FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
department of radio electronics

MULTI-OBJECTIVE OPTIMIZATION OF EM STRUCTURES WITH VARIABLE NUMBER OF DIMENSIONS

VÍCE-KRITERIÁLNÍ OPTIMALIZACE EM STRUKTUR S PROMĚNNÝM POČTEM DIMENZÍ

DOCTORAL THESIS (SHORT VERSION)
DISERTAČNÍ PRÁCE (ZKRÁCENÁ VERZE)

AUTHOR — ING. MARTIN MAREK
AUTOR PRÁCE

CZECH REPUBLIC, BRNO 2020

KEYWORDS

Evolutionary algorithms, EM structures design, FOPS, multi-objective optimization, variable number of dimensions, VND-GDE3, VND-MOPSO.

KLÍČOVÁ SLOVA

Evoluční algoritmus, FOPS, návrh EM struktur, proměnný počet dimenzí, více-kriteriální optimalizace, VND-GE3, VND-MOPSO.

THE THESIS IS AVAILABLE AT:

Brno University of Technology, Dept. of Radio Electronics, Technická 3082/12, 616 00 Brno, The Czech Republic.

ISBN 80-214-
ISSN 1213-418X

Contents

About the author

Martin Marek was born in Český Krumlov, Czech Republic, in June 1991. He received the B.Sc. and M.Sc. degrees in Electronics and Communication Technology from the Brno University of Technology in 2014 and 2016, respectively. He is currently working toward the Ph.D. degree in Electronics and Communication Technology at Brno University of Technology. His research interests are global optimization methods and computer-aided design in electrical engineering.

Acknowledgments

I would like to thank my supervisor Ing. Petr Kadlec Ph.D. for his support, motivation, and knowledge in the field every time it was needed.

1 Introduction

These days, optimization is used in almost every discipline of engineering. Optimization is a process of finding the best solution from a set of available solutions. The quality of a solution is defined by fitness values calculated from fitness (objective, cost) functions. The fitness functions describe the behavior of an optimized system with properties called decision variables e.g. dimensions, reliability, or a price of a product. Therefore, fitness values depend on the decision variables of the optimized system. The optimization process is a process of finding minima (or maxima) of fitness functions.

Most of the real-world optimization problems are by its nature multi-objective and the objectives are also conflicting. In such case, the result of optimization is a set of trade-off solutions called Pareto-front [1]. This aimed the research to develop various multi-objective optimization methods.

A common optimization problem has a fixed number of decision variables. Therefore, the optimization algorithm knows the dimensionality of the decision space and tries to find the optimal position. Its fitness function depends only on decision variables. However, there are some optimization problems where the fitness function depends on the number of components of the decision vector as well. Such problems are called problems with a Variable Number of Dimensions (VND). When optimizing VND problems, an algorithm has to find not only the proper position vector but its dimension as well.

2 Theoretical background

Global optimization methods are preferred because the risk of being caught in a local optimum of a fitness function is significantly lower compared to the local optimization methods. Usually, the global optimization method works with multiple sets of decision variables that are modified in each iteration of an algorithm. These population-based and stochastic methods, inspired by the theory of evolution, are generally called Evolutionary Algorithms (EAs), which covers Genetic Algorithms (GA) [2], Differential Evolution (DE) [3], etc.

Although the researchers initially worked with single-objective optimization problems, it was only by assuming huge simplifications. As was mentioned before, most nature-inspired optimization tasks are multi-objective ones. Therefore, the need for multi-objective optimization algorithms soon emerged.

2.1 Multi-objective Optimization

The most popular and well-known multi-objective optimization algorithms are – Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) [4], the successor of GA; Generalized Differential Evolution (GDE3) [5], the successor of DE; and other EAs and their numerous modifications [6].

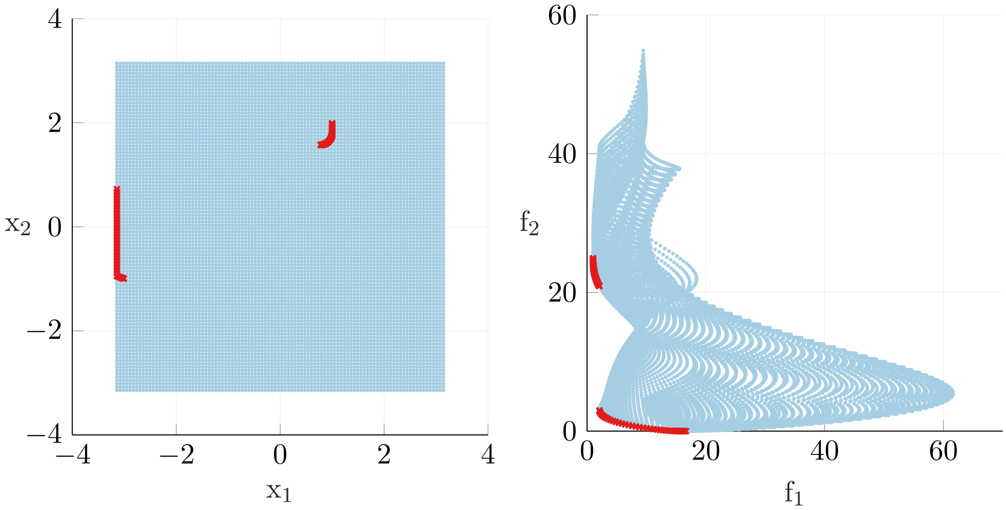
The multi-objective optimization algorithm confronts two fundamental requirements:

- Minimize the distance between found solutions produced by the optimization algorithm and the true Pareto-front.
- Maximize the spread of found trade-off solutions, so the solutions are distributed as uniformly as possible over the whole Pareto-front.

The multi-objective optimization process deals with a finite number of fitness functions. There are two spaces in multi-objective optimization – the decision space and the objective space. Both spaces are connected by fitness functions. Figure 1 shows both spaces of Poloni's study [7] (dark red "×" solutions represent the true Pareto-front).

The result of multi-objective optimization is a set of solutions, which makes it hard to decide whether one set of solutions is better than the other [7]. In order to simplify the decision, a performance metric represents an entire Pareto-front with a single number. The value of a metric depends on the quality of the solution.

There are two metrics used for the performance assessment of the pro-



(a) Decision space **(b)** Objective space

Figure 1: Both spaces of the Poloni's study. Dark red "×" - true Pareto-front, light blue "·" - dominated solutions.

posed method in this thesis – generational distance (GD) [8] and distance hypervolume (dHV) [9].

3 Motivation

The thought of the optimization problems with a variable number of dimensions is almost as old as the optimization algorithms themselves [10]. However, for many years, researchers aimed for the simplicity of the optimization process and worked with fixed-length decision vectors. In 1989, the author of the original Genetic Algorithms stated in [11] that "Nature has formed its genotypes by progressing from simple to more complex life forms" and proposed the Messy Genetic Algorithm (mGA) - a first algorithm to be found working with a variable number of dimensions. Readers interested in the survey of existing VND algorithms are encouraged to read a full version of the dissertation thesis.

In the literature, most of the VND applications are tackled by VND algorithms that are single-objective, but a multi-objective definition of an optimization problem is more natural, because the nature itself is full of contrasts. Leaving out any criterion to fit the problem for a single-objective optimization algorithm may be tricky, if not odd, entirely.

However, only a few multi-objective algorithms were adapted to work with a variable number of dimensions so far [12, 13, 14]. It is also important to note that some papers claim a multi-objective VND algorithm is being proposed, but in truth they are either:

- quasi multi-objective – aggregates several objectives into one [12],
- impure-VND – performs update position operator with fixed-length decision vectors [13, 14].

Aggregating multiple objectives into a single one is a tricky issue. One has to have some additional knowledge about the optimized problem in order to satisfyingly set the aggregating method and therefore obtain a good trade-off solution. However, the fact that the problem properties are unknown is often the original reason the optimization is performed.

The method is called impure-VND in this thesis if a VND problem is being optimized, but a fixed decision space is eventually used in the update position operator. Although such an algorithm may be able to find optimal dimensionality, it necessarily performs position update with the whole position vectors. Therefore, the performance of an algorithm is wasted on exploring unfeasible regions of decision space (regions unused in fitness function evaluation). Note that only a few single-objective VND algorithms in the literature can be considered to be pure VND.

4 Dissertation Objectives

The following list summarizes the most important objectives of this thesis:

- develop an optimization framework suitable for algorithms with a variable number of dimensions,
- create a library of benchmark problems with a variable number of dimensions,
- propose new algorithms for optimization with a variable number of dimensions,
- verify the performance of the proposed methods on a set of benchmark problems,
- exploit new algorithms on several real-world applications.

Implementing novel optimization techniques in any optimization framework is a beneficial step in the design process. The optimization framework not only simplifies the designing of the algorithm but its setting and verifying as well. However, maintaining problems with a variable number of dimensions casts a special requirement on the framework itself. Due to the requirement that any agent in a population can have a different number of components, no existing optimization framework was suitable for our cause. Therefore, a new optimization framework in MATLAB was developed, which makes it easier to implement an algorithm, run a simulation, view its results, or compare its performance to other algorithms.

The library of benchmark problems is a necessary part of the verification of the algorithm's convergence properties. A proper set of benchmarking problems has several individual problems with all kinds of difficultie. Advantageously, if the true Pareto-front of a benchmark problem is known, it is possible to compare the found Pareto-fronts to the true Pareto-front.

Deriving new stochastic optimization methods is a crucial part of this thesis. They are exploited on a special class of optimization problems – problems with a variable number of dimensions. Such an optimization algorithm not only determines proper values of the decision vector but the number of decision variables as well.

Verifying the performance of proposed methods is a substantial step in the design process. Each novel algorithm is compared to the algorithm with a fixed number of dimensions and a hybrid-VND method in a scenario that tries not to favor any of the methods.

Finally, novel algorithms are exploited on several real-world applications related to the field of electrical engineering:

- Anisotropic band-stop filter design,
- Antenna array design,
- Transmitter placement problem,
- Digital circuit design,
- Automated image thresholding,
- Clustering problem.

Note that only the antenna array design problem and the automated image thresholding problem are presented in this shortened version of the thesis.

5 FOPS

The motivation of researches for the development of optimization techniques is that there simply does not exist a universal method suitable for all kinds of optimization problems. Wolpert proposed the "no free lunch" theorem for the optimization [15]. It says that for any algorithm, any elevated performance over one class of problems is exactly paid for over another class in performance. Therefore, when an unknown optimization problem is given, it is common to use various optimization methods and see which one serves the best to our needs. Such practice encourages the development of optimization frameworks.

There are several MOEA frameworks to be found in the literature (see survey in [16]). However, none of them naturally allows an implementation of pure VND algorithm where each agent has a different number of decision variables. Therefore, a new optimization framework – Fast Optimization ProcedureS (FOPS) was developed.

The FOPS is a standalone MATLAB toolbox. Therefore, it is available for various operating systems. It includes 16 methods for single- and multi-objective optimization. Currently, there are seven single-objective (plus one single-objective VND) and four multi-objective (plus two multi-objective VND) optimization methods, almost 110 benchmark problems of various types, and many performance metrics such as generational distance, hypervolume, spread, etc. The FOPS can be controlled from the command line or by graphic user interface.

Visualization of the results is an important part of the optimization process since it is capable of disclosing a relationship between different quantities. The FOPS framework has many features there are beyond the extent of this thesis. Some of them were presented in [MM1, MM2]. Readers interested in the FOPS framework are referred to paper [MM3] where the FOPS is presented or its documentations [MM4].

Figure 2 shows the results suite when results of optimization task are visualized. Tables in the lower half of the figure contain positions and fitness values of the non-dominated set. The upper half of the figure shows the control panel. There is an vizualization of the fitness values of a custom optimization task presented on the right side of the figure. Green points depict the true Pareto-front, red points build the non-dominated set, and blue points depict the fitness values from consecutive iterations.

The main advantage of an in-house optimization toolbox for MATLAB is that the implementation of a new optimization algorithm is simple and straightforward. Within moments, the user implements the unique properties of a given algorithm. The procedures that are common for many optimization algorithms (e.g. initialization of population, non-dominated sorting, etc.) are already defined and can be utilized without effort. Afterward, the verification of a new algorithm is easily performed by predefined optimizing routines on a set of benchmark problems that are already defined too.

The versatility of the framework is demonstrated in Section 9. Various features of the FOPS framework are exploited there. It was utilized in many research papers [MM5, MM1, MM2, MM6, MM7, MM8, MM9, MM10, MM11, MM12, MM13, MM14].

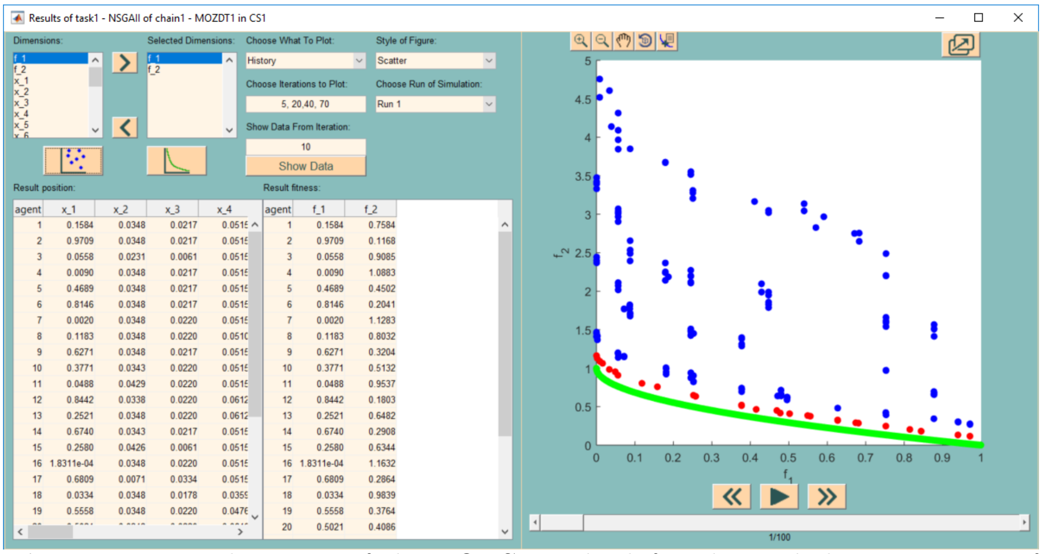


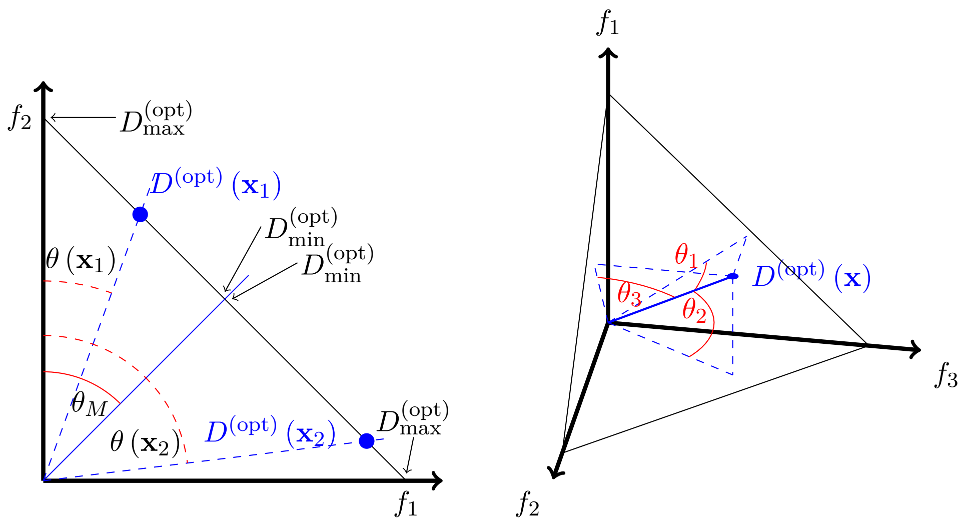
Figure 2: Results suite of the FOPS on the left side, and the animation of the fitness values of MOZDT1 problem on the right.

6 Multi-objective Testing Problems with Variable Number of Dimensions

When comparing the performance of multiple optimization algorithms, it is convenient to use testing problems with known true Pareto-front. Therefore, the non-dominated sets can be compared and qualified.

Single-objective VND benchmark problems were proposed in [17] or [18]. Before these publications, most of the studies were validated by arbitrary and often simplified real-world problems. Speaking of a multi-objective problems with a variable number of dimensions, no such library of benchmark problems can be found in the open literature (to the authors' knowledge).

This subsection proposes the methodology for creating multi-objective VND benchmark problems based on the idea of [19]. Authors of that paper say that different parts of the Pareto-front may have different sizes in real-world optimization problems. Afterward, they constructed a few problems with linearly-shaped Pareto-fronts where the number of decision variables of the optimal solution is determined by the angle between the line connecting the solution with the coordinate's origin and the f_2 axis (2-dimensional Pareto-fronts) or individual coordinate planes (3-dimensional Pareto-fronts).



(a) Two-dimensional objective space.

(b) Three-dimensional objective space.

Figure 3: Construction of Pareto-fronts of general two- and three-objective VND problems.

The dimensionality of an arbitrary solution on the Pareto-front is determined by the following process:

$$D^{(\mathrm{opt})}\left(\mathbf{x}\right) = D_j^{(\mathrm{opt})}, \tag{1}$$

where the dimensionality is selected from the list $D^{(\mathrm{opt})} = \left\{D_{\min}^{(\mathrm{opt})}, \ldots, D_{\max}^{(\mathrm{opt})}\right\}$ with N_D members and the index is defined by:

$$j = \begin{cases} 1 + \left\lfloor \left(1 - \frac{\theta(\mathbf{x})}{\theta_M}\right)(N_D - 1)\right\rfloor, \text{for} \quad \theta\left(\mathbf{x}\right) <= \theta_M \\ \\ 1 + \left\lfloor \quad \left(\frac{\theta(\mathbf{x})}{\theta_M}\right)(N_D - 1)\right\rfloor, \text{for} \quad \theta\left(\mathbf{x}\right) > \theta_M, \end{cases} \tag{2}$$

where θ_M is the maximal angle shown in Figure 3 and is set to 45° in our study. Pareto-fronts are divided into two regions, where the dimensionality of the solution gradually decreases in the first region and gradually increases in the second region. The angle $\theta\left(\mathbf{x}\right)$ for two-dimensional Pareto-fronts is defined as:

$$\theta\left(\mathbf{x}\right) = \arccos\left(\frac{f_2}{\sqrt{f_1^2 + f_2^2}}\right) \tag{3}$$

and for three-dimensional Pareto-fronts:

$$\theta(\boldsymbol{x}) = \arccos\left[\max_{i=\{1,2,3\}}\left(\frac{f_i}{\sqrt{f_1^2 + f_2^2 + f_3^2}}\right)\right]. \tag{4}$$

Note that f_1, f_2, and f_3 denote fitness values of the first, second, and third objective for vector $\mathbf{x}$, respectively. The description of VND modifications of ZDT1 and MODTLZ2 problems can be found in the full version of the thesis. The definition of other benchmark problems used later in Section 8 is also there.

7 VND-GDE3

The idea of handling the variable decision space applied to the multi-objective GDE3 comes from PSO-VND algorithm [20]. At first, the GDE3 algorithm will be described and the VND-GDE3 is presented afterwards. Note that the full version of the thesis shows also a description of VLGDE3 algortihm. The VND-GDE3 is considered to be a pure-VND algorithm while the VLGDE3 is not.

7.1 GDE3

Generalized Differential Evolution (GDE3) is based on a Differential Evolution algorithm proposed in 1997 [3]. It is a population-based real-numbered optimization algorithm with selection and crossover operators. A random initial population is created at the beginning. Agents' positions are then altered to find better solutions in every iteration until a stopping criterion (usually predefined number of iterations) is met.

The crossover procedure is performed by constructing a trial vector ($\mathbf{u}_{i,g}$) for each decision vector ($\mathbf{x}_{i,g}$) of the population. Here, i is the index of an agent in the population, and g is an iteration (generation) index. The trial vector is derived with following pseudocode:

Algorithm 1: Pseudocode of crossover operator in GDE3.

```
r1, r2, r3 ∈ {1, 2, ... N};
r1, r2, r3 ≠ i;
j_rand ∈ 1, 2, ... D;
for (j = 1 : D) do
    if (rnd(1) < P_C ∨ j = j_rand) then
        u_{j,i,g} = x_{j,r3,g} + F · (x_{j,r1,g} − x_{j,r2,g});
    else
        u_{j,i} = x_{j,i};
    end
end
```

where N denotes the number of agents in population, j denotes the j-th decision variable, D is the number of decision variables, F is the scaling factor, r_1,r_2, and r_3 are randomly selected agents' indices (mutually different and different from i). Not all the trial vectors replace all the old vectors. The ratio of replaced vectors is controlled by the crossover probability (P_C).

In multi-objective optimization, where the objectives are conflicting, a set of trade-off solutions constitute the Pareto-front. GDE3 selects trade-off solutions based on the dominance principle [7, Chapter 2] as follows:

- If the old vector dominates the trial vector, the old vector remains as it is for the next iteration.
- If the trial vector dominates the old vector, the trial vector replaces the old vector for the next iteration.
- If the old vector and the trial vector are non-dominated, both vectors are selected, and the population is temporarily extended.

However, the size of the population has to be limited due to increasing computational demands. When the trial and old solutions are non-dominated, both solutions remain in the population. If there are more non-dominated solutions than the number of agents, the extended population is trimmed using the crowding distance metric. This approach enhances the diversity of the solution.

7.2 VND-GDE3

Algorithm VND-GDE3 [MM6] is very similar to its predecessor – GDE3. Nonetheless, its agents can have a different number of decision variables in any iteration. The only differences are related to the crossover of the decision vectors of different sizes. Moreover, if a problem has a fixed number of decision variables, the VND-GDE3 acts identically as the GDE3 algorithm. Although, it might be slightly more computationally demanding. Variability of solution's dimensionality introduces a new user-defined parameter probability of dimension transition (P_{DT}).

The crossover operator in Algorithm 1 mixes the decision variables of three different agents into a single trial decision variable with a probability of crossover P_C. Otherwise, the decision variable remains as it is.

The number of decision variables may differ between the three vectors. Therefore, the dimensionality of a trial vector D_{new} has to be determined beforehand. Dimensionality D_{new} is one of the following:

- dimensionality of the current agent D_i,
- dimensionality of the first randomly picked agent D_{r_1},
- dimensionality of the second randomly picked agent D_{r_2},
- dimensionality of the third randomly picked agent D_{r_3}.

The dimensionality of the trial solution D_{new} is equal to D_i with the probability P_{DT}. Otherwise, it is one of the dimensionalities D_{r_1}, D_{r_2}, or D_{r_3} (picked with equal probability).

Afterward, four artificial agents are derived from the current agent and random agents (r_1, r_2, and r_3), but they all have the same size D_{new}. Note that missing decision variables are filled randomly and that only an undivided part of the decision space vector can be deleted from the ending part of it (please refer to [20], Figure 1).

8 Performance Assessment

This section shows the comparison of VND-GDE3 algorithm, its impure-VND peer VLGDE3, and also Clustered-GDE3. The Clustered-GDE3 represents a non-VND approach used in problems with a variable number of dimensions. In the full version of the thesis the comparison includes also VND-MOPSO and VLMOPSO algorithms.

Controlling parameters of all the GDE3-based algorithms are: the scaling factor $F = 0.2$, the probability of crossover $P_C = 0.2$, the number of agents $N = 400$, and the number of iterations $G = 200$. Note that the number of

Table 1: List of parameter settings used for VND-GDE3 simulations.

SET	N	G	**Algorithm**	$\lvert D\rvert$	D	$D^{(opt)}$
C1	400	200	VND-GDE3	10	{3,4,...,12}	{3,4,5}
C2	400	200	VLGDE3	10	{3,4,...,12}	{3,4,5}
C3	40*	200	Clustered-GDE3	10	{3,4,...,12}	{3,4,5}
C4	400	200	VND-GDE3	50	{3,4,...,52}	{10,11,12}
C5	400	200	VLGDE3	50	{3,4,...,52}	{10,11,12}
C6	8*	200	Clustered-GDE3	50	{3,4,...,52}	{10,11,12}
C7	400	200	VND-GDE3	80	{3,4,...,82}	{15,16,17}
C8	400	200	VLGDE3	80	{3,4,...,82}	{15,16,17}
C9	5*	200	Clustered-GDE3	80	{3,4,...,82}	{15,16,17}
C10	400	200	VND-GDE3	100	{3,4,...,102}	{20,21,22}
C11	400	200	VLGDE3	100	{3,4,...,102}	{20,21,22}
C12	4*	200	Clustered-GDE3	100	{3,4,...,102}	{20,21,22}

*Number of agents N of each cluster.

agents in each cluster in the Clustered-GDE3 algorithm vary according to the number of dimensionalities $|D|$ of the problem. Also note that VND-GDE3 algorithm has the probability of dimension transition P_{DT} parameter. It was set to $P_{\mathrm{DT}} = 0.35$ according to comparative study described in the full version of the thesis.

8.1 Comparison of Clustered, Pure-VND, and Impure-VND Approaches

To perform a fair comparative study between the VND algorithm and a standard non-VND GDE3 algorithm, a reasonable approach is to use several simple GDE3 runs (clusters), each with a different number of decision variables. The number of agents of all the clusters summed together is identical to the number of agents of VND-GDE3 and VLGDE3 algorithms. The number of iterations remains fixed for all clusters. The Pareto-fronts from each separate run are combined, and the non-dominated solutions constitute the resulting non-dominated set. We have modified algorithm GDE3 accordingly and it is called the Clustered-GDE3 algorithm.

Three algorithms – VND-GDE3, VLGDE3, and Clustered-GDE3 – were exploited on a set of ten benchmark problems with four dimensionality settings (see Table 1). Figure 4 shows visualization of performance on the modified UF4, UF6, and UF10 problems. The visualization is in the form of standard boxplots. There are 12 boxes for each problem, where each triplet corresponds to a single dimensionality scenario. It is clearly visible that metric values deteriorate as the dimensionality of the problem grows. Average values of distance hypervolume and generational distance metric can be seen in Tables 2 and 3. Table 4 shows results of Friedman's and Wilcoxon's non-parametrical statistic tests on the distance hypervolume metric values.

The decision space in the case of sets C.1 – C.3 is much smaller than

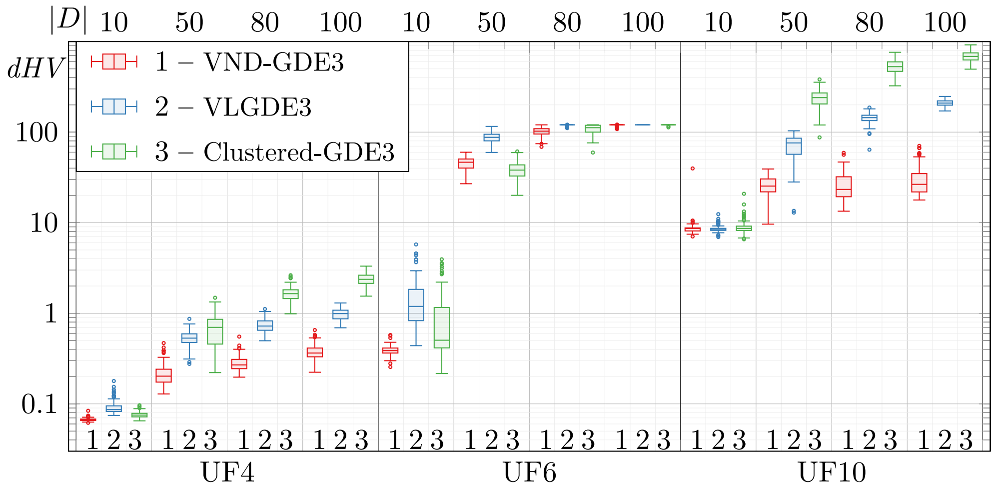


Figure 4: Comparison of VND-GDE3, VLGDE3, and Clustered-GDE3 on UF problems.

Table 2: Average distance hypervolume (dHV) for a given set of settings.

SET	ZDT2	ZDT4	ZDT6	DTLZ2	DTLZ4	LZ6	LZ8	UF4	UF6	UF10
C1	0.0207	0.0271	0.0310	1.7304	0.1236	4.9107	0.2828	0.0671	0.3867	8.8742
C2	0.0024	0.0128	0.0058	0.4432	0.0792	5.0128	0.8391	0.0929	1.4852	8.5796
C3	0.0302	0.0701	0.0090	1.7117	1.7053	3.3450	0.6461	0.0760	0.9408	8.9979
C4	0.1313	15.022	0.2731	3.2684	0.1977	5.9720	32.947	0.2162	45.531	25.819
C5	0.0894	0.7085	0.3519	0.4510	0.0748	6.3874	71.850	0.5374	87.468	70.869
C6	13.139	24.618	49.465	4.3554	10.258	24.974	24.252	0.6988	38.402	237.10
C7	0.2345	50.995	1.2437	3.1595	0.1588	6.2069	91.999	0.2826	101.51	26.826
C8	1.9930	8.2417	14.052	0.2922	0.1864	7.4858	115.75	0.7452	119.90	143.09
C9	26.892	113.62	73.608	9.1589	16.830	61.452	83.214	1.6444	107.76	539.15
C10	0.0777	102.55	6.9278	2.9256	0.1372	6.5641	119.66	0.3813	119.79	30.886
C11	4.5649	29.304	27.997	0.3758	0.3771	8.9248	120.63	0.9875	120.44	209.01
C12	34.939	120.67	79.665	17.118	31.736	97.288	119.80	2.4183	120.32	692.18

Table 3: Average generational distance (GD) for a given set of settings.

SET	ZDT2	ZDT4	ZDT6	DTLZ2	DTLZ4	LZ6	LZ8	UF4	UF6	UF10
C1	0.0110	0.0015	0.0170	0.0064	0.0142	0.0339	0.1284	0.0329	0.4160	2.4548
C2	0.0000	0.0001	0.0002	0.0052	0.0110	0.0277	0.6986	0.0545	1.2193	2.2937
C3	0.0005	0.0033	0.0001	0.0388	0.0423	0.4862	0.4084	0.0367	0.9339	1.2889
C4	0.0165	0.8824	0.0513	0.0067	0.0313	0.0176	8.1427	0.0902	11.133	0.8285
C5	0.0021	0.0012	0.0030	0.0065	0.0152	0.0506	10.526	0.1088	12.267	1.8487
C6	0.6477	1.8461	3.8178	0.1327	0.1117	0.1881	6.4253	0.1648	8.7900	1.9795
C7	0.0202	3.9856	0.0820	0.0066	0.0352	0.0167	20.513	0.1069	25.977	0.5614
C8	0.0361	0.0634	0.3935	0.0151	0.0520	0.0766	14.964	0.1243	17.364	2.2798
C9	1.6960	12.405	5.8774	0.4118	0.3487	0.3188	19.921	0.3756	29.014	3.6947
C10	0.0252	9.4347	0.1221	0.0067	0.0350	0.0195	34.023	0.1243	44.107	0.5244
C11	0.0919	1.7927	1.6603	0.0331	0.1199	0.0950	17.704	0.1390	19.833	2.5466
C12	2.4015	31.700	6.4461	0.7774	0.7345	0.4403	40.204	0.5517	53.312	4.4521

the decision space in the case of sets C.10 – C.12. Therefore, the Clustered-GDE3 was able to find decent Pareto-fronts because the algorithm deliberately searched every dimensionality of the problem in the low dimensionality scenario. Forty agents for each cluster was enough to explore the correspond-

Table 4: Results of Friedman's test / Wilcoxon's test (at significance level $\alpha = 0.05$): + denotes that the first setting is significantly better, – denotes that the second settings is significantly better, = denotes that the difference is not significant.

Compare SET	ZDT2	ZDT4	ZDT6	DTLZ2	DTLZ4	LZ6	LZ8	UF4	UF6	UF10
C.1 vs. C.2	-/-	-/-	-/-	-/-	=/-	=/=	+/+	+/+	+/+	=/=
C.1 vs. C.3	-/-	-/-	-/-	=/=	+/+	-/-	+/+	+/+	+/+	=/=
C.4 vs. C.5	-/-	-/-	=/+	-/-	-/-	+/+	+/+	+/+	+/+	+/+
C.4 vs. C.6	+/+	+/+	+/+	=/+	+/+	+/+	-/-	+/+	-/-	+/+
C.7 vs. C.8	+/+	-/-	+/+	-/-	=/+	+/+	+/+	+/+	+/+	+/+
C.7 vs. C.9	+/+	+/+	+/+	+/+	+/+	+/+	=/-	+/+	+/+	+/+
C.10 vs. C.11	+/+	-/-	+/+	-/-	+/+	+/+	=/+	+/+	+/+	+/+
C.10 vs. C.12	+/+	+/+	+/+	+/+	+/+	+/+	=/=	+/+	+/+	+/+

ing dimensionality sufficiently. Contrarily, the Clustered-GDE3 spent most of its efforts searching in non-optimal dimensions in the high dimensionality scenario.

The same applies when comparing VL to VND algorithms. VLGDE3 performs much better in the low dimensionality scenarios. However, padding decision variables makes the decision space of an optimization problem harder to explore. Therefore, the performance of the VL algorithm deteriorates with the growing number of decision variables of the problem quicker compared to the VND-GDE3 algorithm.

Table 4 shows the results of non-parametric statistical testing. The comparison seems balanced if test problems have only ten possible dimensionalities. However, the VND-GDE3 algorithm outperforms VLGDE3 in most of the problems in the case of a hundred possible dimensionalities. Note that the signs in the table are results of Friedman's and Wilcoxon's non-parametric tests with a level of significance $\alpha = 0.05$. Friedman's unadjusted p-values were adjusted by Holmberg's posthoc procedure.

9 Applications

The FOPS optimization framework was used to study the performance of VND-GDE3 and VND-MOPSO algorithms. Studies were carried out with benchmark problems, i.e., with analytically prescribed fitness functions with known minima. However, strengths of FOPS lie in a real-world application. The FOPS toolbox was designed to be as versatile as possible for real applications that have various needs. It is also used as an internal optimizer of the Antenna Toolbox for MATLAB (AToM [MM15]).

The full version of the thesis includes design of a band-stop filter[MM5, MM3], hybrid optimization problem [MM3], transmitter placement problem [MM3], synthesis of digital circuits problem, and clustering problem. Here, in this shortened version, only the linear antenna array problem and the automated image thresholding problem are presented.

VND algorithms were also exploited in other applications such as [MM9, MM12, MM8, MM10]. However, these applications are not discussed in this thesis.

9.1 Linear Antenna Array Design

This section aims to delineate the difference between problem formulation with a fixed number of dimensions and a variable number of dimensions. A linear antenna array consists of ν antennas distributed alongside the x-axis. In this particular example, all the constituent antennas are identical. Therefore, the formulation of the total radiation vector of the antenna array is simplified to [21]:

$$\mathbf{F}_{\text{tot}}(\mathbf{k}) = \mathbf{A}(\mathbf{k})\,\mathbf{F}(\mathbf{k}), \tag{5}$$

where $\mathbf{k} = k\mathbf{r}$ is the wave number vector, k is the free space wave number, and $\mathbf{r}$ is the position vector. The radiation vector of an elementary antenna $\mathbf{F}(\mathbf{k})$ is multiplied by the array factor $\mathbf{A}(\mathbf{k})$. The array factor is determined by the array configuration, and it is defined as:

$$\mathbf{A}(\mathbf{k}) = \sum_{i=1}^{N} a_i \exp\left(j\mathbf{k}\cdot\mathbf{d}_i\right). \tag{6}$$

Here, j is the imaginary unit, a_i is the complex number representing excitation amplitude and phase, and $\mathbf{d}_i$ is the position of the i-th antenna.

Antenna array properties that are of interest in this study are the Side-Lobe Level (SLL) and the number of antennas ν. The problem formulation for a fixed number of decision variables commonly utilizes the uniform grid (UG). If UG is used, a particular antenna is activated or deactivated according to the decision vector. Therefore, the distribution grid of antennas has to be determined a priori, and it might affect the overall performance of the optimization. Contrarily, the VND formulation of the problem allows an algorithm to not only find the proper number of antennas but to find the optimal positioning on the x-axis as well.

The multi-objective problem using UG is formulated as follows:

$$f_1 = \mathrm{SSL}(\mathbf{x}) \tag{7}$$

$$f_2 = \nu(\mathbf{x}), \tag{8}$$

where both objectives are to be minimized, and the $\mathbf{x}$ is the decision vector. In the case of a uniform grid, the decision vector is a binary string of fixed length equal to the possible number of antennas. In our comparative study, the array can contain up to 100 antennas, and the gap between two grid

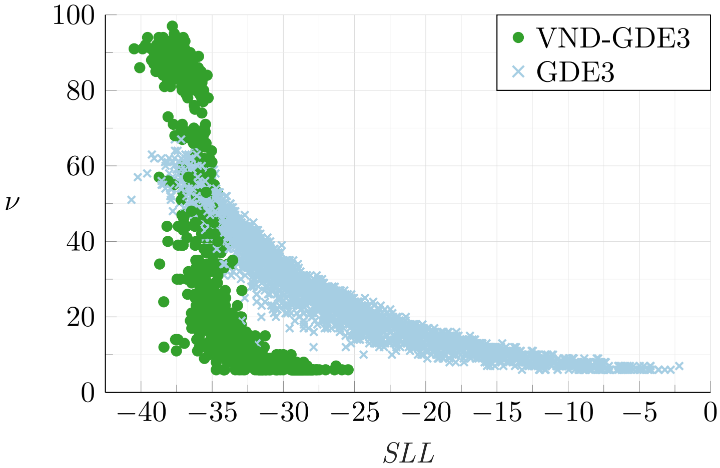


Figure 5: Comparison of VND-GDE3 (VND formulation) and GDE3 (uniform grid formulation) algorithms on the linear antenna array problem.

positions is one-quarter of the wavelength ($0.25\,\lambda_0$).

The variable number of dimensions formulation gives:

$$f_1 = \mathrm{SSL}\left(\mathbf{x}, n\right) \tag{9}$$
$$f_2 = \nu\left(\mathbf{x}, n\right), \tag{10}$$

where $\mathbf{x}$ is the vector of n values expressing the gaps between consecutive elements of the array. Note that the first antenna of the array is placed at $x = 0\,m$. The gap between individual elements can vary according to the interval of $x_i \in [0.25\,\lambda_0, \lambda_0]$. The number of elements in the antenna array is $\nu = n + 1$.

The problem with the uniform grid was optimized by the standard GDE3 algorithm with default properties as defined in [MM4]. The VND formulation of the problem was optimized with the VND-GDE3 algorithm with default settings (i.e. SET A.3). Both algorithms used 200 agents over 200 iterations. Results shown in Figure 5 accumulates 100 independent runs. The solutions marked with "$\times$" signs belong to the uniform grid formulation and the solutions marked with "$\bullet$" signs belong to the VND formulation. The VND-GDE3 algorithm outperforms the standard GDE3 algorithm, especially from the viewpoint of the number of the used antennas. The only drawback of the VND-GDE3 method is that the average number of non-dominated solutions found per one run is lower ($N = 10.62$) than that of standard GDE3 ($N = 24.88$).

9.2 Image Thresholding Problem

Image thresholding is the simplest method for digital image segmentation. The idea of image thresholding is to find the optimal threshold value that divides the gray-level image into "object" pixels (gray level is greater than the threshold value) and "background" pixels (gray level is lower than the threshold value) [22].

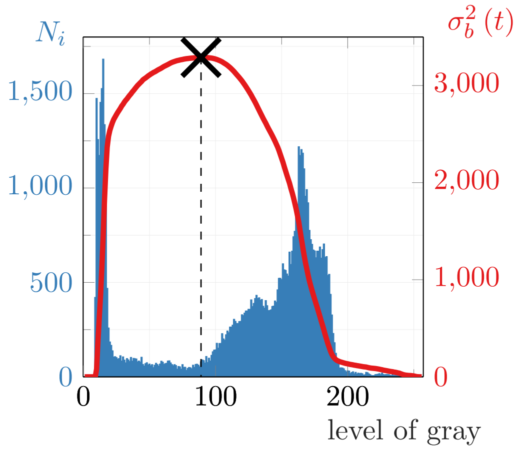
Examples of thresholding applications can be a document image analysis (Optical Character Recognition) [23], x-ray computed tomography [24] or license plate image recognition [25].

The most famous automated image thresholding method is Otsu's method [26]. It is a histogram shape-based method and assumes two distinct peaks in the histogram. Therefore, the threshold value separates the two classes of gray-levels so that the intra-class variance $\sigma_b^2(t)$ is maximized.

Otsu's method calculates the inter-class variance exhaustively. Figure 6b shows the histogram of the cameraman testing image (blue bar graph) and the corresponding values of inter-class variance (red line). The black cross marker shows the threshold value found by Otsu's method – $t = 89$. Figure 6a shows the cameraman testing image translated into a binary image using the threshold $t = 89$.

Therefore, in thresholding tasks where multiple thresholds are sought, the number of inter-class variance calculation exponentially grows. The motivation for using an evolutionary algorithm to solve a multi-threshold thresholding problem is evident. The problem is formulated as a multi-objective one

(a) Thresholded cameraman image.



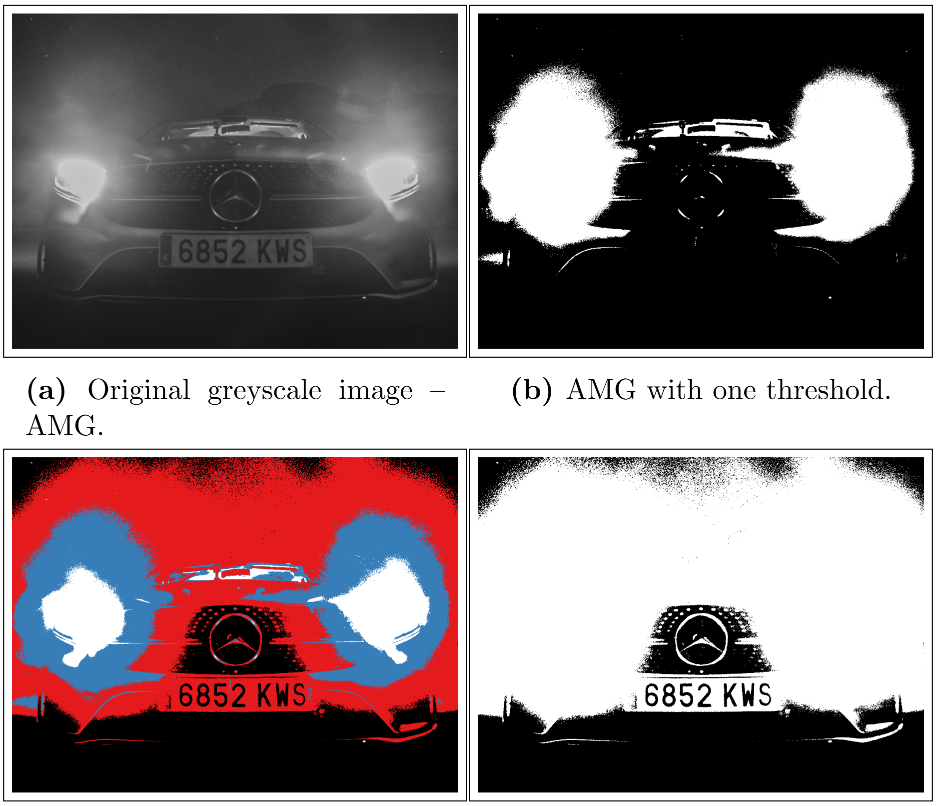
(b) Histogram of the cameraman image (blue) with inter-class variance values found by Otsu's method (red).

Figure 6: Otsu's method explained on the cameraman testing image.

with a variable number of dimensions where the value of Otsu's inter-class variance is used as the first fitness function. The number of thresholds stands for the second fitness function. Note that the variability of the number of thresholds is in accordance with the VND formulation.

9.2.1 License Plate Recognition by Using Thresholding

License plate recognition (LPR) is widely used in many real-life situations including parking lot attendance, traffic laws enforcement, etc. An image with a license plate is pre-processed before the optical character recognition of the license plate by image thresholding [25]. However, the thresholding in the license plate recognition is not a trivial task [28]. The following figures in this subsection show that lighting conditions have a major impact on the performance of the thresholding method. Note that image analysis in LPR consists of three parts: localization of the license plate in the image, thresholding of the license plate region, and optical character recognition of the characters. However, the localization and OCR steps are out of the



(a) Original greyscale image – AMG. **(b)** AMG with one threshold.

(c) AMG with three thresholds. **(d)** Three thresholds – first layer.

Figure 7: LPR testing image – AMG [27].

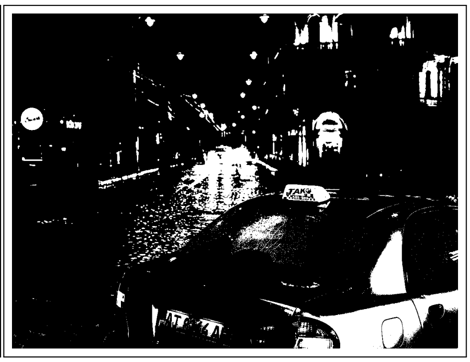
scope of this thesis. Therefore, the image thresholding is performed on the whole testing image instead of the section with the license-plate. Such an approach makes the thresholding more challenging. Nonetheless, it better demonstrates the advantage of the variable number of dimensions approach.

The thresholding technique uses a variable number of dimensions representation with two objectives to find trade-off solutions with multiple threshold values. It is shown that the number of thresholds needed for the proper image segmentation cannot be determined a priori.

Figures 7 – 8 show testing images for license-plate recognition. There are four subfigures for each figure where the top-left shows the original greyscale image and the remaining three subfigures show results of the thresholding method. Note that only a limited number of subfigures is shown in this shortened version of the thesis.

Figure 7 shows the AMG testing image. In this picture, the light from the headlamps is much brighter than the background of the license plate. Therefore, using just one threshold creates two layers, but the license plate is apparent in neither of them. Contrarily, if three thresholds are used, the

(a) Original greyscale image – Taxi.



(b) Taxi with one threshold.

(c) Taxi with two thresholds.

(d) Two thresholds – first layer.

Figure 8: LPR testing image – Taxi [29].

license plate is clearly visible in the first layer (see Figure 7d). The subsequent OCR routine would most likely yield the desired "6852 KWS".

Figure 8 shows the Taxi testing image. Light conditions in this image are very difficult. Moreover, the illumination from the license plate lamps makes the thresholding task even harder. It can be seen in Figure 8b that most of the characters in the license plate blends with the background. Figures 8c and 8d shows the thresholding with two thresholds. In this case all the characters in the license plate fell into the first layer. Contrarily, the whole background of the license plate is in the second or third layer. Therefore, the character recognition task is simple if two thresholds are used.

10 Conclusion

This dissertation thesis deals with multi-objective evolutionary optimization with variable number of dimensions. Many real-life optimization tasks use decision vectors of variable length by nature. Although standard optimization algorithms with a fixed number of dimensions can solve such tasks, either the computational demands are much higher compared to the algorithms with a variable number of dimensions, or the representation of the problem has to be simplified. Therefore, the risk of losing decision space resolution emerges.

The idea of optimization algorithms with variable number of dimensions is probably as old as optimization algorithms itself. However, the research of optimization methods with variable number of dimensions is rather marginal compared to the fixed-length one. The survey of work in the field of optimization with a variable number of dimensions showed that there are many of them, but they are mostly only single-objective or can not work with the decision vectors of uneven lengths in the pure-VND nature.

Particle Swarm Optimization for Variable Number of Dimensions is one of the algorithms that is considered to be the pure-VND algorithm. However, it is a single-objective optimization algorithm. Nonetheless, the employed methodology for handling the vectors of different lengths was successfully applied in a multi-objective version of Particle Swarm Optimization. That gave birth to the VND-MOPSO algorithm. Similarly, a multi-objective Differential evolution-based algorithm with a variable number of dimensions was derived from GDE3 – the VND-GDE3 algorithm [MM6].

Novel methods were verified by comparative studies against their impure-VND peers and also against the Clustered-GDE3 method. The Clustered-GDE3 method represents the standard algorithm with a fixed number of dimensions applied to problems with a variable number of dimensions. It was shown that a pure-VND algorithm outperforms both opponents, especially if the number of dimensionalities is large.

Before the comparison of the novel methods against others, the study of their setting parameters had to be carried out. Both these controlling parameters (the probability of dimensions transition in the VND-GDE3 and the probabilites to follow in the VND-MOPSO) control the behavior of the algorithm in search of the optimal dimensionality of the problem. Studies of other controlling parameters, common with non-VND peers, are unnecessary because the VND methodology does not change the basic principles of the corresponding predecessor. Therefore, the setting of parameters from various studies in the literature still applies.

The verification of the methods utilizes a library of testing problems with a variable number of dimensions. This library was created by modifying the well-known libraries for multi-objective optimization (namely: DTLZ, LZ, UF, and ZDT libraries). Therefore, the convergence properties of such testing problems are ensured. The methodology is well described and is easily applicable to any scalable multi-objective problem.

All the methods and testing problems are included in the FOPS optimization framework [MM3, MM4]. The development of the framework is an important part of this thesis, although its development began before my doctoral studies. The reason for the creation of FOPS is that there did not exist any framework where optimization methods with a variable number of dimensions could be implemented. The use of a framework is essential if various and numerous comparative studies are to be composed, executed, and visualized.

FOPS is a unique tool for the optimization of all kinds. It has already been exploited in various papers [MM7, MM8, MM9, MM10, MM11, MM12, MM13, MM14, MM15]. The last chapter of this thesis presents several real-life applications published in [MM5, MM1, MM2]. All of them were carried out in the FOPS framework. This demonstrates the versatility of the FOPS. Moreover, most of the applications are problems with a variable number of dimensions from the field of electrical engineering.

The first VND application is the Optimal placement of transmitters. It was published in [MM3] This problem is a perfect demonstration of the VND problem that can not be tackled with a non-VND algorithm without considerable limitations. Either the number of transmitters is defined a priori, or the decision space is sampled so the transmitters at predefined positions can be enabled or disabled. The sampling of the decision space is shown in the next application – the linear antenna array problem.

The linear antenna array problem presents a synthesis of dipole array where the side-lobe level and the number of active dipoles are optimized. The problem shows two different representations of the problem – representation with a variable number of dimensions tackled by the VND-GDE3 algorithm and representation with a fixed number of dimensions using a uniform-grid. Uniform-grid representation is tackled by the standard GDE3 algorithm. It is shown that better values of Side-lobe Level with fewer antennas used are achieved with the VND-GDE3 algorithm. The problems was published in [MM6] and [MM12].

Another application is the synthesis of digital circuits. In this application, the 3-input, 6-input, and 11-input multiplexers were synthesized by the VND-GDE3 algorithm. The number of product terms in the SOP expression was arbitrary. Therefore, the digital circuits were synthesized without the use of Karnaugh maps or other time-consuming methods.

The next-to-last application is the automated image thresholding. The standard, exhaustive approach is compared to the evolutionary approach in the first part. Afterward, the optimization algorithm with a variable number of dimensions is used to segment the testing images by multiple thresholds, and the last part utilizes the multiple threshold approach in license-plate recognition.

Finally, the clustering problem is solved by a multi-objective evolutionary algorithm with a variable number of dimensions. The evolutionary approach eliminates the main disadvantage of the widely used K-means clustering method. Several clustering benchmark datasets were tackled by two standard clustering methods and one exploiting the VND-GDE3 algorithm. It was shown that for most clustering datasets, the VND-GDE3 approach was the most successful method.

References

[1] V. Pareto, *The mind and society*, vol. 1. Jonathan Cape, London, 1935.

[2] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine learning*, vol. 3, no. 2, pp. 95–99, 1988.

[3] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[5] S. Kukkonen and J. Lampinen, "GDE3: The third evolution step of generalized differential evolution," in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 1, pp. 443–450, IEEE, 2005.

[6] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

[7] K. Deb, *Multi-objective optimization using evolutionary algorithms*, vol. 16. John Wiley & Sons, 2001.

[8] D. A. Van Veldhuizen and G. B. Lamont, "Evolutionary computation and convergence to a Pareto front," in *Late breaking papers at the genetic programming 1998 conference*, pp. 221–228, Citeseer, 1998.

[9] E. Zitzler, *Evolutionary algorithms for multiobjective optimization: Methods and applications*, vol. 63. Citeseer, 1999.

[10] L. J. Fogel, A. J. Owens, and M. J. Walsh, *Artificial intelligence through simulated evolution.* John Wiley & Sons, 1966.

[11] D. E. Goldberg, B. Korb, K. Deb, *et al.*, "Messy genetic algorithms: Motivation, analysis, and first results," *Complex systems*, vol. 3, no. 5, pp. 493–530, 1989.

[12] B. Rekiek, P. De Lit, F. Pellichero, T. L'Eglise, P. Fouda, E. Falkenauer, and A. Delchambre, "A multiple objective grouping genetic algorithm for assembly line design," *Journal of intelligent manufacturing*, vol. 12, no. 5-6, pp. 467–485, 2001.

[13] J. Handl and J. Knowles, "Evolutionary multiobjective clustering," in *International Conference on Parallel Problem Solving from Nature*, pp. 1081–1091, Springer, 2004.

[14] 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*, pp. 876–883, ACM, 2007.

[15] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[16] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.

[17] 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*, vol. 31, no. 4, pp. 635–672, 2005.

[18] 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)*, vol. 40, no. 2, pp. 298–319, 2009.

[19] H. Li and K. Deb, "Challenges for evolutionary multiobjective optimization algorithms for solving variable-length problems," in *Evolutionary Computation (CEC), 2017 IEEE Congress on*, pp. 2217–2224, IEEE, 2017.

[20] P. Kadlec and V. Šeděnka, "Particle swarm optimization for problems with variable number of dimensions," *Engineering Optimization*, vol. 50, no. 3, pp. 382–399, 2017.

[21] S. J. Orfanidis, "Electromagnetic waves and antennas," 2002.

[22] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic imaging*, vol. 13, no. 1, pp. 146–166, 2004.

[23] M. Kamel and A. Zhao, "Extraction of binary character/graphics images from grayscale document images," *CVGIP: Graphical Models and Image Processing*, vol. 55, no. 3, pp. 203–217, 1993.

[24] W. Oh and B. Lindquist, "Image thresholding by indicator kriging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 7, pp. 590–602, 1999.

[25] K. Omar, A. S. Zaini, M. Petrou, and M. Khalid, "Determining adaptive thresholds for image segmentation for a license plate recognition system,"

[26] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[27] P. Plashkin, *Car parked to contemplate the night views of sky full of stars.* unsplash.com, October 2019. [Online; accessed November 19, 2020].

[28] J. Parker and P. Federl, "An approach to license plate recognition," 1996.

[29] E. O'Donnell, *Grey taxi during nighttime.* unsplash.com, May 2019. [Online; accessed November 19, 2020].

Author's Bibliography

[MM1] M. Marek and P. Kadlec, "Discretization of decision variables in optimization algorithms," in *Proceedings of the 24th Conference STUDENT EEICT 2018*, pp. 320–324, april 2018.

[MM2] M. Marek and P. Kadlec, "Using a tolerance-based surrogate method for computer resources saving in optimization," *Radioengineering*, vol. 28, no. 1, pp. 9–18, 2019.

[MM3] 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*, vol. 30, no. 9, p. e22335, 2020.

[MM4] M. Marek, P. Kadlec, and M. Cupal, "FOPS Documentation." http://antennatoolbox.com/download_file?file=FOPS_documentation.pdf, 2017. [Online; accessed 24-November-2020].

[MM5] M. Marek, Z. Raida, and P. Kadlec, "Synthesis of electromagnetic equivalents of composite sheets by multi-objective optimization of anisotropic band-stop filters," in *2018 12th European Conference on Antennas and Propagation (EuCAP 2018) (EuCAP 2018)*, (London, United Kingdom (Great Britain)), apr 2018.

[MM6] M. Marek and P. Kadlec, "Another evolution of generalized differential evolution," *Engineering Optimization*, vol. 53, to be published, accepted on 6th November 2020.

[MM7] D. Warmowska, M. Marek, and Z. Raida, "Matlab-based multi-objective optimization of broadband circularly polarized antennas," in *Proceedings of LAPC 2017*, pp. 1–5, october 2017.

[MM8] P. Kadlec, V. Šeděnka, M. Štumpf, and M. Marek, "Solution of an inverse scattering problem using optimization with a variable number of dimensions," in *2017 International Conference on Electromagnetics in Advanced Applications (ICEAA)*, pp. 976–979, IEEE, 2017.

[MM9] P. Kadlec, M. Marek, M. Štumpf, and V. Šeděnka, "PCB decoupling optimization with variable number of capacitors," *IEEE Transactions on Electromagnetic Compatibility*, 2018.

[MM10] 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 *2018 International Symposium on Electromagnetic Compatibility (EMC EUROPE)*, pp. 585–589, IEEE, 2018.

[MM11] P. Kadlec, M. Marek, and M. Štumpf, "Fast lightning stroke localization in the time domain," in *Proceedings of the 2019 International Symposium on Electromagnetic Compatibility (EMC Europe 2019)*, no. 1, (IEEE), pp. 54–58, IEEE, september 2019.

[MM12] P. Kadlec, M. 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 *2020 30th International Conference Radioelektronika (RADIOELEKTRONIKA)*, pp. 1–6, IEEE, 2020.

[MM13] P. Kadlec, M. Marek, and M. Štumpf, "Lightning stroke localization–a time-domain approach based on evolutionary optimization," *IEEE Transactions on Electromagnetic Compatibility*, 2020.

[MM14] P. Kadlec, M. Marek, and M. Štumpf, "Thin-film screen time-domain shielding effectiveness: Multi-objective optimization of the testing pulse," in *2020 International Symposium on Electromagnetic Compatibility-EMC EUROPE*, pp. 1–5, IEEE, 2020.

[MM15] M. Čapek, P. Hazdra, V. Adler, P. Kadlec, V. Šeděnka, M. Marek, M. Mašek, V. Losenický, M. Štrambach, M. Mazánek, and J. Rýmus, "AToM: A versatile MATLAB tool for antenna synthesis," in *2018 12th European Conference on Antennas and Propagation (EuCAP 2018) (EuCAP 2018)*, (London, United Kingdom (Great Britain)), apr 2018.

Abstract

This dissertation thesis deals with multi-objective evolutionary optimization algorithms with a variable number of dimensions. Such an algorithm enables us to solve optimization tasks that are otherwise solved only by assuming unnatural simplifications. The research of the optimization algorithms with a variable number of dimensions required the development of a new optimization framework. This framework contains, apart from various optimization methods including two novel multi-objective algorithms for a variable number of dimensions – VND-GDE3 and VND-MOPSO, a library of various benchmark problems. A set of multi-objective benchmark problems with a variable number of dimensions is a part of the library designed to assess and verify the novel methods with a variable number of dimensions. Novel methods are exploited on several miscellaneous real-life optimization problems in the final chapter of this thesis.

Abstrakt

Tato dizertační práce pojednává o více-kriteriálních optimalizačních algoritmech s proměnným počtem dimenzí. Takový algoritmus umožňuje řešit optimalizační úlohy, které jsou jinak řešitelné jen s použitím nepřirozených zjednodušení. Výzkum optimalizačních method s proměnnou dimenzí si vyžádal vytvoření nového optimalizačního frameworku, který obsahuje vedle zmíněných vícekriteriálních metod s proměnnou dimenzí – VND-GDE3 a VND-MOPSO – i další optimalizační metody různých tříd. Optimalizační framework obsahuje také knihovnu rozličných testovacích problémů. Mezi nimi je také sada více-kriteriálních testovacích problémů s proměnnou dimenzí, které byly navrženy pro nastavení a ověření nových metod s proměnnou dimenzí. Nové metody jsou dále použity k optimalizaci několika různorodých optimalizačních úloh z reálného světa.