

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

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

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

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

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



Multi-Objective Optimization Methods Based on Artificial Neural Networks

Sara Carcangiu, Alessandra Fanni and Augusto Montisci
*Electrical and Electronic Engineering Department, University of Cagliari,
Cagliari, Italy*

1. Introduction

During the last years, several optimization algorithms have been presented and widely investigated in literature, most of which based on deterministic or stochastic methods, in order to solve optimization problems with multiple objectives that conflict with each other.

Some multi-objective stochastic optimizers have been developed, based on local or global search methods, in order to solve optimal design problems. Despite the significant progress obtained in this field, there are still many open issues. In fact, both the deterministic and stochastic approaches present hard limits.

In the first case, although the number of function evaluations needed to reach the optimal solution is generally small, the risk to be trapped in local minima is very high, whereas in the second case, the probability to reach the optimal solution is higher but the computational cost could become prohibitive.

In particular, this is the case of the electromagnetic problems. Electromagnetic devices are fundamental in the modern society. They are used for storing and converting energy (Magele, 1996), manufacturing processes (Takahashi et al., 1996), magnetic resonance imaging (Gottvald et al., 1992), telecommunications, etc.

The design optimization of the electromagnetic devices is one key to enhance product quality and manufacturing efficiency. Definition of geometric boundaries to achieve specific design goals together with nonlinear behaviour of ferromagnetic materials often give rise to multimodal, non-linear, and non-derivable objective functions. For this reason, resorting to numerical approaches, such as the Finite Element Method (FEM), to evaluate objective functions in many cases is compulsory.

When the number of design parameters to be optimized is considerable, the number of objectives evaluations to be performed could be of the order of thousand and the use of numerical solution during the optimization process can be unfeasible.

Approximating techniques have been proposed as a way to overcome the time consuming numerical procedure (Alotto et al., 2001, Canova et al., 2003, Wang & Lowther, 2006). One of the most effective approximation approaches is based on Artificial Neural Networks. In fact, an alternative method to numerical evaluation consists of applying the optimization procedure to the approximation of the objective function, rather than to its numerical model (Abbass, 2003, Fieldsend & Singh, 2005, Carcangiu et al., 2008). On the other hand, the quality of the solution of the optimization problem depends on the error introduced by the approximation

model. In order to take under control such approximation error the constructive algorithm presented in (Carcangiu et al., 2009a) can be used to build the neural model.

The search algorithms presented in this chapter resort to a procedure able to solve inverse problems by inverting the NN approximation model (Cherubini et al., 2005). This procedure consists in imposing the value of the desired objective functions and by searching for the corresponding values of the design parameters.

The proposed approach allows one to look for the Pareto front solutions directly in the objectives space, rather than in the design parameters one, allowing both to uniformly sample the Pareto front, and to limit the computational load (Carcangiu et al., 2009b).

Moreover the search for the Pareto points directly in the objectives space allows one to exploit the *a priori* knowledge of the Decision Maker (DM), guiding the search towards non-dominated solutions having predefined characteristics. In the following, an *interactive* approach is proposed. The DM can impose his own fitting criterion in the objective space fixing in this way a trajectory along which the Pareto points can be searched. In such a way, sampling of the whole Pareto front is avoided, and the deterioration of the different objective functions during the search for the Pareto optimal solution is kept under control.

The remainder of this chapter is organized as follows. In the Section 2 an overview of the multi-objective optimization is given. Section 3 describes the method to construct neural models having a prefixed precision degree. In Section 4 the Neural Network Inversion procedure is illustrated. In Sections 5 and 6 the search algorithms are described. Analytical and electromagnetic applicative examples are presented in Section 7, and the results are discussed. In Section 8 conclusions are drawn.

2. Multi-objective optimization

Multi-objective Optimization Problems (MOPs) usually present a possibly uncountable set of solutions, called Pareto optimal solutions, which, when evaluated, produces vectors whose components represent trade-offs in objective space. A DM has to choose acceptable solutions by selecting one or more of these vectors.

The MOPs objective functions are designated $f_1(\underline{x}), f_2(\underline{x}), \dots, f_k(\underline{x})$, where k ($k > 1$) is the number of distinct objective functions and \underline{x} is the decision vector. The objective functions form a vector function $\underline{f}(\underline{x})$ defined by $\underline{f}(\underline{x}) = [f_1(\underline{x}), f_2(\underline{x}), \dots, f_k(\underline{x})]^T$. The optimization problem can be formalized as follows:

$$\begin{aligned} \min \quad & \underline{f}(\underline{x}) \\ \text{s.t.} \quad & g_j(\underline{x}) \leq 0, \quad j = 1, \dots, q \\ & h_j(\underline{x}) = 0, \quad j = q + 1, \dots, m \end{aligned} \quad (1)$$

where $g_j(\underline{x})$ and $h_j(\underline{x})$ define the feasible region Ω of the decision variables.

Two Euclidean spaces are considered in MOP formulation:

1. The n -dimensional space \mathfrak{R}^n of decision variables, where each coordinate axis corresponds to a component of vector \underline{x} (decision space).
2. The k -dimensional space of objective functions \mathfrak{R}^k , where each coordinate axis corresponds to a component of vector $\underline{f}(\underline{x})$ (objective function space).

Each point in the decision space represents a solution and corresponds to a certain point in the objective function space (Fig. 1).

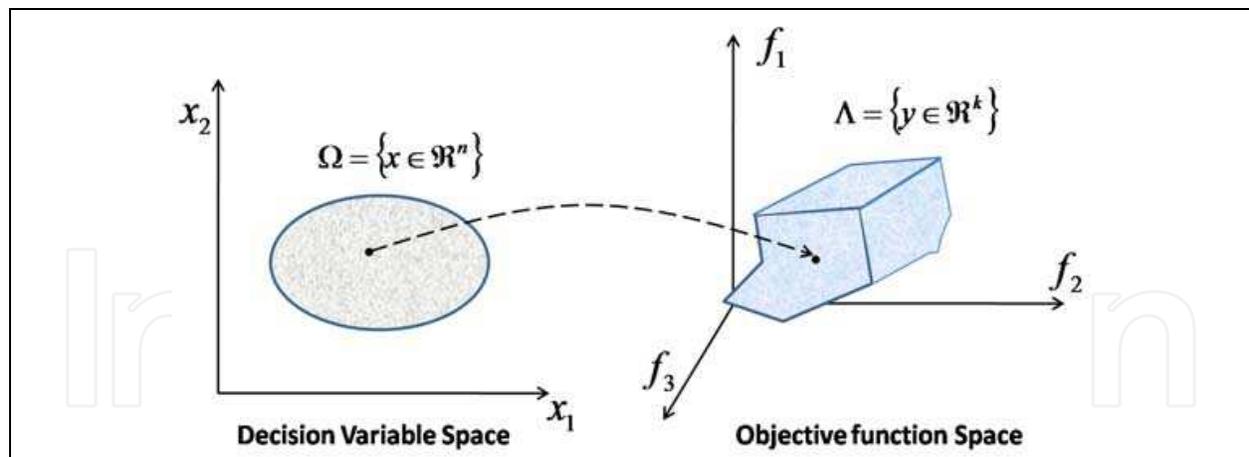


Fig. 1. Variable space mapped into the objective space.

In other words, one wishes to determine, among the set of all values satisfying (1), the particular set $\{x_1^*, x_2^*, \dots, x_n^*\}$ yielding optimum values for all the simultaneously considered k objective functions. The vector \underline{x}_i^* is reserved to denote the i^{th} optimal solution as MOPs often have many “optimal” solutions.

In a MOP, a solution \underline{x}_d is dominated by another solution \underline{x}_{nd} if \underline{x}_{nd} is better than \underline{x}_d on all objectives, and it will be denoted, here, by $\underline{x}_d \prec \underline{x}_{nd}$. A solution \underline{x}_p is a Pareto optimal solution if no objective function can be improved without worsening at least one other objective function. Such solution is not unique, and the set of the Pareto optimal solutions are known as the Pareto front.

In the objective domain search space it is possible to identify the **Utopia** and **Nadir** points, whose components are the best and the worst values respectively of all the objective functions. The Utopia point is then defined as the ideal solution in which all the objective functions have their optimum of the problem considering each objective separately. The Utopia is usually unfeasible.

In the objectives space two regions can be distinguished: the former is the feasible region whose points correspond to existing solutions in the parameters space, the latter is the unfeasible region, which is the complementary region in the objectives space. The frontier that separates these regions is not necessarily composed by non-dominated solutions; therefore in general the Pareto Front is included in such frontier. In general, it is not easy to find an analytical expression of the line or surface that contains the Pareto optimal points, and the normal procedure consists in computing a Pareto Optimal set of points. When a sufficient number of these points have been obtained, the DM may proceed to take the final decision.

Various approaches can be used to guide the DM towards a final solution among the Pareto optimal solutions (Pareto front): *a priori*, *a posteriori*, and *interactive* approaches, which make use of some utility criteria.

In the *a priori* approaches, the DM combines the objectives into a global utility function, thus transforming the MOP into a standard scalar optimization problem, which can be solved using traditional optimization methods. Although they have been widely used in the past, *a priori* techniques suffer from various drawbacks: they do not work properly with non convex Pareto fronts; they provide a single Pareto optimal solution, which is very sensitive to the scalarization of the objectives and to the choice of the parameters (e.g., weighting

coefficients, target values, starting point) associated with the preferences of the DM. In the *a posteriori* approaches, firstly the multi objective search is performed in order to sample the Pareto front and then a fitting criterion is applied to perform the ultimate choice. The *interactive* approaches, like that proposed in this chapter, are capable to overcome the previously described drawbacks.

The performance of a MOP solver can be evaluated on the basis of different criteria: capability of finding Pareto optimal solutions; capability of uniformly sampling the Pareto front; limited computational cost. All these criterion have been considered in evaluating the performance of the algorithms presented in this chapter.

3. Neural Network approximation model

As previously mentioned, one of the most effective approximation approach is based on Artificial Neural Networks. A Neural Network (NN) suitably trained with a limited number of configurations can be successfully used in the optimization procedure, and it can evaluate the objectives values instead of the costly assessment of the numerical procedure. Indeed, it is well known that NNs are effective tools for modelling the non-linear interactions among multiple variables (Principe et al., 2000, Haykin, 1998).

In our procedure a three layers MultiLayer Perceptron (MLP) is trained to capture the functional relationship between the design parameters and the objective functions of the optimization problem. The structure of the used MLP is shown in Fig. 2.

The functional relationship is expressed as in the following:

$$\begin{cases} \underline{W}_2 \cdot \sigma(\underline{W}_1 \underline{x} + \underline{b}_1) + \underline{b}_2 = \underline{u} \\ \underline{x} \in D_x \end{cases} \quad (2)$$

where \underline{x} is the input of the network (corresponding to the design variables), \underline{u} and \underline{h} are the input and the output of the hidden layer, respectively, $\sigma(\cdot)$ is the hidden logistic activation function, and \underline{u} is the output of the network (corresponding to the specified target). In the case of Fig. 2, the output layer has a linear activation function.

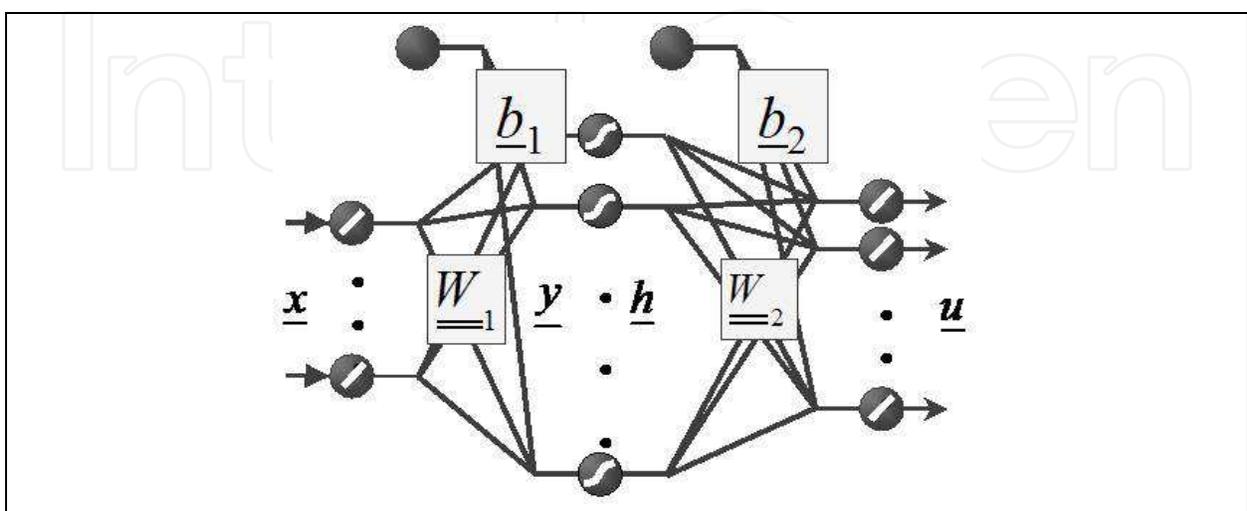


Fig. 2. MLP neural network architecture with a single hidden layer.

\underline{W}_1 and \underline{W}_2 are the weights of the connections between the input and hidden layers, and the hidden and the output layers, respectively, and \underline{b}_1 and \underline{b}_2 are the biases of the hidden and output layers, respectively.

The problem in synthesizing such network is to determine both the number of nodes in the unique hidden layer and the weights of the synapses between the layers. The former part of the problem is usually solved by a trial and error procedure, i.e., by training several networks of increasing or decreasing size on the same training patterns, whereas the latter part is solved by using learning algorithms based on error minimization to find the connection weights.

Recently, different interpretations for neural networks have been given, leading to approach the problem in a completely different way with regard to the classical error correction methods. In particular, a geometrical interpretation of the neural network has been proposed by some authors (Delogu et al., 2008) where each neuron behaves as a linear separator and the connection weights converging to that neuron are equal to the coefficients of the hyper-plane that defines the separation. In particular, in (Delogu et al., 2008) the authors proposed a new method to synthesize MLP networks with a single hidden layer, which are able to correctly classify whatever finite real valued training-set. The key idea of such method is to project the training set in a high-dimensional space, called feature space (Vapnik, 1998), by means of a set of step functions. The images of the two classes of training examples are always linearly separable in the feature space, therefore it is possible to define a unique output neuron that performs the definitive separation.

3.1 Synthesis of the Neural Network

In (Carcangiu et al., 2009a) the method for the synthesis of neural classifiers described in (Delogu et al., 2008) has been adapted to the synthesis of neural regressors.

In order to synthesize the MLP neural network, the function approximation problem has to be firstly converted in a classification problem. To this end, the continuous values of the function to be approximated have to be quantized. Associating a binary coding to each interval, a classification problem can be defined, where each digit is associated to an output node of the network. By choosing a coding with $\log_2 N$ bits, where N is the number of quantization levels, the minimal dimension of the output layer is obtained. The activation functions of both hidden and output nodes are step functions. The number of nodes in the hidden layer, as well as the synaptic weights of the network, will be automatically set during the training phase following the procedure in (Delogu et al., 2008).

In order to obtain a continuous approximation function, the step activation functions of the hidden nodes of the previously synthesized neural network are substituted with sigmoid functions:

$$\sigma(y_i) = \frac{1}{1 + \exp(-\alpha_i y_i)}$$

where y_i is the input to the i^{th} hidden neuron, and α_i tunes the slope of the i^{th} sigmoid. Furthermore, the second layer of connections and the output neurons are substituted by a unique linear output neuron and the corresponding connections. The connections weights between the hidden layer and the output node have to be evaluated, in order to minimize

the mean squared error on the training set. As the output of the hidden layer and the output of the network are linearly related, the best approximation is obtained with the regression hyper-plane. The coefficients of such hyper-plane are used as connections weights of the second layer. The value of the mean squared error depends on the correlation between the output of the hidden layer and the output of the network, which in turn is affected by the slope coefficients α_i . The best values of the coefficients can be found by means of an iterative optimization procedure. In this way we can construct an MLP that is able to approximate a continuous function with a prefixed precision.

As an example, let us consider the analytical Schwefel function defined in a 1-D input space:

$$f(x) = -x \cdot \sin(\sqrt{x}) \quad 0 \leq x \leq 500$$

This function presents several local minima. In Fig. 3, the normalized Schwefel function is reported (continuous line). In order to convert the function in a discrete one, $N=32$ intervals are chosen (see Fig. 3); hence the corresponding binary coding has 5 digits.

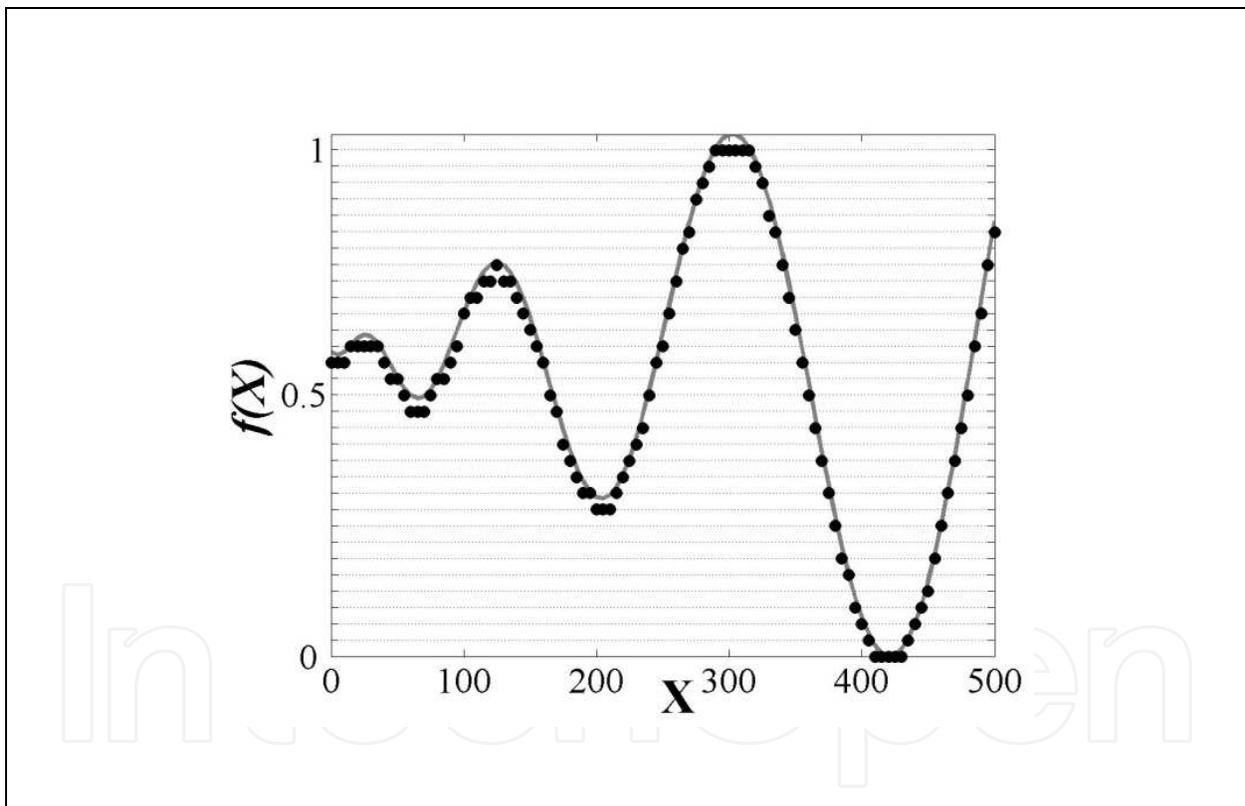


Fig. 3. **Continuous line:** Schwefel function; **dots:** the training set.

The neural network to be synthesized has one input node, corresponding to the x values, and 5 output nodes that correspond to the 5 digits of the binary coding. A training set of 100 examples is selected (see the dots in Fig. 3), and the synthesis of the network is obtained running the procedure in (Delogu et al., 2008).

In Fig. 4 a) the Schwefel function approximated by the smoothed neural model is shown, whereas in Fig. 4 b) the corresponding approximation error is reported. The values of the parameters α_i have been obtained minimizing the MSE evaluated on 500 examples.

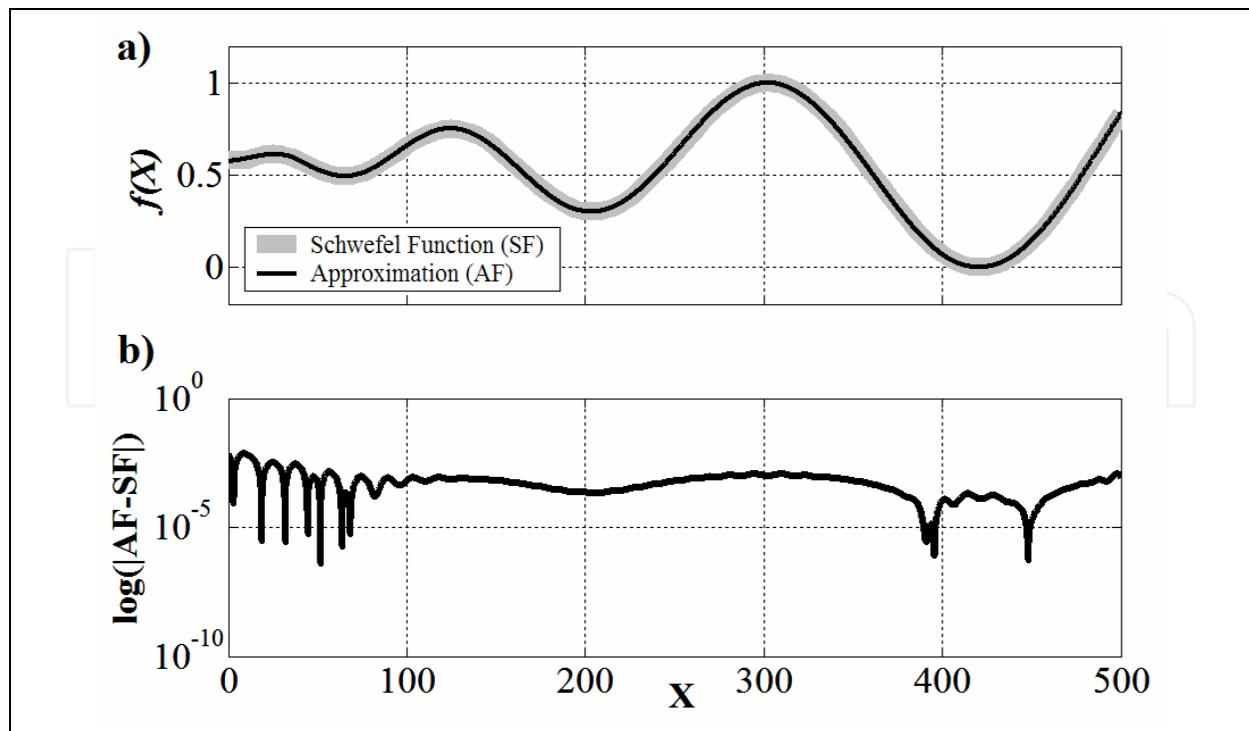


Fig. 4. a) **Continuous gray line:** Schwefel function (SF); **Continuous black line:** function approximated by the neural model (AF); b) absolute approximation error.

4. Neural Network inversion

The procedures proposed in this chapter consist in inverting the NN model of the problem. The aim of the inverting procedure is to determine the inputs that correspond to prefixed target outputs (Jensen et al., 1999, Bao-Liang Lu et al., 1999).

Referring to Fig. 5, the inversion of the MLP consists in finding a solution of the non linear neural network equations system:

$$\begin{aligned}
 a) \quad & \underline{W}_1 \cdot \underline{x} + \underline{b}_1 = \underline{y} \\
 b) \quad & \underline{W}_2 \cdot \underline{h} + \underline{b}_2 = \underline{u} \\
 c) \quad & \underline{h} = \sigma(\underline{y})
 \end{aligned} \tag{3}$$

The non linearity is introduced by the non linear activation function of the hidden neurons.

Since non linear equations can have multiple solutions, it seems that also the direct neural network model has a built-in non-invertible character (Rico-Martinez et al., 2000).

Nevertheless, if the target of the network can be specified with a prefixed degree of error, the iterative procedure in (Cherubini et al., 2005) can be run in order to find (if it exists) a solution whose value differs from the specified target less than the error threshold imposed. If this solution does not exist, the error constraints can be relaxed until a solution is found.

By means of equation (3.a), the input domain D_x can be linearly projected into the space Y where the vector \underline{y} is defined, obtaining the domain D_y . This means that, in order to match the constraints of the input, the vector \underline{y} has to belong to D_y . On the other hand, by means of equation (3.b), the output domain D_u can be linearly projected into the space H where the

vector \underline{h} is defined, obtaining the domain D_h . In order to match the constraints of the output, the vector \underline{h} has to belong to D_h . The equation (3.c) states a biunivocal relationship between \underline{y} and \underline{h} , so that the domain D_y can be projected into the space \mathbf{H} throughout the hidden layer obtaining the domain D'_y . Note that, also D_h can be projected into the space \mathbf{Y} throughout the hidden layer. A point, which belongs to the intersection between the two domains D_h and D'_y matches at the same time the constraints of the input and the constraints of the output, and then it is a solution of the design problem.

In literature, there are several papers that deal with projection algorithms for finding intersection points between two convex sets (Elser et al., 2007; Bauschke et al., 2002). In particular, in (Bauschke et al., 2002) the convergence of the Fienup's algorithm (Fienup, 1982) to the intersection of two convex sets has been proven. No results are present in literature, which demonstrated the convergence for non convex sets. In fact, in the majority of literature, which deals with most difficult real problems, the convergence is based almost entirely on a large body of empirical evidence, as claimed in (Elser et al., 2007).

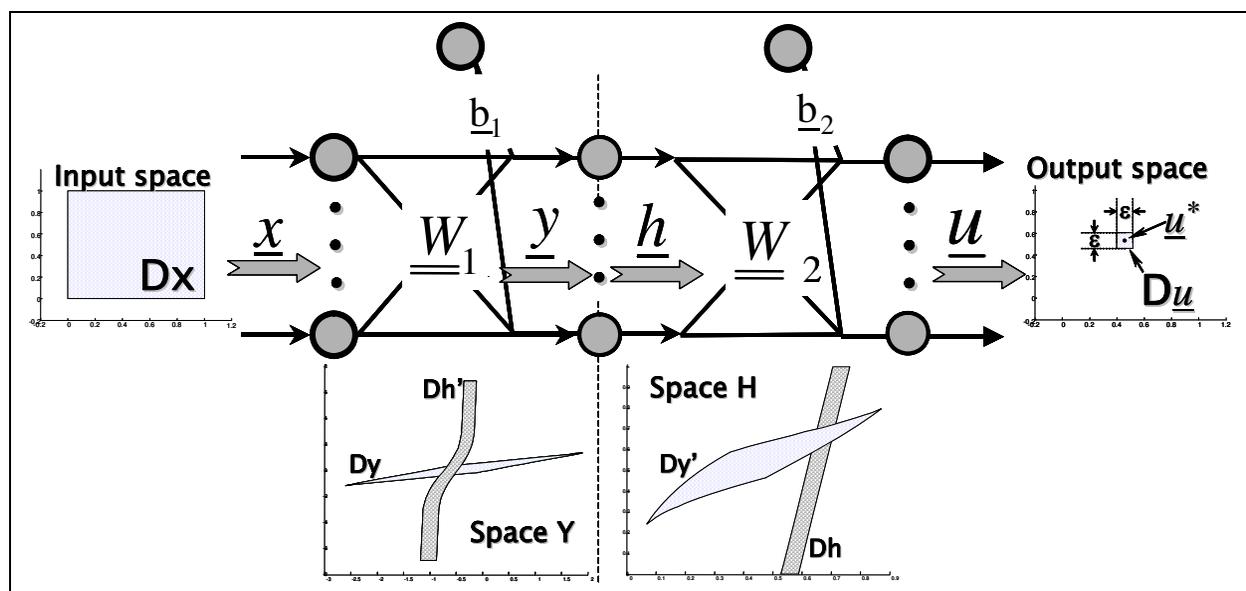


Fig. 5. MLP network structure: D_x is the input domain, D_u is the output domain, D_y and D_h are the domains of \underline{y} and \underline{h} respectively.

The problem of finding the intersection between non convex sets has been also dealt with in (Carcangiu et al., 2009b). Let us suppose, to fix the ideas, that the domain D_h has been projected into the space \mathbf{Y} (before the hidden layer), and we are trying to find out a point of the intersection I_{yh} between the nonlinear projected domain D'_h and the linear domain D_y . Starting from a point external to a linear domain, it is easy projecting such point on the domain, namely to find the point of the domain nearest to the starting point. Projecting is more difficult in the case of nonlinear domains. The easiness to project a point on a linear domain is due to the fact that one must follow a linear path in the direction orthogonal to one plane. This is no longer valid for a nonlinear surface, because a univocal orthogonal direction does not exist, so the shortest path to reach the surface is not known a priori. In the case on study, one has difficulties on projecting points on the nonlinear domain D'_h , whereas there is no difficult if the same projection is performed in the space \mathbf{H} , where the domain is linear. In order to exploit the algorithms available for linear domains, the nonlinear facets of

the domain I_{yh} are approximated by means of hyperplanes. In doing this, we are aided by the fact that each coordinate of the space \mathbf{Y} is related to its corresponding coordinate of the space \mathbf{H} independently from the others. Therefore, we can approximate the nonlinear facets by substituting the sigmoidal functions with their first-order approximation:

$$w_1\sigma(y_1) + \dots + w_n\sigma(y_n) \cong w_1[\sigma(y_{10}) + \sigma'(y_1) \cdot (y_1 - y_{10})] + \dots + w_n[\sigma(y_{n0}) + \sigma'(y_n) \cdot (y_n - y_{n0})] \quad (4)$$

where $\sigma(\cdot)$ is the sigmoidal function assumed for the neurons of the hidden layer, $\sigma'(\cdot)$ represents the first-order derivative of the sigmoid, $\mathbf{y}_0 = (y_{10}, \dots, y_{n0})$ represents the starting point from which the projection has to be calculated. The goodness of the projection on the approximated domain depends on the precision of the approximation (4), and then on the length of the step between the starting and the projected points.

Anyway, the substitution (4) allows one to treat the domain I_{yh} as if it were linear, the starting point can be projected on the approximated domain and a new linear approximation is calculated in the arrival point.

In Fig. 6 a sequence is shown that describes how the projection procedure performs. Starting from a generic point \mathbf{y}_0 of the space \mathbf{Y} , e.g. a point belonging to D_y , firstly a first order approximation of the sigmoid functions is calculated, taking the point \mathbf{y}_0 as reference point. The domain D'_h is transformed in the domain L_h and a point of the intersection with D_y is searched. If such point belongs also to I_{yh} , the procedure ends, otherwise the obtained point is assumed as starting point in the successive iteration.

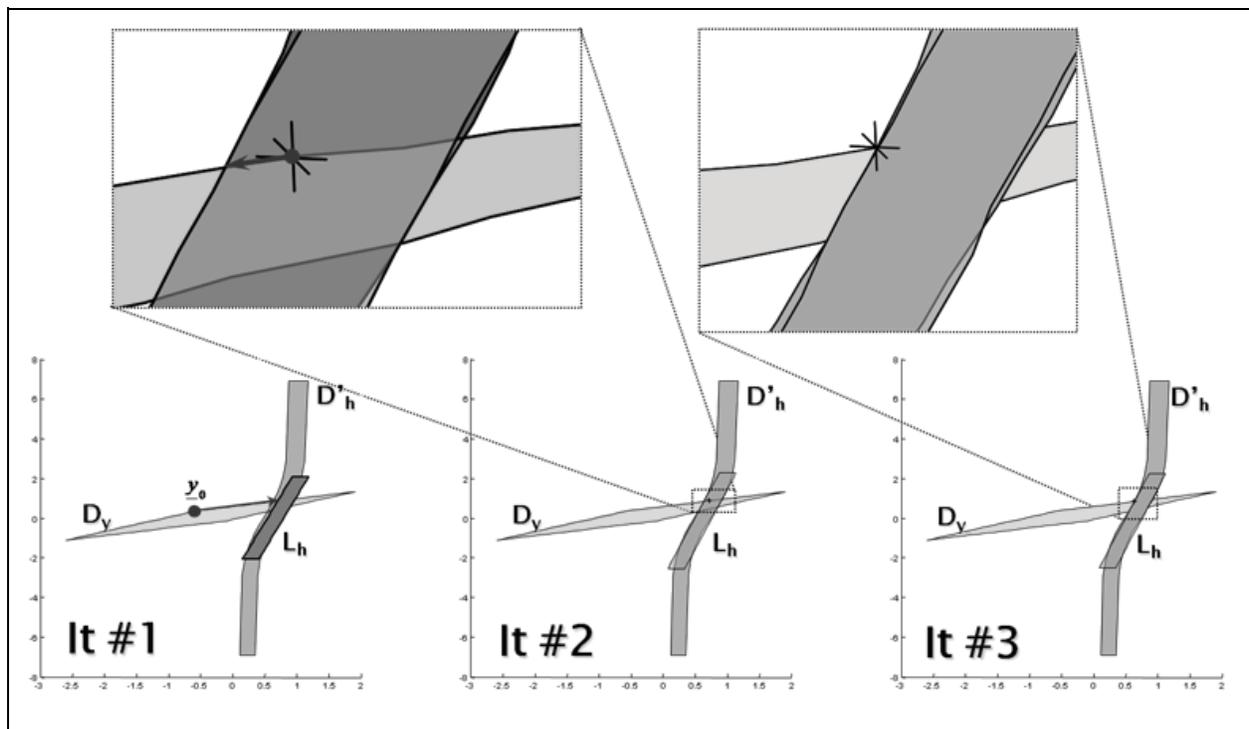


Fig. 6. Procedure for intersection searching.

The intersection between D_y and L_h could be empty, but this not implies that there isn't a solution of the inversion problem. Hence, if no point is found that belongs to the intersection between D_y and L_h , one of the two nearest points between such domains is taken as solution.

A suitable relaxation of the constraints on one or both the input and the output domains will make the intersection I_{yh} not empty and the procedure can be performed once again.

In Fig. 7, the result of the procedure applied to the neural model, which approximates the 1-D Schwefel function, is reported.

The iterative procedure converges to the optimum (circle in Fig. 7 a)). As can be noted from Fig. 7 b), the trajectory followed by the inversion algorithm, starting from a randomly chosen point, crosses a local minimum and reaches the global minimum in 14 iterations.

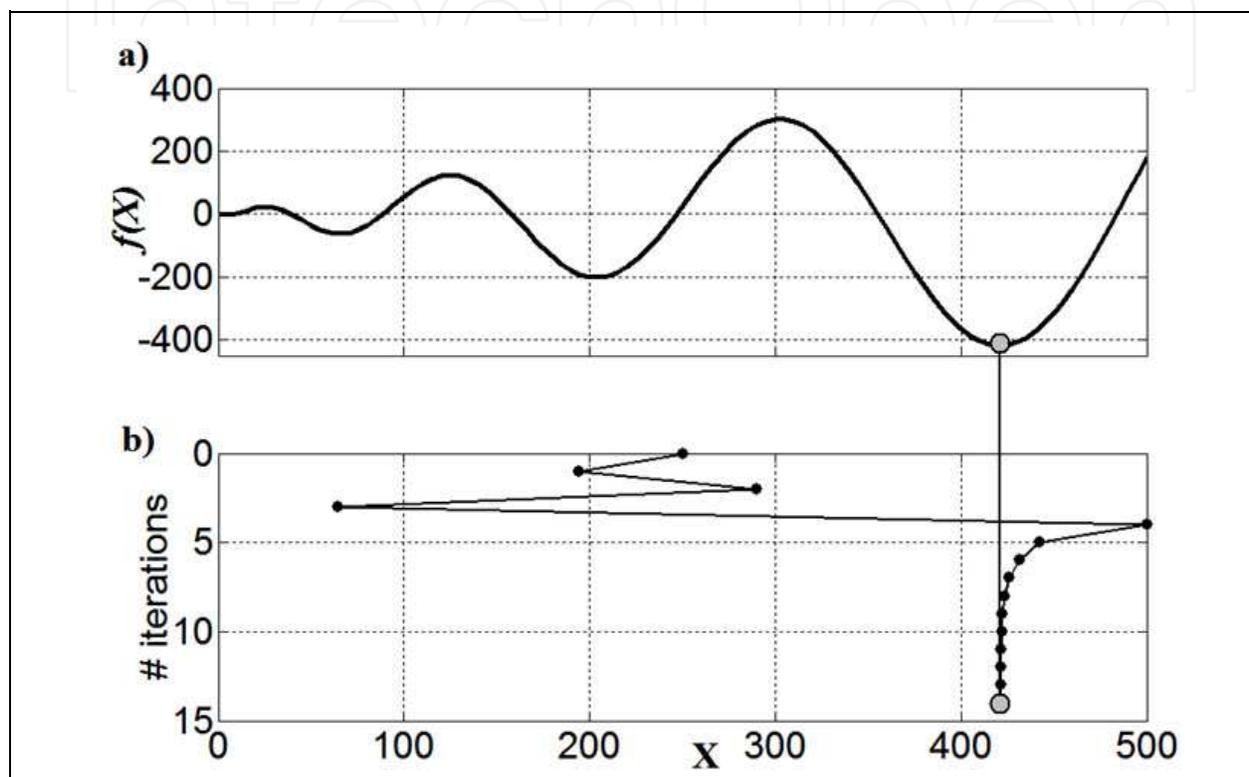


Fig. 7. a): Continuous line: Schwefel function approximated by the neural model; dot: optimal objective function value founded; b) Trajectory followed by the inversion algorithm.

5. Sampling of the whole Pareto Front (IN-MO)

Using an MLP to describe the relationship between parameters and objective functions allows one to obtain two advantages: reducing the computational cost of function calls, which is the main obstacle to use some search strategies; exploiting the neural model in order to search the Pareto points directly in the objective space rather than in the decision space.

In the following, a search algorithm is described that allows us to sample the frontier (with a user-defined sampling step) starting from whatever point in the objectives domain. In Fig. 8, this method is illustrated for a problem with two objectives to be maximized.

Starting from a feasible point in the objective space, one can move toward the utopia point until an unfeasible solution is reached, which means that the frontier has been reached. Starting from this first point (0) one can start to sample the frontier. Firstly, the desired sampling step Δ of the frontier is set. Then, a circle is ideally drawn with centre placed on

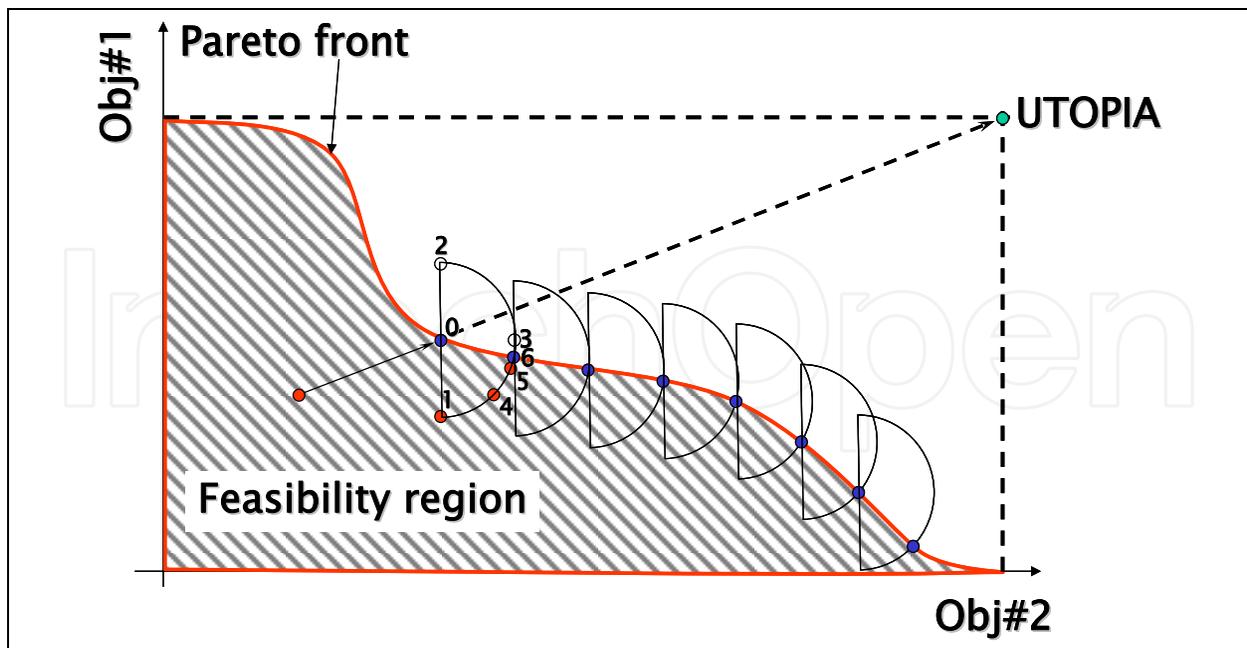


Fig. 8. Sampling of the Pareto front with IN-MO algorithm.

the first point (0) and radius equal to the sampling step. Two points (1) and (2) can be identified on the opposite sides of the circle: one in the feasibility region (1), and one in the non feasibility region (2). In a polar coordinates system with origin in the point (0), these points have coordinates:

$$p(1) = \begin{cases} r = \Delta \\ \theta = -\pi/2 \end{cases} \quad p(2) = \begin{cases} r = \Delta \\ \theta = \pi/2 \end{cases}$$

By means of, e.g., a bisection method, the intersection between the circle and the Pareto front (point 6) is reached. The new point becomes the centre of a new circle and the procedure is iterated until the whole frontier is reconstructed.

Once the frontier is available, a control on the dominance allows one to select the subset of frontier points that belong to the Pareto Front. The final choice is taken *a posteriori* by the decision maker on the basis of a fitness criterion.

6. Strategy-Driven Search Algorithm (SD-MO)

The large majority of approaches to a MOP consist of two stages: firstly a multiobjective search is performed in order to obtain a set of optimal solutions called Pareto front or non-dominated solutions, and then a fitting criterion is applied to perform the ultimate choice. Hence, the selection of a single solution from the set of non-dominated solutions is an *a posteriori* operation and the final solution consists of the optimal point that best fits the requirements of the decision maker.

Instead, in the approach here proposed, the usually unfeasible utopia solution is evaluated and then a strategy is assumed in order to iteratively improve the values of the objective functions with respect to a starting feasible solution, until an unfeasible solution is reached. Said strategy corresponds to a trajectory in the objective space (see Fig. 9), and the aim of the procedure is to find the intersection between such trajectory and the Pareto front. The choice of that trajectory is left to the DM.

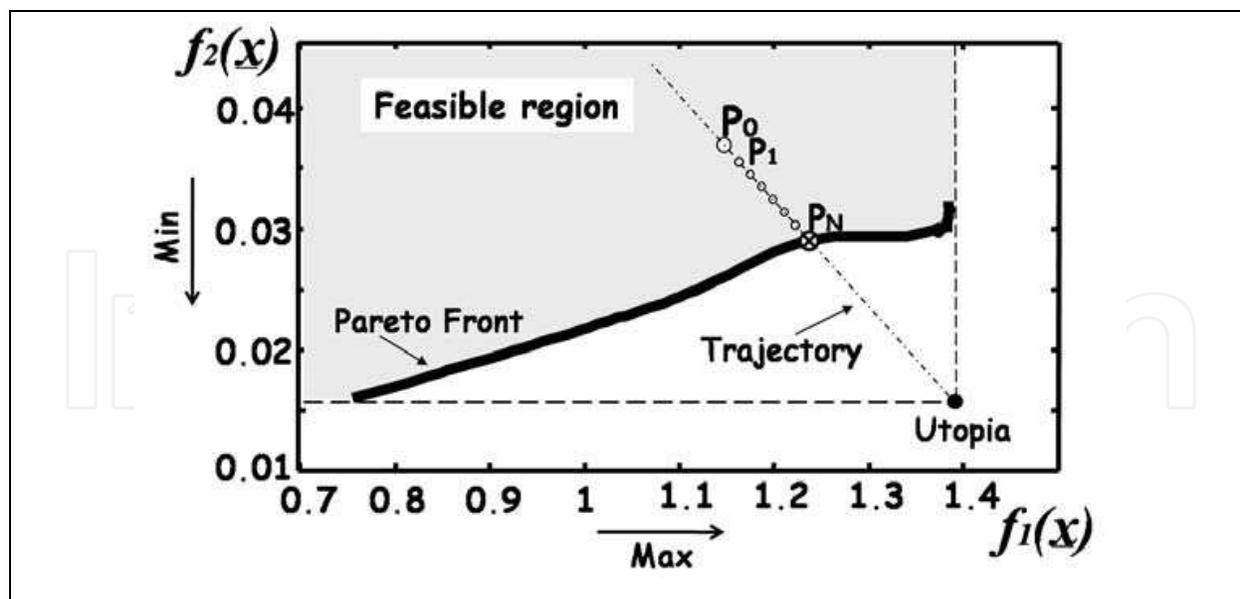


Fig. 9. Pareto front and search trajectory for a MOP.

Thanks to the NN inversion algorithm, such trajectory can be run along directly in the objective space rather than in the decision space. Starting from an initial feasible point P_0 on the trajectory (see Fig. 9) the algorithm iteratively move towards the utopia point until the unfeasible region is reached.

In this way, both the *a posteriori* information consisting in the feasibility of the found point and the *a priori* knowledge, imposed by the DM, are used to guide the search towards the desired non-dominated solution.

In order to find a frontier solution that belongs to the given trajectory, the proposed procedure implements a two-phases method.

In the first phase a starting point is searched, which belongs both to the feasible region and to the trajectory. Generally, such point will be dominated (i.e., it will not belong to the Pareto front).

The second phase consists in searching a frontier point, by following the trajectory.

Firstly, let us suppose the trajectory be a straight line. In the following subsection, the extension to the case of piecewise linear trajectory is given. Let

$$f_k = \underline{m}^T \cdot \underline{f}_{-1, \dots, K-1} + n \quad (5)$$

be the linear trajectory in the objective space, where K is the dimension of the objectives space, $\underline{f}_{-1, \dots, K-1}$ is the vector of all the objectives but the K -th, \underline{m} and n are the coefficients of the linear trajectory, carried on the K -th objective function f_k . The values of objective functions in (5) can be expressed as a function of the parameters of the MLP neural network model:

$$\underline{W}_{2,K}^T \cdot \underline{h} + b_{2,K} = \underline{m}^T \cdot \left(\underline{W}_{-1, \dots, K-1} \cdot \underline{h} + \underline{b}_{-1, \dots, K-1} \right) + n \quad (6)$$

where $\underline{W}_{2,K}$ is the K -th column of \underline{W}_2 , and $b_{2,K}$ is the K -th element of \underline{b}_2 . By re-ordering the equation (6), the following equation is obtained:

$$\left(\underline{W}_{2,K}^T - \underline{m}^T \cdot \underline{W}_{2,(1\dots K-1)} \right) \cdot \underline{h} + \left(b_{2,K} - \underline{m}^T \cdot \underline{b}_{2,(1\dots K-1)} \right) = n \quad (7)$$

The equation (7) represents a constraint for a new MLP neural network represented in Fig. 10. That network has only one output, the same input and hidden layers of the MLP in (3), and the same connections weights \underline{W}_1 and \underline{b}_1 , whereas the connections weights between hidden and output layers and the output bias are respectively equal to:

$$\begin{aligned} \underline{A} &= \underline{W}_{2,K} - \underline{W}_{2,(1\dots K-1)}^T \cdot \underline{m} \\ \beta &= b_{2,K} - \underline{m}^T \cdot \underline{b}_{2,(1\dots K-1)} \end{aligned} \quad (8)$$

Note that the parameters \underline{A} and β are known as they are expressed in terms of the parameters of the neural model in (3) and in terms of the coefficients of the linear trajectory, which are imposed by the DM. Thanks to the equations in (8), the first phase of the procedure can be solved by inverting the one-output neural network in Fig. 10.

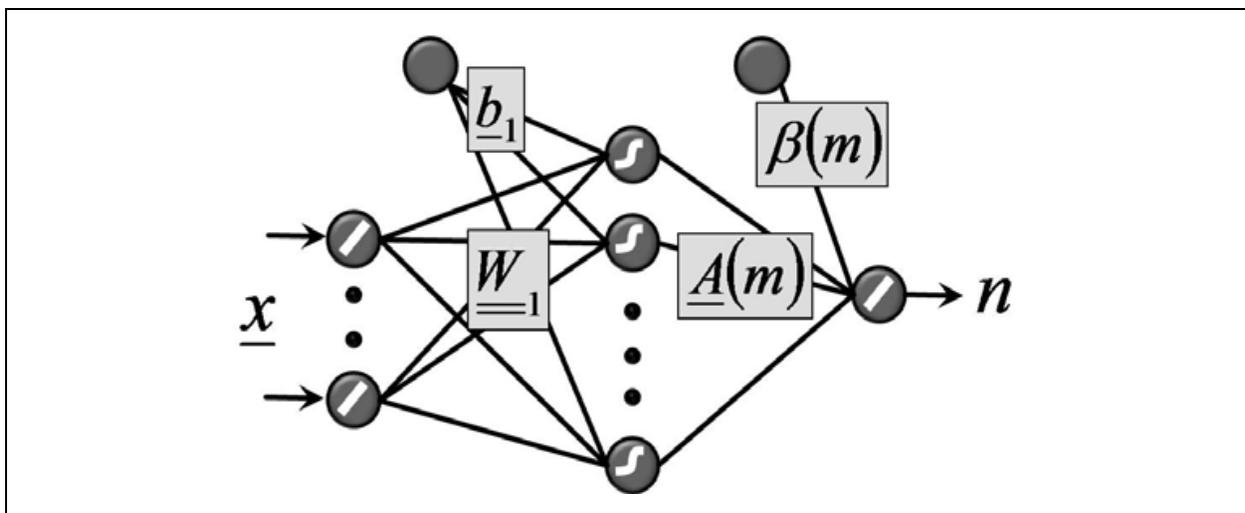


Fig. 10. NN for the search of the trajectory in the objective space.

Using as target output the value of the coefficient n , a corresponding input \underline{x} is obtained. Then, by forwarding the obtained input \underline{x} through the neural model in (3), a point $P_0 \equiv P_0(\underline{f})$ in the objective space is found, which surely lies on the trajectory (see Fig. 9). Once the point P_0 has been found, a step is performed on the linear trajectory towards the utopia point, and the endpoint of the step is assumed as target output in the inversion of the neural model in (3). If the inversion process leads to a feasible configuration, the procedure is iterated until the inversion process does not converge. The last visited feasible point P_N is assumed as belonging to the frontier of the feasible region. The step size is chosen according to the desired approximation of the final solution P_N to the frontier of the feasible region.

Such frontier point could not belong to the Pareto front depending on whether it is placed on a concave portion of the feasible region, or because the Pareto front is discontinuous (see Fig. 11, referring to the analytical problem reported in (Deb et al., 2005), which is discontinuous and non-convex).

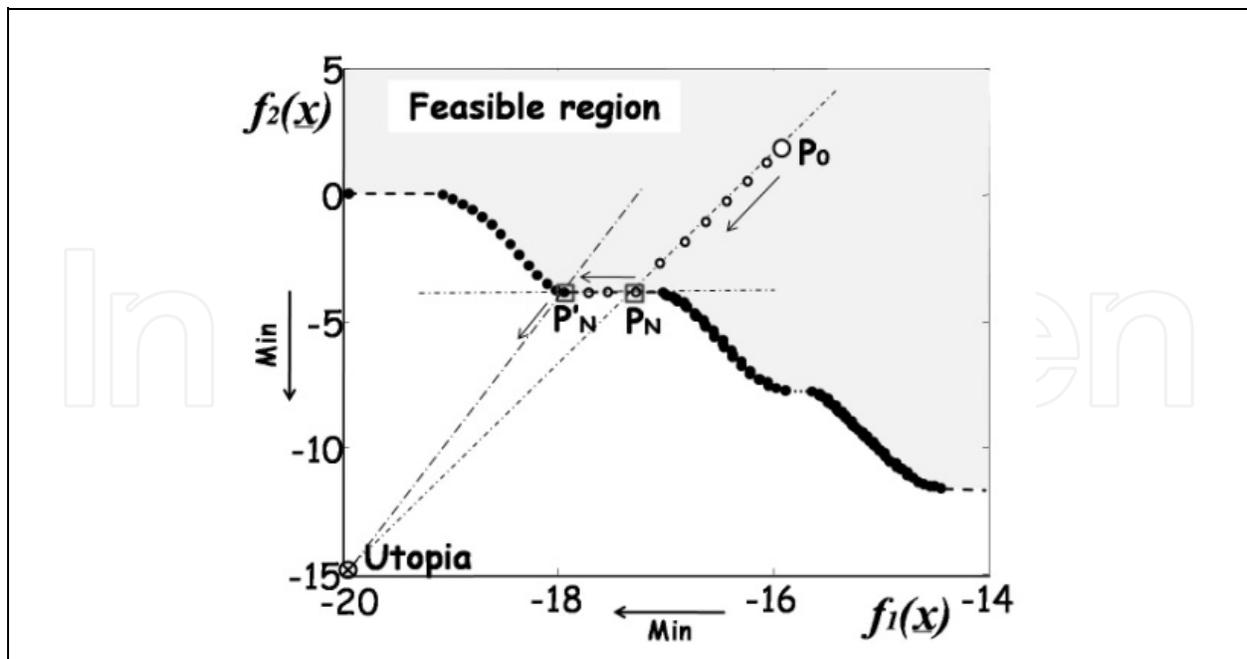


Fig. 11. Pareto front (●) and search trajectory (- -) for the Kursawe analytical problem (Deb et al., 2005).

For this reason, a strategy has been implemented to verify if the final point P_N belongs to the Pareto front rather than simply to the frontier of the feasible region. To this end, starting from P_N , one at a time, we try to improve the value of each objective, leaving the remaining ones at the values corresponding to the frontier point.

If this search gives positive result, P_N is substituted by the new dominating point (P'_N in Fig. 11). As in general also P'_N could not belong to the Pareto front, the decision maker has to define a new linear trajectory along which to search a new frontier point.

6.1 Piecewise linear trajectory

A piecewise linear trajectory formalizes a possible strategy introduced by the DM. By assuming an enough fine segmentation of the piecewise linear trajectory, a continuous curve can be also well approximated.

As an example, in Fig. 12, referring to the electromagnetic problem reported in (Di Barba & Mognaschi, 2005), a piecewise linear trajectory is chosen (U_0, U_1, U_2, \dots).

Following the linear trajectory U_0-U_1 , the two objectives deteriorate at the same rate, but the DM could be interested in introducing further constraints, e.g., on the maximum deterioration of objective function f_1 , hence the new linear trajectory U_1-U_2 is followed whenever that constraint is violated. The new trajectory deteriorates only f_2 . When f_2 deteriorates too much, the new trajectory U_2-U_3 is adopted.

Each segment of the trajectory represents a linear strategy where the current utopia point is represented by its non-dominated end point (U_0, U_1, U_2, \dots in Fig. 12). If the final point P_N^0 belongs to the segment U_0U_1 , the procedure terminates and P_N^0 is the searched frontier point, otherwise U_1 is assumed as new utopia point and the linear trajectory U_1-U_2 is assumed as new strategy. The procedure is iterated for all the segments until a solution is found that lies in the current segment of the piecewise linear trajectory (P_N^2 in Fig. 12).

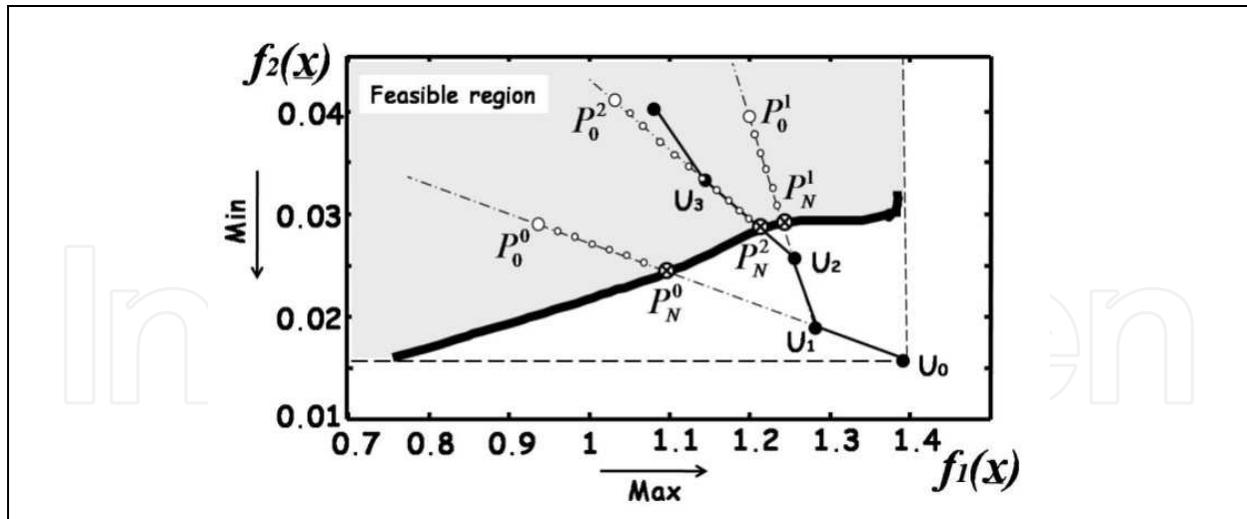


Fig. 12. Piecewise linear trajectory strategy (-●-) applied to the magnetic pole problem (Di Barba & Mognaschi, 2005).

7. Application and results

In order to evaluate the performance of the proposed approaches, both analytical and electromagnetic benchmarks have been used. The first two examples are electromagnetic problems for which the Pareto front has been sampled using the procedure presented in Section 5. One of such examples has been used to test the optimization strategy described in Section 6, together with an analytical problem.

7.1 High field superconducting dipole magnet

The dipole magnet (Fig. 13) consists of a pair of identical saddle-shaped coils of rectangular cross section (chequered area in Fig. 13). A circular shape iron yoke (dashed) is used to improve the field quality and intensity over the bore cross-section. An outer cylinder made of austenitic steel encloses the whole dipole assembly and provides pre-compression at cryogenic temperature due to the differential contraction.

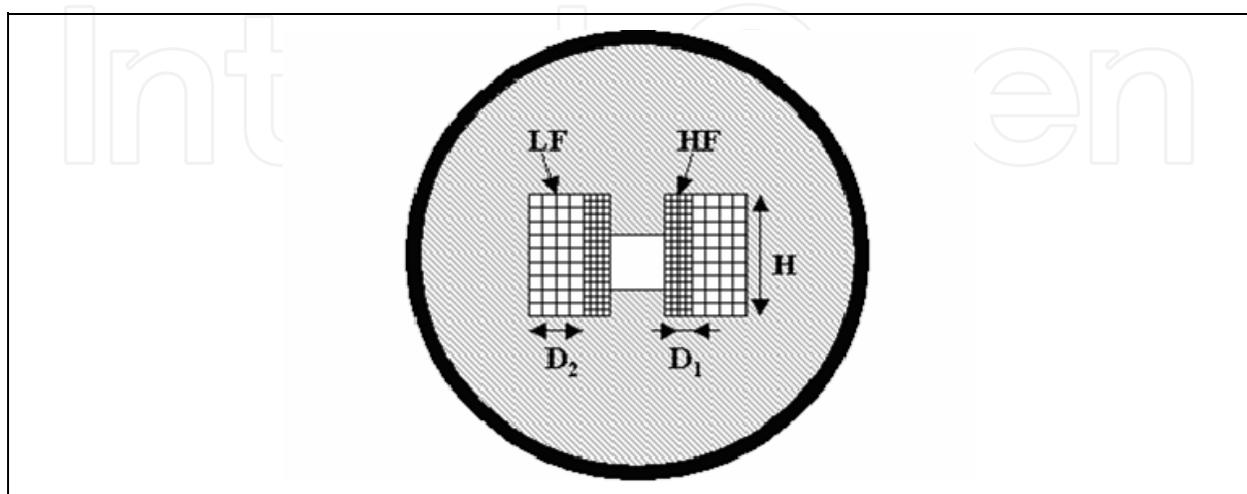


Fig. 13. Dipole assembly (Portone et al., 2006).

Each coil is made of a High Field (HF) section and a Low Field (LF) section. All conductors carry the same operating current, all turns being in series. The HF grade differs from the LF grade in the outer dimensions of the superconducting strands, resulting in a different current density between HF and LF sections. The HF and the LF sections together constitute the winding pack. The design problem consists in finding the optimal values of the HF and LF winding areas, therefore reducing the superconducting cable, in order to have a 12.5 T magnetic field value in the dipole axis. The multiobjective optimization problem can be stated as the minimization of the cost of the superconducting coils, while the prescribed magnetic field value can be considered as a constraint or as objective function to be maximized. The design variables of the MOP are reported in Fig. 13: the width D_1 of the HF section, the width D_2 of the LF section, and the height H of the winding pack.

This benchmark is a three parameters-two objectives problem, in which the design parameters are D_1 , D_2 , and H (see Fig. 13), and the objectives are the magnetic induction B and the total winding pack area A . The neural network model has 3 input neurons, 12 hidden neurons, and 2 output neurons, corresponding to B and A . The training, validation and test sets consist of 2448, 306, and 306 couples of input-output patterns.

When the learning phase ends, the MSE in the validation set is equal to $2.9e-006$. The maximum distance between a point analytically calculated and the corresponding point approximated by the neural model is 0.0564. In this case, the sampling step Δ has been set equal to 0.01, which corresponds to the denormalized steps $\Delta B=0.39$ T, and $\Delta A=0.003$ m².

In Fig. 14 the Pareto front obtained with the search algorithm is presented. The point corresponding to the actual design parameters in (Portone et al., 2006) is indicated with a circle.

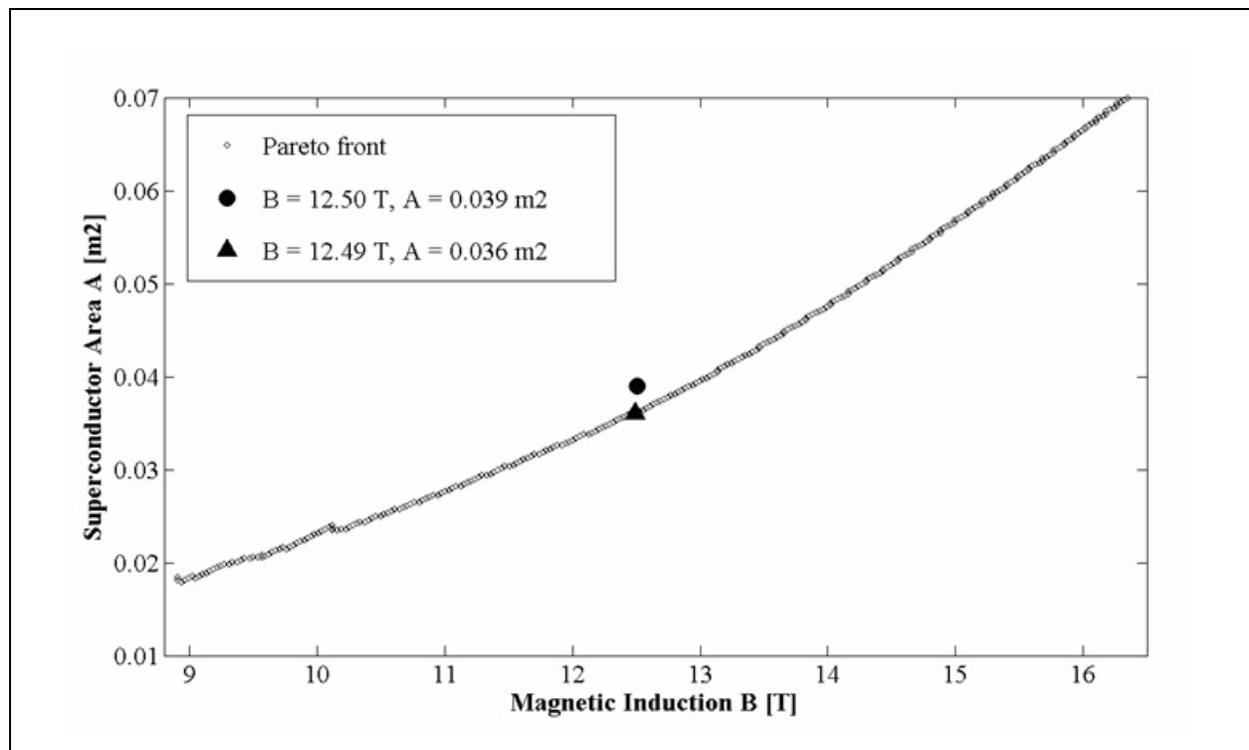


Fig. 14. Pareto front for the dipole magnet.

The closest point of the sampled Pareto front (triangle in Fig. 14) corresponds to the following design parameters: $B= 12.489$ T; $A= 0.036$ m²; $D_1=0.0289$ m; $D_2=0.1838$ m; $H=0.1696$ m. A saving of about 8% in the superconducting material is obtained, with a magnetic field variation less than 1%.

7.2 Optimal electromagnetic devices design (IN-MO algorithm)

The optimal shape design of a magnetic pole is considered (Di Barba & Mognaschi, 2005) to critically evaluate the suitability of the proposed algorithm in the field of electromagnetic devices design.

In Fig. 15, the model of the device is shown. Because of the symmetry with respect to the x-axis, only a half of the magnetic pole rectilinear section has been modeled. The current density is uniform in the winding and is zero elsewhere. The non-linear permeability of the ferromagnetic material is taken into account.

As far as the inverse problem is concerned, four design variables y_1, y_2, x_3, x_4 are selected. The feasible region of the design variables is defined by the conditions of both geometric congruency and non-saturation of the material.

Two objective functions are defined:

- f_1 is the maximum component of magnetic induction in the y-axis direction along the air gap midline, to be maximized;
- f_2 is the average component of magnetic induction in the x-axis direction in the winding, to be minimized.

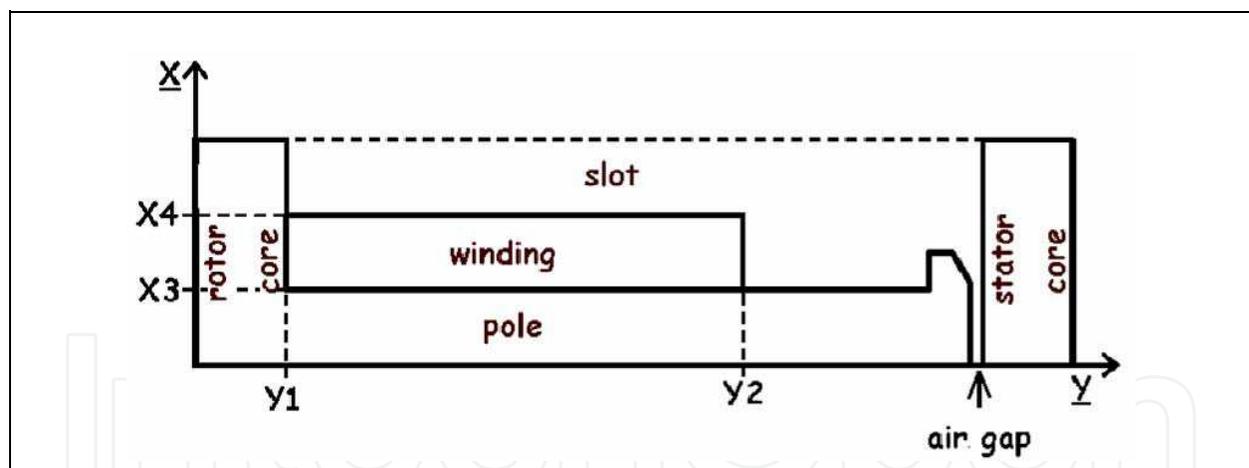


Fig. 15. Magnetic Pole.

The neural model has 4 inputs corresponding to the design variables, whose values are normalized in the range $[-1, 1]$. The hidden layer has 40 neurons, having hyperbolic tangent activation function. The output layer has two neurons, corresponding to the objective functions. Also the output values are normalized in the same interval of the inputs. The neural network is trained to associate the input vector of the design variables to the corresponding values of the objective function. To this end a set of input-output pairs is selected, and the network is modified in order to minimize the difference between the expected output and that one calculated with the network. The training examples are calculated by means of FEM simulations.

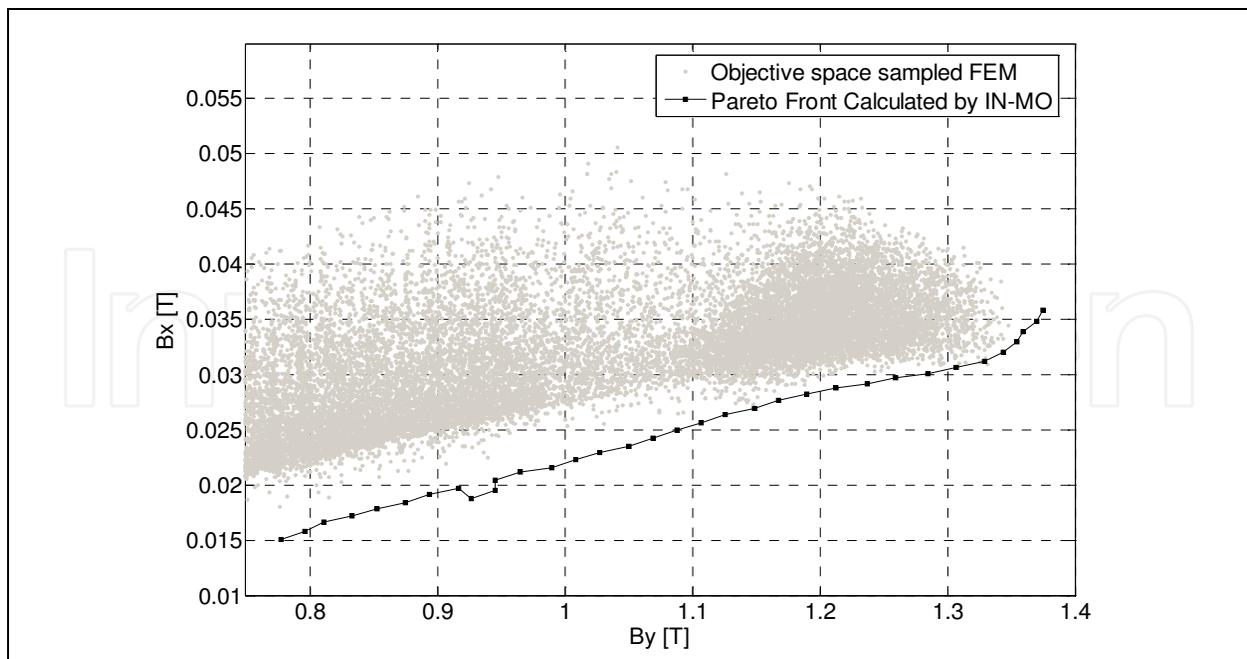


Fig. 16. Magnetic pole: approximated Pareto front.

Fig. 16 shows a random sampling of the objectives space and the Pareto front with 34 sample points obtained by means of the procedure proposed in Section 5.

It has to be highlighted that, the approximation of the Pareto front mostly depends on the accuracy of the neural approximation model, which is known a priori. This is a considerable advantage, especially in engineering problems.

7.3 Analytical test for the SD-MO algorithm

The Kursawe function is a three parameters-two objectives problem (Deb et al., 2005), where the two cost functions to be minimized are stated as:

$$\min \begin{cases} f_1(x_1, x_2, x_3) = \sum_{i=1}^2 \left(-10 \cdot e^{-0.2 \cdot \sqrt{x_i^2 + x_{i+1}^2}} \right) \\ f_2(x_1, x_2, x_3) = \sum_{i=1}^3 \left(|x_i|^{0.8} + 5 \cdot \sin(x_i)^3 \right) \end{cases} \quad (9)$$

where: $-5 \leq x_1, x_2, x_3 \leq 5$.

The MLP neural network model has 3 input neurons, corresponding to the values x_1 , x_2 , and x_3 , 50 hidden neurons, and 2 output neurons, corresponding to f_1 and f_2 .

In Fig. 11, the analytically calculated Pareto front is reported. As the Pareto front is discontinuous and non convex, the trajectory chosen by the DM could intersect the frontier of the feasible region in a point (P_N) that does not belong to the Pareto front. The proposed procedure looks for points dominating P_N , moving along the Cartesian coordinates in the objective space, as described in Section 6, and is able to automatically find a new point P'_N , which belongs to the Pareto front. A further search is then performed along the new trajectory from P'_N to the utopia point. As can be seen from Fig. 11, no other dominating points do exist, and the procedure stops.

7.4 Optimal electromagnetic devices design (SD-MO algorithm)

As in Section 7.2, in the following the optimal shape design of a magnetic pole (Di Barba & Mognaschi, 2005) has been considered in order to test the computational performance of the SD-MO algorithm.

Note that, the computational cost of the algorithms proposed in literature (Konak et al., 2006, Zitzler & Thiele, 1999) for MOPs exponentially grows with the number of objectives. Conversely, the aim of the SD-MO algorithm is to find a unique solution that both is a Pareto point and fulfills the requirements imposed by the DM. The computational cost of the algorithm is independent from the number of objectives, but only depends on the dimension of the sampling step chosen to follow the trajectory.

As an example, in Fig. 17, a possible piecewise linear trajectory is reported, which formalizes the strategy introduced by the DM. The algorithm starts using as strategy the linear trajectory U_0-U_1 . A starting point P_0^0 is found, which lies on this trajectory. Then, a number of steps are performed moving on the trajectory until the frontier of the feasibility region is intersected in the point P_N^0 . Such point does not belong to the segment U_0U_1 of the trajectory, hence U_1 is assumed as new utopia point for the new linear strategy U_1-U_2 .

A new starting point P_0^1 is found and the algorithm, moving towards U_1 , intersects the frontier in the point P_N^1 , which does not belong to the segment U_1U_2 . Finally, the linear trajectory U_2-U_3 is assumed as new strategy, with U_2 as utopia point. In this case, the search terminates in the point P_N^2 , which belongs both to the frontier of the feasible region and to the segment U_2U_3 of the strategy.

This point represents the non-dominated solution of the MOP.

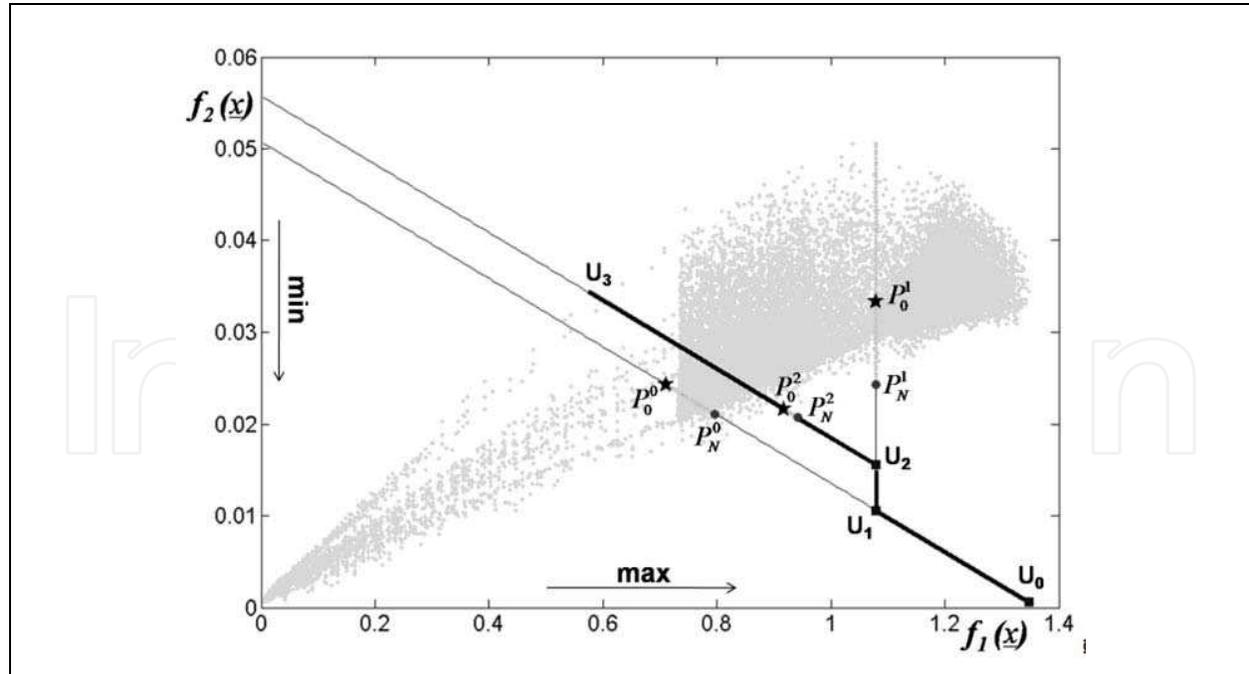


Fig. 17. Piecewise linear trajectory strategy (-●-) applied to magnetic pole problem.

Table I reports the found Pareto optimal solution. The approximated solution given by the neural model has been recalculated using FEM analysis. The error introduced by the approximation model is negligible. Fig. 12 reports the magnetic field distribution in the pole.

Design Parameters				NN Values		FEM Values	
y_1 [m]	y_2 [m]	x_3 [m]	x_4 [m]	f_1 [T]	f_2 [T]	f_1 [T]	f_2 [T]
7.63	14.93	14.49	14.45	0.942	0.02	0.893	0.02

Table 1. Pareto optimal solution of the Magnetic Pole.

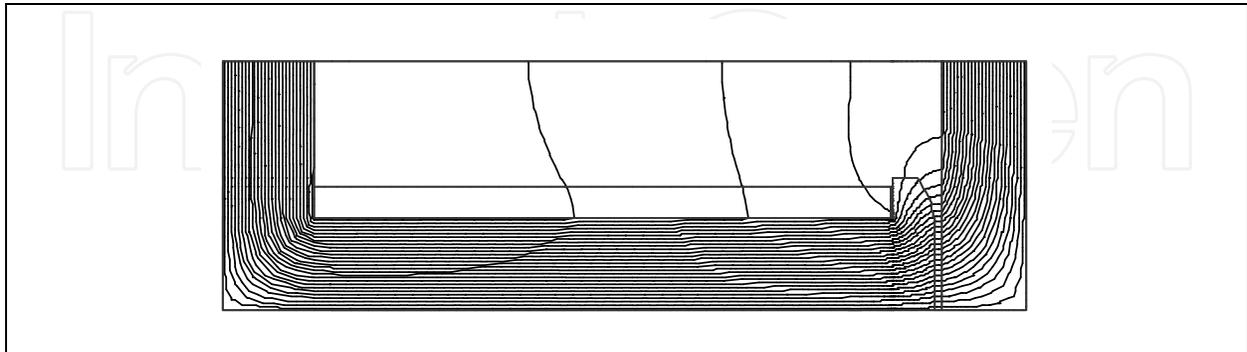


Fig. 12. Magnetic field distribution in the pole.

8. Conclusions

The optimal design of electromagnetic devices is of crucial importance in the modern industry. On the other hand, Multi-Objective Optimization reflects much better than the Single-Objective one the job of the designer, because the best project represents always the best compromise among conflicting exigencies. The performance of an approach that implements a Multi-Objective Optimization is evaluated on the basis of capability of finding Pareto optimal solutions; capability of uniformly sampling the Pareto front; limited computational cost. In this chapter some techniques have been described and the performances are evaluated with some analytical and electromagnetic benchmarks retrieved from the literature.

The capability of finding Pareto solutions has been enhanced by modelling the problem by means of neural networks synthesized by a constructive algorithm that takes under control the approximation error.

A technique that allows one to invert neural networks permits to perform the Pareto points search directly in the objective space. In this way the Pareto front can be uniformly sampled.

The use of neural models of the problem at hand together with an interactive search method, allows one to greatly reduce the computational cost of optimization.

The obtained results confirm the suitability of using the proposed methods in order to find Pareto optimal solutions in all the studied cases.

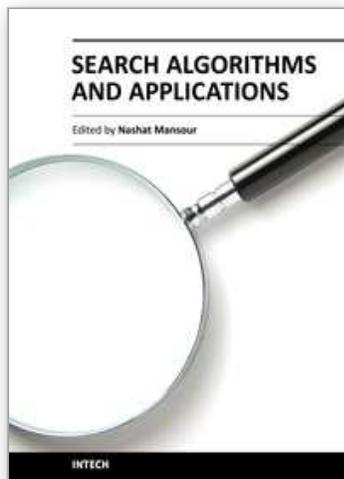
9. References

- Abbass, H.A. (2003). Pareto neuro-evolution: constructing ensemble of neural networks using multi-objective optimization, *Proceedings of CEC'03*, n. 3, pp. 2074 - 2080
- Alotto, P. & Nervi, M. A. (2001). An efficient algorithm for the optimization of problems with several local minima, *International Journal for Numerical Methods in Eng.*, n. 50, pp. 847-868

- Bao-Liang Lu, Kita, H. & Nishikawa, Y. (1999). Inverting feedforward neural networks using linear and nonlinear programming, *IEEE Transactions on Neural Networks*, vol. 10, no. 6, pp. 1271 - 1290
- Bauschke, H.H., Combettes, P.L. & Luke, D.R. (2002). Phase retrieval, error reduction algorithm, and Fienup variants: a view from convex optimization, *Journal of the Optical Society of America A.*, Vol. 19 No. 7, pp. 1334-45
- Canova A., Grusso G. & Repetto M. (2003). Magnetic design optimization and objective function approximation, *IEEE Transactions on Magnetics*, Vol 39, no. 5, pp. 2154-2162
- Carcangiu, S., Fanni, A. & Montisci, A. (2008). Multiobjective Tabu Search Algorithms for Optimal Design of Electromagnetic Devices, *IEEE Transactions on Magnetics*, Vol 44, no. 6, pp. 970-973
- Carcangiu, S., Fanni, A. & Montisci, A. (2009). A constructive algorithm of neural approximation models for optimization problems, *The International Journal for Computation and Mathematics in Electrical and Electronic Engineering (COMPEL)*, vol. 28-5; pp. 1276-1289, ISSN: 0332-1649
- Carcangiu, S., Fanni, A. & Montisci, A. (2009). Multi Objective Optimization Algorithm Based on Neural Networks Inversion, *Lecture Notes In Computer Science*, vol. 5517; pp. 744-751, Springer-Verlag, ISBN:978-3-642-02477-1, Berlin, Heidelberg
- Cherubini, D., Fanni, A., Montisci, A. & Testoni, P. (2005). Inversion of MLP neural network for direct solution of inverse problems, *IEEE Transactions on Magnetics*, vol. 41, no. 5, pp. 1784-1787
- Deb, K., Thiele, L., Laumanns, M. & Zitzler, E. (2005). Scalable test problems for evolutionary multi-objective optimization, *TIK-Technical Report.*, no. 122
- Delogu, R., Fanni, A., Montisci, A. (2008). Geometrical synthesis of MLP neural networks, *Neurocomputing*, Vol. 71, Issue 4-6, pp. 919-930
- Di Barba, P. & Mognaschi, M.E. (2005). Recent Experiences of Multiobjective Optimisation in Electromagnetics: a Comparison of Methods, *The International Journal for Computation and Mathematics in Electrical and Electronic Engineering (COMPEL)*, vol. 24, no. 3, pp. 921-930, ISSN: 0332-1649
- Elser, V., Rankenburg, I. & Thibault, P. (2007). Searching with iterated maps, *Proceedings of the National Academy of Sciences*, Vol. 104 No. 2, pp. 418-23
- Fieldsend, J.E. & Singh, S. (2005). Pareto evolutionary neural networks, *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 338 - 354
- Fienup, J.R. (1982). Phase retrieval algorithms: a comparison, *Applied Optics*, Vol. 21, pp. 2758-69
- Gottvald, A., Preis, K., Magele, C., Biro, O. & Savini, A. (1992). Global optimization methods for computational electromagnetic, *IEEE Transactions on Magnetics*, Vol. 28 No. 2, pp. 1537 - 1540
- Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation, 2nd edition*, Prentice Hall PTR, ISBN:0132733501, Upper Saddle River, NJ, USA
- Jensen, C.A., Reed, R.D., Marks, R.J., II, El-Sharkawi, M.A., Jae-Byung Jung, Miyamoto, R.T., Anderson, G.M. & Eggen, C.J. (1999). Inversion of feedforward neural networks: algorithms and applications, *Proceedings of the IEEE*, Vol. 87 No. 9, pp. 1536 - 1549
- Konak A., Coit D.W. & Smith A.E. (2006). Multi-Objective Optimization Using Genetic Algorithms: A Tutorial, *Reliability Engineering and System Safety*, Vol. 91, pp. 992-1007

- Magele Ch. (1996). TEAM Benchmark Problem 22, available at: www-igte.tu-graz.ac.at/team
- Portone, A., Salpietro, E., Bottura, L., Bruzzone, P., Fietz, W., Heller, R., Rifflet, J.-M., Lucas, J., Toral, F., Raff, S. & Testoni, P. (2006). Conceptual Design of the 12.5 T Superconducting EFDA Dipole, *IEEE Trans. on Appl. Superc.*, vol. 16, pp. 1312-1315
- Principe J. C., Euliano N. R. & Lefebvre W. C. (2000). *Neural and Adaptive Systems*, J. Wiley & Sons, Inc.
- Rico-Martinez R., Adomaitis R. A. & Kevrekidis I. G. (2000). Noninvertibility in neural networks, *Computers & Chemical Engineering*, Vol. 24, pp. 2417-2433
- Takahashi N., Muramatsu K., Natsumeda M., Ohashi K., Miyata K. & Sayama K. (1996). Solution of problem 25 (Optimization of die press model), *Proceedings of ICEF'96*, pp. 383-386
- Vapnik V. N. (1998). *Statistical Learning Theory*, Wiley, New York
- Wang, L. & Lowther, D.A. (2006). Selection of approximation models for electromagnetic device optimization, *IEEE Transactions on Magnetics*, Vol. 42 No. 4, pp. 1227-30
- Zitzler E. & Thiele L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach, *IEEE Transactions on Evolutionary Computation*, Vol. 3, pp. 257-271

IntechOpen



Search Algorithms and Applications

Edited by Prof. Nashat Mansour

ISBN 978-953-307-156-5

Hard cover, 494 pages

Publisher InTech

Published online 26, April, 2011

Published in print edition April, 2011

Search algorithms aim to find solutions or objects with specified properties and constraints in a large solution search space or among a collection of objects. A solution can be a set of value assignments to variables that will satisfy the constraints or a sub-structure of a given discrete structure. In addition, there are search algorithms, mostly probabilistic, that are designed for the prospective quantum computer. This book demonstrates the wide applicability of search algorithms for the purpose of developing useful and practical solutions to problems that arise in a variety of problem domains. Although it is targeted to a wide group of readers: researchers, graduate students, and practitioners, it does not offer an exhaustive coverage of search algorithms and applications. The chapters are organized into three parts: Population-based and quantum search algorithms, Search algorithms for image and video processing, and Search algorithms for engineering applications.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Sara Carcangiu, Alessandra Fanni and Augusto Montisci (2011). Multi-Objective Optimization Methods Based on Artificial Neural Networks, Search Algorithms and Applications, Prof. Nashat Mansour (Ed.), ISBN: 978-953-307-156-5, InTech, Available from: <http://www.intechopen.com/books/search-algorithms-and-applications/multi-objective-optimization-methods-based-on-artificial-neural-networks>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen