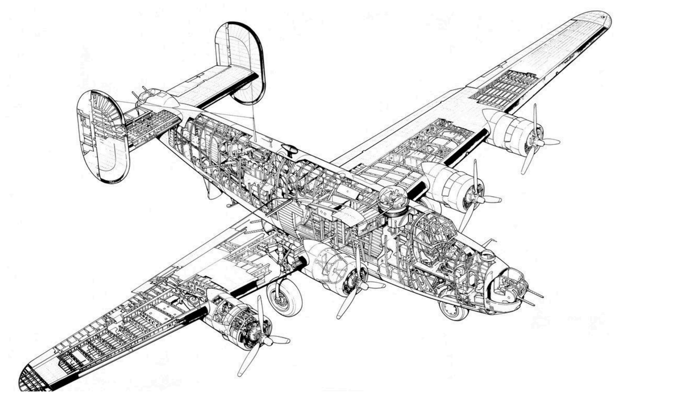
DEVELOPMENT OF A DECISION SUPPORT SYSTEM FOR DESIGN OPTIMIZATION USING METAHEURISTIC ALGORITHMS:

A Case Study at Turkish Aerospace

ENİS ÇALIŞIR - GİZEM ÖZBİLİR - ÖYKÜ ÇORAPÇIOĞLU - ECE ZIRHLI

Assoc. Dr. Fehmi Burçin ÖZSOYDAN - Adem TUNÇDAMAR

Dokuz Eylul University Faculty of Engineering



ABSTRACT

With striking and notable characteristics, metaheuristic algorithms offer quite unique opportunities in solving particularly constrained optimization problems. This project aims to develop a decision support system using reputable population-based metaheuristic algorithms for aerospace mechanical design optimization, especially for stiffened panels in airframes. A MATLAB GUI interface is designed to obtain parameters such as moment, shear forces, tensile strength, elongation forces and compressive strength from decision makers. Users can also set the metaheuristic algorithm parameters. The stiffened panel design is optimized considering strength and buckling constraints using Particle Swarm Optimization, Differential Evolution, Arithmetic Optimization and Genetic Algorithms. An extensive experimental study and nonparametric statistical tests show promising results of this approach.

INTRODUCTION

The process of optimizing a function in the best or worst way under certain constraints is called optimization. The main goal of optimization is to ensure that a process is performed with maximum efficiency and effectiveness. However, most real-world problems cannot be solved exactly due to the limitations of classical optimization techniques. The main reason for this is related to the search procedures within each technique. The general structure common to all optimization processes consists of a defined space, a set and the mappings between them. The defined space of the problem allows us to identify the types of possible solutions each time we face an optimization problem. The problem space of an optimization problem is a set of possible solutions. A solution candidate is an element of the problem space of a given optimization problem. The union of all solutions to an optimization problem is called the solution space. The search space of an optimization problem is the set of all elements that can be handled by search operations. This structure allows optimizing problems with specific constraints, objetive functions, and variables.

Figure 1. Project Fishbone Diagram

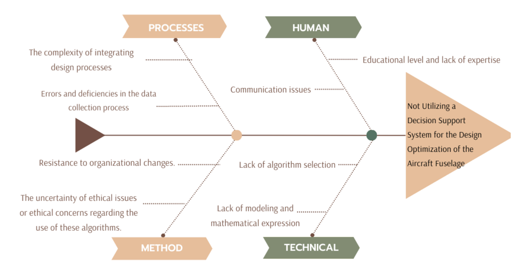
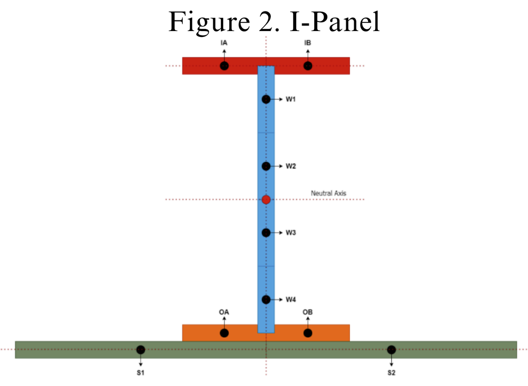


Figure 2. I-Panel



METHODOLOGY

Particle Swarm Optimization

In the implementation of Particle Swarm Optimization (PSO), the social behavior model among birds is considered. Each bird exchanges information about their positions, velocities, and fitness among themselves, and this exchange influences the behavior of the swarm to increase the likelihood of migration towards more optimal fitness regions. It is assumed that during their flights, each bird in a swarm continuously processes information about its own current position and velocity, as well as among the positions of other birds in the swarm.

Figure 4. Particle movement strategy

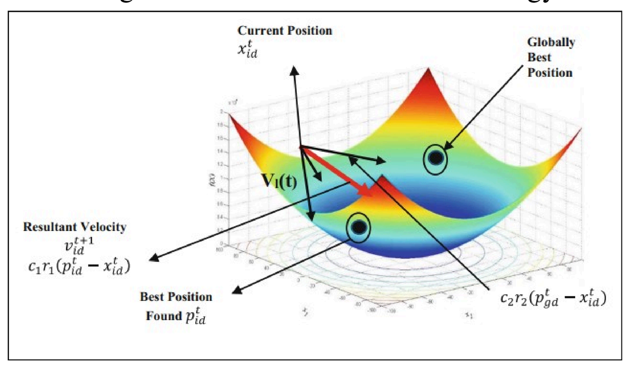


Figure 5. PSO Pseudo Code

```
Create and initialize a D- dimensional swarm with P particles.
repeat
   for each particle i= 1,..., P do
      if f(Xi) > f(Pbesti) then // f() represent the fitness function
         pbesti = Xi
      end
      if f(pesti) > f(nbesti) then
         nbesti = Pbesti
      end
   end
   for each particle i=1,...,P do
      update the velocity vector
      update the position vector
   end
until stopping condition is true;
```

Aritmetic Optimization Algorithm

This generated solution space is then refined through a set of optimization rules iterated under a given objective function. This method forms the basis of optimization techniques. In addition, under a given mathematical problem, the probability of obtaining the global optimum solution increases with a sufficient number of solution spaces and random iterations.

Figure 8. The search phases of the AOA

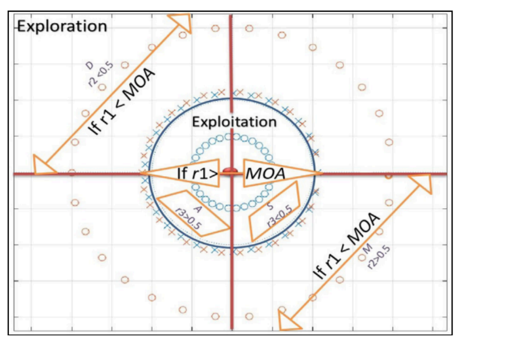


Figure 9. AOA Pseudo Code

```
1: Initialize the Arithmetic Optimization Algorithm parameters α, μ
2: Initialize the solutions' positions randomly. (Solutions: i=1, ..., N.)
3: while (C Iter < M Iter) do
4:    Calculate the Fitness Function (F F) for the given solutions
5:    Find the best solution (Determined best so far).
6:    Update the MOA value using Eq. (2).
7:    Update the MOP value using Eq. (4).
8:    for (i=1 to Solutions) do
9:       for (j=1 to Positions) do
10:         Generate a random values between [0, 1] (r1, r2, and r3)
11:         if r1 >MOA then
12:           Exploration phase
13:           if r2 >0.5 then
14:             Apply the Division math operator (D " ÷ ").
15:             Update the ith solutions' positions using the first rule in Eq. (3)
16:           else
17:             Apply the Multiplication math operator (M " × ").
18:             Update the ith solutions' positions using the second rule in Eq. (3)
19:          end if
20:         else
21:           Exploitation phase
22:           if r3 >0.5 then
23:             Apply the Subtraction math operator (S " − ").
24:             Update the ith solutions' positions using the first rule in Eq. (5).
25:          else
26:             Apply the Addition math operator (A " + ")
27:             Update the ith solutions' positions using the second rule in Eq. (5)
28:         end if
30:       end for
31:    end for
32:    C Iter=C Iter+1
33: end while
34: Return the best solution (x).
```

Differential Evoution Algorithm

The Differential Evolution Algorithm is an algorithm for solving optimization problems. Initially, a random population is created where each individual represents a solution vector of the problem. In each iteration, the population evolves towards better solutions and eventually converges to a stopping point or the best solution. The Differential Evolution Algorithm has been widely used as an effective tool for solving complex and high-dimensional optimization problems.

Figure 6. An illustration of the mutation phase

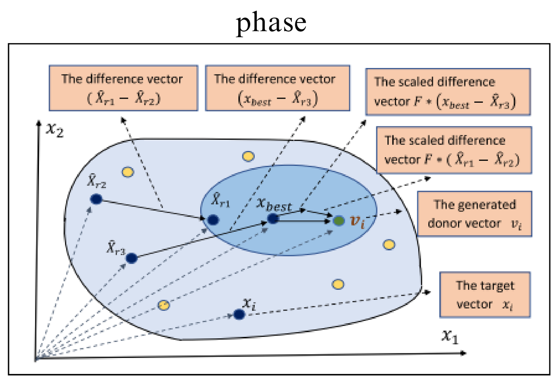


Figure 7. DE Pseudo Code

```
Begin
Set iteration t=1.
Define problem dimension.
Generate initial population and evaluate the fitness.
   While (termination condition not reached).
      for each population in current generation do
         Generate objective Function
         Create offspring using Differantial Evalution Operators
         Mutation
         Crossover
         Select the best offspring for the next generation.
      end for
set iteration t= t+1.
end while
end
```

Genetic Algorithm

The Genetic Algorithm (GA) is a search heuristic inspired by natural selection and genetics, used to solve optimization and search problems. It starts with an initial population of potential solutions represented as chromosomes. The process iterates through evaluation, selection, crossover and mutation until a termination condition is met, resulting in an optimal or near-optimal solution.

Figure 10. GA and Data Structure

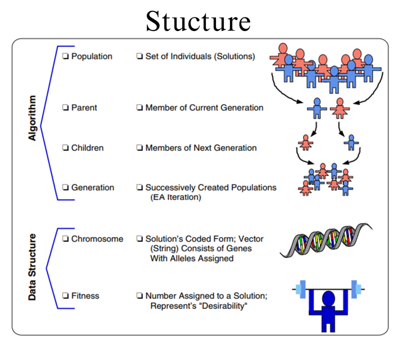


Figure 11. GA Crossover and Mutation Operator

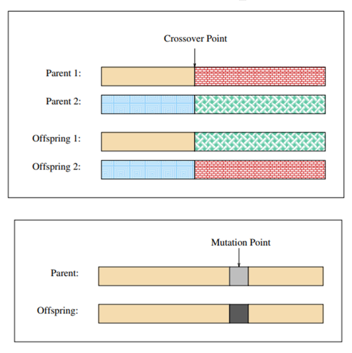


Figure 12. AOA Pseudo Code

```
1:  begin
2:     t=0;
3:     initialize P(t);
3:     evaluate P(t);
4:     while (termination criterion not met) do
5:        recombine P(t) to obtain C(t);
6:        mutation;
7:        evaluate C(t);
8:        select P(t+1) from P(t) and C(t)
9:        t=t+1;
10:    end
11: end
```

STRENGTH ANALYSIS

The strength analysis of the stiffened panel involves a detailed calculation and optimization process to ensure structural integrity and safety. The process begins with defining the main components of the panel and identifying key design parameters such as width (W) and thickness (T). These parameters are essential for calculating the sectional properties of the panel, including the moment of inertia, cross-sectional area, and center of gravity.

Subsequently, the Principal and Von Mises stresses are determined for each component, providing insight into the stresses the material will experience under various loads. Finally, the Reserve Factor (RF) is calculated for both stress types, indicating the safety margin between actual and allowable stress levels. This comprehensive analysis ensures that the components can safely withstand the applied loads.

Table 1. The Formulas Used for Strength Analysis

Step	Description	Formula
Sectional Properties	Moment of Inertia about x-axis	$I_{xx,i} = \frac{b \cdot h^3}{12}$
	Total Cross-Sectional Area	$\sum A = \sum_{i=1}^{n} b_i \cdot h_i$
	y-coordinate of Center of Gravity	$Y_{COG} = \frac{\sum_{i=1}^{n} A_i \cdot y_i}{\sum A}$
	Moment of Inertia about Center of Gravity	$I_{xx,\text{cog}} = \sum_{i=1}^{n} I_{xx,i} + A_i \cdot \Delta y^2$
Stress Calculations	Shear Stress	$\tau = \frac{T}{\sum A}$
	Principal Stress	$\sigma_i = \left(\frac{N}{\sum A} + \frac{M_{\text{sum}} \cdot y_i}{I_{xx,\text{cog}}} \right)$
	Von Mises Stress	$\sigma_{vm,i} = \sqrt{\sigma_i^2 + 3 \cdot \tau^2}$
Reserve Factor (RF)	With Principal Stress	$RF_i = \frac{F_{tu}}{\sigma_i}$
	With Von Mises Stress	$RF_i = \frac{F_{tu}}{\sigma_{vm,i}}$

STABILITY ANALYSIS

This section presents a stability analysis of the stiffened panel structure. Local stability analysis has been applied to each component of the structure. If the stress value calculated in the elements of the structure is negative, the system begins to buckle. The surface geometry, loads, and material priorities of the structure under investigation should be examined. Subsequently, the plate boundary conditions, such as free, hinged, and fixed edges, are analyzed to determine which case of single compressions they are subjected to. The single compressions evaluated on flanges and plates will determine the fixed parameters for the structural components in the calculations, according to their respective cases. The following material properties are used:

Table 2.Material Properties for Stability Analysis

F_{tu}	Tension Ultimate Strength
F_{ty}	Tension Yield Strength
F_{cy}	Compression Yield Strength
n	Ramberg Osgood Coefficient in Tension
n_c	Ramberg Osgood Coefficient in Compression
E	Elasticity Modulus
n_u	Poisson Ratio

Figure 3. Aircraft Fusalage Construction Design

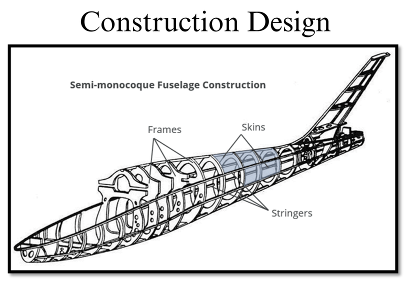


Table 3. The Formulas Used for Stability Analysis

Formula	Description
The buckling stress of a thin flat plate is the stress at which a change to the stable configuration of equilibrium occurs.	Plate Buckling Stress
$\sigma_{cr} = \eta \times K_c \times \left(\frac{e}{b}\right)^2$	Buckling Stress Expression
$K_c = \frac{\pi^2 \times k_c}{12 \times (1-\nu_e^2)}$	Plate Buckling Factor
$\epsilon = \left(\frac{\sigma}{E}\right) + 0.002 \times \left(\frac{\sigma}{\sigma_{0.2}}\right)^2$	Ramberg and Osgood Model for Strain
$E_s = \frac{\sigma}{\epsilon}$	Elastic Stress
$\nu = \left(\frac{E_s}{E} \times \nu_e\right) + \left[\left(1 - \frac{E_s}{E}\right) \times \nu_p\right]$	Elastic-Plastic Poisson Ratio
Plastic Correction Factor	Plastic Correction Factor for Buckling Stress
$RF_i = \frac{F_{tu}}{\sigma_{cr}}$	Reserve Factor

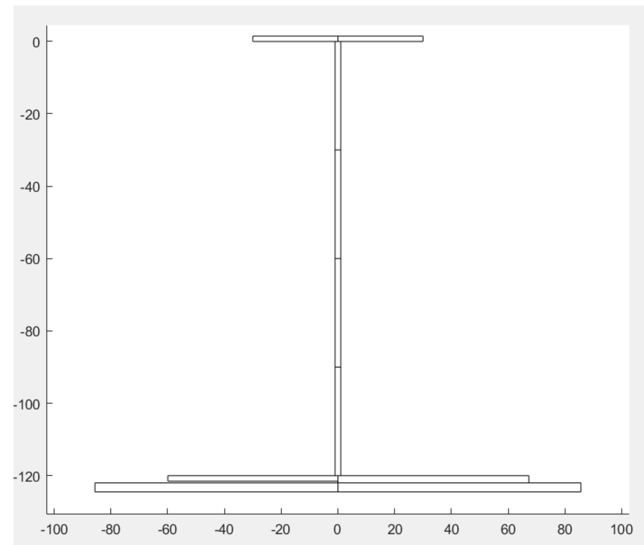
DECISION SUPPORT SYSTEM AND RESULTS

Parameters for use in optimization algorithms were obtained from the user through a MATLAB GUI interface. The user was prompted to determine the feasible range of thickness and length values, force values and algorithm parameters for optimization process.

Figure 13. Matlab GUI Interface

To visualize the real-world applicability of the obtained dimensions, MATLAB was used to create 'plot' graphs. These plots provided a visual representation of the shapes for each run, allowing for an intuitive assessment of the practicality and feasibility of the design parameters in a real-world context. This visualization step was crucial in ensuring that the optimized dimensions not only met theoretical requirements but also translated effectively into practical applications.

Figure 14. DE Plot Graph of I-Panel



CONCLUSION

Each algorithm was run 50 times over 3000 iterations, and the results were meticulously recorded and analyzed. The primary goal was to determine which algorithm provided the most reliable and efficient solution. To compare the results obtained from the algorithms, the average, highest, lowest values, and standard deviations were calculated. These metrics provide a comprehensive understanding of each algorithm's performance, highlighting their consistency and efficiency across multiple runs.

Table 4. Algorithm's Performance Metrics

	AOA	GA	PSO	DE
Highest Value	-0,00345623830	-0,00361045940	0,37842594285	-0,00361131348
Lowest Value	-0,00359215569	-0,00361131876	-0,00357134236	-0,00361132173
Standard Deviation	0,000030010973828	0,00000016496446	0,13194981114240	0,00000002173047
Average Value	-0,00353804462	-0,00361113512	0,07228021768	-0,00361131903

Based on the comparative analysis of the results, it is evident that Differential Evolution (DE) performs the best for the given mechanical optimization problem. It not only achieves the most consistent results, as evidenced by the lowest standard deviation, but also produces a highly reliable performance with its average value closely matching that of the Genetic Algorithm (GA).

In summary, considering all the constraints, objectives, and performance metrics, Differential Evolution (DE) provided the best results in optimizing the mechanical design of the I-panel. Its consistent performance, minimal variability, and ability to reliably reach near-optimal solutions make DE the preferred choice for similar optimization challenges.

REFERENCES

- Bo, R. F. (2009, August). Genetic algorithm-based approach to concept solving for mechanical product in conceptual design. In 2009 Fifth International Conference on Natural Computation (Vol. 4, pp. 254-258). IEEE.
- Byrne, J., Cardiff, P., & Brabazon, A. (2014). Evolving parametric aircraft models for design exploration and optimisation. Neurocomputing, 142, 39-47
- Elegbede, C. (2005). Structural reliability assessment based on particles swarm optimization. Structural Safety, 27(2), 171-186.
- Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., & Gandomi, A. H. (2021). The arithmetic optimization algorithm. Computer Methods in Applied Mechanics and Engineering, 376, 113609.
- Oruc, R., & Baklacioglu, T. (2024). Cruise range modeling of different flight strategies for transport aircraft using genetic algorithms and particle swarm optimization. Energy, 294, 130917.
- Saruhan, H., & Uygur, İ. (2003). Design optimization of mechanical systems using genetic algorithms. Sakarya University Journal of Science, 7(2), 77-84.

ACKNOWLEDGEMENT

We would like to express our respect and gratitude to our advisor, Associate Professor Dr. Fehmi Burçin ÖZSOYDAN, and our industry consultant, Adem TUNÇDAMAR, for their support in the realization of this study.

- This study is supported by Turkish Aerospace Industries Inc. within the scope of Lift Up Project. (Project no: 64c15b897e889)
- This study is supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) within the scope of 2209-B University Students Research Projects for Industry.