

Simulation of Highly Coupled Programmable Photonic Circuits

Erica Sánchez , Graduate Student Member, IEEE, Aitor López , Graduate Student Member, IEEE, and Daniel Pérez-López , Member, IEEE

Abstract—Multipurpose programmable photonic processors have recently emerged as a solution to provide cost-effective and general-purpose functionality for a myriad of photonic applications. Inspired by programmable electronics, the design of these devices lies on generic photonic integrated hardware based on two-dimensional waveguide meshes comprised of tunable couplers and phase actuators. Extending these meshes allows the implementation of more complex structures and functionalities. However, there are design trade-offs arising from parasitic effects coming from fabrication and their dynamic operation. Since the time spent during the design cycle and validation of these complex systems usually involves a costly, risky and time-consuming processes, this work proposes and compares two simulation tools to predict the spectral response of any 2D integrated photonic mesh circuit composed of an arbitrary number of coupled cells. These methods reduce development costs, speed up the growth of new circuit designs, and are a fundamental tool for the development of programmable photonic libraries.

Index Terms—Computational photonics, integrated optics, programmable photonics.

I. INTRODUCTION

PROGRAMMABLE multifunctional photonics (PMP) enables the configuration of multiple optical and electro-optical processing operations employing reconfigurable photonic integrated platforms [1], [2]. The core of the architecture relies on the interconnection of Programmable Unit Cells (PUC) -which can be implemented with Mach-Zehnder interferometers [4] or programmable directional couplers [4], [5]. The architecture includes two phase actuators that can rely on a wide

Manuscript received 4 May 2022; revised 23 June 2022; accepted 24 July 2022. Date of publication 29 July 2022; date of current version 3 October 2022. This work was supported in part by the Spanish MINECO Juan de la Cierva Program, ERC Advanced under Grants ERC-ADGB2016-741415 UWMP-Chip and PROMETEO/2021/015, in part by Disruptive Photonic Technologies for 2030 under Grant IDIFEDER/021/050, and in part by Infraestructura para caracterización de Chips Fotónicos under Grants EQC2018-004683-P and H2020-ICT-2019-2-871330 NEOTERIC. The work of iPronics team was supported in part by European Union's Horizon Europe SME Research and Innovation Program under Grant Agreement 101057934, and in part by National CDTI Neotec 2020. (Erica Sánchez and Aitor López contributed equally to this work.) (Corresponding author: Daniel Pérez-López.)

Erica Sánchez and Aitor López are with the iTEAM Research Institute, Universitat Politècnica de València, Camino de Vera s/n 46022 Valencia, Spain (e-mail: ersango@iteam.upv.es; ailoher@iteam.upv.es).

Daniel Pérez-López is with the iTEAM Research Institute, Universitat Politècnica de València, Camino de Vera s/n 46022 Valencia, Spain, and also with the iPronics, Programmable photonics S.L., Camino de Vera s/n 46022 Valencia, Spain (e-mail: dperez@iteam.upv.es).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/JLT.2022.3194973>.

Digital Object Identifier 10.1109/JLT.2022.3194973

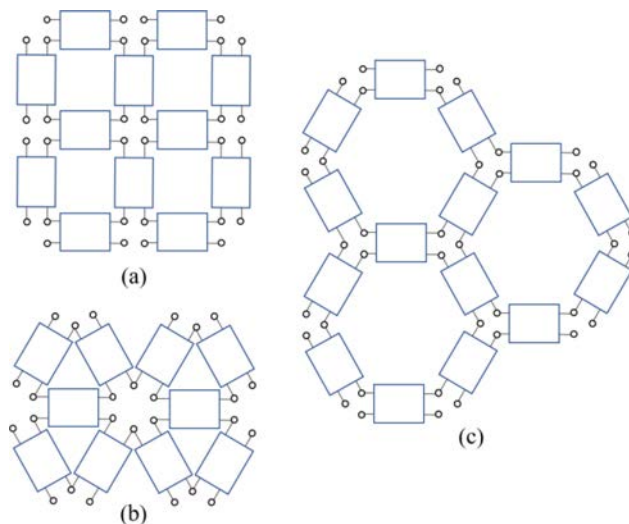


Fig. 1. Joint representation of several reported mesh topologies. (a) Rectangular mesh, (b) Triangular mesh and (c) Hexagonal mesh.

range of physical effects. Depending on the interconnection pattern, different programmable photonic circuit topologies has been demonstrated. Fig. 1 illustrates examples of hexagonal [7], [8], square [3] and triangular [4], [6] arrangements. The resulting architecture enables the synthesis and discretization of optical circuit topologies within the programmable photonic cells, originating multipurpose photonic processors or Field Programmable Photonic Gate Arrays (FPPGAs) [9].

The application of these programmable photonic circuits provides a potential alternative to application-specific photonic integrated circuits (ASPICs) implementing the main required functionalities in microwave photonic systems and radio-over-fiber transmission [3], [10]. It also supports the emulation of arbitrary linear transformations, a key operation in many other fields of application such as quantum processing [8], [11]–[18], boson sampling [19]–[20] and neuromorphic computing [21], [22]. Hence, the support of these features -amongst many others- will allow this technology -in cooperation with current digital processing solutions- to become a key enabler of additional, enhanced signal processing schemes and applications.

However, the performance of some of the previous applications is associated to the scalability and upgrade of current programmable circuits. In practice, there exist several limitations: footprint optimization [23], accumulated losses, imperfect coupling splitting ratios [24], phase control, parasitic

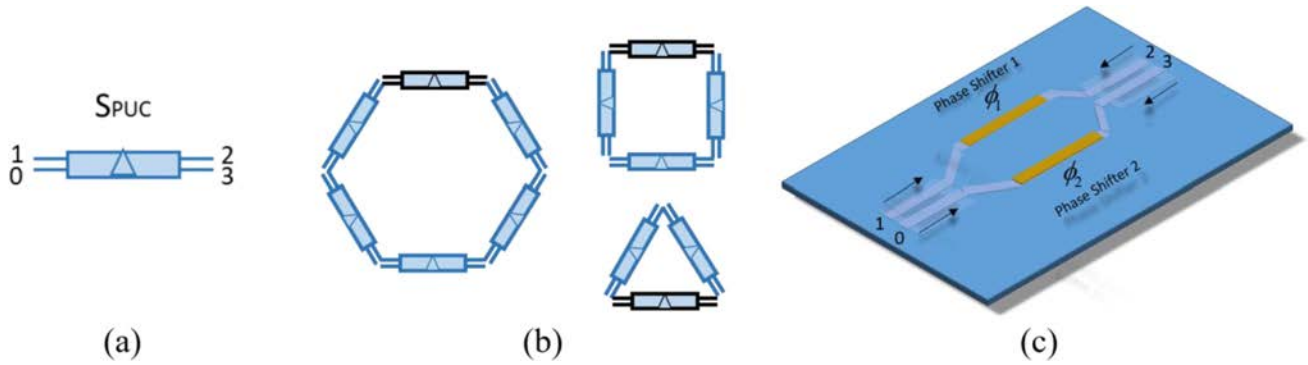


Fig. 2. (a) Lattice/PUC used as a basic building block, (b) Interconnection of several PUCs forming closed cells with different topologies (hexagonal, squared and triangular) using a single PUC as a basic building block and (c) MZI used to implement each programmable unit cell.

back-reflections, loss imbalances, fabrication errors (gradients through the circuit in thickness or temperature) and drift in time [26]. This calls for efficient alternatives to check the viability of such circuits when employing current fabrication techniques and imperfect photonic components, as well as to provide a means for carrying out statistical analysis of their targeted performance to predict and avoid any possible issues arising, leading to significant cost reduction in hardware fabrication. In addition, the availability of reliable and efficient circuit simulation tools is mandatory to sustain and foster the development of programmable photonic circuit routines and automated tools.

On this front, the work proposed in [26] presented a simulation method for arbitrary photonic waveguide meshes and demonstrated its performance. The work relied on the inductive method derivation employing subsets (lattices) of waveguide meshes as primitive cells. The use of subsets limited the arbitrariness of the simulation method and demanded considerable customization efforts for alternative architectures. Moreover, computation efficiency was not reported.

In this work, we propose two alternative simulation approaches that aim to reproduce the spectral response of arbitrary photonic circuit arrangement based on hyper-coupled cells. Section II presents the first approach, inspired by the work done in [26], including the derivation and application of the inductive method to a single-cell approach. The second one, presented in Section III, tackles this issue from a graph theory perspective following the work introduced in [27]. In addition, in Section IV, we analyze and compare their computational elapsed times and key benefits and limitations under different application scenarios. In short, the performance of each method is shown in Section IV.

II. INDUCTIVE SINGLE-CELL APPROACH

The first method that we present in this paper pursues the simulation of the frequency response (scattering matrix) of large-scale 2D waveguide meshes formed by Programmable Units Cells. Following a similar strategy as in [26], we employ a mathematical inductive computation [28] that builds up the mesh iteratively adding pieces sequentially. The key differentiating factor is that previous works employ a set of PUCs connected in

a Y-shape (trilattice) as a basic building block. Here, we employ a single PUC as the basic building block, allowing us to re-use the model for arbitrary waveguide topologies, Fig. 2(a). This enables the implementation of meshes with different topologies (hexagonal, square, triangular, feedforward etc. (see Fig. 2(b)) without the need to change the shape of the block that we are using as a base of the inductive method. For the remaining of the work, the single-cell approach is demonstrated on hexagonal meshes.

To start, we need to define and model the scattering matrix of the PUC. This 2x2 device must be able to provide arbitrary splitting factors and phase response through the control of its phase actuators. The model of the PUC depends on its architecture. Specifically, it can be implemented with beamsplitters, phase actuators or tunable beamsplitters [3]–[5]. However, the classical configuration employs a Mach-Zehnder Interferometer topology (Fig. 2(c)).

Moreover, to simplify the modeling, we can consider negligible backscattering or signal reflections as these will not dominate over optical crosstalk contributions. Equation (1), represents the 4x4 scattering matrix (S_{PUC}) that models the internal behavior of each PUC.

$$\begin{aligned}
 S_{PUC} &= \begin{bmatrix} S_{00} & S_{01} & S_{02} & S_{03} \\ S_{10} & S_{11} & S_{12} & S_{13} \\ S_{20} & S_{21} & S_{22} & S_{23} \\ S_{30} & S_{31} & S_{32} & S_{33} \end{bmatrix} = \\
 &= -je^{-j\Delta} \begin{bmatrix} 0 & 0 & \cos \theta & -\sin \theta \\ 0 & 0 & \sin \theta & \cos \theta \\ \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \end{bmatrix} \cdot \alpha e^{-j\omega\tau}
 \end{aligned} \quad (1)$$

where Δ is the sum of the phases of each arm of the MZI (ϕ_1 and ϕ_2) divided by 2, θ is the subtraction of these phases divided by 2, α are the Insertion Losses (IL) of the couplers, ω is the optical frequency and τ is the propagation delay.

We then compute the scattering matrix of every PUC, employing their phase configurations. Next, we proceed with the inductive method employing the scattering matrix of n-order mesh using the matrix $S(n-1)$ of the immediately previous mesh

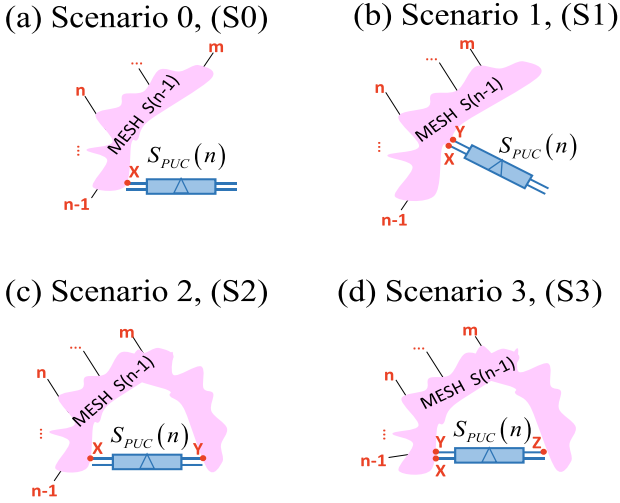


Fig. 3. Description of the inductive method for obtaining the scattering matrix of a hexagonal 2D waveguide mesh composed of n PUCs by adding a unit $S_{PUC}(n)$ to a hexagonal 2D waveguide mesh composed of $n-1$ PUCs. (a) Interconnection scenario 0. (b) Interconnection scenario 1. (c) Interconnection scenario 2. (d) Interconnection scenario 3.

and $S_{PUC}(n)$ of the new block to be connected. The calculation process depends on how the new block is connected to the older structure - the number of new ports added to the mesh by performing a new join- and whether a cell is being closed. As a result, there are four possible interconnection scenarios between the new additional lattice and the previous lower-order mesh. As shown in Fig. 3., these scenarios are classified according to the number of ports that are being used to make the connection. Note that although the same scenarios are described in [26], the resulting equations will be different due to the change on the minimum block architecture employed.

Scenario 0 (S0): This case represents the union of the new lattice with the structure of order $(n-1)$ through a single port, X (see Fig. 3(a)). Port number 1 of the PUC is used as a connection node X with the previous mesh or structure. Therefore, the number of ports of the resultant structure is increased by two, thus increasing the size of the scattering matrix of the new mesh. This scenario is the easier case that we can use to enlarge the mesh.

Scenario 1 (S1): As we can notice in Fig. 3(b), the union nodes between the structures are two, X and Y . Ports 0 and 1 on the same side of the PUC are used to perform the connection and the resulting arrangement does not increase the number of free ports that it has.

Scenario 2 (S2): Here, the addition of the new PUC does not increase the number of free ports either. Again, two nodes, X and Y , form the union but these are located one on each side of the PUC (see Fig. 3(c)). Should be noted that in this scenario, the number of closed cavities/feedback loops increase by one. This scenario will originate recirculation of signals between the interface nodes and the newly added lattice.

Scenario 3 (S3): In S3, not only the number of free ports in the mesh is not increased, but the resulting structure also has two fewer free ports. As we can observe in Fig. 3(d), in this case, 3 ports, X , Y and Z , are used for the addition. Again, the possible

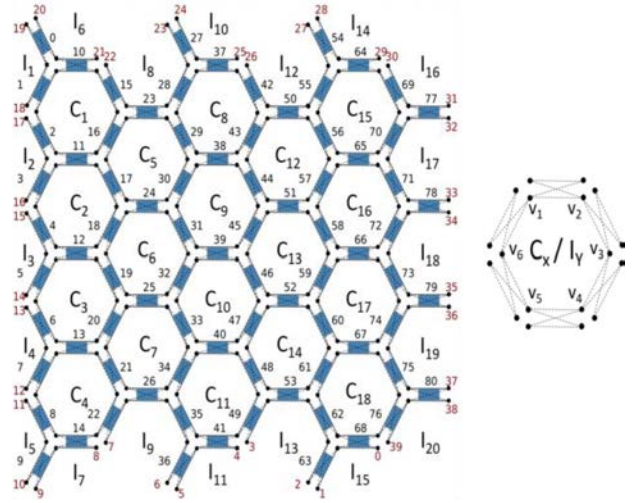


Fig. 4. Joint portrayal of an 81-PUC waveguide mesh along with its graph representation. We denote each vertex by referring to its constituting cell index (' C_x/I_y '), both numbered from top to bottom and from left to right) and its position inside of it, numbered clockwise. Some examples: upper left port from PUC 51 would be referred to as ' $C_{12}v_5$ ', while ports 27 and 28 of the waveguide mesh would be ' $I_{12}v_2$ ' and ' $I_{14}v_6$ '.

recirculation between the nodes X and Z , and Y and Z of the interface and the new PUC joined.

It is important to mention that, in all scenarios, the unions will always be made through the last port(s) of the mesh $S(n-1)$. The detailed procedure to follow in each case, the resulting equations and graphs that allow the computation of the final dispersion matrix $S(n)$ are provided in Appendix, including pseudocode to facilitate programming. The programming and simulation have been carried out in Python programming language.

The performance of this algorithm can be summarized as a methodology to compute the full scattering matrix that is not computationally time-dependent on the system configuration. Its application and benchmark are presented in Section IV.

III. GRAPH-BASED APPROACH

We propose in this section another simulation tool for the synthesis of photonic circuits in a waveguide mesh arrangement based on graph theory. As the name implies, such discipline relates to the study of graphs, mathematical structures consisting of a set of vertices (also called nodes or points) connected by a set of edges (also called links or lines) with associated weights, or costs [29].

Any interferometric structure can be displayed by its reciprocal graph representation, as proposed in [27], [30]. Indeed, the programmable unit cell can be modeled as a set of two-input, two-output vertices (corresponding to its input and output optical ports) and four edges reproducing the device's internal connections. The weights of those edges are defined in [27] as transmission distances (TD) and can refer to any figure of merit (power consumption, insertion loss, basic unit delay, transfer function, etc.). We can observe in Fig. 4 how these structures can be arranged to model the original waveguide mesh topology. We denote every graph vertex (optical connection) by a set of

two indices: a first one referring to the closed polygon to which it belongs and a second one used to identify its position inside of it (oriented clockwise). To identify all input/output optical ports at the mesh perimeter, we define an additional set of ‘imaginary’ cells (I_1 - I_{20}) surrounding it.

Based on this information, we can run a script based on the pathfinder algorithm presented in [27] to retrieve all optical paths connecting one input and one output node from the waveguide mesh during a given timeframe. This is a key difference concerning more ‘classic’ shortest-path evaluation approaches, which usually deliver one single path for which the sum of the weights of its constituent edges is minimized.

In this work, we define the TD as the spectral response of the PUCs at the central wavelength. Thus, we can estimate the overall accumulated cost of any path traversing the graph by directly multiplying their spectral responses at such wavelength.

We can accelerate the process by defining a power threshold (in amplitude). In such a way, any path featuring an optical power below this value will be discarded and therefore we prevent that branch from extending further through the mesh. This can be particularly useful for large arrangements, as only a few branches will survive after a few iterations, thus reducing the computation time significantly. However, this parameter must be carefully elected as studied in the next section.

The performance of this algorithm can be summarized as a methodology to compute a subset of a scattering matrix that is computationally time-dependent on the system configuration. Its application and benchmark are presented in the following section.

IV. RESULTS

We present a scalability statistical analysis between the approach proposed in [26] with the two methods presented in this work for different mesh configurations.

To perform a fair and practical benchmark, it is important to first study the dependency of the graph-based approach with the *threshold* parameter with the simulation error and the computational time. To do so, we configured a set of programmable photonic processors with different sizes (134, 198, 397 and 599 PUCs) to emulate a Ring-Assisted MZI filter (RAMZI). Next, we computed the frequency response of the system with the inductive method and with the graph-based approach. For the latter, we employed *threshold* levels of -20 , -30 , -40 and -50 dB. Since the frequency response of the inductive method does not employ any hyperparameter, it is considered as the error-free trace.

Fig. 5(a) illustrates the overlap of the frequency response of the RAMZI gathered within the tests. We can observe that the fitting of the inductive method and the graph-based approach degrades for lower values of threshold. On the same line, Fig. 5(b) illustrates the measured elapsed time of the computation and the obtained minimum square error for the different thresholds. To cite an example from the figure, we can observe how a medium-size mesh such as the 397-PUC one features a mean elapsed time of 5.93, 9.44, 13.57 and 16.84 s with associated mean squared errors of 4×10^{-3} , 5×10^{-4} , 8×10^{-5} and 5×10^{-6} respectively

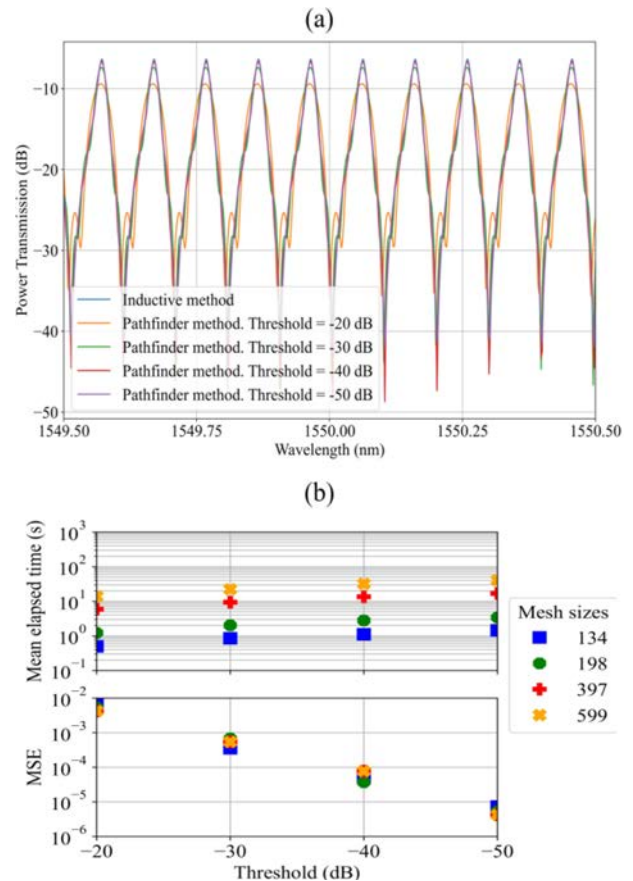


Fig. 5. (a) Comparison of the simulated power spectra between pathfinder method (under several power thresholds) and inductive one for the synthesis of a balanced RAMZI filter in an 81-PUC waveguide mesh. (b) Variation of the elapsed time and the mean square error (MSE) between these methods with power threshold for the synthesis of the same circuit for four different mesh sizes.

for the aforementioned threshold values to produce the RAMZI. All remaining meshes under study follow a similar trend, so we can conclude how the elapsed time increases linearly with the *threshold* value, while the MSE does the opposite exponentially. A MSE below 10^{-3} is sufficient for most applications, as non-ideal effects like optical crosstalk and receiver sensitivities can be dominant. In addition, the threshold selection will be application dependent, since we can expect this parameter to have a significant impact on the synthesis of infinite-impulse response (IIR) filters specially, as their spectral response provides an infinite number of non-zero terms by definition. After repeating the parametric test for the applications studied in this paper we confirmed that the threshold of -40 dB always returned MSE better than $5 \cdot 10^{-4}$. For the remaining section, we employ the threshold of -40 dB.

To end the comparison, it is worth mentioning that the threshold levels of the previous example are translated into a reduction in the number of contributive paths. Precisely, we obtain 4, 6, 8 and 11 paths for the aforementioned threshold levels, respectively.

Next, we apply and compare the two simulation methods to a set of applications to benchmark the elapsed time and illustrate

its applicability. The configuration examples are: all passive state case, an 8x8 unitary matrix, 1x8 beamsplitter, a Side-Coupled Integrated Spaced Sequence of Optical Resonators (SCISSOR) filter and a balanced RAMZI filter. For each case, we sweep over two parameters: the wavelength resolution (going through 3, 11, 101 and 1001 wavelength points for a span of 1 nm centered at 1550 nm) and the number of PUCs, which will determine the mesh size. For all examples (unless otherwise specified) we go through meshes of 34, 36, 45, 72, 81, 87, 134, 198, 397, 599, 799 and 1002 PUCs. The insertion loss of each PUC is configured at 0.2 dB.

Each circuit configuration involves the targeted circuit topologies and the configuration of the access waveguides to east and west ports. Once configured, we compute 10 times each simulation to enable the statistical analysis, employing two equally-equipped desktop computers: 4-core, and 3.60 GHz processors.

The first circuit is the balanced RAMZI filter previously introduced in section III. Fig. 6(a) illustrates its synthesis in an 81-PUC mesh, using ports 15 and 33 as input/output. Hence, the lower arm of the RAMZI would traverse PUCs 33, 40 and 47 after being split in PUC 25 while the upper one would do so through PUCs 32, 39 and 46. The cavity of the structure is configured by PUCs 30, 31, 38, 44 and 45, all in bar state. The simplified schematic is illustrated in Fig. 6(b).

Fig. 6(c) contains the average elapsed time required for each simulation example. First, we can see that the graph-based approach timings overlap for different wavelength resolutions, indicating a very good scalability with the number of points. However, the inductive method, scales linearly with the number of points. As an example, the time of 1001 is 1000 times higher than the 3-point simulation at 1000 PUCs.

Next, if we focus on the elapsed time, the graph-based approach outperforms the inductive method. For 1000-PUC meshes, the performance improvement is 5.6x, 5.8x, 8.3x and 32.8x, for resolutions of 3, 11, 101 and 1001 wavelength points, respectively. We can also observe how it outperforms the trilattice-based approach for large number of wavelength points. The graph-based methodology has found only eight optical paths arriving at the destination port before the power threshold is reached.

Additionally, we tested the handling of non-ideal components during the simulation. In such scenario, PUCs may present a phase response drift (resulting in deviated configurations) as a result of the combined effects of optical, thermal and electrical crosstalk. More information about these effects and how to mitigate it can be found in [26], [33]. To do so, we modelled the coupling coefficients of each of the PUCs forming the RAMZI as truncated gaussian variables centered at their original (ideal) values and featuring a standard deviation (drift) of 0.02. Values lying below 0 (bar state) or above 1 (cross) are ‘mirrored’ by adding or subtracting respectively its absolute value to the original one. As illustrated in Fig. 6(d), the graph-based approach significantly increases its computational time. In particular, focusing on the 600-PUC example, it is around 2.5 times slower and 2.5 times faster than the inductive method for 3 and 1001 points, respectively. In addition, it is 7 times slower than the graph approach with the perfect component case.

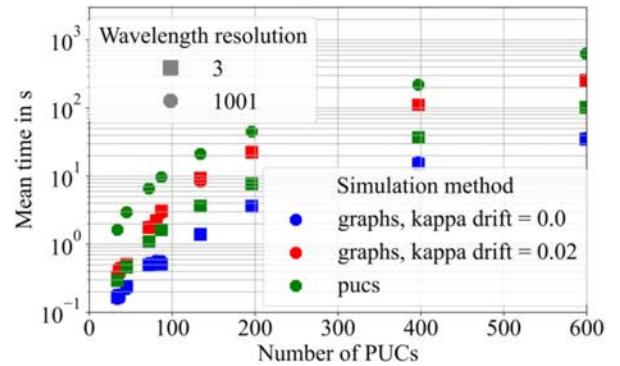
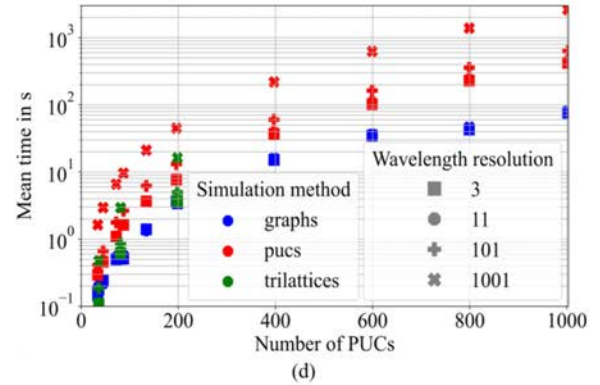
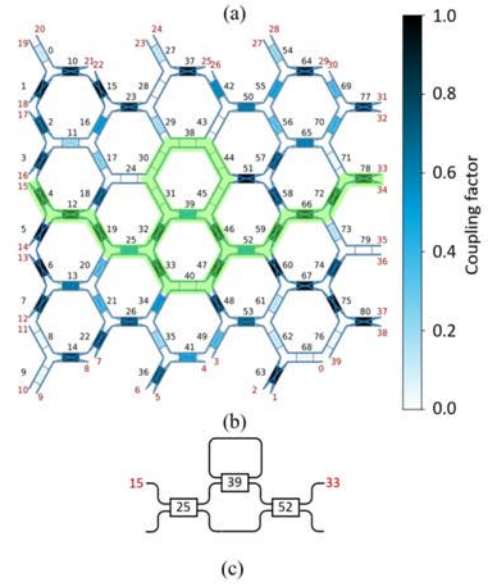


Fig. 6. (a) Synthesis of a balanced RAMZI filter in a simulated 81-PUC waveguide mesh using 15 and 33 as input/output ports, (b) equivalent circuit synthesized, (c) evolution of the mean elapsed time (in seconds) with the number of PUCs for the synthesis of this circuit for four different wavelength resolutions with ideal coupling coefficients and (d) with coupling coefficients modelled as random variables featuring a drift with a standard deviation of 0.02.

Next, we present the simulation results for the optical SCISSOR defined in Fig. 7(a) between ports 12 and 38. To configure this circuit, we set PUCs 2, 7, 11, 13, 17, 20, 30, 33, 44, 47, 57, 60, 65, 67, 70, 75 and 80 to cross state and PUCs 18, 19, 31, 32, 38, 40, 45, 46, 58 and 59 to bar state. At the same time, PUCs 24, 25, 51 and 52 are set in a tunable coupler state to allow light to travel back and forth and be coupled throughout the structure. Fig. 7(b) illustrates the simplified scheme.

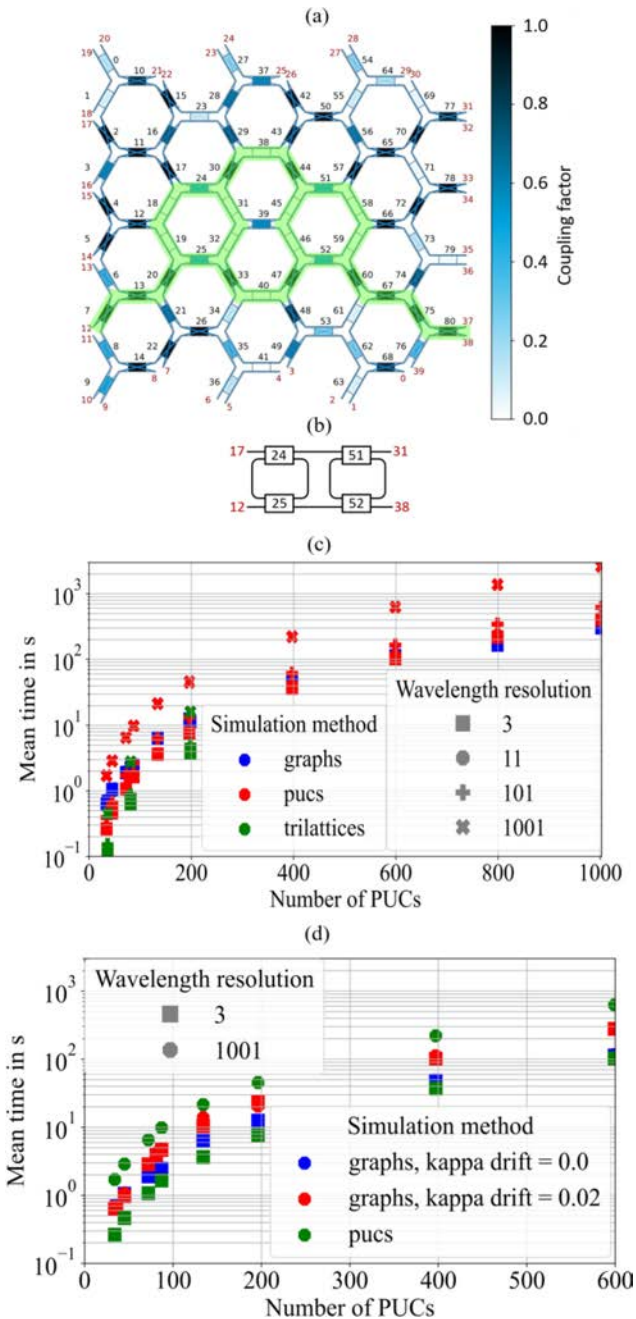


Fig. 7. (a) Representation of a SCISSOR in a simulated 81-PUC waveguide mesh using 15 and 33 as input/output ports, (b) Schematic of equivalent circuit synthesized, (c) Evolution of the mean elapsed time (in seconds) with the number of PUCs for the synthesis of this circuit for four different wavelength resolutions with ideal coupling coefficients and (d) with coupling coefficients modelled as random variables featuring a with a standard deviation of 0.02.

As with the previous example, Fig. 7(c) illustrates the elapsed time for the different simulation approaches. In this case, the graph-based approach, applied to a second order IIR, demands more computational time as the number of power contributions arriving at the output port reaches up to 31. Consequently, the elapsed computation time of graph-based approach is larger than that of a trilattice-based one for smaller

size meshes and lies much closer to that of a single-cell inductive method for larger size meshes and small wavelength resolutions.

Repeating the test for imperfect components with the same methodology as in the previous example, we can observe in Fig. 7(d) that graph-based approach still outperforms inductive single-PUC approach for large wavelength points, offering better scalability. However, for a smaller number of points (3), the inductive method provides a faster computational time. For example, for 600-PUC meshes, the computation times are 275.85 and 103.8 seconds for the graph and inductive approach, respectively. Hence, growing to larger order IIR structures might prevent us from using this strategy and opt for inductive-based approaches.

Moving to a different configuration example, Fig. 8(a) presents a 1x8 beam-splitter, outlined in Fig. 8(b). Such circuit takes port 15 as input and 0, 5, 7, 22, 29, 31, 32 and 33 as outputs. Here, light is split in PUCs 12, 18, 19, 25, 33, 40 and 77, all in Tunable Coupler (TC) state.

To produce the spectral response of this circuit using graph-based approach, we run the single pair algorithm recursively considering the eight output ports. As covered in Fig. 8(c), for the mesh-sizes considered, the graph-based approach performs faster than the inductive one for lower number of spectral points, and much faster for larger spectral points. The efficiency of graph-based approach for this example relies on the rapid discarding process of optical paths during the execution of the algorithm. This no longer occurs, however, if we introduce non-ideal components such as in Fig. 8(d). In this case, graph-based approach's computation time increases dramatically as a result of the appearance and spreading of multiple paths during the execution of this method that may not be discarded up to final stages of its execution. As an example, for the 600-PUC mesh and larger spectral vectors (1001 wavelength points), the graph-based approach requires around 1.6x more time than the inductive method, while the gap becomes of 10x for lower (3) number of points. This particular application demands higher resolution and number of points when validating the channel ripple conditions. Otherwise, the expected spectral behavior is inherently flat. The presence of non-ideal configurations and the increment of optical outputs favor the selection inductive method approach.

The next example focuses on multiple input – multiple output applications. In particular, we configure the 8x8 optical matrix appearing in Fig. 9(a) and represented in Fig. 9(b). Here, we performed simulations for three mesh sizes of 198, 397 and 599 PUCs since it was not possible to allocate this structure in smaller-size meshes. In the figure, mesh ports 33, 34, 35, 36, 37, 38, 39 and 40 work as input ports and mesh ports 1, 2, 3, 4, 5, 6, 7 and 8 operate as output ports. The configuration of an arbitrary linear matrix requires the configuration of the tunable couplers, driving PUCs 10, 11, 12, 13, 28, 29, 30, 43, 44, 45, 46, 61, 62, 63, 76, 77, 78, 79, 94, 95, 96, 109, 110, 111 and 112. In addition, we configure the access waveguides following a similar approach as in the beamsplitter case. The PUCs interconnecting the TC-configured cells are set into cross-state. The configuration of

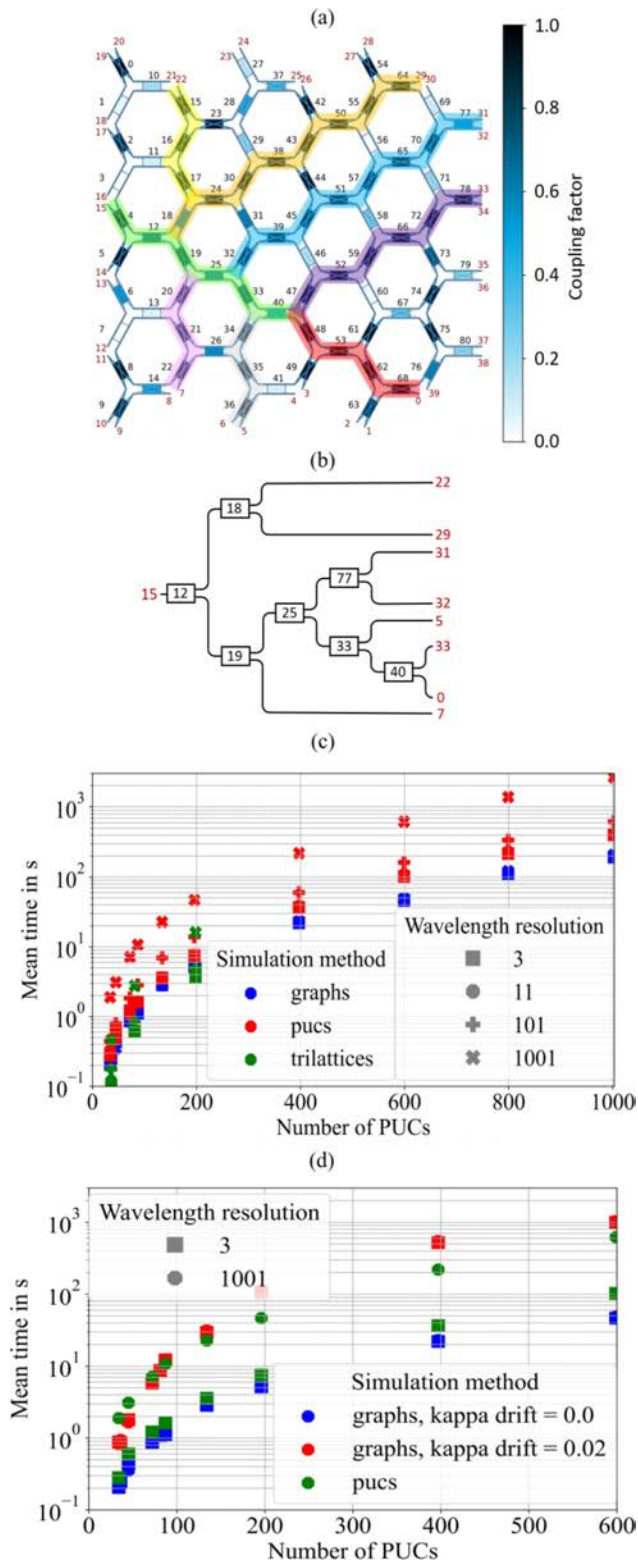


Fig. 8. (a) Representation of a 1x8 beam splitter in a simulated 81-PUC waveguide mesh using port 15 as input port and 0, 5, 7, 22, 29, 31, 32 and 33 as output ports, (b) Schematic of equivalent circuit synthesized, (c) Evolution of the mean elapsed time (in seconds) with the number of PUCs for the synthesis of these circuits four different wavelength resolutions with ideal coupling coefficients and (d) with coupling coefficients modelled as random variables featuring a standard deviation of 0.02.

the TCs are selected employing random distributions to maintain the arbitrariness of the configuration process.

In this case, the graph-based approach is run iteratively for each input/output pair to complete the triangular matrix of the resulting scattering matrix. The results, in Fig. 9(c), illustrate how the elapsed time of the graph approach is around 2 and 3 orders of magnitude larger than that of the inductive method for the larger and shorter wavelength vectors, respectively.

Finally, we conclude our set of experiments by simulating the full response (i.e., using all mesh ports as inputs and outputs) of the waveguide mesh under a random configuration. This represents a scenario with arbitrary complex configurations present on certain applications or uncalibrated circuits. An interesting application of this feature deals with neuromorphic computing engines [22], [31]. The results of the analysis are presented in Fig. 10. For this application, motivated by the exigent computational times of the graph-based approach, we selected mesh sizes up to the 196-PUC waveguide mesh only.

As in the previous case, the graph-based approach runs iteratively for each input/output pair to complete the triangular matrix of the targeted scattering matrix. As for the beamsplitter and the 8x8 circuits, the randomness of PUC transmission states favors the existence of many candidate paths that take long by pathfinder to discard. This effect is exacerbated by the presence of internal feedback loops. Precisely, the performance of the graph-based approach is around 3 and 4 order of magnitude slower than the inductive method approach, for the longest and shortest wavelength vector, respectively.

To summarize the performance of the proposed methods we can conclude the following: First, the inductive method requires the computation of the whole scattering matrix of the structure, even if a subset of matrix elements is required. The computation time of this method is independent from the configuration of the PUCs. In addition, it scales linearly with the number of spectral points.

In contrast, the graph-based is by nature port-pair oriented. However, it can be employed sequentially to extract complete scattering matrices as demonstrated in this work. Multicore electronic processors could be employed to improve the efficiency of this task with parallel processing. The performance of the graph-based approach is significantly invariant with the number of wavelength points and depends on the system configuration. For this reason, the selection of the simulation method is application dependent. For applications with many closed optical paths and lower number of feedback and feedforward loops, the graph-based methodology performs faster simulations. This is also true when only a small subset of elements is required from the scattering matrix. In contrast, the use of the inductive approach provides faster simulation times for arbitrary complex circuits requiring large number of optical ports. Moreover, both methods provide additional flexibility when compared to the existing lattice-based inductive simulation method, allowing the design and its application to arbitrary circuit topologies.

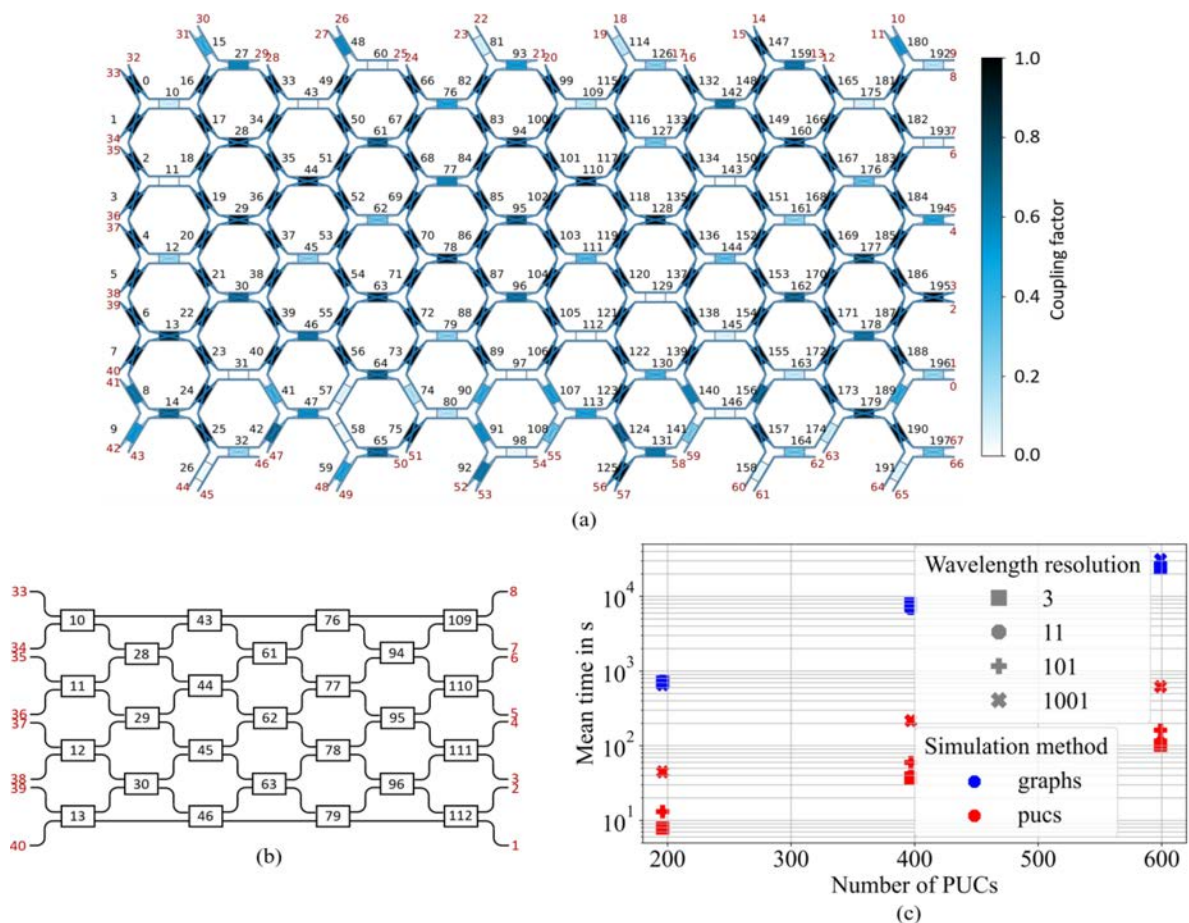


Fig. 9. (a) Representation of an 8x8 identity matrix in a simulated 198-PUC waveguide mesh using ports 33, 34, 35, 36, 37, 38, 39, and 40 as input ports and 1, 2, 3, 4, 5, 6, 7 and 8 as output ports, (b) Schematic of the equivalent circuit synthesized and (c) Evolution of the mean elapsed time (in seconds) with the number of PUCs for the synthesis of these circuits four different wavelength resolutions.

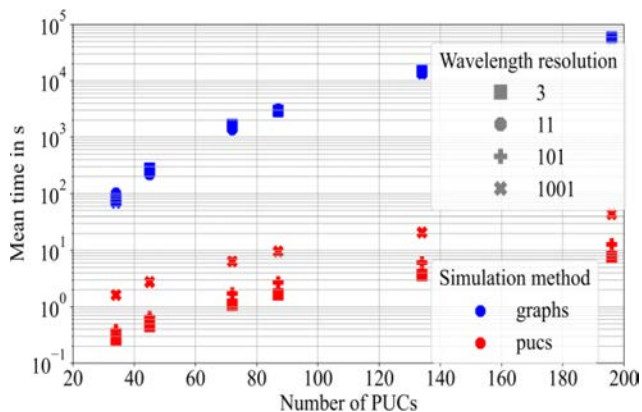


Fig. 10. Evolution of the mean elapsed time (in seconds) with the number of PUCs for the obtention of the response of all ports in a passive waveguide mesh for four different wavelength resolutions.

V. CONCLUSION

The simulation of highly coupled waveguide elements with dynamic configuration and performance is critical for the design

and configuration of scalable programmable circuits. In this work, we have proposed two computational methods based on single-cell inductive approach and graph-based methodologies. Next, we have reported and benchmarked the elapsed computational times and accuracy trade-off of each method for a set of representative application examples and circuit sizes. We concluded that for applications with many closed optical paths and lower number of feedback and feedforward loops, the graph-based methodology performs faster simulations. This is also true when only a small subset of elements is required from the scattering matrix. In contrast, the use of the inductive approach provides faster simulation times for arbitrary complex circuits requiring large number of optical ports. In addition, all methods presented here can be applied to any highly coupled circuit topology (square, hexagonal, triangular, feedforward, etc.). As a consequence, both of them unfold as effective and versatile tools for the study of emerging multipurpose programmable photonic processors. Finally, the expansion of the technology will demand efficient characterization and calibration protocols [32], improved automated configuration routines [33], [34] and resource optimization libraries. These are being developed and will be disseminated in future works.

APPENDIX

SINGLE-CELL APPROACH METHOD AND SCENARIOS

In this appendix, we are going to explain the procedure to obtain the scattering matrix of hexagonal arbitraries waveguide meshes. Then, for each proposed interconnection scenario, its signal flow graph and the new structure scattering matrix obtained after the joint would be described.

Scenario 0: This scenario is the easiest case that we can use to enlarge the mesh. As can be seen in Appendix Fig. 1. (A1), it represents the union of the new lattice with the structure of order (n-1) through a single port, X. Therefore, the number of ports of the resultant structure is increased by two, thus increasing the size of the scattering matrix of the new mesh. The signal flow diagram shown in Appendix Fig. 11. (A2), defines the interconnection possibilities between the n-1 order mesh and the new PUC added through the interface node X. Using the diagram, a system of Equations (A1) is defined, considering all the possible path contributions between the free ports of the older structure. Thus, the coefficients of the scattering matrix that models the mesh are recalculated.

The resulting matrix, shown in Appendix Fig. 11. (A3) can be divided into four submatrices. The first submatrix (Submatrix 1, SM1) defines the coefficients related to the connection between the input-output ports of the n-1 order structure, ergo, it excludes the input-output paths that involve the additional ports that have been created after the union of the new lattice. This means that the SM1 coefficients can be inherited directly from the n-1 matrix we already knew, so we do not need to recalculate them. Afterward, the second case (Submatrix 2, SM2) relates the interconnection creates between the input ports from the older mesh (n-1) and the output ports of the new PUC ($S_{PUC}(n)$). Then, the third (Submatrix 3, SM3) models the paths between input/output ports of the joined lattice. Finally, the last possibility (Submatrix 4, SM4) describes the inputs of the new PUC to the output ports of mesh n-1.

Submatrix1

coefficients:

$$S_{m,n}^N = X' = S_{m,n}^{N-1}$$

Submatrix2

coefficients:

$$S_{\{n\dots n+2\},n}^N = TS$$

Submatrix3

coefficients:

$$S_{\{n\dots n+2\},\{n\dots n+2\}}^N = Th_{xx}G + Z$$

Submatrix4

coefficients:

$$S_{n,\{n\dots n+2\}}^N = GU \quad (A1)$$

Scenario 1: Here, the union nodes between the structures are two, X and Y. Consequently, the connection the resulting

arrangement do not increase the number of free ports that it has, as can be observed in Appendix Fig. 11. (B1). Appendix Fig. 11. (B2). and (B3) illustrates the graph and the resulting scattering matrix obtained, after solving the system of equations, following the same methodology that has been explained in the case of S0. As can be seen in equations (A2), the resulting equations are more complex than (A1), since two interface nodes (X and Y) are required.

$$\begin{aligned} \text{SM1:} & S_{m,n}^N = X = S_{m,n}^{N-1} \\ \text{SM2:} & S_{\{n-1\dots n\},n}^N = OE' + TS \\ \text{SM3:} & S_{\{n-1\dots n\},\{n-1\dots n\}}^N = OP(h_{yy}+GN) + T(Gh_{xx}+MP) + Z \\ \text{SM4:} & S_{n,\{n-1\dots n\}}^N = F'P - UG \end{aligned} \quad (A2)$$

Scenario 2: This joint does not increase the number of free ports either. Again, two nodes, X and Y, form the union but these are located one on each side of the PUC (see Appendix Fig. 11. (C1)). In this case, can appear a recirculation of light between the interface nodes and the newly added lattice as shown, by connections V and W, in the signal flow graph (observe Appendix Fig. 11. (C2)). Therefore, using the diagram, we can model the equations (A3) of each submatrix in this scenario. Formerly, we can observe in the definition of the equations (A3) a geometric sum of contributions due to the possibility of recirculation inside the mesh. In addition, in this case, it should be noted that the SM1 coefficients cannot be inherited from the n-1 order scattering matrix, as occurs in scenarios 0 and 1. Hence, they must be recalculated following the aforementioned system of equations.

Finally, by solving the system of (A3), we can provide the matrix coefficients that characterize the new structure (see Appendix Fig. 11. (C3)).

Scenario 3: This scenario is the most complex because not only the number of free ports in the mesh is not increased, but the resulting structure also has two fewer free ports. As we can observe in Appendix Fig. 11. (D1), three ports, X, Y and Z, are used for the addition. For this reason, the interconnection diagram (observe Appendix Fig. 1. (D2)) involves three interfacing nodes (X, Y and Z). Highlighting, again, the possible recirculation between the nodes X and Z, and Y and Z of the interface and the new PUC joined, modeled by the connections C, E, K and L of Appendix Fig. 11. (D2).

The procedure to obtain the coefficients of the submatrices is similar to the three previous cases. The equations have greater complexity and density, so they will not be defined here. Although, we can be expressed the equation system using the signal flow graph (see Appendix Fig. 11. (D2).) and apply the same methodology as the other scenarios. Then, the resulting scattering matrix sections of the scenario 3 is shown in Appendix Fig. 11. (D3).

$$\begin{aligned} \text{SM1: } S_{m,n}^N &= \left[\frac{-(B'E'V + DSW + B'SVWH_{yy} - B'E'MVW - DNSVW + DE'VWH_{xx})}{(MW + NV - MNVW + VW h_{xx} h_{yy} - 1)} \right] + X \end{aligned}$$

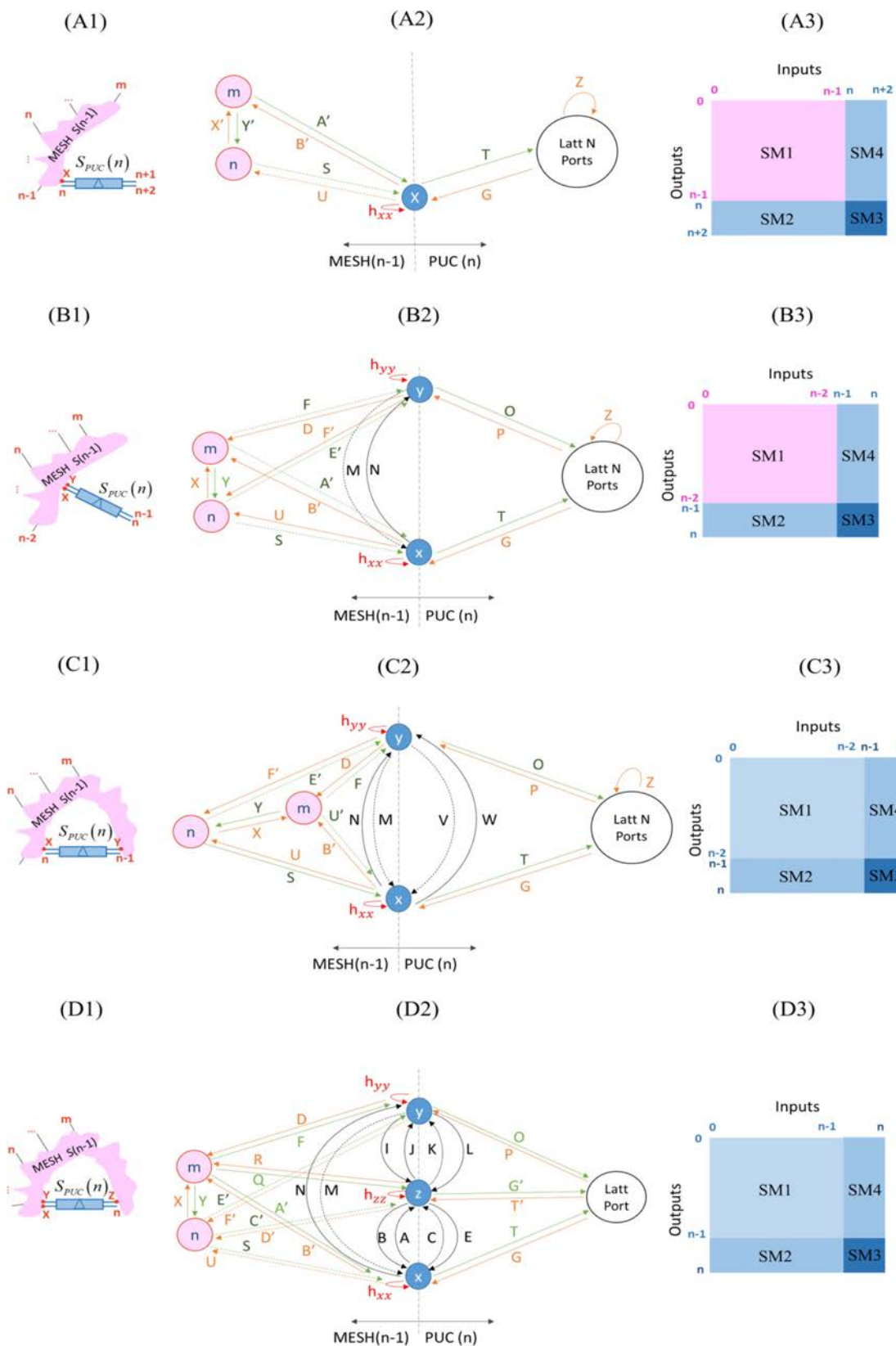


Fig. 11. (A) Scenario 0, (B) Scenario 1, (C) Scenario 2 and (D) Scenario 3 and (1) Connection scheme of the additional lattice with mesh $n-1$ of each scenario, (2) Interconnection graph diagram with the labeled contributions of each scenario (3) Resulting scattering matrix sections of each scenario.

Pseudocode A1. Programming of the inductive method presented in this work.

```

# variable initialization
Require: PUCs_per_column           #Mesh definition (Total NPUCs and columns)
Require: N_wvl_points              #Wavelength resolution
load technology_parameters
Require: IL                          #Insertion losses of each PUC
Require: passive_PUCs_phases       #Passive Phases of each PUC
Require: kappas                     #Coupling factor of each PUC
Require: SPUC = compute_Scattering_PUC #Scattering matrix of each PUC

pattern = generate_pattern_of_mesh_distribution
for column in range(pattern):
  S(n) = SFUC(1)
  S(n-1) = S(0)
  for N in range(PUCs_per_column):
    if column == 0:
      p_c0 = pattern_col_0
      for scenario in p_c0:
        if scenario == S0:
          S(n) = f(S(n-1), S(n))
        end if
        if scenario == S1:
          S(n) = f(S(n-1), S(n))
        end if
        if scenario == S2:
          S(n) = f(S(n-1), S(n))
        end if
        S(n-1) = S(n)
        S(n) = SFUC(N)
      end for
    if column == 1:
      p_c1 = pattern_col_1
      for scenario in p_c1:
        if scenario == S0:
          S(n) = f(S(n-1), S(n))
        end if
        if scenario == S1:
          S(n) = f(S(n-1), S(n))
        end if
        if scenario == S2:
          S(n) = f(S(n-1), S(n))
        end if
        S(n-1) = S(n)
        S(n) = SFUC(N)
      end for
    if column == 2:
      p_c2 = pattern_col_2
      for scenario in p_c2:
        if scenario == S0:
          S(n) = f(S(n-1), S(n))
        end if
        if scenario == S1:
          S(n) = f(S(n-1), S(n))
        end if
        if scenario == S2:
          S(n) = f(S(n-1), S(n))
        end if
        S(n-1) = S(n)
        S(n) = SFUC(N)
      end for
    if column == 3:
      p_c3 = pattern_col_3
      for scenario in p_c3:
        if scenario == S0:
          S(n) = f(S(n-1), S(n))
        end if
        if scenario == S1:
          S(n) = f(S(n-1), S(n))
        end if
        if scenario == S2:
          S(n) = f(S(n-1), S(n))
        end if
        S(n-1) = S(n)
        S(n) = SFUC(N)
      end for
    if column == 4:
      p_c4 = pattern_col_4
      for scenario in p_c4:
        if scenario == S0:
          S(n) = f(S(n-1), S(n))
        end if
        if scenario == S1:
          S(n) = f(S(n-1), S(n))
        end if
        if scenario == S2:
          S(n) = f(S(n-1), S(n))
        end if
        S(n-1) = S(n)
        S(n) = SFUC(N)
      end for
  end for
end for
S = S(n)

```

$$\begin{aligned}
\text{SM2: } & S_{\{n-1\dots n\},n}^N \\
& = \frac{(O(E'(MW-1)-SWh_{yy})+T(S(NV-1)-E'Vh_{xx}))}{(MW+NV-MNVW+VWh_{xx}h_{yy}-1)}
\end{aligned}$$

$$\text{SM3: } S_{\{n-1\dots n\},\{n-1\dots n\}}^N$$

$$= \left[\frac{-(O(Ph_{yy}+G(N-MNW+Wh_{xx}h_{yy}))+T(Gh_{xx}+P(M-MNV+Vh_{xx}h_{yy})))}{(MW+NV-MNVW+VWh_{xx}h_{yy}-1)} \right] + Z$$

$$\begin{aligned}
\text{SM4: } & S_{n,\{n-1\dots n\}}^N \\
& = \frac{(F'(P(NV-1)-GWh_{xx})+U(G(MW-1)-PVh_{yy}))}{(MW+NV-MNVW+VWh_{xx}h_{yy}-1)} \quad (\text{A3})
\end{aligned}$$

After the definition of each scenario and the presentation of the equations that must be used for each case, we expose the following methodology to do this process sequentially (see Pseudocode A1). First, we have to define the mesh shape (variable PUC_per_col), to be able to develop a pattern and identify which scenario will be the next in each step. Moreover, we already know that this will depend on the number of ports that are interconnected with each other. In addition, for each interconnection, we must identify the ports to be connected in the n-1 order structure and the new lattice. Next, we define the number of wavelength resolution points we want (variable N_wvl_points). This variable will cause variations in the computational time of the inductive method as we have seen in the results section. Therefore, we should find a trade-off between resolution and time depending on the application that we are synthesizing. Then, we load the technology parameters of each PUC as group index, Basic Unit Length (BUL), etc. because the method is greatly flexible and we can calculate the value of the scattering function in different technologies. Subsequently, we define the parameters that will model each PUC, these variables can be insertion losses (variable IL), internal coupling factors (variable kappas), passive phases (variable passive_PUCs_phases), mesh configuration, whether arbitrary or not, etc. Finally, the matrix of each S_{PUCs} is calculated independently. After initializing all variables, the inductive method is performed to compute the scattering matrix of the given mesh. Assuming that the calculation is straightforward since the new scattering matrix is evaluated as a function of the matrix from the previous step and that of the new lattice. Moreover, we know that the pattern followed by the hexagonal meshes consists of five independent columns, and then the rest of the mesh can be created as a combination of these columns.

ACKNOWLEDGMENT

We acknowledge the software team of iPrionics programmable photonics S.L., specially, David Sánchez and Zhenyun Xie for their contributions to the computational workflows and upgrades of the algorithms. We acknowledge iPrionics team for the visual tools employed within this work. In addition, we acknowledge Dr. Ana Gutierrez for his discussions and insights.

Contributions: A.L. and D.P.L. conceived graph-based simulation approach and A.L. performed the associated simulations. E.S. and D.P.L. conceived single-cell inductive approach and E.S. performed the associated simulations. A.L., E.S., D.P.L. analyzed the data and wrote the paper. D.P.L., managed, coordinated, and supervised the project.

REFERENCES

- [1] W. Bogaerts et al., “Programmable photonic circuits,” *Nature*, vol. 586, no. 7828, pp. 207–216, Oct. 2020.
- [2] D. Pérez and J. Capmany, *Programmable Integrated Photonics*. London, U.K.: Oxford Univ. Press, 2020.
- [3] L. Zhuang, C. G. H. Roeloffzen, M. Hoekman, K.-J. Boller, and A. J. Lowery, “Programmable photonic signal processor chip for radiofrequency applications,” *Optica*, vol. 2, no. 10, pp. 854–899, Jun. 2015.
- [4] D. Pérez, A. M. Gutiérrez, E. Sánchez, P. DasMahapatra, and J. Capmany, “Integrated photonic tunable basic units using dual-drive directional couplers,” *Opt. Exp.*, vol. 27, no. 26, pp. 38071–38086, 2019.
- [5] 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, Apr. 2018.
- [6] D. Pérez, I. Gasulla, and J. Capmany, “Reconfigurable lattice mesh designs for programmable photonic processors,” *Opt. Exp.*, vol. 24, no. 11, pp. 12093–12106, May 2016.
- [7] D. Pérez et al., “Multipurpose silicon photonics signal processor core,” *Nature Commun.*, vol. 8, no. 1, Sep. 2019, Art. no. 636.
- [8] D. Pérez et al., “Silicon photonics rectangular universal interferometer,” *Laser Photon. Rev.*, vol. 11, no. 6, Nov. 2017, Art. no. 1700219.
- [9] D. Pérez, I. Gasulla, and J. Capmany, “Field-programmable photonic arrays,” *Opt. Exp.*, vol. 26, no. 21, pp. 27265–27278, Oct. 2018.
- [10] D. Marpaung, J. Yao, and J. Capmany, “Integrated microwave photonics,” *Nature Photon.*, vol. 13, pp. 80–90, Feb. 2019.
- [11] E. Knill, R. Laflamme, and G. J. Milburn, “A scheme for efficient quantum computation with linear optics,” *Nature*, vol. 409, pp. 46–52, Jan. 2001.
- [12] P. Kok, W. J. Munro, K. Nemoto, T. C. Ralph, J. P. Dowling, and G. J. Milburn, “Linear optical quantum computer with photonic qubits,” *Rev. Modern Phys.*, vol. 79, pp. 135–174, Jan. 2007.
- [13] J. L. O’Brien, A. Furusawa, and J. Vučković, “Photonic quantum technologies,” *Nature Photon.*, vol. 3, pp. 687–695, Dec. 2009.
- [14] M. G. Thomson, A. Politi, J. C. Matthews, and J. L. O’Brien, “Integrated waveguide circuits for optical quantum computing,” *IET Circuits, Devices Syst.*, vol. 5, no. 2, pp. 94–102, Mar. 2011.
- [15] A. Politi, J. Matthews, M. G. Thomson, and J. L. O’Brien, “Integrated quantum photonics,” *IEEE J. Sel. Topics Quantum Electron.*, vol. 15, no. 6, pp. 1673–1684, Dec. 2009.
- [16] A. Politi, M. J. Cryan, J. G. Rarity, S. Yu, and J. L. O’Brien, “Silicon-on-silicon waveguide quantum circuits,” *Science*, vol. 320, no. 5876, pp. 646–649, May 2008.
- [17] K. Kieling, J. L. O’Brien, and J. Eisert, “On photonic controlled phase gates,” *New J. Phys.*, vol. 12, Art. no. 013003, Jan. 2010.
- [18] J. B. Spring et al., “Boson sampling on a photonic chip,” *Sci.*, vol. 339, no. 6121, pp. 798–801, Dec. 2012.
- [19] M. A. Broome et al., “Photonic boson sampling in a tunable circuit,” *Sci.*, vol. 339, no. 6121, pp. 794–798, Dec. 2012.
- [20] A. Crespi et al., “Integrated multimode interferometers with arbitrary designs for photonic boson sampling,” *Nature Photon.*, vol. 7, pp. 545–549, May 2013.
- [21] Y. Shen et al., “Deep learning with coherent nanophotonic circuits,” *Nature Photon.*, vol. 11, pp. 441–446, Jun. 2017.
- [22] H. Peng, M. A. Nahmias, T. F. de Lima, A. N. Tait, and B. J. Shastri, “Neuromorphic photonic integrated circuits,” *IEEE J. Sel. Topics Quantum Electron.*, vol. 24, no. 6, pp. 1–15, Dec. 2018.
- [23] D. Pérez-López, “Programmable integrated silicon photonics waveguide meshes: Optimized designs and control algorithms,” *IEEE J. Sel. Topics Quantum Electron.*, vol. 26, no. 2, Mar./Apr. 2020, Art. no. 8301312.
- [24] J. C. Mikkelsen, W. D. Sacher, and J. K. S. Poon, “Dimensional variation tolerant silicon-on-insulator directional couplers,” *Opt. Exp.*, vol. 22, pp. 3145–3150, 2014.
- [25] M. Hochberg and L. Chrostowski, *Silicon Photonics Design: From Devices to Systems*. Cambridge, NY, USA: Cambridge Univ., 2015.
- [26] D. Pérez and J. Capmany, “Scalable analysis for arbitrary photonic integrated waveguide meshes,” *Optica*, vol. 6, no. 1, pp. 19–27, 2019.
- [27] A. López, D. Pérez, P. DasMahapatra, and J. Capmany, “Auto-routing algorithm for field-programmable photonic gate arrays,” *Opt. Exp.*, vol. 28, no. 1, pp. 737–752, Jun. 2020.
- [28] D. S. Gunderson, *Handbook of Mathematical Induction: Theory and Applications*. New York, NY, USA: CRC Press, Taylor and Francis, 2010.
- [29] R. J. Trudeau, *Introduction to Graph Theory*. North Chelmsford, MA, USA: Courier Corporation, 1993.
- [30] X. Chen, P. Stroobant, M. Pickavet, and W. Bogaerts, “Graph representations for programmable photonic circuits,” *J. Lightw. Technol.*, vol. 38, no. 15, pp. 4009–4018, Apr. 2020.
- [31] B. J. Shastri et al., “Photonics for artificial intelligence and neuromorphic computing,” *Nature Photon.*, vol. 15, pp. 102–114, Jan. 2021.
- [32] D. P. Pérez, A. L. Hernández, P. DasMahapatra, and J. Capmany, “Field-Programmable photonic array for multipurpose microwave photonic applications,” in *Proc. IEEE Microw. Photon. Conf.*, Ottawa, ON, Canada, 2019, pp. 1–4.
- [33] D. Pérez-López, A. López, P. DasMahapatra, and J. Capmany, “Multipurpose self-configuration of programmable photonic circuits,” *Nature Commun.*, vol. 11, no. 1, pp. 1–11, 2020, doi: [10.1038/s41467-020-19608-w](https://doi.org/10.1038/s41467-020-19608-w).

Erica Sánchez (Graduate Student Member) received the B.Sc. degree in telecommunications technology engineering from the Universidad Miguel Hernández de Elche, Elche, Spain, and the M.Sc. degree in telecommunication technologies, systems, and networks from the Universitat Politècnica de València, València, Spain, where she is currently working toward the Ph.D. degree in integrated photonics, focusing on the design of a universal MWP programmable signal processor with Photonics Research Labs.

Aitor López (Graduate Student Member) received the B.Sc. and M.Sc. degrees in telecommunication engineering from Universidad Politécnica de Madrid, Madrid, Spain. He is currently working toward the Ph.D. degree in programmable photonics with Universitat Politècnica de València, València, Spain, and the second B.Sc. degree in physics with the Spanish National University of Distance Education, Madrid, Spain. His research interests include optical computing and machine learning.

Daniel Pérez-López (Member, IEEE) received the M.Sc. degree and the Ph.D. degree from Photonics Research Labs, Universitat Politècnica de València, València, Spain. He has spent his academic life on the creation, development of a programmable photonics: such giving to large-scale photonic integrated circuits flexibility through run-time software programming. His career started with his Ph.D. and a two-year postdoc research with the Universitat Politècnica de València’s Photonics Research Labs, where he investigated novel photonic integrated circuit (PIC) designs and architectures, which can be reconfigured to perform multipurpose functionalities. These works led to the 2017 Graduate Student Fellowship by the IEEE Photonics Society and 2018 Extraordinary Ph.D. Thesis Award from UPVLC. During 2019–2020, he was with Xanadu Quantum Computing, Toronto, ON, Canada, and founded iPrionics Programmable Photonics S.L, Valencia, Spain. In 2019 and 2021, he was the recipient of the prestigious Juan de la Cierva Fellowship grants that allowed him to initiate his research on high-dense large-scale programmable photonic integrated circuits and its applications to signal processing and computing.