Application-specific Scaling in Programmable Photonic Circuits

(Student Paper)

Iman Zand [1,2] Xiangfeng Chen [1,2] Wim Bogaerts [1,2]
[1]*Photonics Research Group, Department of Information Technology, Ghent University-IMEC, 9052 Ghent, Belgium*
[2]*Center of Nano and Biophotonics, Ghent University, 9052 Ghent, Belgium*
e-mail: wim.bogaerts@ugent.be

ABSTRACT

We performed a scaling analysis of programmable hexagonal two-dimensional (2D) waveguide meshes with different shapes. We show that the shape of the meshes can impose restrictions such as high path losses or large foot-prints for certain configurations and applications. We use the graph-based strategy to guide our application-specific scaling analysis. The design practices are discussed for implementing the programmable mesh for beam splitters and $N \times N$ switches. This will provide a guideline of the shape and size for the future programmable photonic circuits.

Keywords: programmable photonic circuits, scaling analysis, hexagonal waveguide meshes, beam splitters, switches, graph-based strategy

1 INTRODUCTION

Programmable photonic integrated circuits (PIC) [1], at their core, consist of a linear optical processor, which allows arbitrary linear transformations between optical ports. This can be implemented as a circuit mesh composed of tunable couplers and phase shifters, which can be 'programmed' to perform routing or filter functions. One of the important choices in the design of such programmable meshes is their size and shape; in fact, scaling of meshes can cause adverse parasitic effects, high loss, or large footprint of the photonic circuit [2]. Here, we explore how shape of such meshes influences their scaling and the programming of desired functions.

2 METHODOLOGY

We start from a mesh with hexagonal unit cells for our 2D meshes as proposed in [3] and organize the unit cells in a radial, square, or rectangular fashion, as shown in Fig. 1a. We add "whiskers" with an additional tunable coupler on all boundaries. In these designs, we assume that each cell has three phase shifters which are connected to the couplers through waveguides. The expansion parameters (i.e. how the mesh grows) for a radial, square, and rectangular mesh are r, w, and (a, b), respectively.

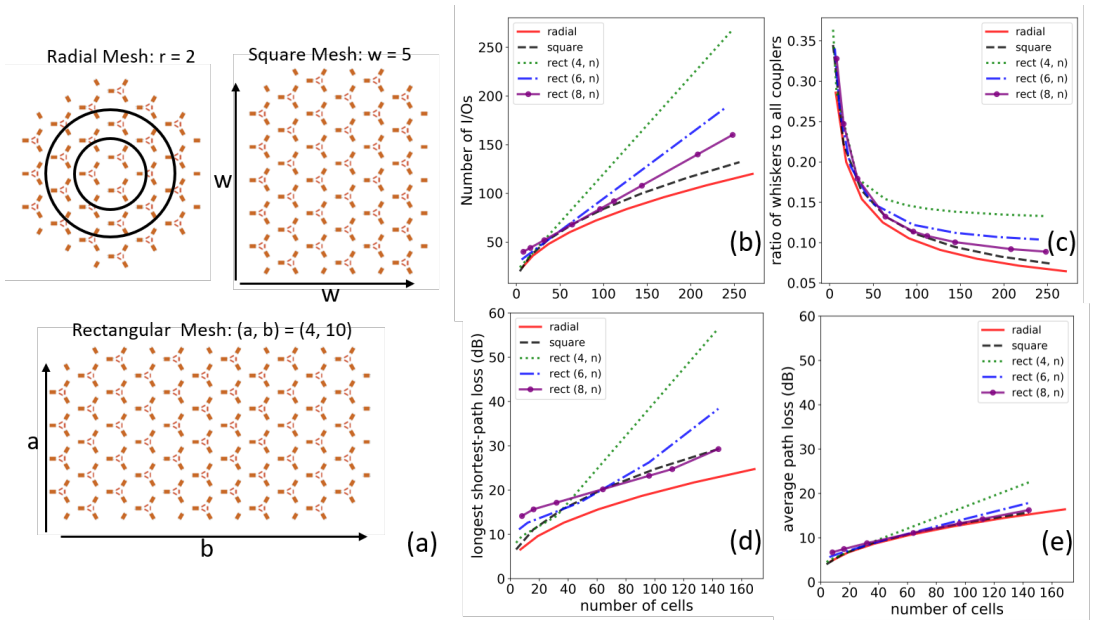

Figure 1. a) Schematic of the waveguide meshes using hexagonal topology and with radial, square, and rectangular shape. b) Number of ports, c) Ratio of the whiskers to all couplers (edge to area ratio), d) Loss of the longest shortest-path, e) Average path loss . Here, we have considered three types of the rectangular meshes with fixed number of rows (b=4, 6, 8).

3 MESH PROPERTIES

For the scaling analysis, we consider four mesh properties: (1) the number of I/O's, (2) the ratio of edge-to-area indicated by the number of whiskers divided by the number of couplers, (3) the loss of the longest shortest-path (i.e. the longest straight path between two ports on the edge, which is an important metric for routing performance), and (4) average loss of the mesh (Fig. 1(b-d). The loss calculations are based on recent results on silicon photonic MEMS devices from the MORPHIC project, estimated as 0.75 dB for the paths segment loss [4], [5]. smallest average distance of the ports from each other. As a result, it shows the lowest loss for the longest shortest path. Moreover, the radial mesh has a lower average path loss indicating a lower loss for multi routing. While the radial mesh scales better in terms of losses, it may impose restrictions for some applications, which will be discussed later. Meshes with a higher edge-to-area ratio have a worse scaling behavior.

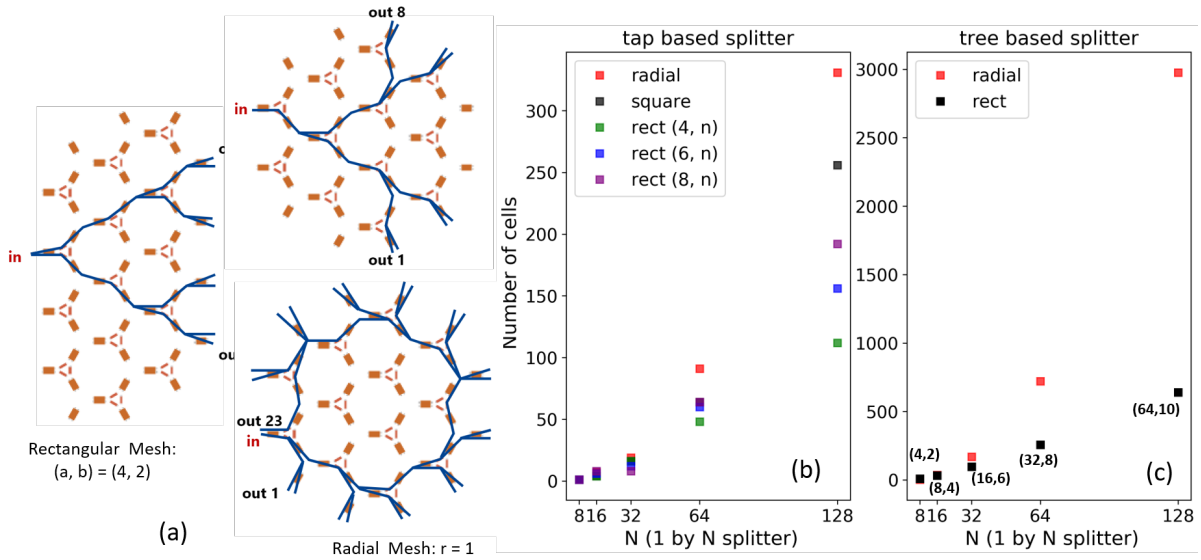

Figure 2. a) Schematics of a 1×8 tree-based beam splitter in rectangular and radial meshes; and, also, a 1×23 tap-based beam splitter in radial mesh ($r = 2$). The number of cells of different meshes required to b) configure tap-based $1 \times N$ beam splitters in radial, square, and three different rectangular meshes. c) tree-based $1 \times N$ beam splitters in radial and rectangular meshes with $(a, b) = (4, 2), (8, 4), (16, 6), (32, 8), (64, 10)$.

4 SCALING FOR DIFFERENT APPLICATIONS

4.1 Beam Splitters

One of the important functions in a programmable mesh is a splitter network for multicasting or the distribution network for an optical beam former. Here, *tap-based* and *tree-based* beam splitters have been studied in the meshes shown in Fig. 1a. Figure 2a shows a few examples of splitter implementations in radial and rectangular meshes. Tap-based splitters do not have balanced paths but can lead to a more compact mesh; a radial mesh with $r = 1$ can support a 1×23 tap-based beam splitter, while in a fully balanced tree it can only be programmed for a 1×8 splitter. For the different mesh shapes we calculated the required size of the mesh to implement a tap or tree-based splitter with different numbers of outputs. Comparing the results of tap-based splitters for different mesh shapes (Fig. 2b) shows that a steady increase of the mesh foot print for the tap-based splitter, and this effect is even more adverse for the tree-based splitter (Fig. 2c). Results show that using rectangular meshes provides better splitting performance compared to the radial meshes.

4.2 $N \times N$ Switching Circuits

The programmable mesh with a rectangular shape scales up in both directions (a, b) to enable $N \times N$ switching. In Fig. 3, we provide an example graph representation [6] for such hexagonal mesh with $(a, b) = (8, 7)$. The degrees of freedom for the choice of input-output ports of a switch circuit is large. An $N \times N$ means that any input can be connected to any output on a one-to-one basis. It quickly becomes computationally expensive to route all these possible pair combinations. Thus, we assume that if a mesh can allocate N input-output pairs in both full-parallel and full-diagonal assignment, it can probably allocate the full $N \times N$ switch circuit. For example, Fig. 3a-b shows simultaneous multi routing for 8 input-output pairs in parallel and in

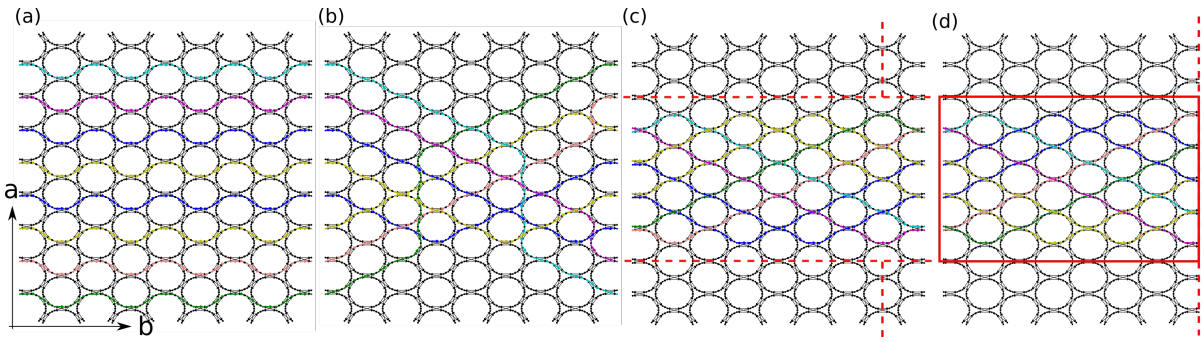

Figure 3. Graph-based routing in a rectangular-shaped meshes with 7×8 hexagonal cells. (a) Multi-routing finding the shortest paths routing for 8 input-output pairs in parallel and (b) in diagonal assignment. (b) shows a sparse use of 16 input-output ports (one per whisker) while (c) shows a dense use (two per whisker). The boundaries in dashed red lines shows the reduction in required mesh size, which is further improved in (d), representing the minimum mesh size to implement 8×8 switching

diagonal assignment, with a sparse assignment of inputs and outputs (one per whisker). We observe that every path will cross 7 other paths in Fig. 3b. so, as long as we reserve sufficient crossing couplers, we can reduce the cell number a in vertical direction, as shown in Fig. 3c. For a given number of inputs/outputs, we use heuristic-based congestion negotiation algorithms to sweep the values of both a and b. These simulations verify that the minimal size mesh for an $N \times N$ switch from left to right has boundary conditions as the mesh scales up: When b is an odd number as in Fig. 3c, the East side is symmetric with the West side, whereas when b is an even number as in Fig. 3d, the East side has more output ports than the West side. We choose the symmetrical configuration as the minimal requirement: solutions are found when $b \geq N - 1$ (b odd) and $a \geq N/2$. When b is even, the conditions are $b \geq N - 2$ and $a \geq N/2$. If the mesh is smaller than these conditions, generic left-to-right switching is no longer guaranteed.

5 CONCLUSION

We performed a scaling analysis for 2D waveguide meshes with different shapes (based on hexagonal unit cells). Simulation results show that the shape of the meshes can considerably affect the loss scaling and restrict the implementation of application-specific functions such as beam splitters and $N \times N$ switches. This paper provides a design guideline for the future programmable photonic circuits.

ACKNOWLEDGEMENTS

This work was funded by the EU through grant 780283 (MORPHIC) and 725555 (PhotonicSWARM)

REFERENCES

[1] D. Pérez, I. Gasulla, and J. Capmany, "Programmable multifunctional integrated nanophotonics - Review Article," *De Gruyter - Nanophotonics 2018*, vol. 7, no. 8, pp. 1351–1371, 2018.

[2] I. Zand and W. Bogaerts, "Effects of Coupling and Phase Imperfections in Programmable Photonic Hexagonal Waveguide Meshes," *Photonics Research*, vol. 8, no. 2, 2019.

[3] D. Pérez, I. Gasulla, J. Capmany, and R. A. Soref, "Reconfigurable lattice mesh designs for programmable photonic processors," *Optics Express*, vol. 24, no. 11, p. 12093, 2016.

[4] W. Bogaerts, H. Sattari, P. Edinger, Y. A. Takabayashi, I. Zand, X. Wang, A. Ribeiro, M. Jezzini, C. Errando-Herranz, G. Talli, K. Saurav, M. Garcia Porcel, P. Verheyen, B. Abasahl, F. Niklaus, N. Quack, K. B. Gylfason, P. O'Brien, and U. Khan, "MORPHIC : Programmable photonic circuits enabled by silicon photonic MEMS," *Proc. SPIE*, vol. 11285, pp. 11 285–1, 2020.

[5] C. Errando-Herranz, A. Y. Takabayashi, P. Edinger, H. Sattari, K. B. Gylfason, and N. Quack, "MEMS for Photonic Integrated Circuits," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 26, no. 2, pp. 1–1, 2019.

[6] X. Chen and W. Bogaerts, "ME2.2 - A Graph-based Design and Programming Strategy for Reconfigurable Photonic Circuits," *2019 IEEE Photonics Society Summer Topical Meeting Series (SUM)*, pp. 1–2, 2019.