Programmable Integrated Silicon Photonics Waveguide Meshes: Optimized Designs and Control Algorithms

Daniel Pérez López, *Member, IEEE*

(Invited Paper)

***Abstract*—Programmable Integrated Photonics is a recent area of research that aims to integrate a very-large scale of reconfigurable photonic components to enable flexible and versatile photonic integrated circuits. In this paper, we review the state of the art of general-purpose waveguide mesh arrangements with a special focus on those that allow the synthesis of optical feedback loops. Moreover, we propose for the first time, a new design approach to generate waveguide mesh patterns with equally-oriented components. This innovation is of special relevance to improve performance and to mitigate one of the main scalability limitations, the integration density. The paper finalizes with an introduction to control algorithms for waveguide mesh arrangements based on derivative methods and non-derivative methods. These control methods provide a proof for the self-reconfiguration of large-scale waveguide mesh arrangements. In particular, we apply the computational optimization algorithms to program a hexagonal waveguide mesh to emulate a 1×8 beamforming network and an optical filter based on an unbalanced MZI design. All in all, the paper comprises recipes to achieve truly practical software-defined photonic integrated circuits.**

***Index Terms*—Programmable photonics, reconfigurable circuits, signal processing, integrated circuits, control algorithms.**

I. Introduction

PROGRAMMABILITY, re-usability, reconfigurability, versatility and flexibility are key features in past and future revolutions. Almost all technology-based research areas are severely impacted when these features are incorporated to their processes and performances [1], [2]. For example, in electronics, the paradigm shift from application specific integrated circuits (ASPICs) to general-purpose electronic processors and Field-Programmable Gate Arrays had become one of the strongest bases of the third and fourth industrial revolutions [3]. As another example, traditional network management is being progressively substituted by Software-defined networking, enabling a more efficient dynamic network configuration [4]. Moreover, other areas like software-defined radio incorporate general-purpose processors and a set of widely employed hardware electronics and high-speed electronics to enable dynamic configuration of communication protocols, frequency ranges and optimization algorithms [5]. Photonics is not an exception. Closely mimicking the past electronic trends, in the last 50 years, light-based systems have evolved from the first proposals of discrete integrated components to the growth of a complete ecosystem tailored to produce Application Specific Photonic Integrated Circuits (ASPICs) [6], [7]. As in electronics integrated circuits, the up-front non-recurring engineering costs (NRE) including custom mask tooling, design hours and specific process developments reduce the cost effectiveness for low-volume applications. Only few application fields like transceivers and datacenters have shown enough volume fabrication to compensate the overhead costs [8]. As a solution leading to mass production and subsequent cost reduction for PIC manufacturing, multi-project wafers are fabrication runs where different designs or ASPICs from different users are combined on the same wafer providing cost sharing and additional advantages, such as on-wafer testing. However, the time-to-market and time-for-development of PIC-based science and technology is still limited by the custom design processes and by the large development periods of around of 12–24 months per design-fab-packaging-test iteration, depending on the chip complexity.

Replicating a prior roadmap followed by electronics, in the last decade, the reconfigurability degree of photonic integrated components has been progressively increasing. The introduction of integrated optical actuators makes PICs configurable and programmable, modifying the light propagation conditions locally, either in amplitude or phase with the proper electrical control signals [2]. The very-large scale integration of actuators is the base under the concept of general-purpose photonic integrated circuits. Highly-inspired, by the electronics processors, a general-purpose photonic processor incorporates widely-employed resources that are enabled, routed and selected on demand by software definition [9], [10]. As a possible candidate for the core of this processor, recent advances in programmable photonics have demonstrated PICs relying on a very-large scale interconnection of a set of beam-splitters

Manuscript received June 30, 2019; revised September 9, 2019; accepted October 2, 2019. Date of publication October 17, 2019; date of current version November 12, 2019. This work was supported in part by the ERC ADG-2016 UMWP-Chip, in part by the ERC PoC-2019 FPPA, in part by the Generalitat Valenciana PROMETEO 2017/017 research excellency award, and in part by the COST Action CA16220 EUIMWP.

The author is with the Photonics Research Labs, Universitat Politècnica de València and iPronics Programmable Photonics S.L., Valencia 46022, Spain (e-mail: dperez@iteam.upv.es).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTQE.2019.2948048

and phase actuators (sometimes referred as Tunable Basic Units (TBUs)) that, inspired by electrical FPGAs, allow the synthesis and programming of photonic integrated circuits discretized into primitive blocks of TBUs [11]–[14].

In parallel, alternative waveguide mesh arrangements of TBUs allowing feed-forward-only propagation and multi-interferometry of the optical signal [15] have demonstrated promising results for quantum computing [16]–[18], as hardware accelerators for deep-learning applications [19] and general linear signal processing [20], [21].

In this paper, we focus our work on general-purpose waveguide mesh cores that enable the programmability of not only multi-interferometry feed-forward propagation of light but also optical signal feedback-loops and cavities. In particular we review the state-of-the-art and report optimized architectures to mitigate the scalability issues dealing with footprint and performance. The control and practical operation of such circuits is a current open research question. In Section III, we propose and demonstrate the use of computational optimization methods to enable a practical control and self-configuration of very-large scale PICs based on general-purpose waveguide mesh arrangements. Both derivative-based and non-derivative-based optimization techniques are proposed. We finish the paper with the conclusions and envisioning the future roadmap of multipurpose programmable photonic integrated circuits.

II. Waveguide Mesh Cores

Fig. 1(a) illustrates a conventional ASPIC example that includes an optical filter based on an add-drop ring resonator accessible through port 1 ($p1$). The *through port* output is split by two ports, p_2 and p_3. The latter terminates a waveguide 1.5 times larger than the former and includes a tunable phase actuator. The *drop-port* output is labeled as p_5 and an additional input port is defined by p_4. Any minimal circuit upgrade or modification forces to start again a time-consuming and costly custom design-fabrication-packaging-test circle. As illustrated, the optical circuit design in Fig. 1(a) is composed of a well-organized set of fourteen waveguide elements, two phase shifters and two beam-splitters or tunable couplers. Moreover, the combination of these basic building blocks is enough to implement the vast majority of the most simple and complex photonic linear processing architectures. This premise is one of the two fundaments of integrated waveguide meshes. A second key feature, is given by the fact that all these basic building blocks (discrete delay lines, optical switches, tunable couplers or phase shifters) can be emulated by a 4-port photonic component by suitable programming of its control signals. This programmable photonic block is called Tunable Basic Unit (TBU). A description is included below.

Under these premises, the combination and interconnection of a large-enough number of TBUs describing a waveguide mesh arrangement can be employed as a common hardware photonic platform to program a wide variety of photonic linear processing architectures and design parameters. As illustrated in Fig. 1(b), the independent configuration of each TBU either as a fixed delay line (optical switch in cross or bar state), as a tunable coupler and/or as a phase shifter results in the emulation or synthesis of the circuit in Fig. 1(a).

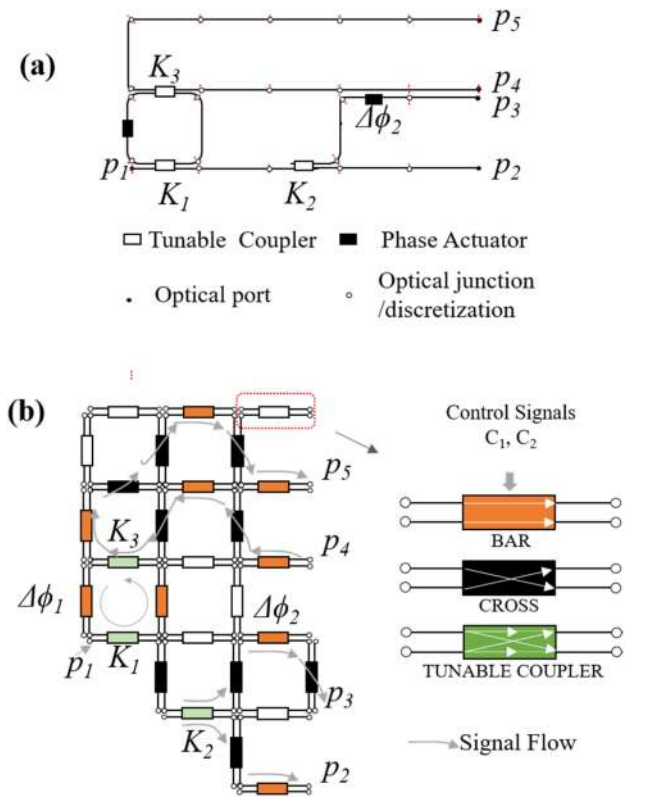

Fig. 1. (a) Schematic of a conventional photonic integrated circuit showing its discretization into waveguide elements, couplers and phase shifters, (b) waveguide mesh arrangement configuring the previous circuit and inset of different configurations of the Tunable Basic Unit.

In the previous example, the TBU interconnection follows a square mesh topology. Notwithstanding, circuit discretization, space and power consumption constraints play a critical role in the design of photonic circuits in general and of optical meshes in particular. The limited area available for growing the optical mesh, the mapping of standard circuits into a TBUs, and the need to reduce to a minimum the number of switching elements required for implementing a set of optical core topologies call for a careful analysis of possible geometries for the mesh interconnection implementing the optical core.

A. Tunable Basic Units

Before considering the waveguide mesh analysis, it is essential to fully understand the tasks and fundamentals of its primitive block, the TBU. A TBU is a 2×2 photonic component that employs beam-splitters and phase actuators to synthetize any coupling state and a common phase, independently. The interconnection between TBUs allows the user to synthetize a light-path on demand after fabrication by tuning each TBU defining the path where the optical signal goes through. Although there are multiple alternatives with different port count, tuning mechanisms and architectures [2], [14], [22], the simplest approach is the TBU that relies on a balanced Mach-Zehnder Interferometer

with an independent phase actuator on each arm. Most of the experiments to date had relied on this configuration and employ thermo-optic phase actuators as their tuning mechanism. Finding the optimum phase actuator technology is a current research area, where shorter lengths, reduced footprints, and low crosstalk and power consumption are the key performance indicators.

For a balanced MZI loaded with heaters on both arms, the splitting ratio is obtained by increasing the effective index due to the Joule effect in the upper or lower arm, producing a ϕ_{upper} and ϕ_{lower} phase shift respectively. Once set, a common drive in both heaters will provide a common phase shift, leading to independent control of the amplitude ratio and the phase [11], [22]. The simplified transfer matrix is defined by:

$$h_{TBU} = je^{j\Delta} \begin{pmatrix} \sin\theta & \cos\theta \\ \cos\theta & -\sin\theta \end{pmatrix} \gamma e^{-j\omega BUD}, \qquad (1)$$

where θ is $(\phi_{\text{upper}} - \phi_{\text{lower}})/2$, Δ is $(\phi_{\text{upper}} + \phi_{\text{lower}})/2$, and *BUD* is the time that takes the signal to go through the TBU or *basic unit delay*. The coupling factor *K* is then defined as $\cos^2(\theta)$.

The key features of a TBU are its *excess loss*, the *power consumption* to get a π-phase shift, total length including access waveguides or *basic unit length* (*BUL*), the *BUD* and its longitudinal and transversal dimensions. They are precisely defined in [2].

B. General-Purpose Waveguide Mesh Design Topologies

Very recently, alternative arrangements allowing both the synthesis of circuits including feed-forward light propagation and optical feedback loops and cavities have been proposed and experimentally demonstrated. Fig. 2 illustrates them.

1) Square Waveguide Meshes: Square waveguide mesh arrangements were the first proposed mesh topology. In their pioneering work, L. Zhuang and co-workers developed the concept where the interconnection of a large number of integrated balanced MZIs with two actuators could lead to the synthesis of a wide variety of PICs [11]. As we saw in Fig. 1(b1) this is achieved by discretizing conventional circuits into TBU with specific configurations. The proposed interconnection topology enables the routing of the optical signal path to follow orthogonal directions where the repetition of a given direction is not allowed for two consecutive TBUs. For example, in Fig. 2(b1), it is not possible for the signal flow to go from TBU5 to TBU 12 and TBU19. However, the topology is flexible enough to allow the synthesis of waveguides or discrete delay lines, tunable couplers as well as phase shifters and thus more complex building blocks like optical cavities and unbalanced MZIs. Fig. 2(b1) illustrates an example with the longitudinal axis orientation specified.

2) Triangular Waveguide Meshes: The triangular waveguide mesh topology was proposed in [2], [22]. In this case, three TBUs describing longitudinal orientations of 0° (horizontal plane), 60° and −60° are interconnected through 6 points resulting in a triangular pattern, as illustrated in Fig. 2(c1). Notice that the angle described by the longitudinal axis of two connected TBU is always 60°. Again, it can be shown that the interconnection scheme directly impacts on the degrees of freedom related to propagation directions for the light flow across two consecutive TBUs and consequently the allowed light-paths and circuits that can be implemented inside the arrangement. This topology is particularly useful to increase the integration density of TBUs, and to implement optical cavities with reduced cavity lengths. The triangular pattern achieves a better discretization resolution of unbalanced filter structures as compared to the square topology. They have been experimentally proven using Dual-Drive Directional Couplers as their TBU [14].

3) Hexagonal Waveguide Meshes: As we saw for the square and triangular topologies, the interconnection schemes between TBUs lead to regular uniform patterns with different topologies. The hexagonal topology was proposed in [2], [22]. It is a more efficient 3-point optical interconnection scheme that resembles the arbitrary linear interferometer layouts with the difference that the proposed pattern also allows the synthesis of optical feedback loops and optical cavities. As illustrated in Fig. 2(a1), the light flow propagates through TBUs with longitudinal axis orientation of ±60° and 0°. However, in contrast to triangular waveguide meshes, the angle described by the longitudinal axes of two connected TBU is always 120°. The main advantages are their versatility and flexibility in fitting conventional circuit programming, their enhanced resolution compared to the two previous topologies and their synthesis efficiency (See point 5). In addition, as opposed to the two previous patterns, it enables the synthesis of Sagnac mirrors, which are an essential building block for Fabry-Perrot Cavities. One of the main drawbacks of waveguide meshes, that is exacerbated in the hexagonal topology, is that the overall mesh footprint can be compromised if the longitudinal TBU dimension is too large. They have been experimentally proven showing the programming of multiple circuits [2], [12], [22]–[24].

4) Longitudinally Parallel Waveguide Meshes: In the previous subsections, we saw that the proposed waveguide mesh circuit topologies arise from the selected optical interconnection nodes between TBUs following regular and uniform spatial geometries. We have given as much importance to the resulting pattern as to the interconnection scheme. However, in this section, we propose an optimized design layout for waveguide mesh arrangements, where we eliminate the dependencies between interconnection nodes and resulting topologies.

The motivation is three-fold. First, the current topologies are geometrically constrained by the patterns originated by their connections. In other words, once selected one of the interconnection topologies, our arrangement will inherit a series of capabilities, characteristics and performance metrics (analysed in [2], [22]). As a preliminary example, hexagonal topology is characterized by a better versatility, and programming flexibility but the associated footprint is compromised by the space-consuming hexagonal shapes.

Secondly, the integration of TBUs with different angular orientations implies the integration of some building blocks with performance metrics that are orientation-sensitive. In particular, 3-dB couplers like directional couplers or multi-mode interferometers designs are more reproducible and robust to fabrication if the longitudinal orientation is kept fixed. In addition, some phase actuators are orientation-sensitive and are

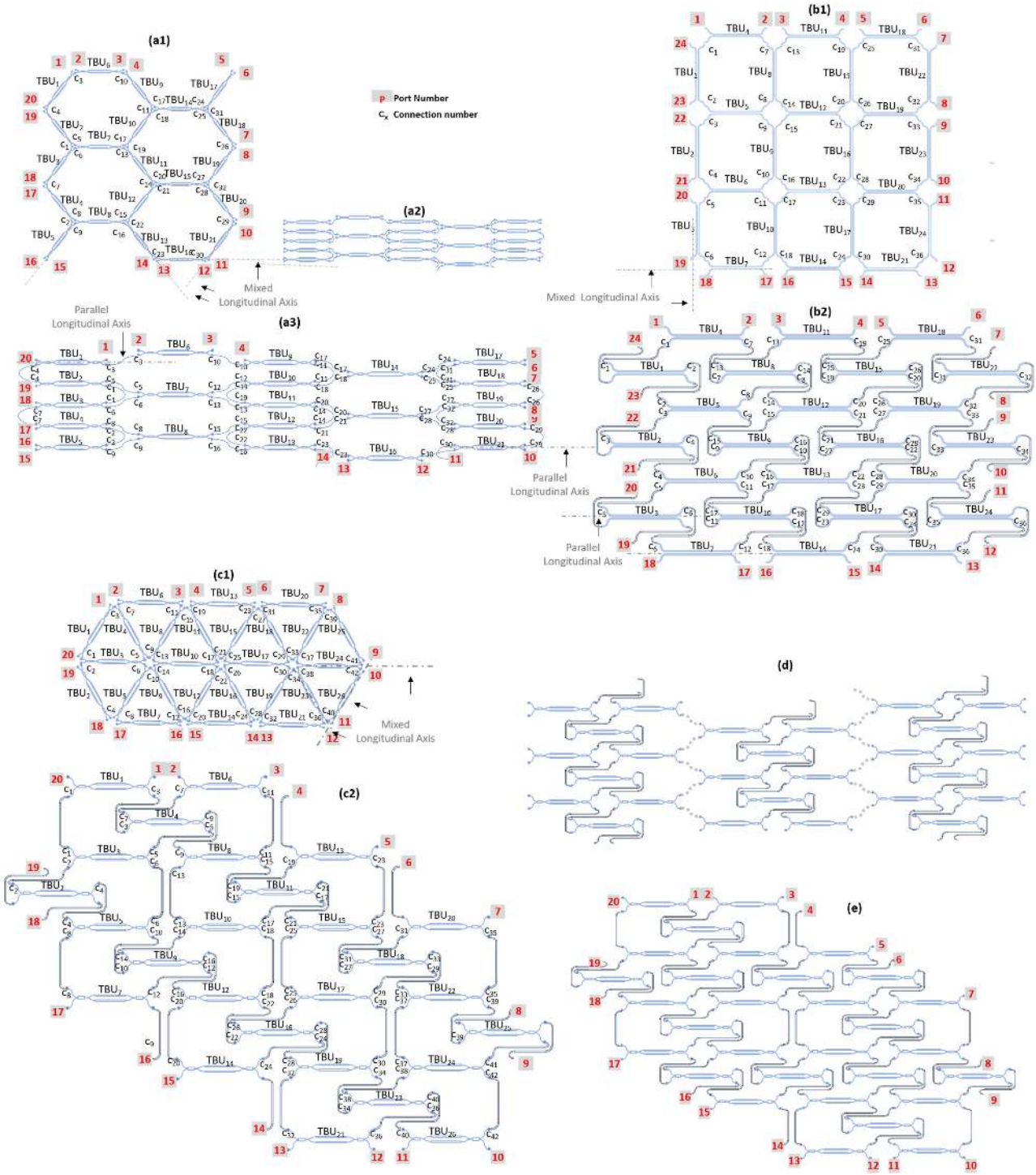

Fig. 2. Schematic diagram example of the implementation and interconnection of the (a1) hexagonal pattern and (a2) the longitudinally- parallel mesh topology homologue to the hexagonal optical interconnection, (a3), a labelled and augmented version of (a2), (b1) square pattern and (b2) the longitudinally- parallel mesh topology homologue to the square optical interconnection, (c1) triangular pattern and (c2) the longitudinally- parallel mesh topology homologue to the triangular optical interconnection, (d-e) examples of longitudinally-parallel implementations where a translation to a conventional uniform topology is not possible. Note that the TBUs in (a3), (b2), (c2), (d) and (e) are not in the same scale as in (a1), (b1), (c1) and the access waveguide dimensions have been exaggerated.

more efficient if single or orthogonal orientations are maintained [25], [26].

Finally, the fabrication costs are directly related to the total footprint of the integrated device. The use of any of the aforementioned topologies results in an increment of the unused areas in the device, particularly in designs with longitudinally large TBUs. Although it might be convenient to maintain a safety gap for optical and tuning-derived crosstalk, we would like to decouple this parameter from the selected waveguide mesh interconnection scheme.

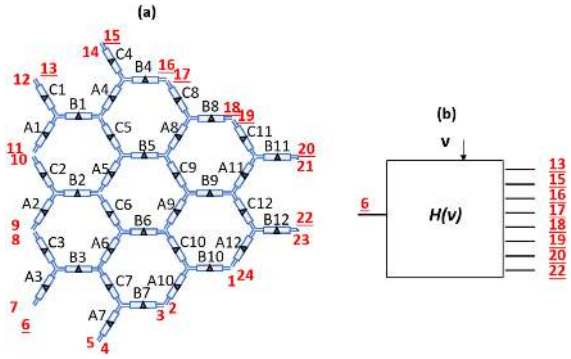

Fig. 3. (a) Schematic of the waveguide mesh arrangement employed in the examples, (b) Targeted black-box system of a 1 × 8 beamsplitter with labelled ports equivalent to (a).

In the light of the previous considerations, we describe here the use of parallel waveguide meshes to potentially solve all the aforementioned issues. Since the main performance metrics are related to the optical interconnection schemes or optical nodes, there are multiple possibilities for the definition of their architectures. However, in contrast to the previous patterns, they benefit from maintaining all TBU with the same longitudinal orientation. Compared to the square, hexagonal and triangular lattices, more complex access paths are required in the definition of the TBU for their interconnection, however, their impact is mitigated in integration platforms with high index contrast due to the reduced bend radius.

Fig. 2 illustrates the concept by comparing the previous topologies with their homologue parallel/flat configurations. See Fig. 2(a3, b2, c2, d, e). Note that the access waveguides sizes have been exaggerated to enhance the figure readability and to highlight the optical interconnections. Consider that if the bend radius in the access waveguide is small, the overall design footprint becomes narrower in the vertical direction much more than wider in the horizontal direction and the resulting integration density increases for every interconnection scheme. For an unaltered footprint comparison see Fig. 2(a1) and Fig. 2(a2). In addition, it can be shown that during the design of longitudinally- parallel meshes, there is more freedom to define the optical interconnection node between TBUs. In the non-parallel implementations, one is constrained to a set of few geometrical patterns. As an example, Fig. 3(d–e) illustrates two alternative parallel waveguide mesh geometries that do not have a straightforward associated uniform topology. Note that Fig. 2(d) is particularly interesting for the implementation of Sagnac mirrors with a shorter cavity length when compared to the hexagonal mesh.

In short, the longitudinally parallel waveguide mesh arrangement version maintains the main characteristics of their associated conventional topologies while increasing the integration density. The final integration density is mainly limited by the safety distances between phase actuators to prevent tuning crosstalk and by the required space to enable the effective routing of the metal layers.

5) Comparison Between Waveguide Meshes: The conventional square, triangular and hexagonal mesh geometries have been analysed and compared against a set of different figures of merit that measure their programmability efficiency and integration properties [22]. The summarized analysis is included in Table I, with additional key performance estimators. For reference, *the spatial tuning resolution step* quantifies which is the minimum step in BUL units by which the arm length mismatch or the cavity length can be increased or decreased, the *reconfiguration performance* of the mesh is given by the number of filters with different spectral period values that can be implemented given a maximum value X (in BUL units) for the path imbalance in unbalanced Mach Zehnder Interferometers (UMZIs) or the cavity length in Optical Ring Resonators (ORRs). The *replication flexibility* gives the number of possible alternative geometries for a specific filter implementation, while the *switching elements per unit area* denotes the amount of required BULs per surface unit to achieve an equivalent functionality. Also, the table includes the tolerance ratios to the bend radius and angles. Moreover, it is shown that although the triangular mesh offers the highest

TABLE I
SUMMARY OF VALUES FOR THE FIGURES OF MERIT OF THE DIFFERENT MESH DESIGNS (SEE DEFINITIONS IN [22])

Figure of Merit	Triangular	Square	Hexagonal
ORR cavity spatial tuning resolution step in BUL units (the lower the better)	**3**	4	2* **The first and second step has a resolution of 6 and 4.**
MZI arm imbalance spatial tuning resolution step in BUL units (the lower the better)	3	4	**2**
ORR reconfiguration performance (the higher the better) (for X=25 *BUL*)	8	6	**9**
MZI reconfiguration performance (for X=25 *BUL*)	8	6	**12**
Switching elements per unit area compared to *square* % (the lower the better for a fixed value of reconfiguration performance)	+65.00%	0.00%	**-36.66%**
Replication Ratio for ORR structures up to 16 BUL cavity length (the higher the better).	1	**2.68**	1.31
Replication Ratio for MZI structures up to 12 BUL interferometric length (the higher the better).	1	3	**3.36**
$L_{access}/L_{access\ square}$ % for a fixed radius of curvature *Ra* (the lower the better)	+33.33%	+0.00%	**-33.33%**
Ra/Ra_{square} % for a fixed BUL value (the higher the better)	-25.00%	+0.00%	**+50.00%**
Integration density (Parallel version)	**Moderate/High (High)**	Moderate **(High)**	Low **(High)**
Multiport interferometers	**Yes**	Low-efficiency	**Yes**
Sagnac Loops implementation	No	No	**Yes**

integration density, the longitudinally parallel implementation version of all of them increase the overall integration density, as illustrated in the comparison between Fig. 2(a1) and Fig. 2(a2). Finally, two final rows qualify the capability to perform multiport interferometers like the implemented with feed-forward meshes [15]–[21] and Sagnac-loops efficiently. A complete definition of the figures of merit can be found in [22].

All in all, the table reflects that the hexagonal waveguide mesh offers the best performance for the task of programming conventional photonic integrated circuits with more resolution in the delay lines definition. However, its integration density is lower than the one provided by the triangular and the square patterns. As a solution, the longitudinally parallel implementation of the hexagonal pattern mitigates this issue and improves the integration density. This is achieved by reducing the empty space areas inside the cells up to a safety margin that prevent from tuning-related crosstalk. Note that most of the figures of merit covered in the table are relative to the optical interconnection scheme, so equivalent results are obtained for the standard- and their associated longitudinally-parallel versions. This effect can be explained through graph-based theory [27]. If we assign a vertex to each interconnection node, and an edge to the four signal flow possibilities in a 2×2 TBU, we will find that the number of edges and vertex and their interconnection schemes are equivalent for the standard and longitudinally-parallel designs. This property is known as *isomorphism* between the two graphs.

In addition, for most applications, the access waveguides that interconnect the TBUs should be of equal length in order to preserve the same *BUL* and *BUD* over the waveguide mesh. Also, an efficient design would reduce the number of bends and would try to compensate the number of bends and their orientation for every access waveguide. The design of this access path is straightforward in all the proposed arrangements.

The design of this access paths is straightforward for most of the proposed arrangements. However, longer access waveguides are employed inevitability for some arrangements (See Fig. 2(c2)), while in other patterns longer accesses are employed if we make the arrangement more compact in the horizontal direction (see Fig. 2(b2)). In the latter, we can avoid this length increment by allowing a wider aspect ratio. Finally, other interconnection schemes, like the employed in Fig. 2(a2–a3), do not require large access waveguide increments. Since the BUD definition includes the delay at the TBU and the access waveguides, the impact of longer access waveguides must be seriously considered in terms of waveguide mesh resolution, phase coherence and extra losses.

Finally, some applications might find useful a variable non-uniform BUL, giving birth to non-uniform waveguide meshes.

III. Optimization Methods for Waveguide Mesh Cores

In the previous section, we saw that optimized mesh layouts enable denser PIC. As the number of reconfigurable integrated components increase, the need for a control system and routines enabling a practical use of the PIC becomes mandatory. Historically, the complexity of reconfigurable photonic circuits has been moderate and either manual or semi-automated iterative routines have been enough to enable their operation. The control system can be composed of electrical driving circuitry for the actuators, an electrical circuitry to monitor the optical signals at specific points in the PIC, and a logic unit to run the computational algorithms. Based on the feedback collected by the readout system, a microcontroller computes the new required driving signals for the actuators based on software routines. Examples of these schemes have been employed in the configuration of optical bank filters based on cascaded MZIs [28], optical ring resonators [29], and moderate-size multiport interferometers propagating in feed-forward waveguide mesh arrangements [18]–[21], [30]. In the latter circuits, the optical signal monitoring is located after each TBU, increasing the control system overhead and limiting the scalability of the circuit. In order to reduce the number of readout points, it has been theoretically proposed the use of external monitoring at the output ports in feed-forward-only waveguide mesh arrangements [31].

In this section we propose and demonstrate the use of optimization methods for the control and configuration of general-purpose waveguide mesh arrangements like the ones covered in the previous section. The main difference with the algorithms and architectures published to date is that we aim for an effective optimization method that allows us to configure waveguide mesh arrangements that allow both feed-forward and optical feedback loops. Moreover, we aim for optimization algorithms that avoid signal preparation at the inputs and optimize the number of optical readouts. A straightforward approach consists of the integration of optical power monitors on every TBU output. Once obtained the coupling response of each TBU one can program the overall mesh by using global configuration algorithms [32].

However, as in the case of feed-forward-only arrangements, the integration of readouts at every TBU implies a serious limiting overhead for the PIC scalability both in its packaging and electrical interfacing and in the optical power budget. In order to relax this limit, we propose the reduction of the number of optical readouts points to every output port of the optical mesh, i.e., without internal monitoring.

The optical monitoring can be based on integrated photodetectors after collecting a portion of the signal through a directional coupler or by using advanced structures like CLIPP detectors [21] or low-efficiency optical monitors [33].

Then, we can develop and apply software routines that include adaptions of mathematical optimization methods to program the actuators and get the desired functionality of the system. Mathematical optimization methods deal with the problem of finding minima/maxima of a cost function. In our case, we define application-specific cost-functions. They are computed based on the specifications of the targeted functionality. As the proper algorithm, *per se*, the design of the cost function is essential for the success and faster convergence of the algorithm. The cost function can combine several features *f* and being multiplied by a correction factor (*c)* to specify its weight in the final metric:

$$CF(\mathbf{v}) = \sum_n c_n f_n(\mathbf{v}), \tag{2}$$

where *n* is the number of features considered and **v** is the vector including the variables that are tuned in the optimization process. In our present work, **v** includes a set size equal to the double of the number of TBU enabled during the optimization $(2N_{TBU})$. The first half of elements deals with the differential drive and actuates only the upper phase shifter of each TBU. The second half deals with the common drive and actuates over the sum of both phase shifters in the TBU.

As far as features are concerned, an experimental demonstration would consider postprocessed data from the optical readout system. In the present work, we employ the data obtained by a scattering matrix analytical model derived in [13] that computes the scattering matrix H(**v**) of the targeted waveguide mesh arrangement that considers the specified driving of each phase actuator (**v**). In addition, we emulate the random passive initial state of real TBU imposing an additional phase term to the upper phase shifter. This value is not specifically treated during the optimization process and is only considered for verification and monitoring. Although analytical solutions enable a safer and faster development environment, obtaining the scattering matrix is more computationally expensive and time consuming than obtaining the data from real readouts. For simplicity we will obviate that H is a function of **v** in the formulation.

As an example, if our functionality is a programmed light-path between ports 3, 8 of an arbitrary waveguide mesh arrangement we could define two features for the cost function before starting the minimization process of (2) as:

$$\begin{aligned} c_1 &= -1, \quad f_1 = |H_{8,3}|^2, \\ c_2 &= 0.5, \quad f_2 = \sum_{p=1}^{P} |H_{p,3}|^2, \end{aligned} \tag{3}$$

where $H_{o,i}$, is the optical field response from a channel defined by an input port *i* and an output port *o*.

Note that the first feature is accompanied by a negative correction factor to ensure that the minimization of the CF leads to the targeted functionality. The second feature aims to minimize the power in the remaining output ports during the optimization process.

The number of alternative computational optimization techniques is wide-ranging and they are applied in very different application fields. However, no computational optimization techniques have been proposed for general-purpose waveguide mesh arrangements. Here we will report optimization solutions to program the mesh arrangement of Fig. 3. These include the description of the targeted functionality, the definition of the engineered cost function and the development and application of the software-routine. The selected PIC is a general-purpose waveguide mesh arrangement composed of 36 TBUs based on a MZI, (72 phase actuators) and 24 optical outputs. Although the standard shape is employed, note that its application to the flattened longitudinally-parallel version would be equal.

A. Derivative Methods (First-Order Optimization Algorithms)

For the minimization of the cost function, these set of techniques employ the multivariable generalization of the derivative of the CF for each variable in **v**. The resulting vector ***g*** is the gradient of the function and it provides the direction tangential to the error surface at the evaluation point defined by **v**:

$$\mathbf{g} = \nabla_{\mathbf{v}} CF(\mathbf{v}). \tag{4}$$

This direction is employed to advance on the opposite way to progress in the minimization of the error function. A wide range of first-derivative optimization methods are reported in the literature [34]. The simplest and more extended is the gradient-descent algorithm. It renders the next configuration state of our system settings by sequentially applying:

$$\mathbf{v}^t = \mathbf{v}^{t-1} - \eta \mathbf{g}, \tag{5}$$

where η is the learning rate. In data science and statistics, parameters associated to the optimization routine are called hyper-parameters. The learning rate must be selected by the practitioner and regulates the step size between iterations. A large value can make the process converge earlier but can also lead to noisy results and prevent the algorithm to achieve the optimal value of **v**. Strategies to find optimum hyperparameters are discussed in [34].

A straightforward approach for computing the derivatives of the error function is to use finite differences approximation. This can be done by perturbing each variable in turn, and approximating the derivatives by the following expressions:

$$g_i = \frac{CF(v_i + \varepsilon) - CF(v_i)}{\varepsilon} + O(\varepsilon), \tag{6}$$

$$g_i = \frac{CF(v_i + \varepsilon) - CF(v_i - \varepsilon)}{2\varepsilon} + O\left(\varepsilon^2\right), \tag{7}$$

In (6–7) the gradient employs the evaluation of the CF in **v** perturbing the position *i* by a small amount ε. In our case, we define ε equal to 0.3 10^{-3} rads. The rationale behind is that the finite-differences approximation is better approximated if we are close to the evaluation point. However, the lower limit of ε will be imposed by the resolution of our electrical drivers and the noise of the readout system. In (7), we see the central differences equation to get a significantly better approximation of the gradient. However, the number of computational steps is almost doubled when compared to (6).

As the reader can infer, getting the gradient straightforwardly in a real waveguide mesh system, as proposed, implies performing the perturbation of one actuating variable, getting the associated CF from the postprocessed signal from the readout system, repeating the procedure with a negative perturbation and then computing (7). Then, this procedure is iterated for every optimization variable till getting the full vector **g**. Next, we apply (5) to get the next actuation driving variables $\mathbf{v}^t$ and if the exiting conditions are not achieved, we iterate again.

To enhance the convergence speed, and to overcome noisy gradients, we can make use of the momentum algorithm. It accumulates an exponentially decaying moving average of past gradient vectors $\boldsymbol{\nu_{mo}}$ and use a proportion (α_{mo}) of it to set the new direction. The new $\mathbf{v}^t$ is

$$\mathbf{v}^t = \mathbf{v}^{t-1} + (\alpha_{mo}\mathbf{v}_{mo} - \eta \mathbf{g}), \tag{8}$$

where α_{mo} is another hyper-parameter in the range [0, 1) that determines the relation between the new gradient and the accumulated.

The following procedure incorporates the algorithm. If α_{mo} equal to 0 is selected, we get the conventional gradient descent.

---

procedure gradientDescent_momentum_mesh()
  require: initial $\boldsymbol{v}$ (can be random or 0)
  require: set initial $\boldsymbol{\nu_{mo}}$ to zeros.
  setting η, α_{mo}
  while (stopping conditions or maxiter)
    get gradient (Eq. (4 and 2))
    get next variable vector (Eq. (8))
  end while

To test the procedure, we start with a first application. In this case we want to get the optimum actuation variables $\mathbf{v_{opt}}$ to achieve a mesh performance equivalent to a 1×8 beam splitter as illustrated in Fig. 3(b). Since the splitting of the input signal by 8 will introduce a 9.03 dB penalty and, assuming average penalties of 0.1 dB per TBU, we will set our targeted channel loss to 10 dB. The selected input port will be port 6, whereas ports 13, 15, 16, 17, 18, 19, 20, 22, will be the output ports.

These specifications can be translated to the following cost function variables to be evaluated in (2):

$$\begin{aligned} &c_1 = 1/8,\ c_2 = 0, \quad c_3 = 1/8, \\ &f_1 = \sum_{op} \left(10\log_{10}\left(|H_{op,6}|^2\right) + 10\right)^2, \\ &f_2 = \sum_{p \neq op} \left(10\log_{10}\left(|H_{p,6}|^2\right)\right), \\ &f_3 = \begin{pmatrix} \sum_{op} \max\left(10\log_{10}\left(|H_{op,6}|^2\right)\right) \\ -\min\left(10\log_{10}\left(|H_{op,6}|^2\right)\right) \end{pmatrix}, \end{aligned} \tag{9}$$

where *op* references the optical ports under use by this configuration. Note that, in this case, we have disabled the information coming from the signals from the remaining ports in the second feature to reduce the number of reads by a practical readout system. As for the third feature, it considers the ripple of the channel to achieve a better cost function. This is an example of a cost function that employs spectral information, meaning that if low-speed diodes are employed a laser swept would be required in a real system implementation. Alternatively, a filtered WDM spectrum could be photodetected at each spectral channel, increasing the complexity of the system. The use of extra features and the consideration of non-used or secondary ports is particularly interesting for larger-scale waveguide meshes.

Fig. 4 illustrates an example of the resulting optimization procedure were η, α_{mo} are 7.2 10^{-2} and 0.8, respectively. First, in Fig. 4(a), we can see two $c_i f_i$ products and the CF for each iteration. By regulating the coefficients c_i we can weight their contribution in the overall CF. In this example, we appreciate that *feature* 1 is the dominant until arriving to iteration 12. Then *feature* 3, which is proportional to the mean ripple becomes dominant until the end of the process. At iteration 45, the selected exiting criteria is satisfied, giving as a result the optimum $\boldsymbol{v}$. For the stopping conditions, we selected the ripples mean value below 0.2 dB and a mean error in the channels of $c_1 f_1 = 0.12$, that ensures an average error of around 0.34-dB optical power deviation per channel. The latter was achieved at iteration 32. Fig. 4(b) illustrates the coupling factor modification of each TBU during the optimization process. It is worth mentioning again

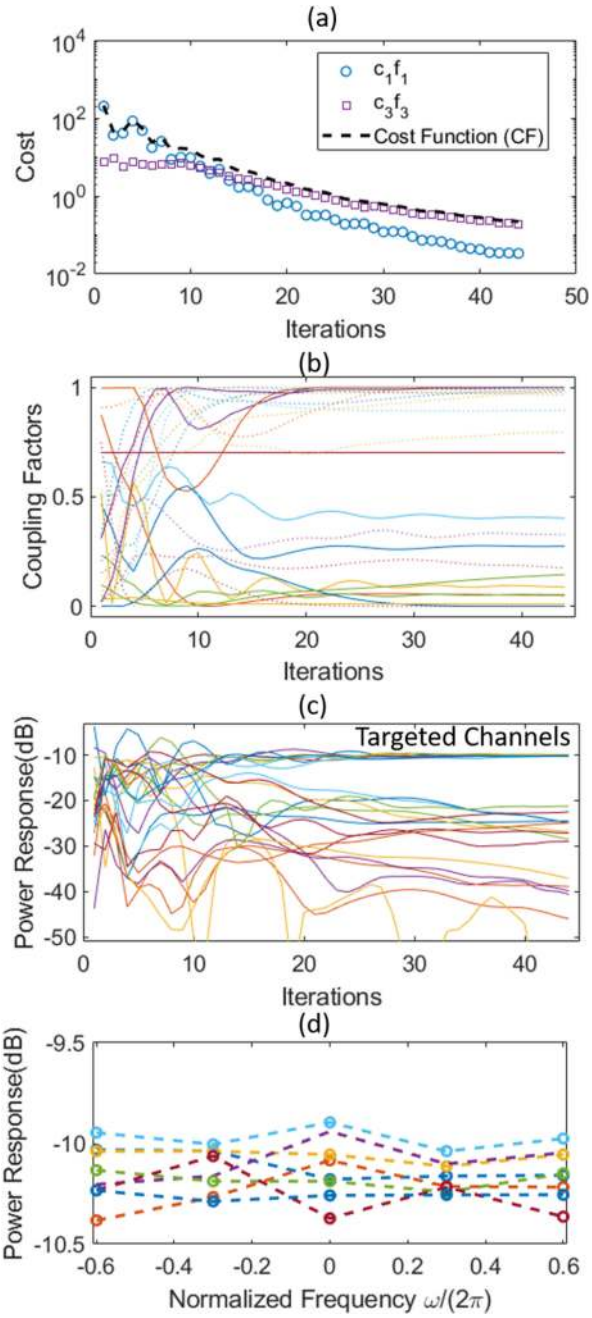

Fig. 4. (a) Cost function minimization versus iterations, (b) Coupling coefficient tuning during optimization process, (c) Optical power monitoring at every waveguide mesh output during the optimization process, (d) final normalized frequency response normalized to one single TBU delay. Only five spectral points are considered per channel.

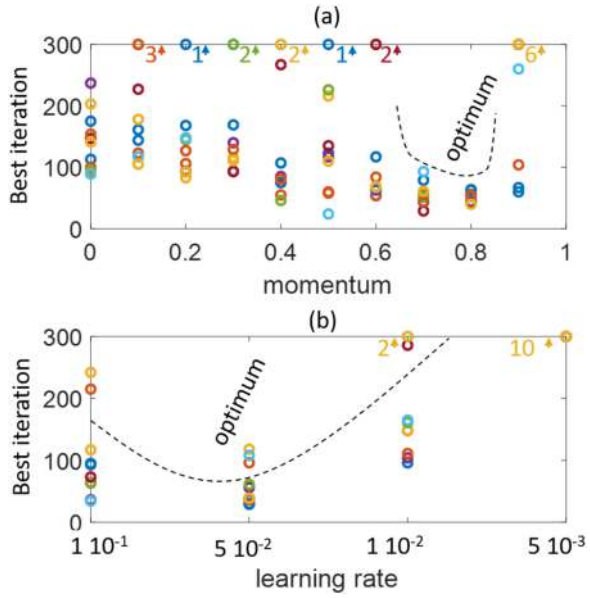

Fig. 5. (a) Test results containing the best iteration (convergence) for the 1×8 beamsplitter for (a) different momentums at a fixed learning rate of 7.2e-2, (b) for different learning rates and fixed momentum of 0.8.

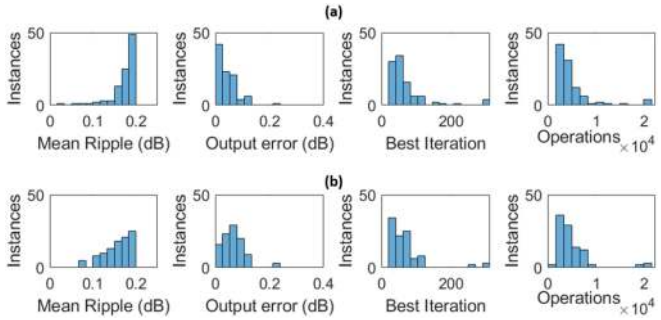

Fig. 6. Statistical results obtained after performing the optimization process of the 1×8 beamsplitter 100 times. (a) feature 2 is not considered ($c_2 = 0$), and (b) the feature 2 is slightly considered ($c_2 = 0.01$).

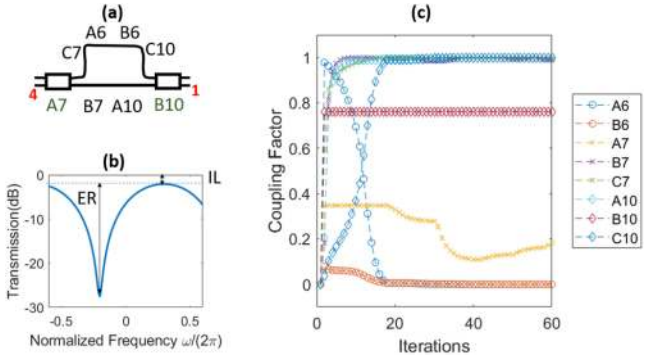

Fig. 7. (a) Targeted unbalanced MZI architecture with labelled ports mapping the code of Fig. 4(a). (b) Resulting transfer function after optimization process with labelled performance and (c) coupling factor variation of the enabled TBUs during the optimization process. ER: Extinction Ratio, IL: Insertion loss.

that this data have not been employed during the optimization process and they is only employed for verification purposes. In Fig. 4(c) we can see the power responses of every channel in the waveguide mesh arrangement. We can appreciate that the 8 targeted ports converge at the desired -10 dB-level. Since we did not included *feature* 2 in (9), the optical crosstalk is slightly better than 10 dB for the unused channels at iteration 45. Both including the feature and adding more restrictive stopping conditions lead to an enhancement in the optical crosstalk with the unused ports. Finally, Fig. 4(d) illustrates the flat optical power spectral response achieved for each targeted optical channel at the end of the optimization, with all ripples below 0.5 dB.

In order to show the implications of the hyper-parameter selection we maintain the learning rate and ε fixed with the aforementioned values. Then we sweep the momentum coefficient α_{mo} from 0 (standard gradient descent) to 0.9 in steps of 0.1. For each value we perform 10 tests allowing a maximum of 300 iterations per test. Each test starts with random and unknown passive phases at each TBU. If the targeted exit conditions are achieved, then it exits the test and saves the best iteration. In this example the exit conditions are maintained as in the previous example.

We will consider that the ones that require more than 300 iterations do not properly converge. Fig. 5(a) illustrates the resulting values. We can appreciate that the optimum values of enhanced convergence are achieved by $\alpha_{mo} = 0.8$. In this case, they require less than 63 iterations to converge. Fig. 5(b) represents the results where the momentum is fixed to 0.8 and the learning rate is varied, showing the impact of this hyper-parameter.

In order to test the proper behavior of the algorithm, we perform the test with the optimal hyper-parameters 100 times and perform a statistical analysis of the achieved ripples, output mean error in the selected channels and the number of required iterations and operations performed. As mentioned for the previous example, for each test, the initial phases are selected from a random uniform distribution, to imitate the random nature of a fabricated waveguide mesh with fabrication errors. The analysis plotted in Fig. 6 is performed two times: in Fig. 6(a) the feature 2 is not considered ($c_2 = 0$), and in Fig. 7(b) the feature 2 is slightly considered ($c_2 = 0.01$). In short, the test reveals that a small impact in the definition of the cost function impacts over the performance of the optimization process. In this case, the mean ripples values are slightly better when feature 2 is included, with no serious implications in overall process performance.

As a second application example, we want to synthetize an optical filter based on a MZI interferometer. In this case, we are going to specifically select the TBUs that we want to enable during the circuit optimization to achieve a lower-dimensional optimization variable space. This is particularly interesting in large-scale arrangements. Looking at the mesh depicted in Fig. 4(a) we can employ TBUs A7 and B10 as the input and output couplers of the interferometric structure, a long arm including TBUs C7, A6, B6, C10 and a shorter arm including TBUs B7, A10. As in the 1×8 beam splitter, we start by defining a new cost function. For this application, typical features of optical filters are Extinction Ratio (ER), insertion loss (IL) and bandwidth. After testing with more than 100 cost

functions alternatives developed by choosing different features and weights, we appreciate the importance on the final selection on elevating the success or convergence rates. Indeed, we saw that in most iterations, minimizing some of the features led to an increase of the others, being the origin of noisy gradients if weights (c_i) are not properly selected. Here one can obtain more optimum cost functions and/or use alternative computational optimization algorithms like conjugate gradient descent, RMSProp, Adam, to cite a few [34]. As an alternative, we can develop specific routines for each application. For example, in this case, we developed a routine that employ two cost functions. The employed cost function for each iteration is selected depending on the current performance conditions, leading to better results when compared to straightforward use of a common CF. This approach, however, requires some knowledge about the fundamentals of the targeted circuit behaviour. Precisely, in this application example we employ a cost function that optimize the channel loss of the MZI filter when the channel loss is greater than 1.5 dB and selects the second cost function enabling a single TBU for the optimization of the ER.

$$\left.\begin{array}{l} c_1 = 0.005,\ c_2 = 0, \quad \text{if } f_2 < -1.5 \text{ dB} \\ c_1 = 0,\ c_2 = 0.5, \qquad \text{if } f_2 > -1.5 \text{ dB} \end{array}\right\}$$

$$f_1 = \begin{pmatrix} \sum_{op} \max\left(10\log_{10}\left(|H_{op,4}|^2\right)\right) \\ -\min\left(10\log_{10}\left(|H_{op,4}|^2\right)\right) \end{pmatrix},$$

$$f_2 = \sum_{op}\left(10\log_{10}\left(|H_{1,4}|^2\right)\right), \tag{10}$$

As an example of the obtained results, Fig. 7(a) illustrates the targeted MZI structure and the employed TBUs. Fig. 7(b) illustrates the achieved filter performance, where the ER and IL parameters are depicted. Finally, Fig. 7(c) illustrates the changes in the coupling response of each TBU involved in the optimization process.

As for the required reconfiguration times of the mesh, they are application sensitive. For the two approaches followed in this section, the starting driving configuration will set the initial $\boldsymbol{v}$. Then, the targeted application in the CF shape will require a certain amount of iterations to converge (N_{iter}). Each iteration internally computes the gradient with a number of operations approximately equal to the number of enabled actuators (N_{act}). Note that the size of $\boldsymbol{v}$ is the double of the number of TBUs participating in the optimization process. For each actuation, the required time delay (τ) to tune the actuator, getting the necessary data from the readouts to build-up the cost functions and save the data is technology dependent. Thus, the total reconfiguration time is:

$$RT = N_{iter} N_{act} \tau, \tag{11}$$

Assuming an approximated τ of 1 ms, the reconfiguration times in the previous 1×8 beam splitter example results in a time delay in the range of 5.76 ms and 21 ms. These figures are doubled if central differences are employed, as in (7). The analytical model employed requires an approximated τ of 175 ms for 5 wavelength points. That translates into computational times of few minutes.

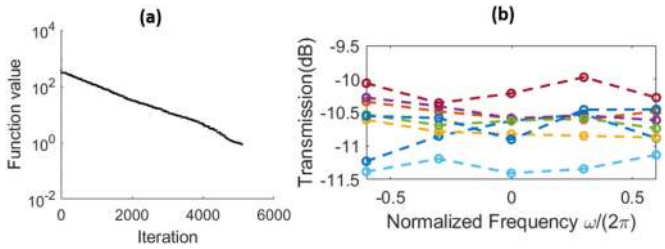

Fig. 8. Results of the N-M algorithm in the 1×8 beamsplitter example: (a) Cost function minimization versus iterations, (b) final normalized frequency response normalized to one single TBU delay. Only five spectral points are considered per channel.

B. Non-Derivative Methods (Zero-Order Optimization Algorithms)

As the reader can infer, getting the gradient straightforwardly in a real waveguide mesh system, as proposed, means performing the perturbation of one actuating variable, getting the associated CF from the post-processed signal from the readout system, repeating the procedure with a negative perturbation and then computing (6-7). Then, this procedure is iterated for every optimization variable till getting the full gradient.

An alternative to the gradient computation is to employ non-derivative computational optimization methods, like the Nelder-Mead simplex algorithm, (N-M) [35]. It consists on a direct search computational optimization method that evaluates the cost function (10) in the variable space and select the next evaluation point by comparison among a set of previously selected points. In this section we applied the N-M algorithm to the previous 1×8 example. Here, the stopping criteria relates to the decrement in the CF for a certain number of iterations. First, we applied to 100 trials of random mesh initialization a softer criterion where the process exits if the cost function is not reduced by 0.05 during 500 iterations. Then, we test another 100 trials of the method where the procedure terminates if the cost function is not reduced by 0.01 during 500 iterations. In addition, the vector $\boldsymbol{v}$ is the active driving phase of each phase actuator.

An example of the optimization process is illustrated in Fig. 8 for one of the cases employing the softer stopping criteria. We can see that the targeted performance has been achieved properly, obtaining both spectrally flat responses and optical powers close to -10 dBm.

Fig. 9 illustrates the statistical analysis performed over the 100 optimization trials. We can see that in this case, the use of a harder criterion results in better ripple reduction in the optical channel and better optical output approximation to -10 dB. Remember that the optical output error is given by feature 1 in (9). The cost to pay comes with an increment in the number of iterations. As opposed to derivative methods, here the number of operations is close to the number of iterations.

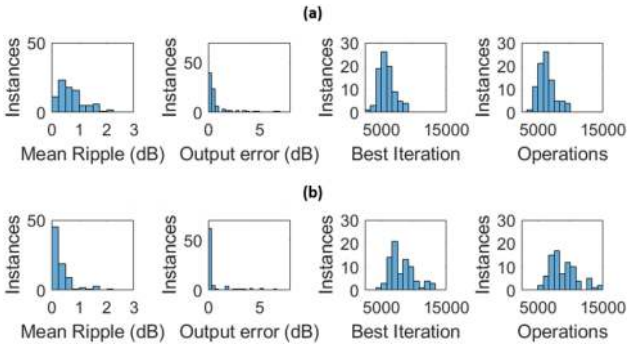

Fig. 9. Statistical results obtained after performing the optimization process of the 1 × 8 beam splitter 100 times employing a non-derivative method. (a) softer stopping criteria, (b) harder stopping criterion.

IV. Discussion

In Section II we introduced a new approach to address the footprint limitation of large-scale waveguide mesh arrangements. However, a more limiting issue arises from enabling the practical driving of a large number of photonic actuators simultaneously. The methods proposed in Section III enables a key feature in software-based programmability and merge the study of photonic integrated circuits with other interesting scientific and technological fields like system optimization, machine learning and artificial intelligence.

Precisely, the gradient descent and the N-M methods employed in this work are simple routines and have been selected to validate the application of optimization algorithms for controlling waveguide mesh arrangements. However, the selection of the hyper-parameters and cost functions play a critical role in the final performance and convergence speed. Sometimes they are selected after a grid-search, random-search or Bayesian optimization [36]. In order to relax both the selection of the hyperparameters and the cost function engineering, the application of advanced algorithms and customized routines have been widely employed in other research fields [34], [36]. For example, algorithms with adaptative learning rates mitigate the fact that the cost is often highly sensitive to some directions in parameter space and insensitive to others [36]. As another example, the use of genetic algorithms seems a promising approach to avoid standstill states when approaching local minima during the PIC configuration process, and thus, relaxing the cost function selection.

Applying these solutions and exploring alternative optimization routines will enable reliable self-configuring and self-healing photonic integrated circuits with faster reconfiguration rates.

V. Conclusion

In this paper we have reviewed the state-of-the-art of general-purpose waveguide mesh arrangements. These architectures are called to play a key role in future programmable photonic integrated circuits. However, the best performance is offered by an architecture that is more space-consuming when compared to the other alternatives. To compensate this drawback, we have proposed a novel architecture that while implementing the same optical interconnection scheme allows the integration of every tuning element following the same longitudinal axis orientation. This innovation has been presented for all the existent mesh topologies and is expected to reduce the scalability issues and enhance the circuit robustness and performance.

Next, the large-scale integration of general-purpose waveguide meshes is currently limited by our capability to properly drive all the units automatically to get a desired functionality. In this work we have proposed the use of computational optimization methods as the basis to configure automatically the circuits. The actuation variables are optimized to reduce a cost function that is application- dependent. We provided a proof-of-concept that both gradient-based algorithms and non-derivative methods can succeed on this task. In the light of these results, a new generation of self-optimized photonic integrated circuits based on general-purpose waveguide mesh arrangements is envisioned. Future actions should be focused on applying alternative algorithms that allows a reduction on the number of iterations for noisy cost functions and on the development of more efficient cost functions.

Acknowledgment

The author acknowledges his ERC-Advance-Grant-UCHIP-team members for the fruitful discussions.

References

[1] J. Lyke, C. G. Christodolou, G. A. Vera, and A. H. Edwards, "An introduction to reconfigurable systems," *Proc. IEEE*, vol. 103, no. 3, pp. 291–317, Mar. 2015.
[2] J. Capmany and D. Pérez, *Programmable Integrated Photonics*. London, U.K.: Oxford Univ. Press, 2019.
[3] S. M. Trimberger, "Three ages of FPGAs: A restrospective on the first thirty years of FPGA technology," *Proc. IEEE*, vol. 103, no. 3, pp. 318–331, Mar. 2015.
[4] P. Samadi, D. Calhoun, H. Wang, and K. Bergman, "Accelerating cast traffic delivery in data centers leveraging physical layer optics and SDN," in *Proc. Int. Conf. Opt. Netw. Des. Model.*, 2014, pp. 73–77.
[5] J. Mitola, "The software radio architecture," *IEEE Commun. Mag.*, vol. 33, no. 5, pp. 26–38, May 1995.
[6] M. Hochberg and T. Baehr-Jones, "Towards fabless silicon photonics," *Nature Photon.*, vol. 4, pp. 492–494, 2010.
[7] J. G. M. Van der Tol *et al.*, "Indium phosphide integrated photonics in membranes," *IEEE J. Sel. T Quantum Electron.*, vol. 24, no. 1, Jan./Feb. 2018, Art. no. 6100809.
[8] D. Inniss and R. Rubenstein, *Silicon Photonics: Fueling the Next Information Revolution*. San Mateo, CA, USA: Morgan Kaufmann, 2016.
[9] D. Pérez, I. Gasulla, and J. Capmany, "Software-defined reconfigurable microwave photonics processor," *Opt. Express*, vol. 23, pp. 14640–14654, 2015.
[10] D. Pérez, I. Gasulla, and J. Capmany, "Field-programmable photonic arrays," *Opt. Express*, vol. 26, pp. 27265–27278, 2018.
[11] L. Zhuang *et al.*, "Programmable photonic signal processor chip for radiofrequency applications," *Optica*, vol. 2, pp. 854–859, 2015.
[12] D. Pérez *et al.*, "Multipurpose silicon photonics signal processor core," *Nature Commun.*, vol. 8, 2017, Art. no. 636.
[13] D. Pérez and J. Capmany, "Scalable analysis for arbitrary photonic integrated waveguide meshes," *Optica*, vol. 6, no. 1, pp. 19–27, 2019.
[14] D. Pérez *et al.*, "Integrated photonic tunable basic units using dual-drive directional couplers," *Opt. Express*, submitted for publication.
[15] M. Reck, A. Zeilinger, H. J. Bernstein, and Y. P. Bertani, "Experimental realization of any discrete unitary operator," *Phys. Rev. Lett.*, vol. 73, no. 58, pp. 58–61, 1994.
[16] J. Carolan *et al.*, "Universal linear optics," *Science*, vol. 349, no. 711, pp. 711–716, 2015.

[17] P. L. Mennea *et al.*, "Modular linear optical circuits," *Optica*, vol. 5, no. 9, pp. 1087–1090, 2018.
[18] N.C. Harris *et al.*, "Quantum transport simulations in a programmable nanophotonic processor," *Nature Photon.*, vol. 11, pp. 447–452, 2017.
[19] Y. Shen *et al.*, "Deep learning with coherent nanophotonic circuits," *Nature Photon.*, vol. 11, pp. 441–446, 2017.
[20] A. Ribeiro *et al.*, "Demonstration of a 4×4-port universal linear circuit," *Optica*, vol. 3, no. 12, pp. 1348–1357, 2016.
[21] A. Annoni *et al.*, "Unscrambling light—Automatically undoing strong mixing between modes," *Light: Sci. Appl.*, vol. 6, no. 12, 2017, Art. no. e17110.
[22] D. Pérez, I. Gasulla, J. Capmany, and J. R. A. Soref, "Reconfigurable lattice mesh designs for programmable photonic processors," *Opt. Express*, vol. 24, pp. 12093–12106, 2016.
[23] D. Pérez *et al.*, "Silicon photonics rectangular universal interferometer," *Lasers Photon. Rev.*, vol. 11, 2017, Art. no. 1700219.
[24] D. Pérez, E. Sánchez, and J. Capmany, "Programmable true time delay lines using integrated waveguide meshes," *J. Lightw. Technol.*, vol. 36, no. 19, pp. 4591–4601, Oct. 2018.
[25] M. Rudé *et al.*, "Optical switching at 1.55μm in silicon racetrack resonators using phase change materials," *Appl. Phys. Lett.*, vol. 103, no. 14, 2013, Art. no. 141119.
[26] J. Zheng *et al.*, "GST-on-silicon hybrid nanophotonic integrated circuits: A non-volatile quasi-continuously reprogrammable platform," *Opt. Mater. Express*, vol. 8, no. 6, pp. 1551–1561, 2018.
[27] R. J. Trudeau, *Introduction to Graph Theory*. New York, NY, USA: Dover, 1993.
[28] S. Ibrahim *et al.*, "Demonstration of a fast-reconfigurable silicon CMOS optical lattice filter," *Opt. Express*, vol. 19, pp. 13245–13256, 2011.
[29] S. Romero-García *et al.*, "Wideband multi-stage CROW filters with relaxed fabrication tolerances," *Opt. Express*, vol. 26, pp. 4723–4737, 2018.
[30] D. A. Miller, "Setting up meshes of interferometers—Reversed local light interference method," *Opt. Express*, vol. 25, no. 23, pp. 29233–29248, 2017.
[31] D. A. B. Miller, "Self-configuring universal linear optical component," *Photon. Res.*, vol. 1, pp. 1–15, 2013.
[32] D. Pérez, "Integrated microwave photonic processors using waveguide mesh cores," Ph.D. dissertation, Dept. Commun., Univ. Politècnica de València, Valencia, Spain, 2017.
[33] J. J. Ackert *et al.*, "High-speed detection at two micrometres with monolithic silicon photodiodes," *Nature Photon.*, vol. 9, no. 6, 2015, Art. no. 393–396.
[34] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[35] D. M. Olsson and L. S. Nelson, "The Nelder-Mead simplex procedure for function minimization," *Technometrics*, vol. 17, no. 1, pp. 45–51, 1975.
[36] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer, 2006.

Daniel Pérez López was born in Yecla, Murcia, Spain. He received the bachelor's degree in telecommunications engineering, the master's degree in technologies, systems, and communications networks, and the Ph.D. degree in communications from the Universitat Politècnica de València (UPV), Valencia, Spain, in 2014, 2015, and 2017, respectively. In 2016, he had a brief stint at the Optoelectronics Research Centre, University of Southampton, U.K. He is currently a full-time Postdoctoral Researcher with the Photonics Research Labs Research Centre, under the European Project ERC-ADG-2016 UMWPCHIP. He has authored/coauthored 60 scientific contributions in international magazines and congresses, focusing on programmable photonic integrated processors. He was the recipient of the National COIT-AEIT Award for the Best Final Career Project and the Best Doctoral Thesis in Fundamentals and Basic Technologies of Information and Communications, and Its Applications in 2015 and 2018, respectively. His doctoral thesis was also awarded by the Institute of Electrical and Electronics Engineers Photonics Society with the International Best Student Graduate Fellowship Award in 2017, and the Extraordinary Doctoral Thesis UPV awards in 2019.