Discrete global optimization algorithms for the inverse design of silicon photonics devices

Olivier Teytaud, Pauline Bennet, Antoine Moreau

Discrete global optimization algorithms for the inverse design of silicon photonics devices

Olivier Teytaud[a], Pauline Bennet[b], Antoine Moreau[b]

[a] *Facebook AI Research Paris, France,*
[b] *Universite Clermont Auvergne, Clermont Auvergne INP, CNRS, Institut Pascal, F-63000 Clermont-Ferrand, France*

Abstract

The photonics community counts on inverse design approaches to provide new designs of miniaturized photonic components. Although the structures are composed of two materials only, the dominant method used to tackle such a problem is to make the problem continuous through a relaxation of the binary constraint and to use a gradient-based approach. Global optimization algorithms working on continuous problems actually often fail to produce satisfying solutions in such a context characterized by a large number of parameters – especially when space is discretized into a large number of pixels or voxels. However, we show here for three different photonics problems that global *discrete* optimization algorithms, which are adapted to this kind of problems, can provide efficient solutions which are relatively regular, physically understandable and close to being buildable. Such algorithms constitute overlooked but valuable tools for inverse design problems. Our preferred method is using a starting point provided by a gradient based algorithm, and run an optimization using Lengler's algorithm equipped with a simple smoothing operator to make the algorithm aware of the physical nature of the problem.

Keywords: Inverse design, Discrete optimization

PACS: 0000, 1111

2000 MSC: 0000, 1111

1. Introduction

Photonic structures are increasingly integrated in electronic devices[1]. How far this may lead us depends on whether the photonics community will be able to provide efficient miniaturized structures to meet the various needs of electrical engineers. The main obstacle is that there are no design rules based on a physical understanding of complex 2D or 3D structures.

All the physical analysis and design tools developed for multilayers[2] are useless in such a context, so that we do not even know which devices can be imagined and what will be their physical characteristics. The community has thus turned to numerical optimization to provide original designs: whenever a measure of the discrepancy between the performances of a given design and the ideal or desired performance can be defined, then optimization algorithms trying to minimize such a cost function naturally provide new structures, a process called inverse design.

However, given the difficulty of even the simplest optimization problems, there is generally no realistic way to know whether a solution provided by an algorithm is the true optimal minimum or a local minimum. This means that a photonic design provided by optimization is in general not the best possible design, and it is even difficult to estimate how far it is from the optimum[3]. The doubt easily put on any solution makes it often difficult to convince that inverse design is even the right path to follow. But all recent results suggest that even extremely miniaturized devices which are not fully satisfying can present interesting performances[4, 5].

Given the complexity of any optimization problem in photonics, for a long time parametric optimization has been preferred in the community. The idea is globally to describe a photonic structure as a collection of objects whose optical or geometrical characteristics can be described using a few continuous variables (e.g. the radius of a rod, the length and width of a block, the position of the object, its refractive index). This allows to describe the whole design using a relatively small set of variables, making the use of standard global optimization algorithms for continuous variables (eg PSO, DE) possible[6]. Recent results have shown that above 50 variables typically, problems in photonics tend to become too difficult for most of global optimization algorithms[7]. The main drawback of parametric optimization is probably the inherent biases associated with the way the structure is described.

More recently, an approach originally used to solve mechanical problems and called topology optimization[8] has been used to tackle problems in photonics[9, 10, 4]. Contrarily to parametric optimization, the general idea is to constraint as little as possible the geometry of any solution. A possibility is to divide the structure into numerous pixels (or voxels). In mechanics as in photonics, each pixel can be filled with a material or with air making the whole problem discrete. Such a setting is not adapted to the use of gradient-based methods because it is then not described by continuous variables. However, for each pixel, a continuous variable can usually be introduced – allowing to describe intermediate optical or mechanical properties, making the problem continuous and even differentiable. Typically, for each pixel in photonics, the value 0 is used for air and a value of 1 is used for the high index material. The gradient of the cost function is then cheap to compute[11, 9, 5]. Given the large number of pixels involved (several hundreds), which makes the problem difficult for global optimization algorithms, gradient based methods appear as an obvious choice.

In mechanics, such an approach provides solutions that are actually buildable[12] quite easily. The results are easily reproducible, indicating the optimization problem (the so-called "landscape") is relatively simple, and finally the devices produced are physically understandable[13]. This is just not as simple in photonics. Even the most simple photonic structures (e.g a single dielectric layer) are resonant, which implies the existence of numerous local minima. Solutions provided are generally not buildable (with many intermediate values for the permittivity) so that the discretization step, where the continuous variable describing each pixel needs to be converted into 0 or 1, implies a large modification of the structure and of its optical properties. This last step is always detrimental to the performances.

Overall, topology optimization is less successful in photonics, producing efficient but complex structures with small features hindering commercialization for now[5]. This suggests that the gradient based approach is perhaps not sufficient in that case. Numerical optimization being an incredibly dynamic field with enormous potential applications, a large number of well tested algorithms suited to the present, discrete case are available. Optimizations (also called experiments) can easily be run using freely available and easy to deploy libraries[14, 15, 16].

Here we show that discrete optimization algorithms can actually provide improved designs, especially when they come after a first gradient-based optimization. We show this is the case on three test cases provided in the freely available SPINS-B package[17, 18]: a waveguide bend, a demultiplexer and a grating coupler. We rely on the Nevergrad platform[19, 16] to run the experiments using several different discrete algorithms. Importantly, even though we have not especially imposed any buildability constraint, the structures finally produced are rather buildable and relatively regular, so that they can probably be understood physically – suggesting high quality solutions have been actually produced. We emphasize that the present work has been made possible because the authors of previous works have taken care to freely distribute their codes and methods, so that ours have been released too and are available without any restriction[20].

2. Methods

2.1. Freely available packages

We rely on the three test cases released in the SPINS-B freely available python package[18]. The core of the package is a Finite Difference Frequency Domain (FDFD) solver allowing to evaluate the optical response of a 2D structure divided into pixels and to compute the cost function corresponding to three different cases (i) a square domain meant to force an incoming guided mode to make 90 degrees turn (ii) a square domain able to separate two incoming wavelength (under the form of a guided mode), sending them in two different waveguides and (iii) a grating coupler, designed to couple an incoming plane wave to a guided mode, a common problem in photonics. The cost functions are thoroughly described[17], easy to interface with any python code and can be considered as black box functions on which different algorithms can be compared.

The package is well documented, and constitutes a lighter version of the SPINS code[17], meant to help design full 3D photonic components and which is not freely available. By solving an adjoint problem, the gradient of any cost function can be computed for a low computational cost, so that a Broyden–Fletcher–Goldfarb–Shanno (BFGS[21]) method can then be implemented to generate a design by gradient based computation. More precisely, the algorithm is a limited-memory and bounded version of the method (L-BFGS-B) particularly adapted to handle a large number of parameters. It is simply called BFGS in the following. For the discretization step following BFGS which transforms a structure with continuously varying permittivity into a design with only two materials, we rely on the technique provided in SPINS-B based on sigmoids, which has been tailored to provide relatively satisfying discretized structures. Globally, we use the BFGS method exactly as it has been implemented in the SPINS-B package. Eventually, we catch the design before it undergoes any discretization to evaluate the performances of the continuous structure. We underline that there are other approaches[9] in which rely on the penalization of intermediate values of the permittivity. Such methods were alas not implemented in the SPINS-B package.

We use the Nevergrad platform to provide us with state of the art algorithms, whether they are meant for continuous or discrete variables. Nevergrad is a freely available benchmarking platform for optimization algorithms that are not based on gradient, as its name suggests[19].

2.2. *Benchmarking methodology*

In order to be able to compare different methods, each run is attributed only a limited number of evaluations of the cost function (assuming this is the most costly part of the computation). This is called the budget. This is the classical way to compare algorithms and methods fairly, even though what really matters for applications is the best design overall, whatever the method or the budget used. All the results we have obtained are averaged over at least 5 different runs with different starting points (5 for the waveguide bend, at least 10 for the others).

The BFGS method being intrinsically a local optimization method able to converge quickly, it then stagnates at a local minimum. The results can be vastly improved by restarts: once it has converged to a solution, BFGS is restarted with a randomly chosen starting point. We stop the restarts when the budget is reached and we keep only the best solution provided. We give BFGS a slight advantage here: we count each step as only one evaluation of the cost function whereas each step is roughly 2.5 times more costly.

Whenever we need to start anew any algorithm, we begin with a completely random design where each pixel is randomly and independently chosen in the 0.4-0.6 refractive index range. On some occasions (for the difficult multiplexer problem), we randomly draw the initial point identically over blocks of $h \times h$ pixels instead of 1×1. The marginal distribution for each pixel is the same as in the original SPINS-B, but instead of independence we use identical values *per block*.

2.3. *Discrete optimization algorithms*

We have used global optimization algorithms provided in Nevergrad that work on the discrete problem (problems with only two materials). We use many methods from the black-box optimization world: simple evolutionary methods, Fast Genetic Algorithms [22], the portfolio variant[23], various flavors of differential evolution [24] including Holland crossovers[25], Lengler's algorithm[26], particle swarm optimization[27], compact genetic algorithms [28], covariance matrix adaptation [29], memetic algorithms based on first Lengler's algorithm followed by the Discrete $(1+1)$, some wizards[30] and others. In some cases, we have tried to start with a random design and to apply the algorithms directly, but as will be clear below, our best runs are when we use a two stages optimization: we start with a BFGS and proceed to a discretization, to provide a sort of initial guess, then we run Nevergrad algorithms.

We insist that our discrete algorithms are completely unaware of the physical nature of the problem: there is nothing which tells any algorithm that two pixels are actually adjacent physically. The simplest way to make an algorithm more aware of such a situation is to introduce what we call a smoothing of the structure. The process is simple: every 55 newly generated structure, by one mean or another, we try the simplest possible filter, and keep the result if such a filtering improves the solution. For Lengler's algorithm, this means less than 2% of the solutions generated will be perturbed by the smoothing. This is enough to make the algorithm aware of the bidimensional nature of the pixels, without disturbing too much the way it functions.

For the filtering, we consider the values (0 or 1 for air and silicon respectively) of a pixel and its 8 neighbours. The pixel's value is set to the value which has the majority among these 9 pixels (said otherwise, we take the median of the different values), for one fourth of the pixels, randomly chosen. This is summarized in annex in Alg. 1. Such a smoothing is the simplest possible for a pixelized structure, but others could be thought of. This is fundamentally different from a constraint and does not hinder *a priori* small details to appear during the optimization.

While the two first cases in the SPINS-B code are 2D structures, the third case concerns a grating[31, 32]. The approach in SPINS-B is then completely different: after a run of BFGS and a discretization, a continuous parametric optimization is performed. We compare in the following this approach to ours, which stays essentially the same (a two stage optimization with BFGS and a discretization providing the starting point).

3. Results

3.1. Waveguide bend

The waveguide bend case provided by SPINS-B has served as a basis to compare methods and algorithms, and to determine which one is the most adapted to photonics problems. The main reason is that all the most successful designs have a lot of resemblance, meaning that it is possible to tell even from the look of the solution whether it is satisfactory or not. In order to get more interesting results, we increase the resolution of the problem to 42×42 instead of

Budget 300

Algorithm	Score
AdapDisc$(1+1)$	0.804
DiscBSO$(1+1)$	0.579
DiscLengler$(1+1)$	0.752
Disc$(1+1)$	0.636
FastGADisc$(1+1)$	0.636
PortfolioDisc$(1+1)$	0.621
BFGS continuous	1.004
BFGS discretized	0.860

Budget 1200

Algorithm	Real Score
AdaptDisc$(1+1)$	0.988
DiscBSO$(1+1)$	0.711
DiscLengler$(1+1)$	0.899
Disc$(1+1)$	0.735
FastGADisc$(1+1)$	0.901
NGOpt	0.899
PSO	0.568
PortfolioDisc$(1+1)$	0.679
Disc3	0.978
BFGS continuous	1.004
BFGS discretized	0.882

Budget 9600

Algorithm	Score
AdaptDisc$(1+1)$	0.997
DiscBSO$(1+1)$	0.519
DiscLengler$(1+1)$	0.999
Disc$(1+1)$	0.802
FastGADisc$(1+1)$	0.917
NGOpt	0.999
PortfolioDisc$(1+1)$	0.883
discrete3	0.997
BFGS continuous	1.004
BFGS discretized	0.860

Budget 70000

Algorithm	Score
AdaptiveDiscreteOnePlusOne	1.000
DiscreteLenglerOnePlusOne	1.000
DiscreteOnePlusOne	0.970
NGOpt	1.000
RLSOnePlusOne	0.988
discrete3	0.946
discretememetic	1.000
memeticde	0.984
BFGS continuous	1.004
BFGS discretized	0.860

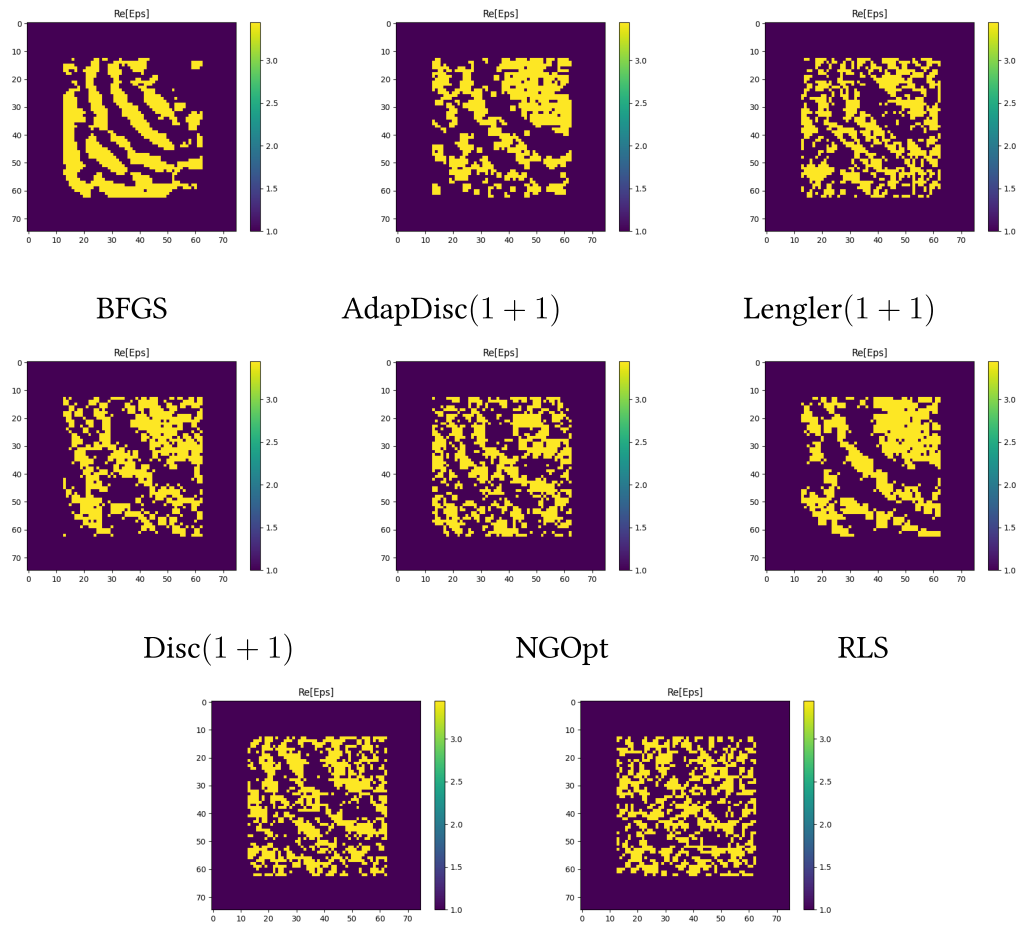
Table 1: Final scores for various methods on the 90 degrees bending waveguide problem, for different budgets. The higher the score, the better, one being theoretically the maximum. Note that BFGS needs the gradient and is therefore roughly 2.5 times slower per iteration than other methods. To be fair, for BFGS, we indicate the score computed before the discretization (it is essentially perfect) and after. We underline that BFGS is restarted whenever a local minimum is reached to be able to explore more.

a 16×16 resolution.

We first compare all the methods starting with random points. For low budgets (around 300), BFGS provides the best results (presented Table 1), but as the budget increases typically above 10^4, most of the discrete algorithms we have considered present better scores. A score of 1, as computed using SPINS-B is a perfect score which can theoretically not be exceeded. However, small numerical errors allow to reach values that are slightly beyond. We insist for instance that the score presented by BFGS before the discretization is already slightly beyond 1, but the discretization is detrimental to this score, so that the discretized structure has a lower score around 0.86.

It seems however that considering the score reached by each optimization is not enough.

We show (Fig. 1) the best designs obtained for budget 30000. While all of the designs seem to exhibit some sort of periodicity, indicating there is probably physically only one type of solution to such a problem, they are not all sufficiently smooth. This is not reflected in the corresponding score, since most of these structures present a score close to 1. When it comes to generating smooth structures, it seems BFGS performs the best.



BFGS AdapDisc$(1+1)$ Lengler$(1+1)$

Disc$(1+1)$ NGOpt RLS

Memetic Lengler/Disc$(1+1)$ PortfolioDisc$(1+1)$

Figure 1: Budget 30000. Designs obtained without BFGS are not smooth: BFGS designs look better.

The cost function does not reflect the smoothness of the design, a characteristics which is however important. In order to know whether smoother structures can reach high scores, we combine eventually BFGS starting from a random point and then discretized and a second stage for which we use a discrete optimization algorithm with the smoothing operator described above.

Visually, all the solution looks better. This is especially the case for BFGS when using the smoothing operator and the overall best is when we use BFGS first, and then a run of

Algorithm	Score
PortfolioDiscrete$(1+1)$	0.759
SMOOTHPortfolioDiscrete$(1+1)$	0.783
SMOOTHDiscrete$(1+1)$	0.799
SMOOTHAdaptiveDiscrete$(1+1)$	0.882
Discrete$(1+1)$	0.899
BFGS (with discretization)	0.948
AdaptiveDiscrete$(1+1)$	0.965
SMOOTHDiscreteLengler$(1+1)$	0.992
DiscreteLengler$(1+1)$	0.993
SMOOTHmemetic	0.996
discretememetic	0.997
BFGS/SMOOTHDiscreteLengler$(1+1)$	0.999
BFGS (without discretization)	1.004

Table 2: Numerical results with budget 3000, averaged over 5 runs each. Besides being the smoothest visually (Fig. 2), the combination BFGS+Lengler+Smoothing operator performs best numerically and equivalently to the best other methods with 10 times more budget (Fig 1).

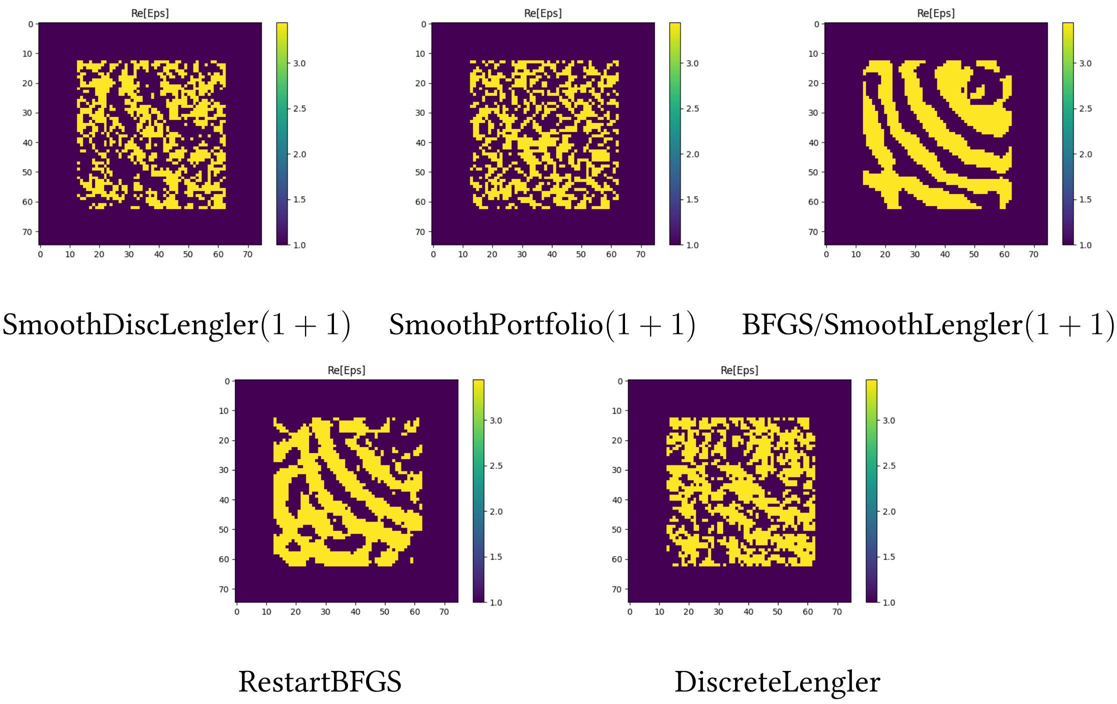


Figure 2: Visualization of waveguide bend designs with budget 3000, including smoothing operators. The light arrives from the middle at the top, and leaves at the middle on the right hand side.

the Lengler[26] algorithm with a smoothing operator (Alg. 2). In that case the result is both particularly efficient and particularly smooth. This makes this solution convincing, so that we will apply this technique to the two other test cases proposed in SPINS-B.

3.2. Demultiplexer

The same method as described above is applied to a wavelength demultiplexer of size $2\mu m \times 2\mu m$ that splits 1550 nm and 1300 nm wavelengths. The objective function is the sum for both wavelength of the squared lost share of transmitted power ($(1 - power)^2$) for the relevant output path.

This case is fundamentally different from the previous one in that the lowest cost function physically reachable (i.e. a value of 0) is never reached by any algorithm, whereas the maximum score of 1 is essentially reached by all algorithms in the waveguide bend case. It could thus be considered as more difficult. We have tried several resolutions (14×14, 42×42 and 202×202).

Results for the Demultiplexer use case are presented in Tab. 3, comparing methods depending on the budget and the resolution case of the structure. The progress represents the diminution of the cost function value when Lengler's algorithm and smoothing operator are combined with BFGS method in comparison to the use of the BFGS method alone. A positive progress means the combining method outperform the BFGS method. Exactly as what occurs in the waveguide bend, BFGS performs well for low budgets. Lengler's algorithm and smoothing operator however leverage larger budget to explore the surroundings in the parameter space, usually finding more efficient solutions and outperforming BFGS. While we present only the results for the highest resolution, the same conclusions can be drawn for lower resolutions – the progress brought by our approach is just slightly lower.

Increasing the resolution of the structure, and so the number of pixels, allows to have more degrees of freedom but as SPINS-B originally uses random uniform pixels with complete independence to generate starting points (each pixel is randomly drawn independently of other pixels), those starting points lack of homogeneity at the wavelength scale. The structure will be seen as an intermediate material by the wavelength with the same characteristics whatever the starting structure really is. We test (gray rows) randomly drawing pixels by groups of $h \times h$, where h is one tenth of the size of the design. Average results and, even more, best results, become significantly better both for BFGS and for our methods. Not only the average value is better, but more importantly the variance is greater by 50% , leading to more progress on best value found (over 50 runs) than with the previous random initialization procedure. This illustrates the dependency of the results on the starting points chosen, something we have not explored further here.

The different structures obtained during the best run we have made are shown in Fig. 3 for the highest resolution and the largest budget. The result of the first stage based on BFGS is shown, and it is not binary at all, presenting a lot of intermediate values for the permittivity. Once the structure has been discretized, its performances can be further improved using our discrete optimization with smoothing. This leads to a different solution, in which however some characteristics of the starting point provided by BFGS can be found. This shows that the algorithm does largely modify the structure to improve its performance. Interestingly, the structure produced is relatively smooth: few black pixels seem to be isolated, even though this is not especially enforced in our method, making the structure easy to grasp. Two channels leading to the two output waveguides on the right can be distinguished.

Budget	BFGS	BFGS + Lengler + Smoothing	Progress
High resolution case (202x202)			
80	0.201	0.186	0.014
160	0.199	0.186	0.013
320	0.199	0.186	0.012
640	0.198	0.186	0.011
1280	0.197	0.185	0.012
2560 (average)	0.195	0.181	0.014
2560 (best)	0.182	0.161	0.021

Table 3: Loss (to be minimized) obtained by BFGS with restart vs BFGS followed by Lengler's method equipped with a smoothing operator for the demultiplexer. The progress (represented in the last column) is positive when the combination of the three algorithms BFGS + Lengler + Smoothing is better than BFGS with restarts. The BFGS is exactly the method used in [17]. It is generally better to use the combined method than the BFGS with restart alone. The values presented here are averaged over 10 runs at least. The two last lines present the results with a different random initialization as described in Section 2: instead of choosing randomly each pixel, we chose the starting value randomly for groups of $h \times h$ pixels. The last line of the table presents however the *best result* reached using BFGS and our approach, something which has more importance when solving a design problem.

Figure 3: The demultiplexer in the highest resolution (202×202) and for the lowest value of the cost function: after BFGS (left), after BFGS and discretization (middle) and after having run the algorithm Lengler equipped with the smoothing operator (right). The material with high permittivity is in black, while air is represented in white. There are significant differences before the discrete algorithm begins and the final result.

3.3. Grating coupler

The latest case in the SPINS-B code deals with a grating coupler, able to transfer the power of a large incident gaussian beam with a 10 degrees angle of incidence to a waveguide placed on top of a substrate. The cost function here is simply $1 - power$, meaning that the power transmitted into the waveguide is maximized. The structure is composed of four materials : a silicon substrate 2 µm under the grating, a silicon oxide layer, a 220 nm thick silicon layer that can be half-etched eventually, and a superstrate of silicon oxyde[31, 32]. The method provided by the SPINS-B package relies on two stages: first BFGS followed by a discretization, and then a parametric optimization where the position and width of the different blocks are modified. The parametrized design optimization is easy to apply here because the design is a one-dimensional array of pixels: the parameters are the positions (x-axis) of the frontiers between areas. The algorithm to reduce the cost function is the Sequential Least Squares Programming (SLSQP). This method is referred to as (BFGS)+(SLSQP) in the following, BFGS being followed by a discretization step and (SLSQP) standing for the discrete parametrized design optimization. The (SLSQP) stage relies on continuous parameters (widths of blocks), and can be deemed discrete in the sense that permittivities are discrete (the two extreme values). We compare this method to our approach with Lengler's algorithm and a smoothing operator after a first stage using BFGS and discretization. (Lengler/smoothing) stands for the Nevergrad tools, namely Lengler + Smoothing run for a budget equivalent to the (halting criterion dependent) budget used by (BFGS).

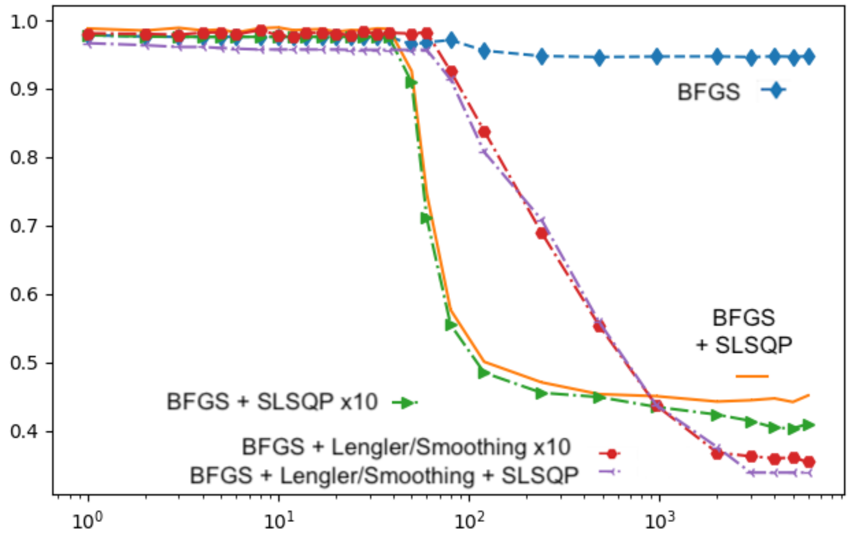


Figure 4: Comparison between various methods used for optimizing the grating design. The common part based on BFGS stagnates quickly: in the original code (BFGS+SLSQP), (SLSQP) uses more time than (BFGS). All methods based on non-differentiable optimization for improving the design were better. The best solution is to combine all: BFGS for the beginning, then the non-differentiable optimization (Lengler+Smoothing), and then the parametric optimization with SLSQP[17]. A suffix $\times 10$ after a method name means how we compute the budget for that method as follows. The budget of BFGS is determined by its halting criterion. (SLSQP) has a halting criterion, but (SLSQP$\times$10) means that we force 10 restarts. Lengler/Smoothing $\times 10$ means that we use the budget used by BFGS, but multiplied by 10. For a fair number of function evaluations for each method, we use random restarts when an algorithm halts before its entire budget is elapsed. The original solution in [17] was similar but without the non-differentiable optimization: in all the benchmarks from [17], adding non-differentiable optimization leads to improved results. When a non-differentiable optimization part is applied, the parametric (SLSQP) part provides only a minor improvement.

Fig. 4 shows the value of the cost function reached, as a function of the budget. For each different value of the budget, new runs are launched, so that each point is independent from the others on the same curve. The BFGS algorithm does not, here, provide a noticeable progress. The cost function remains relatively constant until the beginning of the second stage. While the parametric optimization provided in SPINS-B is quicker to converge, it ultimately converges to a larger value than Lengler's algorithm equipped with smoothing. Restarting BFGS each time it has found a local minimum is not as efficient as for the two previous cases, as the associated values of the cost function remain high. This makes this grating case peculiar. It is interesting to see that our approach works finally just as well as in the two previous cases, despite the fundamental differences.

4. Discussion

As can be seen from the results, the combination of Lengler's algorithm equipped with the simplest possible smoothing operator to make the algorithm more aware of the physical surrounding of each pixel, using a starting point provided by BFGS, provides in each case the best possible results. It thus seems we have identified in this work a discrete algorithm which seems particularly adapted to the optimization of binary structures (made of only two materials) when they are divided into pixels. The starting point provided by BFGS presents variations with typical size that are of the order of the wavelength inside the high index material. This is very different from a purely random starting point because it is influenced by the physics of the problem already, and this explains to our advice, why they are much better starting points.

We have seen that the result produced by the discrete optimization is noticeably different from the starting point provided by BFGS. However, it can be seen that the results of the optimization are essentially composed of well defined blocks, something which we have not imposed. The fact that we have not imposed such a constraint points to a characteristics imposed by the physics of the problem. This is anyway good for the buildability of the design.

Not all the problems we have considered here present the same level of difficulty for the algorithms. The waveguide bend is simple, in that the largest possible score (equivalently the lowest physically possible cost function) is easily reached. This means it is realistic, given the miniaturized size of the device, to have light make a 90 degrees bend. In that case the gradient based approach (BFGS and discretization) as well as our discrete method seem to generate very reliably efficient structures that look quite similar. It is never possible to truly assess whether a global optimum has been reached, so that only the degree of confidence which can be placed in a given solution has always to be discussed. This is what we call the quality of the solution. The kind of solution provided here can be deemed to be of high quality because (i) it emerges systematically, (ii) there is a physical limit to the lowest value of the cost function, which is essentially reached and (iii) the solutions provided can be understood physically, as a structure resembling a tilted Bragg mirror emerges (which is quite common in photonics[33, 7, 34]).

The solutions produced in the demultiplexer problem do not possess the same degree of quality, which indicates that the problem is more difficult. The BFGS algorithm provides solutions which are very sensitive to the starting point[17] and the best values of the cost function are not found systematically. We note, as shown by results in gray in Table 3, that modifying the correlation between pixels at the random initialization can strongly improve results both for BFGS and for our methods. The solutions are more difficult to understand physically, even if some regularity emerges spontaneously, a very important sign identified a long time ago[33]. More precisely, as identified by the authors of SPINS-B[17], in most of the designs produced by the algorithms, two channels leading to the two outputs emerge, which are somehow periodical. This points to an interferential mechanism allowing to reject one wavelength or another from each waveguide.

The grating case can, quite surprisingly, be deemed very difficult for gradient-based algorithms, which are seemingly not able to produce efficient solutions – despite the numerous restarts in the case of a large budget. This is why a second, parametric or discrete stage is absolutely necessary to reach interesting performances, whereas in the previous cases restarting BFGS proved to be largely enough. We underline, too, that even though the solution we have found using Lengler's algorithm is better than any reached using the parametric stage, we do not find it is yet fully satisfying (see appendix) – which is a sign the problem is difficult.

The SPINS platform is actually more of a way to organize the optimization, setting up different stages. We are aware that, in the present case, we compared our algorithms to a simple run of BFGS followed by discretization and that there could be a sequence of optimizations based on BFGS leading to better results than the one we have presented. In their original work, the authors of SPINS use a more complex sequence[17]. Furthermore, some authors rely on techniques involving penalizing non binary structures – a very different approach[9]. We thus do not claim that our approach is definitely the best, but that discrete optimization algorithms can be very serious candidates that have been so far essentially overlooked.

We are rather convinced that nobody, so far, has hardly found any high quality solution to the problem of the multiplexer, despite its practical importance. While we begin to comprehend how the structures produced may work, they are still too complex to be physically analyzed thoroughly. We think however that such a convincing solution is within our reach, and that it can be obtained through a combination of algorithms and maybe some physical input. Given the complexity and the diversity of the solutions, maybe guiding the optimizations could lead to interesting results. In some cases[10] parts of the structure which are not very useful can be manually erased. This is exactly how a gradient based method stuck in a local minimum would look like: a structure with parts that play almost no role, but whose cutting off would be only slightly detrimental to the efficiency. Maybe the path to follow would be to run another optimization after having modified the structure to match our physical intuition. We underline that, generally, any physical input usually helps optimizations immensely.

As we are discussing the physics of the structure, we think an important open question remains. Is the best possible solution a continuous one, with intermediate values of the permittivity, or is it binary ? The authors of SPINS-B seem to lean towards the first answer, as they use the results of BFGS without discretization to estimate the highest reachable efficiency[17]. However, our results show that discrete structures can be particularly efficient. We underline that in multilayered structures it has been proven that the best structures present a maximum index contrast, something which is found in parametric optimization and in particularly high quality solutions[35, 7, 36].

As underlined at the very beginning of this paper, the practical importance of being able to provide efficient photonics components can be far reaching. As widely recognized[5], we are not there yet because the quality of the solution reached is not completely satisfying. Many in the photonics community still doubt inverse design approaches will be up for the challenge – mainly because no design rule can really be deduced from the solutions proposed as long as we are not able to understand how they work in detail. Such a complex problem cannot be solved by one individual or one team. Coming up with good designs requires a lot of work. We were able to contribute here by identifying interesting algorithms and proposing improved designs because we have been relying on openly available codes[17, 19]. We would like to underline that, because this is not always the case in the field of optimization[37, 38]: the domain is going trough a worrisome reproducibility crisis. We thus made the codes which allowed us to produce the present results freely available[20]. We cannot stress more that open science, and the sharing of codes and methods openly[39] is the only way we can ever overcome all the obstacles which will arise on the road to versatile and reliable photonic components.

Acknowledgments

A.M. is an Academy CAP 20-25 chair holder. He acknowledges the support received from the Agence Nationale de la Recherche of the French government through the program Investissements d'Avenir (16-IDEX-0001 CAP 20-25).

References

[1] D. Thomson, A. Zilkie, J. E. Bowers, T. Komljenovic, G. T. Reed, L. Vivien, D. Marris-Morini, E. Cassan, L. Virot, J.-M. Fédéli, et al., Roadmap on silicon photonics, Journal of Optics 18 (7) (2016) 073003.

[2] A. V. Tikhonravov, M. K. Trubetskov, Development of the needle optimization technique and new features of OptiLayer design software, in: Optical Interference Coatings, Vol. 2253, International Society for Optics and Photonics, 1994, pp. 10–20. doi:10.1117/12.192109.

[3] G. Angeris, J. Vuckovic, S. P. Boyd, Computational bounds for photonic design, ACS Photonics 6 (5) (2019) 1232–1239.

[4] A. Y. Piggott, J. Lu, K. G. Lagoudakis, J. Petykiewicz, T. M. Babinec, J. Vučković, Inverse design and demonstration of a compact and broadband on-chip wavelength demultiplexer, Nature Photonics 9 (6) (2015) 374–377, number: 6 Publisher: Nature Publishing Group. doi:10.1038/nphoton.2015.69.
URL https://www.nature.com/articles/nphoton.2015.69

[5] S. Molesky, Z. Lin, A. Y. Piggott, W. Jin, J. Vucković, A. W. Rodriguez, Inverse design in nanophotonics, Nature Photonics 12 (11) (2018) 659–670, publisher: Nature Publishing Group.

[6] P.-I. Schneider, X. Garcia Santiago, V. Soltwisch, M. Hammerschmidt, S. Burger, C. Rockstuhl, Benchmarking five global optimization approaches for nano-optical shape optimization and parameter reconstruction, ACS Photonics 6 (11) (2019) 2726–2733.

[7] M. A. Barry, V. Berthier, B. D. Wilts, M.-C. Cambourieux, P. Bennet, R. Pollès, O. Teytaud, E. Centeno, N. Biais, A. Moreau, Evolutionary algorithms converge towards evolved biological photonic structures, Scientific reports 10 (1) (2020) 1–10.

[8] M. P. Bendsoe, O. Sigmund, Topology optimization: theory, methods, and applications, Springer Science & Business Media, 2003.

[9] J. S. Jensen, O. Sigmund, Topology optimization for nano-photonics, Laser & Photonics Reviews 5 (2) (2011) 308–321.

[10] L. F. Frellsen, Y. Ding, O. Sigmund, L. H. Frandsen, Topology optimized mode multiplexing in silicon-on-insulator photonic wire waveguides, Optics express 24 (15) (2016) 16866–16873.

[11] C. M. Lalau-Keraly, S. Bhargava, O. D. Miller, E. Yablonovitch, Adjoint shape optimization applied to electromagnetic design, Optics Express 21 (18) (Sep. 2013).

[12] E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov, O. Sigmund, Efficient topology optimization in matlab using 88 lines of code, Structural and Multidisciplinary Optimization 43 (1) (2011) 1–16.

[13] M. P. Bendsøe, O. Sigmund, Material interpolation schemes in topology optimization, Archive of applied mechanics 69 (9) (1999) 635–654.

[14] R. Martinez-Cantin, Bayesopt: a bayesian optimization library for nonlinear optimization, experimental design and bandits., J. Mach. Learn. Res. 15 (1) (2014) 3735–3739.

[15] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, D. D. Cox, Hyperopt: a Python library for model selection

and hyperparameter optimization, Computational Science & Discovery 8 (1) (Jul. 2015). doi:10.1088/1749-4699/8/1/014008.

[16] P. Bennet, C. Doerr, A. Moreau, J. Rapin, F. Teytaud, O. Teytaud, Nevergrad: black-box optimization platform, ACM SIGEVOlution 14 (1) (2021) 8–15.

[17] L. Su, D. Vercruysse, J. Skarda, N. V. Sapra, J. A. Petykiewicz, J. Vučković, Nanophotonic inverse design with spins: Software architecture and practical considerations, Applied Physics Reviews 7 (1) (2020) 011407.

[18] M. LLC, Spins-b 0.0.2 (2022).
URL https://github.com/stanfordnqp/spins-b

[19] J. Rapin, O. Teytaud, Nevergrad - A gradient-free optimization platform, https://GitHub.com/FacebookResearch/Nevergrad (2018).

[20] O. Teytaud, This fork: adding black-box optimization in spins-b (2022).
URL https://github.com/teytaud/spins-b/

[21] D. C. Liu, J. Nocedal, On the limited memory BFGS method for large scale optimization, Math. Program. 45 (1-3) (1989) 503–528.
URL https://doi.org/10.1007/BF01589116

[22] B. Doerr, H. P. Le, R. Makhmara, T. D. Nguyen, Fast genetic algorithms, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '17, ACM, 2017, pp. 777–784.

[23] D. Dang, P. K. Lehre, Self-adaptation of mutation rates in non-elitist populations, in: Parallel Problem Solving from Nature - PPSN XIV - 14th International Conference, 2016, pp. 803–813.

[24] R. Storn, K. Price, Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces, J. of Global Optimization 11 (4) (1997) 341–359.
URL https://doi.org/10.1023/A:1008202821328

[25] J. H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, 1975.

[26] B. Doerr, C. Doerr, J. Lengler, Self-adjusting mutation rates with provably optimal success rules, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '19, Association for Computing Machinery, 2019, p. 1479–1487.

[27] J. Kennedy, R. C. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, 1995, pp. 1942–1948.

[28] G. R. Harik, F. G. Lobo, D. E. Goldberg, The compact genetic algorithm, Trans. Evol. Comp 3 (4) (1999) 287–297.
URL http://dx.doi.org/10.1109/4235.797971

[29] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, Evolutionary Computation 11 (1) (2003).

[30] L. Meunier, H. Rakotoarison, P. K. Wong, B. Roziere, J. Rapin, O. Teytaud, A. Moreau, C. Doerr, Black-box optimization revisited: Improving algorithm selection wizards through massive benchmarking, IEEE Transactions on Evolutionary Computation 26 (3) (2021) 490–500.

[31] L. Su, R. Trivedi, N. V. Sapra, A. Y. Piggott, D. Vercruysse, J. Vučković, Fully-automated optimization of grating couplers, Optics express 26 (4) (2018) 4023–4034.

[32] N. V. Sapra, D. Vercruysse, L. Su, K. Y. Yang, J. Skarda, A. Y. Piggott, J. Vučković, Inverse design and demonstration of broadband grating couplers, IEEE Journal of Selected Topics in Quantum Electronics 25 (3) (2019) 1–7.

[33] A. Gondarenko, S. Preble, J. Robinson, L. Chen, H. Lipson, M. Lipson, Spontaneous emergence of periodic patterns in a biologically inspired simulation of photonic structures, Physical review letters 96 (14) (2006) 143904.

[34] Y. Brûlé, P. Wiecha, A. Cuche, V. Paillard, G. C. des Francs, Magnetic and electric purcell factor control through geometry optimization of high index dielectric nanostructures, Optics Express 30 (12) (2022) 20360–20372.

[35] A. V. Tikhonravov, M. K. Trubetskov, G. W. DeBell, Application of the needle optimization technique to the design of optical coatings, Applied Optics 35 (28) (Oct. 1996). doi:10.1364/AO.35.005493.

[36] P. Bennet, P. Juillet, S. Ibrahim, V. Berthier, M. A. Barry, F. Réveret, A. Bousquet, O. Teytaud, E. Centeno, A. Moreau, Analysis and fabrication of antireflective coating for photovoltaics based on a photonic-crystal concept and generated by evolutionary optimization, Physical Review B 103 (12) (2021) 125135.

[37] K. Sörensen, Metaheuristics—the metaphor exposed, International Transactions in Operational Research 22 (1) (2015). doi:10.1111/itor.12001.

[38] M. Hutson, Artificial intelligence faces reproducibility crisis (2018).

[39] J. Jiang, R. Lupoiu, E. W. Wang, D. Sell, J. P. Hugonin, P. Lalanne, J. A. Fan, Metanet: a new paradigm for data sharing in photonics research, Optics Express 28 (9) (2020) 13670–13681.

[40] Y. Collette, N. Hansen, G. Pujol, D. Salazar Aponte, R. Le Riche, Object-oriented programming of optimizers–examples in scilab, Multidisciplinary Design Optimization in Computational Mechanics (2013) 499–538.

Appendix A. Algorithms

We give, in this appendix, an overview of the different algorithms that we have used under a form that will allow the reader to better grasp the details of our experiments.

We described the algorithms with the standard Ask, $Tell$, $Initialize$ and $Recommend$ methods. The Ask method provides a new candidate or set of parameters to evaluate. Once evaluated, the user returns the candidate and the corresponding loss through the $Tell$ method. The $Recommend$ method provides the optimized set of parameters at the end of the optimization. Each algorithm has its own methods, so that the number of parameters for each one depends on the algorithm considered.

More details on the ”Ask and Tell” methods can be found in [40].

Algorithm 1 Smoothing operator. This operator is applied once per 55 mutations. We use the median: if we have two permittivities, we get one of those permittivities as a result.

for each pixel p **do**

  if random $< .25$ **then**

    Replace p by the median value of the 3×3 pixels around p (p and its 8 neighbours)

  end if

end for

Accept the mutation above if fitness is not worse

Algorithm 2 Our combination of BFGS, Lengler and the smoothing operation. The Lengler method starts at the point at which BFGS stagnates.

Require: an objective function f, its gradient function ∇f, a domain D, a budget b

$BFGS: Initialize(D, b)$: we initialize the algorithm given the domain D and the budget b

for $i \in \{1, \ldots, b\}$ with interruption in case of halting criterion raised (BFGS auto-detects its stagnation) **do**

  The algorithm provides $x \leftarrow BFGS: Ask()$ ($x \in D$)

  Compute $f(x)$ and $\nabla f(x)$

  $BFGS: Tell(x, f(x), \nabla f(x))$

end for

$b' \leftarrow$ number of calls to $tell$

$Bfgs - Recommendation \leftarrow BFGS: Recommend()$ with discretization to two permittivities

$Lengler: Initialize(D, b - b', Bfgs - Recommendation)$

for $i \in \{1, \ldots, b\}$ **do**

  $Lengler$ provides $x \leftarrow Lengler: Ask()$ ($x \in D$)

  if 55 divides i **then**

    for each pixel p **do**

      if random $<$ .25 **then**

        Replace p by the median of the 3×3 patch around p

      end if

    end for

  end if

  Compute $f(x)$

  $Tell(x, f(x))$

end for

Appendix B. Grating design

Here is the design provided by our algorithms. In a way, it is disappointing in that no regularity can be particularly found in the result of the optimizations despite the low values of the cost function reached. It is thus difficult to claim the solution we have found is actually a high quality solution (*i.e.* found systematically and physically understandable), but it seems more efficient than any solution found with the parametric optimization, according to the value of the cost function and to the field representedB.6.

Figure B.5: The black color corresponds here to the half-etched silicon embedded in silicon dioxyde.

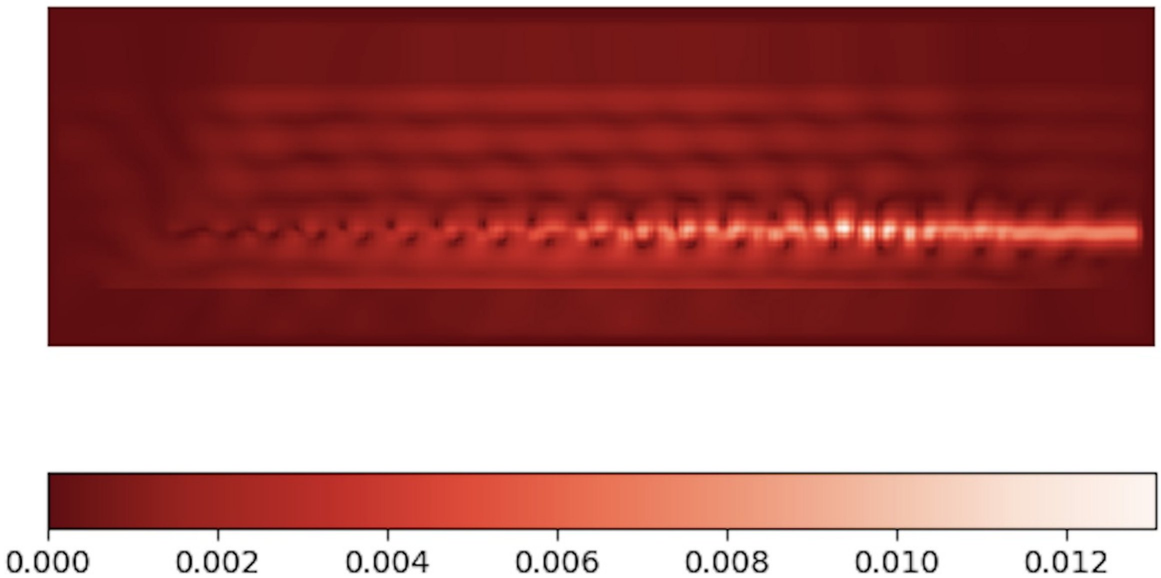


Figure B.6: Modulus of the field showing the coupling of the incident plane wave to the resonance of the grating and finally to the waveguide on the right.