed annual cost of maintenance and operation, which mainly consists of the minimum labor cost of the plant (€/year), and m is the average maintenance cost (€/MW h).

Finally, the present value of cash outflows is:

$$PV_{OUT} = C_c \cdot \frac{K_c \cdot (1 - K_c^{V_u})}{1 - K_c} + C_t \cdot \frac{K_t \cdot (1 - K_t^{V_u})}{1 - K_t} + C_{mo} \cdot \frac{K_{mo} \cdot (1 - K_{mo}^{V_u})}{1 - K_{mo}} \tag{10}$$

where $K_c = \frac{1+r_c}{1+d}$, $K_t = \frac{1+r_t}{1+d}$ and $K_{mo} = \frac{1+r_{mo}}{1+d}$, r_c being the annual increase rate of C_c, r_t the annual increase rate of C_t and r_{mo} the annual increase rate of C_{mo}.

3.3.4. Net present value

The present value (*PV*) of an investment is the present value of cash inflows (PV_{IN}) minus the present value of cash outflows (PV_{OUT}) during the useful lifetime of the plant. Therefore, it can be written as:

$$PV = PV_{IN} - PV_{OUT} \tag{11}$$

The net present value (*NPV*) is defined as the present value (*PV*) minus the initial investment (*INV*):

$$NPV = PV - INV \tag{12}$$

3.3.5. Profitability index

The fitness function that has been used in this work is the profitability index (*PI*), which is defined as follows:

$$PI = \frac{NPV}{INV} = \frac{PV_{IN} - PV_{OUT} - INV}{INV} = \frac{PV_{IN} - PV_{OUT}}{INV} - 1 \tag{13}$$

An investment is profitable when $PI > 0$.

4. Experimental results

The region considered to apply and compare the four metaheuristics (SA, TS, GA and PSO) consists of $128 \times 128 = 16,384$ parcels of constant surface, $S_i = 2$ km^2. The size of the supply area for the power plant is coded by 6 bits, i.e. $2^6 = 64$ different sizes are possible. The region is covered by natural forest vegetation. Forest residues constitute the biomass type. There are parcels where neither extraction of forest vegetation nor placement of the generation plant are possible.

Region under study, including position of the electrical lines, is shown in Fig. 1. A radial feeder is considered to assess the performance of the proposed optimization approach. The radial feeder is connected through a substation to a sub-transmission system. As mentioned in Section 3, the initial investment for construction of the generation plant and required equipment depends on the length of the electric line that connects the power plant to the grid. The number of evaluations of the fitness function needed to find the optimal solution when using exhaustive search is 2^N, N being the number of bits required to represent the solution. For the proposed problem, parameter N is equal to $\log_2(128) + \log_2(128) + 6 = 20$, which involves $2^{20} = 1,048,576$ evaluations. Although this number is not large enough to make exhaustive search impractical, it can become a major deal when changes in the experimental setup are required to assess several technologies, different parameter configurations, other electrical or economic models, etc. In all these cases, the computational time is a critical issue, which justifies the use of modern optimization methods, such as metaheuristics.

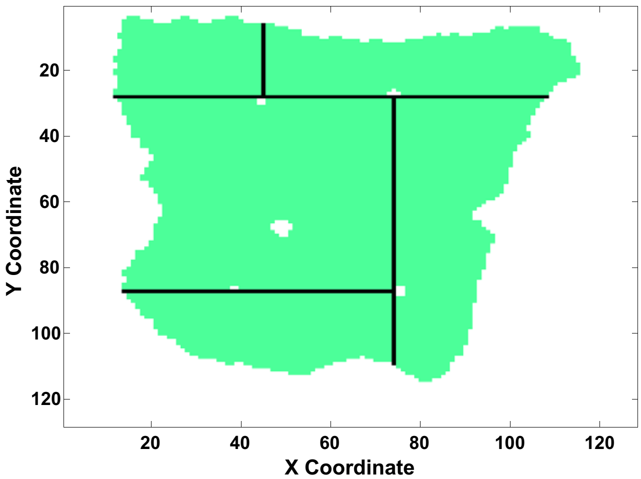

Fig. 1. Region under study, showing position of the electrical lines.

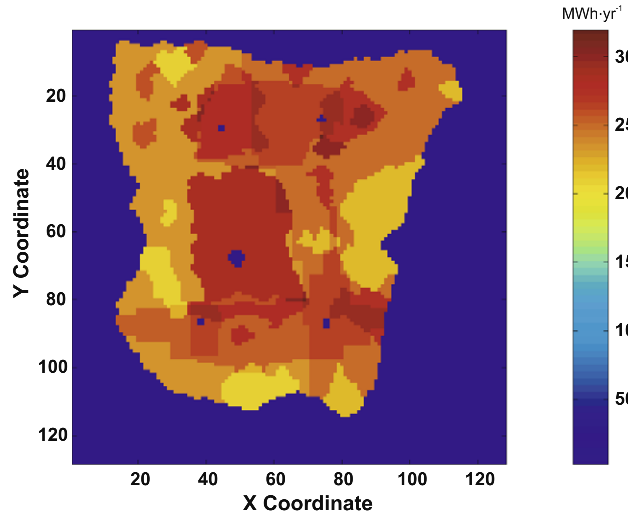

Fig. 3. Available biomass potential (MW h/yr).

Fig. 2 shows the theoretical biomass potential, which is defined from the net density of dry biomass that can be obtained at any parcel, D_i (ton/(km^2 year)), and provides a measure of the primary biomass resource.

Fig. 3 shows the available biomass potential. It has been created taking the following parameters into account: D_i (ton/(Km^2 year)), U_i, S_i (Km^2) and LHV_i (MW h/ton). By multiplying the four variables for all parcels that comprise the entire region, it results in the available biomass potential, expressed in (MW h/yr).

The available information for each parcel comprises $S_i, U_i, D_i, LHV_i, L_p, dist(p,i)$ and C_{cu_i}. Taking into account that gas turbine is the technology considered in this work for producing electric energy in the biomass-fueled power plant, Table 1 shows the remaining parameter values to compute the profitability index.

Once all parameter values required to compute the profitability index have been defined, the four considered metaheuristics (SA, TS, GA and PSO) are applied to the problem of determining the optimum placement and supply area of a biomass-fueled power plant. As just stated, the power plant is fueled by forest residues, which are converted into electric energy by using gas turbine-based technology. In the following, experimental results accomplished by the four considered metaheuristic techniques are reported. Analysis comparative between them is also included. First of all, we are going to assess the performance of the trajectory methods (SA and TS). Then, population-based methods (GS and PSO) are evaluated for the formulated problem.

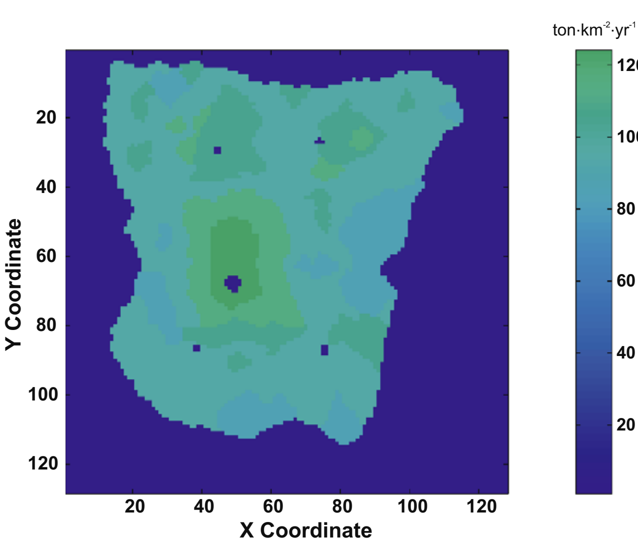

Fig. 2. Theoretical biomass potential (ton/(Km^2 yr)).

4.1. Trajectory methods

Simulation data for the SA algorithm are the following: (1) Initial temperature T^0 ranges from 1 to 10; (2) The total number of iterations for each experiment is $N_{\text{iter}} = 1000$; (3) The number of experiments (realizations) to obtain mean and standard deviation values is 30. SA performs an evaluation per iteration, which results in 1000 evaluations for each experiment. Table 2 depicts the influence of parameter T^0 in the profitability index when optimization is performed by SA.

From Table 2, it results that parameter T^0 does not influence in the profitability index. Experimental results also reveal that SA quickly find near-optimum solutions (i.e. an upper bound is reached in few iterations).

Simulation data for the TS algorithm are now the following: (1) The total number of iterations for each experiment is $N_{\text{iter}} = 1000$; (2) The number of experiments (realizations) to obtain mean and standard deviation values is 30. TS performs a variable number of evaluations at each iteration, because neighbors included into the Tabu list change from one iteration to the next. Table 3 shows the profitability index (mean and standard deviation) when optimization is performed by TS.

The computational cost of TS is higher than that of SA. In our experiments with TS, we have found that the average number of evaluations for each experiment is close to 1200, which involves

Table 1
Standard values for parameters (GAS TURBINE).

Parameter	Value	Parameter	Value
η	0.3	T (h/yr)	7500
INV_f (€)	1.5×10^6	I_s (€/MW)	1.2×10^6
C_L (€/km)	3×10^4	V_u (year)	15
p_g (€/MW h)	100	C_{tu_i} (€/(ton km))	0.3
C_{mof} (€/yr)	2.4×10^5	m (€/MW h)	4.0
d	0.08	r_g	0.04
r_c	0.06	r_t	0.08
r_{mo}	0.04		

Table 2
SA: Profitability index vs. T_0 (mean and standard deviation values).

T^0	Average	σ
1	1.8632	0.0243
2	1.8644	0.0241
3	1.8545	0.0360
4	1.8686	0.0255
5	1.8522	0.0435
6	1.8606	0.0410
7	1.8599	0.0234
8	1.8609	0.0297
9	1.8645	0.0279
10	1.8555	0.0294
Average	1.8604	0.0048

getting better solutions on average (i.e. higher mean value of the profitability index) than SA.

4.2. Population-based methods

Simulation data for the proposed binary PSO algorithm are the following: 1) Several population sizes are considered (P = 10, 20, 30, 40, 50 and 60); 2) The inertia probability at the beginning of the algorithm can take different values ($P_\omega^0 = 0$, 0.05, 0.1, 0.2, 0.3, 0.4 and 0.5); 3) The total number of iterations for each experiment is now $N_{iter} = 70$, which involves $(N_{iter}+1)\cdot P = 71\cdot P$ evaluations; 4) The number of experiments to obtain mean and standard deviation values does not change (30 realizations).

The performance of our binary PSO algorithm has been assessed by computing the influence of probability P_ω^0 on the profitability index for all considered values of parameters P. Experimental results illustrating that influence are shown in Fig. 4, which are based on 30 replicate simulation runs.

From Fig. 4, two main conclusions can be extracted: (1) the quality of the solution provided by the proposed BPSO algorithm increases with parameter P_ω^0 until a maximum is reached at $P_\omega^0 = 0.4$. This behavior has been observed for all considered values

Table 3
Profitability index (mean and standard deviation) when TS is performed.

Average	σ
1.9105	0.0236

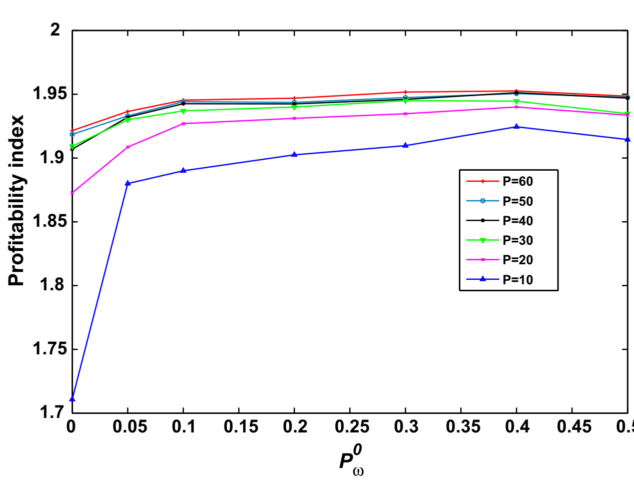

Fig. 4. Mean value of the profitability index vs. P_ω^0 for all considered values of parameter P when optimization by our BPSO algorithm is performed.

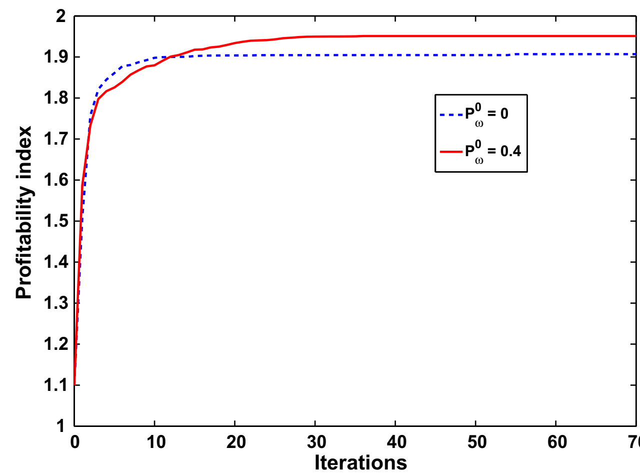

Fig. 5. Effect of parameter P_ω^0 on the convergence rate.

of parameter P (i.e. the profitability index increases with the inertia probability until $P_\omega^0 = 0.4$, regardless of the population size); (2) better solutions are obtained as the population size grows, until an upper bound is reached. It results that a population size above P = 40 implies higher computational cost (more evaluations are required) without increasing the quality of the solution accordingly.

Therefore, at the sight of results in Fig. 4, the proposed BPSO algorithm achieves optimal performance for the problem to be solved when $P_\omega^0 = 0.4$ and $P = 40$.

Fig. 5 shows the effect of parameter P_ω^0 on the convergence curve of the proposed BPSO algorithm. Comparative results between $P_\omega^0 = 0$ (high diversification) and $P_\omega^0 = 0.4$ (high intensification) are reported. The population size is $P = 40$ (the optimum one).

From Fig. 5, it results that $P_\omega^0 = 0.4$ yields better solutions than $P_\omega^0 = 0$ at the expense of more time spent.

Next, we are going to assess the performance of GA (a classical approach is considered) for the problem to be solved. Simulation data are the following: (1) Several population sizes are considered (P = 30, 40, 50, 60, 70, 80 and 90); (2) The mutation probability at the beginning of the algorithm (P_m^0) and the selection rate (SR) have been fixed to be 0.1 and 70%, respectively, which are typical values found in the literature; (3) The total number of iterations for each experiment is $N_{iter} = 70$, which involves $(N_{iter}\cdot\frac{SR}{100}+1)\cdot P$ evalua-

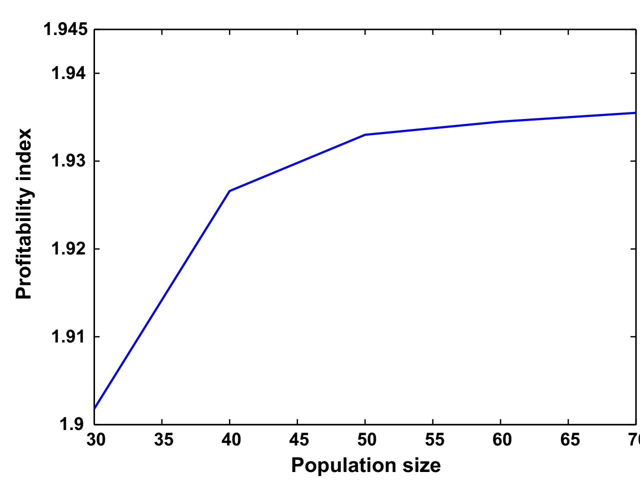

Fig. 6. Mean value of the profitability index vs. population size when optimization by GA is performed.

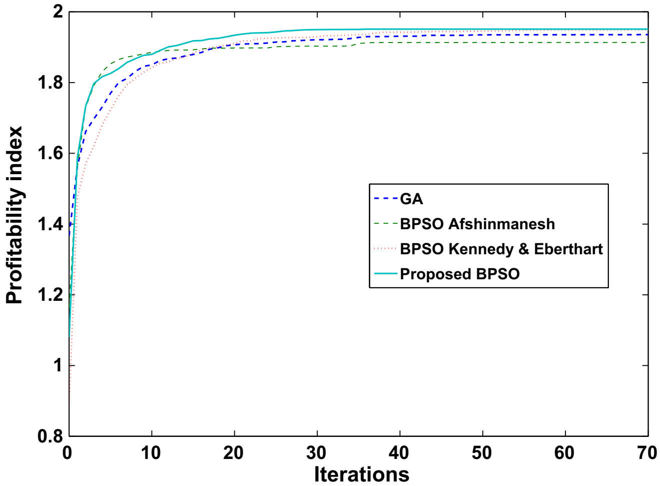

Fig. 7. Convergence characteristics for all considered metaheuristics.

tions; (4) Mean and standard deviation values are achieved with 30 realizations.

We are interested in knowing how the population size P influences the profitability index. Fig. 6 illustrates the behavior of GA with the population size.

From Fig. 6, it results that better solutions are obtained as the population size grows. However, there is a population size above which the quality of the solution does not improve accordingly with the computational cost. In such sense, $P = 60$ has been chosen as the optimum value for the population size.

Comparative results between BPSO and GA are next reported for the problem we deal with. Fig. 7 compares the convergence curve of the profitability index as a function of the number of iterations for all considered metaheuristics. Convergence curves in Fig. 7 show mean values computed over 30 realizations, and have been obtained by using the optimum parameter values of each algorithm. Here, three BPSO algorithms are assessed: the proposed one, the original BPSO algorithm by Kennedy and Eberhart [20] and the BPSO algorithm by Afshinmanesh et al. [27]. The optimum parameter values for all compared algorithms are:

- Proposed BPSO: $P_{\omega}^{0} = 0.4$ and $P = 40$
- GA: $P_m^0 = 0.1, SR = 70\%$ and $P = 60$
- BPSO by Kennedy and Eberhart: $P = 40$
- BSPO by Afshinmanesh et al.: $v_{max} = 18$ and $P = 40$

As shown in Fig. 7, the proposed BPSO algorithm converges to better solutions than GA and the other two BPSO algorithms. By analyzing the convergence curves in Fig. 7, the four assessed algorithms can be ranked as follows: (1) proposed BPSO; (2) BPSO by Kennedy and Eberhart; (3) GA; (4) BPSO by Afshinmanesh et al. Although the BPSO algorithm by Afshinmanesh et al. is the worst ranked, it provides the best performance for a reduced number of iterations (N_{iter} below 10). In that case, the worst ranked algorithm is the BPSO by Kennedy and Eberhart. The computational cost of BPSO is somewhat lower than that of GA. Notice that BPSO involves $P \cdot (N_{\text{iter}} + 1) = 40 \cdot (70 + 1) = 2840$ evaluations, while GA involves $P \cdot (N_{\text{iter}} \cdot \frac{SR}{100} + 1) = 60 \cdot (70 \cdot 0.7 + 1) = 3000$ evaluations.

4.3. Trajectory vs. population-based methods: comparative results

Here, comparison between all considered metaheuristics is reported. Simulations have been performed by using the optimum parameter values obtained for all algorithms. The results in Tables 4 and 5 are referred to trajectory and population-based methods, respectively. The tables show meaningful results concerning our problem (coordinates of the optimal location, supply area, profitability index, net present value, initial investment, distance to grid, generated power), derived from all metaheuristics. The results in both tables correspond to median values computed over 30 realizations.

Table 4
Results derived from trajectory methods.

	SA	TS
x-coordinate	85	29
y-coordinate	74	65
Supply area	870	874
P_e (MW)	4.6752	4.7158
Dist. grid	0	1.4142
PI	1.8763	1.9138
NPV (M€)	14.34	13.78
INV (M€)	7.11	7.20

Table 5
Results derived from population-based methods.

	BPSO by Afshimmanesh	BPSO by Kennedy	GA	Proposed BPSO
x-coordinate	66	108	64	74
y-coordinate	28	26	28	105
Supply area	874	994	876	1012
P_e(MW)	4.7270	4.8812	4.7326	4.8683
Dist. grid	0	2.8284	0	0
PI	1.9323	1.9388	1.9340	1.9518
NPV (M€)	13.86	14.43	13.88	14.33
INV (M€)	7.17	7.44	7.18	7.34

The supply area and the distance to the grid are expressed in Km^2 and Km, respectively. From Tables 4 and 5 it results that population-based methods outperform trajectory methods. The main reason for this fact is the following: trajectory methods explore the search space, while population-based methods explore and exploit the search space. Among the population-based methods, the highest profitability index is obtained by the proposed BPSO algorithm, which outperforms the other two BPSO algorithms considered for comparison. Comparative analysis between the proposed BPSO algorithm and GA (used in this work to validate the results)

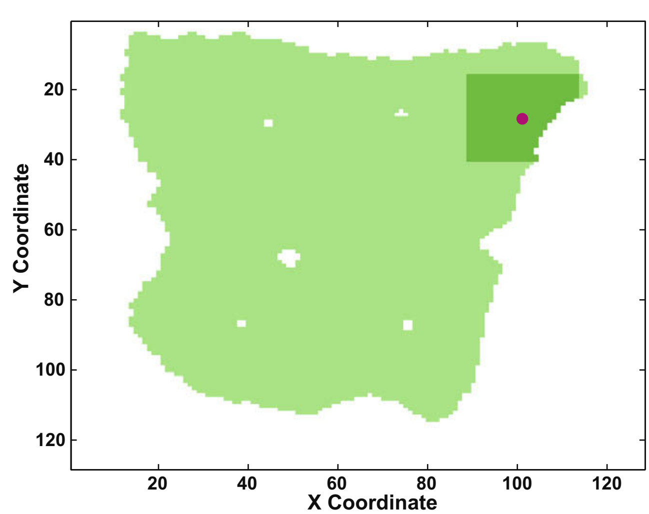

Fig. 8. Placement and supply area of the biomass plant for the best possible solution.

Table 6
Details about the best possible solution.

x-coordinate	101
y-coordinate	28
Supply area (km^2)	1004
P_e(MW)	4.969
Distance to grid (km)	0
PI	1.9918
NPV (M€)	14.86
INV (M€)	7.46

evinces the good performance of our BPSO algorithm. Note that the distance to the grid is usually zero, which means that the electrical line crosses the parcel chosen as optimal placement for the biomass plant.

Fig. 8 shows the placement and supply area of the biomass power plant for the best possible solution. This solution is found when the profitability index takes the highest possible value. Table 6 provides additional information about the best possible solution.

From Table 6, it results that the profitability index for the best possible solution is not far from the median value obtained by the proposed BPSO algorithm. Therefore, it provides near-optimum solutions, being a good candidate for solving discrete optimization problems in real-world applications.

Finally, random walk is applied to the problem we deal with. The median value of the profitability index achieved by random walk, computed over 30 realizations, was 1.8106. This value is the lowest one among those obtained by all tested algorithm, as expected.

5. Conclusion

In this work, four metaheuristics have been are applied and compared in order to determine the optimal placement and supply area of biomass-fueled power plants. In particular, two well-known trajectory methods (Simulated Annealing and Tabu Search) and two commonly used population-based methods (Genetic Algorithms and Particle Swarm Optimization) have been considered for the problem we deal with. Further, a new binary PSO algorithm has been proposed and successfully applied to the problem. The profitability index of the biomass power plant has been used as the fitness function. The power plant is based on gas turbines for producing electric energy from forest residues.

Experimental results show that the proposed BPSO algorithm converges to better solutions than GA and other BPSO algorithms considered for comparison. From the convergence curves, it results that the four assessed algorithms can be ranked as follows: (1) proposed BPSO; (2) BPSO by Kennedy and Eberhart; (3) GA; (4) BPSO by Afshinmanesh et al. Although the BPSO algorithm by Afshinmanesh et al. is the worst ranked, it provides the best performance for a reduced number of iterations (N_{iter} below 10). Experimental results also reveal that population-based methods outperform trajectory methods. Meaningful results about the problem (coordinates of the optimal location, supply area, profitability index, net present value, initial investment, distance to grid, generated power), derived from all metaheuristics, are also reported in the paper. The profitability index for the best possible solution is not far from the median value obtained by the proposed BPSO algorithm. Therefore, it provides near-optimum solutions, being a good candidate for solving discrete optimization problems in real-world applications.

References

[1] REN21 The Renewable Energy Policy Network for the 21st Century, Mexico City; June 2006.
[2] Kumar A, Cameron JB, Flynn PC. Biomass power cost and optimum plan size in western Canada. Biomass Bioenergy 2003;24:445–64.
[3] Varela M, Saez R, Audus H. Large-scale economic integration of electricity from short-rotation woody crops. Solar Energy 2001;70(2):95–107.
[4] Voivontas D, Assimacopoulos D, Koukios EG. Assessment of biomass potential for power production: a GIS based method. Biomass Bioenergy 2001;20(2):101–12.
[5] Jurado F, Cano A. Optimal placement of biomass fuelled gas turbines for reduced losses. Energy Convers Manage 2006;47:2673–81.
[6] IEA Energy Technology Essentials; January 2007. p. 3–4
[7] Jurado F, Ortega M, Cano A, Carpio J. Biomass gasification, gas turbine, and diesel engine. Energy Sources 2001;23(10):897–905.
[8] Willis HL, Scott WG. Distributed power generation: planning and evaluation. New York: Marcel Dekker; 2000.
[9] Blum C, Roli A. Metaheuristics in combinatorial optimization: overview and conceptual comparison. ACM Comput Surveys 2003;35(3):268–308.
[10] Back T. Evolutionary algorithms in theory and practice. Oxford University Press; 1996.
[11] Schwefel HP. Evolution and optimum seeking. John Wiley & Sons; 1995.
[12] Haghifam MR, Falaghi H, Malik OP. Risk-based distributed generation placement. IET Generat Transmission Distrib 2008;2(2):252–60.
[13] Celli G, Ghiani E, Mocci S, Pilo F. A multiobjective evolutionary algorithm for the sizing and siting of distributed generation. IEEE Trans Power Syst 2005;20(2):750–7.
[14] Borges CT, Manzoni A, Viveros EC, Falcao DM. A parallel genetic algorithm based methodology for network reconfiguration in the presence of dispersed generation. In: 17th international conference on electricity distribution, Barcelona, Spain; May 2003.
[15] Jahromi MJ, Farjah E Zolghadri M. Mitigating voltage sag by optimal allocation of distributed generation using genetic algorithm. In: 9th international conference on electrical power quality and utilisation (EPQU-2007); October 2007. p. 1–6.
[16] Nigim KA, Parker P. Heuristic and probabilistic wind power availability estimation procedures: improved tools for technology and site selection. Renew Energy 2007;32(4):638–48.
[17] Dufo-Lopez R, Bernal-Agustin JL. Design and control strategies of PV-diesel systems using genetic algorithms. Solar Energy 2005;79(1):33–46.
[18] Kennedy J, Eberhart RC. Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks; 1995. p. 1942–8.
[19] Eberhart RC, Kennedy J. A new optimizer using particle swarm theory. In: Proceedings of the IEEE sixth international symposium on micro machine and human science; 1995. p. 39–43.
[20] Kennedy J, Eberhart RC. A discrete binary version of the particle swarm algorithm. In: Proceedings of IEEE conference on systems, man and cybernetics; 1997. p. 4104–8.
[21] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. Science 1983:671–80.
[22] Glover F. Future paths for integer programming and links to artificial intelligence. Comput Operat Res 1986;13:533–49.
[23] Holland JH. Adaptation in natural and artificial systems. Ann. arbor: The University of Michigan Press; 1975 [The MIT Press, London 1992].
[24] Goldberg DE. Genetic algorithms in search, optimization and machine learning. Adisson Wesley; 1989.
[25] Michalewicz Z. Genetic algorithms + data structures = evolution programs. Springer Verlag; 1992.
[26] Rastegar R, Meybodi MR, Badie K. A new discrete binary particle swarm optimization based on learning automata. In: Proceedings of IEEE 2004 international conference on machine learning and applications; 2004. p. 456–62.
[27] Afshinmanesh F, Marandi A, Rahimi-Kian A. A novel binary particle swarm optimization method using artificial immune system. In: Proceedings of IEEE international conference on computer as a tool (EUROCON 2005); 2005. p. 217–20.
[28] Amonchanchahigul T, Kreesuradej W. Input selection using binary particle swarm optimization. In: Proceedings of IEEE international conference on computational intelligence for modelling, control and automation; 2006. p. 159.
[29] Sadri J, Suen CY. A genetic binary particle swarm optimization model. In: Proc. IEEE congress on evolutionary computation; 2006. p. 656–63.
[30] Freppaz D, Minciardi R, Robba M, Rovatti M, Sacile R, Taramasso A. Optimizing forest biomass exploitation for energy supply at a regional level. Biomass Bioenergy 2004;26:15–25.