International Conference on Renewable Energies and Power Quality (ICREPQ'14)
La Coruña (Spain), 25th to 27th March, 2015
Renewable Energy and Power Quality Journal (RE&PQJ)
ISSN 2172-038 X, No.13, April 2015

Metaheuristic approach to the Holt-Winters optimal short term load forecast

E.Eusébio[1], C. Camus[1] and C. Curvelo[1]

[1] Department of Power Systems Engineering and Automation
ISEL, Instituto Politécnico de Lisboa
Rua Conselheiro Emídio Navarro nº 1, 1959-007 Lisboa (Portugal)
Phone: +351 218317000, e-mail: eaeusebio@deea.isel.ipl.pt, ccamus@deea.isel.ipl.pt, 31390@alunos.isel.pt

Abstract. Electricity short-term load forecast is very important for the operation of power systems. In this work a classical exponential smoothing model, the Holt-Winters with double seasonality was used to test for accurate predictions applied to the Portuguese demand time series.
Some metaheuristic algorithms for the optimal selection of the smoothing parameters of the Holt-Winters forecast function were used and the results after testing in the time series showed little differences among methods, so the use of the simple local search algorithms is recommended as they are easier to implement.

Key words

Electricity demand, Load forecast, Combinatorial optimization, Evolutionary algorithms.

1. Introduction

Electricity demand forecasting is very important for the management of power systems. Long-term forecasts of the peak electricity demand are needed for capacity planning and maintenance scheduling. Medium-term demand forecasts (from several weeks up to one year) are required for power system operation and planning. Short-term load forecasts are required for scheduling and regulation of power systems. Short-term forecasts are also required by transmission companies when a self-dispatching market is in operation [1]. Error in predicting electricity load has significant cost implications for companies operating in competitive power markets [2].
In the short run, the load is mainly influenced by meteorological conditions, seasonal effects (daily, weekly cycles, calendar holidays) and special events. Weather related variation is certainly critical in predicting electricity demand for lead times beyond a day-ahead. However when the interest is in shorter lead times (four, six hours until a day-ahead), a univariate model will be sufficient because the meteorological variables tent to change in a smooth fashion, which will be captured in the demand series itself. It can be used also for longer lead times (three, four days a week maximum) when there is a lack of readily available weather forecasts.

The stochastic nature of demand as a function of time has been frequently modeled with seasonal ARIMA and state space models. ARIMA modeling is also used by many as a sophisticated benchmark for evaluating alternative proposals such as neural networks. Artificial neural networks (ANNs) have featured prominently in the load forecasting literature [3]. Their nonlinear and nonparametric features have been useful for multivariate modeling in terms of weather variables.
Simpler methods, such as general exponential smoothing are always attractive due to the small number of parameters involved, which make them easy to implement. In a study [4]-[5], methods for short-term load forecast are reviewed to compare a variety of univariate methods. These studies concluded that a double seasonal version of Holt-Winters (HW) exponential smoothing was the most accurate method.

2. Holt-Winters Exponential Smoothing for Double Seasonality

The exponential smoothing models are based on the updating, for each period, of up to three parameters: mean level (simple smoothing model), mean level and trend (Holt model), mean level, trend and seasonality (HW model). These models are also known in the literature as one, two and three parameter exponential smoothing respectively. For the last one several variations can be considered whether the trend and seasonality are additive or multiplicative in the model and also there are studies that refer an increase in the accuracy of the method when the projected trend can be damped [6]. The HW method with multiplicative seasonality has been widely used [7]-[8]. For short term load forecast applications, the double seasonal method, developed to forecast time series with two seasonal cycles is well suited. In this case there is a shorter cycle of 24 hours and a longer one of 168. The application requires an extension of the standard HW exponential smoothing formulation to accommodate the two seasonal cycles in the electricity demand series. This involves the introduction of an additional seasonal index and an extra smoothing equation for the new seasonal index [8]. The formulation for double multiplicative seasonality and additive damped trend is given in the following expressions:

$$n_t = \alpha \frac{y_t}{D_{t-S_1} W_{t-S_2}} + (1-\alpha)(n_{t-1} + \phi b_{t-1}) \tag{1}$$

$$b_t = \beta (n_t - n_{t-1}) + (\phi - \beta) b_{t-1} \tag{2}$$

$$D_t = \gamma \frac{y_t}{n_t W_{t-S_2}} + (1-\gamma) D_{t-S_1} \tag{3}$$

$$W_t = \delta \frac{y_t}{n_t D_{t-S_1}} + (1-\delta) W_{t-S_2} \tag{4}$$

$$y_{t+k} = \left(n_t + \sum_{i=1}^{k} \phi^i b_t\right) D_{t+k-m_1 S_1} W_{t+k-m_2 S_2} + \lambda^k \left(y_t - \left((n_t + \phi b_t) D_{t-S_1} W_{t-S_2} \right) \right) \tag{5}$$

Where, n_t and b_t are the smoothed level and trend; D_t and W_t are the daily and weekly seasonal indices, respectively; α, β, γ, δ are the smoothing parameters; and y_{t+k} is the k step ahead forecast made from origin *t*. The parameter ϕ stands for the damped tend meaning that if $0 < \phi < 1$ the trend is damped and the term involving parameter λ, in the forecast function expression (5), is an adjustment for the first order autocorrelation error. $S_1=24$ and $S_2=168$ are the seasonal cycles size and m_1 and m_2 are the number of corresponding cycles in the k-ahead prediction.

The objective is to find the best exponential smoothing parameters. The forecasting model having the lowest error over a period of time is most desirable so the objective is to select the values of the smoothing parameters associated with the lowest prediction error. The error measure usually used is the sum of squared errors (SSE) or the mean square error (MSE) or its root (RMSE). Other measures of error are the mean absolute error (MAD) or the mean absolute percentage error (MAPE). Anyway, for the objective function, an error function or a convex combination of error functions can be selected and the optimal values of the smoothing parameters can be obtained solving the following non-linear problem:

$$\begin{array}{l} Min \;\; E(\alpha, \beta, \gamma, \delta, \phi, \lambda) \\ s.t. \;\; (\alpha, \beta, \gamma, \delta, \phi, \lambda) \in \Omega \end{array}$$

Where $E(\alpha, \beta, \gamma, \delta, \phi, \lambda)$ is the selected error function and the feasible region Ω is determined by the *4xmxp* recursive equations (1)-(4) where *m* is the number of weeks used to fit the model, *p* is the number of hours within a week, 168, and twelve boundary constraints for $\alpha, \beta, \gamma, \delta, \phi, \lambda$. Implementation of the *4xmxp* equality equations using a spreadsheet enables these expressions to be handled, associating each period with one of the rows of the active spreadsheet [9].

2.1 Model Estimation, Selection and Prediction

The parameters in the HW model with double seasonality and damped trend can be obtained by minimizing the one-step-ahead sum of squared errors, equation (6).

$$SSE = \sum_{t=1}^{n} \left(y_t - \hat{y}_t\right)^2 \tag{6}$$

Where *n* is the number of observations in the series, y_t the demand value for period *t* and $\hat{y}_t$ the corresponding forecast. This objective function is hard to compute if too many data is used. The one-step-ahead forecast $\hat{y}_t$ is computed using equation (5) and the four components of the HW model (level, trend, daily and weekly seasonality) that are updated recursively using equations (1)-(4) for fixed values of $\alpha, \beta, \gamma, \delta, \phi$.

The problem constrains are only the twelve boundary constrains representing the maximum and minimum values of the six parameters to optimize, since all equality constraints are evaluated as elements of the objective function. The problem can be formulated as follows:

$$\begin{array}{ll} Min & SSE = \sum_{t=1}^{n} \left(y_t - \hat{y}_t\right)^2 \\ s.t. & \alpha, \beta, \gamma, \delta, \phi, \lambda \in [0,1] \end{array}$$

The objective is thus to find the combination of parameters $\alpha, \beta, \gamma, \delta, \phi, \lambda$ that minimizes SSE. The four components of the HW model (n_t, b_t, D_t, and W_t) are updated recursively so it is necessary to estimate the corresponding initial values.

After computing the initial values the forecasting parameters can be calculated with an appropriate optimization algorithm.The model is then tested for prediction intervals for k periods ahead. For the tests the MAPE as well as the MSE are estimated.

$$MAPE = \frac{100}{m} \sum_{t=1}^{m} \frac{\left|y_t - \hat{y}_t\right|}{y_t} \tag{7}$$

$$MSE = \frac{SSE}{m} \tag{8}$$

Where *m* is the number of periods ahead tested.

3. Combinatorial Optimization

As mentioned, the optimal combination of parameters $\alpha, \beta, \gamma, \delta, \phi, \lambda$ is going to be obtained by metaheuristics. Metaheuristics are applied to problems for which there is no satisfactory solving algorithm or when it is not practical to implement the solving algorithm. Metaheuristics are procedures based on exploring the search spaces. Metaheuristics can be seen as a frame work that can be applied to different optimization problems with relatively few changes in adapting to a specific problem [10]. Attractive features of metaheuristic algorithms are simplicity, robustness and flexibility. Also, they allow the incorporation of strategies to make more effective solving procedures. Metaheuristic algorithms have parameters to control the search, which have to be carefully tuned to obtain good performance. Hence, the adaptive control mechanism of such parameters has to be tested and it is an important task for success.

First we need to have a good evaluation function and sometimes the objective function is a good choice. Then, we need to represent the search space, which is the set of solutions that may potentially be visited by the algorithm. The metaheuristic algorithms are based on the iteration of the following two steps:

```
        Step 1. Generate initial solutions $u$;
        Step 2. Improve $u$ by using local search (LS).
```

3.1 Model Estimation, Selection and Prediction

One way to search for good solutions is to start from any solution, evaluate its neighbors, move to the best neighbor, and keep doing this until a good solution is found. This search method is called Hill climber (HC) or local improved procedure [11] and its algorithm can be described as follows:

```
Step 1 Start from an initial random solution $u$ and evaluate
       it by calculating $f(u)$;
Step 2 Define some neighbor solutions set $N(u)$ and
       evaluate those solutions;
Step 3 Move to the best neighbor, while a stopping
       criterion is not met go to step 2.
```

The disadvantage with hill climber is that: if the objective function is not a convex one, then there is no guarantee of being close to the optimum when in a local optimum. Also, if there is more than one local optimum, the local solution found depends on the algorithm starting initial solution.

3.2 Simulated Annealing (SA)

To escape local optima the simulated annealing algorithm (SA) follows decisions on choosing a neighbor that depends on a probability function that varies with the neighbors' relative performance [9]. The acceptance probability is controlled by a parameter T called temperature. The value of parameter T is important. For instances, with large T the process is nearly random, with small T the process is nearly greedy.

The SA algorithm varies T along the optimization process, whose idea stems from the physical annealing process [11]. Temperature T is usually set to large (i.e., the probability of moving to a worse solution is high) in the beginning of the search. T is then gradually decreased as the search proceeds. To vary the temperature T the following exponentially decreasing function can be used.

$$T(t) = T_{\max} e^{-Rt} \tag{9}$$

Where, R is the temperature decreasing rate and $T_{\max}$, the initial temperature and t represents time and can be implemented with an iteration counter. The SA algorithm can be stated as follows:

```
Step 1   Initialize, $t = 0$ and $T = T_{\max}$; Choose a solution
         (at random) say $u$; Evaluate the solution by
         calculating $f(u)$;
Step 2   Find (at random) a neighbor solution $v$ and
         evaluate it by calculating $f(v)$;
Step 3   Compute
         $$p = \begin{cases} 1, & f(v) \le f(u) \\ e^{\frac{f(v)-f(u)}{T}}, & f(v) > f(u) \end{cases};$$
Step 4   Accept the new solution $v$ with probability $p$;
         Make $t = t + 1$; Set $T = T(t)$;

         if $T > T_{\min}$ go to Step 2 otherwise stop.
```

3.3 Evolutionary Algorithms, Genetic Algorithms (GA)

The algorithms presented above are local search algorithms; these algorithms search by moving from one solution into one of its neighbors. The evolutionary algorithms are different. They explore the space with a set (population) of solutions at the same time. The evolutionary algorithms manage the survival of a set of solutions based on the principles of natural selection. Evolution takes place in time as the fittest get a chance to combine their genes and contribute to the next generation. At the beginning of a GA, representations for possible solutions, which are often called chromosomes or individuals, must be developed. Different combinations of genes form different chromosomes. Each chromosome is a possible solution of the problem. The set of chromosomes is called the population of the generation. Chromosomes in a generation are forced to evolve toward the next generation by three basic GA operators, reproduction, crossover and mutation [12]. A GA can be stated as follows:

```
Step 1   Make t = 0; Initialize a population of solutions
         P(0), at random and evaluate;
Step 2   $t = t+1$;
Step 3   Select the fittest from P(t-1) to built P(t);
Step 4   Cross P(t);
Step 5   Mutate some solution from P(t);
Step 6   Evaluate P(t); Go to step 2 or stop if close to
         saturation.
```

3.4 Evolutionary Algorithms, Particle Swarm (PS)

The particle swarm algorithm is also a good method to solve this problem in the real-number space [13]. Each particle represents a solution point. The position of each particle i is assigned by vector x_i. Change of position of a particle is addressed as velocity v_i which represents a vector of numbers that are added to the position coordinates in order to move the particle from one step to another.

$$x_i(t)=x_i(t-1)+v_i(t) \quad (10)$$

The direction of movement of each particle is a function of current position and velocity, the location of the individual's best position found and the best neighbor's position. So the formula for changing velocity is the following:

$$v_i(t)=v_i(t-1)+\varphi_1(p_i-x_i(t-1))+\varphi_2(p_g-x_i(t-1)) \quad (11)$$

where p_i is the best state found so far and p_g is the neighborhood best. The φ variables are random numbers. The algorithm for particle swarm in continuous numbers can be as follows:

```
Step 1 Make $t = 0$; Initialize a population of solutions
       $x_i(1)$(particles positions), at random and evaluate
       them; Set randomly the first value for velocity;
       Make $p_i=x_i(1)$;
Step 2 $t = t+1$;
Step 3 Select the fittest $p_g$ from $x(t-1)$;
Step 4 Choose randomly values for $\varphi_1$, $\varphi_2$;
Step 5 Update velocity $v_i(t)$ of each particle
       $v_i(t)=v_i(t-1)+\varphi_1(p_i-x_i(t-1))+\varphi_2(p_g-x_i(t-1))$;
Step 6 If $v_i(t)> V_{max}$ then $v_i(t)=V_{max}$ else if $v_i(t)<-V_{max}$ then
       $v_i(t)=-V_{max}$;
Step7  Update particles' positions. $xi(t)=xi(t-1)+vi(t)$.
       Evaluate new positions and update pi;
Step 8 Go to step 2 or stop if close to saturation.
```

4. Implementation Issues for the Case Study

For obtaining the optimal parameters for HW double seasonal smoothing forecasting equation using the previous described methods the following measures were taken:

The evaluation function was the objective function though it has the drawback of being highly time consuming to perform a solution's evaluation there wasn´t any other good alternative.

The first solution or the first population of solutions was random with Uniform distribution between [0, 1] for each of the six parameters that constitute a solution $x(\alpha, \beta, \gamma, \delta, \phi, \lambda)$.

The neighbor solutions were obtained for each element of current solution by, $n_i=x_i+N(0,0.1)$, where $N(0,0.1)$ is a random observation of a normal distribution with mean 0 and standard deviation of 0.1, and the neighbor solution is feasible (all parameters remain within the boundary constraints).

For the simulating annealing implementation the choosing of parameters T_{max} and R were made by a trial and error process. The values $T_{max}=20000$ and $R=0.05$, performed reasonably. The probability of changing to a worst neighbor was set to 0.7.

For the GA many population sizes were tested between 20 to 100 different solutions. The selection of the fittest solutions was performed by a random tournament between solutions. The crossover was performed randomly between any two of best fitted solutions by combining randomly selected parameters of the parent solutions.

For each offspring there are 64 different possible combinations of the two parents' genes.

The use of real variables as described above showed a quick saturation of population so the use of binary code was also implemented. For using binary code each parameter was set to have a 10^{-3} accuracy so ten bits were necessary for each parameter representation so a solution is composed by 60 bits which gives 2^{60} different possible solutions less 24^{6} (values that are unfeasible), makes $1{,}1\text{x}10^{18}$ different solutions. And the problem can then be considered a NP hard adding the fact that an evaluation of the objective function is highly time consuming.

For the particle swarm algorithm implementation, populations between 10 and 100 particles were tested. The V_{max} parameter was set at 0.5 (values from 0.2 to 0.8 were tested). The control parameters φ, were random $U(0, 0.2)$. The neighborhood topology considered was the interaction of individuals with the best performing individual in the population.

5. Results

To evaluate the forecasting performance, the Portuguese electricity demand, REN, from 3/01/05 to 31/03/07 was used for testing and evaluation. The first 107 weeks data were used to estimate the parameters and the remaining 9 weeks to evaluate post-sample accuracy of forecasts up to 24 hours ahead. Figures 1 and 2 show the performance of some of the algorithms in term of the paths taken towards the best solutions obtained. The algorithms were implemented in MATLAB.

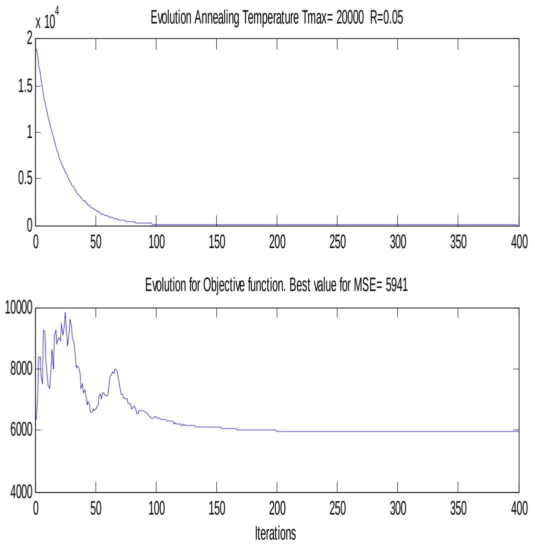

Fig. 1. Evolution of evaluation function for the simulated annealing algorithm.

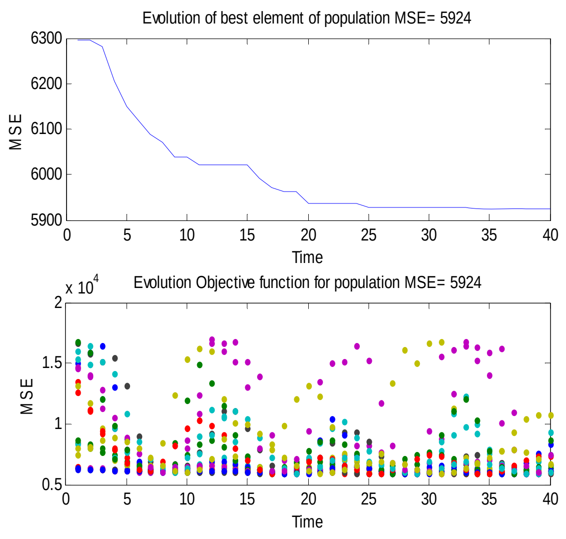

Fig. 2. Evolution of evaluation function for the PS algorithm

The values of the parameters obtained from each algorithm are summarized in Table I. The solutions do not differ much what may indicate that the problem could be convex. The best solutions (those with the smallest MSE) were then used for testing.

Table I. Parameters values obtained with each algorithm

	α	β	γ	δ	ϕ	λ	MSE
HC	0.33874	0.88411	0.48434	0.0003	0.41878	0.47991	5921
SA	0.35593	0.71944	0.47638	0.08282	0.37245	0.50514	5941
GA	0.36181	0.62915	0.49587	0.00013	0.36439	0.5250	5952
GA(bin)	0.3680	0.770	0.5150	0.0580	0.4080	0.5050	5946
PS	0.32119	1.0	0.46712	0.00130	0.44381	0.46747	5918
Solver	0.31204	1.0	0.44155	0.08463	0.45908	0.49953	5879

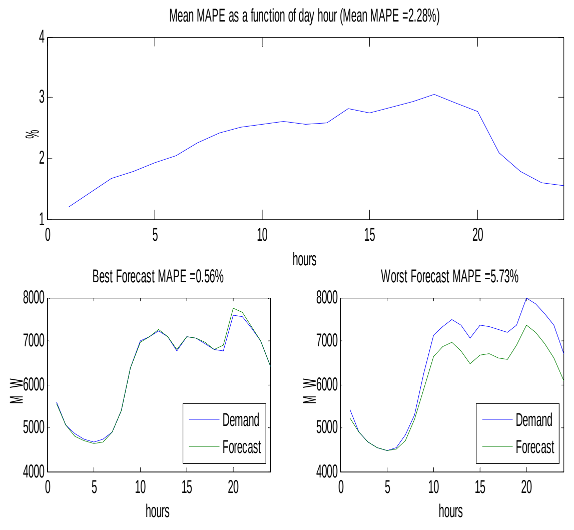

Fig. 3. Evaluation tests for the results obtained with hill climber algorithm for the day-ahead demand forecast.

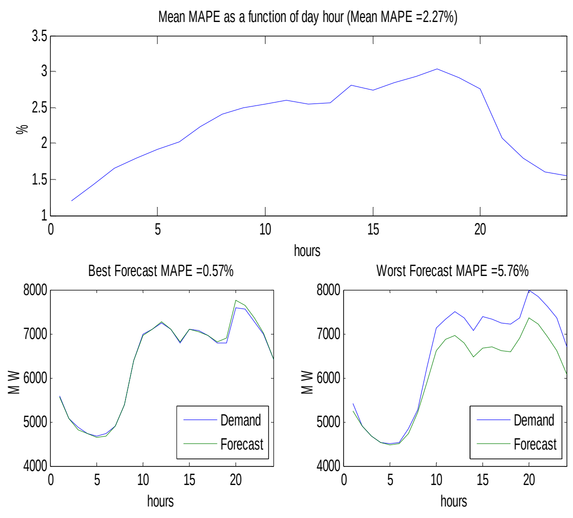

Fig. 4. Evaluation tests for the results obtained with hill climber algorithm for the day-ahead demand forecast.

The post sample forecasting performance was evaluated with the parameters obtained from the various algorithms. Figures 3 and 4 show the results obtained with the parameters optimized by the HC and PS. A comparison between the real and forecasted values is performed for the 70 days used for testing.
The best and worst days' forecast are showed in the graphics. A summary of each method performance is also showed by averaging the MAPE across the 24 hourly lead times. The resulting mean MAPE values are also presented in Figures 3 and 4.
With the parameters obtained with the PS algorithm the post sample forecasting performance was also tested for 48 hours-ahead and 168 hours-ahead forecasts. The results for the best and worst predictions were showed in Figures 5 and 6.
As we can see, the mean MAPE for the overall period increases a little (from 2.5% to 4 %) as its values are bigger (5% to 6%) as the lead time increases in the predictions of more than 48 hours ahead.

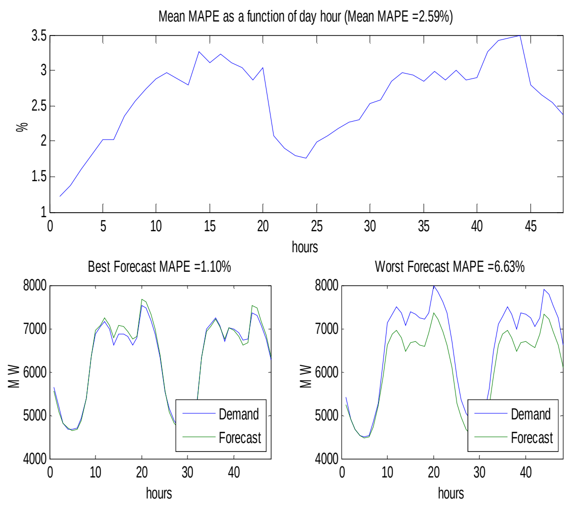

Fig. 5. Evaluation tests for the results obtained with PS for the two days-ahead demand forecast.

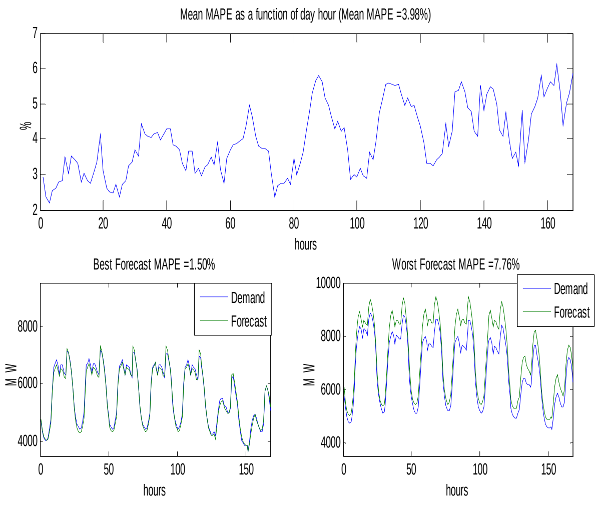

Fig. 6. Evaluation tests for the results obtained with PS for the week-ahead demand forecast.

6. Conclusions

On line short term electricity demand forecasting requires a robust procedure. The robustness of exponential smoothing methods and in special the HW with double seasonality prove to be a good candidate for short-term demand forecast.
The computing of the optimal parameters of the forecasting equation is an important issue for the accuracy of the forecast. Using an evolutionary approach to get the best parameters is a quick and simple method considering the enormous number of possible combinations for the parameters’ values.
The values obtained for the forecasting equation´s parameters using different metaheuristic algorithms were similar as well as the post sample forecasting performance which indicates that the objective function might be convex.
The HC algorithm, for its simplicity, is a good solution to adopt. The same parameters obtained could also be used to forecast up to 48 hours ahead or even a week if necessary. Although the errors increase when the forecasting period increases, when there are no weather data available this univariate method for short-term load forecast could also be a useful tool for weakly planning and scheduling of power systems supply.

Acknowledgement

We would like to thank to REN, Redes Energéticas Nacionais for the total hourly energy comsumption data.

References

[1] M.P. Garcia, and D.S. Kirschen, "Forecasting system imbalance volumes in competitive electricity markets," *IEEE Transactions on Power Systems*, vol. 21, pp. 240-248, 2006.

[2] D.W. Bunn, "Forecasting loads and prices in competitive power markets," Proceedings of the IEEE, vol. 88, pp.163-169, 2000.

[3] H.S. Hippert, C.E. Pedreira, R.C. Souza, “Neural Networks for Short-Term Load Forecasting: A Review andEvaluation,” *IEEE Trans. on Power Systems*, vol. 16, pp. 44-55, 2001

[4] Taylor, J.W., L.M. M. de Menezes, P. E. McSharry. 2006. A Comparison of Univariate Methods for Forecasting Electricity Demand Up to a Day Ahead. *International Journal of Forecasting*, 22,1-16.

[5] J. W. Taylor and P. E. McSharry, Short-Term Load Forecasting Methods: An Evaluation Based on European Data, *IEEE Transactions on Power Systems*, 22, 2213-2219, 2008.

[6] Taylor, J.W. 2003. Exponential Smoothing with a Damped Multiplicative Trend. *International Journal of Forecasting*, 19, 715-725.

[7] Philip G. Gould, Anne B. Koehler, J. Keith Ord, Ralfh D. Snyder, Rob J. Hyndman, Farshid Vahid-Araghi, “Forecasting Time Series with Multiple Seasonal Patterns”, 2005

[8] Taylor, J.W. 2003. Short-Term Electricity Demand Forecasting Using Double Seasonal Exponential Smoothing. Journal of Operational Research Society, 54, 799-805

[9] J.V. Segura, E. Vercher, A spreadsheet modeling approach to the Holt-Winters optimal forecasting”, Elsevier-European Journal of Operational Research, 2001

[10] Hillier, Lieberman, “Introduction to Operations Research” *McGraw-Hill Eighth Edition, 2005 pp618-653*

[11] M. Yagiura and T. Ibaraki, "On Metaheuristic Algorithms for Combinatorial Optimization Problems," *Systems and Computers in Japan, Vol. 32, nº3, March 2001 pp. 33-55.*

[12] C. Blum and A. Roli, "Metaheuristics in Combinatorial Optimization : Overview and Conceptual Comparison", ACM Computing Surveys, Vol 35, nº 3, September 2003, pp 268-308

[13] James Kennedy, Russell Eberhart, “Swarm Inteligence”, The Moran Kaufmann Series in Evolutionary Computation, 2001, pp 287-314.