



Evolutionary Optimization for Solution of Electromagnetic Problems

Habilitation Thesis

Petr KADLEC

Program:
Electrical, Electronic, Communication
and Control Technology

Brno 2020

This thesis may be consulted for the purposes of research or private study. Any other use is subject to copyright restrictions, in particular with regard to the obligation to explicitly state the source when quoting results from this thesis.

Brno, 10 April 2020

Author

Preface

Always do your best. Your best is going to change from moment to moment; it will be different when you are healthy as opposed to sick. Under any circumstance, simply do your best, and you will avoid self-judgment, self-abuse and regret.

Don Miguel Ruiz

Optimization is understood as a process of finding the best option from a set of possible options. Everybody solves several optimization tasks on a daily basis without knowing it: setting the time of an alarm, planning a public transport route from home to work, packing bought items to a backpack in a store, choosing a proper gear at a right time when driving the car etc.

One can be satisfied with a solution that is offered to us more or less randomly, or one can pursue the optimal one. The search for the optimal solution should not shadow the genuine purpose of the activity: e.g. one can search for the optimal connection so long that he misses the appointment. But if properly used, optimization can make a lot of things easier, save human and natural resources, save enormous amounts of money, or improve or even save lives.

The basis for the research described in this thesis originally stemmed from my passion to evolutionary optimization. For me, it was love at first sight. I remember how amazed I was by the simplicity of evolutionary algorithms on one hand and their power on the other hand. The field of applicability of evolutionary algorithms seems to be limitless. The math hidden by the evolutionary algorithms is really simple - approximately on a high school level. Therefore, optimization methods are attractive for a very large number of users.

I would like to thank to my parents and siblings for giving me all the love. I miss my dad, especially thinking of what knowledge we could share together. I would like to thank to my family: wife Iva, son David and other kids (sorry, I do not know your names right now) for giving the purpose to my life and work.

I have to acknowledge my present and former colleagues namely Martin Štumpf, Vláda Šeděnka, Martin Marek, Míla Čapek, Vlasta Koudelka, Zbyněk Raida, Jarda Rýmus, Peter Kovács, Pavel Hazdra, Viktor Adler, Víték Losenický, Michal Pokorný and many others for kind help, valuable discussions and all the encouragements. Also, I have to praise all my students for giving me the valuable feedback and bringing the new points of view to me.

Contents

Preface	i
Contents	iii
1 Introduction	1
2 Optimization Problem Formulations	3
2.1 Single-objective Optimization	3
2.2 Multi-objective Optimization	3
2.3 Optimization with Variable Number of Dimensions	7
3 State-of-the-art Algorithms	11
3.1 Genetic Algorithms	12
3.2 Particle Swarm Optimization	16
3.3 Differential Evolution	20
4 Single-objective Optimization	23
4.1 Lightning Stroke Localization	24
5 Multi-objective Optimization	37
5.1 Multi-objective self-organizing migrating algorithm	38
5.2 Multi-objective Optimization of EMC Testing Pulse	66
6 Optimization with Variable Number of Dimensions	75
6.1 PSO for VND problems	76
6.2 VND Inverse Scattering Problem	91
6.3 VND PCB Decoupling	97
6.4 VND Antenna Array Design	111
7 Teaching Experiences	121
7.1 Subjects	121
7.2 Student Theses	122
8 Research Projects	125
9 Conclusions	127
Bibliography	129

Chapter 1

Introduction

With a massive increase of personal computer power a majority of design engineering tasks transferred from laboratory to computer aided design (CAD) programs. The CAD programs enable to perform thorough analysis of virtual prototypes in a more comfortable way than it is possible with conventional real prototypes. The advantages of the virtual prototyping are obvious:

- lower initial costs,
- decreasing design cycle times,
- decreasing manpower needed,
- decreasing waste of material resources,
- automation of the design processes.

The last item from the aforementioned list goes hand in hand with a massive progress of optimization methods. Optimization can be viewed as any process of finding and comparing feasible solutions of a specified problem until the best solution is assigned. Optimization methods are nowadays used in almost every field of engineering e.g. economics, civil or mechanical engineering, bio-medicine etc. including the domain of electromagnetics (EM).

With the increasing complexity of engineering devices, the number of design variables i.e. degrees of freedom grows. Accordingly, the number of possible combinations for e.g. parametric analysis of a designed device grows exponentially beyond the limits of nowadays computational resources. Therefore, new effective methods are inevitable for solving these design problems that belong to the family of NP-hard problems [70].

According to “No-free-lunch” theorem [198], there simply does not exist one universal method for solving of all kinds of engineering problems. According to this theorem, various optimization algorithms have to be compared on a large sample of problems so that one can say that one algorithm outperforms the other. Therefore, it is almost impossible to select the best algorithm for a specific problem a priori.

Also, the design procedures take different points of view on the device under consideration. These viewpoints can be e.g. price, performance, stability of parameters, resilience to faults, dimensions etc. Some of these objectives can be contradictory which disables to design a device satisfying all the requirements at the same time. The so-called

multi-objective optimization methods enable to reveal the trade-off among the individual objectives [45].

Moreover, some of the design problems force the designer to select a complexity of the device (e.g. to select a number of transmitters to cover a specific area with a broadcasting signal, to select a number of elements of an antenna array, to select a number of clusters when classifying a set of data points etc.) and then search for the optimal values of corresponding design variables. It is very ineffective to process the search for optimal design variables for all the possible dimensions and then select an optimal solution (in terms of the dimension and values of the design variables). Therefore, these problems require optimization algorithms that are capable to handle variable number of dimensions within a single run.

Organization of the thesis is as follows. Various possible formulations of the optimization problems including single-objective, multi-objective and with a variable number of dimensions are defined in Chapter 2. Chapter 3 briefly reviews the selected state-of-the-art evolutionary optimization algorithms. Chapters 4, 5, and 6 review contributions of the author to the single-objective, multi-objective and a variable number of dimensions optimization research, respectively. These chapters include reprints of the most important peer-reviewed journal and conference papers by the author. Then, Chapter 7 describes teaching experiences of the author and Chapter 8 briefly covers the author's engagement to research projects. Finally, Chapter 9 concludes the thesis.

Chapter 2

Optimization Problem Formulations

2.1 Single-objective Optimization

A general single-objective optimization problem (SOOP) is defined:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x} \in \Omega, \\ & g_j(\mathbf{x}) \leq 0, \\ & j = 1, 2, \dots, J \end{aligned} \tag{2.1}$$

where f is the so-called objective (fitness) function that should be either minimized or maximized. Symbol \mathbf{x} denotes the decision space vector composed of all N design variables x_n (some times called as degrees of freedom, DOF). Usually, the feasible space Ω is a N -dimensional subspace formed by a lower limit $x_{n,\min}$ and upper limit $x_{n,\max}$ for every variable from the decision space vector. Finally, there can be up to J constraint functions that further delimit the feasible space of the optimization problem. An example of one-dimensional ($N = 1$) problem constrained by inequality $x \geq x_k$ is shown in Fig. 2.1. The dashed curve denotes the objective function for the full feasible space Ω , while the solid curve denotes the constrained part of the objective function.

Without loss of generality, we will consider only minimization objective functions in the remainder of the text. The maximization objective function can be easily transferred to a minimization one by multiplying it by -1 . This procedure has no impact on the position (\mathbf{x}_{opt}) of the optimal (minimal or maximal) solution as shown in Fig. 2.1.

The result of the SOOP is one decision space vector \mathbf{x}_{opt} having the minimal (or maximal) value of the objective function $f(\mathbf{x}_{\text{opt}})$. Therefore, this approach is applicable to problems where no conflicting objectives are considered: e.g. solution of the inverse imaging problem.

2.2 Multi-objective Optimization

The next class of optimization problems applies multiple (usually conflicting) objectives. Then, the optimization problem is called as a multi-objective one (MOOP) and is defined

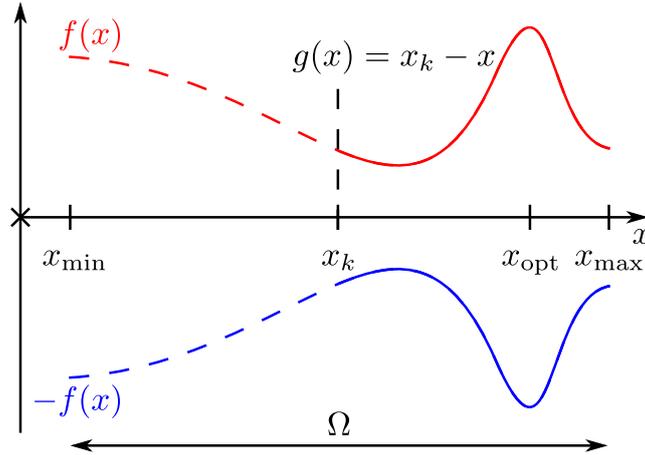


Figure 2.1: Objective function for maximization/minimization constrained one-dimensional optimization problem.

as:

$$\begin{aligned}
 \min_{\mathbf{x}} \quad & \mathbf{f}(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})\} \\
 \text{s.t.} \quad & \mathbf{x} \in \Omega, \\
 & g_m(\mathbf{x}) \leq 0, \\
 & m = 1, 2, \dots, M, \\
 & j = 1, 2, \dots, J
 \end{aligned} \tag{2.2}$$

Here, meaning of all the symbols remains the same as in case of SOOP in (2.1). The only difference is, that up to M objective functions are minimized at the same time. With MOOP, every solution can be viewed in the so-called decision space (using values of the decision space vector \mathbf{x}) or in the so-called objective space (using values of the objective functions \mathbf{f}). A traditional example of the multi-objective problem can be e.g. choice of the transport vehicle from a viewpoint of the traveling time and price. As shown in Fig. 2.2, none of the possibilities is better than any other in both the objectives i.e. the minimal time and price. But knowing the shape of the trade-off among the individual objectives brings an extra information to the designer.

The relation between the decision and the objective space is usually highly non-linear as indicated in Fig. 2.3. The non-linearity is based on the shape of the objective and constraint functions. The solution of the MOOP is the so-called Pareto front. It is depicted in Fig 2.3 with the red curve. It can be viewed as a collection of all trade-off solutions in the objective space. It forms a curve for two-objective problems, a curved plane for three-objective problems etc. There are two goals for the MO algorithms: 1) to find a set of non-dominated solutions that is as close as possible to the true Pareto front, and 2) distribute the set of non-dominated solutions so that the true Pareto front is covered uniformly.

The Pareto front is called as a set of Pareto-optimal solutions when viewed in the decision space. The Pareto front is framed by its extreme solutions that are the best according to individual objectives (please refer to Fig. 2.3 and red circles). Symbol $\mathbf{x}(\mathbf{f}_{1,\text{opt}})$ assigns the best solution according to objective function f_1 . Symbol \mathbf{x}_w denotes the worst possible solution. Finally, \mathbf{x}_u denotes the Utopian (best) solution formed by the best values of individual objective functions. It is highlighted with a blue marker in the objective space.

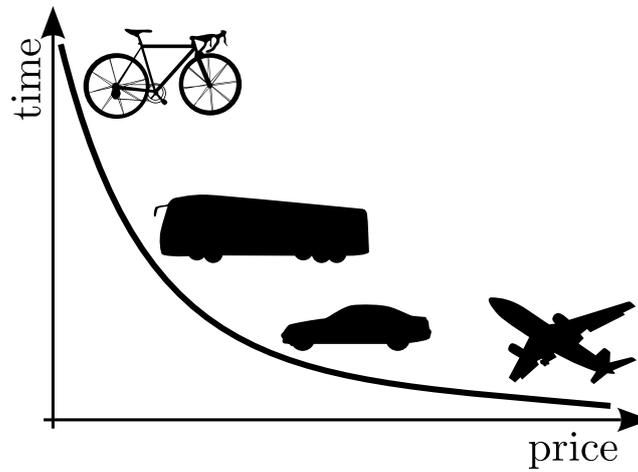


Figure 2.2: Choice of the traveling vehicle as a multi-objective problem: price vs. time [82].

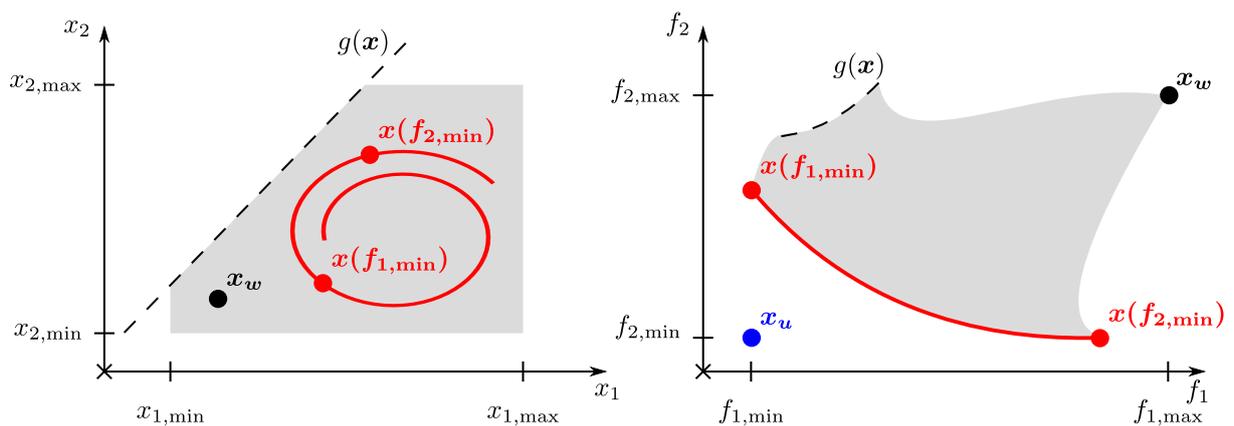


Figure 2.3: Decision space (left) and objective space (right) of the two-objective optimization problem.

Table 2.1: The dominance relations.

Relation	Symbol	Interpretation
strictly dominates	$\mathbf{x}_1 \prec\prec \mathbf{x}_2$	\mathbf{x}_1 is better than \mathbf{x}_2 in all objectives
dominates	$\mathbf{x}_1 \prec \mathbf{x}_2$	\mathbf{x}_1 is not worse than \mathbf{x}_2 in all objectives and better in at least one objective
weakly dominates	$\mathbf{x}_1 \preceq \mathbf{x}_2$	\mathbf{x}_1 is not worse than \mathbf{x}_2 in all objectives
incomparable	$\mathbf{x}_1 \parallel \mathbf{x}_2$	neither $\mathbf{x}_1 \preceq \mathbf{x}_2$ or $\mathbf{x}_2 \preceq \mathbf{x}_1$
indifferent	$\mathbf{x}_1 \mathbf{x}_2$	\mathbf{x}_1 has the same value as \mathbf{x}_2 for all objectives

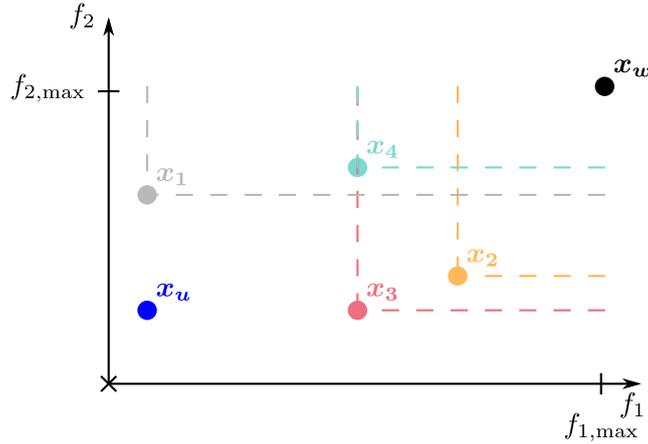


Figure 2.4: Principle of dominance explained on a two-objective optimization problem.

However, it clearly resides outside of the feasible area in the objective space (gray area) and therefore it cannot be found in the decision space and therefore is infeasible.

There is plenty of aggregating techniques that transform a multi-objective (MO) problem to a single-objective (SO) one e.g. Weighted Sum Method, Rotated Weighted Metric Method (see [45]). These methods suffer to deliver high-quality Pareto-front results as shown e.g. in [87]: these methods cannot find solutions on concave parts of the Pareto front and fail to distribute the solutions uniformly along the whole Pareto front. Therefore, pure multi-objective solvers are necessary.

Most of the pure multi-objective optimization algorithms utilize the principle of dominance [107]. Two solutions \mathbf{x}_1 and \mathbf{x}_2 can be compared according to dominance. The possible results of the comparison are summarized in Table 2.1. Mostly, multi-objective algorithms use the simple or weak dominance relation to compare possible solutions.

According to the definition of dominance relation in Table 2.1, every solution dominates a part of the objective space according to values of its objective functions. This phenomenon is shown in Fig. 2.4. There, four solutions are displayed with different colors. Dashed lines mark the subspace that is dominated by the corresponding solution. Based on positions of individual solutions, we can state that solution \mathbf{x}_1 strictly dominates \mathbf{x}_4 . Further, solution \mathbf{x}_3 dominates \mathbf{x}_4 . Solutions \mathbf{x}_1 and \mathbf{x}_3 are mutually non-dominated (incomparable). The same holds for the pair \mathbf{x}_2 and \mathbf{x}_4 . It is obvious, that all the feasible solutions dominate the worst solution \mathbf{x}_w but are dominated by the Utopian solution \mathbf{x}_u .

When working with a set of solutions $\mathcal{P} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_P\}$, it is necessary to select the mutually non-dominated solutions that are the candidates for the Pareto front. The non-dominated sorting algorithm is therefore the corner stone of every multi-objective

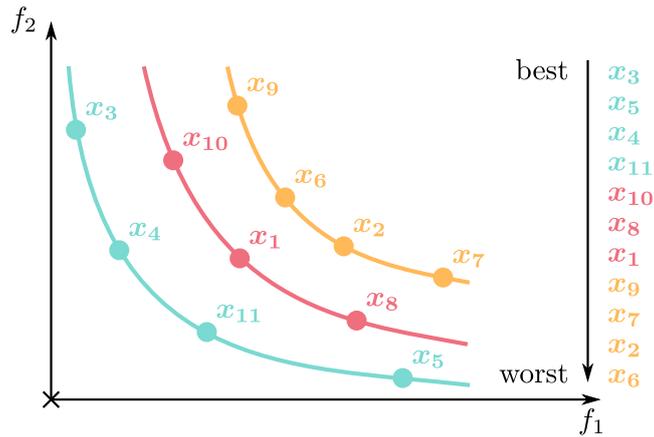


Figure 2.5: Non-dominated sorting: different orders of non-dominated sets.

algorithm. All the solutions can be sorted to non-dominated sets of consecutive orders as shown in Fig. 2.5. There, solutions with the same color build the set of specific order. Every solution from the non-dominated set of the second order is dominated by at least one solution from the set of the first order and so on. Within the set of one order, solutions are mutually non-dominated. However, extreme solutions according to individual objective functions and then less crowded solutions are preferred to maintain diversity of candidate solutions.

The effectiveness of the non-dominated sorting is crucial factor for the speed of MO algorithms. Therefore, lot of efforts has been put into improvement of non-dominated sorting algorithms starting from naive and slow ones [171] and [46] with computational complexity $O(MN^3)$ and $O(MN^2)$, respectively. The efficiency can be improved to $O(M\bar{N}N)$ (where \bar{N} is the number of non-dominated solutions in the population of N solutions) using the so-called Arena's principle introduced in [181]. Next, authors in [129] improve the efficiency of the sorting to $O(MN\bar{N}^{1/2})$. This complexity can be further improved with the help of the hierarchical sorting in [17].

2.3 Optimization with Variable Number of Dimensions

In general, optimization tasks require to set a priori the complexity of a model approximating a designed device. It means, that a fixed number of variables is searched by the optimization algorithm. Plenty of design problems start with a question: "How many of ... is necessary to ...?" These tasks need a precise estimation of number of variables of the model to be done a priori by some experienced designer. Another option is to formulate

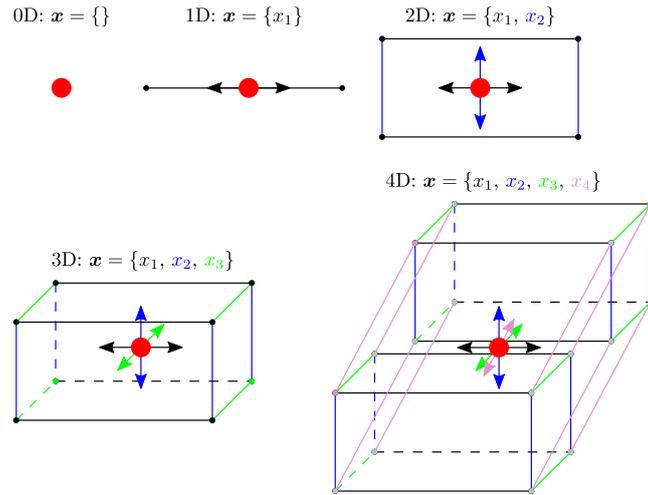


Figure 2.6: Solutions with different dimensions for VND optimization.

the task as a problem with a variable number of dimensions (VND):

$$\begin{aligned}
 \min_{\mathbf{x}} \quad & \mathbf{f}(\mathbf{x}, N) = \{f_1(\mathbf{x}, N), f_2(\mathbf{x}, N), \dots, f_M(\mathbf{x}, N)\} \\
 \text{s.t.} \quad & \mathbf{x} \in \Omega, \\
 & g_m(\mathbf{x}) \leq 0, \\
 & m = 1, 2, \dots, M, \\
 & j = 1, 2, \dots, J, \\
 & N = N_1, N_2, \dots, N_D
 \end{aligned} \tag{2.3}$$

The VND task formulation is the same as for MO task (2.2), except the fact, that some of or all the objective functions depend on the decision space vector \mathbf{x} and on its size N at the same time. The dimension of the decision space vector N can be selected from the list of feasible dimensions $\{N_1, N_2, \dots, N_D\}$ where D is the number of feasible dimensions for the modeled device. It should be noted, that feasible dimensions are usually multiples of some number specifying how many items with the specified number of design variables are used. Therefore, the list of feasible dimensions does not contain consecutive integer values, usually.

There is a large number of representative examples of VND problems. Designing some broadcasting network covering a specified area is clearly a VND problem because the optimal number of transmitters (and their position and power) is not known a priori [48]. The similar behavior is evident for a composite laminate stacking problem [114], a short cantilever design [101], the high-dimensional data clustering [185], the cancer marker identification [135] etc.

The optimization with variable number of dimensions allows to share information between solutions \mathbf{x} that have different number of dimensions as indicated in Fig. 2.6. The main idea of VND optimization is that the information about good values of individual decision space variables can be shared across different dimensions.

Consider the broadcasting network design problem. Every transmitter introduces three new design variables (two for a position on a map and one for a transmitting power). The objectives of the design is to cover a specified area without any overlaps between

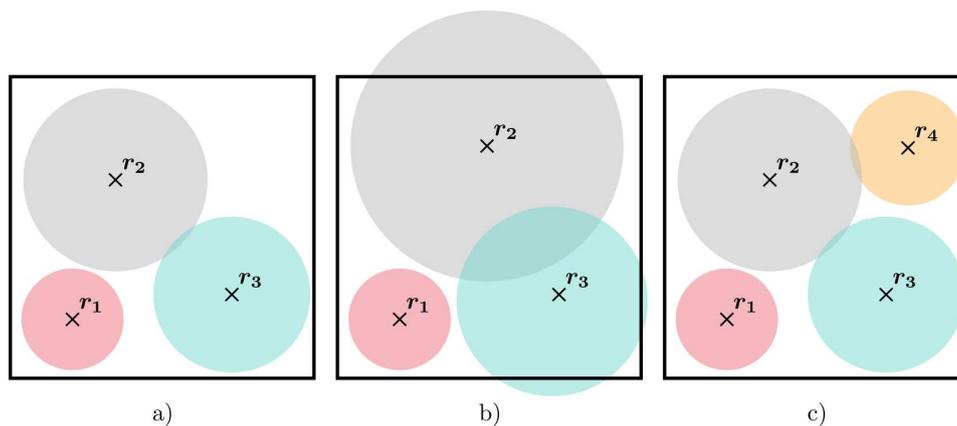


Figure 2.7: The broadcasting network design example: a) initial $N = 9$ solution, b) “improved” $N = 9$ solution (no VND), and c) $N = 12$ solution (using VND).

mutual transmitters and outside of the area that should be covered. Let’s assume that we work with a “good” candidate solution, that utilizes 3 transmitters (i.e. $N = 9$ design variables) as shown in Fig. 2.7 a). Now, if we try to cover the whole area with use of three transmitters, we have to adjust the positions and power of individual transmitters a lot. However, a better coverage with lower overlaps and without a large adjustment of variables corresponding to the transmitters from the “good” candidate solution can be achieved by adding a fourth transmitter (see Fig. 2.7 b) and c)).

Chapter 3

State-of-the-art Algorithms

Conventional optimization methods like Steepest Descent method, Newton method, Conjugate Gradient method etc. [169] are based on the knowledge of gradients of objective functions. These can be replaced by the finite differences when needed. Nevertheless, the knowledge of gradients is not possible in case of problems with objective functions determined by results of CAD-based analysis tools. Moreover, the conventional gradient-based methods rely on a qualified initial estimation. If the initial guess is far from the global minimum or simply at the bad position not so far (in the domain of the other attractor), then these methods tend to end in a local minimum.

The main factors for a massive usage of evolutionary optimization algorithms (EA) are:

- no need of gradients,
- no need of initial guess,
- robustness,
- simplicity of use and modification.

It is fair to mention the drawbacks of the evolutionary optimization methods in comparison to the conventional gradient-based methods that are mainly:

- slower convergence rate,
- higher computational demands,
- randomness of the process.

A pseudocode for a general evolutionary algorithm (regardless if single-/multi-objective or even VND) is shown in Algorithm 1. It accepts inputs defining the problem to be solved i.e. set of objective and constraint functions \mathbf{f} and \mathbf{g} , decision space limits Ω . Next, a set of algorithm parameters \mathcal{S} is defined. In general, \mathcal{S} contains P as a number of agents (solutions) and I as a number of iterations to be proceeded. The remaining elements of \mathcal{S} depend on the type of the algorithm.

The general EA starts with a random or pseudo-random generation of initial positions for all P agents. Then, every agent is “awarded” by objective functions \mathbf{f} . Then, positions of all agents are updated iteratively until any of the stop conditions is fulfilled. Usually,

Algorithm 1: Pseudocode of a general evolutionary optimization algorithm.

```

1 function generalEA ( $\mathcal{S}, \mathbf{f}, \mathbf{g}, \Omega$ );
   Input : Settings parameters  $\mathcal{S}$ , objective functions  $\mathbf{f}$ , constraint functions  $\mathbf{g}$ ,
           decision space limits  $\Omega$ 
   Output: Set  $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_P\} : \forall \mathbf{x}_p \in \Omega$ 
2 Generate random vectors  $\mathbf{x} \in \Omega$ 
3 Compute  $\mathbf{f}(\mathbf{x})$ 
4 while Stop condition not met do
5   | Update  $\mathbf{x}$ 
6   | Check new  $\mathbf{x} \in \Omega$ 
7   | Compute new  $\mathbf{f}(\mathbf{x})$ 
8   |  $i = i + 1$ 
9 end

```

maximal number of iterations I is one of the stop conditions. Other commonly used stop condition is that the algorithm finds a solution reaching specific values of the objective functions.

We have decided to describe here three state-of-the-art algorithms, that have also variants for multi-objective problems and problems with a variable number of dimensions. Namely we describe Genetic Algorithms [78], Particle Swarm Optimization [100] and Differential Evolution [172].

It should be noted here, that the canonical versions of the algorithms are only briefly introduced in this thesis. There exists an enormous number of modifications of the selected algorithms, their problem-oriented modifications and hybridized versions. We kindly refer the reader to the referenced papers.

3.1 Genetic Algorithms

Genetic algorithms (GA) have been introduced by Holland and Goldberg in [60]. It is based on the idea that only high-quality and beneficial genes of some species survive to next generations i.e. the basic idea of the evolutionary theory published by Charles Darwin in [41].

Single-objective Genetic Algorithm

Description of a conventional single-objective GA (SOGA) can be found in [78]. In GA, every variable x_n is coded in form of a binary number of selected length b_n . It means that GA works with a discrete decision space whose density increases with the increase of b_n values. All binary values of a single solution \mathbf{x} build together the so-called chromosome. A set of P chromosomes is one generation denoted as \mathcal{X} .

A pseudocode for a generalized GA is summarized in Algorithm 2. It starts with a random generation of P individuals that are awarded with objective functions values. The SOGA then proceeds with a iteration loop, where the parent population \mathcal{X}_p is first selected using the tournament selection or roulette selection. Both the selection methods are illustrated in Fig. 3.1. With the tournament selection, random pairs of agents are

selected and then compared according to the objective function value. The agent with a better value of the objective function proceeds to the mating pool. The roulette selection works so that a probability value ξ is given to a p -th solution proportional to its quality of objective function. The probability for the maximizing objective function f is

$$\xi_p = \frac{f_p}{\sum_{p=1}^P f_p}, \quad (3.1)$$

while for the minimization objective function f :

$$s_p = \frac{\sum_{p=1}^P f_p}{f_p},$$

$$\xi_p = \frac{s_p}{\sum_{p=1}^P s_p}. \quad (3.2)$$

Then, a random number from interval $\langle 0, 1 \rangle$ is generated P -times to assign solutions that belong to the mating pool. Both the selection methods may leave out the best solution from the mating pool. Therefore, the so-called elitism strategy is applied and selected number of solutions with the best value of objective function are automatically taken to the mating pool.

Algorithm 2: Pseudocode of a generalized genetic algorithm.

```

1 function GA ( $\mathcal{S}, \mathbf{f}, \mathbf{g}, \Omega$ );
   Input : Settings parameters  $\mathcal{S}$ , objective functions  $\mathbf{f}$ , constraint functions  $\mathbf{g}$ ,
           decision space limits  $\Omega$ 
   Output: Set  $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} : \forall \mathbf{x}_n \in \Omega$ 
2 Generate random vectors  $\mathbf{x} \in \Omega$ 
3 Compute  $\mathbf{f}(\mathbf{x})$ 
4 while Stop condition not met do
5     | Select the mating pool  $\mathcal{X}_m$  from parents  $\mathcal{X}_i$ 
6     | Create set of offsprings  $\mathcal{X}_o$ 
7     | Mutate individuals
8     | Evaluate fitness for  $\mathcal{X}_o$ 
9     | Pick individuals for new generation
10    |  $i = i + 1$ 
11 end

```

After the mating pool is selected, the recombination operations are applied to create a set of offspring solutions \mathcal{X}_o . The conventional recombination operations are the crossover and the mutation operators. Figure 3.2 illustrates these two operations. The crossover is controlled by the probability of crossover parameter. If set to 0, no crossover is performed and the offspring population is the exact copy of the parent population. If set to 1, then all the offspring solutions are generated with a help of the crossover. During the crossover, random pairs are selected from the mating pool, these solutions are cut on a randomly selected position and two offspring solutions are generated so that both the offspring solutions have a corresponding part from both the parents. The mutation is applied to

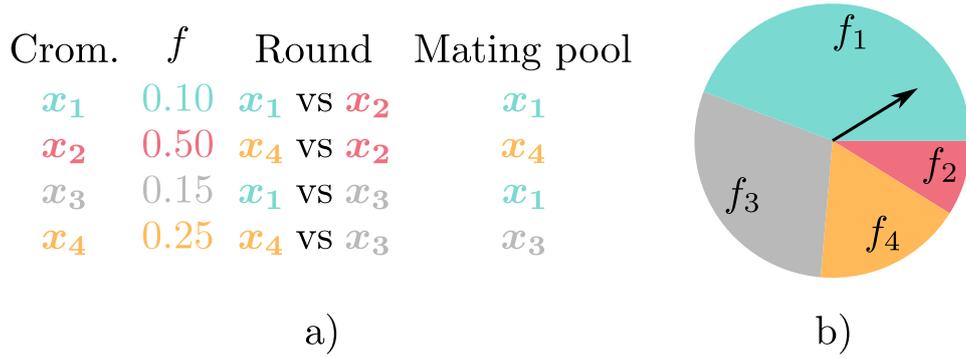


Figure 3.1: Tournament and roulette wheel selection for parent generation selection of GA.

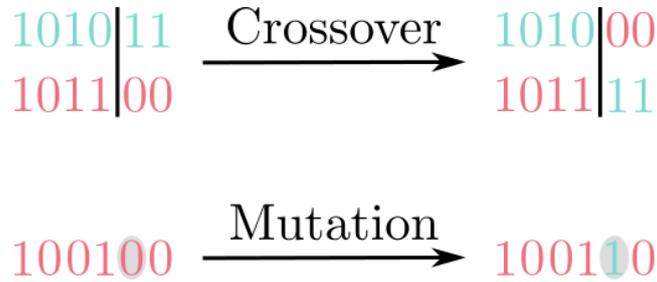


Figure 3.2: Crossover (top) and mutation (bottom) operators for the genetic algorithm.

all the offspring solutions with a priori selected probability of mutation. When applied, a randomly selected bit from the offspring chromosome is reverted.

After the recombination stage is completed, the parent set \mathcal{X}_i and offspring set \mathcal{X}_o are combined and the best solutions build the parent set \mathcal{X}_{i+1} for the next iteration. The crossover should force the algorithm to search for the solutions with the better parts from the parents chromosomes, thus, enhancing the convergence rate of the algorithm. On the other hand, the mutation operator prevents the algorithm to get stuck in the local optimum. The user of GA can balance the exploration/exploitation ability of the algorithm by selecting the mutation and crossover probability.

Multi-objective Genetic Algorithm

The most famous and probably the most used multi-objective optimizer is the Non-dominated Sorted Genetic Algorithm (NSGA-II) introduced in [46]. It follows all the strategies as introduced for the SOGA i.e. it follows the pseudocode as in Algorithm 2. The only difference is the strategy to pick individuals for new generation \mathcal{X}_{i+1} . The selection is made by a two-stage sorting as illustrated in Fig. 3.3.

First, the non-dominated fronts of consecutive orders $\mathcal{F}_1, \mathcal{F}_2, \dots$ are revealed until the number of agents in the advanced fronts is greater or equal than number of agents P . Then, the last advanced front has to be further sorted based on the so-called crowding distance metric [46]. It is applied on the advanced front that cannot be included to the parent set for the next generation \mathcal{X}_{i+1} because the maximal number of solutions would be exceeded. The crowding distance for sorting the solutions within a one non-dominated set is computed according to [46]:

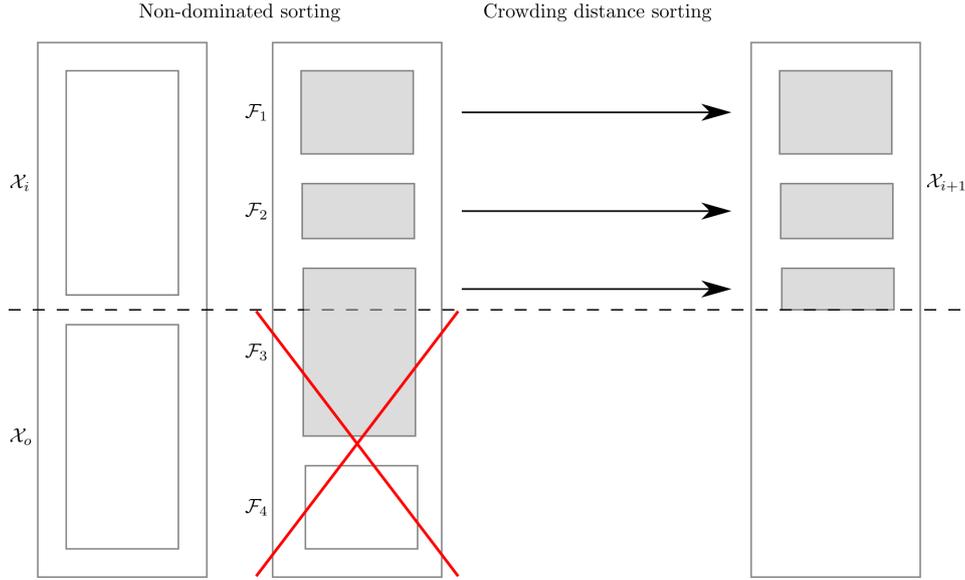


Figure 3.3: Two stage sorting of combined parent and offspring set for the NSGA-II [46].

$$d_{I_j^m} = d_{I_j^m} + \frac{f_m^{I_{j+1}^m} - f_m^{I_{j-1}^m}}{f_{m,\max} - f_{m,\min}}, \quad (3.3)$$

where d denotes the crowding distance, and I_j^m denotes the solution index of the j -th member of the set \mathcal{F} sorted in ascending order according to values of m -th objective function. Value $d = 0$ is assigned to all solutions from set \mathcal{F} at the beginning of the procedure. The extreme solutions with indexes I_1^m and $I_{|\mathcal{F}|}^m$ are automatically assigned with a large value, so that they are always preferred by the crowding distance metric. The crowding distance selection (see Fig. 3.3) takes first the solutions with a larger value of the crowding distance.

Settings for the NSGA-II algorithm remains the same as for the SOGA (see Section 3.1). The trade-off between exploration and exploitation of the decision space of the algorithm can be set via parameters probability of mutation and probability of crossover, respectively. The most important parameter for the convergence rate is the binary precision for individual variables. Further, the speedup of the convergence rate can be achieved by a proper choice of mating pool selection (e.g. tournament or roulette wheel selection) and by the multi-point crossover [112]. The success-rate of these techniques is usually very problem-dependent.

Genetic Algorithm with Variable Number of Dimensions

Many authors in the domain of optimization with a variable number of dimensions name the modified VND algorithms as Variable Length Genom Algorithms (VLGA). Genetic algorithms have been used very often to solve VND problems. Nevertheless, most of the authors add a so-called header to the decision space vector. This header has length N (the number of decision space variables). Individual entries of the header are equal to zero - variable is in not used, or one - variable is used.

Authors in [47] introduce the so-called static (these variables are mandatory for the

decisions space vector) and dynamic part (these variables can be included/excluded based on the complexity of the model) of the chromosome. The crossover operator is applicable only in certain positions (according to the binary precisions of individual design variables). Authors in [170] and [113] introduce an integer or real-valued additional string to the chromosome, respectively. This string then specifies the affiliation of genes with decision space variables.

Ting et al. [184] divides the genome with variable length into the sub-strings according to model parameters. Then, crossover positions are available on the level of whole variables and on the level of individual bits of variables. Although most of the authors of VLGA handle the variable number of dimensions by modifying the crossover operator, authors in [59] change the mutation operator. There are four mutation operators that can change the length of the chromosome: growth, shrink, swap, and replace mutation.

Major part of the VLGAs are devoted to single-objective optimization applications. From the aforementioned works, only [184] solves a multi-objective task (namely the transmitter placement problem). Another multi-objective applications are e.g. [161] where the multi-dimensional clustering problem is solved and [58] where sorting of unsigned permutations by reversals is performed.

3.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is another example of evolutionary algorithm [142]. It was originally introduced by Kennedy and Eberhart in 1995 [100]. It mimics a behavior of swarm of individuals (usually bees) that searches for the best feeding position. The individual agents trace information about places that were visited and share the information about quality of the visited places within the whole swarm.

Single-objective PSO

A comprehensive review of a single-objective PSO algorithm (SOPSO) can be found in [163]. The generalized SOPSO pseudocode is presented in Algorithm 3. As other evolutionary algorithms, it starts with a random generation of positions for individual particles. Then, every particle has its own velocity vector, that is crucial for the update of the position. The velocity is generated randomly within a velocity subspace Π . Usually, a portion (1/3 or 1/10) of the search domain is used for the generation of the initial random velocity e.g.:

$$v_{n,\max} - v_{n,\min} = 1/3(x_{n,\max} - x_{n,\min}). \quad (3.4)$$

In every iteration i , the particles change their respective position \mathbf{x}_p according to:

$$\mathbf{x}_p(i) = \mathbf{x}_p(i-1) + \Delta t \mathbf{v}_p(i) \quad (3.5)$$

where δt is the time step ($\delta t = 1$ for most of the cases), p denotes the index of particle from the swarm of size P and i stands for the iteration number. The velocity is then updated according to:

$$\mathbf{v}_p(i) = w \mathbf{v}_p(i-1) + c_1 r_1 [\mathbf{pb}_p - \mathbf{x}_p(i)] + c_2 r_2 [\mathbf{gb} - \mathbf{x}_p(i)] \quad (3.6)$$

Here, w is the so-called inertia weight chosen by user from interval $\langle 0, 1 \rangle$ (or sometimes linearly decreasing with increasing iteration i from $w_{\max} = 0.9$ to $w_{\min} = 0.4$ [16]). Then,

Algorithm 3: Pseudocode of a generalized particle swarm optimization algorithm.

```

1 function PSO ( $\mathcal{S}, \mathbf{f}, \mathbf{g}, \Omega$ );
   Input : Settings parameters  $\mathcal{S}$ , objective functions  $\mathbf{f}$ , constraint functions  $\mathbf{g}$ ,
           decision space limits  $\Omega$ 
   Output: Set  $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} : \forall \mathbf{x}_n \in \Omega$ 
2 Generate random agents  $\mathbf{x} \in \Omega$ 
3 Generate random velocities  $\mathbf{v} \in \Pi$ 
4 Compute  $\mathbf{f}(\mathbf{x})$ 
5 while Stop condition not met do
6   | Update velocities  $\mathbf{v}_i$ 
7   | Update positions  $\mathbf{x}_i$ 
8   | Apply boundary conditions for  $\mathbf{x}_i \notin \Omega$ 
9   | Evaluate fitness for  $\mathbf{x}_i$ 
10  | Update personal and global best
11  |  $i = i + 1$ 
12 end

```

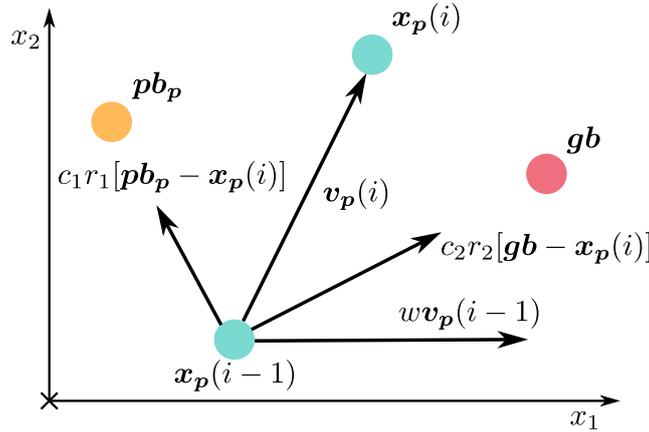


Figure 3.4: Velocity update for the PSO algorithm [10].

c_1 and c_2 stands for a cognitive and social learning factor, respectively. They are both usually set to $c_1 = c_2 = 1.5$. Random numbers r_1 and r_2 are generated from a uniform interval $\langle 0, 1 \rangle$ for every single velocity update. Vector \mathbf{pb}_p stands for the personal best of the p -th particle (position of the p -th particle with the best value of the objective function). Finally, \mathbf{gb} denotes the global best position - the position with the best value of the objective function visited by any of the particles from the whole swarm.

As shown in Fig. 3.4, the particle is driven by the velocity update formula to partly remain in the totally random initial direction, and partly, it is forced to research the area of personal and global best more in detail. However, the corresponding \mathbf{pb} and \mathbf{gb} components of the velocity vector are multiplied by random number to preserve the algorithm to evaluate very similar solutions. Users can control the exploration and exploitation of the algorithm by setting the values for the inertia weight, and learning factors. While large values of w and c_1 favor the exploration, a large value of c_2 forces the algorithm to exploit the area of the current best solution \mathbf{gb} .

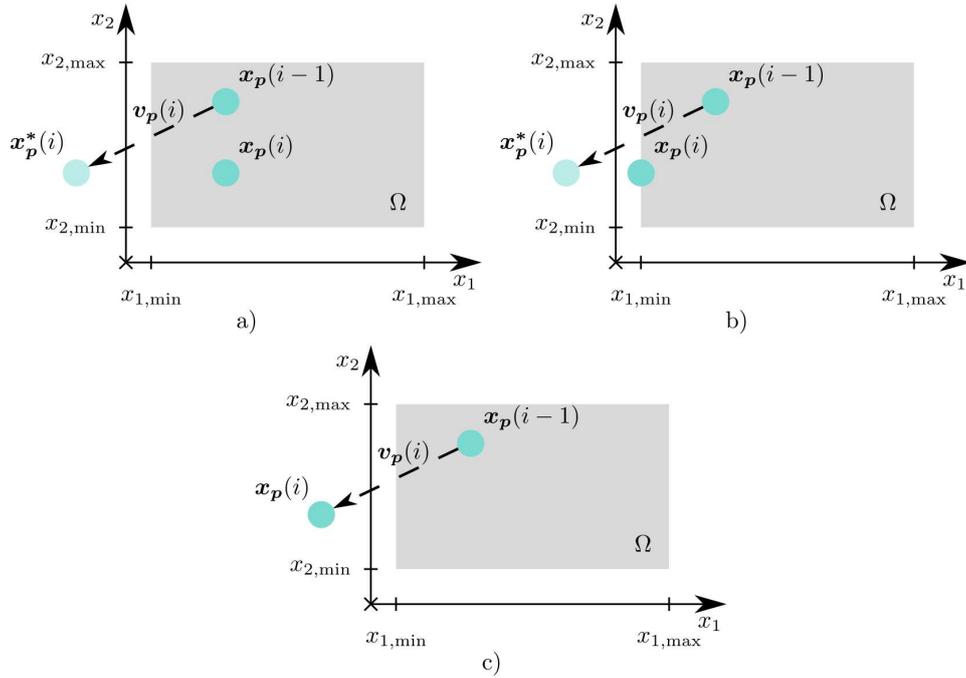


Figure 3.5: Boundary conditions for the PSO algorithm: a) reflecting, b) absorbing, and c) invisible.

After the position update (3.5) particles can reside outside of the feasible decision space Ω . Then, one of the following boundary conditions has to be applied:

- reflecting,
- absorbing,
- invisible.

The boundary conditions are illustrated by Fig. 3.5. The reflecting condition reflects the solution from the updated position $\mathbf{x}_p^*(i)$ according to (3.5) back to the feasible space Ω according to violated boundary ($x_{1,\min}$ in case of Fig. 3.5). In case of the absorbing boundary, the new position of the particle $\mathbf{x}_p(i)$ is fixed to reside on the violated boundary. The invisible boundary condition does not change the position of the particle that is outside the feasible space. Instead, the objective function is worsened significantly (e.g. $f(\mathbf{x}_p(i)) = \infty$). Then, the particle is pulled back into Ω in the next iterations with the velocity update formula (3.6).

Multi-objective PSO

There is a large number of PSO modifications for multi-objective problems. Nice overview of these modifications can be found in [159]. Multi-objective Particle Swarm Optimization (MOPSO) algorithm follows exactly the pseudocode of the simple SOPSO algorithm (see Algorithm 3). The only change is in the velocity update step (3.6) that changes to:

$$\mathbf{v}_p(i) = w\mathbf{v}_p(i-1) + c_1r_1[\mathbf{pb}_p - \mathbf{x}_p(i)] + c_2r_2[\mathbf{gb}_p - \mathbf{x}_p(i)] \quad (3.7)$$

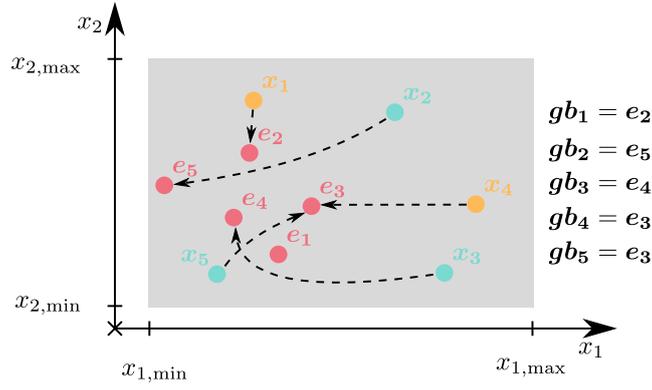


Figure 3.6: Selection of the global best solutions gb from the external archive members e .

Here, a different global best position gb_p is selected for every p -th particle. The global best candidates are saved in the so-called external archive \mathcal{E} . The external archive is updated at the end of every iteration i . It has to be pruned according to crowding distance in case too many non-dominated solutions are found in the combined parent/offspring sets (see Section 11).

There exist two ways how to assign the global best from the external archive:

- based on the Euclidean distance,
- randomly.

With the selection based on the Euclidean distance (in the decision space), the algorithm tends to premature convergence (in case of the multi-objective optimization to non-dominated sets of orders higher than 1) i.e. the local optimums are researched by a large portion of particles. On the other hand, random selection slows the convergence rate significantly. Therefore, the combination of both the approaches is usually the best option. This approach is illustrated in Fig. 3.6. There, global best is selected randomly for three solutions (x_2 , x_3 , and x_5) and the global best is selected based on the minimal Euclidean distance from the external archive members $e_1 - e_5$ for two of the solutions (x_1 and x_4). As the process of gb selection is partly random, some external archive members can be selected for more solutions (e_3 is selected for both x_3 and x_4) while some external archive members are not assigned at all (e.g. e_1).

Users of MOPSO can set the ratio between randomly selected and based on the Euclidean distance with the settings parameter $r_{gb} \in \langle 0, 1 \rangle$. If $r_{gb} = 0$, all global best values are selected based on the Euclidean distance, if $r_{gb} = 1$ all the global best values are selected randomly. According to our experience, it is better to set larger values $r_{gb} > 0.75$ to the ratio r_{gb} to ensure robustness of the MOPSO algorithm.

PSO algorithms with Variable Number of Dimensions

Surprisingly low number of authors presented works that employ PSO algorithm and its modifications to problems with a variable number of dimensions. Although authors in [74] does not solve a VND problem, an important contribution to the VND algorithms can be found there. Additional term to the velocity update formula is introduced. It is called

as “*ebest*” and is composed of random parts from previous global best positions, that are saved by the modified PSO algorithm. It brings the idea that global optimum can be composed from pieces of individual local optimums.

In [115, 143], modifications to conventional PSO algorithm are made so that VND problems can be solved. The problem with different sizes of the particle \mathbf{x}_p and the global best solution \mathbf{gb} is solved so that a random size between sizes of \mathbf{x}_p and \mathbf{gb} is selected. Multi-Dimensional PSO has been introduced in [103]. The dimension of every particle \mathbf{x}_p is driven by a separate single-variable PSO algorithm. Therefore, the variable is switched on/off by the auxiliary PSO algorithms. Dimension Adaptive PSO has been introduced in [202]. Dimensions of individual particles is controlled by a discrete header that is controlled by a discrete variable PSO algorithm. The size of the particle is selected based on the so-called chord-length parametrization in [37]. Further, authors in [201] introduce the Improved Variable-Length PSO algorithm to select the number and optimal values of neurons in a hidden of an artificial neural network. The velocity update formula is changed to preserve the same sizes for \mathbf{x}_p , \mathbf{pb}_p , and \mathbf{gb} components. The architecture of a convolutional neural network is selected by an improved PSO algorithm in [192]. In [185], authors fill in the missing components for components of the velocity update formula are taken randomly from other solutions with a higher dimension.

3.3 Differential Evolution

Differential evolution (DE) is an evolutionary optimization algorithm introduced by Storn and Price in [172]. The single-objective version of DE and modifications made to DE to solve multi-objective problems and VND problems are briefly introduced in the following sub-sections.

Single-objective Differential Evolution

There is plenty of variants of the DE algorithm for single-objective problems [39]. A variant assigned as DE/rand/1/bin is described here more in detail. Two basic operations are used to modify the solutions: a mutation and a crossover. Its pseudocode of DE is summarized in Algorithm 4.

The mutation operation creates a new random vectors according to:

$$v_{p,n}(i+1) = x_{r_1,n}(i) + F[x_{r_2,n}(i) - x_{r_3,n}(i)] \quad (3.8)$$

where p is the index of the agent, n is the index of the variable, r_1 , r_2 , and r_3 are three mutually different indexes of agents that are also different from p . Further, i denotes the number of iteration and F is the user-defined scaling factor ($F = 1.5$, usually).

The new trial vector $\mathbf{u}_p(i)$ is then generated by the crossover operation between the previous position of the p -th agent $\mathbf{x}_p(i)$ and from the new “mutated” vector $\mathbf{v}_p(i)$:

$$u_{p,n}(i+1) = \begin{cases} v_{p,n}(i+1) & \text{if } randb(n) \leq CR \quad | \quad n == rnbr(p) \\ x_{p,n}(i) & \text{if } randb(n) > CR \quad \& \quad n \neq rnbr(p) \end{cases} \quad (3.9)$$

Here, $randb(n)$ denotes the random number from interval $\langle 0, 1 \rangle$. CR stands for the user-defined crossover rate ($CR = 0.1$ usually). Next, $rnbr(p)$ is the index of component of

Algorithm 4: Pseudocode of a generalized differential evolution algorithm.

```

1 function DE ( $\mathcal{S}, \mathbf{f}, \mathbf{g}, \Omega$ );
   Input : Settings parameters  $\mathcal{S}$ , objective functions  $\mathbf{f}$ , constraint functions  $\mathbf{g}$ ,
           decision space limits  $\Omega$ 
   Output: Set  $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} : \forall \mathbf{x}_n \in \Omega$ 
2 Generate random agents  $\mathbf{x} \in \Omega$ 
3 Compute  $\mathbf{f}(\mathbf{x})$ 
4 while Stop condition not met do
5     Perform mutation for every agent  $\mathbf{x}$ 
6     Create trial vectors  $\mathbf{u}$  using crossover
7     Evaluate fitness for trial vectors  $\mathbf{u}$ 
8     Update positions of agents  $\mathbf{x}$ 
9      $i = i + 1$ 
10 end

```

the p -th agent, that is taken from newly mutated vector $\mathbf{v}_p(i+1)$ automatically. This maintains the diversity among agents because at least one component of the trial vector $\mathbf{u}_p(i+1)$ is new (taken from the randomly created vector).

After the trial vector is created for every agent, the objective function is computed for them. The position of p -th agent is updated as follows:

$$\mathbf{x}_p(i+1) = \begin{cases} \mathbf{u}_p(i+1) & \text{if } f(\mathbf{u}_p(i+1)) < f(\mathbf{x}_p(i)) \\ \mathbf{x}_p(i) & \text{else} \end{cases} \quad (3.10)$$

In other words, the new position of $\mathbf{x}_p(i+1)$ is updated to the trial vector $\mathbf{u}_p(i+1)$ if the objective function value is better than at previous position $\mathbf{x}_p(i)$.

The main advantage of the DE algorithm is that it keeps the diversity among individual agents very well. That influences the robustness of the algorithm positively. Other advantage is that only two user-defined parameters are required to be set by users. A better exploitation can be achieved with lower values of the scaling factor F while exploration is preferred when larger values of the crossover rate are used.

Multi-objective Differential Evolution

The most used multi-objective variant of the differential evolution is the so-called Generalized Differential Evolution (GDE3) introduced in [109]. This method extends the conventional SODE algorithm. A modification is needed for the selection of new position of the agents - see (3.10). If the new trial solution $\mathbf{u}_p(i+1)$ and $\mathbf{x}_p(i)$ are non-dominated then both the solutions are accepted for the next iteration.

The overall number of agents cannot grow uncontrollably. If the population exceeds the number of agents P at the end of iteration i , then the set of non-dominated solutions have to be pruned. It is accomplished with use of the crowding distance metric as described in NSGA-II algorithm (see Section 11). As described in the GDE3 paper [109], the selection rules are more complex in case of the multi-objective constrained problem.

Differential Evolution with Variable Number of Dimensions

Only three papers implementing the DE algorithm for problems with a variable number of dimensions can be found in open literature up to date (03/2020). In [204], author presents the so-called Variable-length DE algorithm to solve the automatic circuit design problem. The circuit consists of a fixed part, where only component values are searched and a non-fixed part, where the number of components, their connection and values are a subject of search. The agents with a mutually different sizes for the mutation operation according to (3.8) are either truncated (if they are too long) or filled with zeros (if they are too short). If the crossover is applied to two vectors with different sizes then they are truncated to the size of the smaller one, which favors the agents with a lower number of design variables.

Authors in [168] implement the so-called Variable Length Crossover Differential Evolution to automatically design the fuzzy neural network architecture. However, the crossover operation modification is valid only for the problem under interest. That makes it impossible to use the algorithm for other problems. Finally in [128], modified DE algorithm is used for the image classification. Nevertheless, the algorithm uses a header called as “masker” that contains ones and zeros that determines if some variable is used or not.

Chapter 4

Single-objective Optimization

The single-objective optimization is a field of interest for the optimization research for more than 50 years. There exists a large number of elaborate algorithms, that are dedicated to specific problems. The main contribution of the author is in the application of some algorithms to solve EM problems. In [27], we have applied the genetic algorithm to search for the optimal feeder position of the patch antenna.

A conference paper [86] applies the DE algorithm to find a position of a lightning stroke source based on the voltage measured on the over-head transmission line above a perfect electric ground. This method was further extended in [83] (see the reprint reprinted version in Section 4.1) so that it accounts also the finite losses of the ground under the transmission line. Also, this paper [83] shows that CMA-ES algorithm outperforms GA, PSO and DE in the efficiency of the lightning stroke localization.

We have used the PSO algorithm to search for the optimal position and value of the decoupling capacitor on a printed circuit board in [94] to suppress the unwanted signals between ports on the board. This approach was held fully in time-domain.

Another important contribution of the author is the Fast Optimization ProcedureS (FOPS) software tool available at <http://http://antennatoolbox.com/fops>. Author of the thesis is the initiator and principal author of the tool, not the main programmer. This tool is written in MATLAB and enables to solve single-objective, multi-objective and VND optimization problems. User can define an own problem or the tool can be used to compare various methods on a gallery of testing benchmark problems. The full documentation to the tool can be found in [127]. The main features of the FOPS tool are described in the journal paper [126].

4.1 Lightning Stroke Localization – A Time-Domain Approach Based on Evolutionary Optimization

Originally published as:

Kadlec, P., Marek, M., Štumpf, M.: Lightning Stroke Localization A Time-Domain Approach Based on Evolutionary Optimization. In *IEEE Transactions on Electromagnetic Compatibility*, 2020.

Abstract

In the present paper we analyze a localization problem based on evolutionary optimization algorithms and an analytical EM-field-to-line coupling model. A computationally efficient, analytical model for calculating time-domain lightning-induced voltages serves as the forward solver in the optimization process. The model takes into account electric permittivity and conductivity of a lossy ground. The inverse problem is solved with the aid of a single-objective global optimization algorithm. It is demonstrated that the Covariant Matrix Adaptation - Evolution Strategy algorithm shows the best performance among considered state-of-the-art algorithms. The influence of other problem parameters, e.g. the size of the search space, the fluctuation of the return-stroke pulse width and amplitude, and the presence of noise, on the localization error is discussed. While the growing size of the search domain has a significant impact on the convergence properties of the optimization process, this is not the case for the signal distortion, whose influence can be virtually neglected.

Introduction

Among stages of lightning stroke phenomena the return stroke is potentially the most critical one with regard to the electromagnetic (EM) susceptibility of overhead transmission lines [8, 38, 73, 156]. EM pulses caused by strokes cover the frequency spectrum from low tens of Hz to hundreds of MHz [38, 121]. These signals can be observed on power-delivery transmission lines at distances up to several kilometers from the lightning strike point [141]. In the event of an electric power system failure, with its potentially drastic consequences for industry, medical care and households, a matter of paramount importance is to locate and evaluate the cause of the failure, in view of designing the appropriate protection system [99]. Accordingly, developing an efficient and reliable return stroke localization methodology is the main objective of this work.

The majority of localization techniques is based on the principle of Time of Arrival (TOA) [55, 120, 194]. These systems employ at least three sensors to detect a change of an EM quantity induced by the lightning event. The time difference between the events from every two sensors defines a hyperbola representing the set of potential lightning strike points. The stroke position is then determined as the intersection of two or more such hyperbolas. The data from at least four uncorrelated sensors are necessary to perform a precise localization [136]. A system based on ToA can also work with other than EM signals e.g. acoustic signals [6, 14]. An intriguing idea in this respect is the use of data recorded by mobile phones of several users as proposed in [77]. Another category of lightning localization systems is based on the principle of Magnetic Direction Finding (MDF) [30, 182, 188]. Such MDF-based systems consist of two orthogonal loop antennas

measuring the magnetic field caused by the stroke event. The difference of magnetic field peak values determines the vector pointing towards the stroke location. The data from at least two antenna systems must be used to determine the stroke position precisely. The last group of localization methods relies on image processing [61, 123]. Since such systems mostly process the large amount of image data from satellites, they are mostly implemented on Field-Programmable Gate Array (FPGA) circuits [111, 137].

In a previous work [86], a return stroke localization methodology based on the analytical EM-field-to-line coupling model (see [176]) applying to the perfect ground has been introduced. In that initial study, the inverse problem is solved by combining the DE (Differential Evolution) global optimization algorithm [149] with the closed-form analytical model serving as the forward solver. In the present paper, we extend the methodology by accounting for a finite ground permittivity and conductivity using the analytical model introduced in [177]. On top of this, a comparative study investigating the use of four global optimization algorithms, namely, DE [149], the Particle Swarm Optimization (PSO) [163], Genetic Algorithm (GA) [78] and the Covariance Matrix Adaptation–Evolution Strategy (CMA–ES) [69] is presented and thoroughly discussed. Namely, the influence of problem parameters including the size of the search space, the fluctuation of pulse parameters, the noise level and the presence of an echo in the voltage response, on the localization error is discussed.

Problem Description

The goal of the optimization process is to find the location of the lightning return stroke described by $\mathbf{r} = x_s \mathbf{i}_x + y_s \mathbf{i}_y$. A vertical electric dipole (VED) source modeling a section of the return stroke is located at a height $h > 0$ above the ground plane at $z = 0$. Its pulse shape can be described by a sum of two pulses of the type [141]

$$i(t) = \frac{I_0}{\eta} \frac{(t/\tau_1)^n}{1 + (t/\tau_1)^n} \exp(-t/\tau_2) \quad (4.1)$$

where the amplitude correction factor η reads:

$$\eta = \exp[(-\tau_1/\tau_2)(n\tau_2/\tau_1)^{1/n}] \quad (4.2)$$

where $t > 0$ is the time coordinate, I_0 is the amplitude of the pulse and τ_1 and τ_2 denote the front time and decay constants, respectively. The pulse signature used throughout this work is composed of two pulses shown in Fig. 5.26 (pulse 1: $I_0 = 10.7$ kA, $\tau_1 = 0.25$ μ s, $\tau_2 = 2.5$ μ s, and $n = 2$, pulse 2: $I_0 = 6.5$ kA, $\tau_1 = 2.1$ μ s, $\tau_2 = 230$ μ s, and $n = 2$). Such a pulse shape mimics well real lightning return strokes [36].

To localize the point in the problem configuration (see Fig. 5.25), we employ a right-handed Cartesian system with the origin \mathcal{O} and the three orthogonal base vectors $\{\mathbf{i}_x, \mathbf{i}_y, \mathbf{i}_z\}$. The transmission line of length $L > 0$ is located at a height $z = z_0 > 0$ above a homogeneous ground described by its (scalar and real-valued) electric permittivity ϵ_1 , electric conductivity σ_1 and permeability μ_0 . The transmission line is placed in the homogeneous, isotropic and loss-free free-space, whose EM constitutive properties are defined by permittivity ϵ_0 and permeability μ_0 . The EM wave propagates here at the wave speed given by $c_0 = (\epsilon_0 \mu_0)^{-1/2}$.

The lightning stroke localization is formulated as a single-objective two-dimensional optimization problem. The objective function compares the measured (target) voltage

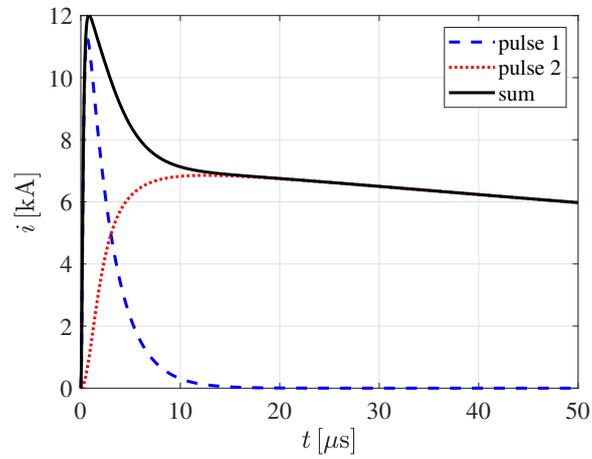


Figure 4.1: Current pulses forming the lightning stroke [86].

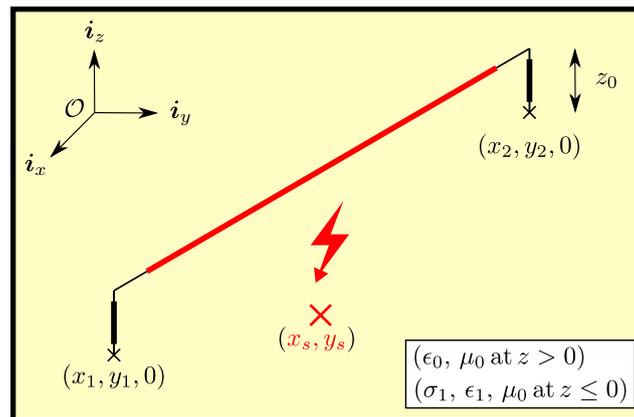


Figure 4.2: The lightning stroke localization problem description.

response $V_{1,m}$ at end 1 of the transmission line with the one simulated (computed) by the forward solver denoted as $V_{1,s}$ [86]:

$$\begin{aligned} \min \quad & f(\mathbf{r}) = \sum_{i=1}^N [V_{1,m}(i) - V_{1,s}(i, \mathbf{r})]^2 \\ \text{subject to} \quad & \mathbf{r} \in \Omega \end{aligned} \quad (4.3)$$

where N is the total number of time samples (i goes from 1 to N) and Ω is the feasible region for the stroke position \mathbf{r} . Equation (6.17) is designed to minimize the squared error between the target and the computed voltage response time samples. The model applied to calculating the both voltage responses: $V_{1,m}$ pertaining to the desired return stroke position, and $V_{1,s}$ pertaining to the (trial) position returned by the optimization algorithm, is described in Appendix 4.1 (see Sec. 4.1). Unless otherwise stated, we assume that the voltage responses $V_{1,m}(i)$ and $V_{1,s}$ are synchronized precisely.

Forward Solver Description

We apply the analytical time-domain coupling model based on the Cooray–Rubinstein formula (see e.g. [26]) which has been recently introduced in [176] and [177]. The analytical method has been validated in [177] against results of other numerical methods presented in [66] and [153]. For the sake of brevity, only the equations necessary for calculating the voltage response $V_{1,s}$ are summarized here and in Appendix 4.1. The formulas apply to the stroke located above the origin at $\mathbf{r} = \{0, 0, h\}$ and to the transmission line occupying $\{x_1 \leq x \leq x_2, y = y_0, z = z_0\}$. In the solution procedure, a straightforward linear transformation is performed whenever the stroke's location is changed.

The voltage response can be decomposed into two parts:

$$V_{1,s}(t) = V_{1,s}^0(t) + \Delta V_{1,s}(t) \quad (4.4)$$

where superscript 0 denotes the voltage response applying to the perfect ground and $\Delta V_{1,s}(t)$ is the correction term accounting for the finite ground permittivity and conductivity [177]

$$\begin{aligned} \Delta V_{1,s}(t) \simeq & Z(t) *_t \partial_t j(t) *_t [\mathcal{M}(x_1, y_0, z_0 + h, t) \\ & - \mathcal{M}(x_2, y_0, z_0 + h, t - L/c_0)] \end{aligned} \quad (4.5)$$

where $*_t$ denotes the continuous time-convolution operator and ∂_t is the time derivative.

Next, the voltage response pertaining to the perfect ground can be found from [176]

$$\begin{aligned} V_{1,s}^0(t) \simeq & -\mathcal{Q}(x_1|x_2, y_0, h - z_0, t) \\ & + \mathcal{Q}(x_1|x_2, y_0, z_0 + h, t) \\ & + \mathcal{V}(x_1, y_0, t) - \mathcal{V}(x_2, y_0, t - L/c_0) \end{aligned} \quad (4.6)$$

for $\{0 < z_0 < h\}$ and

$$\begin{aligned} V_{1,s}^0(t) \simeq & \mathcal{Q}(x_1|x_2, y_0, z_0 - h, t) \\ & + \mathcal{Q}(x_1|x_2, y_0, z_0 + h, t) \\ & + \mathcal{V}(x_1, y_0, t) - \mathcal{V}(x_2, y_0, t - L/c_0) \end{aligned} \quad (4.7)$$

for $\{z_0 > h\}$. Please refer to Appendix 4.1 for the computation of undefined quantities in (4.5)-(4.7).

The computational approach described above incorporates real electrical parameters of the ground under the transmission line, namely relative permittivity and electric conductivity. Using this computational tool extends the localization system proposed in our previous work [86].

In case that a more complex transmission line is considered the applied forward solver can be replaced by any other method, including numerical ones without affecting the optimization procedure. However, the use of an alternative solver may lead to an increase of the computational time.

Numerical Examples

In the numerical experiments that follow, we consider a transmission line of length $L = 1.0$ km located at $\{-L/2 \leq x \leq L/2, y_0 = 0, z_0 = 8.0 \text{ m}\}$ above a lossy ground described by $\epsilon_1/\epsilon_0 = 10$ and $\sigma_1 = 0.01 \text{ S/m}$. In every single optimization run, the position of stroke is generated randomly in domain Ω that is defined as a rectangle occupying $\{-2L < x < 2L, L/100 < y < L\}$. Owing to the problem symmetry (with respect to the transmission line), the half-plane left to the transmission line is assumed only. The problem with ambiguous stroke positions can be solved by using the data from sensors at the ends of two different non-parallel and non-crossing transmission lines. The optimization process is then executed separately for every transmission line which produces two sets of two stroke locations with the correct one being present in both the sets of solutions. The accompanying increase of the computational time can be handled by parallelizing the data processing.

The length of the TD voltage response is $\{0.0 \leq c_0 t/L \leq 2.5\}$ at 1200 samples. The simply vectorized MATLAB code for the voltage response is executed approximately in 0.13s on a standard PC with an AMD Ryzen 7 1700X platform and 32 GB of RAM. The computational time could be further reduced by using a proper code optimization, the use of Python or C language etc. However, this is out of the scope of the paper. Statistical data presented in this study are based on 100 repetitions for every algorithm and parameters combination.

The complexity of the optimization problem as defined by Eq. (6.17) is shown in Fig. 4.3. The objective function is overall very flat with a very thin needle-shaped global minimum and plenty of local minimums and maximums. Nevertheless, the difference between the global minimum value and those in local minimums is more than three orders of magnitude. Also, the problem is two-dimensional only, which indicates that it can be solved efficiently and reliably in a reasonable time. The performance of four different global optimization algorithms and the influence of several problem parameters is discussed in the following subsections.

Optimization Algorithms Comparison

As can be observed in Fig. 4.3, the objective function consists of a high number of local minimums and maximums, which prohibits from using efficient gradient-based algorithms. Therefore, four global state-of-the-art algorithms are applied to solve the problem: GA [78], PSO [163], DE [149], and CMA-ES [69]. A detailed description of the algorithms

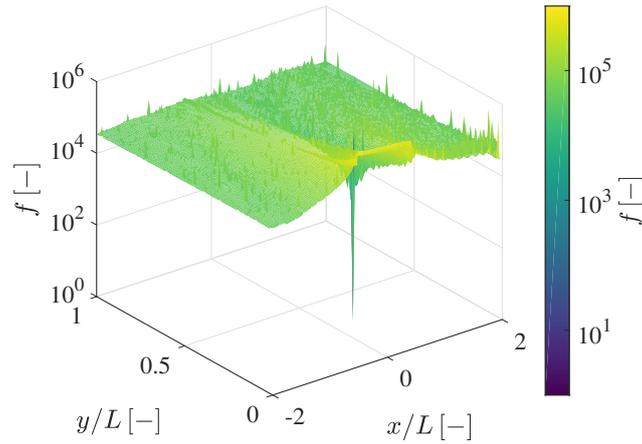


Figure 4.3: Values of the objective function for all possible return stroke locations in domain Ω . The stroke's location was associated with the peak observed at $\{L/3, L/3\}$.

settings is summarized in Tables 4.2-4.5 in Appendix 4.1. To get a fair comparison, all algorithms use 30 agents and 50 iterations, so that the number of fitness-function evaluations is still the same.

The results are summarized in Table 4.1 which contains the average, standard deviation, minimum, and maximum values of the localization error defined by

$$\varepsilon = \|\mathbf{r} - \mathbf{r}_s\| \quad (4.8)$$

where \mathbf{r} denotes the stroke location found by the optimization algorithm and \mathbf{r}_s is the true stroke event location. Another observed metric is the so-called percentage success rate that is defined as:

$$SR = 100(N_s/N) \quad (4.9)$$

where N is the total number of trial runs, and N_s is the number of successful runs. A run is considered as successful if the localization error ε drops under 3.0 m.

Results for individual algorithms are presented also in the form of conventional box plots in Fig. 4.4. Here, the red line marks the average value, the blue bottom and top edges mark the 25-th and 75-th percentile, respectively, and red markers denote the so-called outliers (7% of data that are the most abnormal according to the normal distribution of probability).

The algorithm CMA-ES clearly outperforms the other algorithms. It achieves the best values in all the observed metrics. Remarkably, as shown in the column CMA-ES* of Table 4.1, the CMA-ES algorithm achieves comparable or even better results with respect to DE and GA with only a quarter of fitness function computations (400 compared to 1500).

The reason behind such an outstanding performance of the CMA-ES algorithm is in the noisy shape of the objective function (see Fig. 4.3) for which this algorithm works very well, which can be demonstrated by various comparative studies (e.g. [68,189]). This high performance can be explained by the mechanism that CMA-ES uses for the creation of its new trial positions. It produces them using the multi-dimensional normal distribution that is distorted according to the covariance matrix computed from the previous trial

positions with an emphasis on the best solution. Consequently, every part of the search domain can be explored with a great detail, but the algorithm prevents from getting stuck in the local minimum, especially for the later iterations of its run. Algorithm CMA-ES achieves the outstanding search rate $SR = 98$ [%] with the worst localization error only $\varepsilon = 3.74$ m. Only the algorithm PSO achieves an acceptable result of the search rate $SR = 85$ %, while the other two - DE, and GA - end up with $SR = 10$ % and $SR = 4$ %, respectively. Results of the comparative study shown in Fig. 4.4 indicate that all the considered algorithms achieved different levels of the localization error for the fixed computational resources (see ε_{avg} and ε_{min} in Tab. 4.1). Algorithms PSO, DE, and GA can solve the problem as well, but at the expense of a much longer computational time to reach the same localization error level as for CMA-ES.

The localization methodology has been improved (1) by incorporating the influence of electric conductivity and permittivity, and (2) by employing the CMA-ES algorithm. Consequently, the average localization error decreased to $\varepsilon_{\text{avg}} = 0.16$ m compared to $\varepsilon_{\text{avg}} = 1.34$ m, which was achieved in Ref. [86] assuming the idealized PEC ground model. The overall search rate increased to $SR = 98$ % compared to $SR = 78$ % from [86].

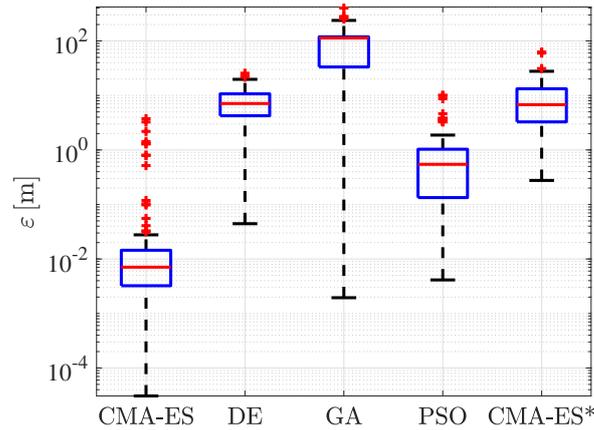


Figure 4.4: Box plots of the stroke localization error obtained for different optimization algorithms.

Table 4.1: Comparison of the return stroke localization error for different optimization algorithms. Simulation denoted as CMA-ES* used less agents 20 and iterations 20.

Algorithm	CMA-ES	DE	GA	PSO	CMA-ES*
ε_{avg} [m]	1.64×10^{-1}	8.29	9.83×10^1	1.15	9.44
σ_{ε} [m]	5.84×10^{-1}	5.11	7.94×10^1	1.86	1.01×10^1
ε_{min} [m]	3.07×10^{-5}	4.44×10^{-2}	1.95×10^{-3}	4.13×10^{-3}	2.76×10^{-1}
ε_{max} [m]	3.74	2.56×10^1	4.00×10^2	1.01×10^1	6.26×10^1
SR [%]	98	10	4	85	21

Influence of Problem Parameters

At first, we add the Additive White Gaussian Noise (AWGN) to the measured voltage response $V_{1,m}$. The box plots for $SNR = \{10, 15, 20, 25, 30\}$ (in dB) are shown in Fig. 4.5. Here we can observe that the average error decreases slightly with the growing SNR values. Also, the search rate enhances very slightly with the growing SNR as $SR = \{94, 97, 96, 98, 98\}$ (in %) for the corresponding SNR values. A small variation of the localization error with the growing noise level implies a good stability and robustness of the proposed methodology.

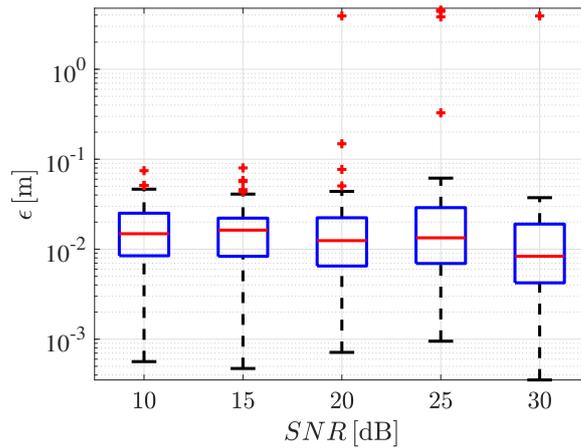


Figure 4.5: Box plots of the stroke localization error for the target response corrupted by AWGN with different SNR (CMA-ES algorithm).

Next parameter whose influence will be studied is the size of search domain Ω . We shall limit our analysis to rectangular domains Ω . Namely, we consider a rectangular domain of a height $h = w/2$ and a width $w > 0$, the latter is taken to be a multiple of L . As can be seen in Fig. 4.6, the size of Ω has to be chosen thoroughly since it has the dominant impact on the success rate of the localization. Our methodology seems to be applicable to domains with approximately up to $w/L = 5$ ($\varepsilon_{\text{avg}} = 0.38$ m). Increasing its dimensions further, the localization error grows very quickly (the average value for $w/L = 10$ reaches $\varepsilon_{\text{avg}} = 73.3$ m). A better performance could be achieved at the expense of an extended computational time, as more agents and iterations would be needed. It is evidenced by the convergence plots for different w/L sizes in Fig. 4.7. The convergence of localization error ϵ based on number of objective function computations n_f . The localization error $\epsilon < 3$ m can be reached under approximately 1000 objective function computations for sizes $w/L \leq 5$. It would be necessary to use approximately 4000, and 8000 n_f to achieve the same level of localization error for $w/L = 10$ and $w/L = 20$, respectively.

The measured response can be corrupted also by echo signals. Echo signal V_e represents a reflected signal arriving at the observation point from the stroke location. The echo signal is, in fact, an attenuated and delayed copy of the primary (direct) signal V_d . Their sum is observed at the location of interest. An example of such a combination of the primary signal with the echo is shown in Fig. 4.8. The echo was attenuated 10-times with respect to the primary signal and its delay is $c_0 t = 200$ m. The influence of the presence of echo to the optimization results can be observed in Fig. 4.9. Here, a histogram of

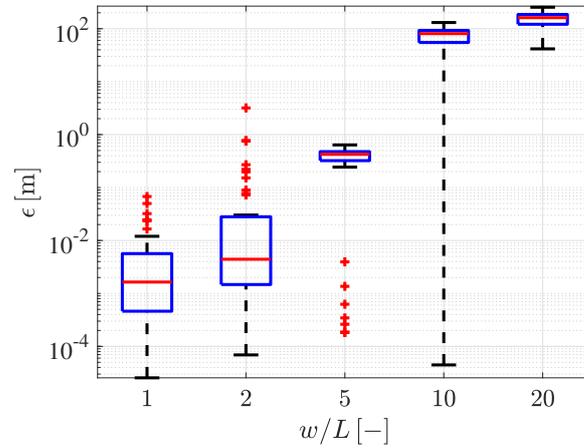


Figure 4.6: Box plots of the stroke localization error obtained for different sizes of the search domain Ω .

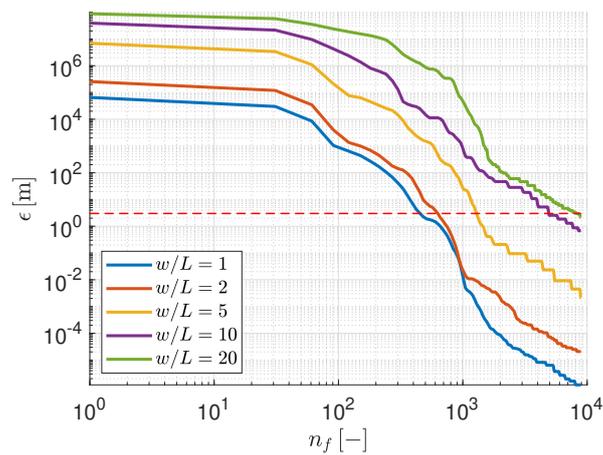


Figure 4.7: Convergence plots of the localization error ϵ vs. the number of objective function computations n_f for different search domain sizes Ω (expressed by the w/L ratio). Red dashed line marks the *SR* limit $\epsilon = 3$ m.

the localization error ε for the measured signal $V_{1,m}$ with $(V_d + V_e)$ and without (V_d) the echo signal are compared. As can be seen, the error rate remains almost unchanged. The search rate is very high for the both scenarios: $SR = 98\%$ for the primary signal only and even higher, i.e. $SR = 99\%$ for the voltage response with the echo signal. The presence of the echo signal does not change the strong initial part of the voltage response. Therefore, the optimization algorithm can deduce the correct stroke position from the early-time response. The remaining corrupted part of the response increases the objective function value for the correct position, but also for fault positions.

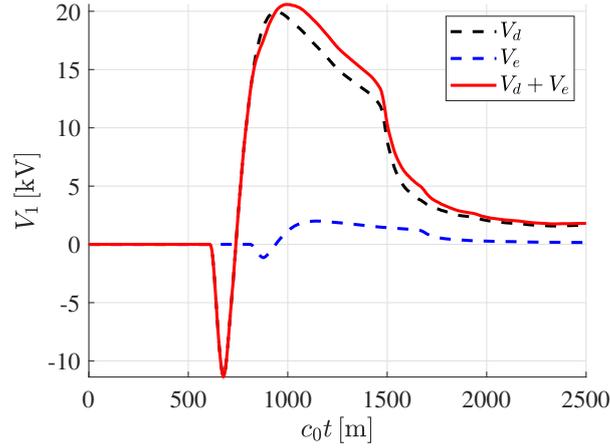


Figure 4.8: Sum of primary wave voltage response V_d with the reflected echo response V_e .

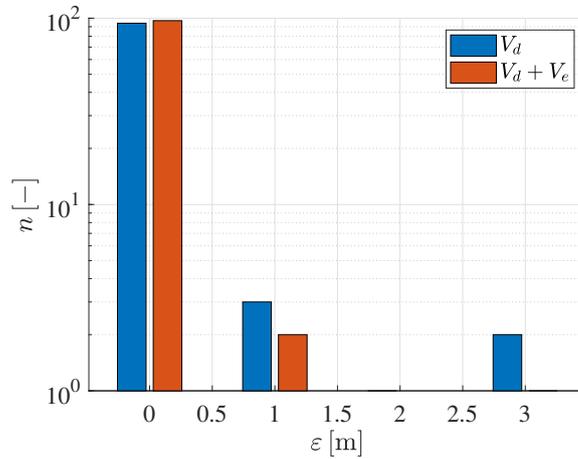


Figure 4.9: Histogram of localization error when the voltage response of a direct wave V_d and a wave with the reflected echo V_e is used.

The pulse parameters defined in Sec. 5.2 (see Fig. 5.26) used for the computation of voltage response V_s were experimentally set to model actual return stroke pulses [141]. The robustness of the localization methodology against fluctuations of the pulse parameters is a very limiting factor. Therefore, we examine here the effect of the variable pulse width and pulse peak amplitude (see Eq. (6.24)) as used to calculate the "measured" voltage response V_m on the method's error. The stability of the localization error to the

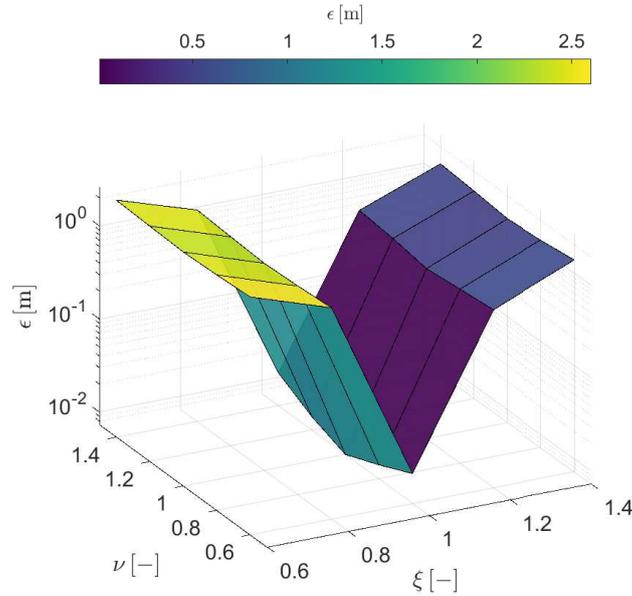


Figure 4.10: The stroke localization error obtained for different pulse width ratio ξ and pulse peak amplitude ratio ν values.

pulse parameters fluctuation is shown in Fig. 4.10. The pulse width change is expressed by means of parameter ξ , that is defined as the ratio of the width of $i(t)$ as used for calculating V_m at the level of $i(t) = 8$ kA and the width of $i(t)$ as used for calculating V_s at the same current level. This implies that if $\xi = 1.0$, the same pulse for V_m and V_s is used. The parameter ν denotes the pulse peak amplitude ratio. It is defined as the ratio of the peak value of $i(t)$ used for calculating V_m and the amplitude value of $i(t)$ used for calculating V_s . This means that if $\nu = 1$, the same pulses are used for calculation of V_m and V_s . As can be seen in Fig. 4.10, the localization error does not grow significantly with the change of the pulse amplitude. On the other hand, the error increases with the change of the pulse width. Anyway, the localization error remains below three meters throughout the entire range of the considered ξ and ν values.

Conclusion

In this paper, a fast lightning localization technique based on evolutionary optimization algorithms and a closed-form TD analytical model for calculating lightning-induced voltages on an overhead transmission line has been described. It has been demonstrated that the incorporation of the ground electric permittivity and conductivity improves the proposed localization system. The localization problem was defined as a causation inverse problem and the analytical coupling model based on the Cooray–Rubinstein formula was used as a forward solver. The solution of the inverse problem was obtained with the help of four global optimization algorithms. Our comparative study clearly proved that the CMA-ES algorithm is the most suitable candidate for solving the localization problem. The influence of other problem parameters has been investigated, namely, the influence of the level of noise added to the voltage response, the presence of the echo signal in the measured voltage response, the size of the return stroke search domain, and the fluctuation of the return stroke pulse width. Based on the results of this work it can be concluded that

only the size of the search domain reduces the success rate of the CMA-ES algorithm significantly.

Appendix 1: Voltage Response Computation

The space-time function $\mathcal{M}(x, y, z, t)$ used in (4.5) is computed:

$$\mathcal{M}(x, y, z, t) = (1/2\pi R)\mathcal{N}(x, y, z, t)\mathbb{H}(t - R/c_0) \quad (4.10)$$

with $\mathbb{H}(t)$ being the Heaviside step function and

$$\begin{aligned} \mathcal{N}(x, y, z, t) = & (c_0 + x)^{-2} [c_0 t (c_0 t + 2x) - xR \\ & - \frac{R(c_0^2 t^2 - x^2) + c_0 t (y^2 + z^2)}{R + c_0 t}] \end{aligned} \quad (4.11)$$

in which

$$R = R(x, y, z) = \sqrt{x^2 + y^2 + z^2} \quad (4.12)$$

In Eq. (4.5), $j(t)$ (in A · m) represents a traveling current pulse

$$j(t) = i(t - h/v) \exp(-h/\lambda) \Delta h \quad (4.13)$$

where $i(t)$ is the current pulse defined by Eq. (6.24), $\Delta h > 0$ is the spatial step for integration along the stroke channel height. The velocity of the stroke inside the channel is $v = 1.30 \times 10^8$ m/s and $\lambda = 2$ km. The time-domain surface impedance $Z(t)$ is computed from

$$Z(t) = \zeta_1 \{ \delta(t) - (\chi/2) [\mathbb{I}_0(\chi t/2) - \mathbb{I}_1(\chi t/2)] \mathbb{H}(t) \} \quad (4.14)$$

where $\mathbb{I}_{0,1}(t)$ are the scaled modified Bessel functions of the first kind, ζ_1 is the wave impedance corresponding to the “high-frequency limit” of the surface impedance $\hat{Z}(s)$ (see [177])

$$\zeta_1 = \sqrt{\mu_0/\epsilon_1} \quad (4.15)$$

and

$$\chi = \sigma_1/\epsilon_1 \quad (4.16)$$

For the PEC ground response as computed by (4.6) and (4.7), we used

$$\begin{aligned} \mathcal{Q}(x_1|x_2, y, z, t) = & \zeta_0 \partial_t j(t) *_t [\mathcal{F}(x_2, y, z, t - L/c_0) \\ & - \mathcal{F}(x_1, y, z, t)] \end{aligned} \quad (4.17)$$

and

$$\begin{aligned} \mathcal{F}(x, y, z, t) = & [z/4\pi(y^2 + z^2)] \\ & \times \mathcal{P}(x, y, z, t) \mathbb{H}(t - R/c_0) \end{aligned} \quad (4.18)$$

with

$$\begin{aligned} \mathcal{P}(x, y, z, t) = & 1/Rc_0 t \left[xc_0 t - x^2 \right. \\ & - \frac{R(c_0^2 t^2 - x^2) + c_0 t (y^2 + z^2)}{R + c_0 t} \\ & \left. + \frac{c_0^2 t^2 (y^2 + z^2)}{R^2} \right] \end{aligned} \quad (4.19)$$

Furthermore, in Eq. (4.6) and (4.7) we substitute

$$\mathcal{V}(x, y, t) = \mathcal{U}(x, y, h - z_0, t) - \mathcal{U}(x, y, h + z_0, t) \quad (4.20)$$

for $\{0 < z_0 < h\}$ and

$$\begin{aligned} \mathcal{V}(x, y, t) = & 2\mathcal{U}(x, y, 0, t) - \mathcal{U}(x, y, z_0 - h, t) \\ & - \mathcal{U}(x, y, z_0 + h, t) \end{aligned} \quad (4.21)$$

for $\{z_0 > h\}$, where

$$\mathcal{U}(x, y, z, t) = \zeta_0 \partial_t j(t) *_t \mathcal{G}(x, y, z, t) \quad (4.22)$$

in which

$$\begin{aligned} \mathcal{G}(x, y, z, t) = & (1/4\pi) [(c_0^2 t^2 - x^2 - y^2)^{-1/2} \\ & - z c_0 t R^3] \text{H}(t - R/c_0). \end{aligned} \quad (4.23)$$

Appendix 2: Optimization Algorithms Settings

For the meaning of individual parameters used in following tables please refer to [125].

Table 4.2: Settings of the Genetic Algorithm.

PC [-]	PM [-]	BP [-]	TS [-]	nCP [-]
0.9	0.7	20	2	1

Table 4.3: Settings of the Particle Swarm Optimization Algorithm.

W [-]	$C1$ [-]	$C2$ [-]	BT [-]
[0.9, 0.4]	1.5	1.5	'Reflecting'

Table 4.4: Settings of the Differential Evolution Algorithm.

F [-]	PC [-]
0.2	0.2

Table 4.5: Settings of the Covariance Matrix Adaptation - Evolutionary Strategy Algorithm.

μ [-]	α_c [-]	BT [-]
15	2	'Reflecting'

Chapter 5

Multi-objective Optimization

The multi-objective optimization of EM problems was the main concern of the Ph.D. thesis of the author [81]. The dissertation was dedicated to extension of the Self-organizing Migrating Algorithm to handle multi-objective problems. The new algorithm was called as Multi-objective Self-organizing Migrating Algorithm (MOSOMA). The main contributions of the thesis were summarized in form of two book chapters: [92] summarizes the algorithm and [91] shows results of the algorithm on various design problems. The MOSOMA was then applied to design of various types of layered filters (a low-pass, band-pass and band-stop filter) and the design of the Yagi-Uda antenna in the journal paper [90] (see the reprinted version in Section 5.1). The results of the MOSOMA algorithm on these design problems are compared to other state-of-the-art algorithms (MOPSO and NSGA-II).

A conference paper [84] presents a multi-objective optimization of the testing pulse for a time-domain shielding effectiveness of thin metal sheets (please see the reprinted version in Section 5.2). The MOPSO algorithm is used to search for optimal values of pulse parameters (namely the rise time, the pulse rising power and the ringing frequency) to alter the test sensitivity according to specific requirements. Results show the theoretical level of the test adjustments and the influence of the individual pulse parameters to the shielding-effectiveness.

5.1 Multi-objective self-organizing migrating algorithm applied for design of electromagnetic components

Originally published as:

Kadlec, P., Raida, Z.: Multi-objective self-organizing migrating algorithm applied for design of electromagnetic components. In *IEEE Antennas and Propagation Magazine*, 55(6), pp. 50-68, 2013.

Abstract

Real life design problems of electromagnetic components are usually highly non-linear. These problems can be efficiently solved with stochastic global optimization algorithms. This paper deals with an application of a novel Multi-objective Self-organizing Migrating Algorithm (MOSOMA) to the design of two electromagnetic components: layered dielectric filters and Yagi-Uda antennas. Optimization of dielectric filters considers two objectives: minimization of reflection in the pass band and its maximization in the stop band. Band-pass, low-pass and band-stop filters having seven dielectric layers are optimized here. An option for treatment of both continuous parameters and discrete ones and dealing with optimization constraints without any change in the optimization algorithm is briefly discussed. Yagi-Uda antenna optimization deals with two-objectives: maximization of gain and minimization of relative side-lobe level, while impedance matching is considered as a constraint for the proposed designs. Yagi-Uda antennas are analyzed using 4NEC2 software based on the method of moments. Co-operation of Matlab optimization script and 4NEC2 software in a non-interactive mode is explained. Results for four- and six-element antennas are presented. Results of both the problems are compared to results from available references.

Introduction

Since almost every optimization problem can be viewed from more than one side, importance of efficient multi-objective optimizers grows. These algorithms are able to find so called Pareto front of solved problem. This set is built by solutions that are optimal from the viewpoint of all the objectives at the same time. Pareto front expresses the trade-off between individual objectives.

The aggregation methods (summation of weighted objectives) suffer with various problems [45]. Mostly, aggregation of multiple objectives into one large fitness function brings a problem of proper settings of weights for individual objectives. Consider, you have to set the weights a priori which is almost impossible without a good experience with a solved problem. For example assuming these weights to be equal (we do not prefer any of the objectives), optimization process usually does not find a solution from the middle of the Pareto front as expected, but the proposed solution prefers one of the objectives significantly [45]. Next, individual objectives have to be normalized with their maximum value to keep all the entries of the aggregated function in the same scale. Finding of maximum for individual objectives is again an optimization task. Moreover, most of the

aggregation methods are not able to find solutions from the concave part of the Pareto front as shown in [87].

On the contrary, searching for so called Pareto front of the problem with multiple objectives gives the user some additional knowledge about limits and dependencies of the considered objectives and enables to choose carefully the final solution of the problem. This approach gives a designer chance to take into account the true importance of individual objectives. The final solution can be then selected according to extra information about the solved problem either manually or by a machine. With knowledge of the Pareto front, the chosen trade-off solution then truly respects your preferences for particular objectives (weight of individual objectives is selected according to the shape of the found Pareto front). On the contrary, in case of the methods aggregating all the objectives, you don't know, if the weights for particular objectives (you have to select a priori before the start of optimization) respect the true trade-off between objective function values of the found solution (just a single one).

Multi-objective self-organizing migrating algorithm (MOSOMA) is one of the most recent multi-objective optimizers. Two-objective version of MOSOMA has been introduced in [88]. In [89], MOSOMA has been extended so that it is able to solve efficiently problems with any number of objectives. MOSOMA has shown very good performance on various mathematical benchmark problems [88, 89] where results of various convergence metrics are compared with commonly used multi-objective algorithms NSGA-II [46] and SPEA2 [211]. MOSOMA achieves at least comparable results in all watched metrics.

Following two paragraphs try to explain the differences between previous papers concerning MOSOMA and this one. Paper [87] has compared pure multi-objective algorithm MOSOMA with conventional aggregating methods on benchmark problems to justify the derivation of pure multi-objective optimization method based on self-organizing migration. Almost the whole paper [89] was focused on extension of MOSOMA to solve problems with more than two objectives. Functionality of our approach has been shown on design of dielectric filters. The filter design has been considered as a three-objective problem (reflection minimization in the pass-band, reflection maximization in the stop-band and total length minimization). Then, MOSOMA has been employed for control of time domain adaptive beam-forming of slot antennas array in [155]. As it was a joint paper covering much larger area than EM multi-objective optimization MOSOMA application could not be discussed in appropriate detail.

This paper deals with application of generally working (within the meaning of objective space size) algorithm MOSOMA for design of EM components: dielectric filters and Yagi-Uda antennas. First, the run of the algorithm is described especially from the programming point of view (Matlab scripts of the algorithm can be downloaded for free from authors website). Then, practical issues arising during the design process like constraints handling and working with decision space built by continuous and discrete variables at the same time are discussed. Also co-operation of our optimization algorithm with external EM solver is explained here.

Venkatarayalu et al. formulated the optimization of widths and relative permittivities of individual layers of a filter as constrained two-objective problem in [191] where authors proposed a new evolutionary algorithm (MOEA) for its solution. Goudos et al. used a multi-objective algorithm based on swarm intelligence (MOPSO) for the solution of band-pass, low-pass and band-stop filter design [62].

The optimal design of Yagi-Uda antenna is a challenging problem for various authors

from the 20-th century. Cheng [31] put a lot of efforts to the application of gradient based methods to the search of optimal lengths of elements and their spacing. Cheng observed that gain of Yagi-Uda antenna is described by a highly non-linear function and therefore, the solution provided by gradient based methods strongly depends on an initial guess. Therefore, lot of authors tried to apply global optimization methods: Jones and Joines employed the single-objective binary-coded genetic algorithm (GA) [79]. Baskar et al. [18] used comprehensive learning particle swarm optimization (CLPSO). All previously mentioned authors solved the Yagi-Uda design problem as a single-objective task using one aggregated fitness function composed of individual objectives considering gain, relative side lobe level and impedance matching. Nevertheless, several variants of Yagi-Uda antennas were optimized as a constrained two-objective problem (maximization of gain and minimization of side lobe level as objectives and impedance matching as a constraint) in [190] by Venkatarayalu et al. Kuwahara considered the impedance matching as a third objective instead of constraint in [110].

In this paper, properties of the multi-objective optimization are briefly reviewed to be introduced to readers. Then, basic principles of MOSOMA are described and an illustrative run of MOSOMA on a simple benchmark problem is derived. Finally, experimental results of the application of MOSOMA to both the design problems are discussed and compared with solutions obtained by other authors.

Multi-objective optimization

Optimization can be understood as a process of finding and comparing feasible solutions of a solved problem until the best solution is found. Quality of the solutions is expressed by means of an objective function. Usually, the solved problem has to be described by more than one objective (e.g. price, robustness, reliability, size of some device, etc.). If the objectives are conflicting, optimization leads to a set of so-called Pareto-optimal solutions that represent a trade-off among all objectives.

The multi-objective optimization problem can be formulated as follows:

$$\begin{aligned}
 \min_{\mathbf{x}} \quad & F_m(\mathbf{x}) \\
 \text{s.t.} \quad & \mathbf{x} \in \Omega, \\
 & g_j(\mathbf{x}) \leq 0, \\
 & m = 1, 2, \dots, M, \\
 & j = 1, 2, \dots, J
 \end{aligned} \tag{5.1}$$

where M denotes the number of objective functions F_m , N stands for the number of decision variables, \mathbf{x} is the vector of decision variables for individual solution, $x_{n,\min}$ and $x_{n,\max}$ are lower and upper bounds for the n -th variable from the N -dimensional decision space. The symbol J stands for the number of constraint functions g_j . Every proposed solution \mathbf{x} defined in the decision space (see Figure 5.1) can be represented in the objective space by means of values of the objective functions. Constraint functions g_j divide the objective space into a feasible and an infeasible subspace.

Operation of almost every stochastic multi-objective optimization algorithm is based on a principle of dominance. The principle of dominance compares two solutions from the viewpoint of all objectives at the same time [45]:

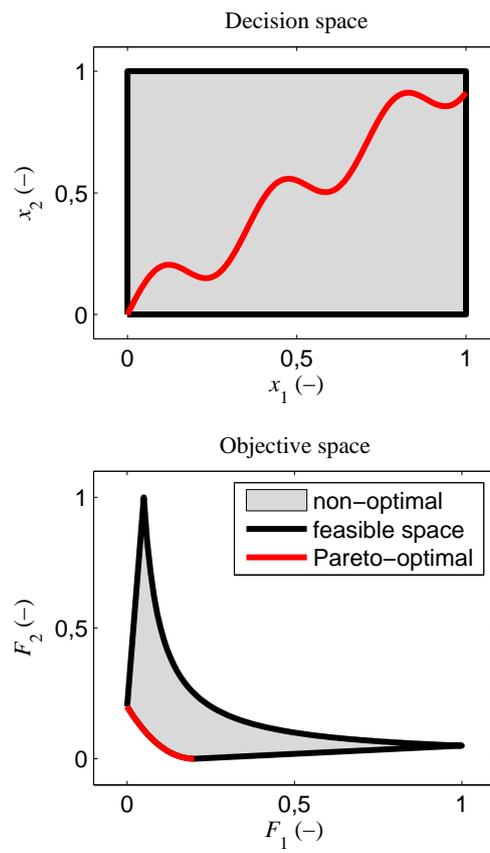


Figure 5.1: Pareto optimal solutions depicted in the decision and objective space.

Solution \mathbf{x}_1 is said to dominate the other solution \mathbf{x}_2 , if both conditions 1 and 2 are met:

1. Solution \mathbf{x}_1 is no worse than \mathbf{x}_2 in all objectives.
2. Solution \mathbf{x}_1 is strictly better than \mathbf{x}_2 in at least one objective.

Such a comparison can end with two different results: either one solution dominates the other (it is at least the same or better in all objectives) or both the solutions are non-dominated (any solution is better in all objectives).

The principle of dominance is observed in the objective space (see Figure 5.2). We have here a set \mathcal{Q} consisting of five solutions. The dashed lines mark out regions in the objective space that are dominated by a corresponding solution. The solution \mathbf{x}_1 dominates the solutions \mathbf{x}_2 and \mathbf{x}_3 , the solution \mathbf{x}_4 dominates the solution \mathbf{x}_5 . The solutions \mathbf{x}_1 and \mathbf{x}_4 are non-dominated. These two solutions build a non-dominated set (front) \mathcal{P} of the first order. The solutions \mathbf{x}_2 , \mathbf{x}_3 and \mathbf{x}_5 are also non-dominated and build a front of the second order.

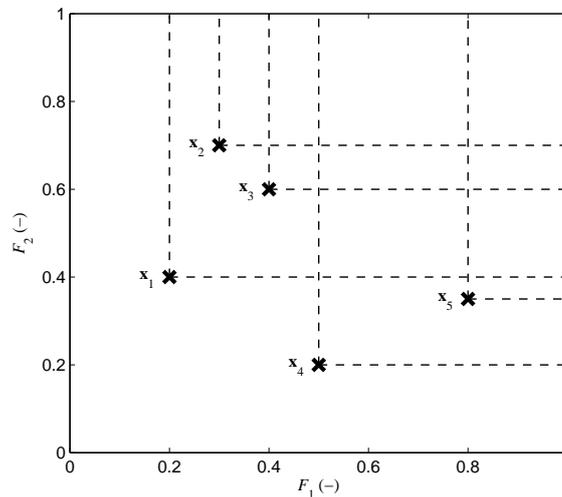


Figure 5.2: Principle of dominance in two-dimensional objective space.

Obviously, if the researched set \mathcal{Q} contains all possible solutions from the whole objective space, then non-dominated set contains all the Pareto-optimal solutions PF of the problem. Therefore, the concept of dominance can be used for an effective classification of proposed solutions from the viewpoint of multiple objectives.

The process of finding a non-dominated set \mathcal{P} is crucial for a fast convergence of every multi-objective optimizer. Intuitively, the comparison of the whole set of solutions \mathcal{Q} for evaluating the dominance is ineffective. Therefore, a continuously updated algorithm for assigning the set has been proposed in [46]. Its pseudocode is depicted in Figure 5.3 and the working Matlab code can be found at <http://www.urel.feec.vutbr.cz/kadlec/?Downloads:MOSOMA>.

Every multi-objective optimizer is aimed to reveal the whole Pareto front. This task contains two contradictory minor goals. First, members from the found set \mathcal{P} should be as close to the true Pareto front PF as possible. On the contrary, members from \mathcal{P} should be distributed along the whole Pareto front uniformly. Following both these goals is very

```

Start
  Insert  $\mathbf{x}_1$  from  $Q$  to  $P$ 
  For  $q = 2 : |Q|$ 
     $insert = 1$ 
    For  $p = 1 : |P|$ 
      If  $\mathbf{x}_q$  dominates  $\mathbf{x}_p$ 
        Delete  $\mathbf{x}_p$  from  $P$ 
      End
      If  $\mathbf{x}_p$  dominates  $\mathbf{x}_q$ 
         $insert = 0$ 
        Break incrementation  $p$ 
      End
    End
    If  $insert == 1$ 
      Insert  $\mathbf{x}_q$  to  $P$ 
    End
  End
End

```

Figure 5.3: Pseudocode of continuously updated algorithm for assigning set \mathcal{P} .

important for a proper understanding of trade-offs between objectives of an examined problem.

MOSOMA

A novel Multi-Objective Self-Organizing Migrating Algorithm (MOSOMA) extends the single-objective optimization method SOMA (Self-Organizing Migrating Algorithm) [206]. The multi-objective variant of the algorithm was derived in [88, 89]. Here, the main principles of MOSOMA will be described and then we walk through an example run of MOSOMA on a simple benchmark problem to illustrate the algorithm from the implementation point of view.

Algorithm overview

MOSOMA works in both the optimization domains. So called agents (vectors of state variables) migrate through the N -dimensional decision space and evaluate researched positions with values of M objective functions. These values are used for a non-dominated sorting of the whole group of agents. All agents migrate then towards the members of a so called external archive. So, the migration leads all agents closer to the true Pareto front as indicated in Figure 5.4. The overall functionality of MOSOMA can be described by following steps:

- Step 1: Defining controlling parameters of the algorithm.
- Step 2: Generating an initial population, evaluating objective functions.
- Step 3: Choosing an external archive from the current population.

- Step 4: Migrating agents to the members of the external archive. Evaluating objective functions for new positions. Updating the external archive. Selecting migrating agents for the next migration loop.
- Step 5: Testing stopping conditions. If no stopping condition is accomplished, go back to Step 4.
- Step 6: Choosing a final non-dominated set from the current external archive.

The user can enhance the convergence of the algorithm by felicitous setting of its controlling parameters. These parameters can be summarized as follows: Q_1 denotes the initial population size, T stands for the number of migrating agents, ST is the number of steps in one migration, PL denotes a relative length of a path for one migration, PR means the probability of perturbation and $N_{ex,min}$ stands for the minimal size of the external archive. The recommended intervals derived on behalf of our experience with the algorithm are summarized in Table 5.1. These intervals are defined in multiples of N (the number of optimized parameters) or $|Q_1|$ (the initial population size).

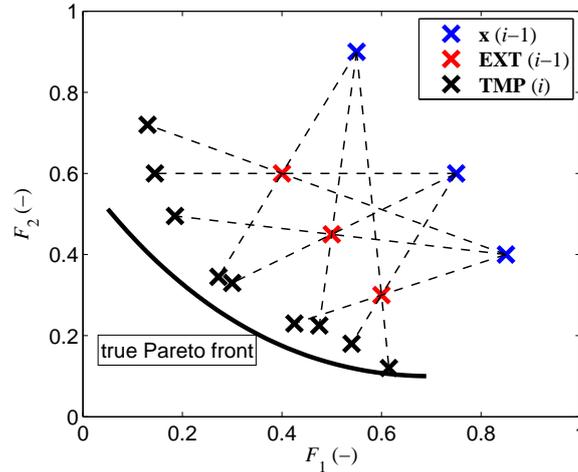


Figure 5.4: Migration of agents towards the external archive [88].

Table 5.1: Recommended values for controlling parameters of MOSOMA.

Parameter	Recommended interval
Q_1/N	$\langle 5, 12 \rangle$
T/N	$\langle 5, 10 \rangle$
PL	$\langle 1.1, 1.7 \rangle$
ST	$\langle 2, 5 \rangle$
PR	$\langle 0.1, 0.4 \rangle$
$N_{ex,min}/ Q_1 $	$\langle 1/3, 2/3 \rangle$

The procedure of MOSOMA is as follows. First, positions of Q_1 agents are randomly defined by the equation:

$$x_{q,n} = x_{n,\min} + rnd_{q,n}(x_{n,\max} - x_{n,\min}) \quad (5.2)$$

where $x_{q,n}$ denotes n -th variable of q -th agent, $\langle x_{n,\min}, x_{n,\max} \rangle$ denotes the feasible interval for n -th variable and $rnd_{q,n}$ is a random number from the interval $\langle 0, 1 \rangle$ with the uniform distribution of probability.

Then, objective functions are computed for the whole group Q_1 so that the non-dominated sorting can be performed. The non-dominated set \mathcal{P} is saved into the external archive EXT . If the size of the external archive is lower than its minimal size $N_{ex,\min}$ defined by the user, the archive is completed with best solutions from the fronts of advancing orders.

Now, the iterative process of finding the Pareto-optimal solutions is performed within the maximal $MIGS$ migration loops. During the i -th migration loop, selected agents migrate towards current members of the external archive. According to our experience, we should choose the group of migrating agents partly randomly (the premature convergence is suppressed) and partly from the members of the current EXT (the region of the so far found best solutions is researched carefully to speed-up the convergence). The migration procedure is indicated in Figure 5.4.

The positions visited during the i -th migration of the agent \mathbf{x}_t from \mathcal{T} towards the agent \mathbf{x}_p from \mathcal{P} are calculated by:

$$\mathbf{TMP}_{t,s} = \mathbf{x}_t(i-1) + s/ST[\mathbf{x}_p(i-1) - \mathbf{x}_t(i-1)]PL * \mathbf{PRTV}_{t,s} \quad (5.3)$$

where $\mathbf{TMP}_{t,s}$ is the vector specifying s -th position during the migration, and ST defines the number of steps for one migration ($s = 1, 2, \dots, ST$). Parameter PL denotes the multiple of the distance between agents \mathbf{x}_t and \mathbf{x}_p . Parameters PL and ST should be set such a way so that the migrating agent does not visit the position of the agent from EXT :

$$s \frac{PL}{ST} \neq 1 \forall s = 1, 2, \dots, ST. \quad (5.4)$$

So called perturbation vector \mathbf{PRTV} has the same size as the vector defining the position of an individual \mathbf{x} and consists of zeros and ones. The vector is defined for each migration by N randomly generated numbers:

$$\mathbf{PRTV}(n) = \begin{cases} 1 & \text{if } rnd(n) > PR \\ 0 & \text{if } rnd(n) \leq PR \end{cases} \quad (5.5)$$

where PR denotes a probability of perturbation defined by user. The perturbation has a similar influence on MOSOMA as the mutation for genetic algorithms. The perturbation protects the algorithm against a bottleneck in the local optimum (e.g. front of advancing order). The migration in the decision space with the explanation of the influence of the perturbation can be seen in Figure 5.5.

MOSOMA uses the absorbing boundary condition as defined in classical particle swarm optimization [163] when the agent visits during the migration places outside the feasible decision space. If any of the variable x_n is lower than $x_{n,\min}$ (or higher than $x_{n,\max}$), it is set the value $x_{n,\min}$ (or $x_{n,\max}$).

Typically, the size of the external archive grows with consecutive migration loops. This behavior slows the procedure of MOSOMA (T migrating agents migrates towards increasing number of agents in EXT). Therefore, three stopping conditions are combined: the total number of migration loops $MIGS$, the maximal size of the external archive $N_{ex,\max}$ and the limit for objective functions computations FFC . The algorithm stops

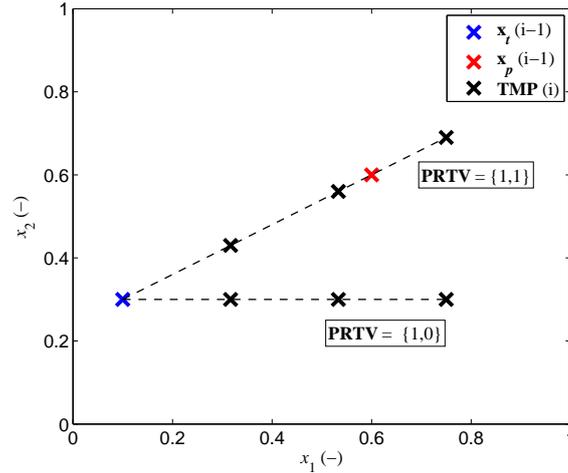


Figure 5.5: Migration of agent in the decision space with and without influence of the perturbation ($ST = 3$, $PL = 1.3$) [82].

whenever any of these conditions is fulfilled. The detailed pseudo-code of MOSOMA can be found in Figure 5.6. Usually, final external archive contains much more solutions than requested by user. Working with too large Pareto-optimal set slows down the final choice of the trade-off solution, if this is made manually. Therefore, we have to save into final non-dominated set \mathcal{P} those solutions from the EXT that cover the Pareto front uniformly. The detailed description for two-objective and M -objective variant of this procedure can be found in [88] and [82], respectively.

Illustrative run of MOSOMA

In this subsection, we will walk through a typical run of MOSOMA when solving a relatively simple benchmark problem. Interested readers can follow up also the codes available from <http://www.urel.feec.vutbr.cz/kadlec/?Downloads:MOSOMA> where three Matlab scripts can be found: `mosoma.m` that controls the run of the optimization, `crit.f.m` that evaluates the proposed solutions with objective functions and `NDsort1.m` that performs the non-dominated sorting of current set of proposed solutions as shown in Figure 5.3. The variables mentioned in this text that are used with the same name in the Matlab codes are marked with .

We describe here the run of MOSOMA on a two-objective problem having two input variables:

$$\begin{aligned} F_1 &= x_1, \\ F_2 &= \frac{1 + x_2}{x_1}. \end{aligned} \quad (5.6)$$

where x_1 can vary within the interval $\langle 0.1, 1.0 \rangle$ while x_2 within the interval $\langle 0.0, 5.0 \rangle$. Both the objective functions are to be minimized.

Settings of MOSOMA are as follows: initial population has $Q_1 = 5$ agents, migration proceeds in $ST = 3$ steps on the relative path length $PL = 1.3$, probability of perturbation is $PR = 0.1$ and minimal size of the external archive is $N_{ex,\min} = 3$. The number of migrating agents is set to $T = 3$. Two of migrating agents are taken from the initial

```

Start
  Define initial population of  $Q_1$  agents
  Compute objective functions in TMP
  Find external archive EXT
  Select  $T$  migrating agents
  While  $i < MIGS \mid FFC < N_{f,max} \mid |EXT(i)| < N_{ex,max}$ 
    For  $q = 1 : T$ 
       $\mathbf{x}_q$  migrates to all members of  $EXT(i - 1)$ 
      Compute objective functions in TMP
    End
    Find  $EXT(i)$  from  $TMP \cup EXT(i - 1)$ 
    While  $|EXT(i)| < N_{ex,min}$ 
      Find advancing front and crowding distance
      Fill  $EXT(i)$  with best agents from advancing front
    End
    Select  $T$  migrating agents for  $i + 1$ 
     $i++$ 
  End
  Choose final set  $P$  from  $EXT(i)$ 
End

```

Figure 5.6: The pseudocode of MOSOMA [82].

population ($T_1 = 2$) and position of the third agent is chosen randomly. The optimization process stops immediately if total number of migration loops $MIGS = 10$ proceeds, objective functions are computed 1000-times ($FFC = 1000$) or size of the external archive is 10-times higher than total number of expected non-dominated solutions $N_{ex,f}$.

We start with random generation of five agents according to equation (5.2). Following the Matlab script `mosoma.m`, the optimized variables are stored in matrix `AGENTS` (one row per variable, one column per agent). Two objectives are evaluated for all the agents executing the function `crit.f.m` and stored in the matrix `F` (one row per objective, one column per agent):

$$F = \text{crit.f}(\text{AGENTS});$$

The positions of the agents and corresponding objective function values are stored in Table 5.2.

Table 5.2: Positions of randomly generated agents and their objective functions values.

Agent	$x_1(-)$	$x_2(-)$	$f_1(-)$	$f_2(-)$
\mathbf{x}_1	0.60	3.80	0.60	8.00
\mathbf{x}_2	0.48	1.88	0.48	6.00
\mathbf{x}_3	0.25	0.25	0.25	5.00
\mathbf{x}_4	0.70	1.10	0.70	3.00
\mathbf{x}_5	0.78	4.46	0.78	7.00

Now, we have to determine the current external archive. Most members of this archive are the non-dominated solutions of the first order (see Figure 2) from the set described by `F`. The function `NDsort1.m` implements approach described in Figure 5.3 for finding the non-dominated set:

PSORTED = NDsort1(F(:,1:Q1));

Here, PSORTED are indices of the non-dominated solutions from current set of agents. If you consider the values of objective functions from Table 5.2 the non-dominated set is built by two solutions PSORTED = $\{\mathbf{x}_3, \mathbf{x}_4\}$. Both the agents are saved into the current external archive EXTARCH. Number of PSORTED elements is lower than minimal size of the external archive $N_{ex,min} = 3$. Therefore, the non-dominated sorting of the set F without agents \mathbf{x}_3 and \mathbf{x}_4 has to be performed to determine non-dominated set of the second order. It contains just a single solution \mathbf{x}_2 in case of our example which is saved into the EXTARCH as well. Now, the external archive has the expected minimal size and contains three agents: EXTARCH = $\{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_2\}$. In case that number of non-dominated solutions of the first order would be larger than $N_{ex,min}$, they would be all saved into the EXTARCH.

Further, the three migrating agents ($T = 3$) have to be selected. Two of them are randomly selected from the current population. Lets assume now, that agents \mathbf{x}_1 and \mathbf{x}_2 were chosen. Position of the third one is generated randomly using equation (5.2). In the Matlab code mosoma.m, positions of these three agents are saved into the variable MIGRATORS which has the same form as the variable AGENTS.

Now, all the selected agents migrate towards all members of the external archive. If the distance between two agents from the migrating pair is too low, the migrating agent travels towards randomly chosen position. Every migration proceeds in three steps ($ST = 3$). The temporary positions TMP are computed according to equation (5.3) and (5.5). The migration of solution \mathbf{x}_1 towards the first member of EXTARCH - agent \mathbf{x}_3 - is summarized in Table 5.3. Here, rnd_1 and rnd_2 are random numbers from interval $\langle 0; 1 \rangle$ that controls the perturbation process (see Figure 5.5) and TMP_1 , TMP_2 and TMP_3 are positions of the agents in the first, second and third step of the migration, respectively. If any rnd_n value is lower than probability of perturbation (in our case $PR = 0.10$) corresponding variable x_n remains the same. This happens in case of the second variable for temporary position TMP_2 in Table 5.3 ($rnd_2 = 0.08 < 0.10$) and therefore this variable remains $x_2(TMP_2) = 3.80$.

Table 5.3: Migration of agent \mathbf{x}_1 towards agent \mathbf{x}_3 .

Agent	$x_1(-)$	$x_2(-)$	$rnd_1(-)$	$rnd_2(-)$
\mathbf{x}_1	0.60	3.80	-	-
\mathbf{x}_3	0.48	1.88	-	-
TMP_1	0.45	2.97	0.82	0.24
TMP_2	0.30	2.13	0.71	0.08
TMP_3	-0.21	1.13	0.96	0.34

Using the same procedure, 27 temporary positions are obtained (3 migrating agents 3 members of the external archive 3 steps per every migration). All these positions have to be checked, if they remain in the feasible part of the decision space. For example in Table 5.3, the variable x_1 of the temporary position TMP_3 is -0.21 that is lower than the minimum limit for this variable ($x_{1,min} = 0.10$). Therefore, absorbing condition has to be applied and the value is set to the value $x_{1,min}$. Intuitively, if any temporary position value would be higher than maximum limit, it would be set to $x_{1,max}$.

After the boundary condition is applied, objective functions are computed for all the TMP positions. The new external archive is selected applying non-dominated sorting on a union of the objective values of TMP positions and previous external archive $\{\mathbf{TMP}_1, \mathbf{TMP}_2, \mathbf{TMP}_3\} \cup \{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_2\}$. Then, selected agents migrate towards all members of new external archive, objective values are evaluated in temporary positions and so on until any of the stopping conditions is fulfilled. After this procedure stops, the last task is to choose defined number $N_{ex,f}$ of final solutions REDUCT from current external archive. Objective function values of the final external archive are stored in variable FINAL. Usually, the final external archive contains much more members than $N_{ex,f}$ (if this is not true, we have to fill the rest of the REDUCT set with agents from advancing fronts or be content with lower number of Pareto-optimal solutions than $N_{ex,f}$).

Lets assume that MOSOMA stopped with the external archive having 7 agents. They are depicted as blue crosses in Figure 5.7. Their objective values are summarized in Table 5.4. Here, the whole set is in ascending order according to values of the first objective F_1 (index i is for the ordered set). The Euclidean distance d_i between the neighbors in the ordered set is computed. So, the total length of the found external archive d_{tot} can be computed as a sum of all d_i :

$$d_{tot} = \sum_{i=1}^{|EXT|-1} d_i \quad (5.7)$$

The distance between two ideally placed agents can be then computed:

$$d_{ideal} = \frac{d_{tot}}{N_{ex,f} - 1} \quad (5.8)$$

In our case, the total length of the found Pareto front is $d_{tot} = 9.08$ and distance between two ideal neighbors in the ordered set is $d_{ideal} = 2.27$. First, the best and worst agent according to F_1 is saved into the final set REDUCT. Now, we have $REDUCT = \{\mathbf{x}_3, \mathbf{x}_5\}$. Then, we have to choose three other agents. We always pick following agent from the sorted list until the sum of Euclidean distances between neighbors from the first one to the actual one (having index i) is greater than the ideal length (d_{ideal} multiplied by $i - 1$). When this is satisfied, the i -th agent is saved into the REDUCT set. If the i -th agent has been saved there already, the $(i + 1)$ -st is saved into the final set. Following this procedure for our example, the final set $REDUCT = \{\mathbf{x}_3, \mathbf{x}_5, \mathbf{x}_7, \mathbf{x}_4, \mathbf{x}_1\}$ will be formed. These solutions are marked with red plus signs in Figure 5.7. As can be seen here, the set REDUCT covers the true Pareto front uniformly.

Experiments

This subsection describes a design of two electromagnetic structures: a multilayer dielectric filter and a Yagi-Uda antenna. Both these problems have been solved with other optimization techniques. Therefore, results obtained by MOSOMA can be compared with independent references. All the tests were made so that MOSOMA computes objective function (the most time consuming part of the whole optimization) maximal same-times as it was set in other references to keep the comparisons fair. The significant differences between various global optimization tools presented here may be caused by the fact that both the solved problems have very large and complicated decision space and the time

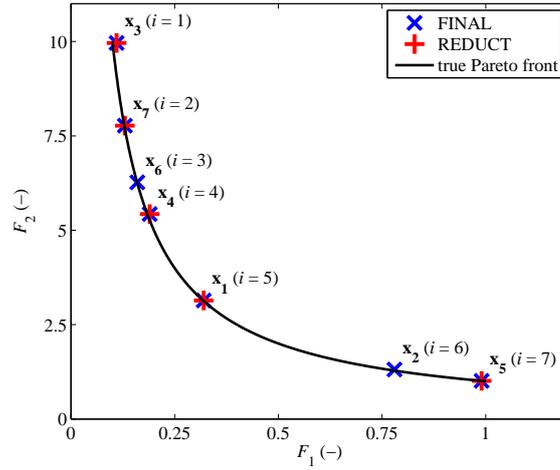


Figure 5.7: Choice of the final non-dominated set (FINAL) from the external archive (FINAL).

Table 5.4: The migration of agent 1x towards agent 3 x : experiments.

i	$F_1(-)$	$F_2(-)$	agent	$d_i(-)$
1	0.11	0.96	\mathbf{x}_3	2.19
2	0.13	7.77	\mathbf{x}_7	1.50
3	0.16	6.27	\mathbf{x}_6	0.84
4	0.19	5.43	\mathbf{x}_4	2.29
5	0.32	3.14	\mathbf{x}_1	1.89
6	0.78	1.31	\mathbf{x}_2	0.37
7	0.99	1.01	\mathbf{x}_5	-

devoted for the optimization was not satisfactory to reach the global optimum. Main problems faced during the optimization process (implementation of constraints, discrete input variables etc.) and their solution will be discussed here.

Multilayer dielectric filter

Design of a dielectric filter for microwave frequency bands involves an optimization of a relative permittivity and a width of individual layers of the filter. Considering the filter having N layers, $2N$ parameters are changed during the optimization process. The layered medium is depicted in Figure 5.8. Here, \mathbf{k}_0 stands for the wave vector of the impinging wave, l_n denotes width of n -th layer, $\epsilon_{r,n}$ denotes relative permittivity of n -th layer, α_n is the incident angle for n -th interface and R_n is the reflection coefficient of n -th interface. Interface between the first and second dielectric layer is denoted by R_2 .

Considering homogeneous lossless nonmagnetic materials ($\sigma = 0$, $\mu_r = 1$), generalized recursive reflection coefficient R_n for n -th interface can be derived [32]:

$$R_n = \frac{r_n + R_{n+1} \exp(2j\mathbf{k}_n l_n)}{1 + r_n R_{n+1} \exp(2j\mathbf{k}_n l_n)} \quad (5.9)$$

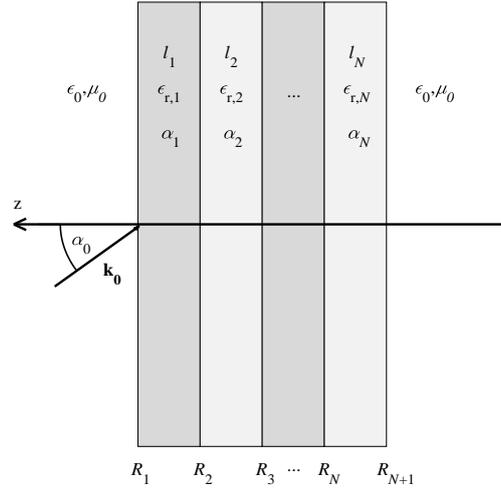


Figure 5.8: Description of the layered medium [82].

where the wave vector in the n-th layer can be computed using an equation [32]:

$$\mathbf{k}_n = \frac{2\pi f}{c} \sqrt{\epsilon_{r,n}} \quad (5.10)$$

After few simplifications the reflection coefficient can be derived for TE mode:

$$r_{n,TE} = \frac{\sqrt{\epsilon_{r,n-1}(1 - \sin^2 \alpha_{n-1})} - \sqrt{\epsilon_{r,n}(1 - \sin^2 \alpha_n)}}{\sqrt{\epsilon_{r,n-1}(1 - \sin^2 \alpha_{n-1})} + \sqrt{\epsilon_{r,n}(1 - \sin^2 \alpha_n)}} \quad (5.11)$$

and for TM mode:

$$r_{n,TM} = \frac{\sqrt{\epsilon_{r,n}(\epsilon_{r,n-1} - \sin^2 \alpha_{n-1})} - \sqrt{\epsilon_{r,n-1}(\epsilon_{r,n} - \sin^2 \alpha_n)}}{\sqrt{\epsilon_{r,n}(\epsilon_{r,n-1} - \sin^2 \alpha_{n-1})} + \sqrt{\epsilon_{r,n-1}(\epsilon_{r,n} - \sin^2 \alpha_n)}} \quad (5.12)$$

where the angle of incidence for the n-th layer is defined by:

$$\alpha_n = \sin^{-1} \left(\frac{\sqrt{\epsilon_{r,n-1}}}{\sqrt{\epsilon_{r,n}}} \sin \alpha_{n-1} \right) \quad (5.13)$$

The coefficient between the free space and the first medium denoted R_1 expresses then the reflection properties of the filter.

In [62, 191], two objective functions for the design of a filter with seven layers were defined:

$$\begin{aligned} F_1 &= \sum_{p=1}^P [|R_{1,TE}(f_p)|^2 + |R_{1,TM}(f_p)|^2], \\ F_2 &= \sum_{s=1}^S [2 - |R_{1,TE}(f_s)|^2 + |R_{1,TM}(f_s)|^2], \end{aligned} \quad (5.14)$$

where f_p and f_s denote the pass and stop frequencies of the filter, respectively, and P and S stands for the size of used frequency vectors. The objective function F_1 minimizes the

reflection of the layered media in the pass band while the other function F_2 maximizes the reflection in the stop band. Under this definition, Pareto fronts obtained by different authors cannot be compared because values of objective functions are influenced by the discretization of the frequency axis. Therefore, we propose a slight modification of the objective functions:

$$\begin{aligned} F_1 &= \frac{1}{P} \sum_{p=1}^P [|R_{1,TE}(f_p)|^2 + |R_{1,TM}(f_p)|^2], \\ F_2 &= \frac{1}{S} \sum_{s=1}^S [2 - |R_{1,TE}(f_s)|^2 + |R_{1,TM}(f_s)|^2], \end{aligned} \quad (5.15)$$

Now, both the functions are normalized to the number of examined frequency points and are fully comparable.

The definition of the optimization problem is fully completed by the formulation of the constraint functions for the pass band and the stop band [191]:

$$\begin{aligned} 20 \log |R_{1,TE}(f_{pc})| &< -10 \text{ dB}, \\ 20 \log |R_{1,TM}(f_{pc})| &< -10 \text{ dB}, \\ 20 \log |R_{1,TE}(f_{sc})| &< -5 \text{ dB}, \\ 20 \log |R_{1,TM}(f_{sc})| &< -5 \text{ dB}, \end{aligned} \quad (5.16)$$

where f_{pc} and f_{sc} denote the pass and stop frequencies considered for constraints, respectively. The constraint limits were selected exactly as in [191] to keep the comparison between MOSOMA and other references fair.

So called penalty function approach [133] can be used for handling with constraints. Violation of any constraint has an impact on worsening the objective functions. First, violation V_j from the normalized j -th constraint function g_j for the i -th solution \mathbf{x}_i is calculated:

$$V_j(\mathbf{x}_i) = \begin{cases} |g_{j,n}(\mathbf{x}_i)| & \text{if } g_{j,n}(\mathbf{x}_i) < 0 \\ 0 & \text{otherwise} \end{cases} \quad (5.17)$$

Thereafter, objective functions $F_{c,m}$ considering the violations from all constraint functions are slightly modified to:

$$F_{c,m}(\mathbf{x}_i) = F_m(\mathbf{x}_i) + R_m \sum_{j=1}^J V_j(\mathbf{x}_i) \quad (5.18)$$

where R_m stands for a penalty parameter, which is introduced to equalize magnitudes of both addends in the equation (5.18). Since values of both the objective functions should vary in the interval $0; 2$ the penalty operator was set for both the objective functions to $R = 5$. This procedure disqualifies the solutions violating any constraint from further search of the algorithm.

The design of the seven-layer filter evolves the optimization of 14 parameters. The incidence angle was fixed to $\alpha_0 = \pi/4$ [191]. The width of every layer x_{1-7} can vary in the interval $\langle 1; 10 \rangle$ mm. The relative permittivity of all layers can be chosen from commercially

available dielectric materials $\{1.01, 2.20, 2.33, 2.50, 2.94, 3.00, 3.02, 3.27, 3.38, 4.48, 4.50, 6.00, 6.15, 9.20, 10.20\}$ [191]. MOSOMA works initially only with continuous input variables. The problem with discrete input variables can be solved using a relatively simple approach. As we have 15 discrete values, the input variables x_{8-14} are set from the interval $\langle 0; 15 \rangle$. This interval is divided uniformly into 15 subintervals, each corresponding to one value of an available dielectric permittivity (e.g. value $x_8 = 6.35$ corresponds to the seventh value from the list 3.02). The variables are within the algorithm treated as continuous. Only objective functions are evaluated with the corresponding value of the relative permittivity.

This procedure brings obviously some shortcomings. If both the agents that participate on the migration have similar values of the input variable, all steps of the migration can cause, that the continuous variable does not leave the original subinterval and the same permittivity is examined again. Another problem is caused by the fact that different values of the input variable means one value of the relative permittivity. Then, the result of the migration depends on the initial value of the migrating agent in the subinterval. Let us consider the equation (5.3) with no influence of perturbation and parameters $PL = 1.3$ and $ST = 3$ and two different migrating agents having only one variable $x_q = 0.1$ and $x_q^* = 0.99$. Now, let them migrate towards the member of the external archive $x_p = 3.5$. The resulting steps of the migration correspond to different dielectric materials $\mathbf{TMP} = \{1.62, 3.13, 4.65\}$ and $\mathbf{TMP}^* = \{2.51, 4.02, 5.54\}$. Beside all the shortcomings, the algorithm is able to solve problems with continuous and discrete variables simultaneously without any change of MOSOMA program. The only change comprises the evaluation of objective functions - discrete values are used according to subintervals of the input variable.

The controlling parameters of MOSOMA were set such a way so that the results can be compared with results published in [62, 191]. The settings are summarized in Table 5.5.

Table 5.5: Settings of MOSOMA parameters for the dielectric filter design.

Par.	FFC	PL	ST	Q_1	T	$N_{ex,min}$
-	15000	1.3	5	30	20	15

Band-pass filter

The first experiment is aimed to design a band-pass filter for frequencies from 28 GHz up to 32 GHz. The frequency bands for the filter and for the constraint functions are summarized in Table 5.6.

Table 5.6: Frequency bands for the band-pass filter optimization.

Band	Lower b. (GHz)	Upper b. (GHz)
f_p	28	32
f_s	24; 32	28; 36
f_{p^c}	29	31
f_{s^c}	24; 34	26; 36

The Pareto front of the optimized problem is depicted in Figure 5.9. Obviously, some of the Pareto-optimal solutions are violating the constraint functions, because their value of the objective function is higher than 2. Three solutions are highlighted here: the best

solution according to the first objective (red marker) and the second (green) objective and the trade-off solution (blue). Figure 5.10 depicts the frequency behavior of the reflection coefficients for these solutions. Here, colors correspond to markers in Figure 5.9. The red solution ideally satisfies the first objective, but the last two constraints are violated. On the contrary, the green solution meets the second objective but violates first two constraint functions. Finally, the blue solution respects both the objectives and does not violate any constraint function.

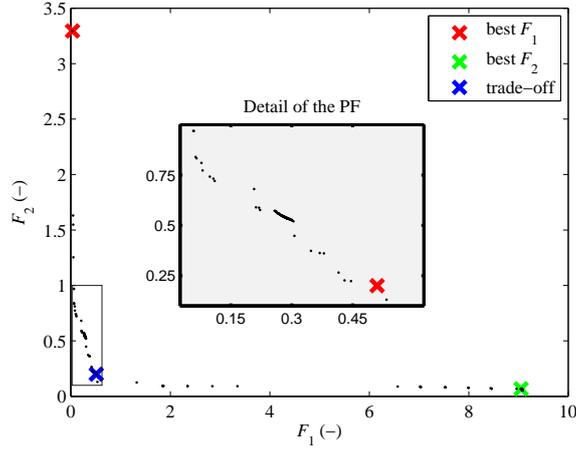


Figure 5.9: Pareto front of the band-pass filter multi-objective optimization using MO-SOMA. The detailed plot depicts the trade-off solutions non-violating the constraints.

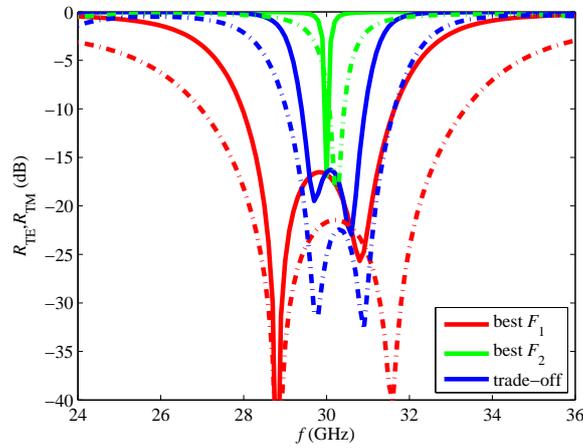


Figure 5.10: Reflection coefficient TE (solid line) and TM (dashed) for three band-pass filters designed by MOSOMA: red line (the best solution according to F_1), green (the best F_2) and blue (trade-off).

The trade-off solution composed of layers having the widths $\{4.686, 1.995, 4.739, 1.001, 1.003, 1.002, 8.663\}$ mm and relative permittivities $\{10.20, 1.01, 10.20, 1.01, 1.01, 2.94, 2.35\}$ was chosen as the final trade-off solution. Figure 5.11 compares its reflection coefficients with solutions published in [62, 191]. The total width of our design is 23.08 mm compared to 33.44 mm [191] and 21.35 mm [62]. The reflection coefficient for our solution remains

below -16 dB for the TE mode and -22 dB for the TM mode in the operational band of the filter. Coefficients R_{TE} and R_{TM} of our design decrease steeper at the boundaries of the desired frequency band. Results obtained by MOSOMA seem to be better than results obtained by other global algorithms published in [62, 191], although the same number of objective function evaluations was made. It shows that MOSOMA converges faster in this case.

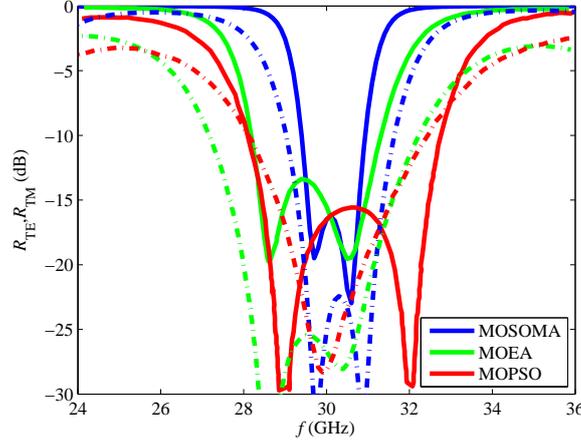


Figure 5.11: Comparison of the TE (solid line) and TM (dashed) reflection coefficient for band-pass filter design obtained by MOSOMA (blue), MOEA [191] (green) and MOPSO [62] (red).

Low-pass filter

The frequency bands for the low-pass filter operating up to 30 GHz and corresponding to constraint functions frequency bands are summarized in Table 5.7.

Table 5.7: Frequency bands for the low-pass filter optimization.

Band	Lower b. (GHz)	Upper b. (GHz)
f_p	24	30
f_s	30	36
f_{pC}	24	28
f_{sC}	32	36

Figure 5.12 depicts the Pareto front obtained by MOSOMA. Again, best solutions according to the F_1 and F_2 objectives are marked with the red and green color, respectively. Solutions non-violating constraints are emphasized in the detailed subplot. The solution chosen for the final design is marked with the blue cross here. This solution is composed of layers having widths $\{8.195, 1.489, 1.758, 1.001, 1.001, 1.153, 1.003\}$ mm and relative permittivities $\{2.20, 1.01, 10.20, 3.02, 1.01, 10.20, 6.15\}$. Figure 5.13 compares then the distinguished solutions from Pareto front at Figure 5.12. We can see, that the red and blue solutions are similar for the passing band (F_1 is just slightly better for the red one) while the blue one is significantly better in the stopping band. Further, we can see that the solution best according to F_2 reflects almost in the whole frequency band.

Frequency response of the reflection coefficient for the trade-off solution is depicted again in Figure 5.14. The frequency response is compared here with solutions obtained

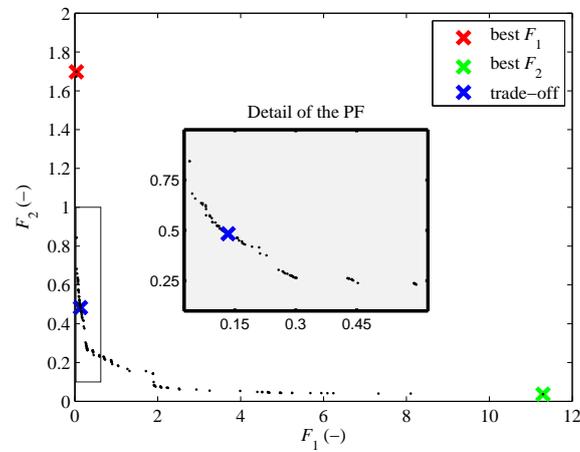


Figure 5.12: Pareto front of the low-pass filter multi-objective optimization using MO-SOMA. The detailed plot depicts the trade-off solutions non-violating the constraints with highlighted solution chosen for the design.

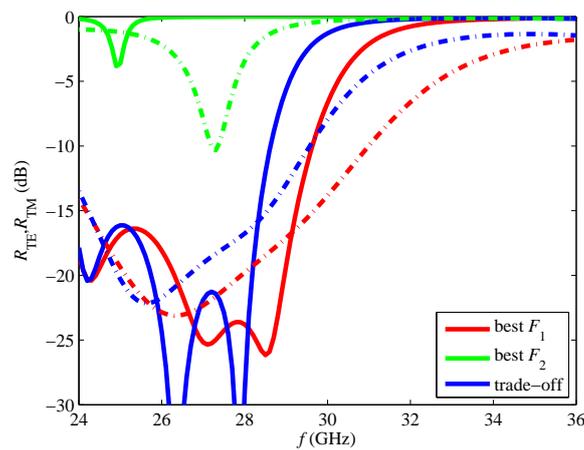


Figure 5.13: Reflection coefficient TE (solid line) and TM (dashed) for three low-pass filters designed by MOSOMA: red line (the best solution according to F_1), green (the best F_2) and blue (trade-off).

by algorithms MOEA [191] and MOPSO [62]. The reflection is in the pass band approximately under -20 dB for the TE mode and -18 dB for the TM mode. Obviously, our solution is better than the solution proposed in [191], because the reflection is lower in the pass band and higher in the stop band. The solution proposed in [62] exhibits similar reflection properties, but the total width of the filter is 16.24 mm while the total width of our filter is 15.58 mm only.

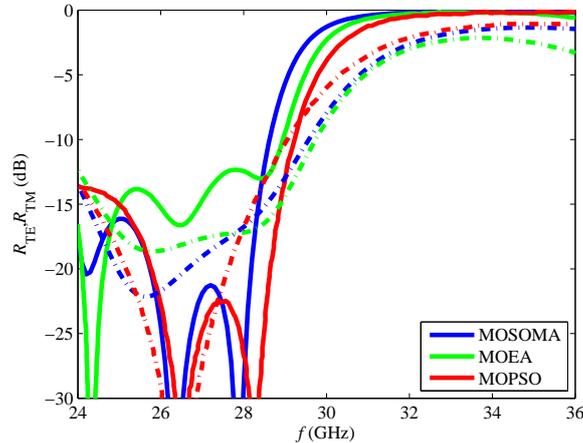


Figure 5.14: Comparison of the TE (solid line) and TM (dashed) reflection coefficient for low-pass filter design obtained by MOSOMA (blue), MOEA [191] (green) and MOPSO [62] (red).

Band-stop filter

The next example considers the design of the band-stop filter for the frequency band from 28 GHz to 32 GHz. The frequency bounds for the filtering properties and constraint functions are summarized in Table 5.8.

We have selected the solution composed from dielectric layers having widths $\{3.090, 5.358, 1.001, 1.000, 5.585, 1.000, 6.020\}$ mm and relative permittivities $\{2.50, 3.38, 10.20, 2.33, 3.38, 10.20, 2.50\}$. This solution is marked in Figure 5.15 of the Pareto front obtained by MOSOMA. Obviously, constraints for the optimization are too strict which causes the increase of the objective function values of the Pareto-optimal solutions.

Table 5.8: Frequency bands for the band-stop filter optimization.

Band	Lower b. (GHz)	Upper b. (GHz)
f_p	24	30
f_s	30	36
f_{pC}	24	28
f_{sC}	32	36

Figure 5.16 compares reflection properties of three distinguished solution: best according to F_1 (red), best according to F_2 (green) and the trade-off (blue). It is obvious, that both the extreme solutions consider just one of the objectives: the red one passes in the whole frequency band (just F_1 is considered) while the green one stops. Finally, the blue one considers both the objectives at the same time.

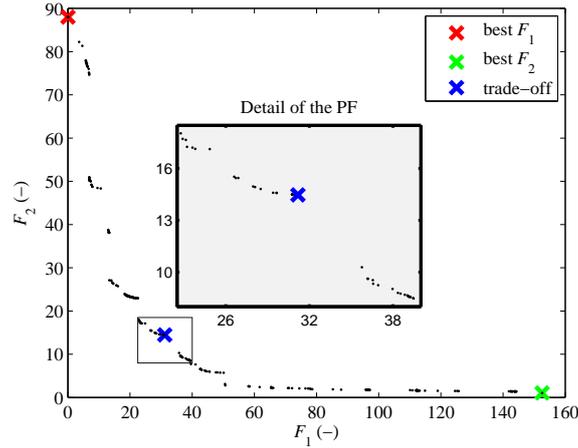


Figure 5.15: Pareto front of the band-stop filter multi-objective optimization using MO-SOMA. The detailed plot depicts the trade-off with highlighted solution chosen for the final design.

The reflection coefficient of this trade-off solution is then compared with solutions published in [62, 191] in Figure 5.16. The reflection coefficient is very high in the whole stop band and falls under -15 dB in both parts of the pass band for both the modes. Reflection properties are comparable for all three algorithms. The total length of our filter is 23.05 mm compared to 29.25 mm in [191] and 20.53 mm in [62].

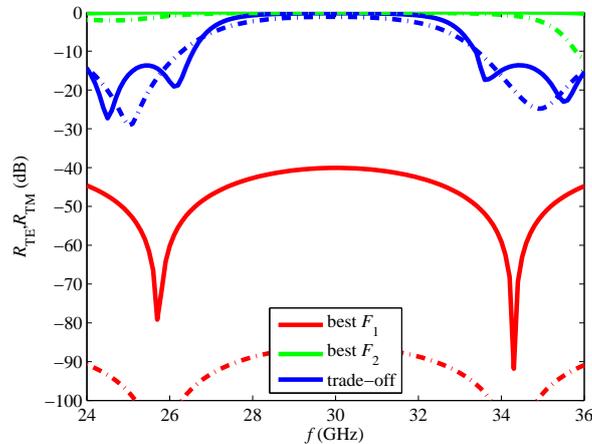


Figure 5.16: Reflection coefficient TE (solid line) and TM (dashed) for three band-stop filters designed by MOSOMA: red line (the best solution according to F_1), green (the best F_2) and blue (trade-off).

Yagi-Uda antenna

The design of the Yagi-Uda antenna requires the optimization of lengths of individual elements and spacing between them. The Yagi-Uda antenna is depicted in Figure 5.18. Here, d_n stands for the total length of the n -th element and s_n denotes the spacing between the n -th and $(n + 1)$ -st element. Considering N elements, $2N - 1$ parameters

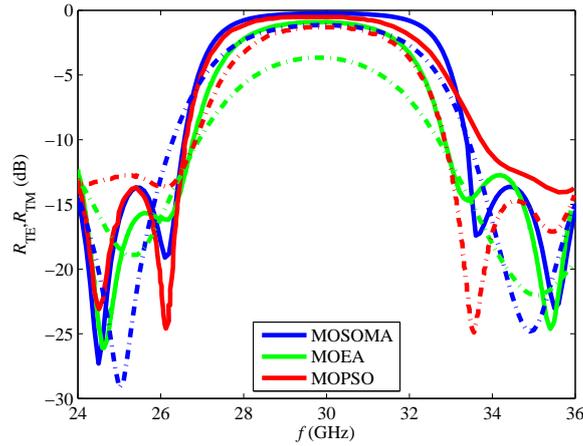


Figure 5.17: Comparison of the TE (solid line) and TM (dashed) reflection coefficient for band-stop filter design obtained by MOSOMA (blue), MOEA [191] (green) and MOPSO [62] (red).

are optimized. Operating frequency was set to 30 GHz as indicated on Figure 5.19. But all antenna dimensions are presented with regards to the corresponding free space wavelength λ . The width of every element can vary in the interval $\langle 0.30; 0.70 \rangle$ while the spacing between neighboring elements can vary within the interval $\langle 0.10; 0.35 \rangle$.

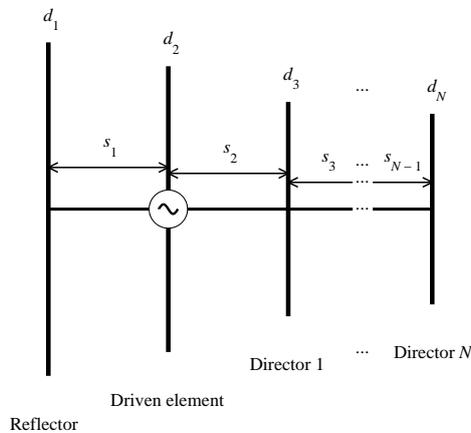


Figure 5.18: Description of the N-element Yagi-Uda antenna with description of optimized parameters.

The settings of MOSOMA was made so that it computes objective functions maximal 36000-times and our results can be compared with results published in [18, 79, 110, 190]. The controlling parameters are summarized in Table 5.9.

Two objectives are considered for the optimization of the Yagi-Uda antenna the maximal gain and the minimal relative side lobe level [190]:

$$\begin{aligned} F_1 &= -G, \\ F_2 &= SLL, \end{aligned} \tag{5.19}$$

```

CM YAGI - UDA
CE 4 elements % description of problem
GW 1,7,0,-0.5,0,0,0.5,0,1e-003 % description of wires
GW 2,7,0.5,-0.5,0,0.5,0.5,0,1e-003
GW 3,7,1,-0.5,0,1,0.5,0,1e-003
GW 4,7,1.5,-0.5,0,1.5,0.5,0,1e-003
% ID, # segments, X1, Y1, Z1, X2, Y2, Z2, radius
GE 0 % end of geometry description
GN -1 % no ground plane
EK % use of extended thin wire model
EX 6,2,4,0,1,1e-006,1,0,0 % excitation
% type of source, ID of exc. wire, exc. segment, optional,
real, imag, magn., phase
FR 0,1,0,0,3e+004,0 % examined frequencies in MHz
RP 0,72,1,1003,-180,0,5,5 % radiation pattern request
% normal mode, # theta, # phi, total gain request, start
theta, start phi, step theta, step phi
EN % end of description

```

Figure 5.19: Example of 4NEC2 input file for four-element Yagi-Uda antenna analysis.

Table 5.9: Settings of MOSOMA parameters for design of Yagi-Uda antenna.

Par.	FFC	PL	ST	Q_1	T	$N_{ex,min}$
-	36000	1.3	5	30	20	15

where G denotes the gain and SLL stands for the relative side lobe level of the antenna. Two constraint functions are defined to ensure a proper impedance matching of the designed antenna [31]:

$$\begin{aligned}
|50 - \Re Z_{in}| &< 5, \\
|\Im Z_{in}| &< 10,
\end{aligned} \tag{5.20}$$

where Z_{in} is the input impedance of the antenna.

The analysis necessary for the computation of objectives was performed by software 4NEC2 based on the method of moments [25]. The software is freely available at the website: <http://home.ict.nl/arivoors/>. The interconnection between the Matlab script that provides the optimization of input parameters and the 4NEC2 software is described in Figure 5.20. Information about the analyzed structure has to be saved in an ASCII text file input.nec in an appropriate format defined in [160]. An example for a four-element Yagi-Uda antenna with the description of individual entries can be viewed in Figure 5.19. The software 4NEC2 can be executed from Matlab in a non-interactive mode using the expression:

```
! ... \4nec2.exe ... \input.nec -I
```

The results can be read from ouput.out text file generated by 4NEC2 software in the folder ... 4nec2\out.

Four-element design

Radius of all the wires was set to 0.00225λ [79]. The best Pareto front from ten runs of MOSOMA is depicted in Figure 5.21. Here, Pareto-optimal solutions are compared with solutions obtained by MOEA published in [79]. MOSOMA algorithm achieved significantly better results because most of the solutions proposed in [79] are dominated by

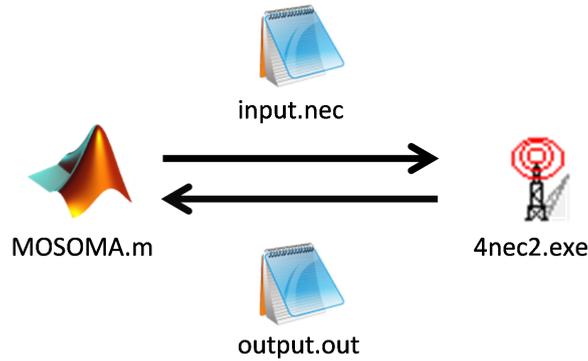


Figure 5.20: Connection between optimization script in Matlab and analysis tool 4NEC2.

the solutions obtained by MOSOMA, except of few solutions in a region of the minimum of the second objective.

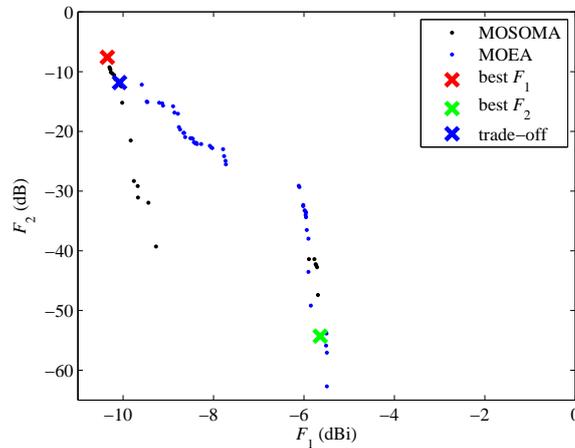


Figure 5.21: Pareto front of four-element Yagi-Uda antenna design obtained by MOSOMA (black dots) and MOEA 0 (blue dots).

The distinguished solutions from the obtained Pareto front are highlighted in Figure 5.21. Their radiation patterns are depicted with corresponding colors in Figure 5.22. The solution depicted in red color represents the maximal gain $G = 10.35$ dBi. The green solution corresponds to the minimal level of side lobes ($SLL = -54.29$ dB), but exhibits a very poor gain $G = 5.64$ dBi. The chosen trade-off solution shows a very good gain $G = 10.08$ dBi and a satisfactory side lobe level $SLL = -11.81$ dB.

Parameters of designs depicted in Figure 5.22 are summarized in Table 5.10. Results obtained by MOSOMA are compared here with the designs proposed in [18, 79, 190]. The design of the Yagi-Uda antenna was treated as a pure multi-objective problem only in [110, 190], but in [110], the impedance matching is considered as the third objective function, which disables us to compare these results with ours. In [18, 79], two objectives were aggregated to build one fitness function, which was then solved by single-objective modifications of genetic algorithms and particle swarm optimization, respectively. Any of the solutions from all the references is not better in all considered objectives than our final trade-off solution.

Table 5.10: Four-element Yagi-Uda antenna design parameters obtained by MOSOMA, MOEA, GA and CLPSO algorithms.

Algorithm	MOSOMA				MOEA [190]				GA [79]		CLPSO [18]			
	best F_1	best F_2	trade-off	best F_1	best F_2	aggregated	aggregated	aggregated	aggregated					
	$d(\lambda)$	$s(\lambda)$												
1	0.473	0.313	0.494	0.245	0.474	0.295	0.480	0.270	0.628	0.204	0.490	0.283	0.476	0.311
2	0.445	0.350	0.469	0.196	0.463	0.260	0.474	0.186	0.488	0.195	0.472	0.179	0.466	0.205
3	0.439	0.288	0.402	0.272	0.440	0.249	0.436	0.274	0.436	0.114	0.442	0.279	0.434	0.279
4	0.434	-	0.651	-	0.433	-	0.434	-	0.582	-	0.424	-	0.412	-
G (dBi)	10.35		5.64		10.08		9.6		5.5		9.84		9.44	
SL (dB)	-7.6		-54.29		-11.81		-12.14		-62.6		-14.5		-15.02	
Z_{in} (\backslash Omega)	45.18 + 6.96i		45.32 - 1.70i		45.36 + 3.65i		47.59 - 5.67i		45.51 + 7.83i		38.50 - 2.30i		49.56+0.11i	
$VSWR$ (-)	1.19		1.11		1.13		1.13		1.21		1.3		1.01	

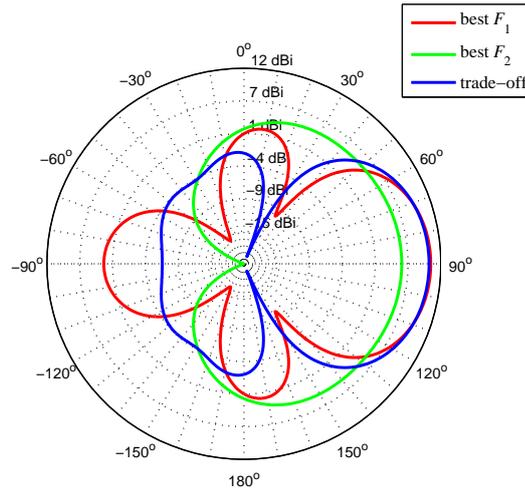


Figure 5.22: H-plane radiation patterns of chosen solutions from Pareto front obtained by MOSOMA for four-element Yagi-Uda antenna.

Six-element design

The six-element design has the same settings as the previous four-element problem. Only the radius of the wire was increased to 0.003369λ [79]. Pareto-optimal solutions obtained by MOSOMA are depicted in Figure 5.23. Here, the extreme solutions according to both the objectives (the maximal gain $G = 12.67$ dBi, the maximal side lobe level suppression $SLL = -38.84$ dB) are highlighted with red and green markers, respectively. One trade-off solution has been also chosen ($G = 12.65$ dBi, $SLL = -12.66$ dB). Radiation patterns of those three designs are depicted in Figure 5.24. Obviously, the radiation patterns of the trade-off solution and the best solution according to F_1 are almost overlapping, but the trade-off solution has a slightly better side lobe level and significantly better impedance matching.

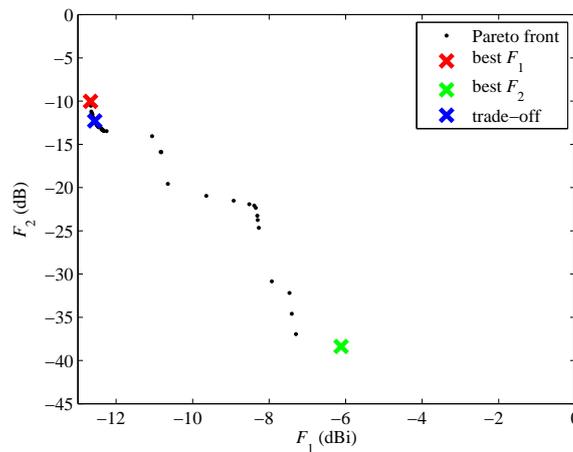


Figure 5.23: Pareto front of six-element Yagi-Uda antenna design obtained by MOSOMA (black dots) with highlighted solutions from Table 5.11.

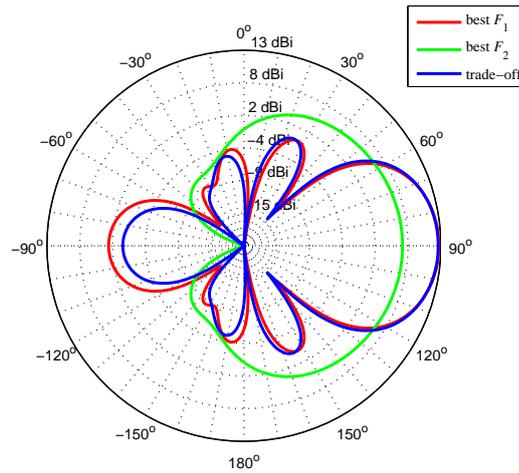


Figure 5.24: H-plane radiation patterns of chosen solutions from Pareto front obtained by MOSOMA for six-element Yagi-Uda antenna optimization.

Conclusions

MOSOMA is a very efficient multi-objective optimizer. MOSOMA can cope both with continuous and discrete parameters at the same time. Also the constraint functions can be considered for the solved optimization problem. The implementation of MOSOMA is easier than genetic algorithms (no binary coding of optimized parameters is needed) and very similar to particle swarm optimization. In case of investigated electromagnetic structures, MOSOMA shows a better (or comparable) performance as other commonly used multi-objective optimizers.

MOSOMA was successfully applied to the design of dielectric filters and Yagi-Uda antennas. MOSOMA has significantly outperformed the algorithm published in [190] in four-element Yagi-Uda antenna multi-objective design. In comparison with aggregation methods that convert a multi-objective problem into a single-objective one, MOSOMA brings additional information about limits of the considered objectives. The trade-off solution chosen from the revealed Pareto front is better or comparable in all considered objectives.

Acknowledgments

Research described in this paper was financially supported by Czech Science Foundation under grant no. P102/12/1274. Support of projects CZ.1.07/2.3.00/20.0007 and CZ.1.05/2.1.00/03.0072 is also gratefully acknowledged.

5.2 Thin-Film Screen Time-Domain Shielding Effectiveness: Multi-Objective Optimization of the Testing Pulse

Originally published as:

Kadlec, P., Marek, M., Štumpf, M.: Thin-Film Screen Time-Domain Shielding Effectiveness: Multi-Objective Optimization of the Testing Pulse. In *2020 Proceedings of the 2020 International Symposium on Electromagnetic Compatibility (EMC Europe 2020)*, 1-5, 2020.

Abstract

In this paper we analyze the shielding effectiveness of a thin-film screen in the time domain (TD). The excitation pulse is optimized with respect to various definitions of the TD shielding effectiveness (SE) concerning the peak value, the maximal derivative and the absorbed energy of the incident and shielded signals. A general approach to set parameters of the power exponential pulse as a key component of the test (namely the rise time, the rising power and the ringing frequency) with respect to different requirements on the device under test is proposed here. Two sample problems are solved using a multi-objective particle swarm optimization algorithm. Results show the theoretical level of the test adjustments and the influence of the individual pulse parameters to the SE.

Introduction

The Shielding Effectiveness (SE) belongs to the most important EMC parameters of thin-film materials. This parameter expresses the ability of the material to reflect, absorb or redirect the unwanted radiation. In most studies, SE is defined in the frequency domain. However, in reality, electronic devices are exposed to transient disturbances rather than harmonic ones. Therefore, the TD SE parameters were defined in [13,28,134]. A statistical approach combining more observation points for the SE was introduced in [12].

As discussed in [28], some devices are more susceptible to different properties of the disturbing signal:

1. the peak value of the electric (magnetic) field,
2. the time variations of the magnetic (electric) flux density,
3. the total energy delivered.

The susceptibility of a device to different parameters of the disturbance signal should be taken into account to ensure the correct choice of a shielding material. Accordingly, a testing system should be tailored to incorporate the most affecting parameter of the disturbing signal for the device under test.

The TD shielding properties of thin-film materials can be analyzed by means of various numerical methods. The transmission-line modeling method [97], the finite difference TD method [29] and the finite integration technique [124] are among the most popular ones. These full wave methods suffer from enormous computational demands. Therefore, a number of analytical methods for the TD shielding effectiveness have been developed in

recent years. A planar conductive shield in the presence of vertical dipole field sources is solved by a modified Cagniard technique in [24]. Shielding properties of graphene materials are examined in [122] by the same technique. An analytical description of the loop-to-loop signal transfer across a thin metal screen with Drude-type dispersive behavior is achieved in [178] using the Cagniard-DeHoop (CdH) technique [42].

In this paper we use the CdH method [178] to examine the theoretical limits of individual TD shielding effectiveness properties (the peak signal level, the time variations of the signal and the total energy of the signal) for a specific thin-film material. In the analysis we employ the unipolar power exponential pulse [43] for the field excitation. A multi-objective optimization [45, Ch. 2] of the excitation pulse is the main contribution of the paper.

The organization of the paper is as follows: Section 5.2 provides the problem description including the definition of the TD SE parameters. Section 5.2 reviews the parameters of the power exponential pulse as it is the main subject of the research. Two numerical examples for a selected thin-film material are presented in Section 5.2. Finally, Section 5.2 concludes the paper.

Problem Description

The problem description is shown in Fig. 5.25. The problem is defined in a standard right-handed Cartesian coordinate system with the origin \mathcal{O} and (standard) base vectors $\{\mathbf{i}_x, \mathbf{i}_y, \mathbf{i}_z\}$. Consequently, a point in the problem configuration can be defined by the coordinates $\{x, y, z\}$. An infinite metal sheet that is described by its thickness δ and electrical conductivity σ_S is placed perpendicular to \mathbf{i}_z at $z = 0$. The sheet is surrounded by the isotropic, homogeneous and lossless medium described by ϵ_0 and μ_0 . Thus, the EM wave propagates here with the speed $c_0 = (\epsilon_0\mu_0)^{-1/2} > 0$.

There are two loop antennas both parallel to the metal sheet. Symbol \mathcal{L}_T denotes the transmitting antenna located at $\{0, 0, h_T > 0\}$ with an area \mathcal{A}_T and \mathcal{L}_R is the receiving antenna located at $\{0, 0, h_R < 0\}$ with an area \mathcal{A}_R . The system is excited by a planar pulsed electric current $I_T(t)$ in \mathcal{L}_T , while the open-circuit electric voltage $V_R(t)$ is induced in \mathcal{L}_R :

$$V_R(t) = Z(t) \overset{(t)}{*} I_T(t) \quad (5.21)$$

where $Z(t)$ denotes the transfer impedance at time t and $\overset{(t)}{*}$ denotes the time convolution. According to [178], the voltage for the scenario described in Fig. 5.25 is computed according to:

$$V_R(t) = -\frac{\mathcal{A}_T\mathcal{A}_R}{(h_T + h_R)^4} \frac{\partial_t^4 I_T(t)}{2\pi\eta_0} \overset{(t)}{*} \frac{t(t^2 - T^2)}{G_L/\eta_0 + 2t/T} \text{H}(t - T) \quad (5.22)$$

where h_T and h_R denote the distance from the screen to the transmitting and receiving antenna, respectively. Symbol ∂_t denotes the time derivative, η_0 is the free-space admittance $\eta_0 = (\epsilon_0/\mu_0)^{1/2}$, $T = (h_T + h_R)/c_0$ is the travel time of the pulse, and $\text{H}(t)$ is the Heaviside unit step function. Finally, the conductance G_L of the screen is $G_L = \delta\sigma_S$.

The TD shielding effectiveness parameters are defined in [13]. The SE parameters are defined there based on the electric or magnetic field values. The SE as considered in the present work, applies to the vertical component of the magnetic field through the voltage induced in the receiving antenna \mathcal{L}_R (see Eq. (6.19)). The signal is measured

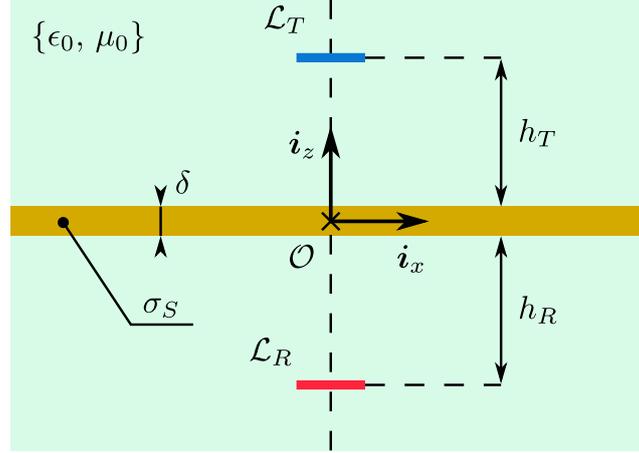


Figure 5.25: Problem configuration: the thin-film metal sheet with the transmitting and receiving antennas \mathcal{L}_T and \mathcal{L}_R , respectively.

in the absence of (denoted by the superscript *in* as “incident”) and in the presence of (superscript *s* as “shielded”) the thin-film screen.

The first parameter is based on the peak reduction of the electric field. It is called as the *peak value reduction shielding effectiveness*:

$$SE_{PR} = 20 \log \frac{V_{R,\max}^{in}(x, y, z, t)}{V_{R,\max}^s(x, y, z, t)}, \quad (5.23)$$

where $V_{R,\max}^{in}$ and $V_{R,\max}^s$ denote the maximal value of the induced voltage observed within a finite time window and at the position of the receiving antenna \mathcal{L}_R .

The next monitored parameter is the reduction of the derivative of the EM transient field in the shielded region. Hence, the *derivative reduction shielding effectiveness* is defined:

$$SE_{DR} = 20 \log \frac{d_t V_{\max}^{in}(x, y, z, t)}{d_t V_{\max}^s(x, y, z, t)}, \quad (5.24)$$

where $d_t V$ denotes the time-derivative of the induced voltage:

$$d_t V(x, y, z, t) = \partial_t V_R(x, y, z, t). \quad (5.25)$$

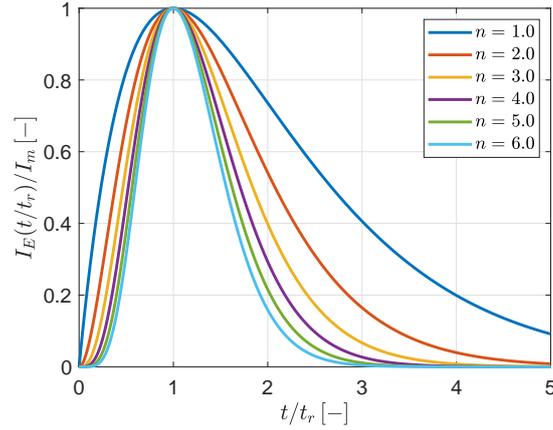
The last parameter is based on the reduction of the energy shielded by the thin-film screen. Thus, the *energy reduction shielding effectiveness* is:

$$SE_{ER} = 10 \log \frac{\mathcal{E}^{in}(x, y, z, t)}{\mathcal{E}^s(x, y, z, t)}, \quad (5.26)$$

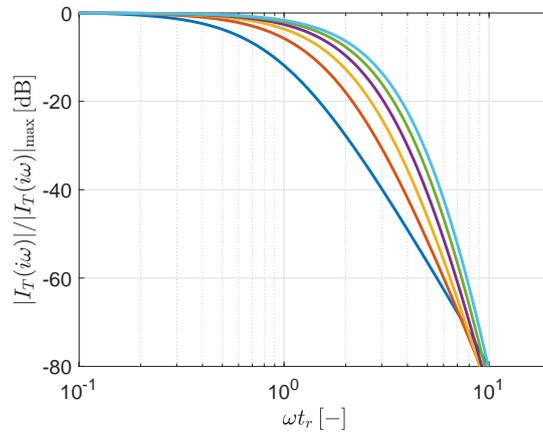
where \mathcal{E} stands for the energy of the sampled signal:

$$\mathcal{E}(x, y, z, t) = \sum_{i=1}^I |V_R(t_i)|^2, \quad (5.27)$$

where t_i is the sampled time and $i = 1, 2, \dots, I$.



(a) Time domain



(b) Frequency domain

Figure 5.26: A power exponential pulse (top) and its frequency spectrum (bottom): dependence on the parameter n .

Optimization of Pulse Parameters

The power exponential testing pulse as used for exciting the transmitting antenna \mathcal{L}_T is defined in [43]:

$$I_E(t) = I_m(t/t_r)^n \exp[-n(t/t_r - 1)]H(t). \quad (5.28)$$

Here, I_m stands for the pulse amplitude, t_r is the rise time, n denotes the pulse rising power. Equation (5.28) defines the envelope of the excitation pulse. It is further modulated by the harmonic signal with the ringing frequency ω_0 . Therefore, the total pulse reads:

$$I_T(t) = I_E(t) \cos \omega_0 t. \quad (5.29)$$

The most significant parameter of the pulse shape is the rising power parameter. The influence of n on the pulse envelope shape and its spectrum is shown in Fig. 5.26. The growing n results in the narrower pulse width and the wider spectrum.

The pulse width t_w is related to the rise time and the rising power via

$$t_w = t_r n^{-n-1} \Gamma(n+1) \exp n, \quad (5.30)$$

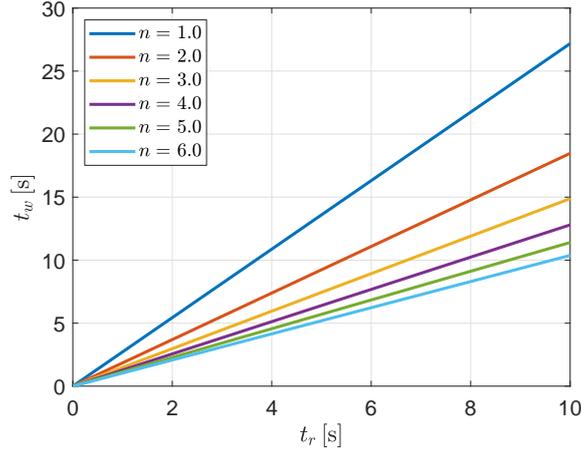


Figure 5.27: The pulse width t_w based on the rising power n and the rise time t_r parameters.

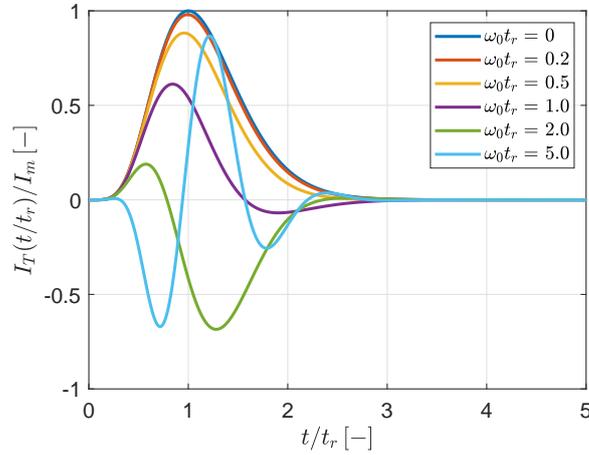


Figure 5.28: The total pulse shape according to change of the angular ringing frequency ω_0 (for $n = 6$).

where $\Gamma(x)$ denotes the Euler Gamma function. The pulse width rises with growing t_r and decreases with growing n as shown in Fig. 5.27.

The pulse amplitude I_m simply scales the pulse envelope I_E . The last parameter of the testing pulse is the angular ringing frequency ω_0 . As can be seen in Fig. 5.28, the pulse is not modulated for the normalized frequencies approximately $\omega_0 t_r < 1$. The combination of t_r and ω_0 can significantly increase or decrease the amplitude of the total pulse I_T .

The multi-objective optimization problem of the testing pulse is defined as follows:

$$\begin{aligned} \max_{\mathbf{x}} \quad & f_m(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x} \in \Omega, \\ & m \in \{1, 2, 3\} \end{aligned} \quad (5.31)$$

It means, that the problem can have one, two or three objective functions f_m . The objective functions are simply maximizing (or minimizing) the shielding effectiveness parameters defined in Section 5.2: SE_{PR} , SE_{DR} , and SE_{ER} . Note that any combination of

objective functions can define a problem (e.g. all functions can be maximized at once, or one can be maximized while one of the remaining or both of them can be minimized etc). Therefore, we can write that:

$$f_m = \begin{cases} SE_{PR}, & \text{if } m = 1 \wedge m \in \mathbb{S}, \\ SE_{DR}, & \text{if } m = 2 \wedge m \in \mathbb{S}, \\ SE_{ER}, & \text{if } m = 3 \wedge m \in \mathbb{S}, \end{cases} \quad (5.32)$$

where \mathbb{S} is any composition (see e.g. [56]) that can be made of a set of integers $\{1, 2, 3\}$.

The decision space vector of the optimization problem (6.17) \mathbf{x} consists of three design variables:

- the rise time t_r ,
- the pulse rising power n ,
- the ringing frequency ω_0 .

The decision space vectors $\mathbf{x} = \{t_r, n, \omega_0\}$ have to be chosen from a feasible part of the decision space Ω that is limited by:

$$\begin{aligned} t_r &\in \langle 0.5, 5.0 \rangle \text{ (in ps)}, \\ n &\in \langle 0.0, 30.0 \rangle, \\ \omega_0/(2\pi) &\in \langle 0.1, 10.0 \rangle \text{ (in THz)}. \end{aligned} \quad (5.33)$$

Note that only integers for the pulse rising power $n \in \{1, 2, \dots, 30\}$ are considered in the present study.

Numerical Examples

A standard multi-objective particle swarm optimization algorithm (MOPSO) [159] is used here for two different testing pulses. The study takes advantage of the implementation of the MOPSO algorithm available in a toolbox FOPS [127] with following controlling parameters:

- the number of agents: 50,
- the number of iterations: 100,
- the decreasing inertia weight: from 0.9 to 0.4,
- the cognitive learning factor: 1.5,
- the social learning factor: 1.5,
- the boundary type: reflecting.

For further details on the algorithm and its settings we refer the reader to [127].

The TD SE of a screen with a width $\delta = 1.0 \mu\text{m}$ made of copper with $\sigma_S = 58.0 \times 10^6 \text{ S/m}$ (see Table 10.2 in [183]) is analyzed. A single computation of the TD SE parameters for the thin screen takes approximately 0.15s on a personal computer with an AMD Ryzen 7 1700X platform and 32 GB of RAM. Every problem was repeated 50 times, so that a stochastic error can be neglected.

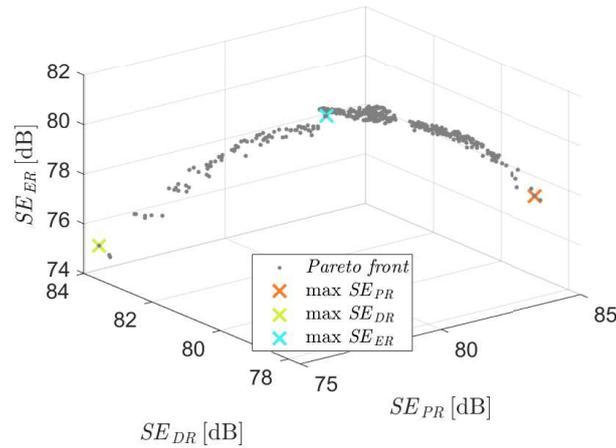


Figure 5.29: The non-dominated solutions for Problem 1 found by the MOPSO algorithm: trade-off between maximizing SE_{PR} , SE_{ER} , and SE_{DR} .

Table 5.12: The pulse parameters and the resulting SE values for the extreme solutions on the Pareto front of Problem 1.

solution	n [—]	t_r [ps]	f_0 [Thz]	SE_{PR} [dB]	SE_{DR} [dB]	SE_{ER} [dB]
max SE_{PR}	19.0	0.5	0.4	83.7	77.9	78.2
max SE_{DR}	22.0	5.0	0.6	75.1	83.7	75.3
max SE_{ER}	20.0	5.0	1.6	79.9	81.0	80.7

Problem 1: $\max SE_{PR}$, $\max SE_{DR}$, $\max SE_{ER}$

Obviously, the most common requirement would be to maximize all the SE parameters simultaneously:

$$\max_{\mathbf{x}} f_1(\mathbf{x}) = SE_{PR}(\mathbf{x}) \quad (5.34)$$

$$\max_{\mathbf{x}} f_2(\mathbf{x}) = SE_{DR}(\mathbf{x}) \quad (5.35)$$

$$\max_{\mathbf{x}} f_3(\mathbf{x}) = SE_{ER}(\mathbf{x}) \quad (5.36)$$

The resulting Pareto front of this problem is displayed in Fig. 5.29. We can see that the non-dominated solutions form a curved line in the three-dimensional space of the SE parameters. The most adjustable parameter is the SE_{PR} that varies from app. 75 dB to app. 85 dB. The peak value of the pulse can be clearly suppressed or amplified by an appropriate combination of the rise time and the ringing signal frequency $f_0 = \omega_0/(2\pi)$. It is also interesting that the solution with the maximal value of SE_{ER} resides exactly in the middle of the Pareto front, equally spaced to the other extreme solutions (maximizing the other two objectives).

Three extreme solutions maximizing the individual objectives are highlighted by the cross markers in Fig. 5.29. The pulse parameters for those solutions are summarized in Table 5.12. Figure 5.30 then displays the resulting pulse shapes: the total pulse containing the ringing signal I_T (solid line) and the pulse envelope I_E (dotted).

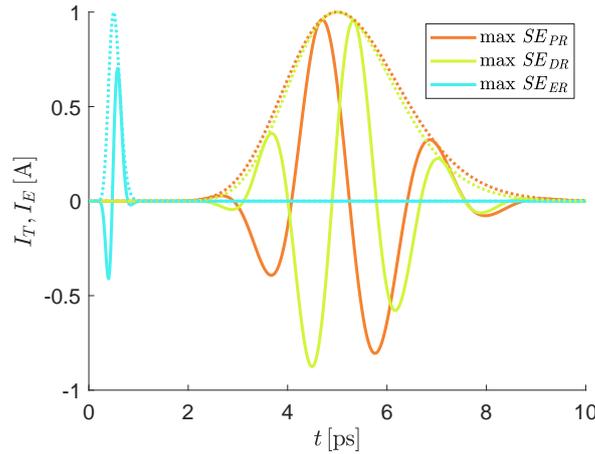


Figure 5.30: The testing pulse shapes for trade-off solutions maximizing the parameters SE_{PR} , SE_{ER} , and SE_{ER} (see Fig. 5.29). The solid line denotes the total pulse I_T while the dotted one corresponds to the pulse envelope I_E .

Problem 2: $\max SE_{PR}$, $\min SE_{ER}$

In the second problem we analyze two conflicting objectives: maximize the SE_{PR} associated with the peak value of the induced voltage while minimizing the SE_{ER} associated with the amount of energy:

$$\max_{\mathbf{x}} f_1(\mathbf{x}) = SE_{PR}(\mathbf{x}) \quad (5.37)$$

$$\min_{\mathbf{x}} f_2(\mathbf{x}) = SE_{ER}(\mathbf{x}) \quad (5.38)$$

With this problem formulation we prioritize the susceptibility of the device under test to the peak value of the disturbance source. The expected result should reveal the trade-off between the two watched objectives and to show the influence of individual pulse parameters.

The Pareto front (non-dominated solutions) found by MOPSO algorithm are shown in Fig. 5.31. We can see, that the level of the test susceptibility to a peak value can vary approximately from 65 dB to 85 dB, while the susceptibility to the energy varies only from 65 dB to 80 dB. The trade-off between the two objectives seems to be linear.

The color of the data markers in the individual subplots of Fig. 5.31 manifests the influence of the individual pulse parameters to the shielding parameters. Both the shielding parameters increase with the increasing value of the exponent n . The rise time dependence does not seem to be so straightforward. The SE parameters increase with the decreasing value of the t_r till approximately $SE_{PR} = 70$ dB. Then, the t_r equals to its maximal value (5 ps) for almost all of the non-dominated solutions having $SE_{PR} > 70$ dB. The ringing frequency ω_0 influences only the part of the Pareto front with maximal values of the SE parameters.

Conclusions

The paper describes a multi-objective optimization problem to set parameters of the power exponential pulse to optimize the TD SE of a metal thin-film screen. The rise time, the

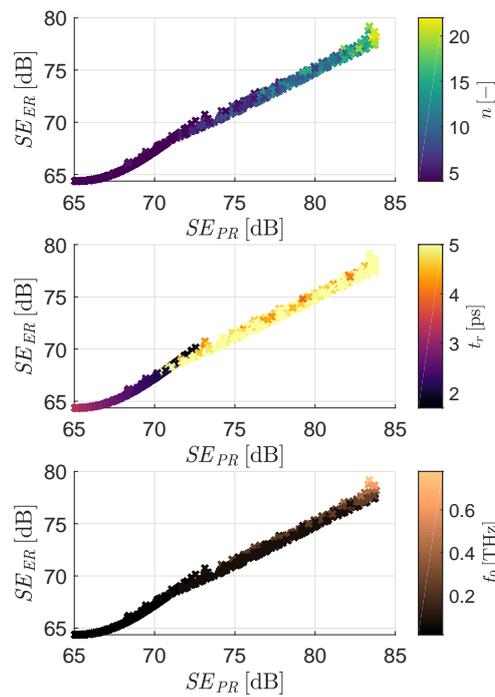


Figure 5.31: The non-dominates solutions for Problem 2: influence of pulse parameters - the rise power (top), the rise time (middle), and the ringing frequency (bottom).

pulse rising power and the ringing frequency of the testing pulse are found with the use of the MOPSO. The trade-off between shielding parameters associated with the peak level, the maximal time derivative, and the total absorbed energy is studied. The results show that individual shielding parameters vary significantly with the pulse parameters. The pulse rising power and the constructive (or destructive) combination of the pulse envelope with the ringing signal has the decisive impact.

Acknowledgments

The research reported in this paper was financed by the Czech Science Foundation under Grant 19-06049S.

Chapter 6

Optimization with Variable Number of Dimensions

The most important contributions of the author can be found in the domain of the optimization with a variable number of dimensions. First, a journal paper [93] introduces a PSO algorithm modification that enables to solve single-objective VND problems (please see the reprinted manuscript in Section 6.1). This paper includes also definitions of various benchmark problems and metrics to test the quality of VND optimization algorithms. The VNDPSO algorithm is compared to other algorithms in the paper.

A conference paper [95] applies the VNDPSO algorithm to solve the inverse scattering problem - namely the reconstruction of the permittivity profile based on the EM wave going through the layered medium. The reprinted version of the paper is presented in Section 6.2.

The VNDPSO algorithm was then employed to search for the optimal positions and values of the decoupling capacitors on a printed circuit board. This contribution was published in the journal paper [85]. The algorithm was asked to search for the number of capacitors also, which redefines the problem introduced in [94] to its VND version. The reprinted version of the journal paper is presented in Section 6.3.

Section 6.4 presents a reprinted version of a conference paper [80]. This paper compares various formulations of the linear antenna array design including single-objective, multi-objective and VND ones. The objectives of the antenna array design is the minimization of the side-lobe level, the first-null beam width and number of elements of the array. The problem formulations are solved using the consecutive variants of the PSO algorithm, including the multi-objective VND formulation as the most complex one. This approach seems to be the most effective one by means of the trade-off between the antenna array parameters quality and number of elements used.

6.1 Particle swarm optimization for problems with variable number of dimensions

Originally published as:

Kadlec, P., Šeděnka, V.: Particle swarm optimization for problems with variable number of dimensions. In *Engineering Optimization*, 50(3), pp. 382-399, 2017.

Abstract

Some real-life optimization problems show apart from the dependence on the combination of state variables also the dependence on the complexity of the model describing the problem. Changing model complexity implies changing the number of decision space dimensions. A new method called Particle Swarm Optimization for Variable Number of Dimensions is developed here. The well-known particle swarm optimization procedure is modified to handle spaces with variable number of dimensions within a single run. Some well-known benchmark problems are modified to depend on the number of dimensions. Novel performance metrics are defined in the article to evaluate convergence properties of the method. Some recommendations for setting the optimization are made according to results of the method on the proposed benchmark test-suite. The method is compared with the conventional swarm strategies able to solve problems with variable number of dimensions.

Introduction

Optimization became an inseparable part of the design process. Because of the complex description of structures, heuristic methods e.g. PSO (Particle Swarm Optimization, see [163]), GA (Genetic Algorithms, see [78]) and SOMA (Self-Organizing Migrating Algorithm, see [206]) are used extensively. These algorithms can be relatively simply implemented which makes them very attractive.

For conventional methods, the designer assigns dimensionality of the problem and an optimizer finds proper decision space vector. In this case, the value of the objective function in the optimum depends just on its position in the decision space. However, some problems can introduce another goal for the optimizer to find out optimal dimensionality as well. With this type of problem, the global minimum value depends on the complexity of the model also.

Consider a real-life example placement of transmitters to cover some area with TV broadcasting. The goal is to find optimal number, location and power of transmitters so that they cover given area with minimal overlap. In that scenario, three decision space variables per transmitter have to be defined (x , y , $power$). However, the number of transmitters is not a priori known. This type of problems will be called as VND (problem with Variable Number of Dimensions).

The aim of this article is to introduce a modification of the PSO algorithm to solve VND problems. The PSO is selected because it has been recently successfully used to solve many problems as described in [145], it works with the continuous decision space comparing to GA (see [78]) and it can be implemented relatively easy as described in [162].

Some articles are devoted to problem-oriented changes to the run of the optimization algorithm: [131, 151, 195, 197]. These methods are not applicable to wider range of problems. In [138], authors transform the VND problem into a multi-objective problem and solve it with NSGA-II by Deb [46]. Minimization of the dimensionality is added as another objective to minimize expected costs. This method requires more complex multi-objective algorithm and decision making from resulting Pareto-optimal set.

Most of the articles are devoted to modifications of GA. These algorithms are called VLGA (Variable Length Genome Algorithms). Decraene [47] introduces so called static (necessary) and dynamic (size is defined by model complexity) part of the chromosome. The crossover operator is applicable at certain valid positions of the chromosome. Some authors add another integer ([170]) or real ([113]) valued string to the chromosome, specifying the affiliation of genes with decision space variables. Ting [184] divides the genome with variable length into the substrings according to model parameters. Then, crossover is possible on level of substrings and on level of alleles. For most of the authors in domain of VLGA, it is typical to handle with the VND problem by modifying the crossover operator e.g. [47, 146, 152, 184]. Only Gheith [59] modifies the mutation operator to solve container pre-marshalling problem.

Few algorithms can solve VND problem in its natural definition: Immune Weed Optimization (IWO) introduced by Mehrabian [130] and Grouping Coral Reefs Optimization (GCRO) by Salzedo-Sanz [165]. For both, every agent can be generated with different decision space size. Nevertheless, individual agents can produce offspring only with the same decision space size. In fact, it is the same like a nave approach (single run per dimensionality).

Surprisingly, few efforts have been devoted to modification of PSO. Jamian [74] does not solve VND problem but adds another term to velocity update formula. The improvement factor (*ebest*) is composed randomly from all *gbest* from previous iterations. Li [114] and O'Neill [143] handle different sizes of particle and *gbest*. The particle changes its dimension towards a value randomly generated between its dimension and dimension of the *gbest*. Kiranyaz [103] introduced MD-PSO (Multi-Dimensional) where the dimension of PSO particle is controlled by another single-variable PSO. Yan [202] proposed the DA-PSO (Dimension Adaptive). Here, the particles dimension is defined by discrete header controlled by another discrete PSO (see Pugh [150]). In Hu [37], size of the particle is updated according to chordal length of the velocity components (particle, personal and global best). In Xue [200, 201], velocity update is changed when dimensions of particle and *gbest* are not same.

All previously published modifications of PSO to solve VND problems show deficiency in convergence properties. First, size of particles remains the same during whole optimization which slows down the search, or second, it forces all the particles to follow dimension of *gbest* which can lead to premature convergence in non-optimal dimension of the problem.

Variable Number of Dimensions Problem

Typical single-objective problem is defined as follows:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & x_{i,L} \leq x_i \leq x_{i,U}, \\ & i = 1, 2, \dots, N \end{aligned} \quad (6.1)$$

Here $x_{i,L}$ and $x_{i,U}$ stand for limits of i -th variable and f is the objective function, N denotes the total number of variables.

Decision space size N becomes a variable for the VND problem:

$$\begin{aligned} \min_{\mathbf{x}, N} \quad & f(\mathbf{x}, N) \\ \text{s.t.} \quad & x_{i,L} \leq x_i \leq x_{i,U}, \\ & i = 1, 2, \dots, N, \\ & N_{\min} \leq N \leq N_{\max} \end{aligned} \quad (6.2)$$

where N_{\min} and N_{\max} stand for minimal and maximal dimensionalities.

This type of problem puts forth another challenge for the optimizer - to search for the optimal number of the dimensions N_{opt} . Physical meaning of every variable has to remain the same for all possible dimensionalities throughout the optimization. Consider the VND problem having $N_{\min} = 3$ and $N_{\max} = 9$ (see Figure 6.1). Vector of three variables implies the only combination $\mathbf{x} = \{x_1, x_2, x_3\}$. Combinations like $\mathbf{x} = \{x_4, x_5, x_6, x_7, x_8, x_9\}$ or $\mathbf{x}\{x_1, x_3\}$ are not valid.

<i>Pos x</i> <i>Pos y</i> <i>Power</i>	<i>Pos x</i> <i>Pos y</i> <i>Power</i>	<i>Pos x</i> <i>Pos y</i> <i>Power</i>	$N_{\min} = 3$ $N_{\max} = 9$
$x_1 \quad x_2 \quad x_3$			$N_p = 3$
$x_1 \quad x_2 \quad x_3$	$x_4 \quad x_5 \quad x_6$	$x_7 \quad x_8 \quad x_9$	$N_p = 9$
$x_1 \quad x_2 \quad x_3$	$x_4 \quad x_5 \quad x_6$		$N_p = 6$
	$x_4 \quad x_5 \quad x_6$	$x_7 \quad x_8 \quad x_9$	NA
$x_1 \quad x_2 \quad x_3$	$x_4 \quad x_5$		NA
$x_1 \quad x_3$			NA

Figure 6.1: Valid and invalid decision space vectors for example with transmitters placement example.

With VND problems, one can find different minima $f(\mathbf{x}_{N,opt})$ for each possible N -dimensional space. Therefore, the optimizer can be stuck either in a local minimum caused by the nature of the objective function or by the dimension of the problem.

There are two types of VND problems depending on the position of the minimum of the N -dimensional space $\mathbf{x}_{N,opt}$:

1. the value of $x_{N,opt,i}$ remains the same for each possible N ,
2. the value of $x_{N,opt,i}$ changes with N .

For example, optimal vectors for $N = 1$ and $N = 2$ $\mathbf{x}_{1,opt} = \{x_1\}$ and $\mathbf{x}_{21,opt} = \{x_2, x_2\}$, x_1 for $N = 1$ is equal to x_1 for $N = 2$ for type 1 and is not for type 2. This behavior can be used to enhance the convergence rate towards the optimum.

Description of PSO-VND

The conventional PSO algorithm with linearly decreasing inertia weight (described in [167]) is modified to handle variable dimensionalities. The developed modification can be applied to any PSO-based algorithm with improved properties (see e.g. [71, 118, 119]) to enhance optimization convergence rate. The general flowchart of the method is Figure 6.2.

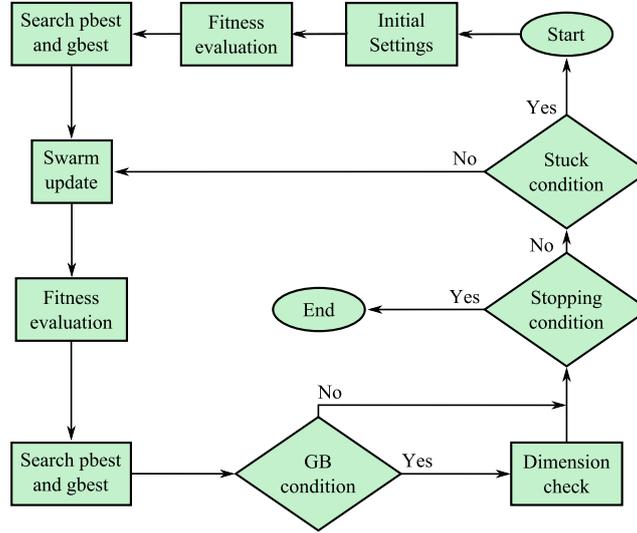


Figure 6.2: Flowchart of the PSO-VND method.

Initial settings

The following parameters remain unchanged with respect to the conventional PSO by [167]: P (number of particles), T (number of iterations), w (inertia weight), c_1 , c_2 (cognitive, social coefficients).

For the purposes of the PSO-VND approach, three new probabilities are introduced:

- p_1 adapting N to dimensionality of **gbest**,
- p_2 adapting N to dimensionality of **pbest**,
- p_3 adapting N to dimensionality of the particle.

Dimensionality of the p -th particle N_p , initial positions x_i and initial velocities v_i are generated randomly using:

$$x_i = x_{i,L} + rnd(x_{i,U} - x_{i,L}) \quad (6.3)$$

where rnd stands for a random value from uniform interval $\langle 0; 1 \rangle$. It is necessary to ensure that i) the swarm is diversified maximally and ii) deals with all the possible dimensions from N_{\min} to N_{\max} .

Swarm update

The position of the particles in every iteration t is:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \Delta t \mathbf{v}_t \quad (6.4)$$

where Δt denotes the time step ($\Delta = 1$) and the actual velocity vector \mathbf{v}_t is computed:

$$\mathbf{v}_t = w\mathbf{v}_{t-1} + c_1 rnd(\mathbf{pbest}_{t-1} - \mathbf{x}_{t-1}) + c_2 rnd(\mathbf{gbest}_{t-1} - \mathbf{x}_{t-1}) \quad (6.5)$$

For VND problem, equations (6.4) and (6.5) can contain vectors with up to three different dimensions: \mathbf{x}_{t-1} (\mathbf{v}_{t-1}), \mathbf{pbest}_{t-1} and \mathbf{gbest}_{t-1} . The pseudocode of the swarm update is depicted in Figure 6.3. Three scenarios can be followed:

1. all the three vectors have the same number of dimensions,
2. any two of them have the same number of dimensions,
3. all three have different number of dimensions.

```

for every  $\mathbf{x}_p$ 
  find  $\{s_1, s_2, s_3\} = \text{size}\{\mathbf{gbest}_{t-1}, \mathbf{pbest}_{p, t-1}, \mathbf{x}_{p, t-1}\}$ 
  if  $s_1 = s_2 = s_3$ 
    compute  $\mathbf{v}_t$  and  $\mathbf{x}_t$  using (6) and (5)
  else
    if any  $s_i = s_j \neq s_k, i, j, k = 1, 2, 3 \wedge i \neq j \neq k$ 
      if  $\text{rnd} < p_i + p_j$ 
        set  $s_n = s_i$ 
      else
        set  $s_n = s_k$ 
      end
    else
      if  $\text{rnd} < p_1$ 
        set  $s_n = s_1$ 
      elseif  $p_1 \leq \text{rnd} < p_1 + p_2$ 
        set  $s_n = s_2$ 
      else
        set  $s_n = s_3$ 
      end
    end
    temporary adapt  $\mathbf{x}_{t-1}, \mathbf{v}_{t-1}, \mathbf{gbest}_{t-1}$  and  $\mathbf{pbest}_{p, t-1}$  to  $s_n$ 
    compute  $\mathbf{v}_t$  and  $\mathbf{x}_t$  using (6) and (5)
  end
end

```

Figure 6.3: Pseudocode of the swarm update procedure to assign a new size s_n for the particle.

The probabilities p_1, p_2 and p_3 will be employed in cases 2 and 3. The probabilities correspond to three possible vector sizes: p_1 to $N(\mathbf{gbest})$, p_2 to $N(\mathbf{pbest}_p)$ and p_3 to $N(\mathbf{x}_p)$. The unit interval is divided into three parts with lengths corresponding to probabilities p_1, p_2 , and p_3 , respectively and compared with the random number. Then, the particles new number of dimensions corresponds to subinterval the random number belongs to.

After the new number of dimensions for p -th particle $N_{new}(\mathbf{x}_p)$ has been determined, vectors with size different from $N_{new}(\mathbf{x}_p)$ are temporary modified and equations (6.5) and (6.4) are applied. If $N_{new}(\mathbf{x}_p)$ is smaller, the last redundant components with indices $N_{new}(\mathbf{x}_p) < iN(\mathbf{x}_p)$ are eliminated. If the size of $N_{new}(\mathbf{x}_p)$ is larger, values of the missing components with indices i so that $N(\mathbf{x}_p) < i \leq N_{new}(\mathbf{x}_p)$ are generated randomly using (6.3). With this approach, the exploration is preferred. If exploitation is to be preferred indices of the redundant components should be determined according to change of fitness value in order to prevent destroying valuable parts of the decision space vector. However, this would require extra fitness function evaluation.

Balancing exploration and exploitation within a single dimension is controlled by well-known coefficients w , c_1 and c_2 . However, frequency of changing an agents dimension can be also viewed as balancing between exploration and exploitation. It is controlled by setting of p_1 (probability to follow **gbest**). With high value of p_1 , the exploitation process is emphasized because all particles are forced to change the dimension according to **gbest**. With a small value of p_1 the exploration process is emphasized because all particles follow their own current size or **pbest** as demonstrated in Figure 6.4.

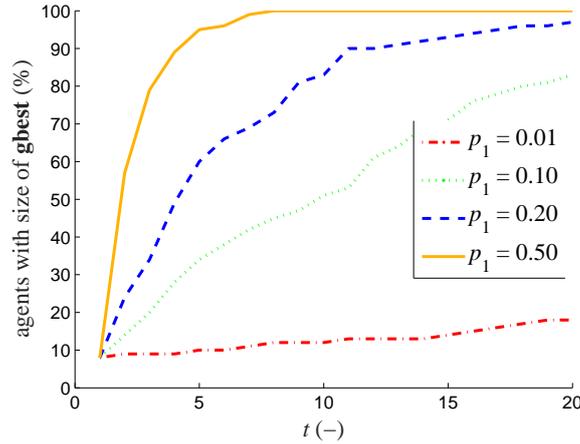


Figure 6.4: Influence of p_1 coefficient on number of agents following size of **gbest** during individual iteration steps, number of agents $P = 100$, maximum decision space dimension $N_{\max} = 10$.

Setting large values (larger than 0.2) for p_1 is not recommended. Then, all the particles in the swarm would be attracted to follow the number of dimensions for **gbest** from an early stage of the algorithm. If all the probabilities p_{1-3} are set to non-zero values, the particle can change the dimensionality with a certain probability which is crucial to protect the algorithm against getting stuck in a local minimum.

According to our experience with the algorithm based on numerous simulations setting the probability parameters so that $p_3 \geq 0.6$, $p_1 = 0.5p_2$ and $p_1 + p_2 = 1 - p_3$ is recommended (only p_3 selected by user).

Global Best Condition and Dimension Check

This approach is based on the idea that some components of the $\mathbf{x}_{N,opt}$ vector may be the same also for a local minimum having $N \neq N_{opt}$. The dimension check is applied only to a **gbest** vector that has significantly changed its position. Components of the **gbest** vector are evaluated in every possible dimension of the problem hoping that the quality of the proposed solution will be increased in other dimensions. For $N < N(\mathbf{gbest})$, some components are removed from **gbest** vector. If $N > N(\mathbf{gbest})$, some components have to be added. These components can be derived: i) randomly using equation (6.3) or ii) taken from the values in the repository of the best values for individual dimensions. The first method is denoted as rand and the second one as best throughout the article.

Convergence Evaluation

In order to evaluate convergence properties of the PSO-VND algorithm, special benchmarks and metrics have to be proposed to express quality of the VND problem solution.

Benchmark Problems

Seven VND benchmark problems can be found in [103]. These are modifications of well-known optimization problems. We will address these problems as follows: KSPH (sphere), KDEJ (De Jong), KROS (Rosenbrock), KRAS (Rastrigin), KGRI (Griewank), KSCH (Schwefel) and KGIU (Giunta).

Other nine problems from Ali [9] are reformulated to form new VND problems: SPH1 (Sphere 1), SPH2 (Sphere 2), MMIC (modified Michalewicz), MRAS (Rastrigin), MDPF (Dixon and Price), MMUM (Multimodal), MDCS (Deflected Corrugated Spring), MALP (Alpine) and MACK (Ackley). All parameters of the modified benchmark problems are summarized in Table 1 (please see the original of the manuscript [93]). The optimal dimensionality is defined by variable N_{opt} .

Variance of the objective function with dimensionality N expresses complexity of the problem. Figure 6.5 depicts the local optima values according to the changing number of dimensions for our new problems. The parameter N_{opt} is set to 5. Very interesting is the MMIC benchmark problem where the second smallest value of the objective function appears for $N = N_{opt} \pm 3$ instead of $N = N_{opt} \pm 1$.

It should be noted that the difference between global and local minimum is not the only parameter determining the complexity of the problem. Also, a shape of the objective function influences complexity of the problem.

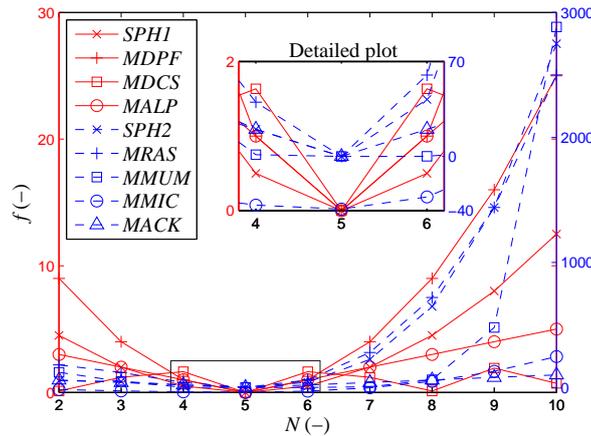


Figure 6.5: Comparison of benchmark suite local minima $f_{N,opt}$ for different decision space sizes. The optimal size for all the problems was set to $N_{opt} = 5$.

Evaluation Metrics

The evaluation metrics should numerically express the quality of the found solution so that solutions from different optimizers or runs can be easily compared. Thanks to non-linearity of optimization problems, the convergence properties should be viewed in both the decision and objective spaces.

The Fitness Error metric (FER) was introduced by Liang [117]:

$$FER = |f_{best} - f_{opt}| \quad (6.6)$$

which is the difference of the minimum objective value found $_{best}$ and the known optimum f_{opt} . The FER measures the distance in the objective space.

Decision Space Error (DER) is computed as the Euclidean distance from the known optimal vector \mathbf{x}_{opt} to the vector \mathbf{x}_{best} :

$$DER = \sqrt{\sum_{i=1}^{N_{opt}} [(x_{i,opt} - x_{i,best})^2]} \quad (6.7)$$

This metric can be evaluated only when size of the \mathbf{x}_{best} is equal to N_{opt} .

The Success Rate (SR) is a ratio between the number of successful runs N_{succ} and total number of runs N_{all} (see Auger [15]):

$$SR = \frac{N_{succ}}{N_{all}} 100\% \quad (6.8)$$

For the purposes of this article, a run of the optimizer is identified as successful if $FER(\mathbf{x}_{best}) < 0.01$

Determining the optimal number of dimensions can be seen as partial success. The Dimension Success Rate (DSR) can be computed using equation (6.8) just with the difference that an optimizer run is considered as successful if the \mathbf{x}_{best} vector is of size N_{opt} . Metric SP (Success Performance) tries to measure the time (expressed by number of FE) necessary to find the optimum. It is defined as the number of function evaluations performed until the algorithm reaches a fixed level of accuracy. The algorithm can fail to find the optimum with certain settings. Therefore, the mean success performance (MSP) has been defined in Auger [15]:

$$MSP = FE_{\max} \frac{100 - SR}{SR} + \frac{1}{N_{succ}} \sum_{k=1}^{N_{succ}} SP_k \quad (6.9)$$

where FE_{\max} is the maximal number of FE per single run, SR denotes the success rate, N_{succ} stands for the number of successful runs and SP_k is the number of fitness evaluations when the algorithm reached the optimum during the k -th successful run.

Experimental Results

The convergence properties of the presented PSO-VND algorithm are evaluated by a large number of tests on the benchmark suite with different settings (Table 6.1). All the tests are executed on the whole benchmark suite with $N_{\max} = 10$, ($N_{\max} = 100$ for chapter 6.1). All the problems have the optimum for dimensionality $N_{opt} = 5$, ($N_{opt} = 20, 50$ or 80 for chapter 6.1). Every problem has been executed hundred times.

Influence of Parameter Settings

As can be seen in Figure 6.6, the modified Rastrigin function (MRAS) problem seems to be the most difficult from the test suite. Although the algorithm has revealed the number of dimensions for almost all runs with different settings, it succeeded to determine the optimum just in a few cases, despite the fact that it was set relatively very robust ($SR = 13\%$ and 10% for SETA.1 and SETA.4 Table 6.2).

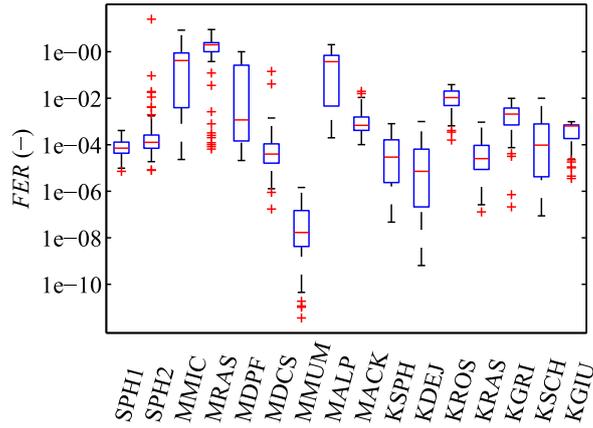


Figure 6.6: Fitness error boxplots for 100 repetitions of PSO-VND set according to SETB.2.

Table 6.1: Different initial parameter settings for PSO-VND used for simulations.

Sett	$P(-)$	$T(-)$	$p_3(-)$	DC type	Sett	$P(-)$	$T(-)$	$p_3(-)$	DC type
SETA.1	100	100	0.8	rand	SETA.4	100	100	0.7	rand
SETA.2	50	100	0.8	rand	SETA.5	100	50	0.8	rand
SETA.3	50	50	0.8	rand	SETA.6	100	100	0.1	rand
SETB.1	100	100	0.8	no	SETB.3	50	50	0.8	no
SETB.2	100	100	0.8	best	SETB.4	50	50	0.8	best

Comparison of the results for settings SETA.2 and SETA.5 shows that it is better to invest FE to extend the number of iteration loops rather than enlarge the swarm (see Figure 6.7). PSO-VND is usually faster in finding the optimum for SETA.2 than for SETA.5. This may be caused by the mechanisms that are incorporated into the algorithm to prevent premature convergence (dimension check and higher value of p_3). The faster convergence rate can be proved by result of Wilcoxon test for the MSP metric presented in Table 5 (see the table in the original manuscript [93]).

Results for settings SETA.1, 3, 4 and 6 examine the influence of parameters p_{1-3} . First, one should compare SETA.1 with SETA.4 where p_3 reaches higher values (larger diversity of N). In this case, there is no significant difference in the search rate and in the convergence rate (SETA.1: $DSR = 96\%$ and $SR = 77\%$, SETA.4: $DSR = 96\%$ and $SR = 76\%$). These results show that PSO-VND is not very sensitive to p_3 settings if chosen from a reasonable interval $\langle 0.6; 0.9 \rangle$. But its performance decreases significantly when p_3 is set to lower values (see Figure 6.8).

Influence of Dimension Check

Dimension check is aimed to enhance the convergence rate of the PSO-VND method. Two types of Dimension check (DC) are studied here: best and rand variant. The effect of this method can be demonstrated on results of six different settings: SETA.1 and 2 and SETB.1-4 (Table 6.2). First, focus is aimed to settings $P, T = 100$ (SETA.1, SETB.1, and SETB.2). The highest overall value of dimension search rate was achieved with best

Table 6.2: Overall results of PSO-VND, conventional PSO, CMA-ES, DE, SOMA and GA on benchmark test suite.

Sett	$FER(-)$	$DSR(\%)$	$SR(\%)$	$MSP(-)$	Sett	$FER(-)$	$DSR(\%)$	$SR(\%)$	$MSP(-)$
SETA.1	2.77e-01	100	0.8	rand	SETA.2	100	100	0.7	rand
SETA.3	1.24e+01	100	0.8	rand	SETA.4	100	50	0.8	rand
SETA.5	8.44e-01	50	0.8	rand	SETA.6	100	100	0.1	rand
SETB.1	4.93e-01	100	0.8	no	SETB.2	50	50	0.8	no
SETB.3	1.32e+01	100	0.8	best	SETB.4	50	50	0.8	best
PSO	7.56e-01	-	44	NaN	-	-	-	-	NaN
CMA-ES	5.51e+01	-	52	NaN	SOMA	2.06e+02	-	28	NaN
DE	5.44e+01	-	35	NaN	GA	5.54e+01	-	31	NaN

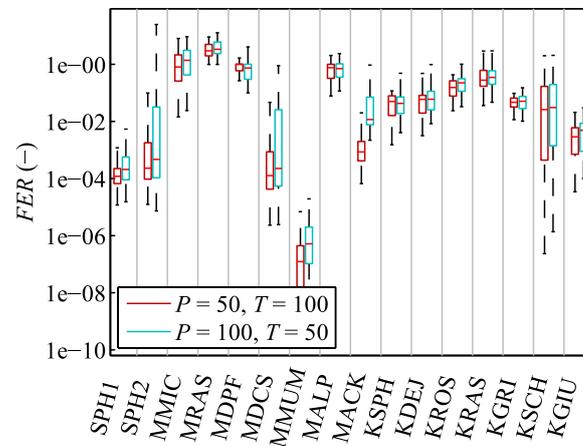


Figure 6.7: Influence of P (number of particles) and T (number of iterations) parameters on FER (fitness error rate) of PSO-VND.

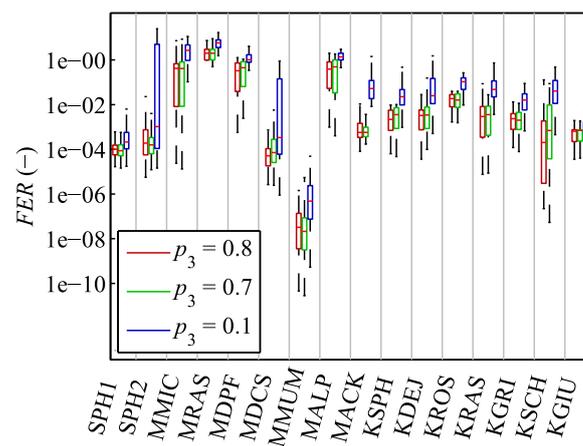


Figure 6.8: Influence of p_3 parameter (probability to explore decision space with the same number of dimensions as the current particle) on FER (fitness error rate) of PSO-VND.

variant of DC (98 %) and second highest for rand DC (96 %).

The best type of DC shows the fastest convergence: overall $MSP = 13675$ for SETB.2 in comparison to 15833 for SETA.3 (rand) and 28636 for SETB.1 (no DC). The convergence rate for best type of DC is higher for problems where the optimum remains at the same position with changing N (e.g. MDPF, MMIC).

As can be seen in Table 5 [93], the best variant of dimension check is just slightly better than the variant rand (SETB.2 vs. SETA.1) but the best variant is significantly better than no application of the dimension check (SETB.2 vs. SETB.1 and SETB.3 vs. SETB.4). This can be proved by FER boxplots depicted in Figure 6.9.

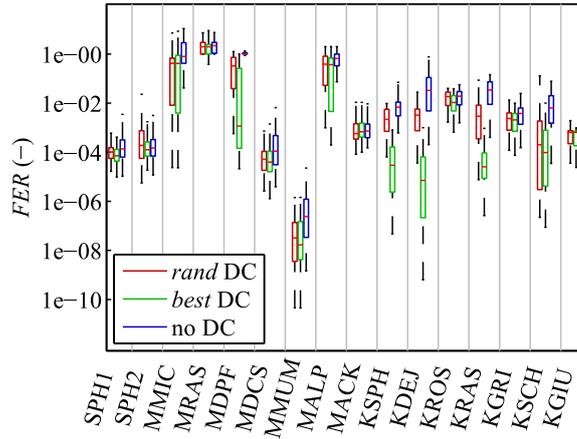


Figure 6.9: Influence of dimension check type on fitness error rate of PSO-VND.

Comparison with other Conventional Methods

In this subsection, PSO-VND is compared with naive solution of VND problems using other conventional methods: PSO as described in [167], CMA-ES (Covariance Matrix Adaptation Evolution Strategy, introduced in [68]), SOMA (Self-Organizing Migrating Algorithm, [206]), DE (Differential Evolution, [172]) and GA (Genetic Algorithm, [78]). These metaheuristics were set to be fairly compared with PSO-VND set according to SETB.2:

- PSO: $T = 100$, $P = 100$, $c_1 = c_2 = 1.494$, decreasing w 0.9 to 0.4;
- CMA-ES: $T = 90$, $P = 55$;
- SOMA: $T = 100$, $P = 100$, path length 1.3, number of steps 5, AllToAll;
- DE: $T = 100$, $P = 100$, crossover probability 0.3, scaling factor 0.2 0.8;
- GA: $T = 100$, $P = 100$, crossover probability 0.8, mutation probability 0.2, tournament size 2, number of bits per variable 10.

This comparison can be quite tricky. It is necessary to consider that if a VND problem is solved using conventional approach, the only way is to try every possible N from a set $\{N_{\min} : N_{\max}\}$. The conventional optimizer should distribute all the available FE

uniformly for each possible N . Results for all the metaheuristics are summarized in Table 6.2.

PSO achieves relatively good value of $FER = 7.36E - 01$ but the resulting search rate is only $SR = 44\%$ which is second best among conventional methods but significantly lower than value $SR = 81\%$ for SETB.2 of PSO-VND. As proved by the Wilcoxon and T-tests (Table 6.3), it is better to use PSO-VND optimizer rather than searching for every possible dimensionality using a conventional PSO.

CMA-ES is the best among tested conventional heuristics in SR metric with $SR = 52\%$. On the contrary, SOMA solves surely only the simplest problems (e.g. SPH2, MMUM) and achieves the worst overall $SR = 28\%$, because its AllToAll variant is strongly oriented on exploration and it has not enough power to search for the global minimum. As GA works with discrete decision space size, its accuracy is limited by choice of binary representation. GA is able to solve more complicated problems (MALP, KDEJ) more often than SOMA, but so as CMA-ES, GA fails to solve the most complicated problems (MMIC, MRAS, KROS, KRAS, KSCH) every time. The same as for GA holds more or less also for results of DE, but it achieves slightly better overall search rate $SR = 35\%$ in comparison with $SR = 31\%$ for GA. Nevertheless, PSO-VND outperforms all of the conventional algorithms.

Another option to compare results of our PSO-VND method could be to modify conventional PSO so that for problem with N_{\max} dimensions PSO works with agents of size $N_{\max} + 1$. Then, the last discrete variable of individual agent would control portion of decision space vector that would be used for evaluation of fitness function. This is similar approach to those published in [103] and [202]. However, increasing the decision space size would decrease convergence properties of used optimizer.

Comparison with MD-PSO

According to our knowledge, there is only one benchmark study of stochastic optimizer able to solve VND type of problems called MD-PSO (see [103]). In the article, results of two variants are presented: MD-PSO in canonical form and MD-PSO with FGBF (Fractional Global Best Formation). FGBF routine works similar to our best DC variant. FGBF forms new potential global best for every possible dimension.

Table 5([93]) compares results of PSO-VND set with results presented in [103]. Settings of PSO-VND: $P = 160, 320$ and 640 , $T = 5000$, $p_3 = 0.8$ once when DC is not applied and once with best DC variant. PSO-VND with no DC clearly outperforms the MD-PSO without FGBF in all watched metrics. This is probably caused by the fact, that in PSO-VND particles can recombine to every possible dimension while in MD-PSO, particle can change the dimension only by one at current iteration. According to results of MSP metric, MD-PSO was slightly better on easier problems with larger swarms (e.g. KGIU and KDEJ for $P = 640$) while PSO-VND was better on more complex problems (see e.g. KROS). This behaviour is probably caused by the power of FGBF method for large enough swarm.

Conclusion

A novel algorithm for solving problems of VND problems has been introduced. The PSO method has been modified (introducing probability parameters p_1-p_3) to handle variable

Table 6.3: Settings of MOSOMA parameters for design of Yagi-Uda antenna.

Comparison	SPH1	SPH2	MMIC	MRAS	MDPF	MDCS	MMUM	MALP
SETA.1	+/+	+/+	+/+	+/+	+/+	+/+	+/=	+/+
SETA.2	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/+
SETA.1	=/=	=/=	=/=	=/=	=/=	=/=	=/=	=/=
SETA.3	+/+	=/=	=/=	=/=	+/+	=/=	=/=	=/=
SETB.2	+/+	=/=	=/=	=/=	+/+	=/=	=/=	=/=
SETB.2	+/+	=/=	+/+	+/+	+/+	+/+	+/+	=/=
SETB.3	-/-	-/-	=/=	-/-	-/-	-/-	-/-	-/-
PSO	-/-	-/-	-/-	-/-	-/-	-/-	+/=	-/-
Comparison	MACK	KSPH	KDEJ	KROS	KRAS	KGRI	KSCH	KGIU
SETA.1	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/+
SETA.2	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/+
SETA.1	=/=	+/+	+/+	=/=	-/-	=/=	-/-	=/=
SETA.3	=/=	+/+	+/+	+/+	=/=	=/=	+/+	+/+
SETB.2	=/=	=/=	+/+	=/=	+/+	+/+	+/+	+/+
SETB.2	=/=	=/=	+/+	+/+	+/+	+/+	+/+	+/+
SETB.3	-/-	-/-	=/=	-/-	-/-	-/-	-/-	-/-
PSO	-/-	-/-	=/=	-/-	-/-	-/-	-/-	-/-

number of dimensions. A benchmark test suite and metrics have been defined to evaluate convergence properties. Presented results have shown that the novel method is much more efficient for solving VND problems than the naive approach using conventional PSO and other heuristic algorithms (CMA-ES, SOMA, DE and GA). The PSO-VND method was compared with the only capable PSO-based method. The recommendations for PSO-VND settings have been derived.

This article can encourage other authors to focus on VND optimization problems. First of all, new procedures can be derived and other evolutionary algorithms (Differential Evolution, Self-Organizing Migrating Algorithm etc.) can be adapted to choose the optimal number of degrees of freedom within the optimization run. Also, much more attention could be paid to derive more complex VND benchmark problems.

Acknowledgments

Research described in this article was financed by Czech Ministry of Education in frame of National Sustainability Program under grant LO1401. Infrastructure of the SIX Centre was used for this research.

6.2 Solution of an Inverse Scattering Problem Using Optimization with a Variable Number of Dimensions

Originally published as:

Kadlec, P., Šeděnka, V., Štumpf, M., Marek, M.: Solution of an Inverse Scattering Problem Using Optimization with a Variable Number of Dimensions. In *2017 International Conference on Electromagnetics in Advanced Applications (ICEAA)*, 976-979, 2017.

Abstract

The inverse scattering problem applies in many areas of engineering such as biomedicine, civil engineering, electromagnetic compatibility or geophysics. In this paper, one-dimensional inverse scattering problem is formulated as an optimization task with variable number of dimensions. Goal is to reconstruct material properties of a layered medium from its reflection coefficient. Then, this problem is solved using PSO-VND algorithm. Its solution is compared with solutions by conventional approaches: PSO, GA and DE optimizers. Results show that VND formulation significantly reduces computational time.

Introduction

The inverse scattering problem is a problem of great interest because of its relative ease of use in industrial and civil engineering, medical diagnostics and buried object detection etc. The radiofrequency (RF) or microwave (MW) systems have some benefits in comparison with commonly used systems such as radars or computer tomography: RF and MW systems are much cheaper and do not burden target with a dose of harmful radiation. This is paid off by the complex reconstruction of a target object from the— scattered field which is a nonlinear and ill-posed problem [19]. There are three types of inverse scattering problem: a) searching for position and simple properties of target body knowing the detailed information about background material [132], b) searching for shape of target [34], and c) reconstruction of the material distributions over defined area [180]. Clearly, the last class of problems is the most challenging one, because it is not known a priori what the number of homogeneous parts is and what are their shapes. In this paper, we shall tackle a problem of type c).

The inverse scattering problem is well-known from biomedicine applications as a diagnostic tool [20, 205]. It is extensively used in civil engineering e.g. for the detection of inclusions in concrete [186] or for imaging of reinforcement bars in concrete objects [7] etc. It is extensively used in geophysics for the ground penetrating radar [157] and in designing shield structures in the field of ElectroMagnetic Compatibility (EMC) [183].

The inverse scattering problem is traditionally approached via iterative solution [44] or using well-known algorithms based on the Born or Rytov approximations with the possibility of employing the unique features of 1D problems [32]. In all these approaches, a unifying concept is the principle of reciprocity a relevant application of which can be found in [23]. Another efficient way for solving 2D/3D inverse scattering problems is based on stochastic optimization techniques, which is the main subject of this paper.

There are plenty of evolutionary algorithms successfully applied to an inverse scattering problem: e.g. GA (Genetic Algorithms) [33,140], DE (Differential Evolution) [35,132], and PSO (Particle Swarm Optimization) [166,205]. When applied to type c) inverse problems, all these methods start with the uniform tessellation of a researched space and then search for material properties of every space element. For a sufficient resolution, a large number of elements with the fixed size has to be used which enlarges the— dimensionality of the optimization problem accompanied with drastic increase of CPU demands.

In this paper, a 1D inverse scattering problem is solved using a recent PSO-VND algorithm (PSO-Variable Number of Dimensions) introduced in [93]. This algorithm is able to handle agents (trial vectors) with different sizes. Using VND formulation of inverse scattering problem enables to tessellate a searched space into parts with varying sizes and material properties which leads to significant reduction of optimization variables and savings of CPU resources consequently.

Problem formulation

In this paper we aim at reconstructing the constitutive parameters of a dielectric layered medium with the material inhomogeneity in the z -axis direction (see Figure 6.10). The layered dielectric medium is assumed in the AOI (Area Of Interest). Our goal is to find unknown number of layers, their width and material properties (only non-magnetic materials with $\mu_r = 1$, only relative permittivity is searched). The AOI is illuminated by a plane wave with the known wavelength λ and the unit amplitude while the pertaining reflection coefficient is being measured. The recursive formula for the reflection coefficient is used [144]:

$$R_i = \frac{\rho_i + R_{i+1} \exp -2jk_i l_i}{1 + \rho_i R_{i+1} \exp -2jk_i l_i} \quad (6.10)$$

where ρ_i stands for elementary reflection coefficient of i -th layer:

$$\rho_i = \frac{n_{i-1} - n_i}{n_{i-1} + n_i} \quad (6.11)$$

where n_i is the refractive index (for a wave impinging from a direction perpendicular to the layers):

$$n_i = \sqrt{\frac{\epsilon_i}{\epsilon_0}} = \sqrt{\epsilon_{r,i}} \quad (6.12)$$

where ϵ_i stands for permittivity of the i -th layer, ϵ_0 is permittivity of vacuum and $\epsilon_{r,i}$ is the relative permittivity of medium. The phase thickness can be calculated from:

$$k_i l_i = \frac{2\pi n_i l_i}{\lambda} \quad (6.13)$$

where l_i denotes the width of i -th layer. Having M layers, reflection coefficient R_{M+1} is equal to the elementary reflection coefficient ρ_{M+1} .

Optimization methods

Particle Swarm Optimization

The Particle Swarm Optimization is a technique proposed by Eberhart and Kennedy in 1996 [51]. It is inspired by the cooperation of a swarm of particles searching for food.

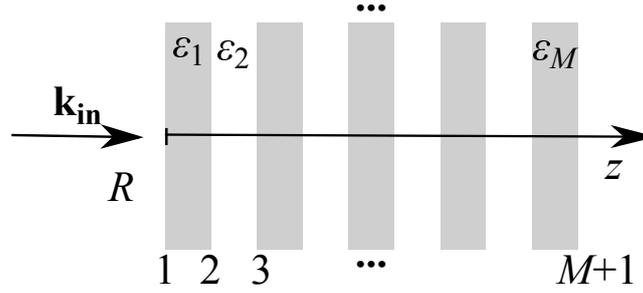


Figure 6.10: One-dimensional inverse scattering problem formulation.

New position of a p -th particle from a swarm in iteration i is determined by [163]:

$$\mathbf{x}_{p,i} = \mathbf{x}_{p,i-1} + \Delta t * \mathbf{v}_{p,i} \quad (6.14)$$

where size of vector \mathbf{x} is $N \times 1$, N stands for number of dimensions, symbol $*$ denotes the element-wise multiplication, Δt is a time step (usually 1) and particles current velocity $\mathbf{v}_{p,i}$ is computed using:

$$\mathbf{v}_{p,i} = w\mathbf{v}_{p,i-1} + c_1 \text{rand}(\mathbf{pbest}_p - \mathbf{x}_{p,i}) + c_2 \text{rand}(\mathbf{gbest} - \mathbf{x}_{p,i}) \quad (6.15)$$

Here, w denotes the inertia weight (large value enforces exploration), c_1 , c_2 are cognitive and social learning factor (c_2 enforces exploitation), rand denotes a random number from interval $\langle 0, 1 \rangle$, \mathbf{pbest}_p is the position with the best value of the fitness function visited by the p -th particle, and \mathbf{gbest} is a position with the best value of the fitness function visited by the whole swarm.

PSO with Variable Number of Dimensions

Some optimization problems show dependence not only on combination of design variables but on the number of these variables too. Such problems can be solved using PSO-VND algorithm whose comprehensive description can be found in [93]. When trying to transform the conventional PSO to the VND algorithm, attention has to be turned to the velocity update formula (6.15), where up to three vectors with different sizes (\mathbf{x} , \mathbf{pbest} and \mathbf{gbest}) can be found. New particles size has to be determined using three probabilities:

- p_1 adapting to dimensionality of \mathbf{gbest} ,
- p_2 adapting to dimensionality of \mathbf{pbest} ,
- p_3 adapting to dimensionality of the particle (\mathbf{x}).

When a new dimension has to be determined, random number from interval $\langle 0, 1 \rangle$ is compared to intervals $\langle 0, p_1 \rangle$, $\langle p_1, p_1 + p_2 \rangle$ and $\langle p_1 + p_2, 1 \rangle$ set by the user. The new dimension is defined by the corresponding interval where the random number belongs to. A number from first interval means \mathbf{gbest} size, second interval means \mathbf{pbest} size and the last one means that the particle keeps the same dimensionality.

Experiment results

As an experiment, the reconstruction of a permittivity profile of total width $2.5\lambda_0$ with five layers of relative permittivities $\epsilon_r = \{1.00, 6.50, 2.20, 1.38, 1.00\}$ and widths $l/\lambda_0 = \{0.50, 0.25, 0.50, 0.75, 0.50\}$ was performed. The reflection coefficient response of this profile on the frequency interval $\langle 55/70, 55/40 \rangle f_0$ with 101 samples were known. For every single algorithm scenario, 500 agents and 200 were run 100-times.

Conventional formulation

Three stochastic algorithms are used for comparison with a solution by PSO-VND: PSO [163], GA [78] and DE [172]. For conventional stochastic optimizers, the searched interval is tessellated into 10 (the smallest number of intervals in agreement with widths of individual layers) and 20 uniform subintervals. Then, for every element (see Figure 6.11), the relative permittivity from an interval $\epsilon_r \in \langle 1, 10 \rangle$ is searched according to the fitness function (valid also for the VND formulation):

$$f(\mathbf{x}) = \sum_{i=1}^S [\Re(R_i(\mathbf{x})) - \Re(R_{i,res})] + \sum_{i=1}^S [\Im(R_i(\mathbf{x})) - \Im(R_{i,res})] \quad (6.16)$$

where S is the number of frequency samples (101 in this study), $R_i(\mathbf{x})$ is the i -th sample of the reflection coefficient for a decision space vector \mathbf{x} , and $R_{i,res}$ is the known reflection response of the system (see Figure 6.12).

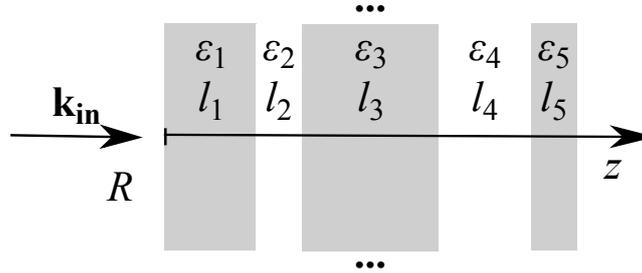


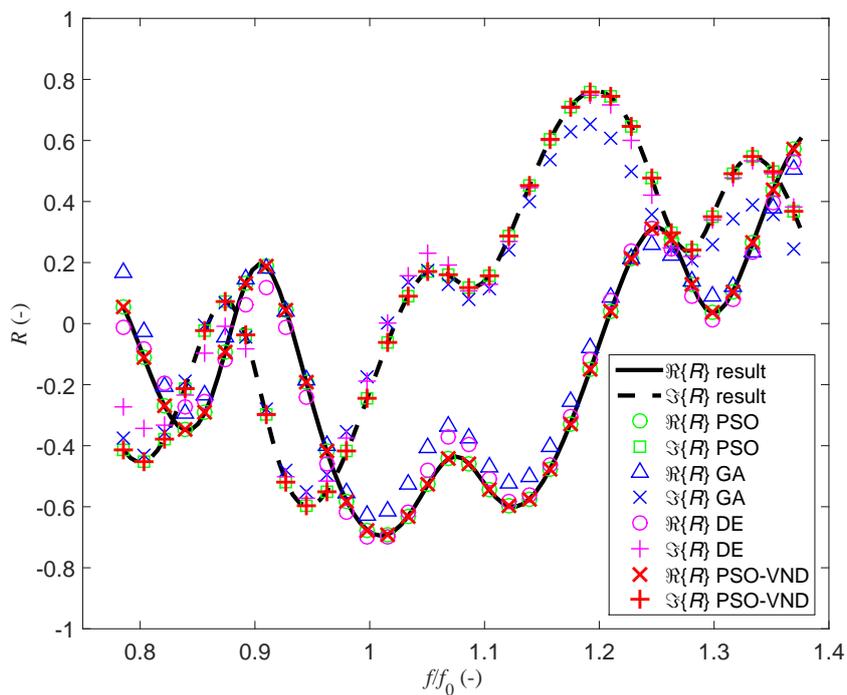
Figure 6.11: A VND formulation of 1D inverse scattering problem unknown number of layers with variable sizes.

PSO was set as follows: decreasing inertia weight from $w = 0.9$ to $w = 0.4$, $c_1 = c_2 = 1.5$, reflecting boundary condition: for GA probability of singlepoint crossover and mutation $p_c = 0.9$, $p_m = 0.7$, respectively, binary precision $b_p = 20$ and tournament selection; for DE probability of crossover $p_c = 0.2$ and scaling factor $s_f = 0.2$.

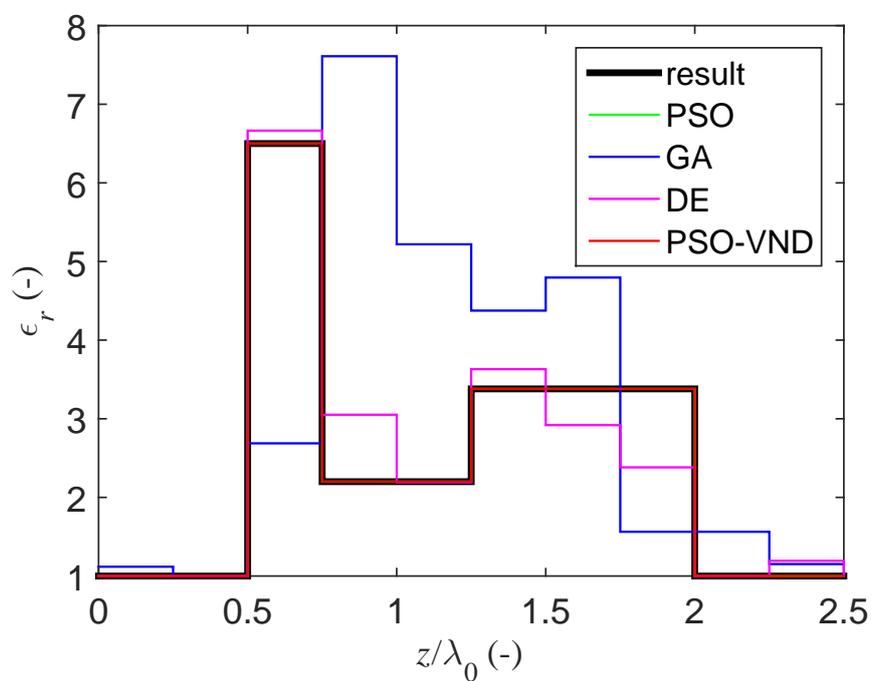
PSO-VND formulation

With the VND formulation, the algorithm has to search for the relative permittivity and width of an unknown number of layers and the start position of the first layer (see Figure 6.11). Therefore, the size of every particle was an odd number from interval $\langle 3, 21 \rangle$. The first variable from the decision space vector was the start position of the first layer.

PSO-VND was set in agreement with conventional PSO, i.e. the relative permittivity was found in the interval $\epsilon_r \in \langle 1, 10 \rangle$ and the width in the interval $l \in \langle 0.1, 1.0 \rangle \times \lambda_0$. PSO-VND special parameters were set as follows: probabilities controlling new size of particle $p_1 = 0.03$, $p_2 = 0.06$ and $p_2 = 0.91$. No dimension check was applied.



(a)



(b)

Figure 6.12: Best results for used optimizers: reflection coefficient (top) and permittivity profile (bottom).

Results

Results from 100 runs of all algorithms are summarized in Table 6.4. The best reflection coefficients and permittivity profiles found by individual optimizers are depicted in Figure 6.12. Despite the best value of the fitness was found by PSO, the PSO-VND clearly outperforms other algorithms because it achieves better values in watched metrics: mean fitness function error and success rate (SR). Run of algorithm was considered as successful if the fitness function value decreased under $f = 0.1$. Despite not so robust settings (only 100000 fitness evaluations for a problem with up to 20 variables) PSO-VND achieved a very high success rate 74%. Only DE was able to come close to the mean value of the fitness function: 1.81 for DE in comparison with 1.18 for PSO-VND.

Table 6.4: Different initial parameter settings for PSO-VND used for simulations.

alg.	layers	mean	st. d.	best	worst	SR
PSO	10	3.91	5.23	2.5e-29	16.76	2
	20	5.00	3.85	1.2e-01	19.35	0
GA	10	2.02	2.65	2.2e-01	13.50	0
	20	7.98	5.59	1.9e-01	16.42	0
DE	10	1.81	0.62	3.9e-01	2.94	0
	20	6.67	1.25	3.7e+00	9.28	0
VND	1:10	1.18	2.07	4.2e-14	8.2	74

Conclusions

A new approach based on the stochastic optimization has been applied to the 1D inverse scattering problem. The PSO-VND algorithm has been applied to search for a material profile from the reflection coefficient. PSO-VND has shown a better convergence in comparison with conventional approaches which is caused by its ability to work with particles having variable dimensions. It enables to regularize the space and decrease the dimension of the decision space vectors. This leads to significant savings of CPU resources.

Acknowledgments

The research reported in this paper was financed by the Czech Science Foundation under Grant 17-05445Y and by the Czech Ministry of Education, Youth and Sports under Grant LD15005 and under Grant LO1401.

6.3 PCB Decoupling Optimization with Variable Number of Capacitors

Originally published as:

Kadlec, P., Marek, M., Štumpf, M., Šeděnka, V. : PCB Decoupling Optimization with Variable Number of Capacitors. In *IEEE Transactions on Electromagnetic Compatibility*, 61(6), pp. 1841-1848, 2019.

Abstract

The decoupling of modern printed circuit boards introduces a very complex task. Powerful stochastic optimizers are usually used to determine values and positions of decoupling capacitors on the board. The number of capacitors used has to be determined a priori by the user which brings problems with convergence of the optimization process or can lead to a waste of resources when the noises are to be attenuated to a certain level. In this paper, an approach based on combination of Time Domain - Contour Integral Method and Optimization with Variable Number of Dimensions is introduced. The optimizer works with models having variable dimensions and searches for the optimal one. The approach is tested on two example power circuit boards with various noise attenuation limits and constraints on capacitors positions and values.

Introduction

It is indispensable to control and suppress the level of conducted and radiated emissions produced by switching components on modern PCBs (Printed Circuit Boards). Decoupling capacitors are used to compensate voltage dips and spikes caused by unintentional noise currents, and for voltage disruptions due to intentional current transients caused by the normal operation of integrated circuits and other switching circuits on the PCB. A proper modeling of decoupling capacitors on PCBs is a problem of great importance as proven by a large number of papers tackling it throughout the last twenty years (please refer to e.g. [49, 53, 104–106, 199, 208]). The automated design of decoupling structures comprises the search for proper capacitance value and position of one or more capacitors.

The indisputable power and ease of implementation of stochastic global optimization methods such as GA (Genetic Algorithms) [78], PSO (Particle Swarm Optimization) [51] or DE (Differential Evolution) [172] lead to their massive use for solving plenty of design problems from a domain of electromagnetics including the decoupling problem. Usually, conventional stochastic optimizers work in two modes: 1) the user selects a parametrized model of designed structure and the optimizer then searches for proper combination of design variables, or 2) components with selected parameters of an uniformly distributed set are enabled/disabled by the optimizer. The dimension of the optimization problem is fixed for both the cases: the optimizer searches for set of real values or for the optimal combination of logical values in the latter case. The number of design variables or switching elements defines the dimension of the problem, which has to be assigned by the user a priori.

The first approach was used in e.g. [11, 96, 98, 148, 179, 187] where authors searched for the position and values of fixed number of capacitors on various power bus structures. Quality of the result is here significantly affected by the choice of the parametrized

model. The fixed model reduces significantly the number of possible realizations before the optimization and makes demands on user who has to understand the problem very well.

Authors in [21, 102, 209, 210] use stochastic optimizers (GA in most cases) to enable/disable capacitors placed in a regular grid. The limiting factor here is the density of the grid. A sparse grid decreases the number of possible capacitor positions. On the other hand, optimization using a dense grid results in exponential growth of the design variables which strongly deteriorates the convergence of stochastic optimization algorithms.

In recent years, a certain strain has been put on the effectiveness of the automated design of decoupling structures. Authors in [108] developed an algorithm minimizing the number of decoupling capacitors needed to meet a predefined impedance characteristics of a PDN (Power Distribution Network) in frequency domain. Capacitors are allowed to be placed only to positions of pin ports. A methodology for reducing the number of decoupling capacitors using a deterministic optimization techniques was published in [193]. Authors in [54] examined the influence of position and the number of decoupling capacitors on a Ball Grid Array package on a PCB. In [50] authors derived a closed-form solution to calculate the grid PDN impedance to speed up the decoupling design process. The prelayout library and equivalent models of decoupling capacitors were published in [203]. Authors in [147] compared different capacitor placement strategies on the gridded PDN to enhance efficiency of GA-based search strategy.

The problem with the choice of the accurate model dimension can be solved with the use of the so-called VND (Variable Number of Dimensions) class of optimization algorithms. These algorithms work with sets consisting of agents that can have different sizes. That introduces another task for the optimizer - not only the optimal combination of design variables, but also the optimal dimensionality, that means the appropriate model complexity. There are many modifications of Genetic Algorithms e.g. [47, 113, 170, 184] and few modifications of Particle Swarm Optimization algorithm [93, 103]. Those VND algorithms were successfully used to solve problems introducing the choice of the model dimension: e.g. the transmitter placement problem [184], the clustering problem [103], or the inverse scattering problem [95].

In [94], we have shown that the optimal position and value of a single decoupling capacitor can be found using a conventional PSO algorithm [51] in cooperation with a TD-CIM (Time Domain - Contour Integral Method) solver [173]. Then, purpose of this paper is to generalize the design methodology to suppress the noise signals propagating on a PCB to a certain level. The search for positions and values of a priori unknown number of decoupling capacitors is driven by the PSO-VND algorithm introduced in [93]. In comparison to previous publications, we want to propose the general procedure to design the set of decoupling components with the optimal size, values, and positions directly in time domain.

Problem Description

The printed circuit board is modeled in this paper by a parallel plate structure that is depicted in Fig. 6.13. Symbol O denotes the origin of the right-handed Cartesian coordinate system and any point in the space can be localized using three mutually orthogonal unit vectors $\{\mathbf{i}_1, \mathbf{i}_2, \mathbf{i}_3\}$. The PCB is modeled by two PEC plates with a shape defined by the closed contour $\partial\Omega$. The top and ground PEC layers lie in the

planes perpendicular to \mathbf{i}_3 . The PEC layers are separated by the dielectric layer with the relative permittivity ε_r and thickness d . The background medium is made up of a vacuum with the permittivity ϵ_0 and the permeability μ_0 . The structure is excited at 'PORT 1' by an electric current pulse representing the noise signal and the voltage response is observed at point of interest labeled as 'PORT 2'. The response is affected by an unknown number N of capacitors. Position of n -th capacitor is defined by a two-dimensional vector $\mathbf{r}_n = r_{n,1} \times \mathbf{i}_1 + r_{n,2} \times \mathbf{i}_2$. The value of n -th capacitor is denoted as C_n .

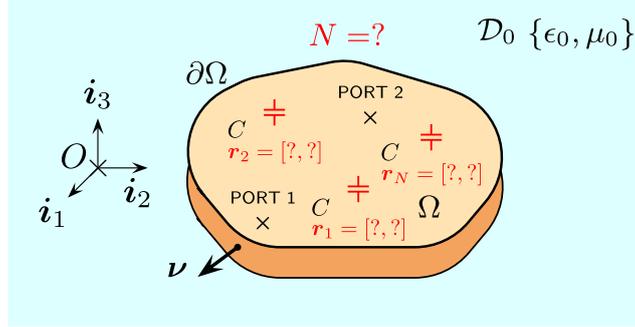


Figure 6.13: Description of a power-ground structure containing unknown number of decoupling capacitors.

The aim of the optimization is to attenuate the received voltage response to the required level L :

$$\begin{aligned} \min \quad & f(N, \mathbf{R}, \mathbf{c}) = |PVR(N, \mathbf{R}, \mathbf{c}) - L| \\ \text{subject to} \quad & \mathbf{R} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \dots \\ \mathbf{r}_N \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_N \end{bmatrix} \\ & \mathbf{r}_n \in \Omega \quad \forall n \in \{1, 2, \dots, N\} \\ & N \in \Gamma, C \in \Lambda \end{aligned} \quad (6.17)$$

where \mathbf{R} is a matrix formed by all N capacitors positions of total size $[N \times 2]$, \mathbf{c} denotes a vector of all capacitance values. Symbol Γ denotes the list of all possible dimensions N (according to the minimum and maximum number of capacitors, that can be used) and Λ stands for the list of available capacitance values. The peak voltage ratio PVR is defined as follows:

$$PVR = \max_{t \in \mathcal{T}} |V_{2,1}(N, \mathbf{R}, \mathbf{c})| / \max_{t \in \mathcal{T}} |V_{2,1}(0, \emptyset, \emptyset)| \quad (6.18)$$

Here, t stands for time, \mathcal{T} is the length of the time window of the excitation pulse (6.24), and $V_{2,1}$ denotes the voltage response observed at 'PORT 2'. The peak voltage of the response with N capacitors is in the numerator while the denominator consists of the peak voltage measured at 'PORT 2' with no capacitor present on the PCB.

The formulation of objective function (6.17) favors solutions with PVR close to the target attenuation value L and penalizes solutions that exceed the target value. The better attenuation level requires use of more decoupling capacitors, which enhances the costs of

the final design (price of the components, place on the PCB etc.). The optimum number of capacitors to meet the target attenuation value can be found by an iterative approach: the conventional PSO is run for a problem with a fixed dimension and if it does not meet the target attenuation value, the dimension is enhanced according to one capacitor. On one hand, this brute-force approach avoids the use of more complex VND algorithm, but on the other hand it leads to a waste of CPU resources. The amount of waste increases with number of components needed. This can be proved by the results of the comparative study concerning the PSO-VND algorithm published in [93]. There, the PSO-VND algorithm has been tested on a suite consisting of 16 VND benchmark problems by means of the *SR* (Search Rate) metric. This metric is defined as percentage of successful algorithm runs. The algorithm run is considered as successful when it finds the global optimum with error lower than 0.01. The overall *SR* of the PSO-VND algorithm raised to 81 % compared to 44 %, 35 %, and 31 % when conventional PSO, DE, and GA algorithms with same computational resources have been used. Furthermore, the conventional algorithms achieved *SR* = 0 [%] for more complex test problems.

Computational Methods

The Time Domain - Contour Integral Method

The reciprocity theorem of time-convolution type enables to formulate the Time Domain - Contour Integral Method as described in detail in [174]. In short only, this method discretizes first the bounding contour $\partial\Omega \simeq \cup_{m=1}^N \Delta\Omega^{[m]}$ and the time axis $\mathcal{T} = \{t_k \in \mathbb{R}; t_k = k\Delta t, \Delta t > 0, k = 1, 2, \dots, NT\}$. Here, N denotes number of line segments, Δt is the chosen time step and NT stands for the number of time steps. Then, the voltage response at any position on the circuit's rim \mathbf{r} and time t can be found from:

$$V(\mathbf{r}, t) \simeq \sum_{m=1}^N \sum_{k=1}^{NT} v_{[k]}^{[m]} T^{[m]}(\mathbf{r}) T_{[k]}(t) \quad (6.19)$$

where $v_{[k]}^{[m]}$ denotes the unknown voltage expansion coefficient and $T^{[m]}$ is the spatial triangle function while $T_{[k]}(t)$ is the temporal triangle function. The desired voltage distribution is found upon solving the following system of equations with the aid of the marching-on in time technique for any time index p :

$$\begin{aligned} & [\mathbf{I} - \mathbf{Q}_{[0]} + \mathbf{L}_{[0]}] \cdot \mathbf{V}_{[p]} \\ &= \sum_{k=1}^{p-1} [\mathbf{Q}_{[p-k]} - \mathbf{L}_{[p-k]}] \cdot \mathbf{V}_{[k]} + \mathbf{F}_{[p]} \end{aligned} \quad (6.20)$$

A detailed description of \mathbf{Q} and \mathbf{I} arrays can be found in [173]. The influence of a decoupling capacitor is accounted for in a separate $[N \times N \times NT]$ 3D-array \mathbf{L} [175, Ch. 7], which makes it possible to change the parameters of capacitors without the need for re-analyzing the entire structure over again which significantly speeds up the optimization procedure.

Particle Swarm optimization with Variable Number of Dimensions

Conventional PSO algorithm introduced in [51] works with a set of P agents called particles. A particle carries the information about currently tested variables. Every particle moves in the decision space and shares information about quality of visited positions expressed by values of specified objective function. The movement of the p -th particle is determined by:

$$\mathbf{x}_p(i) = \mathbf{x}_p(i-1) + \Delta t \cdot \mathbf{v}_p(i) \quad (6.21)$$

where i denotes current iteration index, Δt stands for time step (usually equal to 1), and $\mathbf{v}_p(i)$ is the velocity of the p -th particle in the i -th iteration which is computed:

$$\begin{aligned} \mathbf{v}_p(i) = & w \cdot \mathbf{v}_p(i-1) + c_1 \cdot r_1 \cdot [\mathbf{pbest}_p - \mathbf{x}_p(i)] \\ & + c_2 \cdot r_2 \cdot [\mathbf{gbest} - \mathbf{x}_p(i)] \end{aligned} \quad (6.22)$$

where w is the so-called inertia weight forcing the particle to continue moving in the current direction. The inertia weight is chosen from interval $\langle 0, 1 \rangle$. Symbols c_1 and c_2 denote the cognitive and social learning factor, respectively. The random values r_1 and r_2 are selected from interval $\langle 0, 1 \rangle$ with a uniform probability distribution. So far the best position revealed by the p -th particle is denoted \mathbf{pbest}_p and \mathbf{gbest} stands for the best from all the \mathbf{pbest} values within the swarm.

The application of the canonical PSO as described in [51] to the problem with finding positions of variable number of capacitors would fail. Therefore, the modified version called PSO-VND as introduced in [93] needs to be used. In general, (6.42-6.43) can contain vectors with up to three different sizes. Current particle size (\mathbf{x}_p) can differ from sizes of \mathbf{pbest}_p and \mathbf{gbest} . Three probabilities p_1, p_2 , and p_3 are introduced in [93] to solve the problem with unbalanced sizes:

- p_1 - probability to adapt to dimension of \mathbf{gbest} ,
- p_2 - probability to adapt to dimension of \mathbf{pbest} ,
- p_3 - probability to adapt to dimension of \mathbf{x}_p .

These probabilities are set so that:

$$p_1 + p_2 + p_3 = 1 \quad (6.23)$$

In case of different sizes, the unit interval is divided to three subintervals with sizes corresponding to chosen p_1, p_2 , and p_3 probability values. Then, a random number from interval $\langle 0, 1 \rangle$ is generated and the dimension to be used is selected according to the subinterval where the random number belongs to. Particles with sizes bigger than the selected one are trimmed. Particles with missing parts are filled with random values or with values previously assigned as improving the objective function value. The selection mechanism using the probabilities is depicted in Fig. 6.14. The current particle size is chosen there. Therefore, the last two components of \mathbf{gbest} are discarded, and the missing part of \mathbf{pbest}_p vector is filled with components from the \mathbf{gbest} vector. For detailed informations about the PSO-VND algorithm we refer the reader to [93].

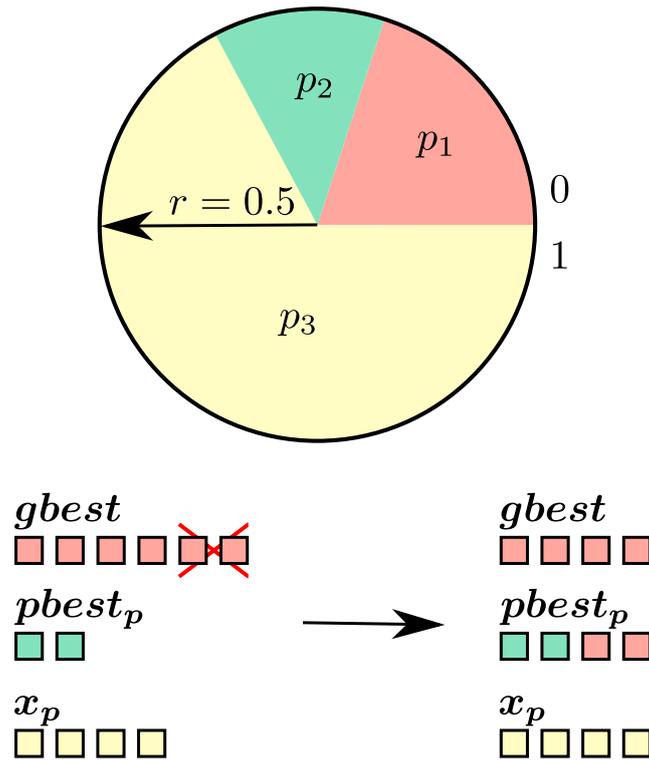


Figure 6.14: Selecting a dimension of particles with different sizes using a user defined probabilities: $p_1 = 0.2$, $p_2 = 0.1$, and $p_3 = 0.7$ in the PSO-VND algorithm.

Numerical Examples

We tested the new approach combining the PSO-VND algorithm with the TD-CIM solver on two sample structures. Namely, we analyze a rectangular and a irregularly-shaped circuit with the dielectric slab of thickness $d = 0.8$ [mm] and relative permittivity $\epsilon_r = 4.5$. For simplicity we assume a non-magnetic material filling with $\mu_r = 1$. The two shapes are shown in Fig. 6.15 and are defined using the following points:

- **rectangular** circuit
 $[0, 0; 0.15, 0; 0.15, 0.1; 0, 0.1]$ [m].
- **irregularly-shaped** circuit
 $[0, 0; 0.12, 0; 0.12, 0.03; 0.075, 0.03; 0.075, 0.07; 0.15, 0.07; 0.15, 0.1; 0.03, 0.1; 0, 0.07]$ [m].

Two ports are located at positions $\mathbf{r} = [0.03, 0.02]$ [m] ('PORT 1') and $\mathbf{r} = [0.12, 0.085]$ [m] ('PORT 2') for the both geometries. They are indicated with the red crosses in Fig. 6.15.

The structure is excited at 'PORT 1' with a bell-shaped pulse defined as [173]:

$$\begin{aligned} \mathcal{I}(t) = A & \left[2 \left(\frac{t}{t_w} \right)^2 \text{H}(t) - 4 \left(\frac{t}{t_w} - \frac{1}{2} \right)^2 \text{H} \left(\frac{t}{t_w} - \frac{1}{2} \right) \right. \\ & + 4 \left(\frac{t}{t_w} - \frac{3}{2} \right)^2 \text{H} \left(\frac{t}{t_w} - \frac{3}{2} \right) \\ & \left. - 2 \left(\frac{t}{t_w} - 2 \right)^2 \text{H} \left(\frac{t}{t_w} - 2 \right) \right] \end{aligned} \quad (6.24)$$

where A denotes the pulse amplitude ($A = 1$ [A]), t_w is the pulse width (defined as the rise time, $t_w = 0.67$ [ns]), and $\text{H}(t)$ stands for the Heaviside unit-step function.

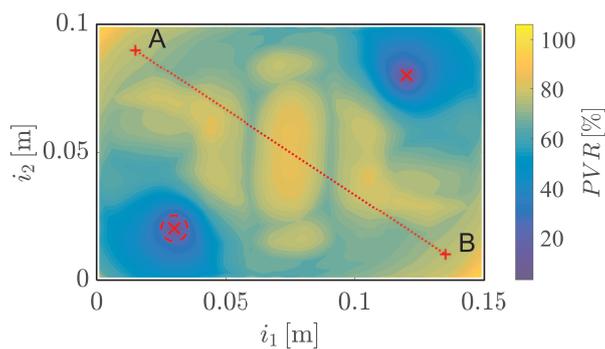
The peak voltage ratio as defined by (6.18) for different positions of the decoupling capacitor with $C = 10$ [nF] is in Fig. 6.15, also. As we can see there, the minimal possible *PVR* value using one 10 [nF] capacitor is approximately 10 [%]. The maximal attenuation values are achieved for positions close to the source of the noise or for positions in the close vicinity of the point that should be decoupled. This is in agreement with the observations published in [105] and [94]. Nevertheless, if we want to further suppress the influence of the noise signal (to ones or tenths of %), we have to use more than one capacitor, definitely.

All the tests were carried out on a personal computer with an AMD Ryzen 7 1700X platform and 32 GB of RAM. A single evaluation of the voltage response took approximately three seconds with our TD-CIM solver in comparison with approximately four minutes using CST Microwave Studio.

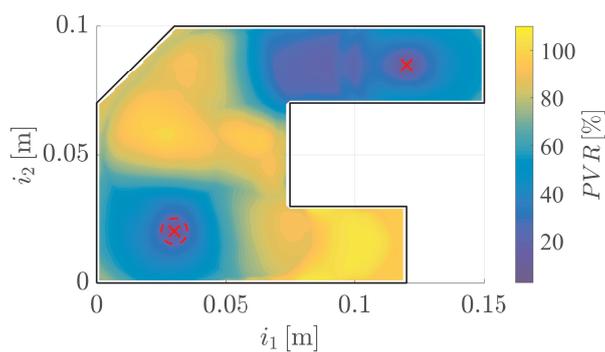
All the optimization tasks were carried out using a FOPS (Fast Optimization Procedures) software which can be downloaded from www.antennatoolbox.com/fops. The PSO-VND algorithm was set with the following parameters:

- number of agents: $nAgents = 50$,
- number of iterations: $nIters = 100$,
- decreasing inertia weight: $w \in \langle 0.9; 0.4 \rangle$,
- cognitive learning factor: $c_1 = 1.494$,
- social learning factor: $c_2 = 1.494$,
- wall boundary: reflecting,
- probabilities: $p_1 = 0.05$, $p_2 = 0.10$, and $p_3 = 0.85$.

The relatively low value of $p_1 = 0.05$ forces particles to maintain their initial dimension. The dimension of ***gbest*** should be adopted only in approximately one case from twenty. Therefore, the diversity in dimension is favored with this settings. All the presented values are the best out of ten repetitions of the algorithm run.



(a) Rectangular circuit.



(b) Irregularly-shaped circuit.

Figure 6.15: The attenuation map for capacitor $C = 10$ [nF] placed on two sample geometries. Red dashed curve denotes a circle used for Fig. 6.16 (a) and some optimization scenarios (capacitors can be placed on the circle only). Dotted line from point A to B denotes positions used for Fig. 6.16 (b).

Influence of ESR and ESL

The ESR (Equivalent Series Resistance) and the ESL (Equivalent Series Inductance) are important parameters influencing behavior of all decoupling capacitor types. The typical values of ESR vary from 10 [m Ω] to 200 [m Ω] while typical values of ESL are from 0.01 [nH] to 1 [nH] according to [104]. Unfortunately, the influence of ESR and ESL cannot be presently added to the simulation using the TD-CIM solver. Therefore, we have modeled the attenuation due to a $C = 10$ [nF] capacitor with various ESR and ESL values using the CST Microwave Studio. We placed the capacitor on subsequent places forming 1) the ring around the 'PORT 1' and 2) line from the top-left corner to the bottom-right corner on the rectangular circuit as shown in Fig. 6.15 (a) with red dashed and dotted curve, respectively. The circle radius is $\rho = 5$ [mm]. The position on the circle is defined:

$$\mathbf{r} = \mathbf{c} + \rho \cdot [\cos(\theta), \sin(\theta)] \quad (6.25)$$

where \mathbf{c} denotes the position of the noise source ('PORT 1') and θ is the angle from the \mathbf{i}_1 axis going from 0 to 2π . The line goes from point $\mathbf{r}_A = [0.015, 0.09]$ [m] to point $\mathbf{r}_B = [0.135, 0.01]$ [m]. Any position on the line can be defined by parameter s going from 0 to 1:

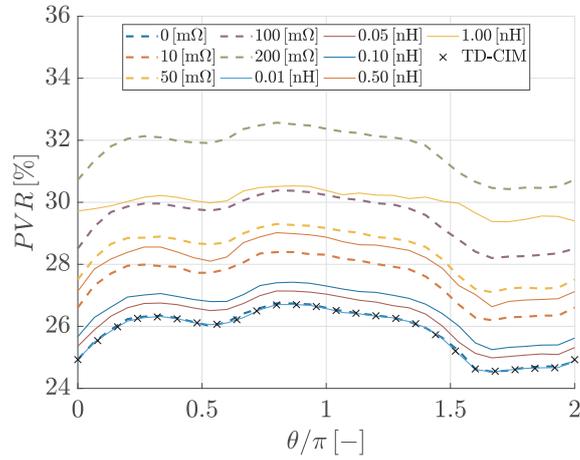
$$\mathbf{r} = s(\mathbf{r}_B - \mathbf{r}_A) \quad (6.26)$$

As we can see in Fig. 6.16, the peak voltage ratio values are different for results obtained by CST Microwave Studio and TD-CIM. What is important, the trend of all obtained curves is the same. The shape of the attenuation curves is slightly changed only for the highest value of $ESL = 1$ [nH]) Nevertheless, the minimum and maximum values can be found for the same angles θ and parameters s for all the curves. Therefore, the influence of ESR and ESL does not affect the capacitor position assuming a linear correction of the required attenuation level value L defined by the user. Considering the time necessary to compute the voltage response of the structure by both the solvers and the influence of ESR and ESL values on the decoupling properties for different places on a PCB, we prefer to use the TD-CIM based solver for the optimization experiments that follow.

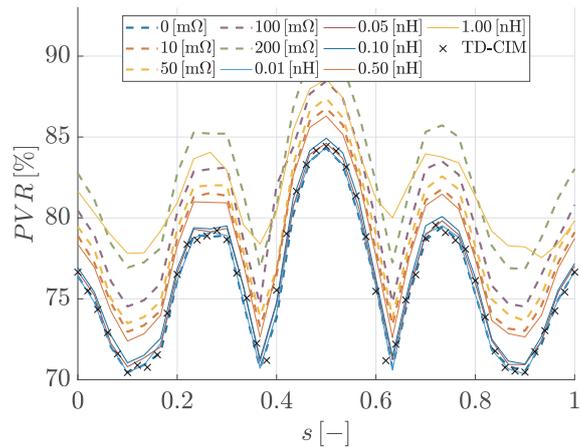
Placing Capacitors Anywhere

First, we used the whole surface of the PCB to find the optimal set of decoupling capacitors to reach peak voltage ratio $PVR = 0.05$ (approximately -13 [dB]). Capacitors with a fixed value of $C = 10$ [nF] can be placed at any position at a distance more than 1.0 [mm] far off the circuit's periphery and the ports. The maximum number of capacitors was set to $N_{max} = 7$ so the decision space size was any value from the set $\{2, 4, \dots, 14\}$. To give an example, discrete variables with 10 samples per a variable would lead to $10^2 + 10^4 + \dots + 10^{14} = 1.01 \times 10^{14}$, which is the enormous number of combinations.

The results are summarized in Table 6.5. As we see there, the optimizer assigned the optimal number of capacitors to be $N = 4$ for both the PCB shapes. The voltage responses with and without the set of decoupling capacitors are shown in Fig. 6.17. Here, we see that the voltage response was attenuated to exactly 0.05 level of the voltage response without any capacitor on the board. Interesting are positions of capacitors depicted in Fig. 6.17 and summarized in Table 6.5. One capacitor is placed very close to the source of the noise or to the place of interest. According to Fig. 6.15, placing a single capacitor here attenuates the voltage response approximately ten times. Other capacitors are then

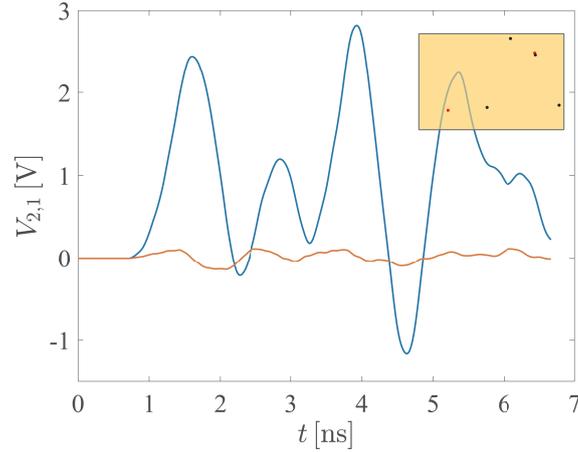


(a) Circle curve.

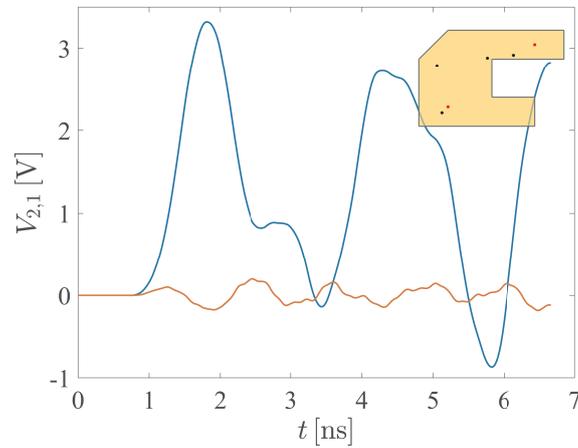


(b) Line segment.

Figure 6.16: Influence of ESR and ESL on the peak voltage ratio considering the capacitor position on the rectangular circuit ($C = 10$ [nF]). Results computed by CST Microwave Studio depicted with curves (solid for non-zero ESL values, dashed for non-zero ESR), results depicted with black cross markers computed by the TD-CIM (no ESR and ESL). The position of capacitor is moving on (a) circle around 'PORT 1' (parameter θ), (b) line from top-left corner to bottom-right corner (parameter s).



(a) Rectangular circuit.



(b) Irregularly-shaped circuit.

Figure 6.17: The voltage response of the example PCBs with (red line) and without (blue) decoupling capacitors optimized for peak voltage ratio $PVR = 0.05 [-]$. The ports are plotted with red dots and the capacitors with black dots in sub-figures depicting geometry of PCBs.

placed to provide exactly chosen L . The better fitness function value f was achieved in case of the irregularly-shaped circuit according to Table 6.5. Though geometrically more complex, the PSO-VND algorithm could find a good candidate on the smaller surface in the early stage of the algorithm run. Then, the algorithm is capable of exploiting the area of the optimum in a great detail, which leads to a very low value of the objective function.

Placing Capacitors on a Circle

We can simplify the optimization task as much as possible knowing that the maximal attenuation can be achieved placing the capacitor to a close vicinity of the element producing the noise or the place of interest. The number of dimensions can be further minimized by regularizing the position of capacitors. We let the optimizer to place capacitors only on a

Table 6.5: Resulting sets of decoupling capacitors with fixed value placed at any position on the rectangular and irregularly-shaped circuit to reach peak voltage ratio $PVR = 0.05 [-]$ and corresponding fitness function values.

	rectangular	irregularly-shaped
r_1 [mm]	{70.05, 22.96}	{23.81, 13.88}
r_2 [mm]	{120.62, 78.04}	{70.43, 71.02}
r_3 [mm]	{95.10, 95.00}	{18.67, 63.00}
r_4 [mm]	{145.00, 25.26}	{98.15, 74.04}
f [-]	2.0×10^{-5}	8.8×10^{-6}

circle with a radius $\rho = 5$ [mm] (depicted with the red dashed line in Fig. 6.15). Then, the position used to evaluate the fitness function (6.17) of any capacitor is computed using (6.25).

The maximal number of capacitors was $N_{max} = 7$ so the decision space vector could have seven different dimensions. The desired peak voltage ratio was set to be $L = 0.01 [-]$ and $L = 0.05 [-]$, respectively.

The obtained optimal angles and resulting fitness function values are summarized in Table 6.6. Defining the capacitor's position using only angle θ we spare one variable per one capacitor, which significantly reduces the decision space size and, in addition, it results in better values of fitness functions. Moreover, the optimizer is able to find a less expensive solution with only three capacitors to achieve 0.05 peak voltage ratio for the rectangular PCB as depicted in Fig. 6.18. The irregularly-shaped PCB has slightly larger PVR values in the close vicinity of the 'PORT 1' as depicted in Fig.6.15. Therefore, four capacitors have to be used. For the lower 0.01 peak voltage ratio, five capacitors are needed for the both rectangular and irregularly-shaped circuits.

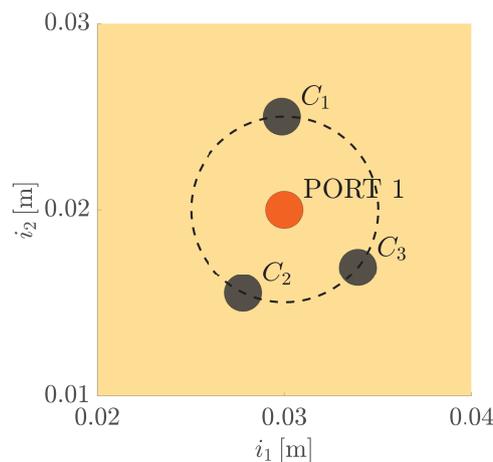


Figure 6.18: The capacitors placed around 'PORT 1' on the circle with a radius $\rho = 5$ [mm] optimized for an attenuation level $L = 0.05 [-]$.

Table 6.6: Resulting sets of decoupling capacitors placed on the circle with a radius $\rho = 5$ [mm] around the noise source on the rectangular and irregularly-shaped circuits. The requested peak voltage ratio is $PVR = 0.01$ [-] and $PVR = 0.05$ [-], respectively.

L [-]	rectangular		irregularly-shaped	
	0.01	0.05	0.01	0.05
θ_1 [rad]	0	1.597	0	0
θ_2 [rad]	2.026	4.257	1.429	1.832
θ_3 [rad]	2.746	5.612	3.836	4.394
θ_4 [rad]	3.947	–	4.247	4.922
θ_5 [rad]	5.212	–	5.868	–
f [-]	7.1×10^{-8}	2.5×10^{-5}	1.7×10^{-8}	3.0×10^{-10}

Searching for Position and Capacitance

A more general optimization task is to search for both positions and values of individual capacitors. In the last example, we searched for the position of an unknown number of capacitors on the circle as in the previous subsection (by means of angle θ) and capacitance values (C). Only standard E-6 series capacitance values from interval $\langle 0.1$ [nF]; 68 [nF] \rangle can be chosen. We have optimized the set of decoupling capacitors only for the attenuation level $L = 0.01$ [-]. All other optimization parameters were set as described in the previous subsection. The results of the optimization for both the rectangular and irregularly-shaped structures are summarized in Table 6.7.

The results show that only four capacitors can be used to attenuate the voltage response to $L = 0.01$ [-] for the rectangular circuit. In comparison, we needed five components when we used only capacitors with the fixed capacitance value $C = 10$ [nF] (see the corresponding column in Table 6.6). As we can see from the optimal angles θ and corresponding values C , the largest capacitance values ($C = 15$ [nF]) are used in areas having the largest impact on the attenuation as seen in Fig. 6.16 (a). Smaller capacitance values are than used to balance the desired attenuation level, which is indicated by the low objective function value $f = 1.3 \times 10^{-5}$ [-].

Five capacitors are needed in case of irregularly-shaped structure as in the case of optimization with fixed capacitance value. The objective function value $f = 4.1 \times 10^{-6}$ [-] is much worse than in case of the fixed capacitance optimization ($f = 1.7 \times 10^{-8}$ [-]). This is caused by the fact that despite the same computational resources allocated to the optimization (50 agents and 100 iterations for both cases), the possible decision space sizes are now much larger. In the present case we work with decision space sizes $\{2, 4, \dots, 14\}$ while available decision space sizes were only $\{1, 2, \dots, 7\}$ in the case of fixed capacitance values.

Conclusion

We have formulated the decoupling capacitor placement problem as an optimization task with variable number of dimensions. This approach makes it possible to solve the problem without knowing a priori the number of capacitors needed to reduce the noise received at

Table 6.7: Resulting sets of decoupling capacitors with variable values placed on the circle with a radius $\rho = 5$ [mm] around the noise source on the rectangular and irregularly-shaped circuits. The requested peak voltage ratio is $PVR = 0.01$ [-].

n	rectangular		irregularly-shaped	
	θ [rad]	C [nF]	θ [rad]	C [nF]
1	0.668	15.0	0.150	10.0
2	1.986	4.7	0.869	22.0
3	3.492	3.3	1.480	3.3
4	5.817	15.0	2.172	3.3
5	—	—	3.703	10
f [-]	1.3×10^{-5}		4.1×10^{-6}	

a chosen place on a PCB to the requested level. We have tested the proposed approach on two sample PCBs considering three different options: placing capacitors with fixed value everywhere on a PCB, placing capacitors with fixed value on a ring near the noise source, and placing various capacitors on the ring. We have demonstrated that thanks to the reduction of the decision space the choice of the problem formulation has an significant impact on the efficiency of the optimization procedure. We have further shown that the tested approach combining power of the PSO-VND optimization algorithm with the TD-CIM solver can be used as a very fast and robust prototyping tool. For final verification of the optimized design, another simulator that can take into account *ESR* and *ESL* values is necessary.

Acknowledgments

The research reported in this paper was financed by the Czech Science Foundation under Grant 17-05445Y and by the Czech Ministry of Education, Youth and Sports under Grant LO1401.

6.4 Design of a Linear Antenna Array: Variable Number of Dimensions Approach

Originally published as:

Kadlec, P., Čapek, M., Rýmus, J., Marek, M., Štumpf, M., Jelínek, L., Mašek, M., Kotalík, P.: Design of a Linear Antenna Array: Variable Number of Dimensions Approach. In *In Proceedings of 30th International Conference Radioelektronika 2020*, 1-5, 2020.

Abstract

A linear antenna array design is usually formulated as a binary optimization task. A global optimization algorithm is then used to turn on/off a fixed number of elements positioned on a uniform grid. This paper introduces a new formulation of the problem: the optimizer searches for distances between individual elements and their total amount at the same time. Such a formulation has to be solved using the optimization with a variable number of dimensions that enables to work with decision space vectors having different lengths. Here, the Particle Swarm Optimization algorithm is used to solve different formulations of the linear antenna array design problem. The “uniform grid” formulation is compared with the “variable number of dimensions” formulation on several optimization tasks encompassing the minimization of selected antenna parameters: the Side-Lobe Level, the First Null Beam Width and the number of used (active) elements.

Introduction

A large number of design problems (not limited to the domain of electromagnetics) involves the decision about amount of some components to be used. The number of components is in fact one of the design variables with a potentially huge impact on the optimized metric (e.g. realized gain, directivity), but a cost and a complexity of the system as well. This is true especially when a designer asks to himself: “How many of ... is necessary to achieve ...?”. This behavior is characteristic for the following problems: the sensor coverage problem [48], the composite laminate stacking problem [114], the short cantilever design [101], the printed circuit board decoupling capacitors placement [85], the dielectric layered filter design [95], the high-dimensional data clustering [164, 185], the cancer marker identification [135] etc. All the aforementioned tasks can be included into the so-called family of variable number of dimensions (VND) optimization problems (referred as variable length problems by some authors).

The antenna array design can be viewed as the VND problem, also. Usually, set of required antenna array parameters is defined and a designer is then asked to guess the optimal number of array elements based on his own experience or a repetitive search for individual dimensions. Another possibility is to formulate the design task as a binary optimization problem [139]. Then, the problem involves a large number of elements positioned with use of a uniform grid (UG) and every variable from the fixed length binary decision space vector switches the corresponding element either on or off.

There is an enormous number of research papers using the UG approach in a cooperation with a global optimization algorithm to solve the antenna array synthesis problem. As the UG formulation involves use of binary variables, most of the authors utilize the

Genetic Algorithms using both the uniform [67, 72, 196] and non-uniform [22, 52, 116] positioning of elements. Also, other global optimization algorithms are utilized to solve this problem: e.g. Particle Swarm Optimization (PSO) [65, 75, 76], Differential Evolution [64] etc. The most important design objectives are the following: the side-lobe level (*SLL*), the first null beam-width (*FNBW*) and number of used elements (ν). Goal of the design is to minimize all these three objectives which leads to a multi-objective optimization because they are contrary to each other [63, 207].

All the UG formulations are suffering from the a priori made choice about possible positions for the individual antenna elements. They define an exponential number of discrete combinations of positions on a relatively large-sized N -dimensional space that contains more suitable positions with a high probability. Only minor efforts were put to try some relaxation techniques as provided in [57] to overcome this issue.

In this paper, we introduce several VND formulations of the linear antenna array design problem. The conventional UG formulations search for the proper combination of fixed number of excitation amplitudes (usually equal to zero or one). On the contrary, the proposed VND formulations search for distances between individual elements and their optimal amount, simultaneously. The VND-formulated problems are solved here using the VND variant of the PSO algorithm introduced in [93]. Results of both the single-objective and multi-objective VND problems are compared to results of conventional UG formulations solved by the conventional single-objective [51] and the multi-objective [159] PSO algorithm.

Linear Antenna Array Design: Problem Formulations

We consider a linear array consisting of ν same antennas distributed alongside the x -axis as shown in Fig. 6.19. Generally, radiation properties of such an array can be analyzed by means of the so-called radiation vector. The full derivation of the radiation vector can be found e.g. in [144]. If $\mathbf{F}_i(\mathbf{k})$ stands for the radiation vector of the i -th element, then the total radiation vector reads:

$$\mathbf{F}_{\text{tot}}(\mathbf{k}) = \mathbf{F}_1 + \mathbf{F}_2 + \dots + \mathbf{F}_\nu \quad (6.27)$$

where $\mathbf{k} = k\mathbf{r}$ is the wave number vector, k stands for the free space wave number ($k = 2\pi/\lambda_0$ with λ_0 standing for the free space wavelength) and \mathbf{r} denotes the position vector. If all elements of the array have the same radiation vector $\mathbf{F}(\mathbf{k})$ then (6.27) can be simplified to:

$$\mathbf{F}_{\text{tot}}(\mathbf{k}) = \mathbf{A}(\mathbf{k})\mathbf{F}(\mathbf{k}) \quad (6.28)$$

Here, $\mathbf{A}(\mathbf{k})$ stands for the array factor which is determined by the array configuration (positions and feeding of individual elements of the array). The total array factor reads:

$$\mathbf{A}(\mathbf{k}) = \sum_{i=1}^{\nu} a_i \exp(j\mathbf{k} \cdot \mathbf{d}_i) \quad (6.29)$$

where j denotes the imaginary unit and a_i and \mathbf{d}_i denotes the complex excitation (amplitude and phase) and the position of the i -th element, respectively. The array factor can be easily expressed in the spherical coordinates θ (the polar angle) and ϕ (the azimuth angle) using the conversion from the Cartesian coordinate system. Only the polar cut $\theta = \pi/2$ is considered in the remainder of the text for the sake of simplicity.

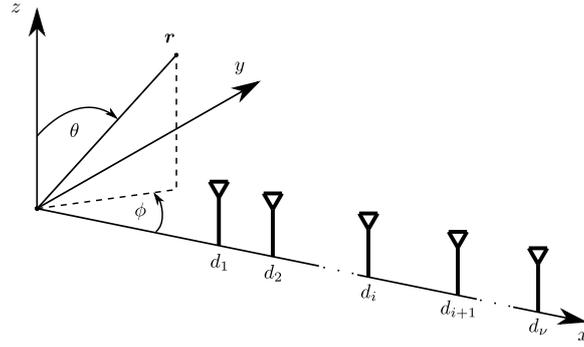


Figure 6.19: Linear antenna array consisting of ν antennas.

The design requirements for an antenna array contain parameters SLL , $FNBW$ and ν . All the design parameters of the antenna array are to be minimized. Please see the definition of the SLL and $FNBW$ parameters in Fig. 6.20.

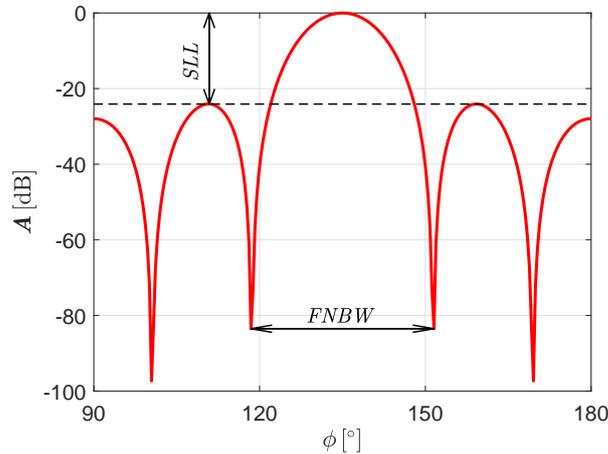


Figure 6.20: Antenna factor \mathbf{A} as a function of the azimuth angle ϕ : a definition of the SLL and $FNBW$ parameters.

The antenna array design problem can be formulated as a single-objective or a multi-objective problem. Both the options can use either a fixed number of decision space variables or variable number of dimensions. The combinations of the problem formulations used in this study are listed in the following subsections. The mutual coupling of individual elements is not considered in this study. However, some computational techniques from [40] could be incorporated to the computational apparatus without loss of generality.

Uniform Grid Formulations

The first type of formulation uses a uniform grid for positions of elements. This approach leads to a binary optimization problem with decision space vector of fixed length. Every component of the vector is equal to either 1 (an element is placed here) or 0 (the position is left empty) as shown in Fig. 6.21. The drawback of this approach is that the feasible space has to be sampled to a very dense grid in order to keep a satisfactory resolution for the design requirements which leads to an enormous number of possible combinations

(2^n) .

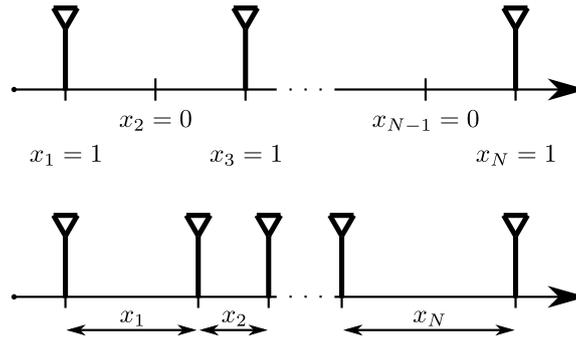


Figure 6.21: Linear antenna array, comparison of the UG formulation (top) and VND formulation (bottom).

In our case, we use a grid with positions for up to 60 antenna elements. A distance between two grid positions is $0.25\lambda_0$. Every variable from the decision space vector can be equal to 0 or 1: $x_i = \{0, 1\}$, $\forall i \in \{1, 2, \dots, 60\}$ which means that $x_i = a_i$ with respect to (6.29) (see the top part of Fig. 6.21). Using 60 decision variables, an optimization algorithm searches within $2^{60} \simeq 1.15 \times 10^{18}$ possible combinations. Assuming that computation of the objective functions for a single combination lasts for 1 ns (which is very unlikely for the real-world simulations), it would still take us almost 37 years to evaluate all the possible combinations.

The remainder of this subsection defines in total four single-objective and two multi-objective UG formulations.

Problem 1 (SO: min SLL)

$$f = SLL(\mathbf{x}) \quad (6.30)$$

Problem 2 (SO: $SLL = -20$ dB)

$$\begin{aligned} f &= [SLL(\mathbf{x}) - L]^2, \\ L &= -20 \text{ dB} \end{aligned} \quad (6.31)$$

Problem 3 (SO: min SLL , min $FNBW$)

$$f = SLL(\mathbf{x}) + FNBW(\mathbf{x}) \quad (6.32)$$

Problem 4 (SO: min SLL , min $FNBW$, min ν)

$$f = SLL(\mathbf{x}) + FNBW(\mathbf{x}) + \nu(\mathbf{x}) \quad (6.33)$$

Problem 5 (MO: min SLL , min $FNBW$)

$$\begin{aligned} f_1 &= [SLL(\mathbf{x}) - L]^2 \\ f_2 &= \nu(\mathbf{x}) \end{aligned} \quad (6.34)$$

Problem 6 (MO: min SLL , min $FNBW$, min ν)

$$\begin{aligned} f_1 &= SLL(\mathbf{x}) \\ f_2 &= FNBW(\mathbf{x}) \\ f_3 &= \nu(\mathbf{x}) \end{aligned} \quad (6.35)$$

Variable Number of Dimensions Formulations

The second family of the problem formulations involves the VND type. It means that decision space vectors with different sizes n can be used during a single optimization run. In this study, the first antenna of the array is placed at the position $x = 0$ m. The i -th variable of the decision space vector is then the distance between i -th and $(i + 1)$ -th element of the array as shown in Fig. 6.21. It means that number of antenna elements is $\nu = n + 1$ for all the VND formulations. The distances between individual elements can vary according to the interval $x_i/\lambda_0 \in \langle 0.25, 1 \rangle$ to keep fair the comparison with the regular grid formulations.

Problem 7 (VNDSO: min SLL)

$$f = SLL(\mathbf{x}, n) \quad (6.36)$$

This formulation is fully comparable with Problem 1. The VND pairings for UG Problems 2-6 are as follows:

Problem 8 (VNDSO: $SLL = -20$ dB)

$$f = [SLL(\mathbf{x}, n) - L]^2 \quad (6.37)$$

Problem 9 (VNDSO: min SLL , min $FNBW$)

$$f = SLL(\mathbf{x}, n) + FNBW(\mathbf{x}, n) \quad (6.38)$$

Problem 10 (VNDSO: min SLL , min $FNBW$, min ν)

$$f = SLL(\mathbf{x}, n) + FNBW(\mathbf{x}, n) + \nu(\mathbf{x}, n) \quad (6.39)$$

Problem 11 (VNDMO: min SLL , min $FNBW$)

$$\begin{aligned} f_1 &= [SLL(\mathbf{x}, n) - L]^2 \\ f_2 &= \nu(\mathbf{x}, n) \end{aligned} \quad (6.40)$$

Problem 12 (VNDMO: min SLL , min $FNBW$, min ν)

$$\begin{aligned} f_1 &= SLL(\mathbf{x}, n) \\ f_2 &= FNBW(\mathbf{x}, n) \\ f_3 &= \nu(\mathbf{x}, n) \end{aligned} \quad (6.41)$$

Optimization Techniques

The Particle Swarm Optimization algorithm mimics a co-operative behavior of a set of individuals to maximize collective profits. Basic properties of the PSO algorithm and its respective variants used to solve the antenna array design problems defined in Section 6.4 are briefly reviewed in the following sub-sections.

Single-Objective PSO

A commonly used version of the single-objective Particle Swarm Optimization (SOPSO) can be found in [163]. It uses a set of decision space vectors called particles that iteratively change their positions to improve their objective function value. The position of the p -th particle at the iteration i reads:

$$\mathbf{x}_p(i) = \mathbf{x}_p(i-1) + \Delta t \mathbf{v}_p(i), \quad (6.42)$$

where the time step $\Delta t = 1$ (for the sake of completeness) and $\mathbf{v}_p(i)$ stands for the velocity of the p -th particle. The velocity is updated according to:

$$\begin{aligned} \mathbf{v}_p(i) = & w\mathbf{v}_p(i-1) + c_1 r_1 [\mathbf{pb}_p - \mathbf{x}_p(i)] \\ & + c_2 r_2 [\mathbf{gb} - \mathbf{x}_p(i)]. \end{aligned} \quad (6.43)$$

Here w denotes the so-called inertia weight, c_1 and c_2 stand for the cognitive and the social learning factor, respectively. Values r_1 and r_2 are chosen randomly from the interval $\langle 0, 1 \rangle$ with the uniform distribution of probability. Symbol \mathbf{pb}_p stands for the best position found by the p -th particle in previous iterations. Finally, \mathbf{gb} denotes the best position from the set of \mathbf{pb} values for the whole swarm.

The user-defined parameters w , c_1 and c_2 determine the trade-off between exploration and exploitation of the algorithm. Recommended values for these parameters are based on numerous comparative benchmark studies (see e.g. [158]): $w = 0.9$ and $c_1 = c_2 = 1.5$.

In this paper, simple binary variant of the PSO algorithm is used to solve the binary UG problems. Before the decision space vector is to be evaluated by the objective function its components are set either to be 0 or 1 with the threshold value 0.5. Then, boundary conditions preventing particles to reach positions outside the feasible part of the decision space Ω can be left out.

Multi-Objective PSO

Plenty of multi-objective modifications of the PSO algorithm (MOPSO) can be found in the open literature. This study uses the MOPSO variant presented in [159]. The only difference between single-objective and multi-objective PSO is the choice of \mathbf{gb} to update the actual velocity vector. The velocity update formula (6.43) slightly changes to:

$$\begin{aligned} \mathbf{v}_p(i) = & w\mathbf{v}_p(i-1) + c_1 r_1 [\mathbf{pb}_p - \mathbf{x}_p(i)] \\ & + c_2 r_2 [\mathbf{gb}_p - \mathbf{x}_p(i)]. \end{aligned} \quad (6.44)$$

Vector \mathbf{gb}_p for the p -th particle is selected randomly from the so-called external archive. The external archive contains the non-dominated solutions so far found by the algorithm (candidates for the Pareto-optimal front). It is also possible to select the closest solution from the external archive by means of the Euclidean distance in the decision space. Unfortunately, this approach would lead to premature convergence of the algorithm. Therefore, a combination of these two approaches is used to benefit from both.

Single-Objective VNDPSO

The conventional SOPSO algorithm works with a set of decision space vectors that have the same number of components (or dimensions if you want). Nevertheless, the conventional SOPSO algorithm was extended to handle problems with a variable number

of dimensions in [93]. The single-objective VNDPSO (SOVNDPSO) follows exactly the same procedures as SOPSPO except the random generation of the initial population and computation of the velocity vector. First, user defines the list of dimensions $\Gamma = \{N_1, N_2, \dots, N_D\}$ that are feasible for the problem. Limits Ω_d have to be defined for all the dimensions up to N_D . Then, the dimension is picked randomly from the Γ list for every particle \mathbf{x}_p . If possible (i.e. $P \geq D$) all the dimensions from the list should be distributed uniformly. Letting particles to have different dimensions becomes a problem in the velocity update formula (6.43). There are three vectors namely \mathbf{gb} , \mathbf{pb}_p and \mathbf{x}_p that can all have a different number of elements N_g , N_p , and N_x , respectively. Therefore, three user-defined probabilities are introduced: p_1 (probability to use N_g), p_2 (probability to use N_p), and p_3 (probability to use N_x).

The sum of all the probabilities is set to equal 1. In case the three components in (6.43) have not the same dimension, random value $r \in \langle 0, 1 \rangle$ is generated. The dimension for the particle in the next iteration is then selected according to:

$$N_x = \begin{cases} N_g, & \text{if } 0 < r \leq p_1 \\ N_p, & \text{if } p_1 < r \leq p_1 + p_2 \\ N_x, & \text{if } p_1 + p_2 < r \end{cases} \quad (6.45)$$

Once the new number of dimensions N_x is determined, vectors \mathbf{gb} , \mathbf{pb}_p and \mathbf{x}_p with a higher dimension are shortened to size of N_x by deleting last elements from the vector. On the other hand, if any of the vectors is shorter than N_x , then the missing elements are either generated randomly or taken from the existing \mathbf{pb} having at least N_x elements. The second approach enhances a convergence speed of the algorithm if the optimal elements position does not change with the change of dimension.

Multi-Objective VNDPSO

Multi-objective PSO with a variable number of dimensions (MOVNDPSO) follows procedure of the conventional PSO. The strategy for \mathbf{gb} and \mathbf{pb} updating is the same as for MOPSPO algorithm (see Sec. 6.4). The problem with variable number of elements in the velocity update formula is solved the same way as presented in the SOVNDPSO algorithm: see (6.45) in Sec. 6.4.

Numerical Examples

Throughout all the simulations presented in this Section, we use the PSO-based algorithms with the following user-defined parameters: size of the swarm $n_a = 200$, number of iterations: $n_i = 100$, learning factors: $c_1 = c_2 = 1.5$, inertia weight: $w = 0.9$, random global best parameter: $r_{gb} = 0.7$, dimension probability vector: $\mathbf{p} = [0.90, 0.05, 0.05]$. Every problem has been solved 100-times with respect to a stochastic nature of the optimization processes. A single evaluation of the array factor took approximately 20 ms. Therefore, a single run of the optimization (regardless of the Problem type) took approximately 400 s.

Single-objective Problems

Here, we compare results of the UG single-objective problems (Prob. 1-4) solved by the SOPSPO algorithm with the VND formulations (Prob. 7-10) solved by SOVNDPSO. The

results of SLL , $FNBW$ and ν (number of elements) are summarized in Table 6.8). The data are presented in form of standard boxplots in Fig. 6.22. We can see, that significantly lower SLL values are achieved for the problems minimizing this parameter with a uniform grid ($-40.21 \text{ dB} \leq SLL_{\text{avg}} \leq -39.73 \text{ dB}$) compared to the VND problems ($-33.13 \text{ dB} \leq SLL_{\text{avg}} \leq -32.17 \text{ dB}$). Also, the results of the $FNBW$ metric are better in case of the UG formulations: $FNBW_{\text{avg}} = 3.10^\circ$ for Problems 3 and 4 while $FNBW_{\text{avg}} = 6.00^\circ$ for the VND Problems 9 and 10. The better performance of arrays designed by the UG formulations is paid dearly by a massive increase of elements used as $\nu_{\text{avg}} = 49$ for the UG formulations while the VND formulations use only $\nu_{\text{avg}} = 27$ elements on the average.

The trade-off between the quality of the array parameters and the number of elements vanishes when we ask the optimizer to find an array with a particular level of some parameter: SLL in case of Problem 2 (UG) and 8 (VND). As evidenced by the corresponding boxplots in Fig. 6.22, both the formulations were able to find a solution with exactly matching level of the SLL parameter $L = -20 \text{ dB}$ with almost zero variance. But, the VND formulation uses only 24 antenna elements on the average while the UG formulation wastes up to 39 elements to reach the same level of SLL . These results indicate that the conventional single-objective optimizer searches for results in a very good agreement with the objective function formulation but in a sub-optimal dimension.

Table 6.8: Comparison of the antenna array parameters SLL , $FNBW$, and ν for the UG and VND formulations.

Problem #	1	2	3	4	7	8	9	10
	UG				VND			
SLL_{avg} [dB]	-40.2	-20.0	-39.8	-39.7	-33.1	-20.0	-32.2	-32.2
$FNBW_{\text{avg}}$ [dB]	-	-	3.1	3.1	-	-	5.9	6.1
ν_{avg} [-]	48.0	39.0	49.0	50.0	27.0	24.0	27.0	26.0

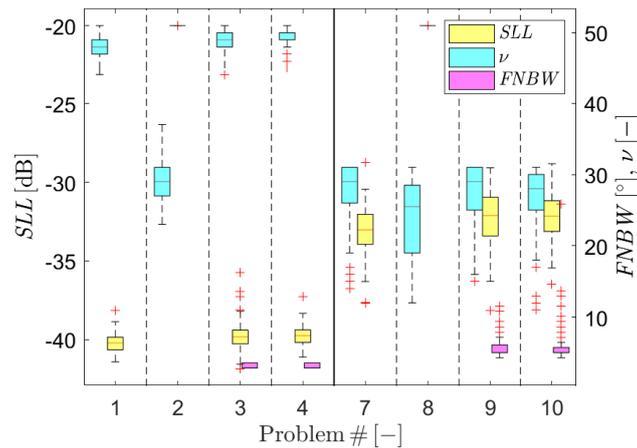


Figure 6.22: Standard boxplots of the antenna array parameters for the UG (Problems 1-4) and VND (Problems 7-10) formulations. The parameter SLL uses the left axis, while the parameters $FNBW$ and ν use the right axis. The vertical solid line separates the UG problems from the VND ones.

Multi-objective Problems

The multi-objective test suite consists of two problems for both the UG and VND formulations: the two-dimensional (SLL and $FNBW$ minimization, see Prob. 5 and 11) and the three-dimensional (minimizing of the number of antenna elements ν is added to the two objectives, see Prob. 6 and 12). All decision space vectors assigned by the MOPSO and VNDMOPSO algorithms as the Pareto-optimal during 100 runs of the 2D problems are compared in Fig. 6.23. We can see that while the UG vectors are better in the SLL values, the VND vectors achieve desired smaller values of the $FNBW$ parameter. Nevertheless, a major part of the Pareto-optimal set for the UG formulation is dominated by the solutions obtained via the VND formulation.

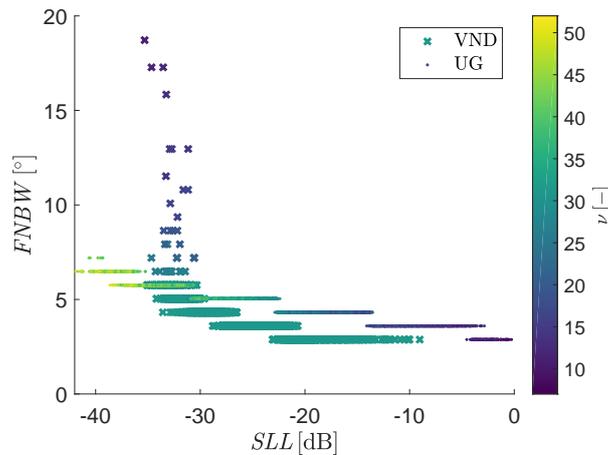


Figure 6.23: Pareto-optimal solutions found for the two-objective Problems 5 and 11. The color of the markers indicates the number of elements needed.

On top of it, the color of the used markers in Fig. 6.23 indicating the number of elements proves the inefficiency of the UG formulation. The Pareto-optimal VND solutions are limited up to approximately 25 or 30 elements while the UG non-dominated solutions use more than 50 elements to build arrays with the best SLL values. The decrease of the SLL value is strongly connected with the number of elements used. The UG formulation is favored here because it can use up to 60 elements compared to maximally 30 elements for the VND formulation. The VND formulation enables the optimizer to set the distances among consecutive elements freely. Nevertheless, the UG formulation has to search in the decision space with a much higher dimension. As a result, the MOPSO optimizer is unable to find the best trade-off solutions using the same computational effort as in case of the VNDMOPSO optimizer.

The two-objective formulations considered minimization of the SLL and $FNBW$ parameters only. Hence, it is not absolutely fair to compare the results of UG and VND formulations from the perspective of the number of elements used. The three-objective Problems 6 and 12 add the number of elements ν as the third objective. The Pareto-optimal solutions found by the MOPSO and VNDMOPSO optimizers, respectively, are shown in Fig. 6.24. It is obvious, that MOPSO solving the UG formulation is able to find more trade-off solutions that occupy a larger area of the three-dimensional objective space. But the extreme solutions are not very good design candidates because they meet

the requirements of a single-objective only. The VNDMOPSO algorithm clearly outperforms the MOPSO algorithm as it dominates the central part of the objective space (the trade-off part). Moreover, no member of the VND Pareto-optimal set is dominated by any member of the UG set.

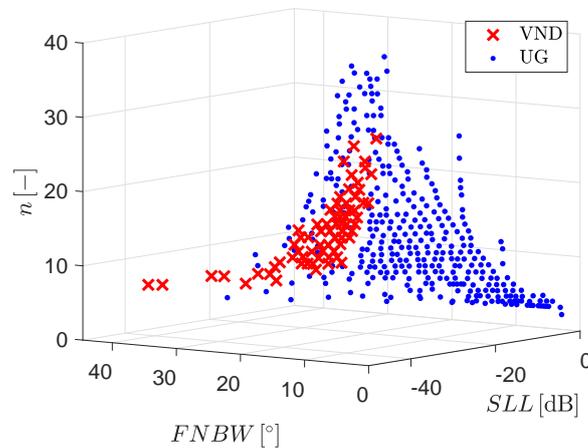


Figure 6.24: Three-dimensional Pareto-optimal solutions considering the array parameters SLL , $FNBW$, and ν found using the UG and VND formulation, respectively.

Conclusions

The paper reviews possible formulations of the linear antenna array synthesis using global optimization methods. The single-objective and multi-objective PSO algorithms and their variants for optimization problems with a variable number of dimensions are briefly reviewed. All the antenna array problem formulations can be split up into two categories: 1) the uniform grid formulations, and 2) the variable number of dimensions. The UG formulation outperforms the VND one for the single-objective problems (both in the SLL and $FNBW$ parameter) at the cost of a significant increase of the elements needed. This is much more obvious when the optimizer is not asked to reach the possible limit of a certain parameter but a certain value of it. In that case, the UG formulation seems to be highly ineffective. The VND formulation clearly outperforms the UG one in case of the multi-objective optimization designs especially when it takes a number of elements used as one of the objectives together with the SLL and $FNBW$ parameters. The performance of the UG formulations is limited by the a priori chosen positions of elements while the VND formulations can distribute them more freely.

Acknowledgments

The research reported in this paper was financed by the Technology Agency of the Czech Republic under Grant TH04010373 and by the Czech Science Foundation under Grant 19-06049S.

Chapter 7

Teaching Experiences

This chapter summarizes my activities in the educational domain. Teaching is for me a corner stone of the research continuity. I see the teaching as a great option for the researcher to think about the aspects of the discussed subject from different angles, because the teacher is forced by the students to use other ways to explain something or find other ways to solve problems.

7.1 Subjects

Fundamentals of Television Technology

After I started my Ph.D. studies in August 2009, I started to teach laboratory classes for the subject Fundamentals of Television Technology (BZTV, Základy televizní techniky in Czech) for Bachelor's students. The course was guaranteed by prof. Stanislav Hanus. In total, I thought these exercises for three years till 2011.

Electromagnetic waves and antennas

In the winter term of the academic year 2012/2013, I was teaching the Computer exercises of the subject Electromagnetic waves and antennas (BEVA, Elektromagnetické vlny, vedení a antény) guaranteed by prof. Zbyněk Raida. The course was thought for Bachelor's students.

CAD in microwave techniques

In the winter term of the academic year 2011/2012, I was teaching the Computer exercises of the subject CAD in microwave techniques (MCVT, CAD v mikrovlnné technice) guaranteed by prof. Zbyněk Raida. The course was dedicated for Master's students.

Electromagnetic compatibility

I was teaching the laboratory exercises of the subject Electromagnetic compatibility (BEMC, Elektromagnetická kompatibilita) guaranteed by Jiří Dřínovský during the winter terms of years 2012 and 2013. The course was thought for Bachelor's students.

Modern electronic circuit design

I have been member of the team of lecturers in the subject Modern electronic circuit design (DRE1, Návrh moderních elektronických obvodů) since 2014. The course is dedicated for Ph.D. students of the Doctoral program Electrical Engineering and Communication. This course is guaranteed by prof. Zdeněk Kolka and I am responsible for three lectures covering the evolutionary optimization.

CAD in Communication Subsystems

In 2013, I started to guarantee the subject CAD in Communication Subsystems (BRKS, Počítačové řešení komunikačních subsystémů). The course was created by prof. Zbyněk Raida and then guaranteed and modified by Petr Vágner. After I started to guarantee the subject, the optimization theory and methods (including the multi-objective ones) were included to the syllabus of the subject. The course was thought for Bachelor's students. The subject was open for Erasmus students in years 2017 and 2018 and therefore, thought in English (CRKS).

Computer Programming 1

In 2018, I started to guarantee the subject Computer Programming 1 (BPC-PP1, Počítačové programování). The course was created as a brand new subject to introduce programming to the Bachelor's students for the re-accredited program BPC-EKT (Electronics and Communication Technologies). Previously, the course was divided to two halves: one dedicated to C language and the other dedicated to MATLAB language. After the re-accreditation, the course is fully dedicated to MATLAB, that enables to introduce the students to algorithmization in a less complicated way compared to the C language. I created brand new materials for the whole subject: new lectures and computer exercises. The subject is open also for long-distance students in program BKC-EKT.

7.2 Student Theses

I have lead six MSc. students and five Bc. students that successfully defended their final theses since year 2009. These students and their theses are summarized in Table 7.1. Student Jan Maloušek received the BUT Rector's price and Josef Hlávka's Price and won the Golden transistor student's competition with his Bc. thesis entitled: Mobile Android sport application.

Ph.D. candidate Martin Marek is currently in the fourth year of his studies. He made the state exam in 2018 with his treatise on Ph.D. thesis entitled: "Multi-objective Optimization of EM Structures With Variable Number of Dimensions". Also, he participated in the Student EEICT conference in 2016 (paper "Comparison of multi-objective optimization methods") and 2018 "Discretization of decision variables in real-coded optimization algorithms").

Table 7.1: Successfully defended Bc. and MSc. student's theses.

Student	Thesis	Level	Year of defense
Marcel Kollár	PIFA Antenna design for GSM band	MSc.	2011
Tomáš Ščiřlak	Mesh Generator for the Finite Element Method	MSc.	2011
Ivan Hoogsteyns	Mechanical and electromagnetic optimisation of carbon fibre reinforced composite materials used in airplanes	MSc.	2013
David Krutílek	Numerical EM analysis of semi-composite model of EV55 aircraft	MSc.	2013
Martin Marek	Smith Chart Visualization Tool	Bc.	2014
Petr Víteček	Radio Network Multiobjective Design	MSc.	2014
Martin Marek	Toolbox for multi-objective optimization	MSc.	2016
Zdeněk Podjůkl	Comparative tool for datasets using method IELF	Bc.	2018
Jakub Michálek	Mobile Android Application for Electromagnetics	Bc.	2018
Jan Maloušek	Mobile Android sport application	Bc.	2018
Miroslav Maluš	Antenna catalogue for Matlab toolbox	Bc.	2019

Chapter 8

Research Projects

At this point, I would like to thank to all the institutions that provided me and my colleagues with the necessary funding!

Modeling and simulation of fields

GA102/07/0688 (Modelování a simulace polí)

Years: 2009 - 2011

Principal researcher: prof. Zbyněk Raida (Brno University of Technology)

This grant project was aimed to improve the current PhD studies in the domain of a fractionalism of the education, excessive specialization in particular areas, insufficient sharing of well-equipped laboratories, insufficient communication and co-operation among students while they use the same techniques of modeling, simulation and optimization. I was a member of the research team. I was co-author of one chapter dedicated to numerical optimization in the monograph [154] which was the most important outcome of the project.

High Intensity Radiated Field Synthetic Environment

7E09008

Years: 2009 - 2012

Principal researcher: prof. Zbyněk Raida (Brno University of Technology)

This project interconnected more than 30 universities and companies around Europe that are specialized in the field of numerical modeling of electromagnetic phenomena. The project aimed to expedite the process of EMC certification of aircrafts. I was member of the team at the Department of Radio Electronics that was responsible for the development of the time-domain and frequency-domain finite element method solver. This solver was called BUTFE and was fully written in MATLAB (for more details please see [2]).

Tools for synthesis of antennas and sensors

TA04010457 (Nástroje pro syntézu antén a senzorů)

Years: 2014 - 2017

Principal researcher: Petr Kadlec (for Brno University of Technology)

This project was solved in cooperation with Czech Technical University Prague (CTU, principal investigator, Miloslav Čapek) and company MECAS ESI s.r.o. The main goal of the project was the transfer of software tools that was created at both the universities with an age-long fundamental research in the domain of antenna analysis and design. I was a member of a highly collaborating team established among researchers at the two universities and the company that created two softwares fully written in MATLAB: AToM (Antenna Toolbox for MATLAB) and FOPS (Fast Optimization ProcedureS). AToM [1] is a versatile tool that enables the characteristic modes analysis of the antennas. FOPS [3] is a standalone tool that enables to solve optimization problems (single-objective, multi-objective and VND ones).

Interdisciplinrn vzkum bezdrtovch technologi (INWITE)

LO1401 (Interdisciplinární výzkum bezdrátových technologi (INWITE))

Years: 2015 - 2019

Principal researcher: prof. Zbyněk Raida (Brno University of Technology)

The project aimed to a research of innovative concepts of wireless communication systems, which can provide high reliability, high speed and capacity, jointly ensuring high digital security with a wide applicability.

Applications of Space-Time Reciprocity in Computational Electromagnetic Compatibility

GJ17-05445Y (Aplikace časoprostorové reciprocity ve výpočetní elektromagnetické kompatibilitě)

Years: 2017 - 2018

Principal researcher: Martin Štumpf (Brno University of Technology)

The project aimed to develop new time-domain analysis tools for transient phenomena of electromagnetic interferences. Therefore, it should help to ensure the problem-free coexistence of new digital systems. I was member of the research team that developed a MATLAB software tool TD-PCBS (Time Domain Printed Circuit Board Solver) that combines a Time-Domain Contour Integral Method with the circuit simulator ngspace. The detailed info about this software can be found in [4].

Virtual Prototyping and Validation of Electromagnetic Systems

TH04010373 (Virtuální prototypování a validace elektromagnetických systémů)

Years: 2018 - 2021

Principal researcher: Petr Kadlec (for Brno University of Technology)

This is an ongoing project that continues in the successful cooperation with the company MECAS Esi Group a.s. (principal researcher Jaroslav Rýmus) and Czech Technical University in Prague established during the solution of project TA04010457. The project aims to transfer knowledge in the domain of virtual prototyping of electromagnetic sensors to the simulator Visual CEM that is developed at MECAS ESI s.r.o. During the solution of the project I was the co-author of the VFTool (Vector Fitting MATLAB Toolbox), that enables the approximation of complex functions in the frequency domain (please refer to [5]).

Chapter 9

Conclusions

The presented thesis covers the theory of evolutionary optimization. First, the optimization problem is formulated as the single-objective one (a single vector of decision space variables is expected as a result of the optimization process), multi-objective one (set of trade-off solutions is found). Moreover, some optimization tasks require to choose the complexity of the solved problem (which means the number of decision space variables) to be estimated a priori when solved by conventional single-objective and multi-objective algorithms. Therefore, the optimization task with the variable number of dimensions is formulated in this thesis.

Then, three state-of-the-art algorithms namely Genetic Algorithms, Particle Swarm Optimization, and Differential Evolution are briefly reviewed in the thesis. The multi-objective extensions of these algorithms are also discussed. Then, the works that introduces the VND variants of these algorithm are summarized.

Only limited number of VND algorithms was presented in otherwise enormous number of optimization algorithms presented. They are often called as Variable Length algorithms based on their introduction in the domain of genetic algorithms, but I believe that the name Variable Number of Dimensions fits better as it is more general. In case of the VND optimization task the number of decision space variables becomes to be unknown. The optimization algorithm has to work with agents (candidate solutions) that have different sizes. On the first sight, this could make the task more complex. Definitely, the algorithm has to be slightly changed. But VND formulation gets rid of the a priori choice of the design complexity, which is a very complicated without a good knowledge about the problem behavior.

In the main part of the thesis reprints of the selected published research papers of me (as the main author) and my colleagues are presented. In the single-objective section, the inverse problem of localization of the lightning stroke based on the voltage induced to the transmission line is shown. The multi-objective part contains the design of the dielectric filter and Yagi-Uda antenna by the MOSOMA algorithm, that was developed during my Ph.D. studies. Then, multi-objective optimization of the testing pulse so the time-domain EMC shielding effectiveness tests of the thin metal sheets are adjusted is presented.

The most important contribution of this thesis in the fundamental research can be found in the domain of the VND evolutionary optimization. First, the paper introducing VND variant of the PSO algorithm is reprinted here. It contains the general approach how to handle variable dimensions of solution candidates, that can be applied to any other algorithm. Also, several benchmark problems and testing metrics to evaluate the

convergence of the VND algorithms are introduced there. Then, this algorithm was applied to solution of various electromagnetic problems: the design of antenna array, the optimization of decoupling capacitors on printed circuit board and the reconstruction of the dielectric profile. The papers have shown that the VND formulation can be more effective in terms of the convergence rate and minimization of the design variables and therefore resources.

The last two chapters cover the teaching experiences and research projects, respectively. During my teaching career, I have been involved in the teaching of seven subjects at the Department of Radio Electronics (from which two have been guaranteed by me) and I have supervised together 11 Bc. and Msc. theses. I have been involved in more than six research projects in which I have co-developed several software tools.

Bibliography

- [1] AToM: Antenna Toolbox for MATLAB. <http://antennatoolbox.com/atom>. Accessed: 2020-04-01.
- [2] BUTFE: Time Domain Finite Element Solver. <https://www.vutbr.cz/vav/vysledky/detail/71460#vysledek-71460>. Accessed: 2020-04-01.
- [3] FOPS: Fast Optimization Procedures. <http://antennatoolbox.com/fops>. Accessed: 2020-04-01.
- [4] TD-PCBS: Time Domain Printed Circuit Board Solver. https://www.urel.feec.vutbr.cz/web_documents/produkty/2018/TD-PCBS_SW_EN.pdf. Accessed: 2020-04-01.
- [5] VFTool: Vector Fitting MATLAB Toolbox. https://www.urel.feec.vutbr.cz/web_documents/produkty/2019/Kadlec_VectorFitting_sw_EN.pdf. Accessed: 2020-04-01.
- [6] S. Abegunawardana, J. A. P. Bodhika, R. Abeywardhana, M. Fernando, and V. Cooray. Sound source localization of lightning discharges. In *2018 34th International Conference on Lightning Protection (ICLP)*, pages 1–6, Sep. 2018.
- [7] K. Agarwal, X. Chen, L. Pan, and S. P. Yeo. Multiple signal classification algorithm for non-destructive imaging of reinforcement bars and empty ducts in circular concrete columns. In *2011 XXXth URSI General Assembly and Scientific Symposium*, pages 1–4. IEEE, 2011.
- [8] M. Akbari, K. Sheshyekani, A. Pirayesh, F. Rachidi, M. Paolone, A. Borghetti, and C. A. Nucci. Evaluation of lightning electromagnetic fields and their induced voltages on overhead lines considering the frequency dependence of soil electrical parameters. *IEEE Transactions on Electromagnetic Compatibility*, 55(6):1210–1219, Dec 2013.
- [9] M. M. Ali, C. Khompatraporn, and Z. B. Zabinsky. A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. *Journal of global optimization*, 31(4):635–672, 2005.
- [10] M. G. Alonso and P. Duysinx. Particle swarm optimization (PSO): an alternative method for composite optimization. In *10 th World Congress on Structural and Multidisciplinary Optimization*, pages 1–10, 2013.

- [11] N. Ambasana and D. Gope. Mesh-sensitivity based decoupling capacitor sizing and placement for power delivery networks. In *Signal and Power Integrity (SPI), 2017 IEEE 21st Workshop on*, pages 1–4. IEEE, 2017.
- [12] R. Araneo, G. Attolini, S. Celozzi, and G. Lovat. Time-domain shielding performance of enclosures: A comparison of different global approaches. *IEEE Transactions on Electromagnetic Compatibility*, 58(2):434–441, April 2016.
- [13] R. Araneo and S. Celozzi. Toward a definition of the shielding effectiveness in the time - domain. In *2013 IEEE International Symposium on Electromagnetic Compatibility*, pages 113–117, Aug 2013.
- [14] R. O. Arechiga, J. B. Johnson, H. E. Edens, R. J. Thomas, and W. Rison. Acoustic localization of triggered lightning. *Journal of Geophysical Research: Atmospheres*, 116(D9), 2011.
- [15] A. Auger and N. Hansen. Performance evaluation of an advanced local search evolutionary algorithm. In *2005 IEEE congress on evolutionary computation*, volume 2, pages 1777–1784. IEEE, 2005.
- [16] J. C. Bansal, P. Singh, M. Saraswat, A. Verma, S. S. Jadon, and A. Abraham. Inertia weight strategies in particle swarm optimization. In *2011 Third world congress on nature and biologically inspired computing*, pages 633–640. IEEE, 2011.
- [17] C. Bao, L. Xu, E. D. Goodman, and L. Cao. A novel non-dominated sorting algorithm for evolutionary multi-objective optimization. *Journal of Computational Science*, 23:31–43, 2017.
- [18] S. Baskar, A. Alphones, P. Suganthan, and J. Liang. Design of yagi–uda antennas using comprehensive learning particle swarm optimisation. *IEE Proceedings-Microwaves, Antennas and Propagation*, 152(5):340–346, 2005.
- [19] M. Bertero and P. Boccacci. *Introduction to inverse problems in imaging*. CRC press, 1998.
- [20] M. Bertero and M. Piana. Inverse problems in biomedical imaging: modeling and methods of solution. In *Complex systems in biomedicine*, pages 1–33. Springer, 2006.
- [21] K. Bharath, E. Engin, and M. Swaminathan. Automatic package and board decoupling capacitor placement using genetic algorithms and M-FDM. In *Proc. 45th Design Automation Conf.*, pages 560–565. ACM, 2008.
- [22] A. Bhargav and N. Gupta. Multiobjective genetic optimization of nonuniform linear array with low sidelobes and beamwidth. *IEEE Antennas and Wireless Propagation Letters*, 12:1547–1549, 2013.
- [23] H. Blok and M. Zeylmans. Reciprocity and the formulation of inverse profiling problems. *Radio science*, 22(07):1137–1147, 1987.

- [24] P. Burghignoli, G. Lovat, R. Araneo, and S. Celozzi. Time-domain shielding of a thin conductive sheet in the presence of pulsed vertical dipoles. *IEEE Transactions on Electromagnetic Compatibility*, 60(1):157–165, Feb 2018.
- [25] G. J. Burke and A. J. Poggio. Numerical electromagnetics code (NEC)-method of moments. 1981.
- [26] C. Caligaris, F. Delfino, and R. Procopio. Cooray-Rubinstein formula for the evaluation of lightning radial electric fields: Derivation and implementation in the time domain. *IEEE Transactions on Electromagnetic Compatibility*, 50(1):194–197, Feb 2008.
- [27] M. Capek, L. Jelinek, P. Kadlec, and M. Strambach. Excitation of optimal and suboptimal currents, 2017.
- [28] S. Celozzi and R. Araneo. Alternative definitions for the time-domain shielding effectiveness of enclosures. *IEEE Transactions on Electromagnetic Compatibility*, 56(2):482–485, April 2014.
- [29] J. Chen and J. Wang. A three-dimensional semi-implicit FDTD scheme for calculation of shielding effectiveness of enclosure with thin slots. *IEEE Transactions on Electromagnetic Compatibility*, 49(2):354–360, May 2007.
- [30] S. M. Chen, Y. Du, L. M. Fan, H. M. He, and D. Z. Zhong. A lightning location system in China: its performances and applications. *IEEE Transactions on Electromagnetic Compatibility*, 44(4):555–560, Nov 2002.
- [31] D. K. Cheng. Gain optimization for yagi-uda arrays. *IEEE Antennas and Propagation Magazine*, 33(3):42–46, 1991.
- [32] W. C. Chew. *Waves and fields in inhomogeneous media*. IEEE press, 1995.
- [33] W. Chien and C.-C. Chiu. Using NU-SSGA to reduce the searching time in inverse problem of a buried metallic object. *IEEE Transactions on Antennas and Propagation*, 53(10):3128–3134, 2005.
- [34] C.-C. Chiu and P.-T. Liu. Image reconstruction of a perfectly conducting cylinder by the genetic algorithm. *IEE Proceedings-Microwaves, Antennas and Propagation*, 143(3):249–253, 1996.
- [35] C.-C. Chiu, C.-H. Sun, C.-L. Li, and C.-H. Huang. Comparative study of some population-based optimization algorithms on inverse scattering of a two-dimensional perfectly conducting cylinder in dielectric slab medium. *IEEE Transactions on Geoscience and Remote Sensing*, 51(4):2302–2315, 2012.
- [36] P. Chowdhuri, J. G. Anderson, W. A. Chisholm, T. E. Field, M. Ishii, J. A. Martinez, M. B. Marz, J. McDaniel, T. E. McDermott, A. M. Mousa, T. Narita, D. K. Nichols, and T. A. Short. Parameters of lightning strokes: a review. *IEEE Transactions on Power Delivery*, 20(1):346–358, Jan 2005.
- [37] H. Chun-yun, H. Bin-jie, and X. Yun-hui. Mobile agent routing using variable-dimension pso algorithm based on chord-length parameterization. 2013.

- [38] V. Cooray and G. Cooray. The electromagnetic fields of an accelerating charge: Applications in lightning return-stroke models. *IEEE Transactions on Electromagnetic Compatibility*, 52(4):944–955, Nov 2010.
- [39] D. Corne, M. Dorigo, F. Glover, D. Dasgupta, P. Moscato, R. Poli, and K. V. Price. *New ideas in optimization*. McGraw-Hill Ltd., UK, 1999.
- [40] C. Craeye and D. González-Ovejero. A review on array mutual coupling analysis. *Radio Science*, 46(02):1–25, 2011.
- [41] C. Darwin. *On the Origin of Species by Means of Natural Selection Or the Preservation of Favoured Races in the Struggle for Life*. H. Milford; Oxford University Press, 1859.
- [42] A. De Hoop. A modification of Cagniards method for solving seismic pulse problems. *Applied Scientific Research, Section B*, 8(1):349–356, 1960.
- [43] A. T. de Hoop and I. E. Lager. Pulsed fields EM interference analysis in digital signal wireless interconnects. In *2011 41st European Microwave Conference*, pages 313–316, Oct 2011.
- [44] M. V. de Hoop and A. T. de Hoop. Wave-field reciprocity and optimization in remote sensing. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 456(1995):641–682, 2000.
- [45] K. Deb. *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons, 2001.
- [46] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.
- [47] J. Decraene, M. Chandramohan, F. Zeng, M. Y. H. Low, and W. Cai. Evolving agent-based model structures using variable-length genomes. In *Proceedings of the 4th International Workshop on Optimisation in Multi-Agent Systems at AAMAS*, volume 11, 2011.
- [48] D. S. Deif and Y. Gadallah. Wireless sensor network deployment using a variable-length genetic algorithm. In *2014 IEEE wireless communications and networking conference (WCNC)*, pages 2450–2455. IEEE, 2014.
- [49] E. Diaz-Alvarez, J. P. Krusius, and F. Kroeger. Modeling and simulation of integrated capacitors for high frequency chip power decoupling. *IEEE Transactions on Components and Packaging Technologies*, 23(4):611–619, 2000.
- [50] W.-Y. Ding, X.-C. Wei, D. Yi, and Y.-F. Shu. A closed-form solution for the impedance calculation of grid power distribution network. *IEEE Transactions on Electromagnetic Compatibility*, 59(5):1449–1456, 2017.
- [51] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pages 39–43. Ieee, 1995.

- [52] F. Enache, F. Popescu, D. Depreanu, A. Enache, and T. Oroian. Multi-criteria optimization of non-uniform linear antenna array using genetic algorithms. In *2016 International Conference on Communications (COMM)*, pages 121–124, June 2016.
- [53] I. Erdin and R. Achar. Efficient decoupling capacitor placement based on driving point impedance. *IEEE Transactions on Microwave Theory and Techniques*, 66(2):669–677, 2018.
- [54] I. Erdin and R. Achar. Multipin optimization method for placement of decoupling capacitors using a genetic algorithm. *IEEE Transactions on Electromagnetic Compatibility*, 2018.
- [55] D. Fagan and R. Meier. Intelligent time of arrival estimation. In *2011 IEEE Forum on Integrated and Sustainable Transportation Systems*, pages 60–66, June 2011.
- [56] P. Flajolet and R. Sedgewick. *Analytic combinatorics*. cambridge University press, 2009.
- [57] B. Fuchs. Antenna selection for array synthesis problems. *IEEE Antennas and Wireless Propagation Letters*, 16:868–871, 2017.
- [58] A. Ghaffarizadeh, K. Ahmadi, and N. S. Flann. Sorting unsigned permutations by reversals using multi-objective evolutionary algorithms with variable size individuals. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 292–295. IEEE, 2011.
- [59] M. Gheith, A. B. Eltawil, and N. A. Harraz. Solving the container pre-marshalling problem using variable length genetic algorithms. *Engineering Optimization*, 48(4):687–705, 2016.
- [60] D. E. Goldberg and J. H. Holland. Genetic algorithms and machine learning. 1988.
- [61] S. J. Goodman, R. J. Blakeslee, W. J. Koshak, D. Mach, J. Bailey, D. Buechler, L. Carey, C. Schultz, M. Bateman, E. McCaul Jr, et al. The GOES-R geostationary lightning mapper (GLM). *Atmospheric research*, 125:34–49, 2013.
- [62] S. Goudos, Z. Zaharis, M. Salazar-Lechuga, P. Lazaridis, and P. Gallion. Dielectric filter optimal design suitable for microwave communications by using multiobjective evolutionary algorithms. *Microwave and optical technology letters*, 49(10):2324–2329, 2007.
- [63] S. K. Goudos, K. A. Gotsis, K. Siakavara, E. E. Vafiadis, and J. N. Sahalos. A multi-objective approach to subarrayed linear antenna arrays design based on memetic differential evolution. *IEEE Transactions on Antennas and Propagation*, 61(6):3042–3052, June 2013.
- [64] S. K. Goudos, K. Siakavara, T. Samaras, E. E. Vafiadis, and J. N. Sahalos. Self-adaptive differential evolution applied to real-valued antenna and microwave design problems. *IEEE Transactions on Antennas and Propagation*, 59(4):1286–1298, April 2011.

- [65] L. A. Greda, A. Winterstein, D. L. Lemes, and M. V. T. Heckler. Beamsteering and beamshaping using a linear antenna array based on particle swarm optimization. *IEEE Access*, 7:141562–141573, 2019.
- [66] S. Guerrieri, M. Ianoz, C. Nucci, C. Mazzetti, and F. Rachidi. Lightning-induced voltages on an overhead line above a lossy ground: a sensitivity analysis. In *23rd International Conference on Lightning Protection*, number CONF, pages 23–27, 1996.
- [67] B. V. Ha, M. Mussetta, P. Pirinoli, and R. E. Zich. Modified compact genetic algorithm for thinned array synthesis. *IEEE Antennas and Wireless Propagation Letters*, 15:1105–1108, 2016.
- [68] N. Hansen, A. Auger, R. Ros, S. Finck, and P. Pošík. Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. In *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*, pages 1689–1696. ACM, 2010.
- [69] N. Hansen, S. D. Mller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18, March 2003.
- [70] R. L. Haupt and D. H. Werner. *Genetic algorithms in electromagnetics*. John Wiley & Sons, 2007.
- [71] M. Hu, T. Wu, and J. D. Weir. An adaptive particle swarm optimization with multiple adaptive methods. *IEEE Transactions on Evolutionary Computation*, 17(5):705–720, 2012.
- [72] A. H. Hussein, H. H. Abdullah, A. M. Salem, S. Khamis, and M. Nasr. Optimum design of linear antenna arrays using a hybrid MoM/GA algorithm. *IEEE Antennas and Wireless Propagation Letters*, 10:1232–1235, 2011.
- [73] W. I. Ibrahim and M. R. Ghazali. Measurements of electric and magnetic fields due to lightning strokes based on single-station detection. In *2012 IEEE Asia-Pacific Conference on Applied Electromagnetics (APACE)*, pages 268–273, Dec 2012.
- [74] J. J. Jamian, M. N. Abdullah, H. Mokhlis, M. W. Mustafa, and A. H. A. Bakar. Global particle swarm optimization for high dimension numerical functions analysis. *Journal of Applied Mathematics*, 2014, 2014.
- [75] C. Jang, F. Hu, F. He, J. Li, and D. Zhu. Low-redundancy large linear arrays synthesis for aperture synthesis radiometers using particle swarm optimization. *IEEE Transactions on Antennas and Propagation*, 64(6):2179–2188, June 2016.
- [76] X. Jia and G. Lu. A hybrid Taguchi binary particle swarm optimization for antenna designs. *IEEE Antennas and Wireless Propagation Letters*, 18(8):1581–1585, Aug 2019.

- [77] N. Jin, X. Zhou, C. Lin, L. Wang, Y. Liu, M. L. Wymore, and D. Qiao. ThunderLoc: Smartphone-based crowdsensing for thunder localization. In *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–2, June 2018.
- [78] J. M. Johnson and V. Rahmat-Samii. Genetic algorithms in engineering electromagnetics. *IEEE Antennas and propagation Magazine*, 39(4):7–21, 1997.
- [79] E. A. Jones and W. T. Joines. Design of yagi-uda antennas using genetic algorithms. *IEEE Transactions on Antennas and Propagation*, 45(9):1386–1392, 1997.
- [80] M. Kadlec, P. Čapek, J. Rýmus, M. Marek, M. Štumpf, L. Jelínek, M. Mašek, and P. Kotalík. Design of a linear antenna array: Variable number of dimensions approach. In *Proceedings of 30th International Conference Radioelektronika 2020*, pages 1–6. IEEE, 2020.
- [81] P. Kadlec. *Multi-Objective Optimization of Electromagnetic Structures Based on Self-Organizing Migration*. dissertation, VUTIU, october 2012.
- [82] P. Kadlec. *Multiobjective optimization of EM structures based on self-organizing migration*. PhD thesis, Ph. D. thesis, Brno University of Technology, Czech republic, 2012.
- [83] P. Kadlec, M. Marek, and M. Štumpf. Lightning stroke localization—a time-domain approach based on evolutionary optimization. *IEEE Transactions on Electromagnetic Compatibility*, 2020.
- [84] P. Kadlec, M. Marek, and M. Štumpf. Thin-film screen time-domain shielding effectiveness: Multi-objective optimization of the testing pulse. In *Electromagnetic Compatibility-EMC EUROPE, 2020 International Symposium on*, pages 1–5. IEEE, 2020.
- [85] P. Kadlec, M. Marek, M. Štumpf, and V. Šeděnka. PCB decoupling optimization with variable number of capacitors. *IEEE Transactions on Electromagnetic Compatibility*, pages 1–8, 2018.
- [86] P. Kadlec, M. Marek, and M. Štumpf. Fast lightning stroke localization in the time domain. In *2019 International Symposium on Electromagnetic Compatibility (EMC EUROPE)*, pages 585–589, Aug 2019.
- [87] P. Kadlec and Z. Raida. Comparison of novel multi-objective self organizing migrating algorithm with conventional methods. In *Proceedings of 21st International Conference Radioelektronika 2011*, pages 1–4. IEEE, 2011.
- [88] P. Kadlec and Z. Raida. A novel multi-objective self-organizing migrating algorithm. *Radioengineering*, 20(4), 2011.
- [89] P. Kadlec and Z. Raida. Self-organizing migrating algorithm for optimization with general number of objectives. In *Proceedings of 22nd International Conference Radioelektronika 2012*, pages 1–5. IEEE, 2012.

- [90] P. Kadlec and Z. Raida. Multi-objective self-organizing migrating algorithm applied to the design of electromagnetic components. *IEEE Antennas and Propagation Magazine*, 55(6):50–68, 2013.
- [91] P. Kadlec and Z. Raida. Multi-objective design of em components. In *Self-Organizing Migrating Algorithm*, pages 105–119. Springer, 2016.
- [92] P. Kadlec and Z. Raida. Multi-objective self-organizing migrating algorithm. In *Self-Organizing Migrating Algorithm*, pages 83–103. Springer, 2016.
- [93] P. Kadlec and V. Šeděnka. Particle swarm optimization for problems with variable number of dimensions. *Engineering Optimization*, 50(3):382–399, 2018.
- [94] P. Kadlec, V. Šeděnka, M. Marek, and M. Štumpf. Optimizing a decoupling capacitor on a PCB: A fully time-domain approach based on PSO and TD-CIM. In *Electromagnetic Compatibility-EMC EUROPE, 2018 International Symposium on*, pages 1–4. IEEE, 2018.
- [95] P. Kadlec, V. Šeděnka, M. Štumpf, and M. Marek. Solution of an inverse scattering problem using optimization with a variable number of dimensions. In *Electromagnetics in Advanced Applications (ICEAA), 2017 International Conference on*, pages 976–979. IEEE, 2017.
- [96] S. Kahng. GA-optimization for finding decoupling capacitors to damp the rectangular power-bus resonances. In *Electromagnetic Compatibility, 2007. EMC Zurich 2007. 18th International Zurich Symposium on*, pages 103–106. IEEE, 2007.
- [97] G. Kameswari, V. Gopi, V. B. Kumar, and P. V. Y. Jayasree. An efficient shielding effectiveness analysis of rectangular box perforated with numerous apertures by transmission line model. In *2013 IEEE Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia)*, pages 96–102, Dec 2013.
- [98] A. Kamo, T. Watanabe, and A. Asai. Simulation for the optimal placement of decoupling capacitors on printed circuit board. In *The 2001 IEEE International Symp. Circuits and Systems*, volume 3, pages 727–730, May 2001.
- [99] M. Kazim, A. H. Khawaja, U. Zabit, and Q. Huang. Fault detection and localization for overhead 11 kV distribution lines with magnetic measurements. *IEEE Transactions on Instrumentation and Measurement*, pages 1–1, 2019.
- [100] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks*, volume 4, pages 1942–1948. IEEE, 1995.
- [101] I. Y. Kim and O. De Weck. Variable chromosome length genetic algorithm for progressive refinement in topology optimization. *Structural and Multidisciplinary Optimization*, 29(6):445–456, 2005.
- [102] Y.-J. Kim, H.-S. Yoon, S. Lee, G. Moon, J. Kim, and J.-K. Wee. An efficient path-based equivalent circuit model for design, synthesis, and optimization of power distribution networks in multilayer printed circuit boards. *IEEE transactions on advanced packaging*, 27(1):97–106, 2004.

- [103] S. Kiranyaz, T. Ince, A. Yildirim, and M. Gabbouj. Fractional particle swarm optimization in multidimensional search space. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 40(2):298–319, 2009.
- [104] J. Knighten. PDN design strategies: I. ceramic SMT decoupling capacitors-what values should I choose? *IEEE EMC Society Newsletter*, 207:46–65, 2005.
- [105] J. Knighten. PDN design strategies: II. ceramic SMT decoupling capacitors-does location matter? *IEEE EMC Society Newsletter*, 207:34–41, 2005.
- [106] J. Knighten, B. Archambeault, J. Fan, G. Selli, L. Xue, S. Connor, and J. Drewniak. PDN design strategies: III. planes and materials—are they important factors in power bus design? *IEEE EMC Society Newsletter*, (208):56–67, 2006.
- [107] J. D. Knowles, L. Thiele, and E. Zitzler. A tutorial on the performance assessment of stochastic multiobjective optimizers. *TIK-Report*, 214, 2006.
- [108] K. Koo, G. R. Luevano, T. Wang, S. Özbayat, T. Michalka, and J. L. Drewniak. Fast algorithm for minimizing the number of decap in power distribution networks. *IEEE Transactions on Electromagnetic Compatibility*, 60(3):725–732, 2018.
- [109] S. Kukkonen and J. Lampinen. Gde3: The third evolution step of generalized differential evolution. In *2005 IEEE congress on evolutionary computation*, volume 1, pages 443–450. IEEE, 2005.
- [110] Y. Kuwahara. Multiobjective optimization design of yagi-uda antenna. *IEEE Transactions on Antennas and Propagation*, 53(6):1984–1992, 2005.
- [111] G. Lakota, G. Milev, V. Djurica, and B. Žitnik. Use of high resolution flash density data and GIS analysis methods in power utilities networks. In *2014 International Conference on Lightning Protection (ICLP)*, pages 1229–1233. IEEE, 2014.
- [112] R. Lakshmi and K. Vivekanandan. An analysis of recombination operator in genetic algorithms. In *2013 Fifth International Conference on Advanced Computing (ICoAC)*, pages 223–226. IEEE, 2013.
- [113] C. Lee and E. Antonsson. Variable length genomes for evolutionary algorithms. In *GECCO*, volume 2000, page 806, 2000.
- [114] H. Li, K. Deb, and Q. Zhang. Variable-length Pareto optimization via decomposition-based evolutionary multiobjective algorithm. *IEEE Transactions on Evolutionary Computation*, 23(6):987–999, 2019.
- [115] X. Li, J. Qian, and Z. Zhao. Discrete hidden markov model training based on variable length particle swarm optimization algorithm. *International Journal of Digital Content Technology and its Applications*, 6(20):182, 2012.
- [116] Y. Li, W. Hong, M. Li, and G. Lu. Unequally spaced linear antenna arrays synthesis based on genetic algorithm. In *2018 International Workshop on Antenna Technology (iWAT)*, pages 1–4, March 2018.

- [117] J. Liang, B. Qu, P. Suganthan, and A. G. Hernández-Díaz. Problem definitions and evaluation criteria for the cec 2013 special session on real-parameter optimization. *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report*, 201212(34):281–295, 2013.
- [118] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE transactions on evolutionary computation*, 10(3):281–295, 2006.
- [119] J.-J. Liang and P. N. Suganthan. Dynamic multi-swarm particle swarm optimizer. In *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.*, pages 124–129. IEEE, 2005.
- [120] B. Liu, L. Shi, S. Qiu, H. Liu, Z. Sun, and Y. Guo. Three-dimensional lightning positioning in low-frequency band using time reversal in frequency domain. *IEEE Transactions on Electromagnetic Compatibility*, pages 1–11, 2019.
- [121] L. Liu, S. Yang, G. Ni, and J. Huang. Fast frequency-domain modeling of return stroke including influence of lossy ground. *IEEE Transactions on Magnetics*, 50(2):149–152, Feb 2014.
- [122] G. Lovat, P. Burghignoli, R. Araneo, and S. Celozzi. Time-domain green’s function for a vertical dipole above a graphene sheet. *IEEE Transactions on Nanotechnology*, 17(4):841–851, July 2018.
- [123] D. M. Mach, H. J. Christian, R. J. Blakeslee, D. J. Boccippio, S. J. Goodman, and W. L. Boeck. Performance assessment of the optical transient detector and lightning imaging sensor. *Journal of Geophysical Research: Atmospheres*, 112(D9), 2007.
- [124] M. Maddah-Ali, S. H. H. Sadeghi, and M. Dehmollaian. Efficient method for calculating the shielding effectiveness of axisymmetric multilayered composite enclosures. *IEEE Transactions on Electromagnetic Compatibility*, pages 1–11, 2019.
- [125] M. Marek and P. Kadlec. *FOPS Documentation*, 2018. FOPS version v1.2.
- [126] M. Marek, P. Kadlec, and M. Čapek. Fops: A new framework for the optimization with variable number of dimensions. *International Journal of RF and Microwave Computer-Aided Engineering*, 2020. submitted for publication.
- [127] M. Marek, P. Kadlec, and M. Cupal. FOPS documentation. <http://http://antennatoolbox.com/fops>.
- [128] U. Maulik and I. Saha. Automatic fuzzy clustering using modified differential evolution for image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 48(9):3503–3510, 2010.
- [129] K. McClymont and E. Keedwell. Deductive sort and climbing sort: New methods for non-dominated sorting. *Evolutionary computation*, 20(1):1–26, 2012.
- [130] A. R. Mehrabian and C. Lucas. A novel numerical optimization algorithm inspired from weed colonization. *Ecological informatics*, 1(4):355–366, 2006.

- [131] H. Meunier, E.-G. Talbi, and P. Reininger. A multiobjective genetic algorithm for radio network optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512)*, volume 1, pages 317–324. IEEE, 2000.
- [132] K. A. Michalski. Electromagnetic imaging of elliptical–cylindrical conductors and tunnels using a differential evolution algorithm. *Microwave and Optical Technology Letters*, 28(3):164–169, 2001.
- [133] K. Miettinen and P. Preface By-Neittaanmaki. *Evolutionary algorithms in engineering and computer science: recent advances in genetic algorithms, evolution strategies, evolutionary programming, GE*. John Wiley & Sons, Inc., 1999.
- [134] C. Möller and L. Klinkenbusch. Electromagnetic and transient shielding effectiveness for near-field sources. *Advances in Radio Science*, 5(E. 2):57–62, 2007.
- [135] A. Mukhopadhyay and M. Mandal. Identifying non-redundant gene markers from microarray data: a multiobjective variable length PSO-based approach. *IEEE/ACM transactions on computational biology and bioinformatics*, 11(6):1170–1183, 2014.
- [136] A. Nag, M. J. Murphy, W. Schulz, and K. L. Cummins. Lightning locating systems: Insights on characteristics and validation techniques. *Earth and Space Science*, 2(4):65–93, 2015.
- [137] P. Nannipieri, D. Davalle, S. Nencioni, P. Lombardi, and L. Fanucci. Clustering algorithm for a spaceborne lightning imager: Design, trade-off, and FPGA implementation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pages 1–11, 2019.
- [138] A. J. Nebro, E. Alba, G. Molina, F. Chicano, F. Luna, and J. J. Durillo. Optimal antenna placement using a new multi-objective chc algorithm. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 876–883, 2007.
- [139] B. H. Nguyen, B. Xue, and P. Andreae. A novel binary particle swarm optimization algorithm and its applications on knapsack and feature selection problems. In *Intelligent and Evolutionary Systems*, pages 319–332. Springer, 2017.
- [140] Y. Ning, X.-r. Bai, F. Zhou, and L. Liu. Method for inverse synthetic aperture radar imaging of space debris using improved genetic algorithm. *IET Radar, Sonar & Navigation*, 11(5):812–821, 2017.
- [141] C. A. Nucci, F. Rachidi, M. V. Ianoz, and C. Mazzetti. Lightning-induced voltages on overhead lines. *IEEE Transactions on Electromagnetic Compatibility*, 35(1):75–86, Feb 1993.
- [142] A. E. Olsson. *Particle swarm optimization: theory, techniques and applications*. Nova Science Publishers, Inc., 2010.
- [143] M. O’Neill, F. Leahy, and A. Brabazon. Grammatical swarm: a variable-length particle swarm algorithm. In *Swarm Intelligent Systems*, pages 59–74. Springer, 2006.

- [144] S. J. Orfanidis. *Electromagnetic waves and antennas*. 2002.
- [145] B. K. Panigrahi, Y. Shi, and M.-H. Lim. *Handbook of swarm intelligence: concepts, principles and applications*, volume 8. Springer Science & Business Media, 2011.
- [146] S. N. Pawar and R. S. Bichkar. Genetic algorithm with variable length chromosomes for network intrusion detection. *International Journal of Automation and Computing*, 12(3):337–342, 2015.
- [147] S. Piersanti, F. de Paulis, C. Olivieri, and A. Orlandi. Decoupling capacitors placement for a multichip pdn by a nature-inspired algorithm. *IEEE Transactions on Electromagnetic Compatibility*, 60(6):1678–1685, 2018.
- [148] S. Piersanti and A. Orlandi. Impact of genetic algorithm control parameters on chip PDN decoupling capacitors placement. *IEEE Transactions on Electromagnetic Compatibility*, 60(4):945–951, 2018.
- [149] K. Price, R. M. Storn, and J. A. Lampinen. *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media, 2006.
- [150] J. Pugh and A. Martinoli. Discrete multi-valued particle swarm optimization. In *Proceedings of IEEE swarm intelligence symposium*, number CONF, pages 103–110, 2006.
- [151] Z. Qiongbing and D. Lixin. A new crossover mechanism for genetic algorithms with variable-length chromosomes for path optimization problems. *Expert Systems with Applications*, 60:183–189, 2016.
- [152] M. Qiu, L. Liu, H. Ding, J. Dong, and W. Wang. A new hybrid variable-length ga and pso algorithm in continuous facility location problem with capacity and service level constraints. In *2009 IEEE/INFORMS International Conference on Service Operations, Logistics and Informatics*, pages 546–551. IEEE, 2009.
- [153] F. Rachidi, C. A. Nucci, M. Ianoz, and C. Mazzetti. Influence of a lossy ground on lightning-induced voltages on overhead lines. *IEEE Transactions on Electromagnetic Compatibility*, 38(3):250–264, Aug 1996.
- [154] Z. Raida. *Mikrovlnné struktury z netradičních materiálů*. MJ Servis, 2011.
- [155] Z. Raida, P. Kadlec, P. Kovács, J. Láčák, Z. Lukeš, M. Pokorný, P. Všetula, and D. Wolanský. Multi-objective synthesis of antennas from special and conventional materials. *Radioengineering*, 20(4), 2011.
- [156] V. A. Rakov and M. A. Uman. *Lightning: physics and effects*. Cambridge University Press, 2003.
- [157] A. G. Ramm. Some inverse scattering problems of geophysics. In *Scattering by Obstacles*, pages 217–284. Springer, 1986.
- [158] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions on Evolutionary Computation*, 8(3):240–255, June 2004.

- [159] M. Reyes-Sierra, C. C. Coello, et al. Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International journal of computational intelligence research*, 2(3):287–308, 2006.
- [160] P. Richeson. NEC-2 manual, part iii: Users guide, 1996.
- [161] K. N. Ripon, C.-H. Tsang, S. Kwong, and M.-K. Ip. Multi-objective evolutionary clustering using variable-length real jumping genes genetic algorithm. In *18th International Conference on Pattern Recognition (ICPR '06)*, volume 1, pages 1200–1203. IEEE, 2006.
- [162] V. Roberge, M. Tarbouchi, and G. Labonté. Comparison of parallel genetic algorithm and particle swarm optimization for real-time uav path planning. *IEEE Transactions on Industrial Informatics*, 9(1):132–141, 2012.
- [163] J. Robinson and Y. Rahmat-Samii. Particle swarm optimization in electromagnetics. *IEEE transactions on antennas and propagation*, 52(2):397–407, 2004.
- [164] Ruochen Liu, Xiao Wang, Yangyang Li, and Xiangrong Zhang. Multi-objective invasive weed optimization algorithm for clustering. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8, June 2012.
- [165] S. Salcedo-Sanz, P. García-Díaz, J. Del Ser, M. N. Bilbao, and J. A. Portilla-Figueras. A novel grouping coral reefs optimization algorithm for optimal mobile network deployment problems under electromagnetic pollution and capacity control criteria. *Expert Systems with Applications*, 55:388–402, 2016.
- [166] M. Salucci, L. Poli, N. Anselmi, and A. Massa. Multifrequency particle swarm optimization for enhanced multiresolution gpr microwave imaging. *IEEE Transactions on Geoscience and Remote Sensing*, 55(3):1305–1317, 2016.
- [167] Y. Shi and R. C. Eberhart. Empirical study of particle swarm optimization. In *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)*, volume 3, pages 1945–1950. IEEE, 1999.
- [168] L. Singh, S. Kumar, and S. Paul. Automatic simultaneous architecture and parameter search in fuzzy neural network learning using novel variable length crossover differential evolution. In *2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)*, pages 1795–1802. IEEE, 2008.
- [169] J. A. Snyman. *Practical mathematical optimization*. Springer, 2005.
- [170] J. Sprave and S. Rolf. Variable-dimensional optimization with evolutionary algorithms using fixed-length representations. In *International Conference on Evolutionary Programming*, pages 261–269. Springer, 1998.
- [171] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3):221–248, 1994.

- [172] R. Storn and K. Price. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.
- [173] M. Štumpf. The time-domain contour integral method—an approach to the analysis of double-plane circuits. *IEEE Transactions on Electromagnetic Compatibility*, 56(2):367–374, 2014.
- [174] M. Štumpf. *Electromagnetic Reciprocity in Antenna Theory*. Hoboken, NJ: IEEE Press–Wiley, 2018.
- [175] M. Štumpf. *Pulsed EM Field Computation in Planar Circuits: The Contour Integral Method*. CRC Press, 2018.
- [176] M. Štumpf. Pulsed vertical-electric-dipole excited voltages on transmission lines over a perfect ground - a closed-form analytical description. *IEEE Antennas and Wireless Propagation Letters*, 17(9):1656–1658, Sep. 2018.
- [177] M. Štumpf and G. Antonini. Lightning-induced voltages on transmission lines over a lossy ground - an analytical coupling model based on the Cooray–Rubinstein formula. *IEEE Transactions on Electromagnetic Compatibility*, 2019.
- [178] M. Štumpf and A. T. de Hoop. Loop-to-loop pulsed electromagnetic signal transfer across a thin metal screen with Drude-type dispersive behavior. *IEEE Transactions on Electromagnetic Compatibility*, 60(4):885–889, Aug 2018.
- [179] H. Su, S. S. Sapatnekar, and S. R. Nassif. Optimal decoupling capacitor sizing and placement for standard-cell layout designs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(4):428–436, 2003.
- [180] T. Takenaka, Z. Q. Meng, T. Tanaka, and W. C. Chew. Local shape function combined with genetic algorithm applied to inverse scattering for strips. *Microwave and Optical Technology Letters*, 16(6):337–341, 1997.
- [181] S. Tang, Z. Cai, and J. Zheng. A fast method of constructing the non-dominated set: arena’s principle. In *2008 Fourth International Conference on Natural Computation*, volume 1, pages 391–395. IEEE, 2008.
- [182] W. Tao, Q. Shi, and S. Lihua. Lightning localization using two MDF stations. In *2017 IEEE 5th International Symposium on Electromagnetic Compatibility (EMC-Beijing)*, pages 1–4, Oct 2017.
- [183] F. M. Tesche, M. Ianoz, and T. Karlsson. *EMC analysis methods and computational models*. John Wiley & Sons, 1996.
- [184] C.-K. Ting, C.-N. Lee, H.-C. Chang, and J.-S. Wu. Wireless heterogeneous transmitter placement using multiobjective variable-length genetic algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(4):945–958, 2009.

- [185] B. Tran, B. Xue, and M. Zhang. Variable-length particle swarm optimization for feature selection on high-dimensional classification. *IEEE Transactions on Evolutionary Computation*, 23(3):473–487, 2018.
- [186] X. L. Travassos, D. Vieira, N. Ida, C. Vollaire, and A. Nicolas. Inverse algorithms for the gpr assessment of concrete structures. *IEEE Transactions on Magnetics*, 44(6):994–997, 2008.
- [187] J. N. Tripathi, J. Mukherjee, P. R. Apte, N. K. Chhabra, R. K. Nagpal, and R. Malik. Selection and placement of decoupling capacitors in high speed systems. *IEEE Electromagnetic Compatibility Magazine*, 2(4):72–78, 2013.
- [188] M. A. Uman, Y. T. Lin, and E. P. Krider. Errors in magnetic direction finding due to nonvertical lightning channels. *Radio Science*, 15(1):35–39, Jan 1980.
- [189] K. Varelas, A. Auger, D. Brockhoff, N. Hansen, O. A. ElHara, Y. Semet, R. Kassab, and F. Barbaresco. A comparative study of large-scale variants of CMA-ES. In *International Conference on Parallel Problem Solving from Nature*, pages 3–15. Springer, 2018.
- [190] N. V. Venkatarayalu and T. Ray. Optimum design of yagi-uda antennas using computational intelligence. *IEEE Transactions on Antennas and Propagation*, 52(7):1811–1818, 2004.
- [191] N. V. Venkatarayalu, T. Ray, and Y.-B. Gan. Multilayer dielectric filter design using a multiobjective evolutionary algorithm. *IEEE transactions on antennas and propagation*, 53(11):3625–3632, 2005.
- [192] B. Wang, Y. Sun, B. Xue, and M. Zhang. Evolving deep convolutional neural networks by variable-length particle swarm optimization for image classification. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, 2018.
- [193] Q. Wang, Y. Wang, X. Li, and Q. Liu. Selection of decoupling capacitors for power delivery networks with multiple power ports. In *2015 International Conference on Power Electronics and Energy Engineering*, 2015.
- [194] T. Wang, S. Qiu, L. Shi, and Y. Li. Broadband VHF localization of lightning radiation sources by EMTR. *IEEE Transactions on Electromagnetic Compatibility*, 59(6):1949–1957, Dec 2017.
- [195] W. Wang, H. Shi, P. Huang, F. Wu, D. Fang, X. Chen, and X. Yin. An efficient variable dimension pso algorithm for mobile node tour planning in wsn. In *Proceedings of the 1st Workshop on Context Sensing and Activity Recognition*, pages 47–52, 2015.
- [196] Z. Wang, Y. Sun, X. Yang, and S. Li. Hybrid optimisation method of improved genetic algorithm and IFT for linear thinned array. *The Journal of Engineering*, 2019(20):6457–6460, 2019.
- [197] N. Weicker, G. Szabo, K. Weicker, and P. Widmayer. Evolutionary multiobjective optimization for base station transmitter placement with frequency assignment. *IEEE Transactions on Evolutionary Computation*, 7(2):189–203, 2003.

- [198] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.
- [199] K. Wu, S. Muralidharan, and M. M. Hella. Decoupling structures for millimeter wave integrated circuits in digital CMOS processes. *IEEE Transactions on Electron Devices*, 65(2):788–792, 2018.
- [200] B. Xue, X. Ma, J. Gu, and Y. Li. An improved extreme learning machine based on variable-length particle swarm optimization. In *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1030–1035. IEEE, 2013.
- [201] B. Xue, X. Ma, H. Wang, J. Gu, and Y. Li. Improved variable-length particle swarm optimization for structure-adjustable extreme learning machine. *Control and Intelligent Systems*, 42(4):1–9, 2014.
- [202] Y. Yan and L. A. Osadciw. Density estimation using a new dimension adaptive particle swarm optimization algorithm. *Swarm Intelligence*, 3(4):275, 2009.
- [203] S. Yang, Y. S. Cao, H. Ma, J. Cho, A. E. Ruehli, J. L. Drewniak, and E. Li. Pcb pdn prelayout library for top-layer inductance and the equivalent model for decoupling capacitors. *IEEE Transactions on Electromagnetic Compatibility*, 60(6):1898–1906, 2018.
- [204] H. Yuan and J. He. Evolutionary design of operational amplifier using variable-length differential evolution algorithm. In *2010 International Conference on Computer Application and System Modeling (ICCSM 2010)*, volume 4, pages V4–610. IEEE, 2010.
- [205] S. H. Zainud-Deen, W. M. Hassen, E. Ali, K. H. Awadalla, and H. Sharshar. Breast cancer detection using a hybrid finite difference frequency domain and particle swarm optimization techniques. In *2008 National Radio Science Conference*, pages 1–8. IEEE, 2008.
- [206] I. Zelinka. Somaself-organizing migrating algorithm. In *New optimization techniques in engineering*, pages 167–217. Springer, 2004.
- [207] C. Zhang, X. Fu, S. Peng, Y. Wang, and J. Chang. New multi-objective optimization algorithm for uniformly excited aperiodic array synthesis. *IET Microwaves, Antennas Propagation*, 13(2):171–177, 2019.
- [208] Y.-J. Zhang, Z. Z. Oo, X.-C. Wei, E.-X. Liu, J. Fan, and E.-P. Li. Systematic microwave network analysis for multilayer printed circuit boards with vias and decoupling capacitors. *IEEE Transactions on Electromagnetic Compatibility*, 52(2):401–409, 2010.
- [209] X. Zhao, Y. Guo, X. Chen, Z. Feng, and S. Hu. Hierarchical cross-entropy optimization for fast on-chip decap budgeting. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 30(11):1610–1620, 2011.
- [210] P. Zhou, K. Sridharan, and S. S. Sapatnekar. Optimizing decoupling capacitors in 3D circuits for power grid integrity. *IEEE Design & Test of Computers*, (5):15–25, 2009.

- [211] E. Zitzler, M. Laumanns, and L. Thiele. Spea2: Improving the strength pareto evolutionary algorithm. *TIK-report*, 103, 2001.

DEPARTMENT OF RADIO ELECTRONICS

Technická 3082/12
616 00 Brno, the Czech Republic
tel. +420 541 146 556
e-mail urel@feec.vutbr.cz
www.urel.feec.vutbr.cz

