Critical Comparison of Multi-objective Optimization Methods: Genetic Algorithms versus Swarm Intelligence

Vladimír ŠEDĚNKA, Zbyněk RAIDA

Dept. of Radio Electronics, Brno University of Technology, Purkyňova 118, 612 00 Brno, Czech Republic

xseden01@stud.feec.vutbr.cz, raida@feec.vutbr.cz

Abstract. *The paper deals with efficiency comparison of two global evolutionary optimization methods implemented in MATLAB. Attention is turned to an elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) and a novel multi-objective Particle Swarm Optimization (PSO).*

The performance of optimizers is compared on three different test functions and on a cavity resonator synthesis. The microwave resonator is modeled using the Finite Element Method (FEM). The hit rate and the quality of the Pareto front distribution are classified.

Keywords

Multi-objective optimization, binary genetic algorithm, particle swarm optimization, Pareto front, finite element method.

1. Introduction

When designing microwave components, analytical design formulae are not available usually, and therefore, numerical methods have to be used. Without deeper knowledge of the dependency of component parameters on state variables, the manual design becomes difficult. In order to find the optimal solution automatically and effectively, a proper optimization method has to be used.

According to the number of requested objectives, optimizations are divided into single-objective ones and multi-objective ones. In case of the single-objective optimization, there is usually a single optimal solution. In case of the multi-objective optimization (MOOP), some objectives may be conflicting: when approaching one criterion, other criteria are receding. Then, a single optimal solution can be hardly determined without knowledge of the optimized problem.

Merging all objectives into the single-objective function [1] is one of potential solutions. However, the preferences of each objective have to be fixed by proper setting of weighting coefficients of partial criteria in the aggregating objective function. Further, the functional values of the partial objective functions have to be normalized, which is quite problematic without knowledge about the profile of the optimization problem in the objective space.

Alternatively, a set of solutions can be selected as a result. The resultant set of solutions is called the Pareto optimal set (front). At the end of the optimization process, the designer can select a single solution from the Pareto set considering preferences in the objectives.

Deb [2, Ch. 2, p. 28] describes the Pareto front using the term of the domination (and the non-domination). The solution x_1 is dominated, if there is a solution x_2 meeting:

- The solution x_2 is not worse than the solution x_1 in all the objectives;
- The solution x_2 is strictly better in one objective at least.

The set of the solutions, which are not dominated by any other solution, is called the Pareto optimal set.

Several multi-objective evolutionary algorithms (MOEA) were developed and compared [2], [3]–[5]. In [9], authors compare almost 30 variants of the multi-objective particle swarm optimization and try to summarize basic concepts used there.

This paper compares a novel multi-objective Particle Swarm Optimization (PSO) with a slightly modified Non-dominated Sorting Genetic Algorithm (NSGA-II).

In Section 2, optimization methods investigated in this paper are briefly described. In Section 3, efficiency of investigated optimization methods is tested using three different testing functions and a cavity resonator synthesis problem. Section 4 concludes the paper.

2. Optimization Methods

In this section, a binary coded genetic algorithm and particle swarm optimization are briefly described. Emphasis is placed on differences between single-objective and multi-objective optimizations.

2.1 Binary Coded Genetic Algorithm

The genetic algorithm was proposed by Holland [6]. The algorithm consists in an evolution of binary coded chromosomes. Each set of state variables is encoded in one chromosome, the set of the chromosomes forms a genera-

tion. In order to select best individuals as potential parents of a new generation, a classification has to be performed.

The principle of the classification of the exploited Non-dominated Sorting Genetic Algorithm (NSGA-II) [7], [2, Ch. 6, p. 245] is depicted in Fig. 1.

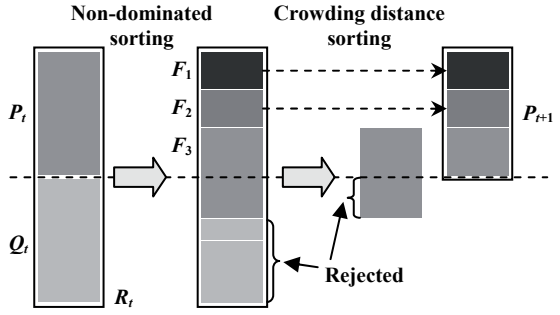

Fig. 1. The NSGA-II procedure [2, Ch. 6, p. 246].

In order to ensure elitism, both the parent population P_t and the offspring population Q_t are sorted.

A coarse classification is given by the non-dominated sorting, which produces multiple Pareto sets indexed F_1, F_2, … F_N. The first Pareto set (the best one) is formed by non-dominated members of both populations P_t and Q_t. Remaining (dominated) members constitute a starting group for the second Pareto set determination, etc. The index of the set forms the coarse criterion.

Within the Pareto set, we cannot determine which solution is better from the fitness viewpoint. However, we can classify the density of solutions surrounding a particular solution by a *crowding distance vector* [2]. This classification is fine and ensures a better distribution of the optimal solutions at the same time.

The procedure always guarantees that the worst distributed chromosome from the first front obtains a better classification than the best chromosome from the second front, etc. New parents are selected by the tournament selection. Afterwards, the cross-over and the mutation are performed as usual.

A two-point crossover with the crossover probability 80 % was used in the test. The probability of the mutation was set to 20 %. Each decision space variable was encoded into 10 bits. Due to the two-dimensional decision space, chromosomes were 20 bits long.

Thanks to the discretized decision space, there is a nonzero chance that chromosomes can repeat. Thus, before each fitness values computation, the chromosome is compared with the previous ones. In case of identity, fitness values are copied instead of repeating the already performed computation.

Using 20 generations with the population size of 30 individuals, the maximum number of fitness evaluations per iteration is 600.

2.2 Particle Swarm Optimization

The particle swarm optimization was introduced in 1995 [8]. In the single-objective variant, each state variable forms one dimension of the decision space, in which the particles move. Each particle stores own personal best solution in a *personal best (*pbest*)* vector, the best solution of the swarm is stored in a *global best (*gbest*)* vector. The velocity vector of the particles is continuously updated according to these vectors and an inertial weight.

The applied multi-objective PSO code was developed on the basis of a single-objective one.

In NSGA-II, a single optimization run is needed in each iteration. In case of a multi-objective PSO, three optimization runs are executed in each iteration. Fortunately, the optimization runs share results mutually which decreases CPU-time demands.

In the first run, the population is divided into multiple subswarms according to the number of objectives (a subswarm per an objective). Each subswarm follows the optimal solution considering only the assigned objective. Along its way, the subswarm can reveal *gbest* values of all the objectives and can store them in the *gbest storage*, which is shared within the whole swarm (*fully connected neighborhood topology* [9]). The *gbest storage* system is depicted in Fig. 2. All single-objective optima f_1^{opt} ... f_k^{opt} are found at the end of the first run.

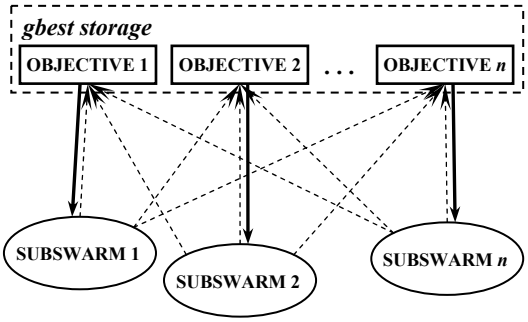

Fig. 2. The *gbest storage* system.

In the second step, all the objective function values are normalized and an aggregating function f_c is created

$$f_c^n = \sum_{k}^{k_{\max}} \frac{f_k^n - f_{k\min}}{f_{k\max} - f_{k\min}} \tag{1}$$

where f_k^n is the n-th fitness value of the k-th objective. The parameter $k_{\max}$ denotes the dimension of the objective space. Now, only $k_{\max}$ particles with initial positions f_1^{opt} … f_k^{opt} will search for a minimum of the aggregating function (in the second run). Due to the smaller number of particles, the second step takes much less time than the first one. The optimization is accelerated thanks to the results of the previous run.

In the third step, Pareto optimal set is found. A poorer half of the set (from the aggregating objective point of

view) form the initial set of the particles for the last PSO run, which minimizes the aggregating objective function f_c again and makes the Pareto set denser.

In order to prevent particles to leave the feasible solution space, some positions and velocity vectors are assigned fixedly. If the number of assigned particles exceeds the value 2^S (S is the dimension of the decision space), 2^S of them are placed into the corners of the decision space. The velocity vector is set to preserve edging to the center of the feasible area. Alternative remaining particles stay in random positions with random velocity vectors.

If the number of assigned particles exceeds the value $2^{(S+1)}$, then 2^S of them are placed into the corners, 2^S of them into the center of the edges and rest is again randomly distributed. Fig. 3 implies the situation in the two-dimensional decision space ($S = 2$).

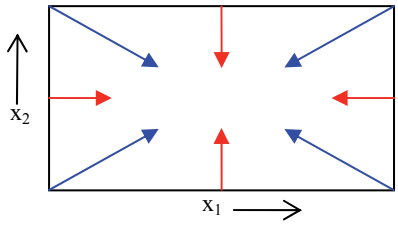

Fig. 3. Improvement of the initial position and the movement vector ("blue" for $p \geq 2^S$, "blue + red" for $p \geq 2^{(S+1)}$).

Compared with NSGA-II, PSO has no mechanism to minimize unnecessary computations. The total number of computations is given by the optimization parameters in the first run, especially. For three test functions discussed in the next section, we used $2^{(S+1)} + 2$ particles and 15 cycles. For the resonator synthesis, 30 particles and 10 cycles were used. The number of cycles in the second run was 20 for the test functions and 10 for the resonator synthesis.

All three runs used invisible walls [11], time step was $\Delta t = 0.4$ s, $c_1 = 1.49$, $c_2 = 1.49$ and inertial weights decreased from 0.9 to 0.5.

In the first run, the whole decision space has to be explored hastily. Thus, the maximal initial velocity for the first run is set to

$$v_{s,\max} = \pm 0.001 \times range_s \tag{2}$$

where s is the current dimension of the decision space and $range_s$ denotes the allowed value range in that dimension. The sign ± indicates two potential directions.

The remaining runs improve the shape and the density of the Pareto front only. Thus, the maximal velocity $v_{s,\max}$ is ten times smaller. The external archive [9], mutation operator [9] or any crowding operator were not used in case of this optimization.

3. Results

Efficiency of the above described methods was tested using three test functions. Then, the methods were applied to the microwave resonator optimization.

Using test functions, each iteration can be executed relatively quickly. Therefore, the optimization can be run several times. We examined the performance over 1 000 independent runs per each test function and each optimization method. In case of the resonator synthesis, only 100 runs were used due to the higher CPU-time demands for the numerical analysis of the resonating structure.

Results are described by various classifiers. The parameter *PF* denotes the number of resulting Pareto front members. The total number of fitness function calculations is given by the classifier *FFC*. The final hit rate *HR* is computed according to

$$HR = \frac{PF}{FFC} \cdot 100[\%]. \tag{3}$$

The higher hit rate indicates that the less of time-consuming fitness calculations were used to find the Pareto front position. The hit rates have to be compared within each solution only. In order to create a universal hit rate quantifier, the relation between the size of the ideal Pareto area and the size of the feasible one should be considered.

The equation for determining the quality of the front distribution was suggested by Deb [2, Ch. 8, p. 328]:

$$\Delta = \frac{\sum_{m=1}^{M} d_m^e + \sum_{i=1}^{|Q|} \left| d_i - d_{\text{avg}} \right|}{\sum_{m=1}^{M} d_m^e + |Q| d_{\text{avg}}} \tag{4}$$

where M denotes the number of the extreme solutions and d^e is the minimal Euclidean distance between the m-th extreme solution and the rest Q solutions of the front, d_i is the consecutive Euclidean distance between i-th and $(i + 1)$-th solutions and d_{avg} is the average value of the Euclidean distances.

The above equation assumes knowledge of the positions of extreme solutions. Since this information is not available in case of the resonator synthesis, sums of the distances d^e are neglected. The Euclidean distances are replaced by *crowding distances* [2, Ch. 6, p. 248]. The quality coefficient is then

$$C_q = \frac{\sum_{i=1}^{\text{PF}} \left| c_i - c_{\text{avg}} \right|}{\text{PF}\, c_{\text{avg}}} \tag{5}$$

where c_i is the *crowding distance* and c_{avg} is the average value of crowding distances. The *crowding distance* is given by the fitness values of the previous solution and the

following one in the front. At the boundaries, we have no more points to evaluate d_i conventionally, so the distance between the previous solution and the current one is doubled to compensate this problem. Lower deviations from the average distance denote a better distribution. The quality coefficient C_q becomes zero in the ideal case.

Each set of results is described by a table containing the minimal values, the average values and the maximal values of above mentioned parameters acquired from all optimization runs. Any row does not need to be the representation of a realizable optimization run. Actually, this situation is highly improbable thanks to equation (3).

3.1 Test Functions

Test functions are very useful, because their functional value is given by a computationally modest relation. Therefore, the feasible area in both the decision space and the objective space can be easily determined. Moreover, the accuracy of approaching the optimum by investigated methods can be evaluated.

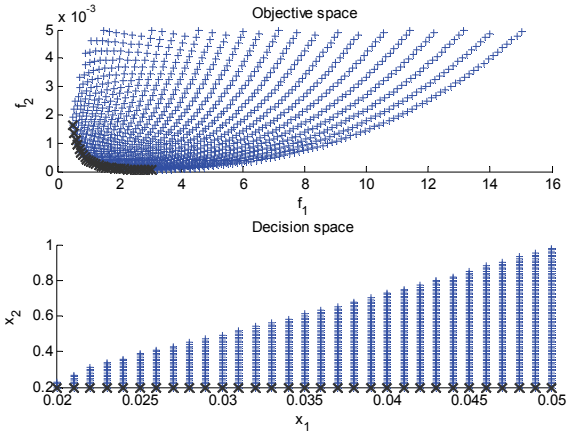

Fig. 4. Feasible area in the decision space and the objective space for the first test function. Blue: feasible solutions, black: Pareto optimal set.

The first test function is described by a cantilever design problem [2]:

$$f_1 = \frac{7800\pi x_1^2}{4} x_2 , \tag{6a}$$

$$f_2 = \frac{64 x_2^3}{621\cdot 10^6 \pi x_1^4} \tag{6b}$$

where

$$x_1 \in \langle 0.02, 0.05 \rangle , \tag{6c}$$

$$x_2 \in \langle 0.2, 1 \rangle , \tag{6d}$$

$$\frac{32 x_2}{\pi x_1^3} \le 3\cdot 10^5 , \tag{6e}$$

$$\frac{64 x_2^3}{621\cdot 10^3 \pi x_1^4} \le 5 . \tag{6f}$$

We try to minimize both f_1 and f_2. The feasible area in the decision space and the objective space is depicted in Fig. 4. Examples of an appropriate optimization are shown in Fig. 5 and 6. Both Pareto fronts in Fig. 5 and 6 are relatively well distributed.

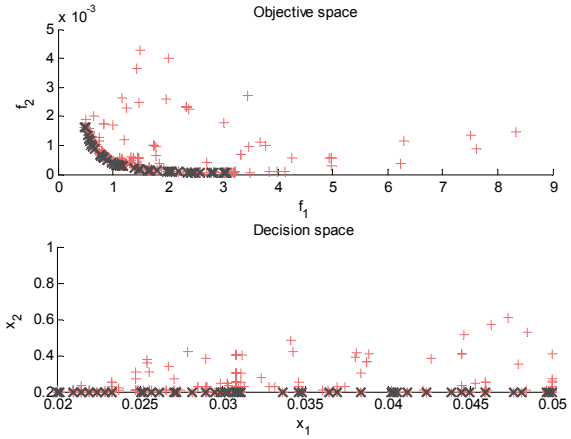

Fig. 5. Test function 1: NSGA-II solution example. Red: findings of the objective function value, black: Pareto optimal set.

NSGA-II	*PF* [-]	*FFC* [-]	*HR* [%]	*Q* [-]
Min	3.0	65.0	3.030	0.362
Avg	50.5	207.2	24.366	0.739
Max	139.0	369.0	49.438	1.412

Tab. 1. Test function 1: statistics over 1 000 runs of NSGA-II.

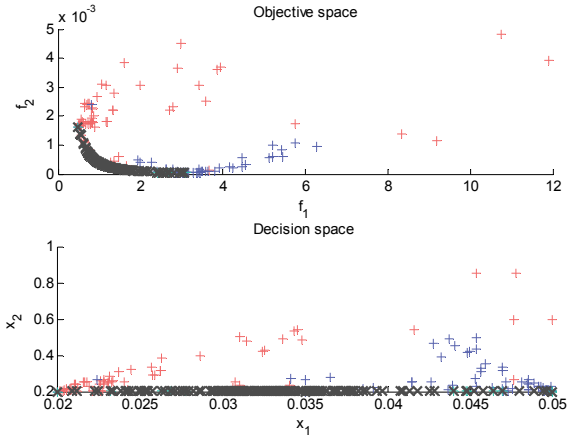

Fig. 6. Test function 1: PSO solution example. Red: findings of the min(f_1), blue: findings of the min(f_2), green: findings of the min(f_c), black: Pareto optimal set.

PSO	*PF* [-]	*FFC* [-]	*HR* [%]	*Q* [-]
Min	27.0	140.0	15.698	0.384
Avg	57.9	188.8	30.697	0.747
Max	102.0	244.0	47.739	1.201

Tab. 2. Test function 1: statistics over 1 000 runs of the PSO.

Obviously, PSO exhibits a better hit rate but the quality of the distribution is slightly worse compared to NSGA-II. Nevertheless, the chances of PSO to bring particles closer to the extreme solutions neglected in equation (5) are better. This property would cause a better performance of the PSO in all researched parameters.

The diversity of the parameter *FFC* is given by the limited decision space. Outside the feasible area, no fitness value is calculated in order to speed-up the optimization.

The second test function is defined in the 3D objective space. We try to minimize all partial objective functions [15]:

$$f_1 = x_1^2 + (x_2 - 1)^2, \tag{7a}$$

$$f_2 = x_1^2 + (x_2 + 1)^2 + 1, \tag{7b}$$

$$f_3 = (x_1 - 1)^2 + x_2^2 + 2 \tag{7c}$$

where

$$x_1 \in \langle -2, 2 \rangle, \tag{7d}$$

$$x_2 \in \langle -2, 2 \rangle. \tag{7e}$$

Fig. 7 depicts the feasible area in the decision space and the objective space. The extreme solutions at (0, –1), (1, 0) and (0, 1) can be observed.

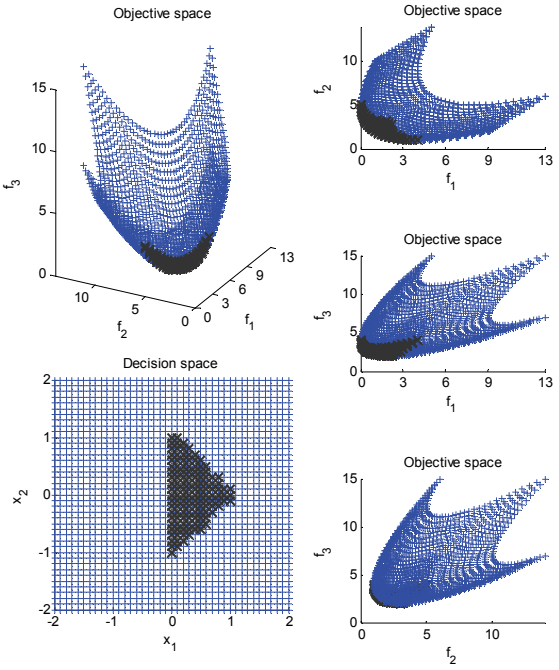

Fig. 7. Feasible area in the decision space and the objective space for the second test function. Blue: feasible solutions, black: Pareto optimal set.

Fig. 8 shows the well distributed front of the NSGA-II. Red points (fitness function calculations) form a crucifix due to only two decision space variables. There is higher possibility that only one variable will change.

Looking at the PSO decision space (Fig. 9), we can observe colored single-objective runs again. Due to the smaller front density, more non-optimal results were evaluated as Pareto front elements (points at $x_1 < 0$).

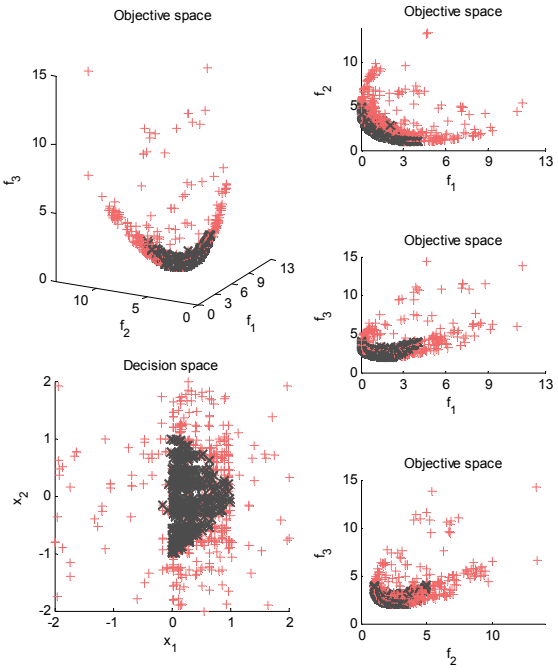

Fig. 8. Test function 2: NSGA-II solution example. Red: findings of the objective function value, black: Pareto optimal set.

NSGA-II	*PF* [-]	*FFC* [-]	*HR* [%]	*Q* [-]
Min	30.0	90.0	17.630	0.435
Avg	186.0	381.4	48.748	0.617
Max	311.0	500.0	71.253	1.088

Tab. 3. Test function 2: statistics over 1 000 runs of NSGA-II.

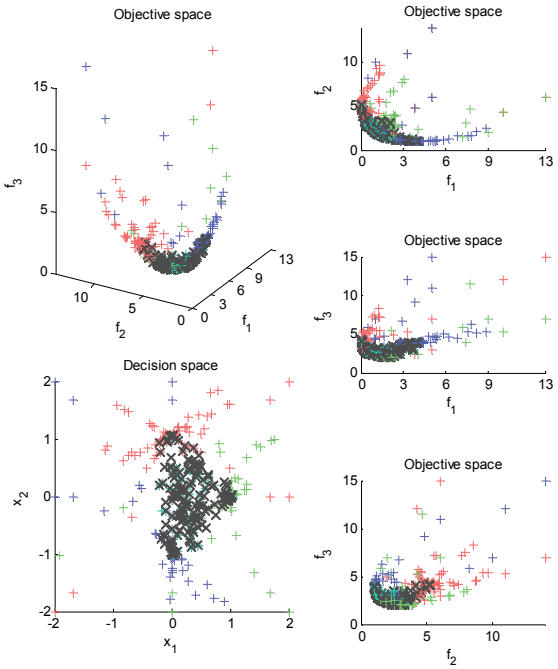

Fig. 9. Test function 2: PSO solution example. Red: findings of the min(f_1), blue: findings of the min(f_2), green: findings of the min(f_c), black: Pareto optimal set.

PSO	*PF* [-]	*FFC* [-]	*HR* [%]	*Q* [-]
Min	120.0	305.0	38.217	0.414
Avg	198.5	355.4	55.645	0.541
Max	282.0	399.0	71.392	0.723

Tab. 4. Test function 2: statistics over 1 000 runs of the PSO.

In case of the second test function, PSO exceeds NSGA-II both in *HR* and *Q*.

The third test function is defined as follows [15]:

$$f_1 = 0{,}5\left(x_1^2 + x_2^2\right) + \sin\left(x_1^2 + x_2^2\right), \tag{8a}$$

$$f_2 = \frac{\left(3x_1 - 2x_2 + 4\right)^2}{8} + \frac{\left(x_1 - x_2 + 1\right)^2}{27} + 15, \tag{8b}$$

$$f_3 = \frac{1}{\left(x_1^2 + x_2^2 + 1\right)} - 1.1\exp\left(-x_1^2 - x_2^2\right) \tag{8c}$$

where

$$x_1 \in \langle -3{,}3 \rangle, \tag{8d}$$

$$x_2 \in \langle -3{,}3 \rangle. \tag{8e}$$

The feasible area and the ideal Pareto optimal set are depicted in Fig. 10. Due to the discontinuous Pareto front all the optimal solutions are very difficult to be found, especially two sections in the area $x_1 \in$ <–3, –1.5> and $x_2 \in$ <–3, 0>.

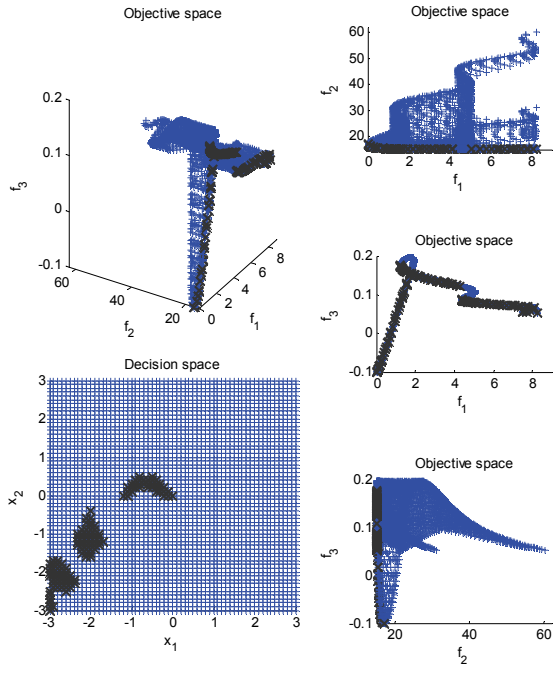

Fig. 10. Feasible area in the decision space and the objective space for the third test function. Blue: feasible solutions, black: Pareto optimal set.

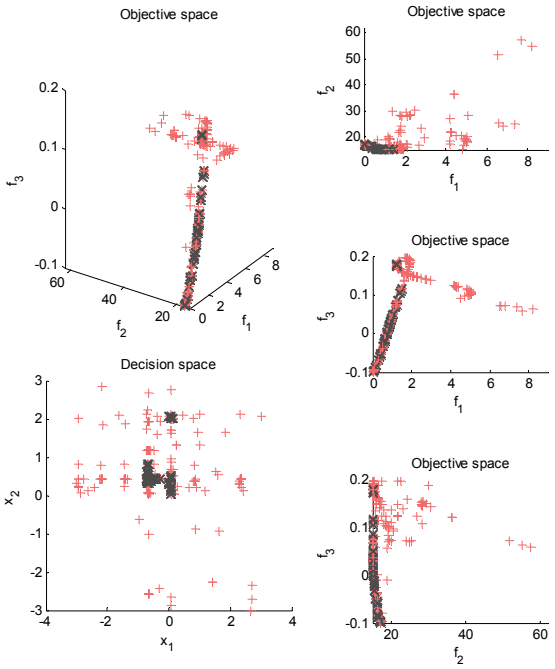

Fig. 11. Test function 3: NSGA-II solution example. Red: findings of the objective function value, black: Pareto optimal set.

NSGA-II	*PF* [-]	*FFC* [-]	*HR* [%]	*Q* [-]
Min	14.0	79.0	7.027	0.444
Avg	63.1	264.4	24.398	0.857
Max	169.0	467.0	64.211	1.394

Tab. 5. Test function 3: statistics over 1 000 runs of NSGA-II.

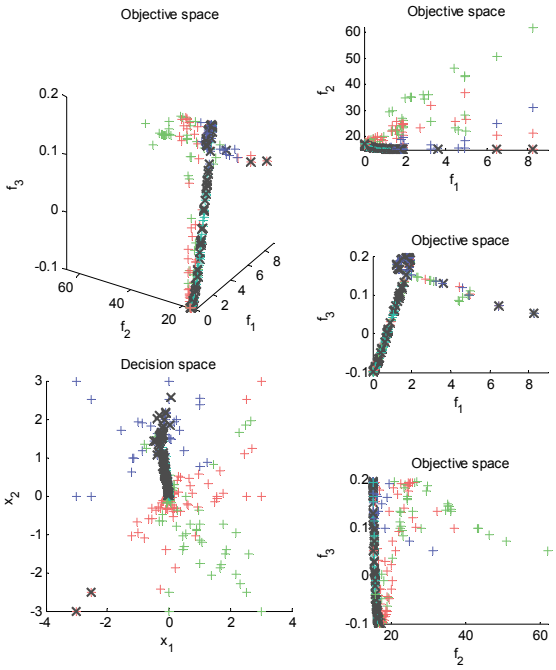

Fig. 12. Test function 3: PSO solution example. Red: findings of the min(f_1), blue: findings of the min(f_2), green: findings of the min(f_c), black: Pareto optimal set.

PSO	*PF* [-]	*FFC* [-]	*HR* [%]	*Q* [-]
Min	17.0	157.0	8.265	0.554
Avg	77.1	278.5	27.351	1.010
Max	153.0	330.0	51.701	1.515

Tab. 6. Test function 3: statistics over 1 000 runs of the PSO.

The PSO exceeds NSGA-II in the hit rate only. Poorer quality of the PSO Pareto front is given by algorithm itself if all minima of the single-objective functions are not found.

The performance of both the optimizers is quite poor in this case (see Fig. 11 and Fig. 12). Pareto front members of two sections in the area $x_1 \in \langle -3, -1.5 \rangle$ and $x_2 \in \langle -3, 0 \rangle$ were not found. Instead, false detection around [0, 2] appears.

3.2 Cavity Resonator Synthesis

Efficiency of both the algorithms is also compared on a conflicting optimization of a cavity resonator. The optimization is aimed to synthesize an air-filled cavity resonator containing perfectly conducting walls. The resonator is longitudinally homogenous and symmetric across both axis of the cross-section. Thanks to the symmetry and longitudinally homogeneity, state variables are created by x and y co-ordinates of the specified number of points and by the depth of the resonator. In the test optimization, three state points (7 state variables) are considered (see Fig. 13).

In order to ensure conflicting objectives, the first cut-off frequency and the total volume of the resonator are observed, and both the parameters are minimized. The cut-off frequency is computed by the finite element method (FEM). Mesh density is given by a *characteristic length* [14], and therefore, the total number of the elements and the CPU-time demands should not vary significantly.

The third objective is related to the total deviation of angles in the cross-section. In order to determine the angle deviation on one edge, the angle between adjacent walls is compared with 90° / 180° (0° denotes the infeasible solution). The total deviation is computed as a sum of the squares of the angle deviation. The right angle implies zero deviation and easier manufacturing, and therefore, we try to get as small total deviation as possible.

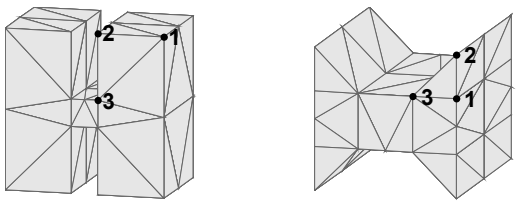

Fig. 13. Example structures 1 (left) and 2 (right).

Both x and y coordinates of three points of the cross-section (see Fig. 13) may vary from 5 mm to 60 mm. The depth of the resonator may vary from 5 mm to 80 mm.

Fig. 13 depicts the geometry of two elements of the Pareto front. Related optimization parameters (decision space variables) and fitness values (objective space) are shown in Tab. 7. We can examine here the frequency / volume dependency and difference in total angle deviation.

	Resonator 1	Resonator 2
x_1 [m]	0.0556	0.0260
y_1 [m]	0.0556	0.0101
x_2 [m]	0.0094	0.0260
y_2 [m]	0.0556	0.0260
x_3 [m]	0.0094	0.0101
y_3 [m]	0.0094	0.0101
depth [m]	0.0740	0.0730
frequency [GHz]	2.6690	3.7660
volume [m^3]	0.0008	0.0001
angle deviation [$(°)^2$]	0	16 200

Tab. 7. Geometry and fitness parameters of the solutions in Fig. 13.

Fig. 14 and Fig. 15 show selected results just in the objective space (since the decision space is 7-dimensional). Related statistics are shown in Tab. 8 and Tab. 9. We added here the total CPU time required for the optimization as a parameter t.

NSGA-II	*PF* [-]	*FFC* [-]	*HR* [%]	*Q* [-]	*t* [s]
Min	49.0	253.0	10.891	0.513	131.1
Avg	86.3	463.5	18.706	0.804	271.6
Max	143.0	514.0	34.387	1.087	564.3

Tab. 8. Optimization of the resonator: statistics over 100 runs of the NSGA-II.

PSO	*PF* [-]	*FFC* [-]	*HR* [%]	*Q* [-]	*t* [s]
Min	56.0	332.0	16.868	0.473	115.5
Avg	137.4	392.6	34.747	0.675	189.6
Max	210.0	461.0	48.724	1.139	297.0

Tab. 9. Optimization of the resonator: statistics over 100 runs of the PSO.

Due to the relatively different number of fitness computations, the only comparable results were selected to exclude the possibility that relative good hit ratio in PSO is caused by a smaller *FFC* number. Thus, values with *FFC* from 410 to 435 were compared only (see Tab. 10).

NSGA-II	*PF* [-]	*FFC* [-]	*HR* [%]	*Q* [-]	*t* [s]
Min	49.0	411.0	11.264	0.756	202.4
Avg	79.5	424.6	18.794	0.932	253.0
Max	114.0	435.0	27.737	1.087	297.9

PSO	*PF* [-]	*FFC* [-]	*HR* [%]	*Q* [-]	*t* [s]
Min	124.0	412.0	30.097	0.480	141.2
Avg	166.6	420.8	39.577	0.635	202.5
Max	210.0	435.0	48.724	0.820	278.4

Tab. 10. Optimization of the resonator: statistics over 10 runs of the NSGA-II and 29 runs of the PSO.

The *FFC* parameter is now nearly equal. Looking at the results, the hit ratio of the PSO is now even higher, so the influence of the smaller *FFC* of the PSO can be excluded.

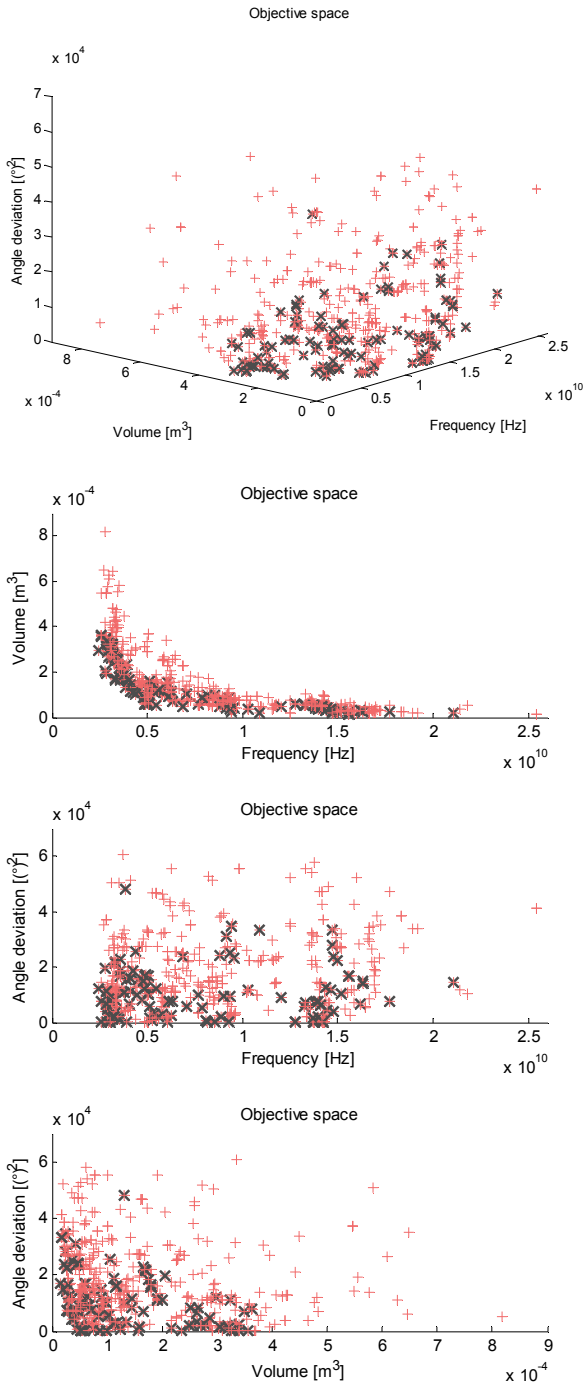

Fig. 14. Optimization of the resonator: NSGA-II solution example. 20 generations, 30 individuals, P_{cross} = 80 %, P_{mut} = 20 %.Red: findings of the objective function value, black: Pareto optimal set.

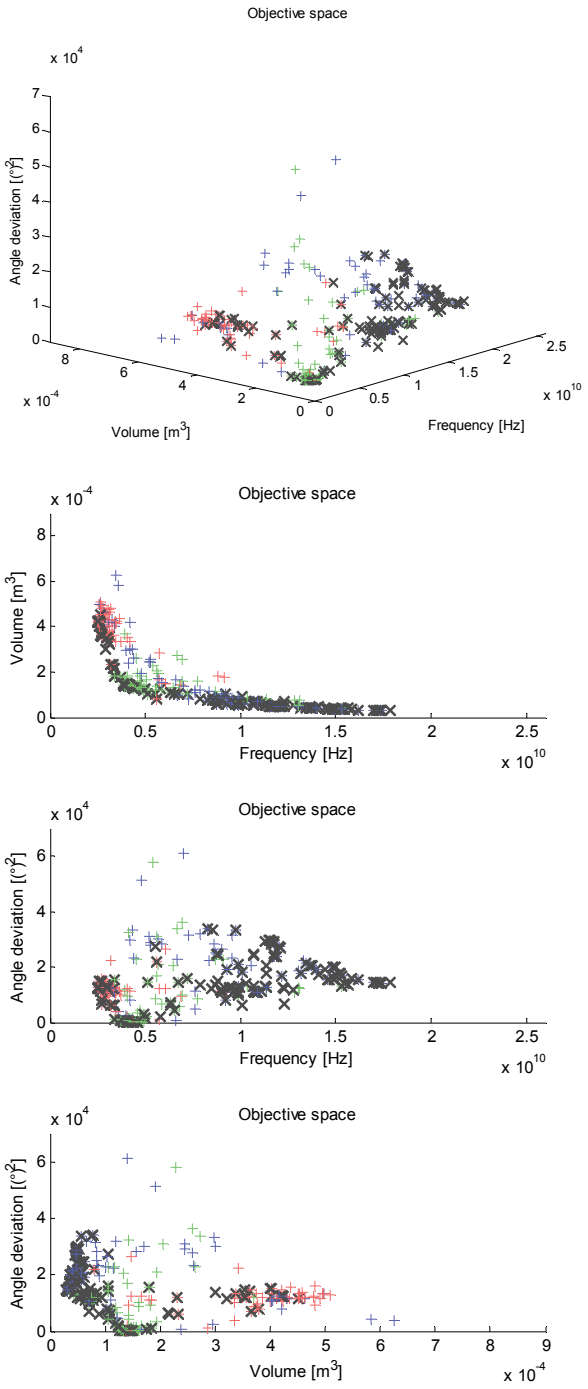

Fig. 15. Optimization of the resonator: PSO solution example. 1st run: 30 particles, 10 cycles; 2nd run: 3 particles, 10 cycles; 3rd run: 41 particles, 3 cycles.

4. Conclusions

Although the PSO is not a native multi-objective one, the novel multi-objective PSO outperforms NSGA-II almost in all the cases. Nevertheless, this may be caused by a not quite optimal setting of the optimization parameters.

The PSO shows better ability to find the extreme solutions. Customizing the optimization variables, the density of the final Pareto set can be chosen. However, their inaccurate combination could produce worse distribution of the solutions.

Thanks to the non-dominated sorting, NSGA-II needs less optimization parameters to set. It is an advantage in term of the service, but the preference for the proximity to the extreme solutions and the density of the Pareto set cannot be driven separately.

Acknowledgements

The research described in the paper has been financially supported by the Czech Science Foundation under grants no. 102/07/0688 and 102/08/H018, and by the research program MSM 0021630513. The research is a part of the COST actions IC 0603 and IC 0803, which are financially supported by the Czech Ministry of Education under grants OC08027 and OC09016.

References

[1] LUKEŠ, Z., RAIDA, Z. Multi-objective optimization of wire antennas: genetic algorithms versus particle swarm optimization. *Radioengineering*, 2005, vol. 14, no. 4, p. 91-97.

[2] DEB, K. *Multi-Objective Optimization using Evolutionary Algorithms*. Chichester: John Wiley and Sons, 2001.

[3] BANDYOPADHYAY, S., SAHA, S., MAULIK, U., DEB, K. A simulated annealing-based multiobjective optimization algorithm: AMOSA. *IEEE Transactions on Evolutionary Computation*, 2008, vol. 12, no. 3, p. 269-283.

[4] ZITZLER, E., DEB, K., THIELE, L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 2000, vol. 8, no. 2, p. 173-195.

[5] GROŞAN, C., DUMITRESCU, D. A comparison of multiobjective evolutionary algorithms. *Acta Universitatis Apulensis*, 2002, vol. 4.

[6] HOLLAND, J. H. Outline for a logical theory of adaptive systems. *J. Assoc. Comput. Mach.* 1962, vol. 3, p. 297–314.

[7] DEB, K., PRATAP, A., AGARWAL, S., MEYARIVAN, T. A fast and elitist multi-objective genetic algorithm-NSGA-II. *KanGAL Report.* 2000, no. 2000001. [Online] Available at: http://www.iitk.ac.in/kangal/papers/tech2000001.ps.gz.

[8] KENNEDY, J., EBERHART, R. C. Particle swarm optimization. *IEEE Conference on Neural Networks.* Perth (Australia), 1995, p. 1942-1948.

[9] REYES-SIERRA, M., COELLO COELLO, C. A. Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International Journal of Computational Intelligence Research*, 2006, vol. 2, no. 3, p. 287-308. [Online] Available at: http://www.softcomputing.net/ijcir/vol2-issu3-paper5.pdf

[10] JOHNSON, J. M., RAHMAT-SAMII, Y. Genetic algorithms in engineering electromagnetics. *IEEE Antennas and Propagation Magazine*, 1997, vol. 39, no. 4, p. 7-21.

[11] ROBINSON, J., RAHMAT-SAMII, Y. Particle swarm optimization in electromagnetics. *IEEE Transactions on Antennas and Propagation*, 2004, vol. 52, no. 2, p. 397-407.

[12] SILVESTER, P. P., FERRARI, R. L. *Finite Elements for Electrical Engineers*. 3rd ed. Cambridge: Cambridge University Press, 1996.

[13] CHEN, L. F., ONG, C. K., NEO, C. P., VARADAN, V. V., VARADAN, V. K. *Microwave Electronics: Measurement and Materials Characterization*. John Wiley and Sons, 2004.

[14] GEUZAINE, C., REMACLE, J. F. *Gmsh Reference Manual*. 233 pages. [Online] Cited 2010-01-05. Available at: http://www.geuz.org/gmsh/doc/texinfo/gmsh.pdf.

[15] JIN, Y., OKABE, T., SENDHOFF, B. Solving three-objective optimization problems using evolutionary dynamic weighted aggregation: Results and analysis. [Online] Available at: http://www.soft-computing.de/moo3D.pdf.

About Authors ...

Vladimír ŠEDĚNKA (*1983 Ostrava, Czech Republic) graduated at the Faculty of Electrical Engineering and Communication (FEEC), Brno University of Technology (BUT), in 2009. Since 2009, he has been a post-graduate at the Dept. of Radio Electronics, BUT.

Zbyněk RAIDA received Ing. (M.Sc.) and Dr. (Ph.D.) degrees from the Brno University of Technology in 1991 and 1994, respectively. Since 1993, he has been with the Dept. of Radio Electronics, FEEC BUT as assistant professor (1993 to 1998), associate professor (1999 to 2003), and professor (since 2004). In 1997, he spent 6 months at the Laboratoire de Hyperfrequences, Universite Catholique de Louvain, Belgium as independent researcher. Z. Raida has authored or coauthored more than 100 papers in scientific journals and conference proceedings. His research is focused on numerical modeling and optimization of electromagnetic structures, application of neural networks to modeling and design of microwave structures, and on adaptive antennas. Prof. Raida is a member of the IEEE Microwave Theory and Techniques Society and since 2003 Senior Member of IEEE. He chaired the MTT/AP/ED joint section of the Czech-Slovak chapter of IEEE (2001- 2003).