Progress In Electromagnetics Research B, Vol. 59, 219–229, 2014

Optimization of a Plasmon-Assisted Waveguide Coupler Using FEM and MMP

Mengyu Wang*, Aytaç Alparslan, Sascha M. Schnepp, and Christian Hafner

Abstract—In this paper, we focus on the problem of optimizing plasmonic structures. A plasmon-assisted waveguide coupler is considered as a test problem, which leads to a five-dimensional optimization problem carried out by an evolution strategy (ES). The optimization results are verified by a comparative analysis between different solvers, i.e., the finite element package `CONCEPTs` and the multiple multipole program (MMP). We also compared with results obtained using a deterministic optimization algorithm, namely the Nedler-Mead method as implemented in the commercial software package `COMSOL Multiphysics`. Some issues concerning deterministic versus evolutionary optimization, in particular, in the field of plasmonics have been discussed.

1. INTRODUCTION AND BACKGROUND

Optical nano antennas are currently promising key elements for sensing and optical communication [1–4]. Like traditional antennas for radio frequencies up to the low THz regime, they are mainly made of metals; however, they exhibit considerable differences because of plasmonic resonances that occur in the optical regime.

Traditional antennas essentially are scalable, i.e., their resonance wavelengths are proportional to the antenna size, which allows one to derive rather simple engineering formulas for the design of such antennas [5, 6]. Typically, the first resonances are exploited in order to keep the antennas small and consequently, the typical antenna size is of order half a wavelength. In many applications, the bandwidth of traditional antennas is increased by special geometries that provide several resonances over a certain band.

In contrast, plasmonic nano antennas may resonate even when they are much smaller than the wavelength. Their bandwidth in the optical range is often broader than required. Although the geometry has a strong impact on resonances and on the antenna performance, the bandwidth is mostly caused by the rather high losses in metals at optical frequencies. As a result, plasmonic nano antennas are not scalable, their quality factor is usually not very high, and it is very hard or even impossible to find simple design rules.

Currently, the most promising approach to designing plasmonic nano antennas for a specific application is to benefit from appropriate combinations of numerical optimizers with simulation tools for electromagnetics that may efficiently simulate plasmonic structures [7, 8]. In the last few years, many numerical optimizations methods have been developed and several numerical optimizers have been embedded in various commercial simulation tools, such as `COMSOL Multiphysics` [9]. The main issues are the following: 1) In order to achieve acceptable simulation accuracy, the simulation time of a single plasmonic structure may be rather long. 2) Working on massively parallel computers may become extremely expensive because of high license costs of commercial software. Therefore, one might prefer using some freely available simulation tool which is typically less user-friendly than a commercial one. 3) The optimizer will usually design many different structures. Some of them may look rather

Received 7 March 2014, Accepted 16 April 2014, Scheduled 18 April 2014
* Corresponding author: Mengyu Wang (wangm@ethz.ch).
The authors are with the Institute of Electromagnetic Fields, ETH Zurich, CH-8092, Switzerland.

astounding. Despite of this, the solver should not fail and return a reasonable result, i.e., the solver must be completely automatic and robust without any support by the user. 4) Inaccuracies of the solver may disturb the numerical optimizer considerably. This especially holds for deterministic optimizers that approximate the gradient information in the parameter search space as soon as they are in the vicinity of an optimum. As a consequence, techniques to speed up the simulation tool by reducing its accuracy may drastically increase the overall cost of the optimization. 5) The overall computation time can become extremely long and depends heavily on the collaboration of the simulation tool with the optimizer. 6) Without any prior knowledge on the complexity of the optimization problem (smoothness of the fitness landscape in the search space, number and shape of local optima, etc.), it is impossible to select an appropriate optimizer. 7) Obtaining information on the complexity of the optimization problem may be hard, especially when the search space is high-dimensional and because even small parts of plasmonic structures may have a considerable impact on the solution.

In this paper, we study a relatively simple, two-dimensional plasmonic waveguide coupler for exciting a guided wave in a dielectric slab in order to illustrate the various difficulties mentioned above. The coupler consists of only two circular metallic particles, which leads to five natural optimization parameters, namely the radii of the particles, their distances from the surface of the waveguide and the distance between the particles. In Sections 2–4, we outline the available numerical optimizers and electromagnetic field solvers. Then in Section 5, we apply the evolution strategy using two fundamentally different solvers, one is MMP based on a boundary discretization method [10], and another is `CONCEPTs` based on a domain discretization method [11, 12]. We perform the statistical analysis of the optimizer using different solvers. Then we show how the requirements of automatic and robust performance of the field solver may be obtained and how the results of a stochastic optimizer may be analyzed in order to obtain information on the complexity of the optimization problem. The optimization results are shown in Section 6. From the results, we find that even a simpler direct optimizer should perform well, at least when a start point near the global optimum of interest is set. Because of this finding, at the end of Section 6, we take advantage of the Nelder-Mead method [13], which is available in `COMSOL Multiphysics`, and compare the results.

2. NUMERICAL OPTIMIZERS

In the design of geometric structures, one often faces high-dimensional optimization problems that may only be tackled numerically. Usually, a real-valued fitness function or cost function is defined and the optimizer has to locate local or global maxima of the fitness function or minima of the cost function. Whether one prefers working with fitness or cost functions is not relevant. In all demanding cases, the optimization problem is non-linear and an iterative search has to be performed. This search may be subject to additional constraints.

For a long time, mathematicians focused on various deterministic algorithms for non-linear optimization in high-dimensional parameter search spaces [14]. These algorithms are very mature and efficient when either the starting point of the search is close to the desired optimum or when the optimization problem is simple enough. Many of these algorithms require gradient information or even second order derivatives of the fitness function, which is usually not available when the fitness is calculated from a numerical solver as in our example. Therefore, only deterministic optimizers without gradient information are useful for most of the engineering applications. A prominent example is the Nelder-Mead algorithm that is also known as downhill simplex method [13]. This algorithm is efficient when the number of dimensions of the parameter search space is rather low, and when there are not many local optima within the search space. In our 5-dimensional test example, Nelder-Mead could be applied, if we knew that there is only one optimum in the search space, or if we could approximately guess its location.

If one is unsure about the problem complexity, one may benefit from an optimization strategy that is not purely deterministic and includes some randomness. Such optimizers have been studied intensively in the second half of the 20th century. Many of them are inspired by optimization processes that seem to be ongoing in physics and nature. An excellent overview may be found in [15].

Currently, the most widely applied nature-inspired algorithms in engineering are probably genetic algorithms (GAs) [15, 16]. These algorithms mimic the evolution of animals (and plants), where the genotype (genetic material in the chromosomes, the genetic code) is separated from the phenotype (the visible properties of the animal). The genotype of a GA is typically a bit string, but the binary basis can be replaced easily by some other basis. The phenotype in the case of an N-dimensional parameter optimization problem is simply a point in the search space, i.e., an N-dimensional vector containing the N optimization parameters. In order to map the genotype, e.g., the bit string, on the phenotype, some decoding routine must be provided by the user of the GA. Since it is not unique at all, a good choice of the decoding is highly important for the GA performance and is often not simple. As a result, standard GAs are usually outperformed by Evolution Strategies (ESs) [15, 17], which are closely related to GAs without using the genotype-phenotype scheme with its requirement for coding-decoding. In an ES, the individual is directly characterized by its N-dimensional parameter vector (the phenotype) and usually, by a variation vector that is also N-dimensional in typical ES implementations. The variation vector indicates how much mutation may modify the individual. Beside these differences both GAs and ESs work with populations of some individuals. They select the best ones as parents for children of the next generation, create the children by crossover of typically two parents followed by mutation with some probability, and discard the most unfit individuals.

For our 5-dimensional optimization (see Section 3) problem, we apply a standard $(\mu+\lambda)$ ES with the number of parents $\mu = 5$ and the number of children $\lambda = 7\mu$. The factor 7 is known to be a reasonable choice for standard ES. This algorithm can be downloaded from [18]. A GA or another nature-inspired or stochastic parameter optimizer might be applied as well, possibly with some additional computational costs.

3. TEST PROBLEM

Our goal is to optimize the performance of a 2D plasmon-assisted waveguide coupler, having the maximum power coupled into the waveguide. With the help of plasmonic structures, the power can feed into the waveguide [3, 4].

Our test problem is shown in Figure 1, where a dielectric waveguide with permittivity 4.0 is mounted on a glass substrate with permittivity 2.25, and the region above is the free space. An H_z polarized planewave with 600 nm wavelength is impinging from top-left at an angle of 45 degrees. Two silver cylinders are embedded in the waveguide. Under this frequency, the permittivity of silver is $-15.855 + 0.432i$ [19] and therefore the cylinders behave as a plasmonic coupler. The thickness of the waveguide is 200 nm, and for the left cylinder, the distances between the upper and lower interface of the waveguide are v_1 and v_2, and for the right cylinder v_4 and v_5. The distance between the two cylinders is v_3.

In order to optimize the power coupled into the waveguide, we place an observer interface $L = 1300$ nm away from the center of the left cylinder. One can choose the power flux through L

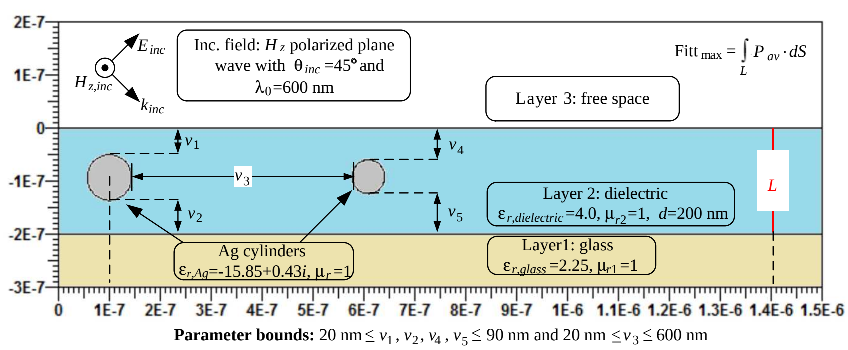

Figure 1. The geometrical configuration of the test problem. The dielectric waveguide with permittivity 4.0 is placed in between the glass and air. Two silver cylinders are embedded in the waveguide. The structure is excited by an H_z polarized plane wave with 600 nm wavelength. The location of the cylinders is characterized by $\bar{V} = \{v_1, v_2, v_3, v_4, v_5\}$, which are the parameters of the optimization problem.

as the fitness function F.

$$F = \int_L P^{AV} dl.$$

In our experiments, in order to compare the results among different methods, we apply the normalized fitness function $\hat{F}$. We determine the best parameters among all methods, and compute the best fitness $F_{\max}$ using these parameters. Then we normalize the fitness function with $F_{\max}$. The normalized fitness function $\hat{F}$ reads

$$\hat{F} = \frac{\int_L P^{AV} dl}{F_{\max}}.$$

The way of selecting the geometrical parameters may seem a bit un-natural at the first glance; however, it works very well with the optimizer. The greatest advantage is that all the five parameters do not interfere with each other. Otherwise, if one selects the radii of the cylinders as the parameters, some additional constrains have to be applied in order to avoid the cylinders to touch the substrates or to collide. Secondly, it is rather simple to map the data into range [0, 1], which fits the standard data interface for our optimizer.

In our experiments, $\bar{V} = \{v_1, v_2, v_3, v_4, v_5\}$ represents the data set with $v_1, v_2, v_4, v_5 \in [20, 90]$ nm, and $v_3 \in [20, 600]$ nm. For the numerical optimizer, we normalize these parameters by linear mapping on $\bar{P} = \{p_1, p_2, p_3, p_4, p_5\}$, with $p_1, p_2, p_3, p_4, p_5 \in [0, 1]$. Hence, the mapping between the two sets is

$$\begin{aligned} v_i &= 20 + 70 p_i \,\text{nm}, \quad i = 1, 2, 4, 5 \\ v_3 &= 20 + 580 p_3 \,\text{nm}, \end{aligned}$$

which represents a 5-dimensional optimization problem with standard parameter set $\bar{P}$.

4. SOLVERS FOR PLASMONIC STRUCTURES

In computational electromagnetics one may characterize the available numerical methods by the way they handle time and space. This leads to the two categories of time-domain and frequency-domain solvers. Time-domain solves such as FDTD attracted much interest and were also used for plasmonics. However, FDTD requires an extremely fine discretization in space (0.5 nm grids) and time (because of the stability criterion) [20] and suffers from undesired stair-casing effects. Another problem is caused by the fact that metals are strongly dispersive, i.e., their permittivity is strongly frequency-dependent at optical frequencies. To handle this problem, simplified Drude and Lorentz models are usually applied, which introduce additional errors, especially in the area of interest, where the real part of the relative permittivity of the metal is changing its sign. Since we want to avoid errors of the field solvers that might disturb the optimizer, and since we intend to optimize the plasmonic waveguide coupler for a certain frequency, we do not consider time-domain solvers in the following.

Since most of the configurations of interest consist of materials separated by interfaces or boundaries, one may have two categories of space discretization, domain discretization, where the entire space is discretized by some elements of finite size (Finite Element Method FEM) or where only the interfaces or boundaries are discretized. When the boundaries are discretized by elements of finite size, the term Boundary Element Method (BEM) is used.

FEM and similar domain discretization methods suffer from the fact that many structures of interest are not finite and need an appropriate truncation, typically by introducing absorbing boundary conditions (ABCs) [21] or perfectly matched layers (PMLs) [22]. ABCs and PMLs require additional implementation effort and may introduce additional errors, which must be carefully checked when numerical optimizations are being performed. However, since FEM is most widely used, we want to apply it also in the following.

In order to get information on the FEM accuracy, we need an accurate simulation tool, preferably one that is based on a boundary discretization technique. Since the boundary discretization by means of boundary elements may introduce sharp wedges and corners — which may cause numerical problems in plasmonics — we use the multiple multipole program (MMP) [10], which is an element-free boundary discretization technique that is very close to analytic solutions and therefore may be used as a reference.

It is important to note that something similar to domain truncation by means of ABCs and PMLs is often not required in boundary discretization methods because the boundaries are usually finite. A typical example is an antenna surrounded by free space. The boundary of such an antenna, i.e., its interface to free space is indeed finite. In our example of an antenna inside a waveguide (for coupling a plane wave into the waveguide), the boundaries of the antenna are finite, but the boundaries of the waveguide are assumed to be infinite since the waveguide is much longer than the wavelength. Consequently, we need some appropriate procedure to discretize such a structure. This will be outlined in Section 5. As a result, we also introduce some error in the MMP model and it is difficult to figure out whether the MMP or the FEM error is higher. Since the two methods are fundamentally different, we can assume that the numerical error is of order of the differences between the MMP and FEM results. Thus, we can at least estimate the error of our simulations.

5. SIMULATION METHODS

5.1. Boundary Discretization — MMP

As a reference solution for the numerical optimization problem described in Section 3, we used MMP together with a genetic optimization algorithm. In order to solve the scattering problem in the given layered medium and calculate the corresponding fitness values for the optimization routine, we use the open source numerical implementation of MMP and layered media Green's functions called OpenMaXwell [23]. By using the automatic expansion distribution routines of OpenMaXwell for the MMP solution, we obtain the results in an efficient and robust way for all the different optimization parameter combinations (the boundary condition mismatch error measured on the scatterers is less than 0.1 percent for the numerical results presented in Section 6). For a detailed description of MMP analysis for layered media, refer to [24–26].

5.2. Domain Discretization — FEM/`CONCEPTs`

We choose the C++ library `CONCEPTs` as our FEM solver, which features high polynomial basis functions and quadrilateral curvilinear elements. Our previous research shows that `CONCEPTs` with its PML implementation provides high accuracy and efficiency for the problems of plasmonic waveguides, the details can be found in [27].

The ES from [18] is rewritten into MATLAB scripts to control the optimization process. For each generation, according to the current fitness distribution, the ES scripts generate a new generation, and then pass the parameters to the mesh generator to generate parametric meshes. The mesh generator is part of `COMSOL Multiphysics`. In order to make it parametrically controllable, we link `COMSOL` with a MATLAB routine with the help of `COMSOL` LiveLink for MATLAB. Then the `CONCEPTs` routine is called to read the parametric meshes and solve the electromagnetic field. Finally, the fitness distribution of the new generation is computed from the `CONCEPTs` results. The process is repeated until the maximum number of iterations is reached. Figure 2 shows the control flow of the optimization.

In practice, a number of issues require attention when using meshes as generated by `COMSOL`. Firstly, `CONCEPTs` only works with quadrilateral meshes; however, in some circumstances, `COMSOL` will generate meshes with mixed quadrilateral and triangular elements, as shown in Figure 3. Secondly, when utilizing quadrilateral curvilinear elements, if there is an element with two adjacent curved edges, as shown in Figure 4, the simulation will fail. As the plasmonic particles have circular shape, the incidence angle at

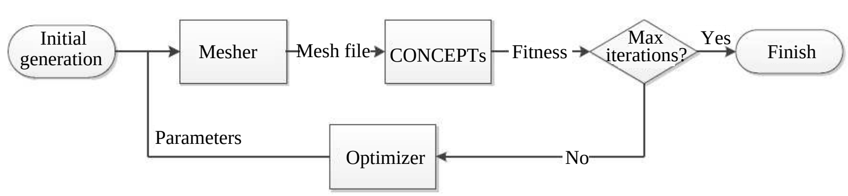

Figure 2. The control flow of the optimization. The framework of parametric mesh generation links `CONCEPTs` with `COMSOL`.

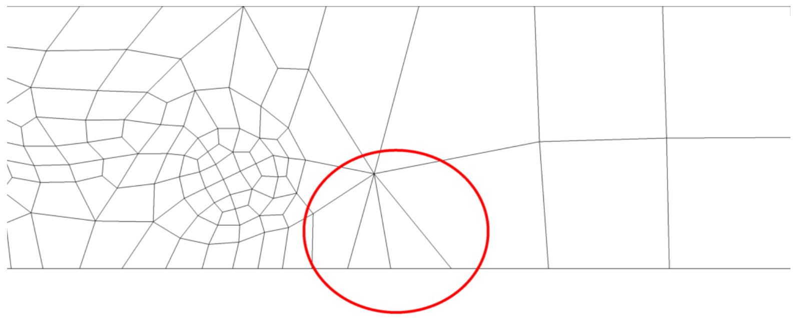

Figure 3. `COMSOL` mesh with mixed quadrilateral and triangular elements, which causes problems when solving with `CONCEPTs`. The red circle points out the problematic triangular elements. This problem can be solved by increasing the overall refinement level.

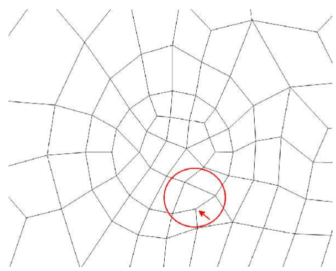

Figure 4. `COMSOL` elements with neighboring curved edges, which causes problems when solving with `CONCEPTs`. The red circle points out one of the problematic elements. This problem can be solved by a further refinement of the circular interface. Note that in the illustration, straight-sided elements are displayed as generated by `COMSOL`. Problems occur only when the elements are subsequently curved leading to coordinate degeneration at the node indicated by the arrow.

the grid node located between the two curved edges is 180 degrees. This leads to a degeneration of the local element coordinates at the node indicated by the arrow.

To overcome the above problems, we need to introduce control parameters in the MATLAB `COMSOL` scripts. For the first problem, in order to control the overall refinement level, we introduce parameter γ_1, which has the meaning that the minimal size of elements is $20/\gamma_1$ nm. When a triangular element is detected, the script will send requirement to increase γ_1. The best empirical values for our problem would be 1.05 as the starting value and 0.15 as the increment, respectively. For the second problem, we introduce parameter γ_2 to control the refinement level of the circular interface. It's defined in the way that the minimal size of the element on the circle is $R/2\gamma_2$, where R is the radius of the corresponding cylinder. When an element with two curved edges is detected, the algorithm will send requirement to increase γ_2. The best empirical values for our problem would be 1.2 as the starting value and 0.15 as the increment step, respectively.

In the `CONCEPTs` simulation, basis functions of polynomial degree up to 7 are used. The polynomial order 7 is a good choice for the fast simulation and high accuracy. The computational time varies according to the refinement level of the meshing step. A typical number of degrees of freedom (DOFs) is some thousands, and a single simulation takes around 6 to 20 seconds, depending on the complexity of the mesh.

6. OPTIMIZATION RESULTS

6.1. ES Optimization

6.1.1. MMP as Solver

For the genetic optimization routine, we use the parameters number of parents $\mu = 5$, number of children $\lambda = 35$, the strength of mutation $\sigma = 0.1$, and the strategy of keeping the best five individuals of the current generation, with a maximum number of solver calls of 1000. As a result, we obtained the parameters $\bar{V}_{\mathrm{MMP}} = \{20.00, 65.91, 434.64, 20.00, 77.34\}$ nm that gives the best fitness value. The distribution of the optimization parameters in the search space and the convergence of the fitness value are plotted in Figures 5(a) and 5(b), respectively. Figure 5(a) shows that the ES converges first to a local optimum. After 300 iterations, it increases the variability and then converges to the global optimum in the search space. Figure 5(b) shows the distribution of the five normalized parameters $\{p_1, p_2, p_3, p_4, p_5\}$ during the ES search. This plot gives a rough impression of the fitness landscape, especially near the optimum found. As one can see, all p_1 and p_4 values at high fitness levels are close to 0. This indicates that one might find better solutions by allowing smaller values for the corresponding

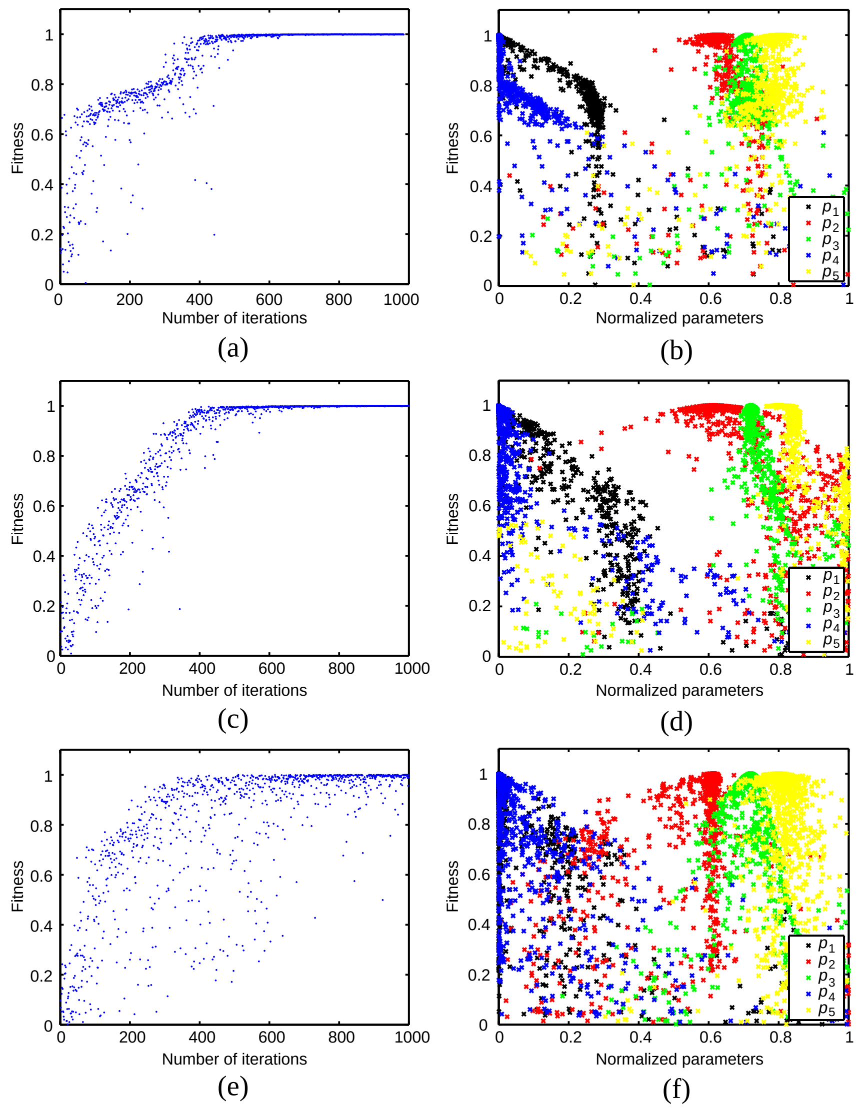

Figure 5. The results of the 5-dimensional optimization. The maximum number of individual simulations is 1000, with the number of parents $\mu = 5$, number of children $\lambda = 35$, and strength of mutation $\sigma = 0.1$. p_1 to p_5 are the normalized geometrical parameters, which are mentioned in Section 3. The strategy keeps the best 5 parent individuals in each generation. (a) Shows the convergence with respect to the number of iterations using MMP, and (b) shows the distribution of the normalized parameters versus the fitness function using MMP. (c), (d) are the corresponding results from the first run of `CONCEPTs` simulation, and (e), (f) are from the second run of `CONCEPTs` simulation. The optimizations look different each time, only due to the randomness of the optimizer.

parameters v_1 and v_4, which are the distances of the cylinders from the top interface. the shape of the envelopes of p_2 and p_5 are rather broad near the optimum. This indicates that fabrication inaccuracies of the corresponding v_2 and v_5 are not affecting the results very strongly. Note that v_2 and v_5 indirectly define the diameters of the cylinders. The sharpest peak can be observed for p_3, which indicates that the fabrication tolerance for the distance between the cylinders will be rather small.

6.1.2. `CONCEPTs` as Solver

We take the same configuration as of MMP, i.e., $\mu = 5$, and $\lambda = 35$, $\sigma = 0.1$, and keep the best five individuals of the current generation. The optimizations are performed twice, with maximum number of solver calls of 1000 each time. Figures 5(c), 5(e) show the convergence results, and Figures 5(d), 5(f) show the distributions of the parameters. As one can see, the ES convergence depends as much on the random initialization of the ES as on the selection of the field solver. The shapes of the envelopes of $\{p_1, p_2, p_3, p_4, p_5\}$ near the optimum are very similar in all cases, indicating that the interpretation in Section 6.1 is also valid here. The optimization reaches best fitness with $\bar{V}_{\mathtt{CONCEPTs}} = \{20.00, 62.89, 437.63, 20.00, 75.86\}$ nm, which are close to the MMP values. Note that the differences of the maximum fitness values found with the two `CONCEPTs` runs are around 0.1% and similar differences between the MMP and `CONCEPTs` solutions may be observed as well. The optimal values of v_1 and v_4 are always 20 nm, i.e., the minimum allowed distances from the top layer. The variations in the optimal v_2 and v_5 values are around 5%, which supports the finding of the rather high fabrication tolerances for v_2 and v_5. The more critical parameter v_3 has a variation of less that 1%, which is also in agreement with the interpretation of Figures 5(b), 5(d), 5(f).

Due to the randomness of the ES, the convergences look some what different, even though the optimization parameters are exactly the same. The first one is more 'lucky' that its convergence is faster and the distribution is denser compare to the second one. The optimization progress with FEM are also different from the one with MMP, this is also because of the randomness. The fact shows that the procedure of the ES optimization is uncontrollable, even with very good empirical optimization parameters.

6.2. Deterministic Optimization with `COMSOL`

We also used a deterministic optimizer to solve the same 5-dimensional optimization problem, in comparison with the optimization results of ES from Section 6.1. We used the Nelder-Mead method, which is embedded in the commercial FEM solver `COMSOL Multiphysics` version 4.3b. In the `COMSOL` simulation, we used the default values of the solver (second order shape functions with the largest mesh size set to be less than 1/10 of the incident wavelength in the corresponding domain) and the built-in

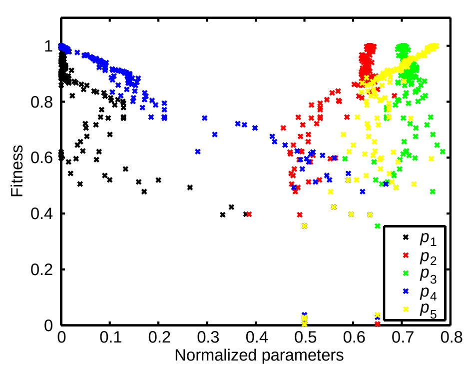

Figure 6. Five-dimensional optimization using Nelder-Mead method in `COMSOL`. There are 172 instances of simulations, with optimality tolerance of 0.01 and the initial values of the optimization set to be the mid-point of the corresponding parameter bounds. p_1 to p_5 are the normalized geometrical parameters, which are mentioned in Section 3. The figure shows the distribution of the parameters versus the fitness function.

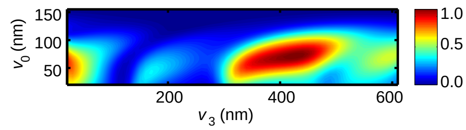

Figure 7. The fitness function of two-dimensional scanning using OpenMaXwell. The scanning parameters are the distance between the two cylinders v_3, and the lower distance from the substrate v_0, respectively.

Nelder-Mead optimization routine (optimality tolerance of 0.01 and the initial values of the optimization set to be the mid-point of the corresponding parameter bounds). The distribution of the optimization parameters in the search space are plotted in Figure 6.

As a deterministic algorithm, Nelder-Mead method requires only 172 instances to converge in this example. However, it works with more strict conditions than ES: the dimension of the optimization should not be very high, and there should not be many local optima within the search space. In order to check the above criteria, we approximated the local optima distribution by a reduced two-dimensional optimization problem. We reduced v_2, v_5 into one parameter, by fixing $v_2 = v_5$, which is noted as v_0. The other two parameter from the original problem are eliminated by fixing $v_1 = v_4 = 20\,\text{nm}$. And the last parameter is v_3, which is the distance between the two cylinders. Then the optimization problem reduces to a two-dimensional optimization problem with parameters v_0 and $v3$. With OpenMaXwell as solver, we scan v_0 from 20 nm to 150 nm, and v_3 from 20 nm to 600 nm. Figure 7 shows the fitness function with respect to the two parameters. From the figure, one can observe the distribution of the local optima for the reduced 2D optimization problem. From this information, one can approximate the distribution of the local optima for the original 5D problem. Our starting point is close to the global optima, therefore the optimization results is the global one.

However, different choices of the starting points can lead to a globally non-optimal solution. It is very important to make good initial guess for the Nelder-Mead method and most of the other deterministic optimization methods. Unfortunately, most of the plasmonic optimizations are high dimensional, and it's hard to make good initial guesses. Therefore, even though the computational cost can be very high, it is recommended to use the ES optimization for plasmonic problems because of the higher probability to find the global optimum.

7. CONCLUSIONS

A plasmon-assisted waveguide coupler was optimized using various methods. We considered five geometry parameters leading to a five-dimensional optimization, which was addressed by means of nature-inspired and deterministic algorithms.

We employed a robust optimizer based on a evolution strategy (ES), representing the category of nature-inspired optimization algorithms. `CONCEPTs` and the multiple multipole program (MMP) were used as different field solvers. `CONCEPTs` is a high-order FEM library. In order to link `CONCEPTs` with the optimizer and the mesh generator, a framework of parametric mesh generation was implemented. MMP is an element-free boundary discretization technique, and it can provide very accurate solutions close to the analytic ones. We performed a statistical analysis of the optimizer, and showed that the results using different solvers converged to very similar optimal parameters. The optimizer behaves robust for our problem. However, due to the randomness, the optimization convergence looked different each time, even with the same solver using exactly the same parameters.

We also used the Nelder-Mead method, a deterministic optimization algorithm, in `COMSOL Multiphysics`. The Nelder-Mead method is very efficient but with the following requirements: the number of dimensions of the optimization is low, and there are not many local optima within the search space. Our test problem is five-dimensional, which fulfills the first requirement. In order to check the second requirement, we scanned the reduced problem with only two parameters, and the approximated local optima distribution proved that Nelder-Mead can be applicable for our problem within the given range of the parameters.

However, for more general problems of plasmonic optimization, they are usually high dimensional, and more importantly, it is very difficult to make a good initial guess, which is likely to lead the deterministic optimizer into a local optimum. Therefore, although the computational cost of performing an ES optimization can be very high, it is outweighed by its much higher probability to find the global optima.

ACKNOWLEDGMENT

This work is supported by the Swiss National Science Foundation, under SNSF 119813. The authors acknowledge Kersten Schmidt, TU-Berlin, Germany, for his help and support in CONCEPTs. The authors acknowledge Nikolay Komarevsky and Martin Spieser, ETH Zurich, Switzerland, for their help and discussion of optimization.

REFERENCES

1. Anker, J. N., W. P. Hall, O. Lyandres, N. C. Shah, J. Zhao, and R. P. Van Duyne, "Biosensing with plasmonic nanosensors," *Nature Materials*, Vol. 7, No. 6, 442–453, 2008.
2. Baron, A., E. Devaux, J.-C. Rodier, J.-P. Hugonin, E. Rousseau, C. Genet, T. W. Ebbesen, and P. Lalanne, "Compact antenna for efficient and unidirectional launching and decoupling of surface plasmons," *Nano Letters*, Vol. 11, No. 10, 4207–4212, 2011.
3. Novotny, L. and B. Hecht, *Principles of Nano-optics*, Cambridge University Press, 2012.
4. Esslinger, M., W. Khunsin, N. Talebi, T. Wei, J. Dorfmüller, R. Vogelgesang, and K. Kern, "Phase engineering of subwavelength unidirectional plasmon launchers," *Advanced Optical Materials*, Vol. 1, No. 6, 434–437, 2013.
5. Kong, J. A., *Electromagnetic Wave Theory*, Wiley, New York, et al., 1986.
6. Balanis, C. A., *Antenna Theory: Analysis and Design*, John Wiley & Sons, 2012.
7. Komarevskiy, N., V. Shklover, L. Braginsky, C. Hafner, and J. Lawson, "Potential of glassy carbon and silicon carbide photonic structures as electromagnetic radiation shields for atmospheric re-entry," *Optics Express*, Vol. 20, No. 13, 14189–14200, 2012.
8. Mihaljevic, J., J. Niegemann, S. M. Schnepp, and C. Hafner, "On the numerical modeling of sharp metallic tips," *Quantum Matter*, Vol. 3, No. 4, 344–354, 2014.
9. COMSOL Multiphysics, 2014, http://www.comsol.com/.
10. Hafner, C., *Post-modern Electromagnetics: Using Intelligent Maxwell Solvers*, Wiley, 1999.
11. Schmidt, K. and P. Kauf, "Computation of the band structure of two-dimensional photonic crystals with *hp* finite elements," *Computer Methods in Applied Mechanics and Engineering*, Vol. 198, No. 13, 1249–1259, 2009.
12. Schmidt, K. and R. Kappeler, "Efficient computation of photonic crystal waveguide modes with dispersive material," *Optics Express*, Vol. 18, No. 7, 7307–7322, 2010.
13. Nelder, J. A. and R. Mead, "A simplex method for function minimization," *Computer Journal*, Vol. 7, No. 4, 308–313, 1965.
14. Bonnans, J.-F., J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal, *Numerical Optimization: Theoretical and Practical Aspects*, Springer, 2006.
15. Brownlee, J., *Clever Algorithms: Nature-inspired Programming Recipes*, Jason Brownlee, 2011.
16. Johnson, J. M. and V. Rahmat-Samii, "Genetic algorithms in engineering electromagnetics," *IEEE Antennas and Propagation Magazine*, Vol. 39, No. 4, 7–21, 1997.
17. Beyer, H.-G. and H.-P. Schwefel, "Evolution strategies — A comprehensive introduction," *Natural Computing*, Vol. 1, No. 1, 3–52, 2002.
18. Ch. Hafner's Generalized Genetic Program, (GGP), 2011, http://alphard.ethz.ch/Hafner/ggp/gp.htm.
19. Jackson, J. D., *Classical Electrodynamics*, 3rd Edition, 1999.
20. Taove, A. and S. C. Hagness, *Computational Electrodynamics, Volume 160*, Artech House, Boston, 2000.

21. Bayliss, A., M. Gunzburger, and E. Turkel, "Boundary conditions for the numerical solution of elliptic equations in exterior regions," *SIAM Journal on Applied Mathematics*, Vol. 42, No. 2, 430–451, 1982.
22. Berenger, J.-P., "A perfectly matched layer for the absorption of electromagnetic waves," *Journal of Computational Physics*, Vol. 114, No. 2, 185–200, 1994.
23. OpenMaXwell, 2014, http://openmax.ethz.ch/.
24. Alparslan, A. and C. Hafner, "Using layered geometry Green's functions in the multiple multipole program," *Journal of Computational and Theoretical Nanoscience*, Vol. 8, No. 8, 1600–1608, 2011.
25. Alparslan, A. and C. Hafner, "Analysis of photonic structures by the multiple multipole program with complex origin layered geometry Green's functions," *Journal of Computational and Theoretical Nanoscience*, Vol. 9, No. 3, 479–485, 2012.
26. Alparslan, A., "Numerical analysis of photonic nanostructures in layered geometries," Ph.D. Thesis, ETH Zurich, Switzerland, 2013.
27. Wang, M., K. Schmidt, A. Alparslan, and C. V. Hafner, "Hp-fem and PML analysis of plasmonic particles in layered media," *Progress In Electromagnetics Research*, Vol. 142, 523–544, 2013.