XXIV ENMC
Encontro Nacional de Modelagem Computacional
XII ECTM
Encontro de Ciência e Tecnologia de Materiais

PERFORMANCE EVALUATION OF THREE PENALTY TECHNIQUES COUPLED TO A BAT ALGORITHM TO SOLVE A MULTIDISCIPLINARY DESIGN OPTIMIZATION OF A UAV

Lucas de Landa Couto[1] - lucas.landa@engenharia.ufjf.br
Josué Yoshikazu de Oliveira Saito[2] - josue.saito@engenharia.ufjf.br
Afonso Celso de Castro Lemonge[3] - afonso.lemonge@ufjf.edu.br
[1,2]Universidade Federal de Juiz de Fora, Dep. Engenharia Mecânica - Juiz de Fora, MG, Brazil
[3]Universidade Estadual de Juiz de Fora, Dep. de Mecânica Aplicada e Computacional, MG, Brazil

Abstract.*Evolutionary algorithms are originally designed to solve unconstrained optimization problems. However, most real-world optimization problems, such as those from engineering, present constraints in their formulations. One way to overcome this issue is to consider penalty functions replacing the constrained optimization problems into unconstrained optimization problems. This paper evaluates the bat algorithm's performance coupled to a parameter-less adaptive penalty method (APM) to handle the constraints of a multidisciplinary design optimization concerning a cargo unnamed aerial vehicle (UAV). The APM handles inequality and equality constraints and does not require the knowledge of the constraint as a function of the design variables. The constrained optimization problem consists of finding design variables concerning the wing and stabilizers plant form, and incidences that maximizes the flight score (e.g., the payload). The numerical experiments compare the performance of the original and one variant of APM, emphasizing the advantages of using a parameter-less adaptive penalty method avoiding the task of pre-defined penalty parameters.*

Keywords: *Bat algorithms; Constraint-handling techniques; Adaptive penalty methods; UAV.*

1. INTRODUCTION

UAVs have been increasing in uncountable fields, such as military, agricultural, surveillance, cargo, and people transportation. According to its mission, the design of such air vehicles can be focused and optimized for one or more purposes. Range, autonomy, stealth capabilities, cargo carrying, and shipping are a few examples.

The student competition organized by the Society of Automotive Engineers of Brazil (SAE Brazil, Schuster et al. (2006)) has as its main goal to develop and to instigate the aeronautical engineering knowledge on engineering students from all over the country. On this challenge, for the last few years, the main focus of the aircraft design is the cargo-carrying capabilities. The aeronautical industry's challenges (such as space and fuel limitations) are adapted and incorporated into the competition, bringing an opportunity to solve real-life multidisciplinary engineering problems to the students. For instance, an issue is the maximization of the payload that an aircraft can carry, leading to many challenges in other areas of its design requiring a multidisciplinary task.

Implementing optimization techniques on such a complex problem provides the opportunity to speed the conceptual and preliminary design phases and are applied extensively in the industry. The Federal University of Juiz de Fora team that has taken part in this competition since 2011 has developed its Multidisciplinary Design Optimization (MDO) algorithm (Pereira et al. (2018)). It has been using it in the design of its aircraft for four competitions so far. As a result, the aircraft configuration definition process has been accelerated from weeks of manual analysis to some hours of computing. Every year the challenge proposed by the competition changes, and with it, the MDO has to be altered. This modification can cause drawbacks, mainly when a new constraint has to be added. Depending on the restriction behavior and influence on the airplane, fitting an appropriate penalty function can slow down the project schedule or limit the MDO's potential if not applied properly. Therefore, the main objective of this paper is to investigate the performance evaluation of a bat algorithm coupled to three penalty strategies, such as a static penalty function and two parameter-less adaptive penalty methods (APM), to handle the constraints of a multidisciplinary design optimization concerning a cargo UAV. The APM presented good performance when coupled to other evolutionary algorithms to solve single and multi-objective constrained optimization problems (Lemonge et al. (2015); Carvalho et al. (2017); Vargas et al. (2019)). The APM handles inequality and equality constraints and does not require the knowledge of the constraint as a function of the design variables. The original and one variant of APM (Barbosa and Lemonge (2008)) are coupled independently to a Bat algorithm, and their performances are compared in the numerical experiments. The constrained optimization problem consists of finding design variables concerning the wing and stabilizers plant form and incidences that maximizes the flight score (e.g., the payload).

The rest of this paper is organized as follows: The basic steps of the Bat algorithm is presented in Section 2.. The multidisciplinary design optimization problem discussed in this paper is formulated in Section 3.. In Section 4. is presented the constraint-handling technique adopted in the constrained multidisciplinary design optimization analyzed in this papeer.

2. BAT ALGORITHM

Proposed by Yang in Yang (2010), the BAT algorithm aims to develop a bio-inspired optimization method on the echolocation behavior of bats. The algorithm combines advantageous features of other metaheuristic algorithms. The Bat algorithm can be considered competitive compared with other evolutionary algorithms to solve optimization problems (Khan and Ashok (2012); Talal (2014)).

The bat optimization searches for a solution by two distinct behaviors: the first one approximates attributes of the best solution in the population. On the other hand, the second one causes a small perturbation on this solution. The hunting pattern of bats begins with a low-frequency pulse, and a big amplitude is adopted, meaning a global search. In the local search, a bat emits a high-frequency pulse with a small amplitude to find the exact location of the prey. Throughout the optimization process, the probability of the bat perform the global search (Step 1) decreases as it evolves. In contrast, the probability of a local search (Step 2) increases for a more refined solution. Details of the Bat algorithm can be found in Yang (2010).

3. MULTIDISCIPLINARY DESIGN OPTIMIZATION

In the aeronautical industry, multidisciplinary design optimization often demands the formulation and solution of multi-objective optimization problems. These formulations can consider two or several objective functions, design variables of various types (discrete, integer, continuous, and mixed, for example), in addition to constraints of multiple types. Thus, for example, the problems can be sizing, shape, topological optimization, or even combination.

However, a flight score equation that already incorporates multidisciplinary aspects of the design is formulated as a single-objective optimization problem on the SAE student competition. The MDO incorporates aerodynamics, stability, performance, and structural analysis. In addition, the maximum take-off weight (MTOW), empty weight, and other aircraft characteristics, essential to the initial design definition, are considered in these analyses. The design variables, represented by the vector (**x**), are the wing and stabilizers plant form and incidences, totaling 12 variables. The dimensions used to define the aircraft are shown in Fig.1. Some variables are defined as the absolute value of their dimensions, while others are expressed as relative dimensions so that the aircraft proportions are coherent. The upper and lower bounds for each variable are shown in Table 1.

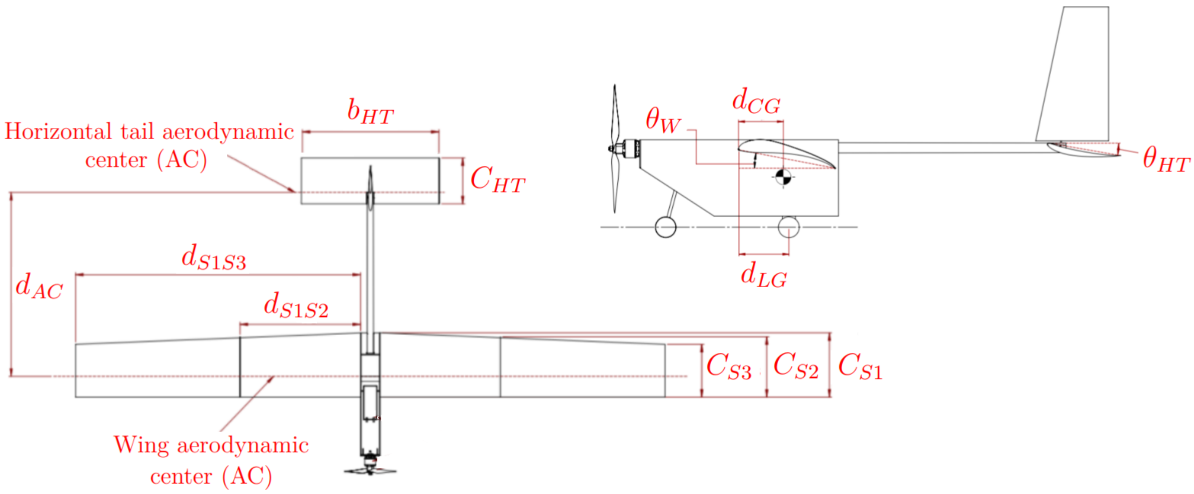

Figure 1- UAV geometry.

Table 1- Upper and lower bounds of the design variables.

Variable	Lower Bound	Upper Bound	Variable	Lower Bound	Upper Bound
θ_{iW} (°)	0	2,5	θ_{iHT} (°)	-2,5	0
C_{S1} (m)	0,22	0,30	d_{AC} (m)	0,5	1,0
d_{S1S3} (m)	1,1	1,4	C_{HT}/C_{S1}	0,6	0,8
C_{S2}/C_{S1}	0,7	1,0	b_{HT}/C_{HT}	2	3
d_{S1S2}/d_{S1S3}	0,4	0,8	d_{CG}/C_{S1}	0,24	0,27
C_{S3}/C_{S2}	0,7	1,0	d_{LG}/C_{S1}	0,45	0,65

The optimization problem is written as:

$$\max \quad FS(\mathbf{x}) = PL(\mathbf{x}) \cdot \left[\frac{0,65}{1 + e^{\left(\frac{1,25 - SE(\mathbf{x})}{0,65} + 0,2\right)}} + 0,4 \right] \tag{1}$$

$$\text{subject to} \quad g_1(\mathbf{x}) = \alpha_{takeoff} - \alpha_{stall} < 0 \tag{2}$$

$$g_2(\mathbf{x}) = Cm_\alpha < 0 \tag{3}$$

$$g_3(\mathbf{x}) = -Cm_0 \leq 0 \tag{4}$$

$$g_4(\mathbf{x}) = Cl_\beta < 0 \tag{5}$$

$$g_5(\mathbf{x}) = -Cn_\beta < 0 \tag{6}$$

$$g_6(\mathbf{x}) = SM - 25 < 0 \tag{7}$$

$$g_7(\mathbf{x}) = -(10 - SM) < 0 \tag{8}$$

$$g_8(\mathbf{x}) = TBV - 36L < 0 \tag{9}$$

where, $FS(\mathbf{x})$ is the flight score equivalent to payload $PL(\mathbf{x})$; α is the aircraft's angle of attack; β is the aircraft's side-slip angle; Cm_α is the pitch coefficient in function of α; Cm_0 is the pitch coefficient for $\alpha = 0$; Cn_β is the Yaw coefficient in function of β; Cl_β is the roll coefficient in function of β; SE is the structural efficiency (Payload/Empty Weight); and TBV is the transportation box volume.

Constraint $g_1(\mathbf{x})$ avoids that the airplane stalls before the takeoff, constraints $g_2(\mathbf{x})$ to $g_7(\mathbf{x})$ are related to the stability and control characteristics of the aircraft, and constraint $g_8(\mathbf{x})$ is imposed by the competition rules that demands that the plane, when disassembled, can fit into a transportation box with maximum volume of 36 liters. A flowchart that explains the MDO workflow is shown in Fig. 2.

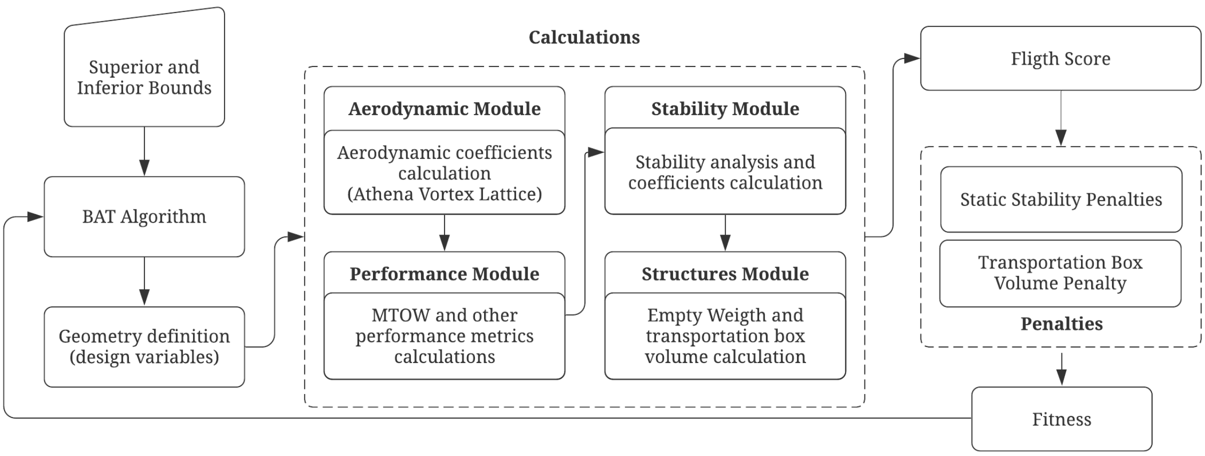

Figure 2- Flowchart of the MDO worflow.

4. CONSTRAINT-HANDLING TECHNIQUE – THE ADAPTIVE PENALTY METHOD (APM)

Most engineering optimization problems have to deal constraints with several degrees of complexity. Population-based metaheuristics are designed to deal with unconstrained optimization problems. When constraints are present in the formulation of the optimization problems, the most immediate way to handle them is by adding penalty functions to the objective function. In this way, a constrained optimization problem is converted into an unconstrained optimization problem and the optimization problem is evaluated by a fitness function denoted by $F(x)$. Penalty functions can be static, dynamic, and adaptive, for example. In the literature, this is an issue of great interest, and there are several strategies for dealing with constraints (Datta and Deb (2014)). Also, penalty functions can be divided in multiplicative or additive. Eqs.10 and 11 describe the Fitness function $F(\mathbf{x})$ for a generic additive and multiplicative technique, respectively.

$$F(\mathbf{x}) = f(\mathbf{x}) + kP(\mathbf{x}) \tag{10}$$

$$F(\mathbf{x}) = kf(\mathbf{x}) \tag{11}$$

where $P(\mathbf{x})$ is the penalty function and k the user-defined penalty coefficient. It's interesting to consider a penalty function $P(x)$ that increases as the constraint's violation values also increase, or, in other words, as more "distant" the solution is from being feasible. The penalty coefficient k determines how severe the penalty will be applied to the infeasible solutions.

This kind of constraint-handling technique is easy to be implemented and can be used in any optimization algorithm since it treats the problem as an unconstrained problem. However, it is too difficult to pre-define the penalty coefficient's values, and it can demand an exhaustive trial-and-error process. An alternative to avoid these trial-and-error processes is using adaptive techniques that do not depend on user intervention to set the penalty parameters.

Examples of adaptive penalty methods can be found in Bean and Hadj-Alouane (1992); Schoenauer and Xanthakis (1993). The technique used in this paper was proposed by Lemonge and Barbosa (2004). The parameter-less adaptive penalty method (APM) handles inequality and equality constraints; it does not require the knowledge of the constraint as a function of the design variables; it does not need any user-defined parameter; and as shown in others studies it is robust. APM can be easily applied to any Evolutionary Algorithm (EA), not only the Genetic Algorithm (GA), for which it was first designed. APM sets penalty coefficients independently for each of the constraints taking into account population information such as the average of the objective function and the level of violation of each constraint. Those penalty factors are updated every generation along with the evolutionary processes. The APM defines the fitness function ($F(x)$) as follows:

$$F(x) = \begin{cases} f(\mathbf{x}) & \text{, if } \mathbf{x} \text{ is feasible} \\ \bar{f}(\mathbf{x}) + \displaystyle\sum_{j=1}^{m} k_j \nu_j(\mathbf{x}) & \text{, otherwise} \end{cases} \tag{12}$$

where m is the total number of constraints of the problem, $v_j(\mathbf{x})$ is the value of violation to the jth constraint of the specific solution x and $\bar{f}(\mathbf{x})$ is defined as

$$\bar{f}(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{, if } f(\mathbf{x}) > \langle f(\mathbf{x}) \rangle \\ \langle f(\mathbf{x}) \rangle & \text{, otherwise} \end{cases} \tag{13}$$

where $\langle f(x) \rangle$ is the average of the fitness function of all individuals of the current population. The penalty parameter k_j, in Eq.12 is defined as

$$k_j = |\langle f(\mathbf{x}) \rangle| \frac{\langle \nu_j(\mathbf{x}) \rangle}{\sum_{l=1}^{m} [\langle \nu_l(\mathbf{x}) \rangle]^2} \tag{14}$$

where $\langle \nu_j(\mathbf{x}) \rangle$ and $\langle \nu_l(\mathbf{x}) \rangle$ are the violation averages for the jth and lth violation, respectively in the current population.

The APM is designed in such a way that the most difficult constraints to be satisfied will present the most severe penalty coefficients. Four new variants of the APM scheme were proposed in Barbosa and Lemonge (2008), each of them with modifications in the updating process of the penalty coefficients throughout the evolution. In this paper, only one variant is considered concerning the APM with monotonic penalty coefficients. This variant does not allow any new penalty coefficient k_j to assume a value smaller than the previous generation, so if $k_j^{new} < k_j^{current}$, then $k_j^{new} = k_j^{current}$.

5. NUMERICAL EXPERIMENTS

Three penalty functions were adopted in the numerical experiments: i) static and multiplicative penalty function setting a value of $k = 1.1$ in Eq. 11, denoted by SPF ; ii) original APM; and iii) monotonic APM. 5 independent runs were performed. For all numerical experiments the population size (η) was set equal to 36, the maximum number of generations equal to 60, the loudness decrease rate (γ) equal to 60, pulse emission increase rate (α) equal to 0,15, and the stop criterion is the maximum number of generations. The computational time of each run was 8 hours executed in a 16GB RAM, i7, Quad Core desktop.

6. RESULTS

Table 2 presents the average, the best, and the worst values of the fitness function at the end of the evolutionary process, according to the curves plotted in Fig. 3. One can observe that the best solution was found setting the original APM with $FS(\mathbf{x}) = -100$. The monotonic APM generated the second best solution with $FS(\mathbf{x}) = -98.37$, followed by the static penalty function with $FS(\mathbf{x}) = -98.27$. Figure 4 is depicted the normalized final design variables, in parallel coordinates, concerning the best solutions. From Fig. 3 to 10, except for Fig. 4, the colours of each one of the curves are denoted by: 1st run (blue), 2nd run (red), 3rd (yellow), 4th (magenta), and finally, 5th (green). The evolution of the fitness function of the best individual in five runs, using the three penalty functions, such as, original APM, and monotonic APM, are provided in Fig. 3.

Table 2- the average, the best, and the worst values of the fitness function at the end of the evolutionary process.

Penalty strategy	Average	Best	Worst
SPF	-97,50	-98,27	-96,39
APM	-97,00	-100,00	-93,97
Monotonic APM	-96,07	-98,37	-93,81

The penalty parameters variation throughout the evolution $k_j, (j = 1, 8)$, concerning the original APM, for each one of the eight constraints, are provided in Figs. 5 and 6. For the monotonic APM, the variation of these parameters is shown in Fig. 7. The constraints violation throughout the generations are presented in Figs. 9 and 10 for the original and monotonic APM, respectively. Constraints violation throughout generations presented by their average, maximum and standard deviation are provided in Figs. 8, 9, and 10, for the static, original APM, and monotonic APM, respectively.

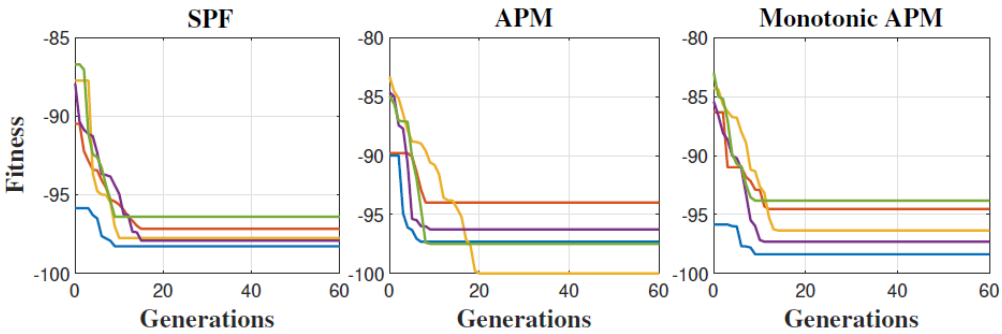

Figure 3- The evolution of the fitness function of the best individual in five runs.

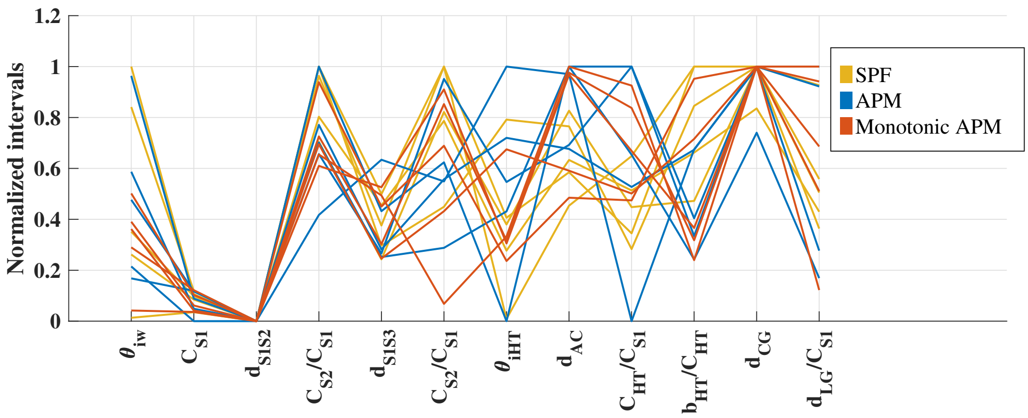

Figure 4- Normalized final design variables, in parallel coordinates, concerning the best solutions.

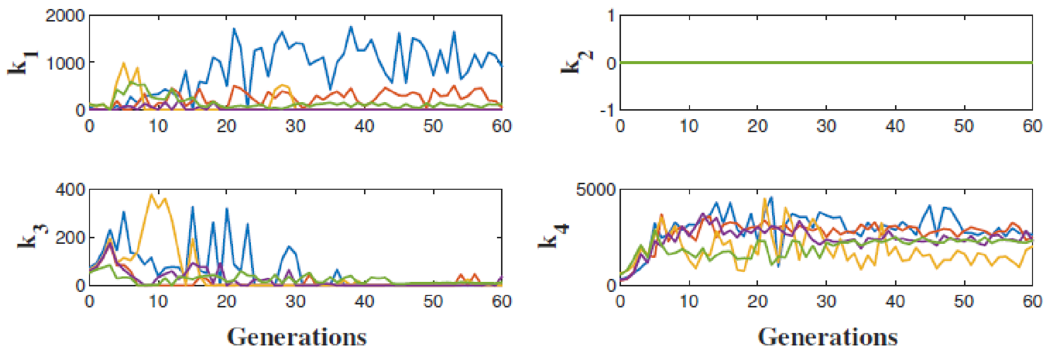

Figure 5- Variation of $k_j, (j = 1, 4)$ obtained by APM.

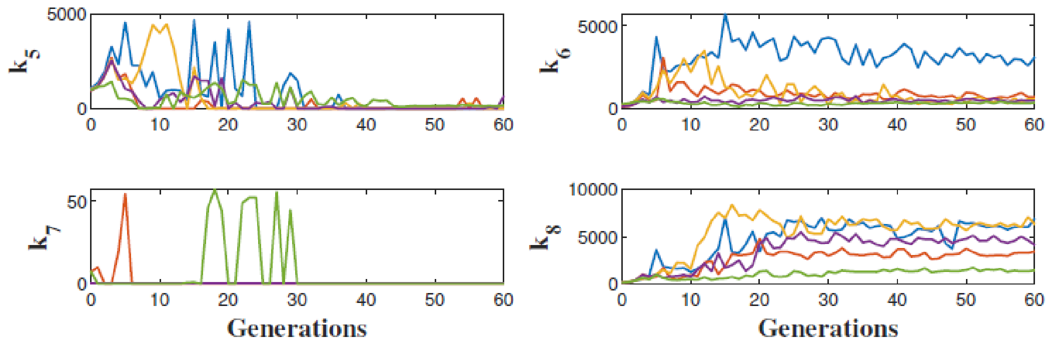

Figure 6- Variation of $k_j, (j = 5, 8)$ obtained by APM.

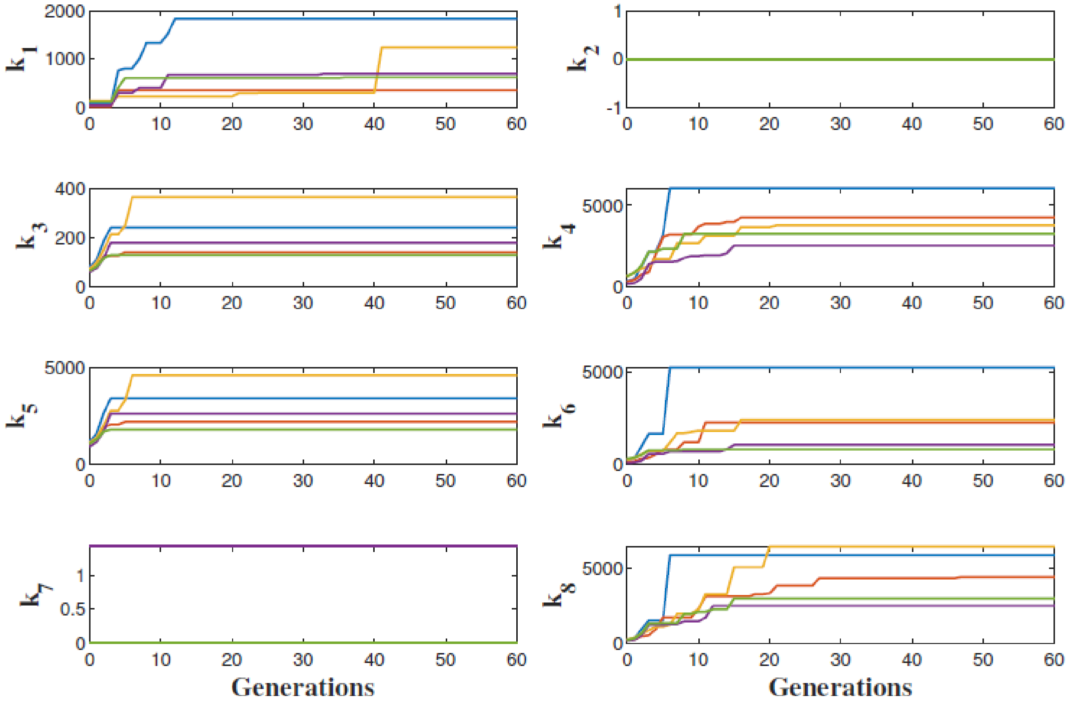

Figure 7- Variation of k_j obtained by monotonic APM

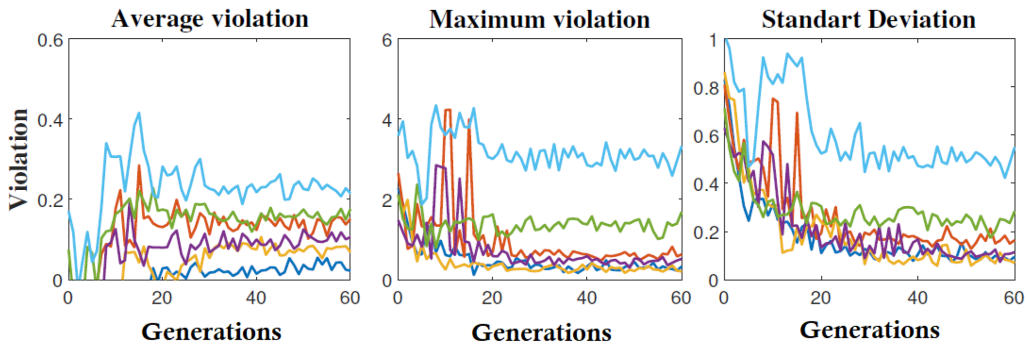

Figure 8- Constraints violation throughout generations for SPF.

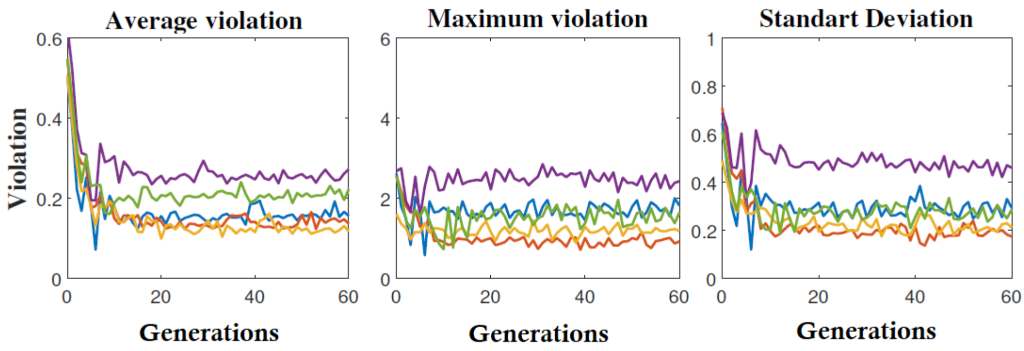

Figure 9- Constraints violation throughout generations for APM.

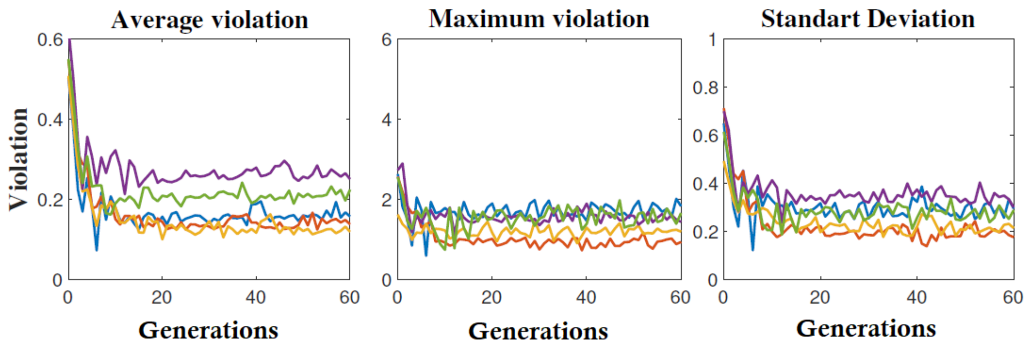

Figure 10- Constraints violation throughout generations for monotonic APM.

7. CONCLUSIONS

The main objective proposed in this paper was to investigate the performance evaluation of a parameter-less constraint-handling technique in comparison with a static penalty function to solve an MDO problem. As a result, APM avoided the trial-and-error process in adjusting static penalty parameters used in previous studies. In this sense, the computational time decreased to search for adequate penalty coefficients. In addition, APM was able to find better solutions when compared to those obtained using a static penalty function.

The k_8 penalty coefficient concerning the TBV constraint was the most sensitive in the optimization problem, reaching the highest values compared to the other ones, indicating that this was the constraint more difficult to be satisfied, as can be observed from Fig.6 and 7. Setting a static penalty function, the best solution found was infeasible concerning TVB constraint. On the other hand, the best solutions found were rigorously feasible for both variants of the APM. Comparing the two APM variants studied, the original APM achieved better results, with the best solution among all runs, and also with better average values of solutions (Table 2).

The study has aggregated great value to the UAV optimization algorithm, adding a robust and parameter-less penalty scheme that has proven to provide better results than the original method. Besides improving the results achieved, easily adapting the optimization for new penalties is also a substantial gain to the project development. Finally, it's concluded that the coupling of APM and BAT algorithm it's a robust strategy for solving optimization problems.

Acknowledgements

The authors thank MATLAB for the student license provided. The third author thanks the Brazilian Agencies CNPq (grant 306186/2017-9), FAPEMIG (Grant TEC PPM-00174-18).

REFERENCES

H.J.C. Barbosa and A.C.C. Lemonge. An adaptive penalty method for genetic algorithms in constrained optimization problems. In H. Iba, editor, *Frontiers in Evolutionary Algorithms*, pages 9–34. I-Tech, Rijeka, Croatia, 2008.

J.C. Bean and A.B. Hadj-Alouane. A dual genetic algorithm for bounded integer programs. Technical report, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI, USA, October 1992. TR No. 92-53.

E.C.R. Carvalho, H.S. Bernardino, P.H. Hallak, and A.C.C. Lemonge. An adaptive penalty scheme to solve constrained structural optimization problems by a craziness based particle swarm optimization. *Optimization and Engineering*, 18(3):693–722, 2017.

R. Datta and K. Deb. *Evolutionary constrained optimization*. Springer, 2014.

K. Khan and S. Ashok. A comparison of BA, GA, PSO, BP and LM for training feed forward neural networks in e-learning context. *International Journal of Intelligent Systems and Applications*, 4, 06 2012. doi: 10.5815/ijisa.2012.07.03.

A.C.C. Lemonge and H.J.C. Barbosa. An adaptive penalty scheme for genetic algorithms in structural optimization. *International Journal for Numerical methods in engineering*, 59(5): 703–736, 2004.

A.C.C. Lemonge, H.J.C. Barbosa, and H.S. Bernardino. Variants of an adaptive penalty scheme for steady-state genetic algorithms in engineering optimization. *Engineering Computations*, 2015.

G.A.B. Pereira, L.R.A. Oliveira, I.C.S. Junior, and C.H. Sant'Ana. Otimização geométrica de aeronaves remotamente pilotadas de perfil cargueiro via ecolocalização. In *50th Brazilian Symposium on Operational Research*, 2018.

M. Schoenauer and S. Xanthakis. Constrained ga optimization. In *Proc. 5th International Conference on Genetic Algorithms*, pages 573–580. Morgan Kaufmann, 1993.

P. Schuster, A. Davol, and J. Mello. In *Student Competitions The Benefits And Challenges*, pages 11.1155.1–11.1155.11, 06 2006. doi: 10.18260/1-2–1055.

R. Talal. Comparative study between the (ba) algorithm and (pso) algorithm to train (rbf) network at data classification. *International Journal of Computer Applications*, 92:16–22, 2014.

D.E.C. Vargas, A.C.C. Lemonge, H.J.C. Barbosa, and H.S. Bernardino. Differential evolution with the adaptive penalty method for structural multi-objective optimization. *Optimization and Engineering*, 20(1):65–88, 2019.

Xin-She Yang. A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)*, pages 65–74. Springer, 2010.