Applied Energy 89 (2012) 347–354

Contents lists available at SciVerse ScienceDirect

Applied Energy

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

Wind speed reconstruction from synoptic pressure patterns using an evolutionary algorithm

L. Carro-Calvo [a], S. Salcedo-Sanz [a,*], L. Prieto [c], N. Kirchner-Bossi [b], A. Portilla-Figueras [a], S. Jiménez-Fernández [a]

[a] Department of Signal Theory and Communications, Universidad de Alcalá, Spain
[b] Department of Physics of the Earth, Astronomy and Astrophysics II, Universidad Complutense de Madrid, Spain
[c] Department of Energy Resource, Iberdrola Renovables, Spain

ARTICLE INFO

Article history:
Received 30 December 2010
Received in revised form 19 July 2011
Accepted 26 July 2011
Available online 3 September 2011

Keywords:
Wind rose reconstruction
Synoptic pressure patterns
Evolutionary algorithms

ABSTRACT

This paper presents an evolutionary algorithm for wind speed reconstruction from synoptic pressure patterns. The algorithm operates in a search space formed by grids of pressure measures, and must classify the different situations into classes, in such a way that a measure of wind speed in a given point is minimized among patterns assigned to the same class. Then, each class is assigned a mean wind speed and direction, so the wind speed reconstruction is possible for a new grid of synoptic pressures. In this paper we present the problem model and the specific description of the evolutionary algorithm proposed to solve the problem. We also show the good performance of the proposed method in the reconstruction of the average wind speed in six wind towers in Spain. The proposed method is applicable to wind speed reconstruction or reconstruction of wind missing data of wind series, specially when there is no other variable or related measure available.

1. Introduction

Wind speed series reconstruction is an important problem currently faced by companies exploding wind farms. Basically this problem is usually faced by obtaining a model for characterizing the wind speed based on previous real wind measures, and then, apply it to obtain values in the past in order to reconstruct wind speed series. Different techniques have been used to obtain these wind speed models, such as statistical methods [1–5], neural networks [6,8,7,9–11], support vector machines [12] or hybridization of some of these algorithms [13].

The majority of the existing techniques to reconstruct wind speed series (and also for long-term wind speed prediction problems) are based on past wind speed data [14], and some of them include other atmospheric variables such as local temperature, radiation or pressure at the measuring point. The main problem with this approach is that these prediction variables are not always available for all the places, so it is sometimes difficult to translate the current techniques or studies to new locations. This problem with local measures is common all over the world, so the idea of considering synoptical information combined with local information has been of interest in the last few years. Synoptic information (mainly atmospheric pressure) is available all over the world and there are reliable records of synoptic pressure fields back to more than one century ago.

The question that arises is then wether we could obtain a reasonably accurate model for wind speed series reconstruction based on this synoptic information (atmospheric pressure), instead on only local information. As has been mentioned before, this idea has been successfully applied to rainfall or pollution prediction in the last few years [15–17]. In these papers, rainfall or pollution measures are explained depending on different pressure synoptic patterns. The objective of this paper is to do something similar with a measure of wind speed in a given point, i.e. obtaining the pressure patterns (pressure clusters) which better explain a wind speed measure in given measurement point. These pressure clustering can be then used to obtain a reconstruction of the wind speed vector (module and direction) in that point. The question to clarify is, first, whether such a reconstruction is possible, and second, how reliable is it with respect to the real measure in a given measurement point.

In this paper we propose to use an evolutionary algorithm to carry out the synoptic pressure clustering for wind speed reconstruction. Evolutionary algorithms are solid population-based approaches which construct the solution to a problem using a loop-fashion procedure, based on the rules of the natural evolution and survival of the fittest individuals. In this paper we present

* Corresponding author. Address: Department of Signal Theory and Communications, Universidad de Alcalá, 28871 Alcalá de Henares, Madrid, Spain. Tel.: +34 91 885 6731; fax: +34 91 885 6699.
E-mail address: sancho.salcedo@uah.es (S. Salcedo-Sanz).

0306-2619/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.apenergy.2011.07.044

the encoding of the problem within the algorithm, and the specific evolutionary operators that we have implemented to design the wind speed reconstruction system. We have tested the performance of the proposed technique in several wind measurement towers in Spain, where we have carried out the reconstruction of a test series after training the evolutionary algorithm. The results are really promising and show the possibility of including synoptic information in the reconstruction or long-term prediction of wind speed.

The rest of this paper is structured as follows: Section 2 presents the problem definition. Section 3 describes the evolutionary algorithm proposed in this paper for wind speed reconstruction. Section 4 presents the results obtained by applying the evolutionary algorithm to data of six measuring towers in Spain. Section 5 closes the paper giving some final conclusions.

2. Wind speed reconstruction based on synoptic pressure patterns

The problem in this paper consists of obtaining a synoptic-scale pressure patterns system which allows to reconstruct the wind speed vector (module and direction) in a given point at a given time, once we have the synoptic pressure pattern associated to that time. Mathematically, this problem can be stated as follows:

Let $\mathbf{d}_t$, $t = 1 \ldots, T$, be a series of daily wind speed real vector (module and direction), measured in a given point (a wind farm, measuring tower, etc.), for a given period of time T. Let P_t, $t = 1 \ldots, T$, be a series of daily synoptic-scale pressure measures in a grid. In our case, each component of P_t is a matrix of 14×13 surface pressure values (182 values), measured in a grid surrounding the Iberian Peninsula (Fig. 1). This problem can be solved by obtaining a pressure clustering, i.e. forming a set of N clusters (centroids) in the space of pressure (space P_t), in such a way that the dispersion of the associated values of $\mathbf{d}_t$ in each cluster is minimized, i.e., in such a way that the following total measure is minimized:

$$f(\mathbf{x}) = \frac{1}{T}\sum_{i=1}^{N}\sum_{t\in\gamma_i}|\mathbf{d}_t - \mathbf{d}^i| \qquad (1)$$

where $\mathbf{x}$ is a vector representing a given synoptic pattern assignment of length T (we consider a series of T pressure patterns to be assigned, in N clusters or centroids), γ_i stands for the set of days belonging to a given class i, and $\mathbf{d}^i$ stands for the mean value of the wind speed within class i. Note that objective function f is a measure of MAE, and tries to explain both wind speed and wind direction. Note also that using f we can reconstruct the average wind rose in the past from the pressure maps.

Thus, we have defined the problem as a clustering problem in the space of matrices P_t, with a function of evaluation in the space of wind speed vectors $\mathbf{d}_t$. These separate working spaces makes the processing of clustering computation and evaluation difficult. Also, note that the high dimension of the pressure space P_t is an extra difficulty. We have tackled the problem by means of an evolutionary-based systems, which will be described and fully analyzed in the next sections.

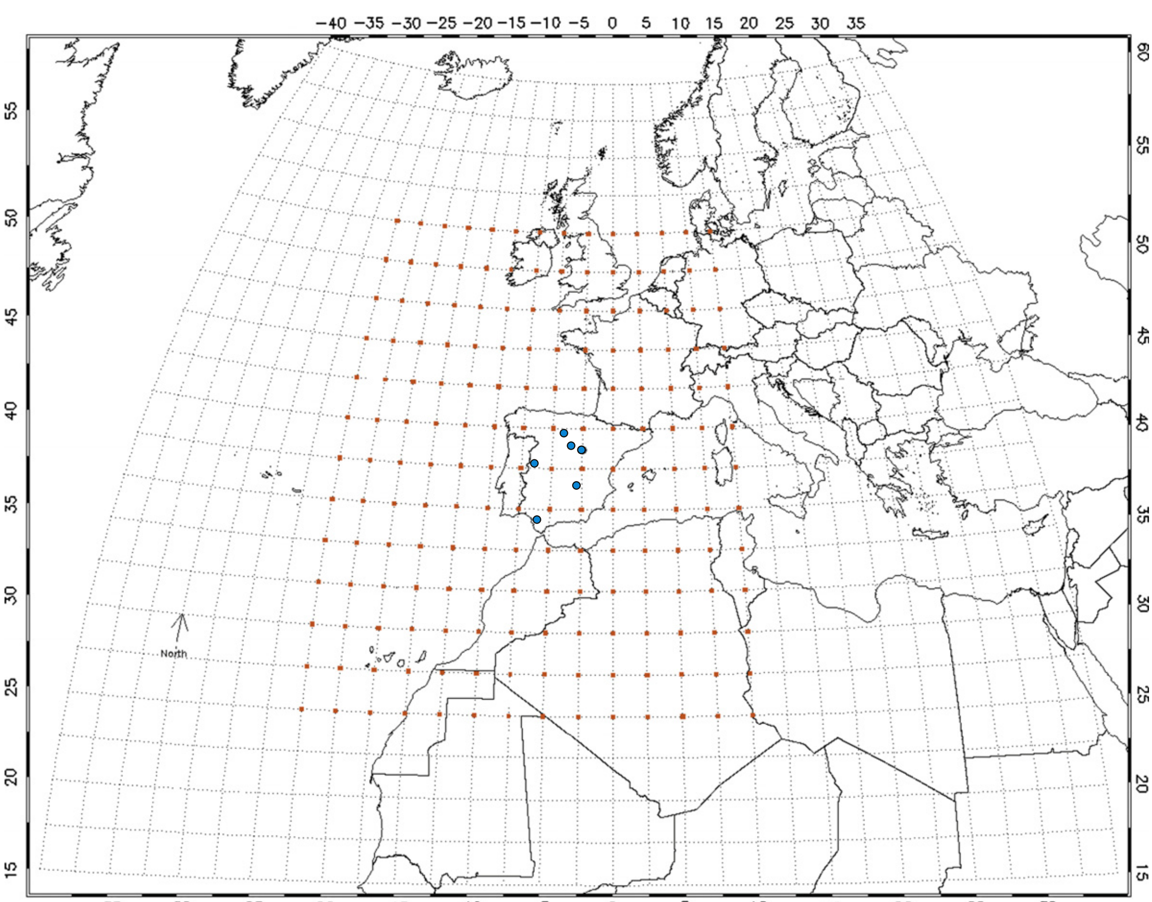

Fig. 1. Pressure measurements grid and location of associated wind speed measurements (in Lambert projection).

3. An evolutionary algorithm for selecting pressure patterns for wind speed reconstruction

In this section we summarize the Evolutionary Algorithm (EA) we propose to obtain a good synoptic pressure clustering for wind speed vector reconstruction. EAs has been applied to many different optimization problems, in a huge range of applications, including energy-related problems [18–22]. Given an optimization problem, an EAs typically start from an initial set, called *population*, of random (candidate) solutions (individuals). These solutions are evolved by the repeated application of a set of evolutionary operators mainly selection, crossover and mutation.

Individuals are typically selected according to the quality of the solution they represent. To measure the quality of a solution, a fitness function is assigned to each individual of the population. Hence, the better the fitness of an individual, the more possibilities the individual has of being selected for reproduction and the more parts of its genetic material will be passed onto the next generations. This is the principle of any selection mechanism incorporated to an EA. The selected individuals are reproduced by means of crossover and mutation operators. In simple terms crossover exchanges some genetic material between two or more individuals, while mutation changes a small part of the genetic material of an individual to a new random value. By applying these operators in a loop fashion, as EA explores the space of possible solutions of an optimization problem. EAs have been shown to be efficient in searching in huge spaces.

In the following sections we describe how we can adapt an EA to look for pressure patterns in an efficient way. We also describe the specific operators implemented in this work to improve the search of the algorithm in this particular problem, following the pseudo-code shown in Fig. 2. Table 1 shows the specific parameters used to run the EA in Section 4.

3.1. Problem encoding

The first important task to face this problem is to find a simplest way to encode matrices P_t. An intuitive and easy form is to reduce the number of points in the grid: instead of using the information of all the points int the grid, we can condense somehow the information by using differences of pressure between points in the grid.

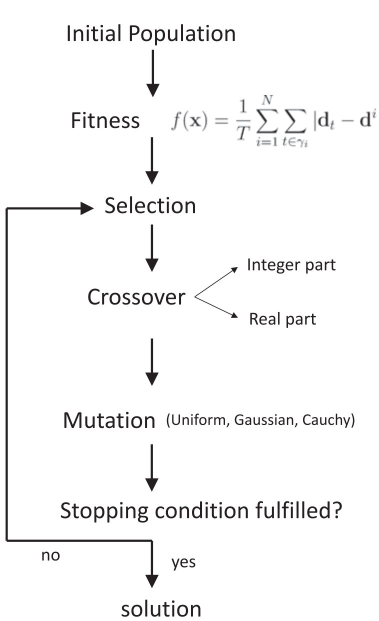

Fig. 2. Pseudo-code of the proposed evolutionary algorithm.

Table 1
Main EA parameters used in the experiments of this paper.

EA parameter	Value
Population length (L)	50
Number of initial centroids (N)	26
Total encoding length	112 (8 integers, 104 reals)
Selection tournaments (p)	80 (80% of 2L)
Mutation probability (P_m)	0.01 per individual
Stopping criteria	Number of generations (500)
Number of runs	30

Of course, different number of differences can be used. In this case, we have encoded the information of the grid by using a set of four pressure differences Dp (eight points in the grid). This way we reduce the space of matrices P_t to an space of four dimensions (space of differences of pressure). Note that we do not fix the points of the grid involved in the calculations of the differences, but the algorithm must locate the optimal points which provide the best possible encoding of the synoptic situation P_t. Thus, the first part of the encoding in the proposed evolutionary algorithm is a set of eight integer numbers $\mathscr{P}_i \in [1,\ldots,182],\ i=1,\ldots,8,$ (number of points in the grid considered), representing four pressure differences, in the following way:

$$Dp_k = P_t(\mathscr{P}_{2k-1}) - P_t(\mathscr{P}_{2k}), \qquad k = 1,\ldots,4 \tag{2}$$

Then, once we have obtained an efficient representation and encoding of matrices P_t (using the four differences Dp_k), we need to encode the different N clusters in the new space of differences of pressure. This can be easy done by encoding each centroid of the cluster in the space, as a four-dimensional vector of real values (one dimension representing each difference of pressure). Thus, each centroid will be represented by a string of 4 real numbers, and the complete set of centroids can be therefore encoded in a vector of 4N real numbers.

The final encoding of the problem in the proposed evolutionary algorithm will be therefore the following:

$$[\mathscr{P}_1,\ldots,\mathscr{P}_8|c_{11},c_{12},c_{13},c_{14},\ldots,c_{N1},c_{N2},c_{N3},c_{N4}] \tag{3}$$

where we have separated the integer part of the encoding from the real part. Note that we apply different operators to the integer and real part of the individuals in the algorithm, as we will describe in the next sections.

3.2. Crossover operator

The crossover operator is known to improve the evolutionary search in many applications [23]. For this problem, we have implemented a mixed crossover approach, different for the integer part of the individual and for the real part. In the case of the integer part (first of the individual in Eq. (3)), we implement a multi-point crossover. After forming couples with the individual in the population, we implement the multi-point crossover by means of a randomly generated binary template of length 8 (length of the integer part of the individual). A 1 in the template means that the corresponding genes of the couple will swap, whereas a 0 means that the genes will not swap. A different template will be generated for all the couples in a generation. Fig. 3 shows a small example of this crossover procedure for the integer part of the individuals.

The crossover operator for the real part of the individual is carried out also implementing a multi-point crossover approach, but in two different modes: first a normal mode, in a similar way as the previous operator defined above, but in the real part of the individual (see Fig. 4). Note that, in this case, the template has length 4N. Also, we consider a second crossover mode, in which

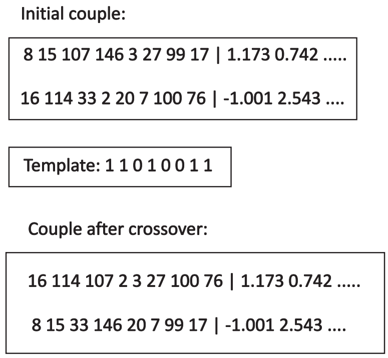

Fig. 3. Crossover applied to the integer part of the individuals.

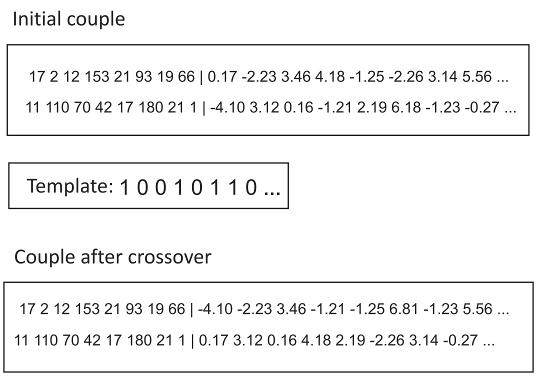

Fig. 4. Crossover applied to the real part of the individuals (parts of centroids interchanged).

we interchange only parts of the individual belonging to a certain centroid (no part of centroids are allowed to be swapped). The length of the template in this second running mode of the crossover is *N*. Fig. 5 shows an example of this second crossover mode for the real part of the individuals. Basically, the first crossover mode generates new centroids by combining existing ones, and the second mode interchanges two centroids from different individuals. Both crossover modes are interesting and have an important role in the evolution of the population. Note that we apply the crossover operator a number of times necessary to obtain an offspring population of the same size as the initial (parents) one (L).

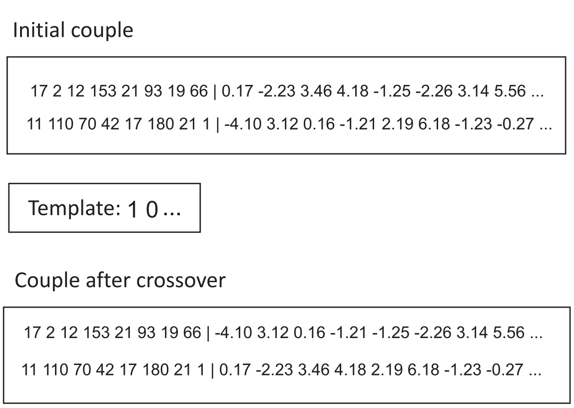

Fig. 5. Crossover applied to the real part of the individuals (complete centroids interchanged).

One important aspect to be taken into account is that, after applying the crossover operators, there may be situations in which a given centroid has no pressure matrix assigned. When one of these cases occur, the void centroid is erased and its components reassigned to the proximity of a valid centroid (one with pressure matrices assigned). This procedure is carried out by assigning the coordinates of the centroid (differences of pressure) and slightly modifying them by adding a Gaussian noise to each coordinates. The pressure matrices are then reassigned to their nearest centroid. With this easy procedure we avoid the presence of void centroids in the evolutionary algorithm, which distorts somehow the efficiency of the search.

3.3. Mutation operator

Mutation operator is applied with a very low probability ($P_m = 0.01$) to each individual in the offspring population. Once a given individual is going to be mutated, the procedure of mutation is divided into two different versions, depending on whether it is applied on the integer part or to the real part of the individual. The mutation of the integer part is carried out by means of a integer randomized substitution of the current values of the individual, by different integers, in the interval [1, 182].

We have tested different operators for the mutation in the real part of the individual. First, we have tested to add uniform noise in the interval $[-5, 5]$, to a number of randomly chosen values of the real part in the mutated individual. We have also tested to change the values in this real part of each individual by means of Gaussian-based mutation and Cauchy-based mutation, which have been proven to improve the search in several problems [24–26]. We will compare the performance of the evolutionary algorithm including all these mutation operators in Section 4.

3.4. Selection operator

In this paper we use a tournament selection, which will be applied to the joint population formed from merging the initial and offspring populations. The result of the selection operator will be a single population, of size L, which will be the parents of the next generation of individuals. Basically, once the complete joint population of parents and offspring is formed, the standard tournament selection, as described in Ref. [24], has two main steps:

- Conduct pairwise comparison over the union of parents and offspring: for each individual, p opponents are chosen uniformly at random from all the parents and offspring. For each comparison, if the individual's fitness is better than the opponent's, it receives a "win".
- Select the L individuals out of the union of parents and offspring that have the most "wins" to be parents of the next generation.

Using this easy procedure, the remaining L individuals act as the parents of the next generations, and the crossover and mutation operators are applied again in a loop fashion, until the maximum number of generations are reached.

4. Experimental part

This section presents the experimental part of this study. First we describe the available data used to test the proposed EA, and then we show the specific reconstruction results in each measuring tower considered.

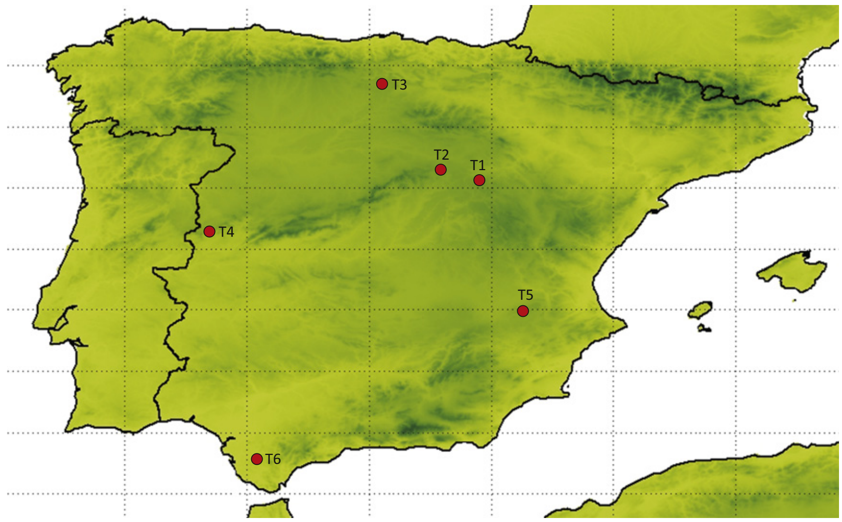

Fig. 6. Locations of the six different meteorological towers considered in this paper.

4.1. Data available

Wind speed real data from 10 years (1999–2008) of six meteorological towers in different points of Spain (Fig. 6 shows the location of the towers in Spain) are available for this study. The data consist of wind speed and direction data, taken at the measurement towers, at 40 m of height every 10 min. Averages over 24 h are considered to obtain daily data vectors $\mathbf{d}_t$ in each measurement tower. On the other hand, average daily pressure maps for the same period have been obtained from the National Center for Environmental Prediction/National Center for Atmospheric Research Reanalysis Project (NCEP/NCAR) [27], which are public data profusely used in climatology and meteorology applications. As previously mentioned, we have considered an uniform grid in latitude and longitude, shown in Fig. 1, with 182 measurement points. Recall that the proposed evolutionary algorithm uses this value as a parameter of the encoding (in the differences of pressure, integer part of the encoding). We have set the two final years (2007 and 2008) of the data for training and the rest of years (1999–2006) as a test set in each tower, where we measure the quality of the proposed EA in reconstructing the wind speed series.

4.2. Results

Table 2 shows the average fitness (over 30 runs) and standard deviation obtained with the proposed EA, when applying different mutation operators (uniform, Gaussian and Cauchy mutations), for the six measuring towers considered in this paper. Since the fitness value (given by Eq. (1)) is a measure of MAE, the lowest values indicate a better performance. In general the EA proposed works well, with small variations depending on the mutation operator used for the real part of the encoding. We have carried out a comparison with the IFEP algorithm described in Ref. [24], as a reference approach. The IFEP approach in Ref. [24] is an evolutionary optimization approach, which uses Gaussian and Cauchy mutations to guide the search. The IFEP can be adapted to work on the proposed encoding for the current problem (some simple modifications must be included to manage the integer part of the encoding). Table 3 shows the results obtained when the IFEP algorithm is applied on the different measuring towers considered. The results show that the IFEP is also able to solve the problem in an accurate way, though the proposed EA obtains in general better results in all the towers considered. This may indicate that in this particular problem the inclusion of crossover improves the performance of the evolutionary algorithm.

Following, we graphically present the results of the reconstruction of average wind speed in each tower in the test years consid-

Table 2
Performance of the proposed evolutionary algorithm with different mutation operators in all the measuring towers considered (from T1 to T6).

Mutation	Av. fitness (Eq. (1))	Std dev.
T1		
Uniform $[-5, 5]$	3.5950	0.1166
Gaussian (0, 1)	3.1479	0.1610
Cauchy ($t = 1$)	3.1554	0.1389
T2		
Uniform $[-5, 5]$	3.1602	0.1158
Gaussian (0, 1)	3.1726	0.1267
Cauchy ($t = 1$)	3.1827	0.1079
T3		
Uniform $[-5, 5]$	3.7857	0.0754
Gaussian (0, 1)	3.7921	0.0921
Cauchy ($t = 1$)	3.7877	0.0686
T4		
Uniform $[-5, 5]$	3.6986	0.1320
Gaussian (0, 1)	3.6703	0.1211
Cauchy ($t = 1$)	3.6882	0.1157
T5		
Uniform $[-5, 5]$	3.1069	0.1444
Gaussian (0, 1)	3.1479	0.1610
Cauchy ($t = 1$)	3.1554	0.1389
T6		
Uniform $[-5, 5]$	2.8835	0.1348
Gaussian (0, 1)	2.8939	0.1145
Cauchy ($t = 1$)	2.8959	0.1349

Table 3
Performance of the IFEP algorithm in the problem tackled (average of 30 runs in all measuring towers).

Tower	Av. fitness (Eq. (1))	Std dev.
T1	3.6499	0.0872
T2	3.2272	0.0908
T3	3.8259	0.0520
T4	3.7262	0.0851
T5	3.3052	0.1439
T6	2.9513	0.1031

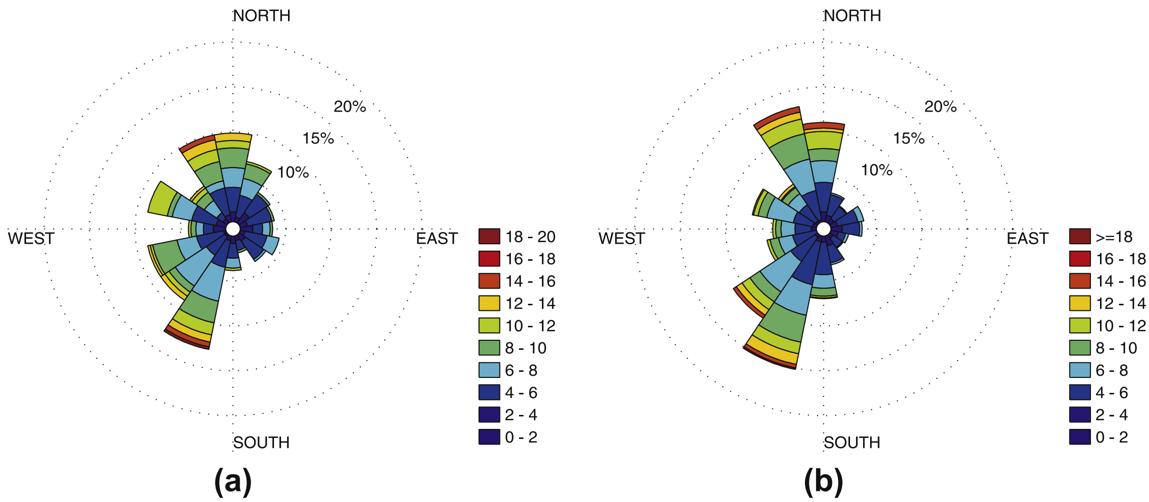

Fig. 7. Tower 1: (a) Real wind rose. (b) EA reconstructed wind rose.

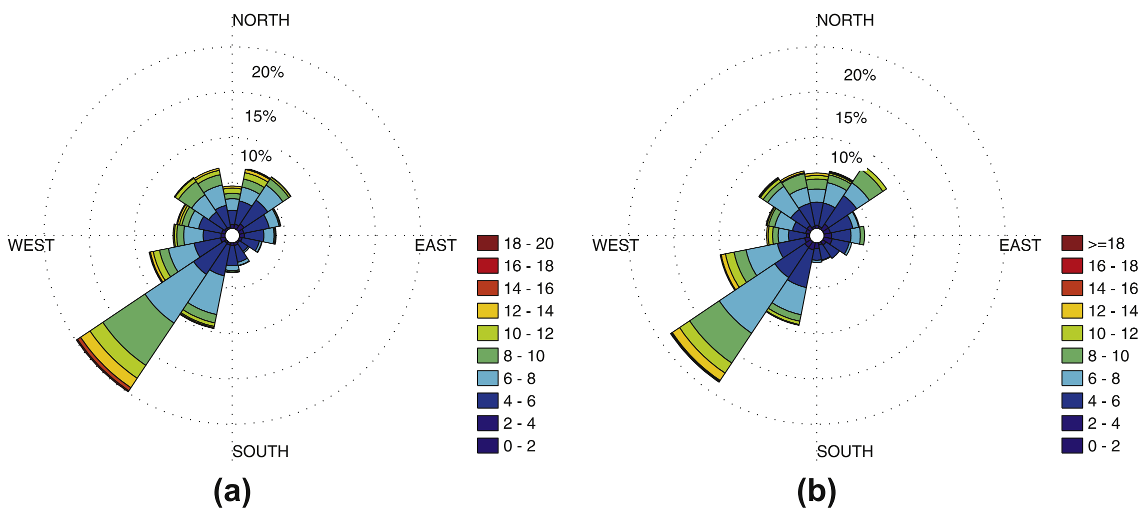

Fig. 8. Tower 2: (a) Real wind rose. (b) EA reconstructed wind rose.

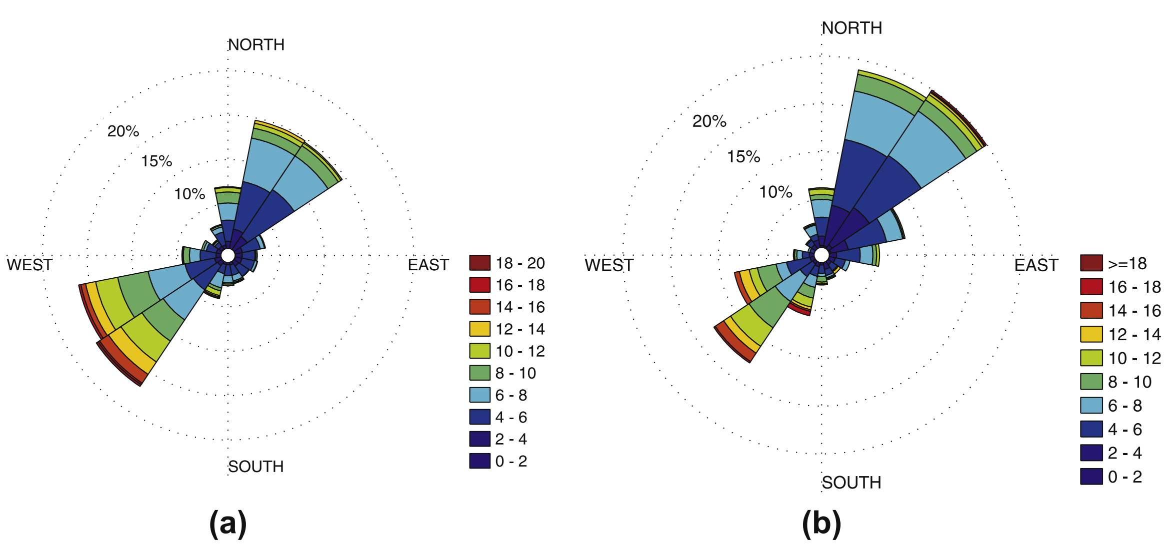

Fig. 9. Tower 3: (a) Real wind rose. (b) EA reconstructed wind rose.

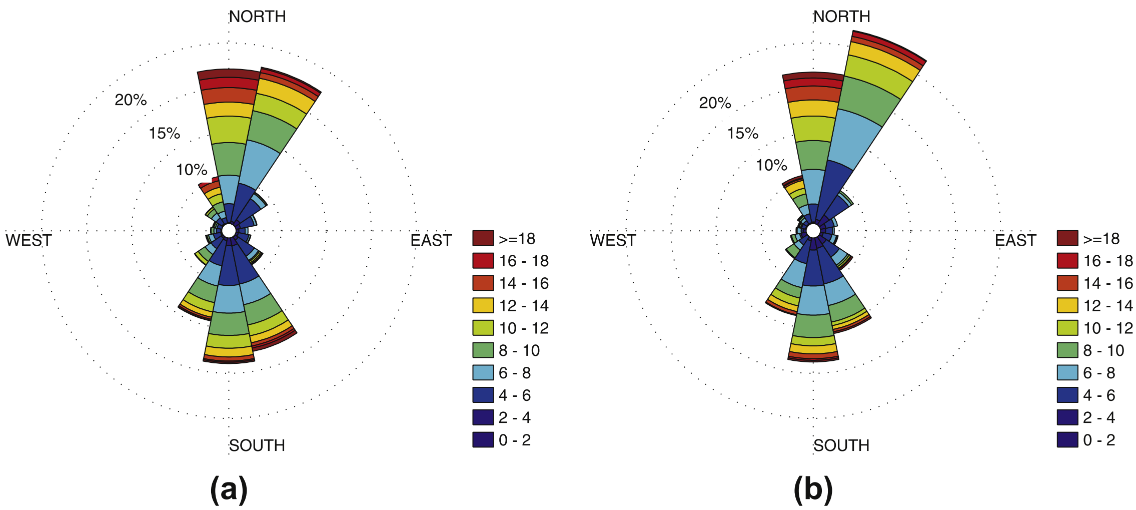

Fig. 10. Tower 4: (a) Real wind rose. (b) EA reconstructed wind rose.

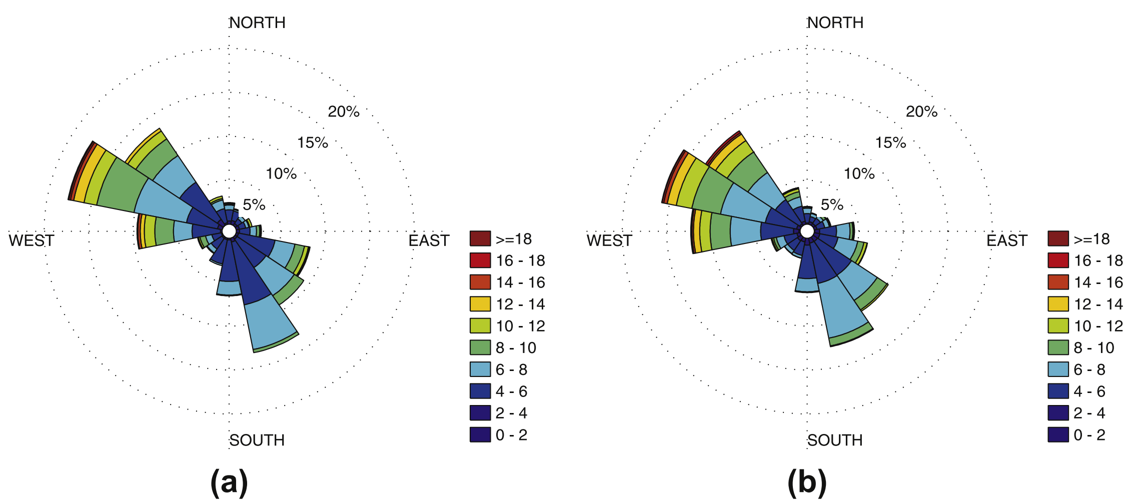

Fig. 11. Tower 5: (a) Real wind rose. (b) EA reconstructed wind rose.

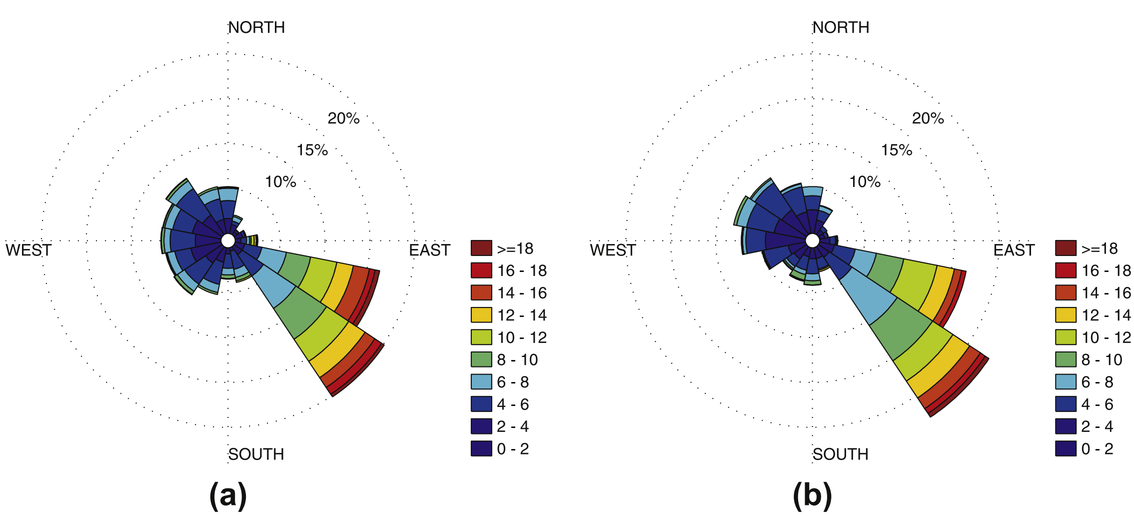

Fig. 12. Tower 6: (a) Real wind rose. (b) EA reconstructed wind rose.

ered (1999–2006), by comparing the reconstructed wind rose (best individual found in the EA evolution) with the real one in each tower. Note that we have not found alternative algorithms in the literature which use synoptic pressure measure to reconstruct the wind speed, so the comparison must be carried from the real measures obtained. Figs. 7–12 show the comparison of the real and reconstructed wind speed using the proposed EA, for Towers 1–6, respectively. Fig. 7 shows this comparison for Tower 1, sited at the north part of Guadalajara, Spain. The wind speed reconstruction for the 8 years considered is quite consistent to the real measure. Small increasing in wind module and direction frequency with respect to the real wind rose is obtained in the EA reconstruction for south, south-west, north-west and north directions, whereas north-east directions have been slightly underestimated by the EA approach. Fig. 8 shows the EA reconstruction and real average wind speed for Tower 2, sited also in Guadalajara, close to Tower 1 (about 20 km). In this case the reconstruction is even better than for Tower 1, and the EA is able to accurate locate the main wind directions, with a slightly decreasing of the wind module and frequency for the south-west direction. Fig. 9 shows the real and reconstructed wind rose for Tower 3, sited in La Rioja, Spain. In this case the reconstruction is not so good, and, in spite of the EA localizes the main wind directions, the reconstruction overestimates the north-east wind components, whereas it underestimates the south-west directions. EA reconstruction for Tower 4 (sited at Salamanca, Spain) is again quite good (see Fig. 10). The EA slightly overestimates the north-east directions, but the southern components are accurately modeled. Wind reconstruction for Tower 5 (Albacete, Spain), shown in Fig. 11 is really accurate. Only slightly underestimation of west components can be observed, but the reconstruction of the average wind speed from pressure data in this tower can be considered as good. Finally, Fig. 12 shows the average wind speed reconstruction of the test period considered for Tower 6, sited in Cádiz, Southern Spain. In this case, the figure shows that the EA slightly overestimates south-east components of the wind, whereas it underestimates south and south-west components. Anyway, the wind speed reconstruction in this tower is really reasonable, since the main wind direction is perfectly obtained by the proposed EA.

Summarizing, the proposed method based on evolutionary clustering in pressure has been shown to be a valid method to reconstruct the wind speed vector (module and direction), in an accurately way. Thus, this method can be used to wind reconstruction in wind farms or obtention of wind missing values from pressure patterns when there is no other variables or measures available.

5. Conclusions

In this paper we have proposed a novel method for wind speed reconstruction based on evolutionary computing. Specifically, we have proposed an evolutionary algorithm which looks for the best clustering of synoptic pressure situations, in terms of an objective function which involves a wind speed measure in a given point of study. We have shown the good performance of the proposed technique in the reconstruction of the average wind speed in six wind towers in Spain, where the proposed algorithm has been able to reconstruct daily average wind speed with some accuracy. The proposed approach is a good option to wind speed reconstruction series when there is no local data available. In these cases, the proposed approach and a synoptic pressure database will be enough to obtain a first wind speed reconstruction.

Acknowledgement

This work has been partially supported by Spanish Ministry of Industry, Tourism and Trading, under an Avanza 2 project, number TSI-020100-2010-663.

References

[1] Torres JL, García A, De Blas M, De Francisco A. Forecast of hourly average wind speed with ARMA models in Navarre (Spain). Solar Energy 2005;79:65–77.
[2] Khashei M, Bijari M, Raissi-Ardali G. Improvement of auto-regressive integrated moving average models using fuzzy logic and artificial neural networks (ANNs). Neurocomputing 2009;72(4-6):956–67.
[3] Morales JM, Minguez R, Conejo AJ. A methodology to generate statistically dependent wind speed scenarios. Appl Energy 2010;87(3):843–55.
[4] Erdem E, Shi J. ARMA based approaches for forecasting the tuple of wind speed and direction. Appl Energy 2011;88(4):1405–14.
[5] Liu H, Erdem E, Shi J. Comprehensive evaluation of ARMA-GARCH(-M) approaches for modeling the mean and volatility of wind speed. Appl Energy 2011;88(3):724–32.
[6] Fadare DA. The application of artificial neural networks to mapping of wind speed profile for energy application in Nigeria. Appl Energy 2010;87(3):934–42.
[7] Mellit A, Kalogirou SA, Hontoria L, Shaari S. Artificial intelligence techniques for photovoltaic applications: a review. Prog Energy Combust Sci 2009;34(5):406–19.
[8] Li G, Shi J. On comparing three artificial neural networks for wind speed forecasting. Appl Energy 2010;87(7):2313–20.
[9] Barbounis TG, Theocharis JB. Locally recurrent neural networks for long-term wind speed and power prediction. Neurocomputing 2006;69(4-6):466–96.
[10] Li G, Shi J. Application of Bayesian model averaging in modeling long-term wind speed distributions. Renew Energy 2010;35(6):1192–202.
[11] Mohandes M, Rehman S, Rahman SM. Estimation of wind speed profile using adaptive neuro-fuzzy inference system (ANFIS). Appl Energy 2011;88(11): 4024–32.
[12] Mohandes MA, Halawani TO, Rehman S, Hussain AA. Support vector machines for wind speed prediction. Renew Energy 2004;29:939–47.
[13] Bouzgou H, Benoudjit N. Multiple architecture system for wind speed prediction. Appl Energy 2011;88(7):2463–71.
[14] Carranza O, Figueres E, Garcera G, Gonzalez LG. Comparative study of speed estimators with highly noisy measurement signals for wind energy generation systems. Appl Energy 2011;88(3):805–13.
[15] Trigo RM, DaCamara CC. Circulation weather types and their influence on the precipitation regime in Portugal. Int J Climatol 2000;20:1559–81.
[16] Paredes D, Trigo RM, García-Herrera R, Trigo IF. Understanding precipitation changes in Iberia in early spring: weather typing and storm tracking approaches. J Hydrometeorol 2006;7:101–13.
[17] Chen ZH, Cheng SY, Li JB, Guo XR, Wang WH, Chen DS. Relationship between atmospheric pollution processes and synoptic pressure patterns in northern China. Atmos Environ 2008;42(24):6078–87.
[18] Kusiak A, Zheng H. Optimization of wind turbine energy and power factor with an evolutionary computation algorithm. Energy 2010;35(3):1324–32.
[19] Kalogirou SA. Optimization of solar systems using artificial neural-networks and genetic algorithms. Appl Energy 2004;77(4):383–405.
[20] Liao CC. Genetic k-means algorithm based RBF network for photovoltaic MPP prediction. Energy 2010;35(2):529–36.
[21] Jursa R, Rohrig K. Short-term wind power forecasting using evolutionary algorithms for the automated specification of artificial intelligence models. Int J Forecast 2008;24:694–709.
[22] Vaisakh K, Srinivas LR. A genetic evolving ant direction DE for OPF with non-smooth cost functions and statistical analysis. Energy 2010;35(8):3155–71.
[23] Eiben AE, Smith JE. Introduction to evolutionary computing. Springer-Verlag; 2003.
[24] Yao X, Liu Y, Lin G. Evolutionary programming made faster. IEEE Trans Evol Comput 1999;3(2):82–102.
[25] Lee CY, Yao X. Evolutionary programming using the mutations based on the Lévy probability distribution. IEEE Trans Evolut Comput 2004;8(1):1–13.
[26] Liang KH, Yao X, Newton CS, Hoffman D. A new evolutionary approach to cutting stock problems with and without contiguity. Comput Operat Res 2002;29(12):1641–59.
[27] http://www.esrl.noaa.gov/psd/data/reanalysis/reanalysis.shtml.