Applied Soft Computing 65 (2018) 1–11

Contents lists available at ScienceDirect

Applied Soft Computing

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

Towards an integrated evolutionary strategy and artificial neural network computational tool for designing photonic coupler devices

Adriano da Silva Ferreira [a,*], Carlos Henrique da Silva Santos [b], Marcos Sergio Gonçalves [c], Hugo Enrique Hernández Figueroa [a]

[a] Department of Communications, School of Electrical and Computer Engineering, University of Campinas, Campinas 13083-852 400, Brazil
[b] São Paulo Federal Institute of Education, Science and Technology, Campus Itapetininga, Itapetininga 18202-000 1561, Brazil
[c] School of Technology, University of Campinas, Limeira 13484-332 1888, Brazil

ARTICLE INFO

Article history:
Received 29 March 2017
Received in revised form 23 October 2017
Accepted 24 December 2017
Available online 30 December 2017

Keywords:
Photonic coupler
Evolutionary algorithm
Artificial neural network
Device optimization

ABSTRACT

Photonics has been widely explored in computing and communications, mainly to rationalize the relationship between device size minimization and data processing/transmission maximization. Generally driven by optimization and modeling techniques, the design of photonic devices is often performed by bio-inspired algorithms integrated to electromagnetic solvers, which have achieved advances but is still time-consuming. As an alternative to a costly finite element method (FEM) solver, a multilayer perceptron (MLP) neural network is proposed for computing power coupling efficiency of photonic couplers, originally designed through an integrated evolutionary strategy (ES) and FEM routine. We address the ES-FEM design of two efficient couplers, present the MLP implementation and the MLP training and testing over the routine generated datasets, and measure MLP and FEM runtime. MLP suitably predicted the power coupling efficiency of a variety of unknown couplers on tests. The measured runtime showed MLP is $\sim 10^5$ faster than FEM. In conclusion, MLP is a potential tool to be integrated to ES on the design of such photonic couplers.

1. Introduction

Photonics has brought cutting-edge technologies for computing and communications, primarily due to its high data processing and transmitting capability through ever-smaller photonic devices. However, as this trend raises the complexity of such devices [1], more sophisticated optimization and modeling techniques are required. This growing demand has led computational electromagnetism work groups to explore parallel algorithms [2,3], novel or improved numerical methods [4,5] and new or adapted optimization procedures [6–10] in order to enable the shape of advanced devices. Likewise, the combination of bio-inspired algorithms and electromagnetic solvers has been used to favorably design complex photonic devices [11–15]. Nevertheless, it has been generally costly due to intensive electromagnetic numerical computations in refined models required by elaborate devices.

In this context, we propose a Multilayer Perceptron (MLP) Artificial Neural Network (ANN) [16] for computing the power coupling efficiency of two sorts of photonic coupler devices, in order to boost the computing performance of the underlying design optimization. Originally, the optimization process has been carried out by an integrated Evolutionary Strategy (ES) [17] and Finite Element Method (FEM) [18] routine, which has designed efficient power coupling photonic couplers. However, it has been computationally expensive due to the intensive FEM's power coupling efficiency calculations (objective function evaluations) of the ES-generated photonic couplers. Thus, an MLP might work as an effective objective function evaluator.

Particularly, we aim at assessing the MLP's power coupling predicting and fast computing capabilities in this work, in order to analyze the feasibility of an integrated ES and MLP tool for optimizing the photonic coupler design. Prior electromagnetism-related works reported the MLP's advantages as a computationally low-cost predictor. However, they explored MLP in distinct applications and measured different electromagnetic quantities, precluding a direct qualitative comparison with our proposal. MLP estimated photonic band-gaps and dispersion relation of bi-dimensional (2D) photonic crystals and reduced the execution time in regard to a Plane Wave Method solver [19]. MLP also forecast promptly and satisfactorily the propagation characteristics of plasmonic transmission lines [20]. Finally, MLP was employed as an objective

* Corresponding author.
E-mail address: adrianof@decom.fee.unicamp.br (A. da Silva Ferreira).

https://doi.org/10.1016/j.asoc.2017.12.043
1568-4946/

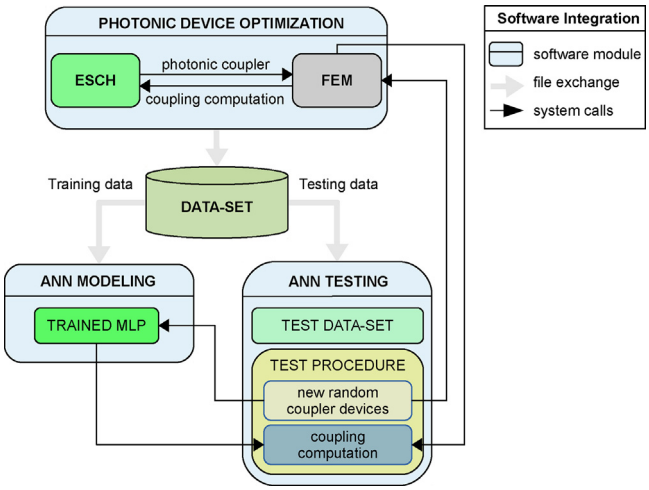


Fig. 1. ESCH, FEM and ANN software integration and work-flow.

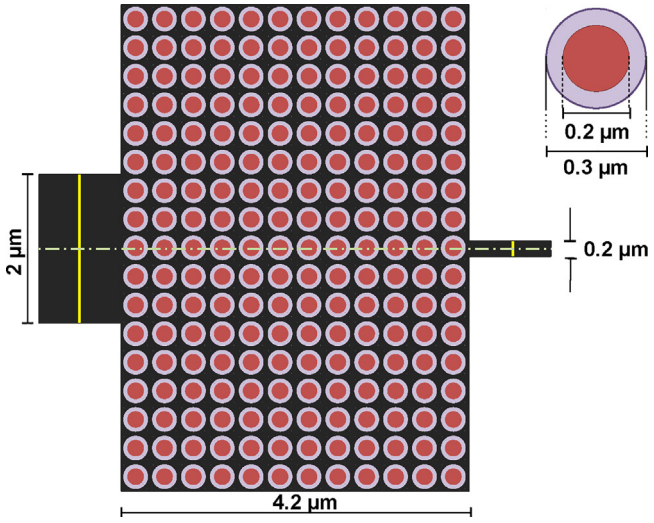


Fig. 2. Proposed micro-to-nano photonic coupler design.

function evaluator for predicting resonance frequency and bandwidth of frequency selective surfaces [21] and effective permittivity of metamaterials [22], decreasing the runtime of the optimization processes in relation to full-wave solvers.

As shown in Fig. 1, our current project has been built by integrating distinct software modules, linked through files or system calls, which provides an automatic way for designing photonic couplers and modeling and testing ANNs. The workflow begins on the *Photonic Device Optimization* module, where an alternative $((\mu/\rho)+\lambda)$ES algorithm (named ESCH on account of its Cauchy-distributed random variable function) is combined with a FEM routine to optimize the photonic coupler design. This ESCH algorithm achieved successful results in telecommunications applications [23], and is available freely online on the MIT NLOpt library, under the global derivative-free algorithms group. The integrated ESCH-FEM routine also stores all ESCH-generated photonic couplers and their respective FEM-calculated, power coupling efficiency values (a percentage between 0% and 100%), providing the means of creating a suitable dataset for training and testing a Multilayer Perceptron (MLP) ANN.

The *ANN Modeling* module provides the functions of reading the generated dataset, modeling the MLP, and exposing an interface for accessing the trained ANN. We have implemented an MLP algorithm due to the flexibility of updating our code in accord with our research purposes.

The ANN model validation is performed on the *ANN Testing* module, which evaluates the trained MLP through a test dataset and an iterative test procedure. The test set contains a collection of unknown samples which are processed and labeled by MLP. The MLP-generated labels are then compared to the respective expected values, in order to assess the quality of MLP's answers. In turn, the iterative test procedure operates changing attributes of an unknown photonic device at random and obtaining the respective FEM and MLP power coupling efficiency computations along the iterations. Through this test procedure, we aim at evaluating iteratively the MLP's outputs along the variation of the FEM's ones for new-created random devices.

We present our work as follows. The second section addresses the works developed in [23,24], describing the photonic coupler design and the optical power equations which provide the means of formulating a 2D Finite Element Method (2D-FEM). The third section presents the ESCH algorithm and summarizes the optimization results of two geometrically identical, material-distinct couplers. The fourth section describes the MLP's concepts and equations which permeate the implemented MLP, and the fifth section presents the MLP's training methodology. The sixth section is dedicated to the results and discussions, where the MLP's answers are assessed through the test dataset and three case studies using the test procedure, and the MLP and FEM runtimes are presented and compared. This section ends this paper by discussing the results of our findings and presenting the conclusions and future works.

2. 2D photonic coupler modeling by finite element method

The proposed photonic device design to be optimized is a micro-to-nano coupler waveguide, as shown in Fig. 2, and was addressed in the works [23,24]. A 2D structure has been considered in order to enable its optimization through a sequential 2D-FEM and to facilitate the validation of our method since the optimization of 3D models would require a long runtime or parallel solutions.

The goal is to maximize the efficiency of the optical power coupling, considering the field of the optical beam should be guided from the 2 µm-width input waveguide, positioned on the left, to the 0.2 µm-width output waveguide, on the right. The output waveguide is ten times lower than the input waveguide, and the distance between them is considerably short (4.2 µm) when compared to inverted tapers and sub-wavelength grating waveguides [25].

Following foundry specifications, the proposed design is based on a rectangular optical coupler geometry filled by material columns (represented by circles in Fig. 2), which are randomly combined to maximize the output power coupling. These columns can assume diameters of either 0.2 µm or 0.3 µm, as well as different materials. A specific configuration of columns' diameters and materials defines one photonic coupler, which is further discretized into a triangular mesh in order to be processed by the 2D-FEM.

The 2D-FEM simulates the coupler's power coupling efficiency through the calculation of the input–output optical power ratio. The simulation occurs in the frequency domain, and the setup considers the fundamental TE_{10} mode with a wavelength of 1.55 µm. The mesh (depicted in Fig. 3) is composed of 36377 triangular elements, generating 72922 unknown variables.

Now, assuming a wave propagating under the fundamental mode in a waveguide, the respective optical power is given by

$$P = \frac{\beta \cdot E_0^2 \cdot w \cdot d_{eff}}{4 \cdot \omega \cdot \mu_0},$$

where β is the constant phase, E_0 the amplitude of the x component of the electric field, w the waveguide depth, ω the angular

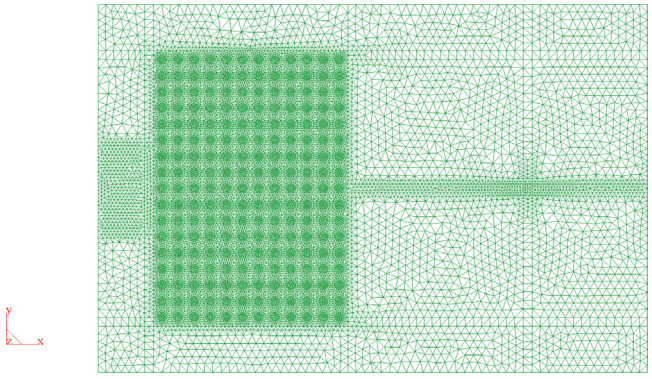


Fig. 3. A triangular mesh of a micro-to-nano photonic coupler.

frequency, μ_0 the magnetic permeability of free space, and d_{eff} the waveguide effective width defined by

$$d_{eff} = d + \frac{2}{\alpha},$$

where d is the waveguide width and $\alpha = \sqrt{\beta \cdot \omega^2 \cdot \mu_0 \cdot \varepsilon_s}$, being ε_s the substrate electric permittivity. Therefore, the input waveguide optical power is given by

$$P_1 = \frac{\beta_1 \cdot E_{01}^2 \cdot w \cdot d_{eff1}}{4 \cdot \omega \cdot \mu_0}, \quad (1)$$

and the output waveguide optical power is given by

$$P_2 = \frac{\beta_2 \cdot E_{02}^2 \cdot w \cdot d_{eff2}}{4 \cdot \omega \cdot \mu_0} \quad (2)$$

To calculate the ratio between the input and output power (Eqs. (1) and (2), respectively), the amplitude E_{01} of the electric field in the waveguide was fixed in 1 (V/m). Thus, the power coupling efficiency of the micro-to-nano photonic coupler is obtained by

$$\frac{P_2}{P_1} = \frac{\beta_2 \cdot E_{02}^2 \cdot d_{eff2}}{\beta_1 \cdot d_{eff1}} \quad (3)$$

A 2D-FEM code was formulated based on (3), providing the means for calculating the power coupling efficiency (objective function to be maximized) of every ESCH-generated photonic coupler, determined by the design presented in Fig. 2. Next, we address the photonic coupler design optimization by means of the ESCH algorithm and present the related results.

3. Adapted evolutionary strategy algorithm: concepts and results

The Evolutionary Strategy Algorithm is a nature-inspired meta-heuristic that considers the parents and off-springs relations and generations [17,26]. Nature inspired meta-heuristics have been broadly explored in many areas of knowledge in order to enhance the final optimization results and therefore provide novel technologies [27], such as these photonic coupler devices.

In this work, the optimization procedure is based on an adapted version of the $((\mu/\rho)+\lambda)$ Evolutionary Strategy Algorithm, where μ and λ are the parents and off-springs population sizes, and ρ is the number of parents recombined to generate an off-spring individual. This version was named ESCH algorithm, once every generated random number follows the Cauchy distribution function. For example, each off-spring is generated by a single-point random combination of two different parents, which are randomly selected by the Cauchy distribution.

Fig. 4 depicts the ESCH work-flow. The procedure starts initializing the parents population size with $\mu = np = 60$, where each

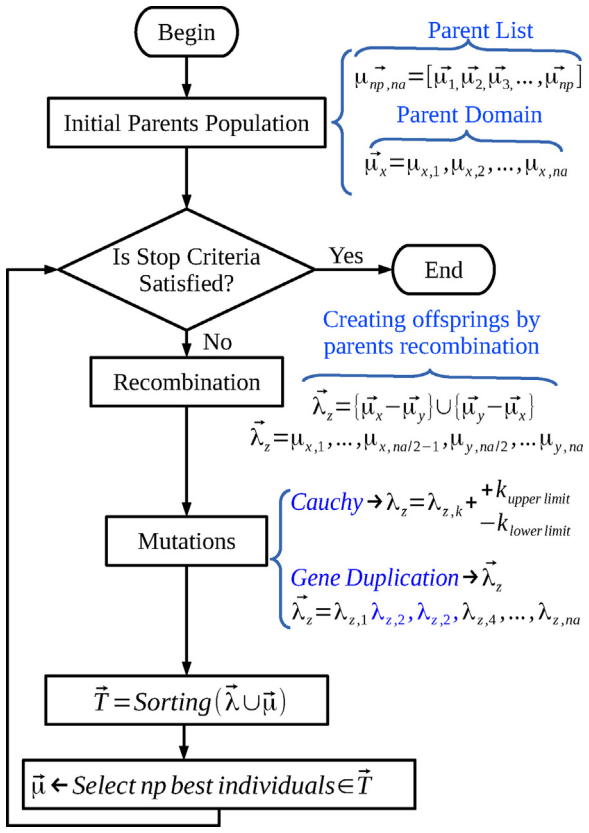


Fig. 4. $\left(\frac{\mu}{\rho}+\lambda\right)$ ES work-flow and its mathematical logical representation.

individual contains *na* attributes. Here, parent and off-spring individuals are represented by the $\vec{\mu}_{i,j}$ and $\vec{\lambda}_{i,j}$ vectors, respectively, being the subscripts *i* and *j* the individual and the attribute indexes. Each individual's attribute is associated to one coupler's column, which might assume different materials.

After creating the initial population, ESCH runs the recombination process, where every off-spring is determined by a single point union of two different parents ($\rho = 2$). The generated individuals are then mutated through two processes. First, a Cauchy random value is added to the individual's attributes, at a 6% rate. Following this, a gene duplication randomly chooses two attributes and replicates the first on the second one, at a 4% rate. The defined off-spring population size is $\lambda = 40$.

Finally, a descending sort function is applied to the parent and off-spring union set in order to generate the *T* list. Then, the first *np* best individuals are selected from this list to create a new parent population for the next generation.

Table 1 lists all photonic coupler materials, determined here by their relative refractive indexes at 1.55 μm wavelength. The lines "*First possible value*" and "*Second possible value*" indicate the materials by which the coupler columns might be made of.

In this work, two different photonic coupler designs were optimized. They are discerned by the columns' possible diameters and

Table 1
Materials' refractive indexes at 1.55 μm wavelength.

Device area	Coupler 1	Coupler 2
Input waveguide	1.45	1.45
Output waveguide	3.5	3.5
Coupler	1.98	1.98
First possible value	1.98	2.4
Second possible value	1.45	1.0

material composition. The first one is referred as Coupler 1 model and was addressed in [24]. The Coupler 1 design can accept both column's diameters, as proposed in Fig. 2. The second is referred as Coupler 2 model and is a new proposal, based on the column's diameter of only 0.3 μm and different column materials.

Column materials are randomly set during the optimization processes, in order to find configurations which could lead to more efficient couplers. Coupler 1 and 2 were optimized following slightly different rules. Assuming only one column diameter, Coupler 2 optimization operates changing the possible column materials. On Coupler 1 optimization, one individual's attribute might assume not only different column materials but also different column diameters. The rule is as follows: internal columns are only considered when they have different materials from external columns, set with the same coupler (rectangle) material. Otherwise, internal columns always assume the external column's materials.

Fig. 5 shows the power coupling efficiency maximization of Coupler 1 and 2 models along the iterations. One may note the Coupler 2 model optimization substantially achieved better results than Coupler 1. The power coupling efficiencies of the best-optimized Coupler 1 and 2 configurations were 73.94% and 85.49%, respectively. Fig. 6 shows the best coupler configurations and the respective simulated 2D and 3D wave scatterings. From these simulations, it is noted the optimized Coupler 2 configuration couples more effectively the electric field and, consequently, the optical power.

A final numerical evaluation of these optimized structures was performed under a bandwidth between 1.35 μm and 1.65 μm, in order to analyze their efficiencies in different wavelengths. The best Coupler 1 configuration was more uniform under the defined bandwidth and more efficient on wavelengths shorter than 1.5 μm, as depicted in Fig. 7. Nevertheless, the optimized Coupler 2 model outperformed Coupler 1 from 1.5 μm wavelength onward, achieving power coupling efficiencies higher than 95%.

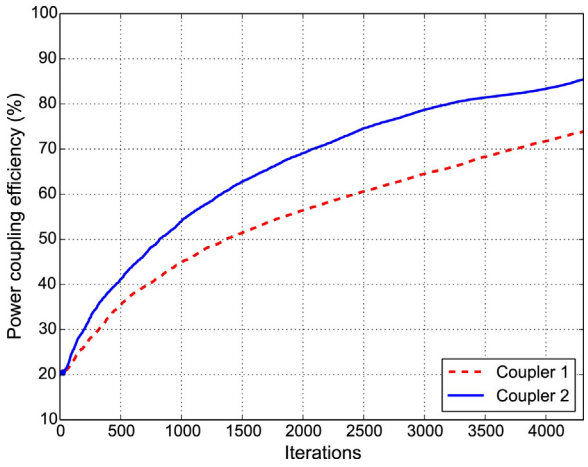


Fig. 5. Convergence of the photonic couplers' optimization processes.

4. Multilayer perceptron: concepts and implemented model

4.1. Brief introduction

Artificial Neural Networks (ANNs) are brain-based machines, designed to model the manner in which the brain executes a specific function or task [16]. In analog to brain – made-up of neu-

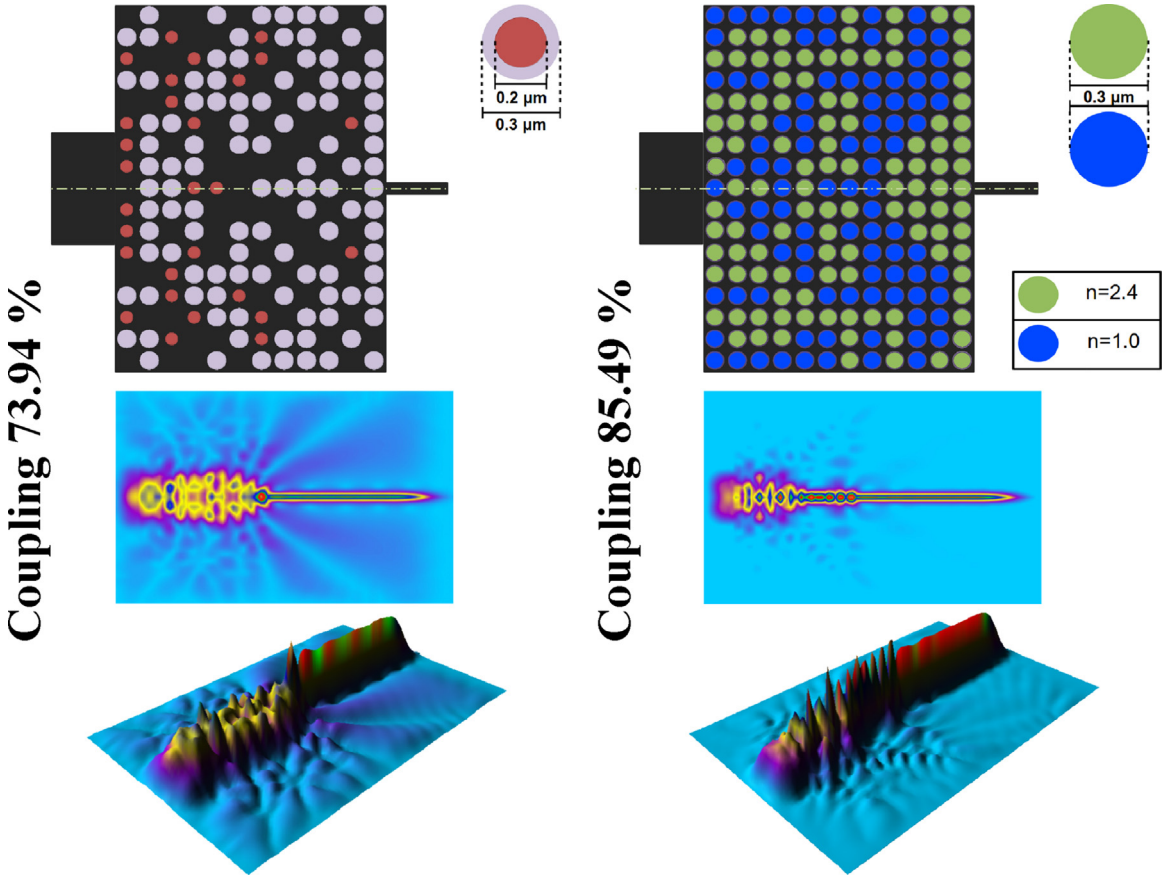


Fig. 6. Optimized photonic coupler devices and the respective electric field profiles.

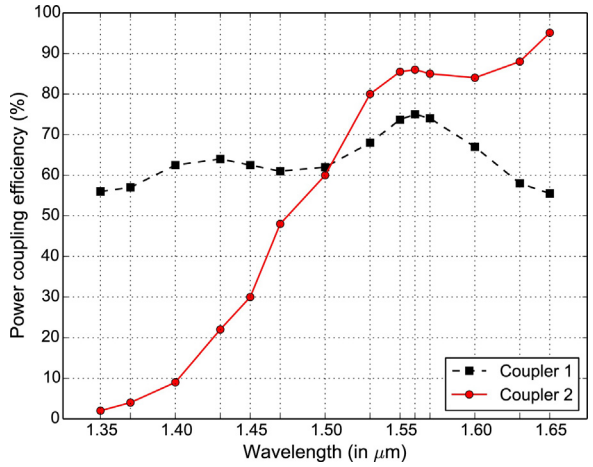


Fig. 7. Optimized photonic couplers' performances under an optical bandwidth.

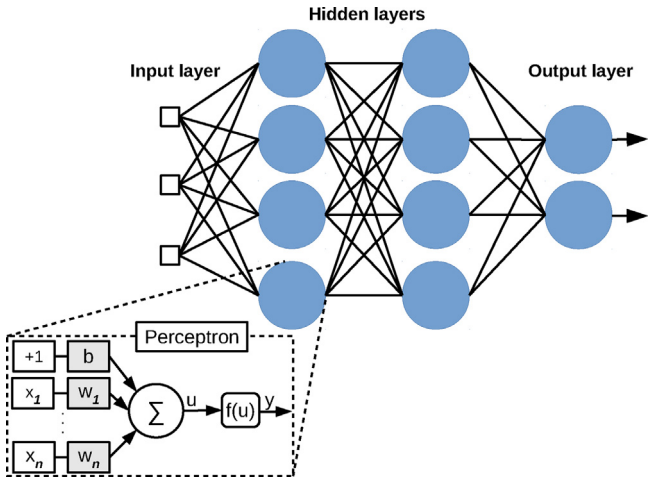


Fig. 8. MLP with two hidden layers and the perceptron model.

rons, which are massively interconnected through the synapses – such machines are composed of a collection of artificial neurons, highly interconnected via synaptic weights, which forms a network structure capable of processing information on a massively parallel-distributed way [28].

An important aspect of the brain is its ability for learning, for which knowledge is acquired, adapted or reinforced from experience and made available for further use. As an information processing system [29], ANNs establish a relation between input–output data, which is completely determined by the configuration of the whole set of synaptic weights. The learning of an ANN is therefore performed by the modification of its synaptic weights [16], so that it may build a suitable relation of input–output information.

In this work, a Multilayer Perceptron (MLP) is employed in order to forecast the power coupling efficiency of the presented photonic coupler devices. MLP is a class of multilayer feed-forward ANN, a type of architecture composed of an input layer (input signal), one or more hidden neuronal layers, and one output neuronal layer. The input signal propagates in a forward direction through the network, where every neuron of a layer receives the signal built by all neurons of the precedent layer and sends its output to all neurons of the following layer. In MLP, the signal is processed by a perceptron, a neural model composed of synaptic weights (one weight for every element of the respective input signal), a summing junction and an activation function. Fig. 8 illustrates a general MLP structure with two hidden layers and the perceptron model.

The output of the kth perceptron that belongs to the mth layer is defined as

$$y_k^m = f\left(\sum_{j=1}^{n} w_{kj}^m y_j^{m-1} + b_k^m\right),$$

where y_j^{m-1} is the jth element of the n-length input signal from the previous layer (the jth element of the input layer's signal is denoted by y_j^0 or x_j), w_{kj}^m is the weight for the jth input signal element, b_k^m is the bias, and y_k^m or $f(\cdot)$ is the activation function, which can be non-linear, characterizing the perceptron – and consequently MLP – as non-linear structures.

4.2. MLP's error signals and cost function

The MLP training process is based on the supervised learning paradigm, which consists in inferring a function from a set of labeled training data. A training dataset $\{(x_i, d_i)\}_{i=1}^{N}$ is thus available, with N input–output pair samples, so that each input pattern x_i has its target output or deserved response d_i. Both variables x_i and d_i can be either scalar or vector quantities.

During the training, MLP processes each x_i, producing an output y_{ik} at the kth output neuron (belonged to the output layer). The difference between d_{ik} and y_{ik} results in the generation of an error signal at the output of the kth output neuron. By adding up the squared error signals of all output neurons, an error energy $E(i)$ can be obtained for the ith training input as

$$E(i) = \sum_{k=1}^{no} (d_{ik} - y_{ik})^2, \tag{4}$$

considering an MLP with no output neurons. By integrating the error energies for all training inputs and normalizing it by N, a mean square error (MSE) can be obtained as a measure of performance for MLP as

$$MSE = J(W) = \frac{1}{N}\sum_{i=1}^{N} E(i), \tag{5}$$

where J is the cost function and W is the whole set of MLP weights. The learning process is therefore accomplished by minimizing (5) through the adjustment to the weights W, in order to obtain an MLP model whose outputs are as close as possible to their respective deserved responses.

4.3. Minimizing the cost function: the back-propagation algorithm

The employed minimization process is performed by an iterative process, which updates the weights W by applying the gradient descent method [30]. As expressed by equation (6), on iteration t, the gradient vector $\nabla J[W(t)]$ is estimated, and a step along its normalized negative value (scaled by the step size α_t and added to the current weights $W(t)$) is taken towards a local minimum of $J(W)$, in order to obtain the new weights $W(t+1)$ of the following iteration.

$$W(t+1) = W(t) - \alpha_t \frac{\nabla J[W(t)]}{\left\|\nabla J[W(t)]\right\|} \tag{6}$$

An iterative correction based on batch mode was adopted for W, on which the entire set of training patterns is presented to MLP before the weights change. An average gradient can be estimated

for (6) by adding the computed gradient of each training pattern and normalizing the sum by N.

The gradient vector is built by means of the back-propagation algorithm [16], which operates propagating the error signals backward (layer by layer), through the whole MLP, measuring the effect of each weight on the current MSE as

$$\frac{\partial J}{\partial w_{kj}} = \delta_{ik} y_{ij}, \tag{7}$$

where J is the cost function and w_{kj} is the jth weight of the neuron k. Considering the training sample i, the influence of w_{kj} on J (first derivative of (7)) is equal to the product between its input element y_{ij} and the sensibility δ_{ik}, that is a common term for all weights of the neuron k. Thus, computing the sensibilities of each neuron is the remaining operation to build the gradient vector, once all input signals have been obtained before on the forward propagation process.

The back-propagation computes the derivative of (7) by applying the chain rule of calculus successively, layer by layer, starting from the output layer M. Considering our MLP model has only one linear output neuron (absence of the activation function), the respective sensibility regarding the ith training pattern can be computed by differentiating (4) with respect to y_{i1}^M as [17]

$$\frac{\partial J}{\partial y_{i1}^M} = \delta_{i1}^M = -2(d_{i1} - y_{i1}^M), \tag{8}$$

where both δ_{i1}^M and y_{i1}^M are the sensibility and the MLP response of the unique output neuron, respectively, and d_{i1} corresponds to the ith target output. An average gradient for the output neuron's weights can be thus obtained by

$$\frac{\partial J}{\partial W^M} = \frac{1}{N}\sum_{i=1}^{N} \delta_{i1}^M (y_i^{M-1})^T, \tag{9}$$

where W^M is a row vector of the output neuron's weights, δ_{i1}^M is the sensibility term of (8), and $(y_i^{M-1})^T$ is the column vector's transpose of the input signal, built by all hidden neurons of the $(M-1)th$ layer for the ith training pattern.

The $(M-1)$th layer's sensibilities are obtained, for the sample x_i, as [17]

$$\delta_i^{M-1} = F'^{M-1}_i(u_i^{M-1})(W^M)^T \delta_{i1}^M, \tag{10}$$

where δ_i^{M-1} is a column vector of the hidden neuron's sensibilities, u_i^{M-1} is a column vector composed of the weighted sum of inputs of every hidden neuron, $F'^{M-1}_i(u^{M-1})$ is a diagonal matrix composed of activation functions' derivatives of the respective hidden neurons, $(W^M)^T$ is the transpose of the output layer's weights, and δ_{i1}^M is the sensibility factor of (8). As the hyperbolic tangent was employed as activation function, the diagonal matrix is given by

$$F'^{M-1}_i(u_i^{M-1}) = \begin{pmatrix} \operatorname{sech}(u_{i1}^{M-1})^2 & 0 & \dots & 0 \\ 0 & \operatorname{sech}(u_{i2}^{M-1})^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \operatorname{sech}(u_{ih}^{M-1})^2 \end{pmatrix},$$

considering an MLP with h hidden neurons on the $(M-1)$th layer. Finally, an average gradient for the weights of the respective neurons is given by

$$\frac{\partial J}{\partial W^{M-1}} = \frac{1}{N}\sum_{i=1}^{N} \delta_i^{M-1} (y_i^{M-2})^T, \tag{11}$$

where W^{M-1} is the matrix of hidden neurons' weights, $(y_i^{M-2})^T$ is the column vector's transpose of the input signal, and δ_i^{M-1} is the sensibility vector of (10).

In addition to an output layer formed by one output linear neuron, the employed MLP model is composed of only one hidden layer. Therefore, the input vector y_i^{M-2} from (11) becomes the input pattern x_i, and the equations from (8) to (11) delineate the back-propagation herein implemented. Finally, the mean gradients obtained on (9) and (11) can be used on (6) in order to update the respective weights for the next iteration.

4.4. MLP's learning rate

The step size α_t is known as the learning rate parameter of the back-propagation algorithm [16], and has to be determined in order to update the weights W in according to (6). This scalar variable represents the step size towards a local minimum of $J(W)$ and plays an important role on the back-propagation algorithm, once small values might lead to long training processes and big ones to solution divergences [16]. The variable α_t is herein determined iteration by iteration through a uni-dimensional searching method, which scales α_t up while the adjustment of W leads to an MSE decrease, and scales α_t down otherwise. We describe the details of the employed uni-dimensional searching method in Section 5.2.

4.5. MLP architecture: aspects of the approximation model

A fundamental aspect of MLP modeling is the definition of the number of hidden neurons and hidden layers that determines a good approximate solution to a particular problem. The MLP approximation capability is intimately linked to its multilayer feed-forward architecture, whenever its activation function is continuous, bounded and non-constant, and a sufficient number of hidden neurons are provided [31]. An MLP – trained in a supervised manner – can be seen as a generic tool for non-linear input–output mappings [17].

The number of hidden neurons directly influences the dimension on which W from (6) are contained in. Therefore, given an appropriate training set to represent a function, a sufficient number of hidden neurons provides a dimension p for W so that its weights can be configured to obtain a good estimator. Formally, let g be a function that maps points from a compact space $X \subset \Re^m$ into points of another compact space $g[X] \subset \Re^n$ as: $g(\cdot) : X \subset \Re^m \to \Re^n$ [32]. A finite training set of N input–output pairs $\{(x_i, d_i)\}_{i=1}^N$ is sampled through a mapping determined by $g(x_i) = d_i$, $i = 1, \dots, N$, $\forall x_i \in X$. Given a weights vector $W \subset \Re^p$ (p is finite) and an approximation model $\hat{g}(\cdot, W) : X \times \Re^p \to \Re^n$, determine the weights $W^* \subset \Re^p$ for which $|g(\cdot) - \hat{g}(\cdot, W^*)| < \varepsilon$, so that $\varepsilon > 0$ should be sufficiently small in order to indicate a reasonable approximation between the functions g and $\hat{g}$. The MSE defined in (5) can provide an average distance measure between g and $\hat{g}$.

From the architecture viewpoint, the MLP mapping capability is enhanced by increasing the number of hidden neurons. Nevertheless, a sufficient number of hidden neurons are required in order to find a viable approximation model $\hat{g}$. Considering a test set $\{(x_j, d_j)\}_{j=N+1}^M$, where $d_j = g(x_j)$, $j = N+1, \dots, M$, $\forall x_j \in X$, an MLP model $\hat{g}$ is said to generalize well if it provides estimates $\hat{d}_j = \hat{g}(x_j)$, $j = N+1, \dots, M$, $\forall x_j \in X$, so that $|d_j - \hat{d}_j| < \varepsilon$. It is important to highlight that a suitable set $W^* \subset \Re^p$ will be obtained by means of an appropriate network architecture. It includes a suitable definition of the number of hidden layers, the number of hidden neurons, and the activation function.

5. MLP training methodology

5.1. Dataset

The databases of Coupler 1 and Coupler 2 models, each one generated by two separated evolutionary optimization processes, are suitable for the MLP supervised learning process. The databases make samples of labeled data available by providing sets of photonic coupler devices, which form the input patterns, and the respective power coupling efficiencies, which determine the output targets.

The couplers are defined by the relative refractive indexes of their material column configurations on the databases. Couplers' columns diameters were not taken into account (recalling Coupler 2 model assumed diameters of only 0.3 μm) in order to reduce the dimensionality of our data (column data might be considered in future tests). According to Table 1, the relative refractive indexes which Coupler 1 and Coupler 2's columns may assume are 1.98 or 1.45 and 2.4 or 1.0, respectively, at wavelength of 1.55 μm. The databases also provide the power coupling efficiencies of both couplers, which are distributed within specific intervals of values. Table 2 summarizes the lower and upper bound values of the available power coupling efficiencies and the databases' sizes.

In addition to their different sizes, the databases also provide more samples of devices with higher power coupling efficiencies. Considering an arrangement of the databases' examples in groups of ten, one may note an unequal distribution of Coupler 1 and Coupler 2 samples, as illustrated in Fig. 9.

The MLP dataset is then generated by reading the databases elements, without normalizing the data. As the coupler model is made of 204 material columns (Fig. 2), an MLP input pattern is a vector $x_i \in \Re^{204}$ which stores 204 relative refractive indexes. The power coupling efficiency percentages, expressed by scalar numbers, form the deserved values $d_i \in \Re^1$. We highlight there are no identical deserved values on the dataset, though some power coupling efficiency values are approximate.

In spite of performing the optimization processes individually for each sort of photonic coupler, we have merged the databases in a unique dataset, once both kinds of couplers are geometrically identical. This strategy has been chosen for two main reasons, considering that MLP learns from data: availability of additional points in another region of the couplings as a function of the relative refractive indexes (when compared to the quantity and variability of points that separated datasets could alone provide); and the possibility of modeling a unique and generalized MLP capable of processing both sorts of couplers.

Before training MLP, the whole dataset is split into 3 different subsets: training, validation and test datasets. MLP is trained through the training dataset. The validation and test datasets are never used for training purposes. The validation dataset is employed here to avoid MLP over-fitting [16] under the training samples and, consequently, to determine the most generalized MLP model (we address how we have employed the validation set during the training next sub-section). The test set is used after the training and plays a key role in the MLP performance evaluation by providing unknown samples for further testing and validation of the trained MLP.

Table 2
Databases' sizes and range of power coupling efficiencies (%).

Databases' features	Coupler 1	Coupler 2
Number of elements	1813	4328
Minimum coupling efficiency (%)	20.05%	20.16%
Maximum coupling efficiency (%)	73.94%	85.49%

Table 3
Training, validation, test and entire datasets' sizes.

Dataset	Size (%)	Number of elements
Training	76.5%	4698
Validation	13.5%	829
Test	10%	614
Entire	100%	6141

Table 4
MLP training parameters and respective values.

Parameters	Values
Initial weight range (uniformly distributed)	(−0.1, 0.1)
Maximum number of iterations	50,000
Minimum training MSE	5.0
Maximum number of continuous rise of validation MSE	6
Monitoring period of validation MSE (in iterations)	10

The whole dataset has been shuffled before being firstly split in training and test datasets. An amount of samples is randomly chosen from the training set to then form the validation dataset. The defined sizes and number of elements of each dataset are listed in Table 3.

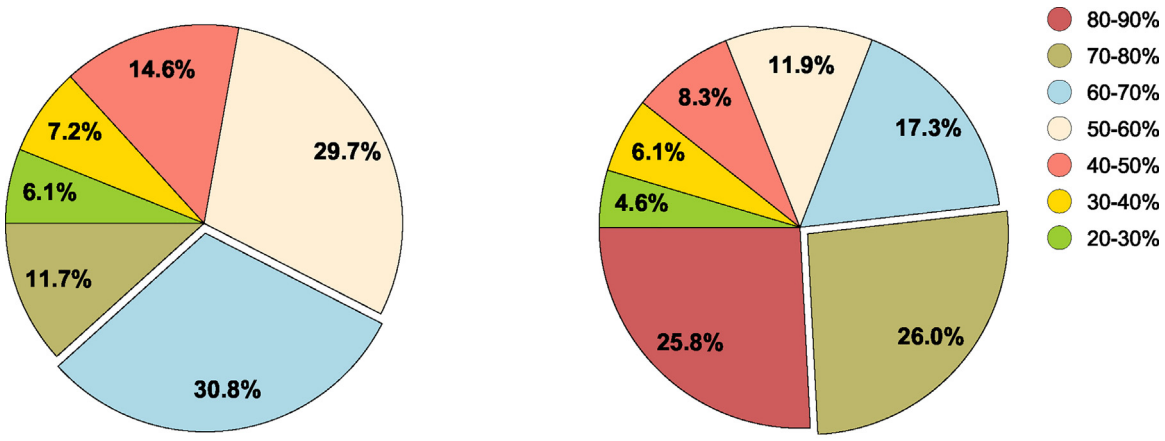
5.2. MLP training method

The training process has been performed on an MLP architecture composed of 150 hidden neurons with hyperbolic tangent transfer functions, and one linear output neuron. We must highlight it might not be the best architecture, once it has not been determined by a systematic process like cross-validation [33], although trial and error tests (gradual increment of hidden neurons) showed this chosen architecture achieved reasonable performances.

As the training has a finite number of iterations, we have defined three distinct stop criteria for ending the MLP training: maximum number of iterations, minimum training MSE, and generalization under validation dataset. The first two are scalar parameters which, once reached, finish the training. The latter operates monitoring periodically the validation MSE during the training to check whether or not it is rising. A continuous increasing of the validation MSE along with a decreasing of the training MSE is considered over-fitting under training data, once MLP is apparently fitting well on these data but not on the validation ones. On this occasion, the training stops and the weight set which has reached the least validation MSE is considered the fittest one, determining the most generalized MLP. Table 4 lists the training parameters and the respective defined values.

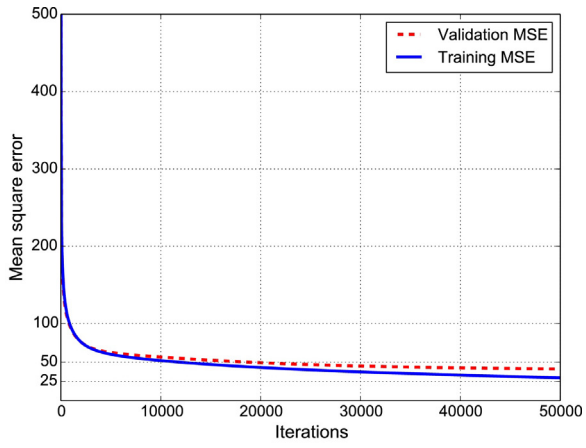
During each iteration of the training, the scalar variable α_t is determined by a uni-dimensional searching method. It uses (6) to search for α_t values able to converge the training faster – by scaling the current gradients up and taking bigger steps towards a local minimum – or to keep the training MSE falling when it is reaching a steady state – by scaling the current gradients down and correcting the current weights smoothly. The variable α_t starts with a defined value at the beginning of the training. At each iteration, the method checks if the current training MSE is lower than the previous one. If yes, it scales α_t up (respecting an upper threshold); otherwise, it scales α_t down until reaching a lower training MSE (or a maximum number of trials). Table 5 lists the defined scales and thresholds of the uni-dimensional searching method.

Fig. 10(a) and (b) illustrate, respectively, the MSE behavior under the training and validation datasets and the learning rate α_t along the training. During the first 5000 iterations, one may note the highest α_t values leading to sharp MSE falls on both datasets since α_t scales the gradients of (6) up and make them reach a local minimum of MSE faster. Also, the approximate minimization of both MSEs

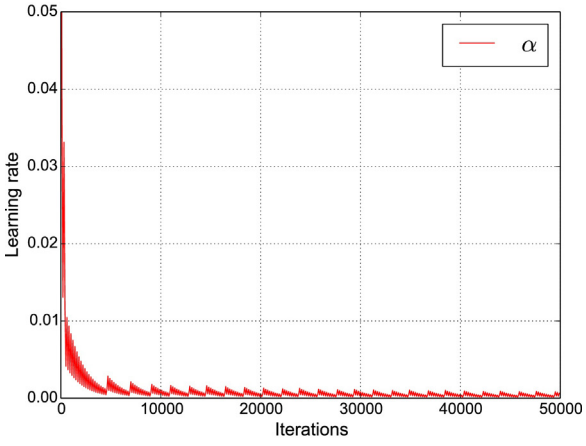


(a) Coupler 1 models (b) Coupler 2 models

Fig. 9. Available samples arranged by tens. The slices of the biggest groups are highlighted.



(a) MSE of training and validation sets



(b) Learning rate α_t

Fig. 10. Training MSE, validation MSE and learning rate during MLP training.

Table 5
Uni-dimensional searching method parameters.

Parameters	Values
Initial α_t value	0.05
Maximum α_t value	0.5
α_t scaled up by	1.05
α_t scaled down by	0.618
Number of trials for scaling α_t down	5

indicates MLP suitably fit the validation data and did not present over-fitting along the training. The reached stop criterion was the maximum number of iterations. At the end, the training and validation MSE values were about 29.6% and 41.0% (a root-mean-square error – RMSE – of about 5.44% and 6.4%, respectively), whereas the α_t value was $\sim 1.29 \times 10^{-4}$.

6. Results and discussion

In this section, we validate the trained MLP by testing its power coupling efficiency answers on unknown photonic couplers and present the computing performance of MLP and 2D-FEM. The tests are performed in the test dataset and the iterative test procedure, where three unknown devices are assessed: a low- and a high-efficiency Coupler 1 models, and a high-efficiency Coupler 2 model. Finally, we discuss the results of our findings, along with the conclusions and future works.

6.1. MLP computation on test dataset

Fig. 11 depicts the MLP power coupling efficiency results for test dataset. Firstly, we have sorted the deserved values in ascending order. One may note MLP results have successfully followed the 2D-FEM ones' ascending curve, clustering its outputs around the 2D-FEM ones. Also, MLP has shown more accurate results for high coupling efficiency photonic couplers. The main hypothesis lies in the larger availability of samples of these devices (Fig. 9), leading MLP to shape its model more precisely in these examples. The test MSE was about 49.82% (RMSE of 7.0583%).

On the other hand, some MLP outputs have considerably deviated from the ascending curve. However, we have not applied outlier detection methods for high dimension [34], nor feature selection methods [35] to either know the distribution of our hyper-dimensional data and (possibly) eliminate outliers or select a subset

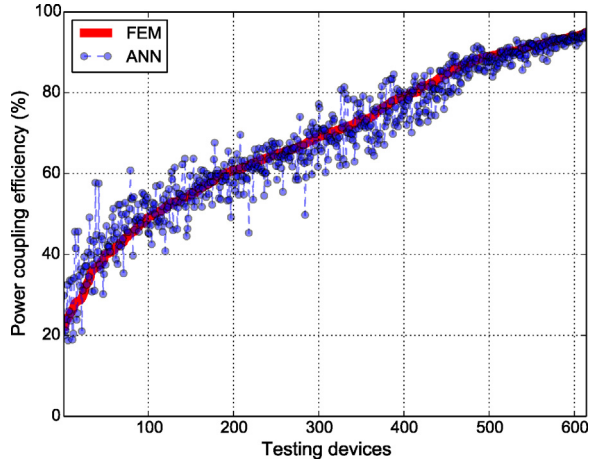


Fig. 11. MLP power coupling efficiency results for test data.

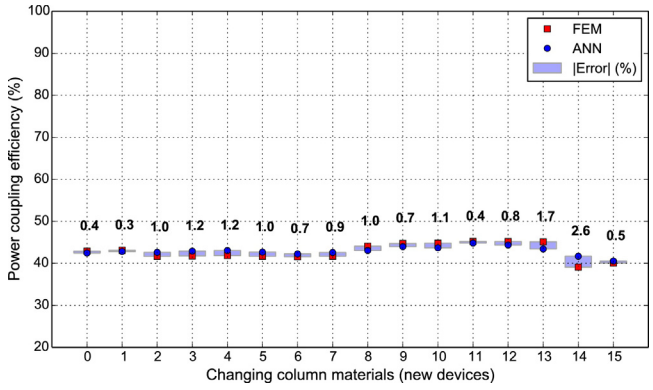


Fig. 12. FEM and ANN coupling results for new varieties of a 42.90% test Coupler 1 model.

which could simultaneously lead to more accurate MLP models and avoid the curse of dimensionality [36].

6.2. MLP computation on new random photonic couplers

Next, we run the iterative test procedure for Coupler 1 and Coupler 2 models. It works as follows. An unknown coupler is chosen from the test dataset. A column is randomly chosen and its relative refractive index (material) is then changed. For Coupler 1 models, a 1.45 relative refractive index is replaced by 1.98, and vice versa; for Coupler 2 ones, 2.4 is replaced by 1.0, and vice versa. The procedure checks if this change has generated a new device and then calculates its power coupling efficiency through the 2D-FEM and the trained MLP. From this new device, a new column is chosen at random and the process is repeated again. Each new device accumulates the prior material changes. Altogether, the procedure is executed 15 times, producing 15 new couplers.

We first show the results of the test procedure on a low power coupling efficiency photonic device. Low-efficiency couplers are less available, and MLP computed less accurately their power coupling efficiencies, as shown in Figs. 9 and 11. Thus, it is reasonable to validate MLP on these sorts of devices.

Fig. 12 shows the MLP and FEM's power coupling efficiency calculations for variations of an unknown Coupler 1 model, which couples about 42.90% of the optical power. The *x*-axis values indicate the number of replaced materials or new devices, starting from the 0th coupler (original) and finishing on the 15th coupler. The absolute error bar represents the modulus of the difference

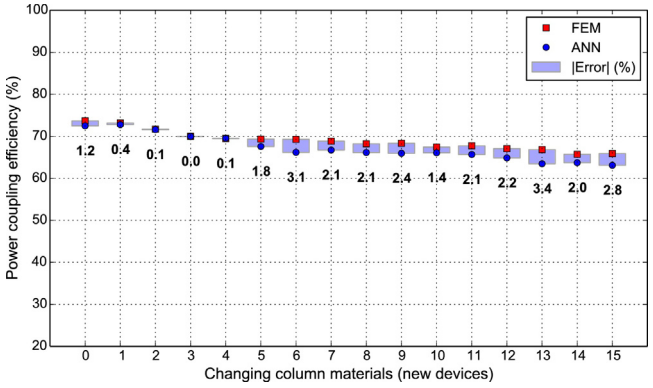


Fig. 13. FEM and ANN coupling results for new varieties of a 73.68% test Coupler 1 model.

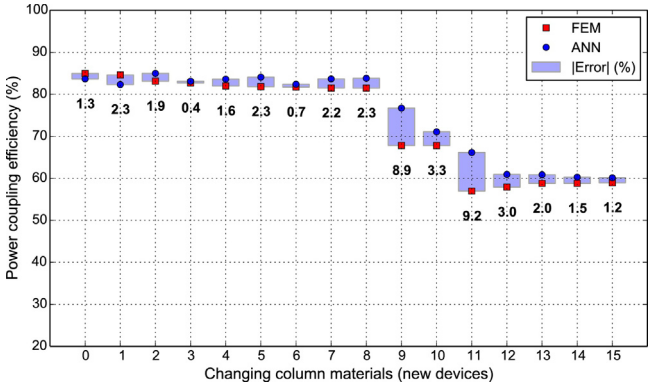


Fig. 14. FEM and ANN coupling results for new varieties of a 84.97% test Coupler 2 model.

between MLP and FEM outputs, quantified by the numbers above or below.

Observing the FEM and MLP's computations on this test, one may note MLP has succeeded in calculating the new random devices' power coupling efficiencies. Among the deserved values (FEM outputs), the results varied up to 6.03%, which in this case occurred between the 13th and 14th couplers (respectively, the highest, 45.12%, and the lowest, 39.09%, coupling efficiencies). The biggest absolute error was 2.6%, which is less than the test RMSE (∼7.06%). The MLP mean absolute error was 1.0% among the new random couplers. Altogether, MLP produced very close results when compared to the FEM ones.

Fig. 13 shows the MLP and FEM's computations for variations of an unknown high-efficiency Coupler 1 model, which couples about 73.68% of the optical power. In this case, the FEM's results varied up to 7.59% (the highest and lowest power coupling efficiencies were 73.68% and 66.09%, belonged to the 0th and 15th couplers, respectively). Observing both 2D-FEM and MLP's output curves, one may note a slight absolute error increase from the 5th coupler, although the respective absolute errors were considerably smaller than the test RMSE. The MLP mean absolute error among the new random couplers was about 1.73%. In general, MLP successfully computed the power coupling efficiencies of the new random devices on this test.

Finally, Fig. 14 shows the results of the test procedure for variations of an unknown very high-efficiency Coupler 2 model, where 84.97% of the optical power is coupled. This test produced the highest power coupling efficiency variation among the deserved values, which was about 28% (the highest and lowest power coupling efficiencies were 84.97% and 56.97% on the 0th and 11th couplers, respectively). Between the 8th and 11th couplers, the

Table 6
MLP and FEM mean elapsed real time.

Process	Time
MLP training (min)	28
MLP power coupling efficiency computation (s)	2.64×10^{-4}
FEM power coupling efficiency computation (s)	87

FEM results varied about 24.51%, showing a strong relation of some specific column materials on the power coupling capability of a photonic coupler. The MLP responses varied considerably on the same interval, but not as sharply as the FEM ones. It resulted in absolute errors slightly bigger than the test RMSE (8.9% and 9.2% on the 8th and 11th couplers, respectively). However, MLP produced very approximate outputs for the other couplers, including those ones created after the 11th coupler, which accumulated all prior material changes that culminated in a accentuated power coupling efficiency fall. The MLP mean absolute error was 2.85% on this test. Altogether, we conclude MLP successfully computed the power coupling efficiencies of these new random devices.

6.3. FEM and MLP computing performance

Table 6 lists the mean elapsed real time of MLP and 2D-FEM for calculating the power coupling efficiency of a unique photonic device. The MLP training time is also considered and listed. The measured times correspond to average times of 5 calls for each process. The computational platform used was a laptop Intel Core i-5 2410M (2.3 GHz, 3 MB L3 cache), DDR3 RAM (6 GB, 1333 MHz), with Linux Ubuntu. MLP and 2D-FEM were coded through MATLAB and Fortran 90, respectively, and are based on serial algorithms. The Linux's *time* function has been used for timing statistics.

From Table 6, one may note the MLP training is a time-consuming process. The main reasons are due to the computation involving a large input pattern matrix, a considerable number of hidden neurons, a high number of iterations, and the efforts for processing the uni-dimensional searching method at each iteration. During the MLP training time, 2D-FEM could compute the power coupling efficiency of approximately 19 photonic couplers.

On the other hand, MLP is five orders of magnitude faster than 2D-FEM in calculating the power coupling efficiency of one photonic coupler. During the 87-s 2D-FEM process, MLP could roughly compute the power coupling efficiency of 329545 photonic coupler devices. Thus, the costly MLP training is compensated by the huge MLP out-performance of 2D-FEM on the power coupling efficiency computation of these kinds of photonic coupler devices.

6.4. Discussion and conclusions

By processing 659 unknown photonic couplers on tests, MLP demonstrated an assured predicting performance, once a unique ANN model was able to satisfactorily compute power coupling efficiency of two different couplers. Also, despite the simplicity of the iterative test procedure, one could analyze the MLP answer capability in a process akin to the ES method (where some random attribute changes generate new individuals) and evidence the MLP has potential as an objective function evaluator. The drawback of the obtained MLP is the answer imprecision primarily for some low-efficiency devices, which may require more training and methods to enhance the MLP accuracy.

On the other hand, MLP demonstrated its greatest advantage by greatly reducing the elapsed time for processing a photonic coupler regarding 2D-FEM. This fast computing capability may pave the way for a wider exploration of the search space, enabling the search for more efficient devices. Particularly, it will be highly effective for optimal broadband Coupler 1 and 2 models, for which the optimization processes will require a large number of objective function evaluations since the analyses are performed in different wavelengths. Also, MLP can be used for an in-depth analysis of the influence of coupler column materials on the power coupling efficiency, contributing to a better grasp of the electromagnetic phenomena inside the coupler, and providing means of evaluating the quality of the proposed photonic device design. Furthermore, once defined MLP as a fast power coupling efficiency estimator, its usage could be extended to 3D problems whose optimization processes are mostly time-critical and computationally costly. In conclusion, MLP is a promising tool to be combined with ESCH in the design of more efficient couplers. Certainly, these optimal devices will effectively contribute to areas such as integrated photonics.

In the future, we consider feature selection methods might determine more accurate MLP models (mainly for low-efficiency devices), by building relevant-feature subsets and reducing data dimension, therefore contributing to enhancing the MLP generalization capability. Feature selection could also diminish MLP data dependence on numerical solvers by providing fewer but appropriate data. Additionally, cross-validation strategies might also provide more precise MLP models by obtaining improved MLP architectures. Despite demanding additional computational effort, these methods will potentially benefit the definition of a stand-alone MLP objective function evaluator, and finally, enable the implementation and use of the ESCH-MLP computational tool.

Acknowledgements

This work was supported by the Coordination for the Improvement of Higher Education Personnel (CAPES), São Paulo Federal Institute of Education, Science and Technology (IFSP) PRP-226/2016, and State of São Paulo Research Foundation (FAPESP) (grant 2012/14553-9). We thank Mr. Christian Edward Harryman for providing language help.

References

[1] K. Aydin, Integrated optics: nanostructured silicon success, Nat. Photon. 9 (6) (2015) 353–355, http://dx.doi.org/10.1038/nphoton.2015.89.
[2] M.S. Goncalves, C. Silva-Santos, H.E. Hernández-Figueroa, A.C. Bordonalli, Parallel three-dimensional full-time domain applied to photonic structures, IET Optoelectron. 5 (1) (2011) 40–45, http://dx.doi.org/10.1049/iet-opt.2009.0086.
[3] T.W.H. Sheu, S.Z. Wang, J.H. Li, M.R. Smith, Simulation of Maxwell's equations on GPU using a high-order error-minimized scheme, Commun. Comput. Phys. 21 (4) (2017) 1039–1064, http://dx.doi.org/10.4208/cicp.OA-2016-0079.
[4] K. Claudio, C.H. Silva-Santos, M.S. Gonçalves, H.E. Hernández-Figueroa, Finite element method to improve the computational performance in optical devices analysis, Microw. Opt. Technol. Lett. 57 (6) (2015) 1423–1426, http://dx.doi.org/10.1002/mop.29103.
[5] T. Fujisawa, T. Sato, K. Saitoh, Full-vector finite-element beam propagation method for helicoidal waveguides and its application to twisted photonic crystal fibers, J. Lightw. Technol. 35 (14) (2017) 2894–2901, http://dx.doi.org/10.1109/JLT.2017.2703889.
[6] J.S. Jensen, O. Sigmund, Topology optimization for nano-photonics, Laser Photon. Rev. 5 (2) (2011) 308–321, http://dx.doi.org/10.1002/lpor.201000014.
[7] A.Y. Piggott, J. Lu, K.G. Lagoudakis, J. Petykiewicz, T.M. Babinec, J. Vučković, Inverse design and demonstration of a compact and broadband on-chip wavelength demultiplexer, Nat. Photon. 9 (6) (2015) 374–377, http://dx.doi.org/10.1038/nphoton.2015.69.
[8] S. Koziel, A. Bekasiewicz, Multi-objective optimization of expensive electromagnetic simulation models, Appl. Soft Comput. 47 (2016) 332–342, http://dx.doi.org/10.1016/j.asoc.2016.05.033.
[9] K. Xu, L. Liu, X. Wen, W. Sun, N. Zhang, N. Yi, S. Sun, S. Xiao, Q. Song, Integrated photonic power divider with arbitrary power ratios, Opt. Lett. 42 (4) (2017) 855–858, http://dx.doi.org/10.1364/OL.42.000855.
[10] A.Y. Piggott, J. Petykiewicz, L. Su, J. Vučković, Fabrication-constrained nanophotonic inverse design, Sci. Rep. 7 (2017) 1786, http://dx.doi.org/10.1038/s41598-017-01939-2.
[11] G.N. Malheiros-Silveira, V.F. Rodriguez-Esquerre, H.E. Hernandez-Figueroa, Strategy of search and refinement by GA in 2-D photonic crystals with absolute PBG, IEEE J. Quantum Electron. 47 (4) (2011) 431–438, http://dx.doi.org/10.1109/JQE.2010.2091107.
[12] A.d.S. Ferreira, G.N. Malheiros-Silveira, H.E. Hernandez-Figueroa, Maximizing complete band gaps in two-dimensional photonic crystals by using artificial

immune network, in: 2015 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC), IEEE, 2015, pp. 1–4, http://dx.doi.org/10.1109/IMOC.2015.7369172.
[13] C. Wang, S. Yu, W. Chen, C. Sun, Highly efficient light-trapping structure design inspired by natural evolution, Sci. Rep. 3 (2013) 1025, http://dx.doi.org/10.1038/srep01025.
[14] J. Zhang, Evolutionary optimization of compact dielectric lens for farfield sub-wavelength imaging, Sci. Rep. 5 (2015) 10083, http://dx.doi.org/10.1038/srep10083.
[15] T. Feichtner, O. Selig, B. Hecht, Plasmonic nanoantenna design and fabrication based on evolutionary optimization, Opt. Express 25 (10) (2017) 10828–10842, http://dx.doi.org/10.1364/OE.25.010828.
[16] S.S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, 2001.
[17] L.N. De Castro, Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications, CRC Press, 2006.
[18] J.-M. Jin, The Finite Element Method in Electromagnetics, John Wiley & Sons, 2015.
[19] G.N. Malheiros-Silveira, H.E. Hernandez-Figueroa, Prediction of dispersion relation and PBGs in 2-D PCs by using artificial neural networks, IEEE Photon. Technol. Lett. 24 (20) (2012) 1799–1801, http://dx.doi.org/10.1109/LPT.2012.2215846.
[20] R.R. Andrawis, M.A. Swillam, M.A. El-Gamal, E.A. Soliman, Artificial neural network modeling of plasmonic transmission lines, Appl. Opt. 55 (10) (2016) 2780–2790, http://dx.doi.org/10.1364/AO.55.002780.
[21] P.H.da.F. Silva, R.M.S. Cruz, A.G. d'Assuncao, Blending PSO and ANN for optimal design of FSS filters with Koch island patch elements, IEEE Trans. Magn. 46 (8) (2010) 3010–3013, http://dx.doi.org/10.1109/TMAG.2010.2044147.
[22] D.R. Luna, C.F.L. Vasconcelos, R.M.S. Cruz, Using natural optimization algorithms and artificial neural networks in the design of effective permittivity of metamaterials, in: 2013 SBMO/IEEE MTT-S International Microwave Optoelectronics Conference (IMOC), IEEE, 2013, pp. 1–4, http://dx.doi.org/10.1109/IMOC.2013.6646572.
[23] C.H.d.S. Santos, M.S. Goncalves, H.E. Hernandez-Figueroa, Designing novel photonic devices by bio-inspired computing, IEEE Photon. Technol. Lett. 22 (15) (2010) 1177–1179, http://dx.doi.org/10.1109/LPT.2010.2051222.
[24] C.H. da Silva Santos, M.S. Goncalves, H.E.H. Figueroa, Evolutionary strategy algorithm applied to optimize micro-to-nano coupler devices, in: 2011 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC), IEEE, 2011, pp. 665–668, http://dx.doi.org/10.1109/IMOC.2011.6169290.
[25] B. Liu, Y. Zhang, Y. He, X. Jiang, J. Peng, C. Qiu, Y. Su, Silicon photonic bandpass filter based on apodized subwavelength grating with high suppression ratio and short coupling length, Opt. Express 25 (10) (2017) 11359–11364, http://dx.doi.org/10.1364/OE.25.011359.
[26] L.N. de Castro, Fundamentals of natural computing: an overview, Phys. Life Rev. 4 (1) (2007) 1–36, http://dx.doi.org/10.1016/j.plrev.2006.10.002.
[27] L.N. de Castro, R.D. Maia, A. Szabo, R.S. Xavier, R. Pasti, D.G. Ferrari, The Grand Challenges in Natural Computing Research: The Quest for a New Science, IGI Global, 2014, pp. 237–250, http://dx.doi.org/10.4018/978-1-4666-4253-9.ch016, Chapter 16.
[28] S.S. Haykin, Neural Networks and Learning Machines, Pearson, 2008.
[29] R. Rojas, Neural Networks: A Systematic Introduction, Springer Science & Business Media, 2013.
[30] R. Battiti, First- and second-order methods for learning: between steepest descent and Newton's method, Neural Comput. 4 (2) (1992) 141–166, http://dx.doi.org/10.1162/neco.1992.4.2.141.
[31] K. Hornik, Approximation capabilities of multilayer feedforward networks, Neural Netw. 4 (2) (1991) 251–257, http://dx.doi.org/10.1016/0893-6080(91)90009-T.
[32] F.J.V. Zuben, M.L. de Andrade Netto, Projection pursuit and the solvability condition applied to constructive learning, International Conference on Neural Networks, vol. 2 (1997) 1062–1067, http://dx.doi.org/10.1109/ICNN.1997.616175.
[33] T. Andersen, T. Martinez, Cross validation and MLP architecture selection, International Joint Conference on Neural Networks (IJCNN '99), vol. 3 (1999) 1614–1619, http://dx.doi.org/10.1109/IJCNN.1999.832613.
[34] C.C. Aggarwal, P.S. Yu, Outlier detection for high dimensional data, SIGMOD Rec. 30 (2) (2001) 37–46, http://dx.doi.org/10.1145/375663.375668.
[35] W. Duch, Filter Methods, Springer Berlin Heidelberg, 2006, pp. 89–117, http://dx.doi.org/10.1007/978-3-540-35488-8, Chapter 3.
[36] M. Verleysen, D. Francois, G. Simon, V. Wertz, On the effects of dimensionality on data analysis with neural networks, Artif. Neural Nets Probl. Solving Methods 2687 (2003) 105–112, http://dx.doi.org/10.1007/3-540-44869-1_14.

Adriano da Silva Ferreira received the B.Sc. degree in information technology and the M.Sc. in electrical engineering from University of Campinas (UNICAMP) in 2009 and 2013, respectively. He is currently working towards the Ph.D. degree in electrical engineering at the Department of Communications, UNICAMP. His research interests include natural computing, electromagnetic simulators, and software engineering.

Carlos Henrique da Silva Santos, (M'06) received his B.Sc. degree in information technology, and the M.Sc. and Ph.D. in electrical engineering from University of Campinas (UNICAMP) in 2003, 2005 and 2010, respectively. Carlos was in research internships at Mathematics Department from Massachusetts Institute Technology (MIT) (2012), Cornell University (2010) and Technische Universität München (TUM) (2009). He is currently an assistant professor at IFSP, campus Itapetininga, and coordinates a Specialization Program in Information and Communication Applied in Educational Environments. Professor Carlos is member of the IEEE Education Society and Brazilian Microwave and Optoelectronics Society (SBMO).

Marcos S. Gonçalves received the B.Sc. degree in electrical engineering from the National Institute of Telecommunication, Santa Rita do Sapucai, in 1998, Brazil, M.Sc., and Ph.D. in electrical engineering from the University of Campinas in 2002, and 2007, respectively, Brazil. From 2007 to 2009, he was with School of Electrical and Computer Engineering of the University of Campinas as a researcher. Since 2009, he has been a professor with the School of Technology of the University of Campinas. His research interests include computational physics, computer-aided engineering, and photonic and microwave devices.

Hugo E. Hernandez-Figueroa received his Ph.D. degree in physics from the Imperial College of Science, Technology and Medicine, London, UK, in 1992. Full Professor at the University of Campinas (UNICAMP), School of Electrical and Computer Engineering, since 2005. He has published over 110 journal papers, 230 international conference papers and 10 patents. He is Co-Editor of the books Localized Waves: Theory and Applications (Wiley and Sons, 2008) and Non-Diffracting Waves (Wiley-VCH, 2013). Fellow of the Optical Society of America class 2011. His research interests concentrate on electromagnetic phenomena applied to integrated photonics, biosensors, nanophotonics, optical fibers, metamaterials, plasmonics, and antennas.