

COMPUTATIONAL AERODYNAMIC OPTIMIZATION OF A LOW-SPEED WING

MARTIN LAHUTA^a, ZDENĚK PÁTEK^{a,b,*}, ANDRÁS SZÖLLÖS^a

^a Výzkumný a zkušební letecký ústav, a,s, Beranových 130, 199 05 Praha – Letňany, Czech Republic

^b Faculty of Mechanical Engineering, Czech Technical University, Technická 4, 166 07 Praha 6, Czech Republic

* corresponding author: patek@vzlu.cz

ABSTRACT. An optimization method consisting of two evolutionary optimization algorithms and a solver using nonlinear aerodynamics is applied to the design of a low-speed wing. The geometric parameterization of the wing uses standard geometric quantities commonly used for describing the wing geometry. The method seems to provide good reliable results with low computer capacity requirements.

KEYWORDS: aerodynamic optimization; evolutionary algorithm; low-speed wing.

1. INTRODUCTION

With the use of new manufacturing technologies, especially technologies with composite materials, conventional and well-proven wing shapes (usually trapezoids or combination of trapezoids) can be abandoned, and complex three-dimensional geometrical shapes can be applied, even for light and general aviation aircraft. A wing can now be designed more closely according to aerodynamic requirements. The shapes of the wings of advanced sailplanes can serve as demonstrative examples [1, 2].

New optimization techniques connected to more precise, more accurate and faster aerodynamic solvers are leading to wings optimized according to several aerodynamic criteria in relatively broad ranges of geometric parameters.

Evolutionary algorithm methods suitable for multi-criterion optimization have been developed over a considerable period of time. Microevolutionary algorithms provide welcome computational-time savings together with an acceptably substantial scan of multi-dimensional design space. Methods are now also available for computing the aerodynamic characteristics of wings that provide sufficiently accurate results for geometrically complex low-speed wings, and that are at the same time very quick. *The necessary parameterization of the geometric shape of the wing uses conventional geometric parameters widely used in the technical description of a wing, rather than mathematical parameters.*

2. OPTIMIZATION CRITERIA AND CONSTRAINTS

In the aerodynamic optimization of a wing, some form of minimization of the wing drag or maximization of the lift-to-drag ratio is usually required as the main optimization criterion. This is not sufficient for a real aircraft wing, where many other constraints are simultaneously applied. Typical aerodynamic constraints that can be used are: the minimum achieved value

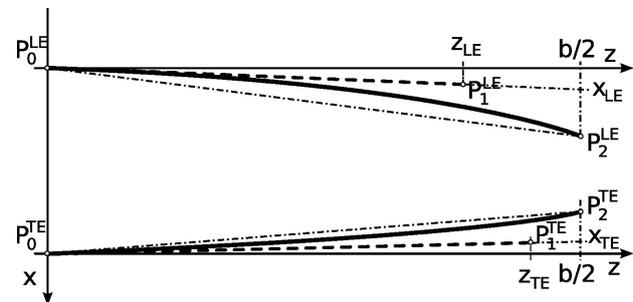


FIGURE 1. Parameterization of the wing planform.

of the maximum lift coefficient $C_{L\max}$, the maximum absolute value of the wing pitching moment coefficient C_m , and the limit of the position of the point where flow detachment begins.

Typical geometric constraints that can be used are: wing area, wing span, wing aspect ratio, wing taper ratio, wing twist, wing dihedral, wing swept angle. The constraints can be given either as limiting values (for example maximum acceptable twist) or as directly required values. A prescribed kind of geometric shape can also be required, for example a wing composed from two trapezoids on the wing half-span.

As the Reynolds numbers along the wing span are involved in the aerodynamic computations, dimensional setting of the wing and the flight speed is preferred.

3. GEOMETRIC DESCRIPTION OF THE WING

The geometry of the wing was restricted by the following two prescribed requirements: area of the wing S_w , and semispan $b/2$.

The leading edge (LE) and the trailing edge (TE) of the wing were represented by a quadratic Bezier curve [3] (see Figure 1) given by the equation

$$\vec{r}(t) = (1-t)^2 P_0 + 2t(1-t) P_1 + t^2 P_2, \quad \vec{r}(t) = [x, z], \quad (1)$$

where P_0 is a point at the root, P_2 is a point at the tip and P_1 is the so-called control point of the curve lying inside the triangle $P_0, P_2, [P_2^x, P_0^y]$, which guarantees that the curve is always convex or concave and lying inside this triangle. The equation (1) of the curve can be expressed with $P_0^{LE,TE}$ shifted to the origin of the coordinate system, without loss of generality. The parametric coordinates of these shifted points are then:

$$\begin{aligned} P_0^{LE,TE} &= [0, 0], \\ P_1^{LE,TE} &= [z^{LE,TE}, dx^{LE,TE}, x^{LE,TE}, z^{LE,TE}], \\ P_2^{LE,TE} &= [b/2, dx^{LE,TE}], \end{aligned}$$

where

$$\begin{aligned} dx^{LE,TE} &\in \langle x_{\min}^{LE,TE}, x_{\max}^{LE,TE} \rangle, \\ x^{LE,TE} &\in \langle 0, 1 \rangle, \\ z^{LE,TE} &\in \langle 0, b/2 \rangle. \end{aligned}$$

These curves with $P_0^{LE,TE}$ at the origin of the coordinate system are then shifted against each other along the x -axis so that the surface of the wing is S_w . Alternatively, the configuration is rejected if the prescribed surface area cannot be achieved for the given parameters. The wing configuration is also rejected when there is non-decreasing local length of the chord with an increasing z -coordinate. The local chords of the wing are also twisted with *consequent* linear dependency along the wing span, where the local twist angle is given as $\epsilon(z) = \frac{z}{b/2} \epsilon^{TIP}$ and $\epsilon^{TIP} \in \langle x_{\min}^{LE,TE} \rangle$ is a parameter describing the twist of the tip of the wing. Overall, there are seven parameters: $x^{LE,TE}$, $z^{LE,TE}$, $dx^{LE,TE}$, ϵ^{TIP} . This parameterization includes a wide range of realistic wing geometries, including trapezoidal geometries.

4. AERODYNAMIC DESCRIPTION OF THE WING

4.1. AIRFOIL SECTIONS

It is necessary to create a database of the airfoils used on the wing in advance, because the aerodynamic characteristics of the airfoils used on the wing and their positions along the wingspan are used as inputs for wing optimization. The airfoils are described by their usual conventional aerodynamic characteristics, i.e., by the lift curves, the drag curves and the moment curves at different Reynolds numbers. It is very useful to have these airfoil characteristics for the angles of attack overrunning the angle of attack of stall by at least two degrees.

4.2. WING

The wing aerodynamic performance is described using the commonly used coefficients of wing lift, wing drag and wing pitching moment, the lift distribution along the span, the lift coefficient distribution along the span

and the point along the span where flow detachment begins.

The point where flow detachment begins is found as the point where, increasing the wing angle of attack, the local lift coefficient reaches the maximum lift coefficient available at the local airfoil section.

5. METHOD FOR COMPUTING THE AERODYNAMIC CHARACTERISTICS OF THE WING

NLWing2 software was used for calculating the aerodynamic characteristics of the wing. NLWing2 is an implementation of the nonlinear lifting line method developed at VZLU [4]. This method allows the use of 2D viscous or non-viscous airfoil analysis (calculated e.g., by Xfoil software or provided by wind tunnel testing) for efficient computation of the nonlinear aerodynamic properties of 3D wing configurations. It employs 2D section data to build a 3D potential vortex model of the flow. It uses a robust Euler-Newton method to track the change in the flow vorticity quantities as the angle of attack progresses. NLWing2 runs under the GNU Octave system [5]. The implementation is very effective, and is a few orders faster than alternative CFD methods.

6. OPTIMIZATION METHOD

6.1. EVOLUTIONARY OPTIMIZATION

The powerful computing facilities that have emerged in recent years have led to many practical optimizations in various areas of science and engineering. As a consequence, various heterogeneous evolutionary algorithms have appeared [6] to [12]. However, according to their functionality, they can be described by the following basic scheme: 1. Initialize the population by randomly generated individuals 2. Evaluate 3. repeat until the criterion for stopping of the optimization is met a. choose individuals for reproduction b. apply variation operators to the selected parents c. evaluate new design candidates d. choose the best design candidates for the next generation end of the loop Genetic algorithms have received much attention in the evolutionary community. Among them, the most popular evolutionary algorithms are perhaps NSGA2, by K. Deb et al. [13], and SPEA2, by E. Zitzler et al. [14]. NSGA2 is widely considered to be a reference algorithm.

The particular achievement of NSGA2 is that it is very easy to use – it is almost parameter-less, and is fast due to its elitist non-dominated sorting with the crowding distance as a diversity metric. SPEA2 has an excellent diversity mechanism accompanied by the so called Pareto-archive, preserving promising non-dominated individuals during the evolution. A detailed explanation of these two approaches can be found in [13, 14].

6.2. DIFFERENTIAL EVOLUTION

A very promising new concept was proposed relatively recently by R. Storn and K. Price [15], and is known as the differential evolutionary (DE) algorithm. Differential evolution is a simple yet powerful means for creating promising new design candidates via combining a parent solution with other population members. Various implementations of the basic idea were soon developed and, interestingly, some of them outperformed both NSGA2 and SPEA2 on benchmark problems.

Now let us quote the original formulation of the authors. Differential evolution utilizes NP D -dimensional design vectors

$$x_i^G, \quad i = 1, \dots, NP$$

as a population for each generation G . DE creates new parameter vectors by adding the weighted difference between two population vectors to a third vector. Let this operation be called mutation. The components of the mutated vector are then mixed with the parameters of another predetermined vector, the target vector, to yield the so-called trial vector. This parameter mixing can be termed as a crossover. If the trial vector has better fitness than the target vector, it replaces the target vector. This last operation is called selection. Each individual in the population has to serve once as the target vector so that NP competitions take place in each generation.

The basic DE-operators are as follows:

Mutation. For each target vector x_i^G , a mutant vector is created according to

$$v_i^{G+1} = x_{r_1}^G + F(x_{r_2}^G - x_{r_3}^G)$$

with mutually different random indices $r_{1,2,3} \in 1, 2, \dots, NP$ and $F > 0$. The randomly chosen integers $r_{1,2,3}$ are also chosen different from the running index i , so that NP must be greater than or equal to four to allow for this condition, and F controls the amplification of the differential variation $x_{r_2}^G - x_{r_3}^G$. It is real, greater than zero and less than 2. In the original proposal it is constant for the whole evolution, while our implementation also allows it to be either constant for the current running index of the design vector or varying with each component of the actual individual.

Crossover. The diversity of the population can be further increased by a crossover. The so-called trial vector is therefore formed:

$$u_1^{G+1} = (u_{1i}^{G+1}, u_{2i}^{G+1}, \dots, u_{Di}^{G+1}),$$

where

$$u_{ji}^{G+1} = \begin{cases} v_{ji}^{G+1} & \text{if randb}(j) \leq CR \text{ or } j = \text{rnbr}(i), \\ x_{ji}^{G+1} & \text{otherwise.} \end{cases}$$

Here, $\text{randb}(j)$ is the j -th evaluation of a uniform pseudo-random number generator with the outcome

from the interval $[0, 1]$. CR is the crossover constant from $[0, 1]$ which has to be determined by the user, and $\text{rnbr}(i)$ is a randomly chosen index belonging to $1, 2, \dots, D$, which ensures that u_{ji}^{G+1} gets at least one parameter from v_{ji}^{G+1} .

In addition, EA1 allows vectors also to be exchanged as a whole, and not just their components.

Selection. In the original proposal, the trial vector is compared to the target vector using the greedy criterion. If vector u_i^{G+1} yields a smaller cost function value than x_i^G then x_i^{G+1} is set to u_i^{G+1} . Otherwise the old value x_i^G is retained. This scheme was tailored for solving merely single-objective optimization problems. Modification for the multi-objective case was straightforward. EA1 is a Pareto-archive oriented algorithm, therefore every trial vector is compared to the archive. If any archive-member dominates it, it is discarded. Otherwise it is included into the archive.

The pseudocode of differential evolution is as follows:

- (1.) Initialize the population of P design candidates by randomly generated individuals.
- (2.) Evaluate.
- (3.) Repeat until the criterion for stopping the optimization is met: For each design vector P_i ($i = 1, \dots, NP$) from P :
 - (a) create candidate C from parent P_i ;
 - (b) evaluate the candidate;
 - (c) if the candidate is better than any of the archive parents, it becomes a new member and replaces it; otherwise the candidate is discarded.
- (4.) Randomly enumerate the individuals in P .

6.3. THE EA1 MULTI-OBJECTIVE EVOLUTIONARY OPTIMIZER

We, too, have proposed an evolutionary optimizer, which is called EA1 (Evolution Algorithms One), using the crowding distance as a diversity metric, together with the Pareto-archive for preserving non-dominated members of the evolution. There are ideas borrowed from the two previously-mentioned algorithms, supplemented by range-adaptation, population-statistics management and elitist-random reinitialization.

Range adaptation is a highly efficient technique for directing the evolution towards interesting regions of the design and criterial space, by controlling the population statistics. The essence of range adaptation is as follows: check the mean values and the standard deviations of the decision variables every k generations. Let us denote the old (after n generations) mean-value vector and its standard deviation as v_{old} and σ_{old} , and the new (after $n + k$ generations) mean-value vector and its standard deviation as v_{new} and σ_{new} , respectively. If the i -th component of v_{new} differs from the i -th component of v_{old} by more than a certain value, defined as $ap * (\sigma_{\text{old}})_i$ (ap is a user-supplied constant

– let us call it an adaptation parameter, and $(\sigma_{\text{old}})_i$ is the i -th component of the old standard deviation-vector), then generate the new reinitialized population in the interval $((v_{\text{new}})_i - (\Delta v_{\text{new}})_i, (v_{\text{new}})_i + (\Delta v_{\text{new}})_i)$, where $\Delta v_{\text{new}} = v_{\text{new}} - v_{\text{old}}$. In other words, let one third of $(\Delta v_{\text{new}})_i$ be the new i -th component of the standard deviation-vector. One third is chosen, because the population is assumed to conform to the Gaussian probability distribution, and 99.7 percent of it is contained within the interval $(-3\sigma, +3\sigma)$. Moreover, if the new standard deviation $(\sigma_{\text{new}})_i$ is less than a certain prescribed scalar value σ_{min} , then $(\sigma_{\text{new}})_i$ is equal to σ_{min} , to ensure that the evolution will not be attracted toward any local optimum. As can be seen from the above, this strategy strives to keep the evolution in a permanently “excited” state by continually perturbing it through forced modification of the population statistics. The values for σ usually go from 1.0 to 1.5, and σ_{min} is mostly equal to 0.2. Therefore, these *are included as* two additional parameters of the evolution. The number of individuals *included in* the population statistics ranges from the upper two to the whole population [7].

Elitist-random reinitialization consists of putting some (usually two) Pareto-archive members into the reinitialized population, which is subsequently selected randomly for mating. It usually utilizes micro-populations going down to four. However, ten-member populations are most commonly used in the evolution. At the beginning, elitist-random reinitialization was a multi-objective micro-genetic algorithm with range-adaptation and elitist-random based reinitialization. Details about our concept are given in [16]. However, it was necessary to broaden its scope by exploiting the new excellent features of differential evolution. After a redesign and a great deal of experimentation with redefining the basic DE-operators to conform with multi-objective microevolution, the result has evolved into a multi-objective optimizer equipped with both genetic and differential evolution operators. The user can now choose between these according to a simple switch that indicates the type of evolution to be used during the optimization. A detailed description will be given in a paper which is currently under preparation.

The evolution starts with a population generated by Latin Hypercube Sampling. After the evaluation, the non-dominated individuals are put into the Pareto-archive to update it. EA1 actually uses two populations: the first population contains only four to ten individuals to produce new information via applying evolutionary operators, and the second population contains the Pareto archive. Each new individual is evaluated by comparing it to the archive. If it is not dominated by any archive member, it is accepted as a new member, otherwise it is rejected. The pseudocode is as follows:

(1.) Initialize the population of P design candidates by randomly generated individuals through Latin Hypercube Sampling.

- (2.) Evaluate and update the archive.
- (3.) Repeat until the criterion for stopping the optimization is met: For each design vector:
- (a) apply evolutionary operators (either genetic or differential);
 - (b) evaluate and update the archive: every n generations:
 - i. update the population statistics,
 - ii. adapt the search range,
 - iii. reinitialize the population by elitist-random reinitialization.

7. EXAMPLE OF OPTIMIZING A WING BY EA1

The method was used for optimizing a low-speed wing for the following problem.

(1.) Geometrical constraints:

- NACA 0012 symmetric airfoil along the whole span;
- area of the wing $S_w = 16 \text{ m}^2$;
- span of the wing $b_w = 18 \text{ m}$;
- the local chord $c(z)$ must decrease monotonically towards the tip of the wing;
- the leading and trailing edge created by the curve are defined by (1);
- the twist is only linear from the wing root to the wing tip, maximum twist at the wing tip 3° in relation to the root;
- longitudinal position of the $\frac{1}{4}$ point of the MAC $x_{0.25 \text{ MAC}} = x_{0.25 \text{ c root}} \pm 0.05 c_{\text{MAC}}$.

(2.) Aerodynamic constraints:

- $C_{L \text{ max}} \geq 1.25$ for $Re = 1.5 \cdot 10^6$;
- point of the beginning of the flow separation not farther than $0.65 b_w / 2$ from the wing root to the wing tip;
- Reynolds number is $Re = 1.5 \cdot 10^6$ (related to c_{MAC}).

(3.) Optimization criteria: In the wing polar curve diagram (relation $C_L = C_L(C_D)$), to minimize the area S (do not confuse with S_w wing area) constrained by:

- value $C_D = 0$ from the left side;
- wing polar curve from the right side;
- value $C_L = 0.1$ from bottom;
- value $C_L = 1.0$ from the top.

(4.) The differential evolution in EA1 has been set as follows:

- population size: 4, 10, 30;
- length of the design vectors: 7;

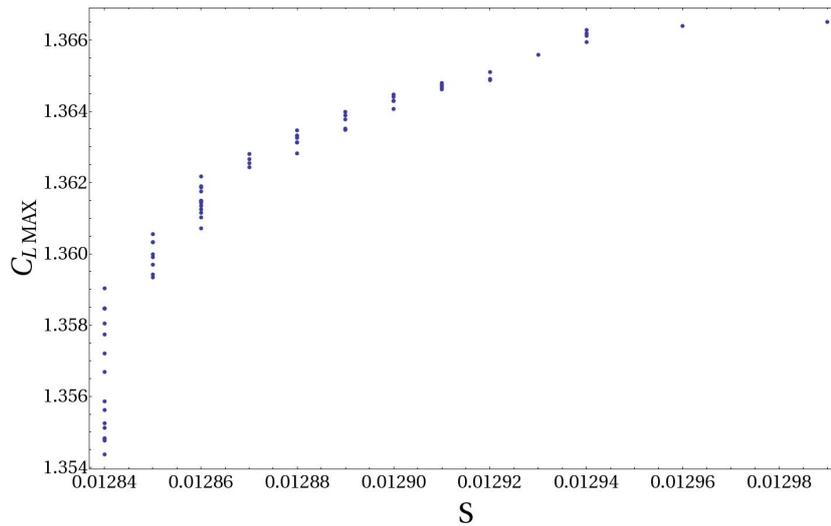


FIGURE 2. Pareto front, $C_{L\max}$ as a function of the optimization criterion S .

Wing	S	$C_{L\max}$	Twist [°]	Separation [$b_w/2$]
1	0.01287	1.363	0.49	0.139
2	0.01290	1.364	0.27	0.249
3	0.01292	1.365	0.27	0.249

TABLE 1. Three examples of optimized wings.

- the fraction of the population included in the calculation of the population statistics: npop 1 for population sizes 4 and 10, and npop 3 for population size 30;
 - Pareto-archive size: 300;
- (5.) The population was reinitialized in each generation:
- adaptation parameter ap : 1.5, σ_{\min} : 0.4;
 - mutation parameter F : 10^{-10} ;
 - crossover parameter CR : 0.1.

7.1. RESULTS

It is seen that the Pareto front (Figure 2) offers many wings which offer a low value of the drag optimization criteria and meet the prescribed constraints. Three examples of the optimized wing planform geometry and twist are included in Table 1 and in Figure 3. The aerodynamic characteristics of these wings are given in Table 1 and in Figure 4. The optimized shapes are generally similar to the planforms of the wings of advanced sailplanes and motorgliders, so these preliminary tests indicate that the method can be used in practical applications.

It can also be seen that the Pareto front (Figure 2) is very flat, as concerns the drag optimization criterion. This means that relatively high deviation from the optimum geometric extreme does not substantially affect the wing aerodynamic performance and the other constraints. This is a favourable result in

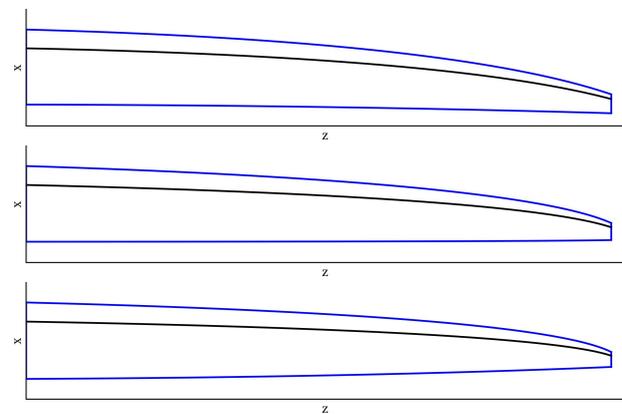


FIGURE 3. Wings 1, 2 and 3.

the sense that non-aerodynamic requirements (e.g., structural and manufacturing requirements) can be broadly applied in the wing design in this case.

8. CONCLUSIONS

A method for optimizing the shape of a low-speed wing has been developed and tested. The method combines the new EA1 evolutionary optimization algorithm with the proven NLwing2 aerodynamic solver, using nonlinear aerodynamic data as inputs. The initial tests seem to prove that the method is applicable for the preliminary design of the wing.

LIST OF SYMBOLS

- b wing span
- C_D drag coefficient
- C_L lift coefficient
- $C_{L\max}$ maximum lift coefficient
- C_m pitching moment coefficient
- c local wing chord
- c_{MAC} wing mean aerodynamic chord
- S value of the optimization criterion
- S_w wing area

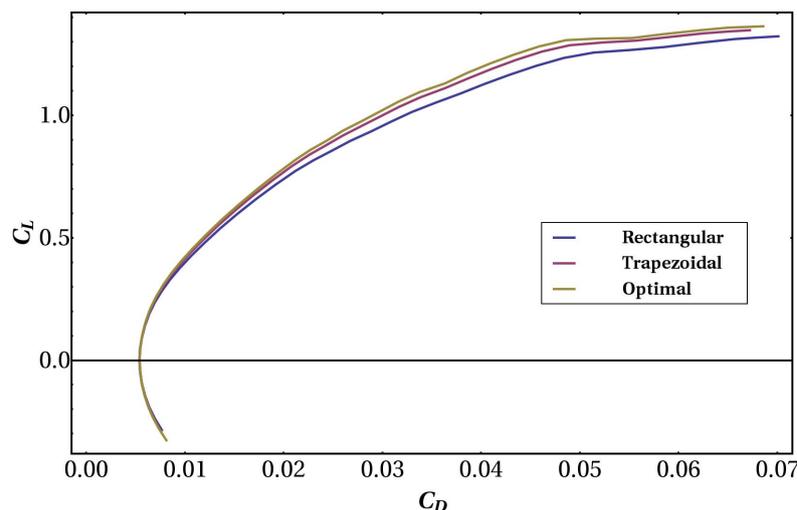


FIGURE 4. Polar diagrams of the three optimized wings and of the rectangular and trapezoidal wing.

α angle of attack
 ϵ twist angle

REFERENCES

- [1] Thomas, F.: *Fundamentals of Sailplane Design*. College Park Press, College Park, Maryland 1999.
- [2] Pajno, V.: *Sailplane Design*. Macchione Editore, Varese 2006.
- [3] Farin, G., Hoschek, J., Myung-Soo, K.: *Handbook of Computer Aided Geometric Design*, North Holland 2002. ISBN: 978-0-444-51104-1. DOI:10.1016/B978-044451104-1/50000-9
- [4] Hájek, J.: *NLWing2 software for viscous analysis of slender wings*, VZLU report R-4722, Výzkumný a zkušební letecký ústav, a.s., Praha 2010
- [5] Octave community: GNU Octave 3.6.4, 2014, <http://www.gnu.org/software/octave/> [2014-12-01].
- [6] Coello Coello, C. A.: *An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design*. Ph.D. thesis, Department of Computer Science, Tulane University, New Orleans 1996
- [7] Coello Coello, C. A., Lamont, G. B. (eds.): *Applications of Multi-Objective Evolutionary Algorithms*. World Scientific, Singapore 2004. DOI:10.1142/5712
- [8] Coello Coello, C. A. Lamont, G. B., Van Veldhuizen, D. A. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Second edn. Kluwer Academic Publishers, New York 2007. DOI:10.1007/978-0-387-36797-2
- [9] Corne, D. W., Knowles, J. D., Oates, M. J.: The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization. In: Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J. J., Schwefel, H.P. (eds.): *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pp. 839–848. *Lecture Notes in Computer Science No. 1917*, Springer, Paris 2000. DOI:10.1007/3-540-45356-3_82
- [10] Deb, K.: *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester 2001.
- [11] Fonseca, C. M., Fleming, P. J.: *Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization*. In: Forrest, S. (ed.): *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 416–423. University of Illinois at Urbana-Champaign, Morgan Kaufmann Publishers, San Mateo 1993
- [12] Goldberg, D. E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Reading 1989. DOI:10.5860/choice.27-0936
- [13] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6 (2002) (2), pp. 182–197. DOI:10.1109/4235.996017
- [14] Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In: Giannakoglou K, Tsahalis D, Periaux J, Papailou P, Fogarty T (eds.) *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, Athens, Greece, pp. 95–100
- [15] Storn, R., Price, K.: Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization* 11 (1997), pp. 341–359. DOI:10.1023/A:1008202821328
- [16] Szöllös, A., Šmíd, M., Hájek, J., 2009: Aerodynamic optimization via multi-objective micro-genetic algorithm with range adaptation, knowledge-based reinitialization, crowding and ϵ -dominance. *Advances in Engineering Software*, 40 (2009) (6) pp. 419–431. DOI:10.1016/j.advengsoft.2008.07.002