



Optimizing large scale problems in a reduced space mapped by autoencoders

Vladimiro Miranda, Fellow IEEE, Vera Palma and Joana da Hora Martins

Abstract – This paper explores a technique to solve large scale optimization problems by reducing the search space dimensionality with the application of autoencoders. The technique applies autoencoders as a reversible mapping between the original problem dimension and a reduced space, which allows an evolutionary metaheuristic to evolve in a reduced space, having its objective function assessed in the original space. The technique is illustrated with an application of an EPSO (Evolutionary Particle Swarm Optimization) algorithm to a Hydro-Wind coordination problem and four benchmark optimization functions. The results obtained suggest that the new technique allows an improvement in the quality of solutions attained.

Index Terms — Hydro-Wind coordination problem, optimization, large scale, neural networks, autoencoders, evolutionary algorithms, metaheuristics.

I. INTRODUCTION

SOLVING problems in high dimensional spaces is both demanding in computing resources and difficult on convergence into satisfactory solutions. One of the major associated problems relates to the curse of dimensionality [1]. These drawbacks usually lead to early termination of runs, inducing the performance of techniques employed to be lower than desirable.

Several approaches have been suggested in literature to address these problems: the application of genetic algorithms [2], evolutionary algorithms [3], cooperative coevolution [4], among others.

This work addresses large scale problems by considering a reduction on the dimensionality of their search spaces. The problem of dimensionality reduction has been addressed with applications in clustering and in image processing. One important technique is Principal Component Analysis (PCA) [5], which is a mathematical procedure that projects the data into a linear subspace: data is multiplied with the eigenvectors from the sample covariance matrix, from where each point is represented by its coordinates along the directions of greatest

variance in the data set.

One research topic that has not been widely explored in literature, and that is explored in this work, is the combination of dimensionality reduction techniques as a general optimization tool for large scale problems. Noting that, to address such topic, it is necessary to transfer into a reduced space not only the data but also the constraints and the objective function of the problem.

The technique developed was firstly presented in [6], and is here designated as LASCA (Large Scale with Autoencoders). The main idea of LASCA is to make an evolutionary metaheuristic to evolve in a reduced dimension space \mathcal{S}' , controlling its progression in the original space \mathcal{S} . The transition between \mathcal{S} and \mathcal{S}' is made with recourse to an autoencoder, applied as a reversible mapping between the two spaces (autoencoders give mappings in both directions between \mathcal{S} and \mathcal{S}'). This way, the evolutionary metaheuristic can evolve in \mathcal{S}' , with its objective function assessed in \mathcal{S} .

The LASCA technique is illustrated with an application of an EPSO (Evolutionary Particle Swarm Optimization) algorithm to 5 case studies: four mathematic optimization functions suggested in [7] and one power system problem concerning the Hydro-Wind coordination problem.

The results obtained showed that the LASCA approach lead to the achievement of better quality solutions for some of the addressed case studies, and these are Alpine, Shifted Sphere, Shifted Rastrigin functions and the Hydro-Wind coordination problem. The tests conducted with the function Griewank did not return a significant gain nor in computational effort or in the quality of the solution achieved, since the convergence obtained with EPSO is already sufficient to achieve the optimum within few iterations.

The paper is structured as follows. Section 2 includes theoretical insights on autoassociative neural networks. Section 3 provides a brief description on the EPSO metaheuristic. Section 4 explains the LASCA technique. Section 5 provides the comparative analysis on the application of LASCA and EPSO techniques to four mathematical optimization functions. Section 6 details the addressed Hydro-Wind coordination problem, including the presentation and discussion of the results obtained with LASCA and EPSO. Section 7 provides the conclusions of the work.

II. AUTOENCODERS

Autoencoders are feedforward neural networks trained to reproduce the input space \mathcal{S} in the output. The adequacy of

This work is partially funded by the ERDF from the EU through the Programme COMPETE and by the Portuguese Government through FCT - Foundation for Science and Technology, project ref. LASCA PTDC/EEA-EEL/104278/2008..

J. Hora (joana.hora@gmail.com), V. Palma (vera.p.ferreira@inescporto.pt) and V. Miranda (vmiranda@inescporto.pt) are with INESC TEC - INESC Technology and Science, coordinated by INESC Porto, Portugal. V. Miranda is also with FEUP, Faculty of Engineering of the University of Porto, Portugal.

autoencoders to reduce dimensionality of data is widely known [8, 9]. This property allows the definition of functions that establish a one-to-one mapping between points in \mathcal{S} of dimension m and a space \mathcal{S}' of dimension n (with $n < m$ without loss of generality). Fig. 1 presents a schematic visualization of an autoencoder, where the bottleneck layer encodes instances from \mathcal{S} into a smaller dimension space \mathcal{S}' by f , and where the reverse process from \mathcal{S}' to \mathcal{S} is allowed by f^{-1} . The quality of this encoding and decoding process depends on the quality of the training.

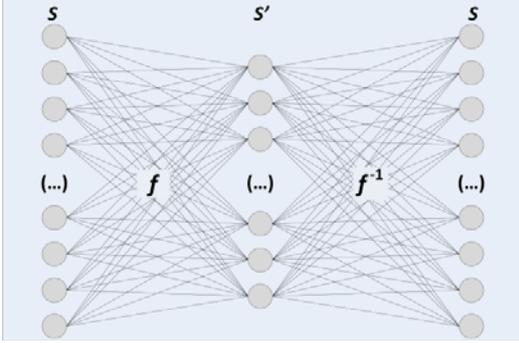


Fig. 1. Schematic of an autoencoder. An instance in space \mathcal{S} is encoded into a reduced space \mathcal{S}' by f , and is further expanded into \mathcal{S} by f^{-1} .

Autoencoders have been applied to perform signal analysis [10, 11], or reconstruct missing sensor signals [12], a problem of “missing data” where autoencoders have been used to reconstruct some missing input data in such a way that the reconstruction appears reproduced in the output, through minimizing a function of the input-output error.

Other applications of autoencoders include the representation of images within a reduced space [13, 14], so that this representation would be subject to distinct processing techniques such as identification and pattern recognition. For instance, face images could be identified and clustered according to sex and distinguished from non-faces [15].

Fig. 1 presents a simple topology of autoencoder with three layers, although any number of layers may be adopted. There is no mandatory condition that both halves of the autoencoder should be symmetric, nevertheless published work has consistently adopted such architecture. It has been found that optimizing the weights of autoencoders with non-linear activation functions became an increasingly heavy and difficult task with the increase in the number of layers in the neural network; therefore, new schemes to achieve a more efficient training are being proposed [16].

And, finally, there is no a priori indication on the optimum reduction rate (the ratio between the no. neurons in the smallest middle layer and the no. neurons in the inputs/output layers) to adopt. This decision has been made, as far as one may perceive from the literature available, dictated by trial and error and by the characteristics of the problem, having in mind that the greatest effective dimension reduction possible is almost always desirable.

III. EPSO

The optimization method used in the tests described in this paper was EPSO, for Evolutionary Particle Swarm Optimization. EPSO is a hybrid in concepts of EA and PSO [17], first proposed in [18] and with an improved version in [XX?]. It is an Evolutionary Algorithm with an adaptive recombination operator inspired in the “movement rule” of PSO (Particle Swarm Optimization). This rule generates a new individual as a weighted combination of parents, which are: a given individual, its best ancestor and the best ancestor of the present generation. This may be seen as a form of intermediary recombination. In this operator, a new individual is formed from a weighted mix of ancestors, and this weighted mix may vary in each space dimension. The mutation operator is only applied to the weights, therefore forming a self-adaptive recombination operator.

An EPSO iteration starts with a swarm of p particles. Each particle originates $r + 1$ descendants, from whose only one will survive. This process flow simultaneously over five main steps, which are schematized in Fig. 2 and detailed next.

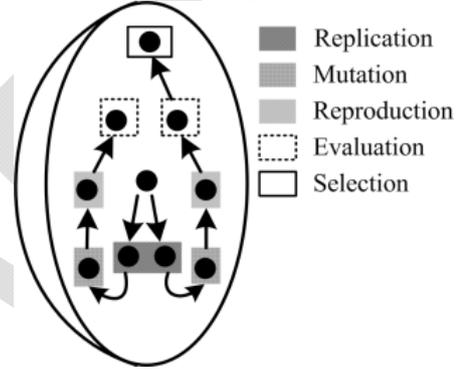


Fig. 2. The five steps elapsing over an EPSO iteration, with $r = 1$.

In **Replication** each particle p_i is replicated r times, originating $k = r + 1$ replications. This way all k replications are similar to p_i . The k^{th} replica of p_i is designated as p_{ik} .

The **Mutation** step is performed by changing the weights of Inertia (w_i), Memory (w_m) and Cooperation (w_c) associated to each replica p_{ik} . The mutation for w_m is presented in (1), where τ is a learning parameter that can be fixed or be subject to mutation. Similar rules are considered for the w_i and w_c .

$$wm_{ik}^{t+1} = wm_{ik}^t + \tau \cdot N(0,1) \quad (1)$$

In **Reproduction** each particle previously replicated and mutated p_{ik} originates a new descendant. This process starts with a mutation on the best position achieved by swarm b_G^t , as specified in (2), which induces agitation into the swarm, even when a convergence to the same space region tends to occur.

$$b_G^{t+1} = b_G^t + \tau' \cdot N(0,1) \quad (2)$$

Next, the velocity for each replica is calculated following the movement rule presented in (3).

$$V_{ik}^{t+1} = w_i^{t+1} \cdot V_{ik}^t + w_m^{t+1} \cdot (b_i - X_{ik}^t) + w_c^{t+1} \cdot (b_G^{t+1} - X_{ik}^t) \quad (3)$$

Finally, the new position of each replica is found by adding the previous position with the new velocity, as specified in (4).

$$X_{ik}^{t+1} = X_{ik}^t + V_{ik}^{t+1} \quad (4)$$

In **Evaluation** each descendant is evaluated, using the objective function.

The last step is the **Selection**. Considering the evaluation results obtained in the previous step, the best particle in each group of descendants (each group has k particles) is identified and attributed with a very high probability $(1 - \alpha)$, being α a small number that simulates the “luck” of a worst particle to be selected as well.

Next a stochastic tournament is performed, considering the probabilities of each particle in each group, from where a particle is chosen to integrate the next swarm generation.

IV. LASCA APPROACH

It is known that, in general, population based (PB) methods, such as Evolutionary Algorithms (EA) like Evolutionary Programming (EP) or Genetic algorithms (GA), or as Particle Swarm Optimization (PSO) or Ant Colony Methods (ACM), used in optimization, exhibit a growingly slow and inaccurate performance with the increase in dimension of the search space where individuals (chromosomes, particles) are defined. This limits the practical application of parallel processing methods in large scale programming problems. However, these problems appear as extremely relevant in a diversity of engineering areas; for instance, in Power Systems it is not at all uncommon to find planning or operation problems with hundreds to tens of thousands of variables.

The original idea reported in this paper is, therefore, to use autoencoder properties to reduce the dimension of the search space, while keeping the solution evaluation accurate so that selection or similar operators may still act and drive the process towards an optimum. This idea can be summarized in the following sequential parts.

Part A

A PB algorithm with individuals (particles) is applied in \mathcal{S} . The solutions obtained over a specified number of iterations are stored. This storage only accepts different particles. Two particles are considered equal if all homologous values are equal for all positions. Moreover, different particles can assume equal fitness values.

The stored solutions are used as a dataset to train an autoencoder. The autoencoder will allow the encoding and decoding of particles between \mathcal{S} and \mathcal{S}' .

Part B

The last swarm obtained in part A, that was evolving in \mathcal{S} , will be transferred to \mathcal{S}' . The information transferred includes the particles and the corresponding best positions, velocities and weights of inertia, cooperation, memory and perturbation.

The transference of particles and corresponding best

positions is made by applying the encoding function f corresponding to the 1st half of the autoencoder.

The transference of velocities is made considering expression (5), which is easily deductible from expression (4). This means that, for each particle, the velocity is calculated in \mathcal{S}' as the difference between the corresponding compressed positions in t and in $t - 1$. This procedure is adopted since the autoencoder was trained to represent particles and not velocities.

$$V_{ik}^t = X_{ik}^t - X_{ik}^{t-1} \quad (5)$$

The transference of weights of inertia, memory and cooperation is made directly.

Once the swarm is completely transferred, it starts to evolve in \mathcal{S}' . The evaluation of the fitness function would not be possible in \mathcal{S}' , since the values that particles assume in the reduced space have no physic meaning. Therefore, when the assessment is necessary, the particles are first decoded into \mathcal{S} , and only then evaluated, as represented in Fig. 3. This procedure ensures that the evolution observed in \mathcal{S}' actually corresponds to an improvement on the problem addressed.

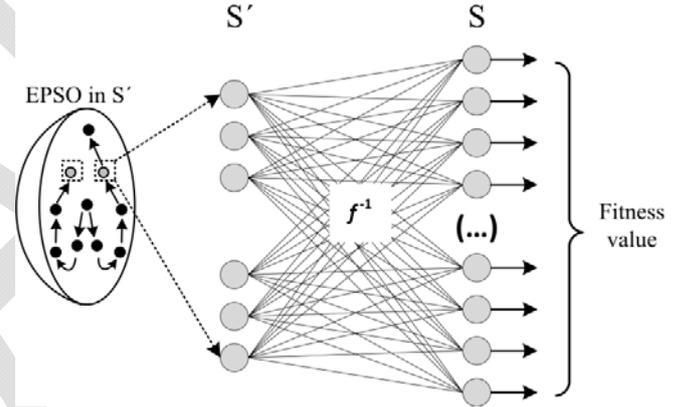


Fig. 3. Particles evolve in \mathcal{S}' but the fitness evaluation is made in \mathcal{S} using the decoding function f^{-1} on the 2nd half of the autoencoder.

Part C

Because the autoencoder is just an approximation and not a representation of the exact mappings $\mathcal{S} \leftrightarrow \mathcal{S}'$, some information will eventually be lost. It is possible then that the exact optimum of the original problem may not be found in \mathcal{S}' – but if the approximation is good enough, a near optimal solution or, at least, the location of the optimum will be found. Therefore, the launching of an efficient post-optimization search, back in \mathcal{S} , is implemented.

This is made by transferring the last swarm obtained in Part B back in \mathcal{S} . This transference includes the same elements and the same procedures described before, now using f^{-1} corresponding to the 2nd half of the autoencoder.

A few additional comments must be made. First, it is true that the half-networks emulating f and f^{-1} only generate approximations to these functions. However, since each point in \mathcal{S}' is associated with a real solution in \mathcal{S} , it is valued exactly (see Fig. 3).

One must realize that the meaning of the variables in space

S' (the output of the neurons in the middle layer) is virtually unknown, but constraints associated with variable limits must be enforced in this space. The strategy adopted has been to observe in the autoencoder training set the values assumed by the variables in S' . From this observation, limits are defined for these variables taking in account the minimum and maximum values registered in the training set.

Notice that the training and test sets used to generate the autoencoder neural network are not obtained through random sampling. In fact, because the sampling is conducted using an evolutionary optimizing method, it becomes very likely that one will have a denser representation of the solution space in regions close to the optimum, which is a very desirable trait.

Many questions remain open, revolving around how and when to apply a LASCA approach and expect a net gain. A partial answer to these points is given in the following sections with the application of LASCA to different case studies, taking EPSO as PB method.

V. BENCHMARK OPTIMIZATION FUNCTIONS

This section includes the results obtained with the application of LASCA approach to four benchmark optimization functions. For each experiment, the comparison with EPSO running solely in S is provided, concerning the equivalent number of iterations used with LASCA.

A. Alpine Function

Alpine is a function to be maximized, and is defined in (6). This function has many local optima, but just one maximum. Considering the search domain $[0,10]$, the maximum is achieved at point $x^* = (7.917, \dots, 7.917)$, corresponding to a maximal value of 2.808^D [19].

$$f(x_1, \dots, x_D) = \prod_{i=1}^D \sin(x_i) \sqrt{\prod_{i=1}^D x_i} \quad (6)$$

Each swarm included 400 particles. The parameters found to best perform with EPSO in S are $\tau_S = 0.4$ and $cp_S = 0.1$. For LASCA, the parameters found to best perform with EPSO at S' were $\tau_{S'} = 0.4$ and $cp_{S'} = 0.95$. The autoencoder was trained with classic backpropagation (PROP), using tangent hyperbolic activation functions at hidden layer and linear at the output layer. The number of iterations used in parts A, B and C were 100, 100 and 100, respectively. Results are provided in Fig. 4, with Parts A, B and C separated by vertical lines.

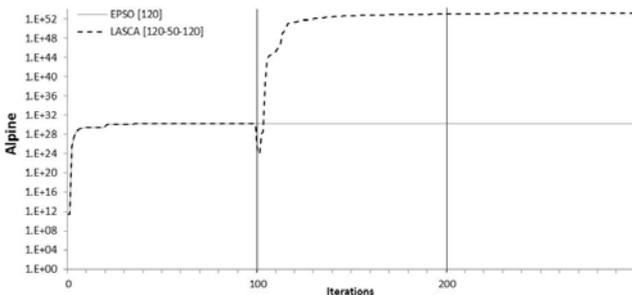


Fig. 4. Results obtained with LASCA[120-50-120] and EPSO[120] for the Alpine function. Average of 10 runs.

The EPSO stabilizes from iteration 80 onwards, with low improvement concerning the fitness evolution. The LASCA includes similar results than EPSO concerning the iterations of Part A, which was expected since the starting point considered for both approaches is the same (meaning the same random seeds were always considered for each pair of compared experiments). Once LASCA enters Part B, a jump occurs impelling the best solution found to values of significantly higher magnitude. The final fitness value achieved with LASCA is significantly higher ($1.51E+53$) than the one obtained by EPSO ($1.66E+30$), for an equivalent number of iterations, knowing that for this dimensionality the maximum value would be of $2.808^{120} = 6.42E + 53$. The transition into a reduced space (part B) allowed the swarm to find a better search zone, and the return to the original space (part C) allowed a further improvement.

This experiment was replicated for spaces with other dimensionalities, including [200-70-200] and [300-150-500], among others. The results obtained with all experiments led to similar conclusions.

B. Shifted Sphere function

The shifted sphere is a function to be minimized, and is defined in (7), with $f_{bias} = -450$. Exploring a domain $[-100,100]$, the minimum value of $f(x^*) = -450$ is achieved at point $x^* = (0, \dots, 0)$ [7].

$$f(x_1, \dots, x_D) = \sum_{i=1}^D x_i^2 + f_{bias} \quad (7)$$

The experiment considered a swarm of 400 particles, for a S dimension of 120, and S' dimension of 50. The parameters found to best perform with EPSO in S were $\tau_S = 0.7$ and $cp_S = 0.9$. Concerning the LASCA approach, the number of iterations used in parts A, B and C were 100, 100 and 100, respectively. The parameters found to best perform with EPSO in S' were $\tau_{S'} = 0.8$ and $cp_{S'} = 0.5$. The autoencoder applied was trained with an Information Theoretical Learning (ITL) criterion, the maximization of Quadratic Mutual Information (QMI) transferred from the input to the hidden layer. The Cauchy-Schwartz (CS) estimator of the QMI was considered, see [20]. Tangent hyperbolic activation functions were considered at both hidden and output layers. The results obtained are detailed in Fig. 5, with Parts A, B and C separated by vertical lines.

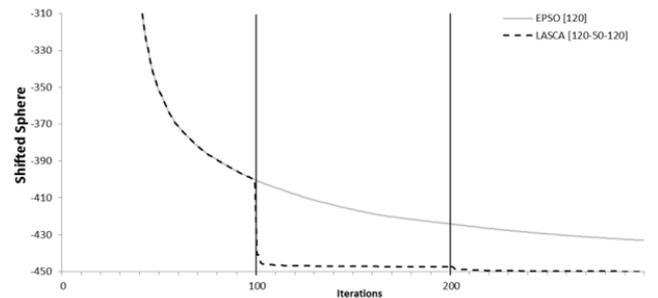


Fig. 5. Results obtained with LASCA[120-50-120] and EPSO[120] for the shifted sphere function. Average of 10 runs.

The EPSO algorithm shows a deeper evolution for the first 150 iterations, from where a stabilization trend is observed. The final fitness value obtained was -433.05. Regarding the LASCA approach, a jump on the fitness value is observed after the transition from \mathcal{S} to \mathcal{S}' . The remainder iterations on Parts B and C evolve steadily, leading to a final fitness value of -449.94, which is a considerable better solution than the one achieved with EPSO for the same number of iterations.

Other experiments were made with this function, assuming the similar parameters and higher dimensionality, including [200-100-200] and [300-100-300]. All experiments returned similar conclusions.

C. Shifted Rastrigin function

Shifted Rastrigin is a function to be minimized. Its definition is provided in (8), where $f_{bias} = -330$. Considering a domain of $[-5.12, 5.12]$, the minimum value $f(x^*) = -330$ is reached at point $x^* = (0, \dots, 0)$ [7]. This function is particularly difficult to solve due to the high number of maxima and minima.

$$f(x_1, \dots, x_D) = A \cdot n + \sum_{i=1}^D [x_i^2 - A \cdot \cos(2\pi x_i)] + f_{bias} \quad (8)$$

The experiment considered swarms of 400 particles. The best parameters found for EPSO in \mathcal{S} were $\tau_S = 0.9$ and $cp_S = 0.9$. Concerning the LASCA approach, the number of iterations employed in parts A, B and C were 100, 30, 100, respectively, and the calibrated parameters for EPSO in \mathcal{S}' are $\tau_{S'} = 0.8$ and $cp_{S'} = 0.4$. The autoencoder was trained with PROP, considering tangent hyperbolic and linear activation functions for the hidden and output layers, respectively. The results obtained are illustrated in Fig. 6, with Parts A, B and C separated by vertical lines.

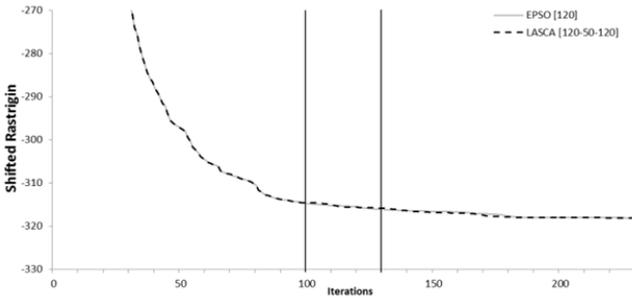


Fig. 6. Results obtained with LASCA[120-50-120] and EPSO[120] for the Shifted Rastrigin function. Average of 10 runs.

The EPSO technique shows a more pronounced development during the first 90 iterations, from where a stabilization trend is observed. The final fitness value obtained with EPSO was -318.08.

The evolution of LASCA in Part B did not provide a significant gain concerning the quality of solution achieved. However, after the swarm transference into \mathcal{S} (Part C), a particularly propitious progress was observed. The final fitness value obtained with LASCA was -318.09. This indicates that both approaches return equivalent quality on the solutions obtained.

Other experiments with higher dimensions were performed. For example, with [300-150-300] and a swarm of 200 particles, keeping the remainder specifications, EPSO achieved a final fitness of -318.26 and LASCA -321.16. Other experiments varying the dimensionality and the iteration number where the transitions occur always returned either equivalent final solutions or solutions slightly better with LASCA approach.

This suggests that LASCA is particularly more efficient than EPSO for cases with higher dimensionalities of \mathcal{S} .

This indicates that the evolution within a smaller search space pushed the swarm to a better positioning.

D. Griewank function

The Griewank function is defined in (9), and is to be minimized. This function has several local minima, with the global minimum $f(x^*) = 0$ achieved at $x^* = (0, \dots, 0)$. Tests were conducted considering the domain $[-30, 30]^{120}$.

$$f(x_1, \dots, x_D) = 1 + \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) \quad (9)$$

The swarm applied was composed of 400 particles. The best parameters found for EPSO are were $\tau_S = 0.6$ and $cp_S = 0.95$. Concerning the LASCA approach, the number of iterations employed in parts A, B and C were 15, 50, 100, respectively, and the calibrated parameters for EPSO in \mathcal{S}' are $\tau_{S'} = 0.7$ and $cp_{S'} = 0.5$. The autoencoder was trained with considering the maximization of QMI with CS. The activation functions considered were tangent hyperbolic and linear, for the hidden and output layers, respectively. The results obtained are illustrated in Fig. 7, with Parts A, B and C separated by vertical lines.

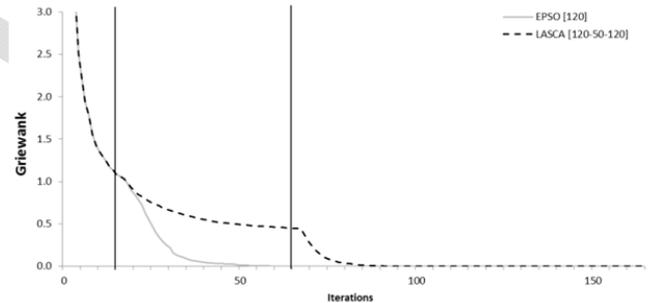


Fig. 7. Results obtained with LASCA[120-50-120] and EPSO[120] for the Griewank function. Average of 10 runs.

EPSO had a particularly fast convergence to the global optimum. This is the reason why the number of iteration considered for Part A was so low. The low number of iterations needed to achieve the global optimum difficult the storage of a sufficient number of different particles to accurately train an autoencoder. This drawback can influence the accuracy of the autoencoder on correctly mapping the manifold between the two spaces. The final fitness value with EPSO was 5.55E-17.

Concerning LASCA, the convergence observed was slower

than EPSO both in Parts B and C. Also, the final fitness value of $3.56E-12$ is worse than the one obtained with EPSO.

VI. HYDRO-WIND COORDINATION PROBLEM

A. General description of the problem

An experimental confirmation of the potential of the LASCA approach is given in this section with a Hydro-Wind coordination problem.

The Hydro-Wind coordination problem aims at maximizing the joint profit of a power system composed by several hydro and wind farms. The maximization is made by changing the water volumes to be pumped and released, given a set of specifications defining each scenario. These specifications include the wind forecast and the water inflow to the system. Due to the high complexity this problem can achieve, the operation planning is normally made with multiple approaches for different horizons of analysis: from the short term (days) to the long term (years).

This problem has obvious similarities with hydro-thermal coordination in the presence of pumping storage facilities and is represented by a complex time dependent formulation if cascading river dams are present. The hydro-thermal coordination is in itself a large scale dynamic difficult problem. Several techniques have been used to deal with it such as Lagrangian relaxation [21], Stochastic Dynamic Programming [22], Dual Dynamic Programming [23] or Genetic Algorithms and Evolutionary Programming [24]. Models for wind-hydro coordination have also been proposed [25, 26, 27]. An insightful review concerning applications and methods for this problem is provided in [28].

The problem formulated in this work is similar to the one proposed in [29], which considers the optimal operation in a deterministic context, meaning that future inflows (of water, of wind energy) are considered as an assumption – although these values are now actually known, the objective is to analyze the system response to a specific situation (a posteriori analysis).

This problem is composed of an independent energy producer that owns a number of cascading hydro power plants, and also wind power plants that are treated as a single source (energy supplied through the transmission grid).

A medium term operation planning is considered. Also, there is differentiation concerning peak and off-peak periods: the power demand suffers high variations concerning day and night periods. Other energy sources, such as nuclear and fossil fuel plants, are inefficient in generating power for short periods of increased demand. On the other hand, hydroelectric generators can be started and stopped almost instantly, making the energy produced in hydro farms timely responsive to peak demands. Water can be stored in reservoirs during off-peak periods, and used to produce energy during the peak periods.

Wind energy value is greatly enhanced if combined with pumped storage so that energy may be delivered to the market during hours of high price but the decision to store must be weighed against the price of selling directly at the moment it is produced in the wind parks.

In this paper, we will apply an EPSO (Evolutionary Particle

Swarm Optimization) algorithm [18] to test problems emulating the wind-hydro coordination context, built with enough complexity to test the optimization techniques under judgment.

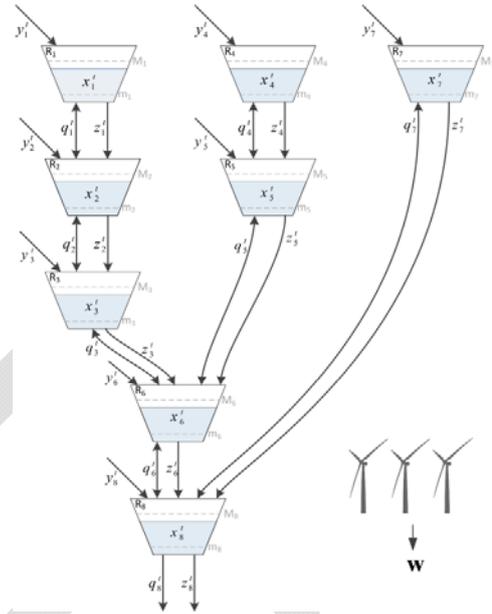


Fig. 8. Scheme for the Hydro-Wind power system considered.

A medium term operation planning or the water resources requires an evaluation of the operation for a period of the order of magnitude of 1 year and estimates of water and wind availability, with the division of the planning period in sub-periods corresponding to different months and different load levels with different estimated energy costs. The dimension of the problem may be very large.

Because the purpose of this paper is to demonstrate the potential and usefulness of the new technique, one will not devote much time to describe the subtleties of the real world problem or analyze the effects of uncertainties and will concentrate on the optimization procedure instead.

The considered system integrates $N = 8$ cascading reservoirs, as detailed in Fig. 8, built from [24]. All reservoirs are admitted to be equipped with pumps allowing a certain amount of water to be moved upstream if convenient, except the 8th reservoir.

The objective is to derive an operation plan that maximizes the profit obtained with the operation of the system throughout T time periods with different buying and selling energy prices. The operation plan will determine:

- Quantity of water to be released or pumped for each hydro power plant in each period of time and energy sold or used;
- Quantity of wind energy to be used for water pumping and the quantity of wind energy to be sold to the electric power system in each period.
- Detailed information about the amount of water storage in each reservoir and water storage capacity available for each period of time

The T time periods are divided in $T/2$ peak periods and $T/2$ off-peak periods. A horizon of 6 months ($T = 12$) is

considered. Six energy prices are defined for each period, also admitting average price forecasting based on market history:

- Hydroelectric energy selling price at peak and off-peak periods;
- Hydroelectric pumping price at peak and off-peak periods;
- Wind energy selling price at peak and off-peak periods.

The variables of this problem are defined in terms of water movement for each reservoir in each period. Ecological spills or evaporation are not considered in this example but present no difficulty to the model.

B. The mathematical model

The electric energy of hydro origin generated in moment t by reservoir n is described by equation (10),

$$H_{n,j}^t = K_{n,j}[h_n(x_n^t) - h_n(q_n^t + z_n^t)] \cdot |q_n^t| \quad (10)$$

where:

$H_{n,j}^t$ - the energy generated by the n^{th} reservoir in period t if $j = turbine$, or the energy consumed if $j = pump$ (in Wh);

N - the number of hydro power plants included in the system;

$K_{n,j}$ - a specified constant for each reservoir, which considers the gravitational acceleration (g), the efficiency of the turbine (η) and the water density (ρ): $K_{n,j} = \eta \cdot \rho \cdot g$. This constant takes different values for pumping ($j = pump$) and generation ($j = turbine$) modes, concerning the different efficiencies involved;

x_n^t - the volume stored in the n^{th} reservoir at the beginning of the period t (in m^3);

q_n^t - volume of water transferred between the n^{th} and the immediately downstream reservoirs, at moment t : assumes negative values for volumes pumped into the reservoir, and positive values for volumes released out of the reservoir (in m^3);

z_n^t - the volume of water spilled during the period t (in m^3);

$h_n(\cdot)$ - function returning the estimation of the water head (height) given a water volume, for the n^{th} reservoir (in m).

The available water volume for each reservoir is calculated for each period considering the variables associated to the reservoir, such as the natural affluences, the volume of water pumped or used in generation, the volume of water spilled and finally the already existing water volume, all of them in the previous period of time, and also considering the variables associated to the operation of the upstream reservoirs such as the quantity of water that was used for generation and now has to be accommodated in the downstream reservoirs and also the water volume spilled from the upstream reservoirs. So in the Hydro-Wind coordination model, the procedure above is mathematically represented as dictated in equation (11).

$$x_n^{t+1} = x_n^t + y_n^t + \sum_{k \in \Omega} [q_k^t + z_k^t] - q_n^t - z_n^t \quad (11)$$

where:

y_n^t - volume of water entering the n^{th} reservoir concerning natural river inflow (in m^3);

Ω - set of hydro reservoirs immediately upstream of the n^{th} reservoir;

Under this case study, the EPSO algorithm is applied to optimize a particle \mathbf{q} , which includes the volumes q_n^t for each reservoir under each temporal moment, as specified in (12), where N refers to the total number of reservoirs, and T to the total number of temporal moments.

$$\mathbf{q} = [q_1^1, q_2^1, \dots, q_N^1, \dots, q_1^T, q_2^T, \dots, q_N^T] \quad (12)$$

Constraints ensuring reservoirs' capacities

Assuming that the system always starts with water volumes x_n^t respecting the corresponding reservoir minimum and maximum capacity limits m_n and M_n , respectively, the model must ensure these limits are satisfied in further temporal moments. Therefore, the volume of the n^{th} reservoir at moment $t + 1$ must respect:

$$m_n < x_n^{t+1} < M_n \quad (13)$$

Combining equation (11) and (13), one can obtain the dynamic constraints specified in (14) and (15), which are applied to each position of \mathbf{q} .

$$x_n^t + y_n^t + \sum_{k \in \Omega} [q_k^t + z_k^t] - z_n^t - M_n < q_n^t \quad (14)$$

$$x_n^t + y_n^t + \sum_{k \in \Omega} [q_k^t + z_k^t] - z_n^t - m_n > q_n^t \quad (15)$$

Constraints ensuring turbines' capacities

For each reservoir, the specifications on the turbines installed were considered, which allowed the estimation of maximum and minimum volumes they are able to release or pump. These constraints are considered in the model as represented in (16).

$$q_n^{min} < q_n^t < q_n^{max} \quad (16)$$

Constraints ensuring the available water to pump

When the decision to pump water is made, the maximum value of volume to pump must also be restricted to the available volume in the immediately downstream reservoir (IDR). This constraint is only meaningful to the pumping case since when releasing water, even if the IDR exceeds its maximum capacity, it would spill out the overflow. Accordingly, the maximum volume of water to pump into the n^{th} reservoir, γ_n^t , is defined in (17), and constrains q_n^t as defined in (18).

$$\gamma_n^t = x_{IDR}^t - m_{IDR} \quad (17)$$

$$\gamma_n^t < q_n^t \quad (18)$$

The energy generated at wind farms per period is estimated

as w^t by an external forecasting procedure – and taken as data in this example of coordination planning. Its value per period is derived from the wind series and each wind farm production characteristic, which can be modeled separately from the optimization procedure. In fact, as there are no “reservoirs for wind”, the generation forecast is a direct function of the wind forecast. An auxiliary vector w is considered, where each element w_n^t refers to the available wind energy for the n^{th} reservoir at moment t , and this vector is updated as further described.

Value of energy produced by water released

When q_n^t takes a positive value, meaning the decision of releasing water was made, the corresponding energy is calculated following equation (10). The value associated to this energy is further calculated by considering the corresponding price, depending if the period type is peak or off-peak, as detailed in (19).

$$V_{n,A}^t = \begin{cases} H_{n,turb}^t \cdot P_{A,1} & ,\text{if under peak period} \\ H_{n,turb}^t \cdot P_{A,0} & ,\text{if under off-peak period} \end{cases} \quad (19)$$

Value of energy consumed to pump water

When q_n^t takes a negative value, indicating the decision of pumping, the energy necessary to pump is calculated using equation (10).

Two possibilities may happen when the decision of pumping water is made: there is enough wind energy available w_n^t , to pump the water, or there is not.

When there is enough wind energy available (i.e. $w_n^t \geq H_{n,pump}^t$), w_n^t is used. In this case, the value of the wind energy spent to pump is calculated by (20).

$$V_{n,B}^t = \begin{cases} H_{n,pump}^t \cdot P_{B,1} & ,\text{if under peak period} \\ H_{n,pump}^t \cdot P_{B,0} & ,\text{if under off-peak period} \end{cases} \quad (20)$$

The second possibility is that there is not enough wind energy available (i.e. $w_n^t < H_{n,pump}^t$). In this scenario, the model spends all the available energy from wind farms and buys the remainder necessary energy from grid (i.e. $G_p = H_{n,pump}^t - w_n^t$). In this situation, the equation (20) is used to calculate the value of the energy consumed from wind farms, and equation (21) is considered to calculate the value of the energy bought from grid to pump.

$$V_{n,C}^t = \begin{cases} (H_{n,pump}^t - w_n^t) \cdot P_{C,1} & ,\text{if under peak period} \\ (H_{n,pump}^t - w_n^t) \cdot P_{C,0} & ,\text{if under off-peak period} \end{cases} \quad (21)$$

For both cases, the wind energy available is updated, to provide accurate energy assessment of all reservoirs under the same temporal moment.

Value of wind energy

When all reservoirs are assessed for a specified temporal moment, the model will calculate the monetary value of the available wind energy at that moment, if any is available, which is considered to be sold to the grid. This value is estimated as defined in (22).

$$V_{n,D}^t = \begin{cases} w_n^t \cdot P_{D,1} & ,\text{if under peak period} \\ w_n^t \cdot P_{D,0} & ,\text{if under off-peak period} \end{cases} \quad (22)$$

System's revenue

The revenue obtained with each reservoir is defined in (23)

$$R_n^t = \begin{cases} V_{n,A}^t + V_{n,B}^t - V_{n,C}^t - V_{n,D}^t & ,\text{if } n \neq N \\ V_{n,A}^t + V_{n,B}^t - V_{n,C}^t + V_{n,D}^t & ,\text{if } n = N \end{cases} \quad (23)$$

Finally, once all reservoirs are assessed for all temporal moments, the profit obtained with the entire system is calculated as defined in (24).

$$Profit = \sum_{t=1}^T \sum_{n=1}^N R_n^t \quad (24)$$

C. Results

The Hydro-Wind coordination problem was simulated with swarms of 50 particles. The parameters found to best perform with EPSO in S are $\tau_S = 0.9$ and $cp_S = 0.7$. For LASCA, the parameters found to best perform with EPSO at S' were $\tau_{S'} = 0.8$ and $cp_{S'} = 0.1$. The number of iterations used in parts A, B and C were 400, 400 and 200, respectively. The autoencoder was trained with PROP, applying tangent hyperbolic and linear activation functions for the hidden and output layers, respectively. Results are provided in Fig. 9, with Parts A, B and C separated by vertical lines.

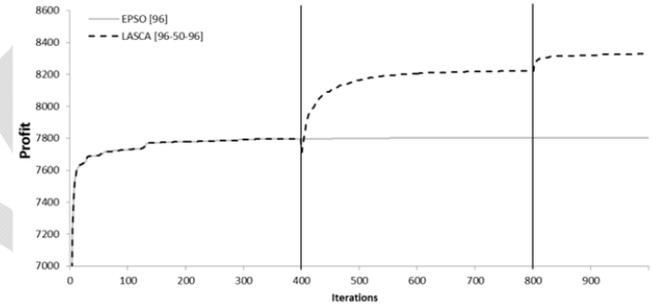


Fig. 9. Results obtained with LASCA[96-50-96] and EPSO[96] for the Hydro Wind coordination problem with 8 reservoirs. Average of 10 runs.

The results obtained show a stabilization of EPSO from iteration 150 (approximately) onwards. The final profit value obtained with EPSO (over 1000 iterations) is 7803.05 euros.

Concerning the LASCA approach, a similar profit evolution over Part A to the corresponding iterations with EPSO is observed, since these iterations are common for both approaches. Once LASCA enters Part B, a slightly deterioration of profit occurs, relating to the loss of information incurred with the transition of particles and associated velocities and weights to the reduced space. The evolution observed in subsequent iterations includes a progressive increase on the profit value till iteration 600, approximately, from where a stabilization would be observed till the end of Part B. The transition to Part C allows a slightly improvement of the profit, with a final profit value for the LASCA approach of 8328.3 euros, representing a significant gain (of 525.25 euros) to the solution obtained with EPSO.

VII. CONCLUSIONS

Metaheuristics or population-based methods are known to lose efficiency in large scale problems: the convergence becomes slow when the number of variables is large and the computing effort to reach the optimum becomes heavy.

One of such problems is the hydro-wind coordination in medium term operation planning, where several elements of complexity are present, namely the spatial and temporal dependence introduced by the cascading hydro power stations and the need to represent a large set of time steps. The practical interest is to evaluate the added value of a hydro-wind coordination strategy when compared with an independent operation of hydro and wind generation systems.

Hydro-wind coordination leads to the concept of virtual power station and a market agent (or a partnership of agents) operating it may extract added value from the renewable resources. This analysis must necessarily be probabilistic, given the uncertainties associated to the renewable energies availability and also to the energy prices, and requires a considerable number of simulations of a diversity of scenarios. Therefore, any technique that provides higher quality solutions becomes extremely valuable.

This paper presents, through the test of benchmark optimization functions and a practical example of hydro-wind coordination problem, a novel method to approach the solutions of large scale problems with population-based metaheuristics by organizing meta-heuristic searches in an equivalent reduced dimension search space.

The approach was coined as the Large Scale Computing with Autoencoders approach, or LASCA. The cleverness of the method lies in the fact that the evolutionary process acts upon individuals represented by chromosomes that are not designed ad-hoc by a human but instead result from an intelligent coding achieved by a first half of an autoencoder neural network, while the fitness function is evaluated by decoding the intelligent chromosomes with the second half of the autoencoder. Because the clever chromosomes are represented in a space of reduced dimension, the optimization process is able to move towards different search zones, this way finding better quality solutions.

This novel technique was applied to benchmark optimization functions and to the hydro-wind coordination problem. The implementation was made with an Evolutionary Particle Swarm Optimization algorithm, but there is no loss of generality because any other population-based method could have been used. The results presented fully demonstrate the interest of the technique, which is of general application. Questions remain unanswered, namely the generality of the application of the method and the characteristics of a problem that make it a good candidate to be dealt with the technique, with computing effort and solution quality gains. However, the results reported in the paper seem interesting enough to motivate research investment in the approach.

VIII. REFERENCES

- [1] T. Bengtsson, P. Bickel and B. Li, "Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems," *Probability and Statistics: Essays in Honor of David A. Freedman*, vol. 2, pp. 316-334, 2008.
- [2] K. Gallagher and M. Sambridge, "Genetic algorithms: a powerful tool for large-scale nonlinear optimization problems," *Computers & Geosciences*, vol. 20, no. 7, pp. 1229-1236, 1994.
- [3] T. S. Duque, K. Sastry, A. C. Delbem and D. E. Goldberg, "Evolutionary Algorithm for Large Scale Problems," in *ISDA 2007 - Seventh International Conference on Intelligent Systems Design and Applications*, Rio de Janeiro - Brazil, 2007.
- [4] Z. Yang, K. Tang and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Information Sciences*, vol. 178, no. 15, pp. 2985-2999, 2008.
- [5] I. T. Jolliffe, *Principal Component Analysis*, New York: Springer Series in Statistics, Springer, 2002.
- [6] L. Costa, "Application of Evolutionary Swarms and Autoencoders to Wind-Hydro coordination," Faculty of Engineering of the University of Porto, Porto, 2008.
- [7] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger and S. Tiwari, "Problem Definitions and Evaluation Criteria for the Special Session on Real-Parameter Optimization," Singapore, 2005.
- [8] S. Haykin, *Neural Networks A Comprehensive Foundation*, Hamilton, Ontario: Pearson - Prentice Hall, 1999.
- [9] M. Daszykowski, B. Walczak and D. L. Massart, "A journey into low-dimensional spaces with autoassociative neural networks," *Talanta*, vol. 59, no. 6, pp. 1095-1105, 2003.
- [10] M. Marseguerra and A. Zoia, "The Autoassociative Neural Network in signal analysis: I. The data dimensionality reduction and its geometric interpretation," *Annals of Nuclear Energy*, vol. 32, no. 11, pp. 1191-1206, 2005.
- [11] M. Marseguerra and A. Zoia, "The AutoAssociative Neural Network in signal analysis: II. Application to on-line monitoring of a simulated BWR component," *Annals of Nuclear Energy*, vol. 32, no. 11, pp. 1207-1223, 2005.
- [12] S. Narayanan, R. J. Marks, J. L. Vian, J. J. Choi, M. A. El-Sharkawi and B. B. Thompson, "Set constraint discovery: missing sensor data restoration using autoassociative regression machines," in *IJCNN'02. Proceedings of the 2002 International Joint Conference on Neural Networks*, Honolulu, Hawaii, 2002.
- [13] G. W. Cottrell, P. Munro and D. Zipser, "Learning internal representations from gray-scale images: An example of extensional programming," in *Ninth Annual Cognitive Science Society Conference*, Seattle (WA), USA, 1987.
- [14] M. K. Fleming and G. W. Cottrell, "Categorization of faces using unsupervised feature extraction," in *IJCNN - International Joint Conference on Neural Networks*, San Diego (CA), USA, 1990.
- [15] B. Golomb and T. Sejnowski, "Sex Recognition from Faces Using Neural Networks," in *Applications of Neural Networks*, A. Murray, Ed., Kluwer Academic Publishers, 1995, pp. 71-92.
- [16] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, no. 5786, pp. 504-507, 2006.
- [17] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks*, Perth, WA, 1995.
- [18] V. Miranda and N. Fonseca, "EPSO - best-of-two-worlds meta-

- heuristic applied to power system problems," in *CEC '02 - 2002 Congress on Evolutionary Computation*, Honolulu, HI, 2002.
- [19] M. Clerc, "The « alpine » function," Maurice Clerc, 1998.
- [20] V. Miranda, J. Krstulovic, J. Hora, V. Palma and J. C. Príncipe, "Breaker status uncovered by autoencoders under unsupervised maximum mutual information training," in *ISAP 2013*, Japan, 2013.
- [21] H. Yan, P. B. Luh, X. Guan and P. M. Rogan, "Scheduling of hydrothermal power systems," *IEEE Transactions on Power Systems*, vol. 8, no. 3, pp. 1358-1365, 1993.
- [22] V. R. Sherkat, R. Campo, K. Moslehi and E. O. Lo, "Stochastic Long-Term Hydrothermal Optimization for a Multireservoir System," *IEEE Transactions on Power Apparatus and Systems*, vol. 104, no. 8, pp. 2040-2050, 1985.
- [23] B. G. Gorenstin, N. M. Campodonico, J. P. da Costa and M. V. F. Pereira, "Stochastic optimization of a hydro-thermal system including network constraints," *IEEE Transactions on Power Systems*, vol. 7, no. 2, pp. 791-797, 1992.
- [24] V. Miranda, D. Srinivasan and L. M. Proença, "Evolutionary computation in power systems," *International Journal of Electrical Power & Energy Systems*, vol. 20, no. 2, pp. 89-98, 1998.
- [25] C. E. Zoumas, A. G. Bakirtzis, J. B. Theocharis and V. Petridis, "A Genetic Algorithm Solution Approach to the Hydrothermal Coordination Problem," *IEEE Transactions on Power Systems*, vol. 19, no. 2, pp. 1356-1364, 2004.
- [26] D. Castronuovo and J. A. Peças Lopes, "On the Optimization of the Daily Operation of a Wind-Hydro Power Plant," *IEEE Transactions on Power Systems*, vol. 19, no. 3, pp. 1599 - 1606, 2004.
- [27] D. Castronuovo and J. A. Peças Lopes, "Bounding Active Power Generation of a Wind-Hydro Power Plant," in *PMAPS 2004 - International Conference on Probabilistic Methods Applied to Power Systems*, Ames, Iowa, 2004.
- [28] J. W. Labadie, "Optimal Operation of Multireservoir Systems: State-of-the-Art Review," *Journal of Water Resources Planning and Management*, vol. 130, no. 2, pp. 93-111, 2004.
- [29] S. Soares and A. Carneiro, "Optimal Operation of Reservoirs for Electric Generation," *IEEE Transactions on Power Delivery*, vol. 6, no. 3, pp. 1101-1107, 1991.
- [30] C. Lyra, H. Tavares and S. Soares, "Modelling and Optimization of Hydrothermal Generation Scheduling," *IEEE Transaction on Power Apparatus and Systems*, vol. 103, no. 8, pp. 2126-2133, 1984.
- [31] S.-C. Chang, C.-H. Chen, I.-K. Fong and P. B. Luh, "Hydroelectric generation scheduling with an effective differential dynamic programming algorithm," *IEEE Transactions on Power Systems*, vol. 5, no. 3, pp. 737-743, 1990.
- [32] S. Soares, C. Lyra and H. Tavares, "Optimal Generation Scheduling of Hydrothermal Power Systems," *IEEE Transactions on Power Apparatus and Systems*, vol. 99, no. 3, pp. 1107-1118, 1980.
- [33] U. S. Department of the Interior Bureau of Reclamation Power Resources Office, "Managing Water in the West - Hydroelectric Power," U.S. Department of the Interior, 2005.

VI. BIOGRAPHIES

Vladimiro Miranda (M'90, SM'04, F'06) received his Ph.D. degree from the Faculty of Engineering of the University of Porto, Portugal (FEUP) in 1982 in Electrical Engineering. In 1981 he joined FEUP and currently holds the position of Full Professor. He is also a researcher at INESC since 1985 and is currently Director at INESC Porto, the leading institution of INESC TEC – INESC Technology and Science, an advanced research network in Portugal. He is also President of INESC P&D Brasil. He has authored many papers and been responsible for many projects in areas related with the application of Computational Intelligence to Power Systems.

Vera Palma graduated from FCUP, in 2007 in Applied Mathematics and is currently a researcher at INESC TEC, Portugal, in the Power Systems Unit.

Joana da Hora Martins received a M.Sc. degree in Industrial Engineering in 2009 at the Faculty of Engineering of the University of Porto (FEUP), and a M.Sc. degree in Quantitative Methods in 2012 at the Faculty of Economics of the University of Porto (FEP), Portugal. She is currently a PhD student at FEUP. She was a researcher at the department of Industrial Engineering of FEUP (April 2009 - August 2011), and at Power Systems Unit of INESC-TEC (November 2011 - June 2013), Portugal.