Optimisation strategies for multi-layered armour plates

ARTICLE TEMPLATE

Optimisation strategies for multi-layered armour plates

I. Reis[a], F. Teixeira-Dias[b] and J. Dias-de-Oliveira[aa]

[a]TEMA, Department of Mechanical Engineering, Universidade de Aveiro, Portugal; [b]School of Engineering, The University of Edinburgh, United Kingdom

ARTICLE HISTORY
Compiled January 30, 2023

ABSTRACT
A set of non-linear optimisation algorithms are combined with a finite element simulation code to analyse an energy absorption and elastic stress wave propagation problem in multi-layer/multimaterial armour systems under ballistic impacts. An Abaqus Python script is used to simulate the ballistic event and to generate the variables and post-processing outputs necessary for the integration with the optimisation algorithms. A number of modelling strategies are considered and three optimisation algorithms are used: Particle Swarm Optimisation (PSO), Genetic Algorithm (GA) and Simulated Annealing (SA). The performance and efficiency of each algorithm are assessed through four benchmark tests with different levels of complexity. A multi-objective optimisation procedure is proposed that uses the most efficient algorithm based on every single-objective formulation, variables and constraints from the benchmark tests, resulting in a highly non-linear problem. The proposed optimisation methods successfully achieve the study purposes both in the simulation of generic ballistic impacts and in the quality of the optimised solutions, demonstrating the potential for this type of optimisation method on terminal ballistic applications, serving as a standpoint for further studies into higher energy impacts and material non-linearities.

KEYWORDS
Design optimisation; Armour systems; Terminal ballistics; Genetic algorithm; Particle swarm optimisation; Simulated annealing; Multi-objective optimisation

1. Introduction and Background

There has been a prevalent need for continuous research on lightweight Armour and Protection Systems (APS) since centuries ago. The use of metals (and some natural materials such as leather) prevailed on most armour systems until the 1960s and the Vietnam war, where for example ceramics started being used in helicopter APS.

1.1. Armour Materials and Structures

There are four main metallic materials (steel, aluminium, magnesium and titanium, and their alloys) that are practical and functional in armour applications. Only steel and aluminium, however, are currently widely used, mostly due to raw material costs and their workability and weldability. These two metals and their alloys have been widely used as they offer reasonable hardness with good ductility and toughness [1]. Their usage in protective structures might, however, become obsolete as there are light-weight non-metallic alternatives available such

CONTACT J. Dias-de-Oliveira. Email: jalex@ua.pt

as ceramics and high-strength fibres and their composites. Nonetheless, it is still likely that metals will be used in APS for some time, also because of current developments in porous or micro- and nano-architecture structures, resulting in promising designs such as cermets (hybrid armour systems made of a ceramic-metal combinations) and auxetic structures.

Trials were done in 1992 to prove that a two-layer design combining a ceramic with aluminium presents reasonable mechanic properties with lower areal density than a single layer metallic plate [2]. Since then, two-layer and multiple layer design configurations have been widely used in modern armour structures. The underlying physical and design principle is to have a hard ceramic front plate, which can break the projectile, and a rear ductile plate (e.g. metallic or polymeric) to help contain the bullet and fragments, and absorb the remaining kinetic energy of the impactor. The hard front plate will also attenuate the concentrated pressure on the rear plate by distributing it on the plate plane [3]. A multi-layer configuration additionally alters the wave propagation velocity, with potential for reducing it and attenuating damage.

Hybrid structures are combinations of two or more materials assembled in such ways as to have attributes not offered by either one of the components alone [4]. These are often obtained by filling gaps in a structure with a second material. They can, however, also be sandwich structures, foams, etc. The properties of hybrid structures are becoming better understood and controlled, and one example is the use of ceramic inclusions to enhance projectile deflection, cause self-sealing of the hole and force shear localisation [5]. Ths use of high-performance fibres (e.g. woven fabrics) is also a common approach, making use of the typically high tensile strength of the fibres ($2-3$ GPa) and their relatively low densities ($1000-1500$ kg/m^3), while simultaneously showing reasonable strain to failure ($3-6\%$) and excellent energy absorbing capability [1].

1.2. Optimisation of Multi-layered Armour Plates

In 2005 Park et al. [6] tested multi-layered plates under ballistic impact in one of the first experimental campaigns targeting the design optimisation of multi-layered plates. The main aim of this study was to minimise the weight of the target by adjusting layer thicknesses without compromising the APS integrity on a high velocity impact, that is, ensuring there is no projectile perforation. A Lagrangian explicit time integration finite element code (NET2D) developed by two of the authors was used for the simulations. The Johnson-Cook constitutive model was implemented due to the high strain-rate plastic deformations involved and temperature-dependent material behaviour. The authors observed that the objective of minimising strain energy to maximise the strength of the target can also be expressed by minimising the average temperature or average equivalent plastic strain (EQPS). Two interesting case studies were looked at: (i) one where the objective was to minimise the average temperature, and (ii) another where the objective was to minimise the EQPS. The restrictions for both cases were identical, and related to layer thicknesses and to limiting the maximum EQPS at critical locations. The optimisation algorithm used was a Response Surface Method (RSM) and, in the end, both objective-functions presented similar results. The optimal results from the optimisation algorithms agreed with the results from NET2D. Although this was one of the first studies on structure optimisation in transient events, the need and the potential for further work is clear.

Yong et al. [7] presented a study on the application of genetic algorithms for optimising the response of composite material structures from impact loads. Two optimisation scenarios were presented and discussed: (i) the low velocity impact on a slender laminated strip and (ii) the high velocity impact on a rectangular plate, using a spherical impactor. The goal was to

minimise the peak deflection of the impacted plate and to minimise the penetration velocity or maximise the rebound velocity, respectively. The variables in both cases were the ply angle stacking sequences. It is worth noting, however, that in the former scenario the ideal stacking sequence was already known to be 0° for all plies. The low velocity case-study was used as benchmark for the algorithm techniques. A genetic algorithm coupled to the commercial hydrocode LS-DYNA was adopted to simulate the impact cases. A comparison with a commercial optimisation package (LS-OPT), that included optimisation methods such as the Monte Carlo and Neural Network techniques, was also made. For the low velocity benchmark, the results from both the Monte Carlo and the neural network techniques were particularly good. It was observed that the Monte Carlo method performed better for smaller test sets while the neural network had better results for larger sets. The adopted genetic algorithm performed better for the high velocity scenario, where the search space is larger and non-linear.

A study by the same authors [8] focused on applying the same genetic algorithm coupled to LS-DYNA to optimise hybrid multilayered plates subjected to ballistic impacts. In this study, low cost but sufficiently accurate models were generated to be used by the optimisation algorithm to generate new hybrid off-springs. The goal was to minimise weight and design costs from a selection of isotropic metals, polymers and orthotropic fibre-reinforced laminates. Both the number of plies and the mechanical properties of the materials were problem variables, while the overall thickness was kept constant. Experimental validation of the optimal designs was successful using a single stage gas gun. Nevertheless, the authors are clear that further work needs to be done with sufficient computational power and using finer meshes, equations of state and sophisticated material models so that hybrid systems can be identified from a wider range of materials, designs and threats.

This work intends to further understand the potential of optimisation strategies for the design of hybrid multilayered plates. It focuses on the way these structures and materials influence the propagation of elastic waves during impact, namely on the influence of material combination of material layers and on the interlayer as an energy absorbing layer. This is done while studying the behaviour of optimisation procedures using different algorithms and problem formulations (i.e. variables, objective functions and constraints).

2. Model design and specifications

This work is based on a ballistic impact model with a target and an impactor. Both are naturally affected by the elastic and/or plastic waves generated by the impact. The one-dimensional elastic wave speed in the projectile and the elastic wave speed in the semi-infinite plate are

$$c = \sqrt{E/\rho} \quad \text{and} \qquad (1)$$
$$c_{\mathrm{p}} = \sqrt{\frac{E_{\mathrm{p}}(1-\nu_{\mathrm{p}})}{\rho_{\mathrm{p}}(1+\nu_{\mathrm{p}})(1-2\nu_{\mathrm{p}})}},$$

respectively, where E is the Young's modulus, ν is the Poisson's ratio and ρ is the material density [1]. In more complex media, such as multi-layer armour plates, the transmission of elastic waves depends not only on material properties but also on the nature of the interfaces and contacts between layers. Assuming perfectly bonded layers, a pressure pulse with intensity σ_{i} arriving at the interface will be partly transmitted into the connected layer and partly reflected back into the layer where it originated. The intensity of the transmitted pulse is σ_{t} and the intensity of the reflected pulse is σ_{r}. The proportions of the incident wave that are

reflected and transmitted depend on the relative mechanical impedance of the system. The mechanical impedance is $Z = \sqrt{E\rho} = \rho c$. Assuming the interface between two layers A and B is in equilibrium, the intensities of the stress pulses can be related as

$$\sigma_{\mathrm{i}} = \sigma_{\mathrm{t}} + \sigma_{\mathrm{r}}. \tag{2}$$

Additionally, continuity at the interface is defined as a function of particle velocities as

$$u_{\mathrm{i}} = u_{\mathrm{t}} + u_{\mathrm{r}}. \tag{3}$$

The transmitted and reflected wave fractions can be found by combining Equations 2 and 3, leading to

$$\frac{\sigma_{\mathrm{t}}}{\sigma_{\mathrm{i}}} = \frac{2\sqrt{E_{\mathrm{B}}\rho_{\mathrm{B}}}}{\sqrt{E_{\mathrm{A}}\rho_{\mathrm{A}}} + \sqrt{E_{\mathrm{B}}\rho_{\mathrm{B}}}} = \frac{2Z_{\mathrm{B}}}{Z_{\mathrm{A}} + Z_{\mathrm{B}}}, \tag{4}$$

$$\begin{aligned} \frac{\sigma_{\mathrm{r}}}{\sigma_{\mathrm{i}}} &= 2\left(\frac{\sqrt{E_{\mathrm{B}}\rho_{\mathrm{B}}} - \sqrt{E_{\mathrm{A}}\rho_{\mathrm{A}}}}{\sqrt{E_{\mathrm{A}}\rho_{\mathrm{A}}} + \sqrt{E_{\mathrm{B}}\rho_{\mathrm{B}}}}\right) \\ &= \frac{2(Z_{\mathrm{B}} - Z_{\mathrm{A}})}{Z_{\mathrm{A}} + Z_{\mathrm{B}}}. \end{aligned} \tag{5}$$

From these wave relations it can be seen that the transmitted wave is always positive (compressive) whilst the reflected wave can be either positive or negative (compressive or tensile, respectively) depending on the ratio of the mechanical impedance of both materials. If layer B has lower impedance the reflected wave will be tensile, and compressive if the mechanical impedance of layer A is the lowest. In the absence of a backing plate (e.g. if material B is air), the incident stress wave is all reflected back.

The multi-layer target system in this work consists of a three-layer plate with a wave-facing area 500×300 [mm^2]. The front and rear plates are 25 mm thick, and the inter-layer is 10 mm thick. The stress pulse is generated by the impact of a blunt cylinder with diameter 20 mm and length 60 mm. Figure 1 shows an example of the observed pressure (stress) pulse as recorded on the rear face of the target. Abaqus was used in this work for the simulation procedures. The element type used for the plates was the eight-node hexaedron with reduced integration (C3D8R), with hourglass control. For the projectile, as a rigid body, a four-node bi-linear rigid quadrilateral (R3D4) was used. Besides using symmetry boundary conditions in the 1/4 simplified model, a boundary condition set was also established regarding how the plate is supported. Three different scenarios were studied: (i) clamping of the side surfaces, (ii) pinning of the side edges of the rear surface and (iii) clamping of the rear surface. The third boundary condition set (shown in figure 2) presents a more stable behaviour throughout the entire impact duration, both in the analysis of through-thickness and in-plane phenomena. This is related to the fact that this configuration reduces interference between the different wave propagation directions, providing a more controlled setup for the studies in this work. Python scripts were used to automate the model generation and update for each ballistic event. This is essential for the optimisation procedures, namely to generate the variables and post-processing outputs necessary for the integration with the optimisation algorithms.

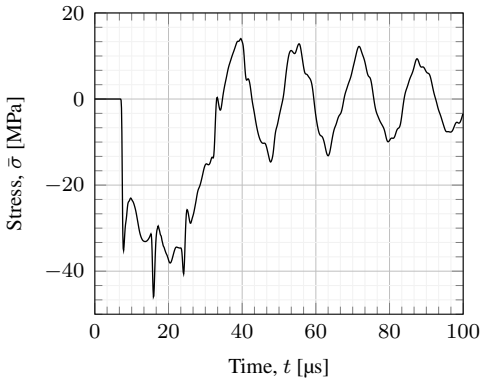


Figure 1.: First 100 µs of the stress pulse at the rear face of the target.

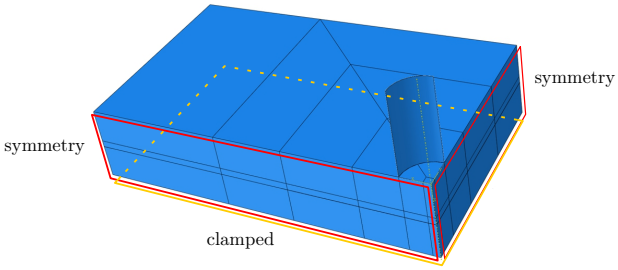


Figure 2.: Ballystic model with impactor and three target layers, including partitions for structured mesh generation, and boundary conditions.

2.1. Mesh Convergence

Pressure pulses can propagate in solids as different types of waves (e.g. transverse, longitudinal) [9]. In numerical simulation, an optimal discretisation is critical to ensure wave propagation is captured accurately. To this end, two separate mesh convergence analyses are done, to ensure the in-plane and through-thickness wave propagation phenomena are captured accurately: (i) a convergence study on the plate's plane and (ii) a convergence study along the plate's thickness. The peak stress at the rear face of the plate is used as convergence parameter. A single plate Finite Element (FE) model is used for this analysis, with eight-node brick elements with reduced integration and hourglass control. The projectile is modelled as a rigid body with four-node bi-linear rigid quadrilateral elements.

For the in-plane analysis, the mesh size is varied while keeping the through-thickness direction size constant and equal to 3 mm. A parametric study is done for 11 different finite element sizes in the range $1 - 50$ mm. Figure 3a shows the evolution of the average peak stress at the plate's rear face and the CPU time for each simulation as a function of the number of elements. It can be seen that the CPU time increases more than ten fold when refining the in-plane mesh from $80,304$ to $564,604$ elements, with an associated increase of 15% in the rear face peak stress. The coarser finite element mesh with $80,304$ is chosen as an optimal compromise (see dashed line in Figure 3a).

The results in Figure 3b show the results for the through-thickness mesh analysis. A similar procedure was used here, where 11 different discretisations were done for mesh sizes in the range $0.1 - 10$ mm. The CPU time increases significantly ($> 37\%$) when refining the trough thickness mesh from $26,880$ to $77,500$ elements, with an increase of the rear face peak stress of only 2.4%. The finite element mesh with $26,880$ is thus chosen as optimal (see dashed line

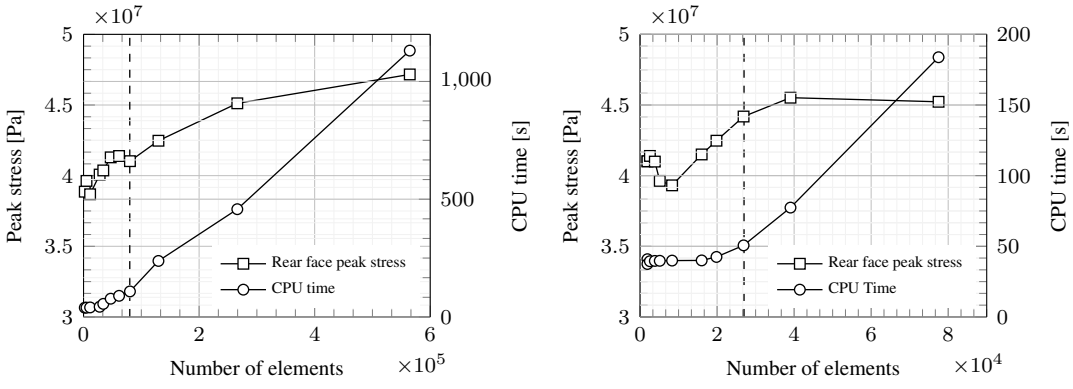


Figure 3.: Mesh convergence analysis: peak stress at the rear face of the plate and CPU time for the (a) in-plane and (b) through-thickness mess size analyses. Dashed line indicates optimised mesh discretisation in each case.

in Figure 3b). The final optimised structured mesh is shown in Figure 4. Element density is not constant on the plate to minimise CPU time without compromising the ability to capture the propagation of stress waves. Figure 5 presents stress isovalues for two example stages during the event, showing stress propagation both through-thickness and in plane.

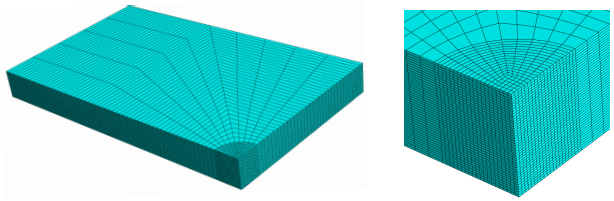


Figure 4.: Optimised finite element mesh showing detail (right) of the impact area and through-thickness discretisation.

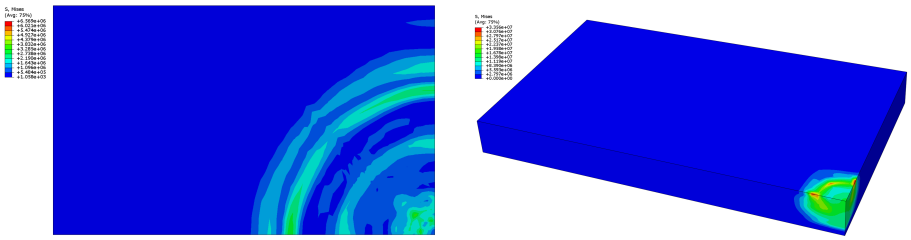


Figure 5.: Illustration of stress propagation in-plane (left) and through-thickness (right).

3. Dynamic response of the system

The transient nature of the dynamics response of the system is analysed and discussed in the following paragraphs. To achieve this it is important to first validate the numerical model, which can be done by comparing measured (numerical) to calculated (analytical) dynamic forces in the system.

3.1. Contact force and resultant stress

A comparison is made between the force calculated due to the acceleration induced in the projectile and the average contact force during the contact period. The force due to the average deceleration of the projectile while in contact with the target can be estimated as

$$F = m \left| \frac{v_1 - v_0}{t_1 - t_0} \right| = 13.08\,\mathrm{kN}, \tag{6}$$

where m is the mass of the projectile, v is the velocity, t is the time and the subscripts 0 and 1 designate the initial and final instants, respectively. The average contact force while the projectile is in contact obtained numerically is $13.07\,\mathrm{kN}$, which corresponds to a difference lower than 0.1%.

Combining the resultant force in Equation 6 and the contact area between the projectile and the plate, the resultant contact stress can be calculated as $\sigma_\mathrm{c} = F_\mathrm{c}/A_\mathrm{c} = 41.7$ MPa. The average stress while the projectile and plate are in contact is 37.1 MPa, which corresponds to a 11.6% difference to the calculated value.

3.2. Wave dynamics

The elastic wave needs to be captured with sufficient resolution to allow for accurate results. To achieve this, it is necessary to ensure: (i) the size of the time increment is low enough to capture the smallest natural period of interest, and (ii) element size is small enough to capture wave length [10].

The target is an aluminium plate with Young's modulus $E = 70$ GPa, Poisson's ratio $\nu = 0.33$ and density $\rho = 2700$ kg/m^3, which gives a theoretical elastic wave speed $c = 6197.8$ m/s (see Equation 2). The corresponding impulse wavelength is $I_\mathrm{w} = 2c\mathrm{d}t = 0.025$ m [10] and, therefore, a mesh size of $h = 1 \times 10^{-3}$ m was used in the simulations. Recalling that $h < I_\mathrm{w}$, this would be sufficient to capture the propagation of the elastic wave. The critical time increment can be calculated as

$$\Delta t = \frac{h}{c} = 1.61 \times 10^{-7}\ \mathrm{s}. \tag{7}$$

It should be noted that the Courant-Friedrichs-Lewy condition (CFL condition) is met [11,12] by using a maximum time increment of 1.0×10^{-7} s and a mesh size $h = 1 \times 10^{-3}$ m.

The numerical elastic wave velocity was calculated by tracking displacements on three nodes, as shown in Figure 6. The displacement along the Ox direction for each one of these nodes is shown in Figure 7, from which the instant when the wave reaches the node — the time of arrival of the wave — can be clearly determined. The numerical elastic wave speed was found to be $\bar{c} = 6134.6$ m/s, which is very close to the theoretical wave speed $c = 6197.8$ m/s (see equation 2), with a relative difference of approximately 1%.

Figure 8 shows the propagation of the initial and reflected elastic waves on the surface of the plate at different instants. The first snapshot, taken at $t = 54$ µs clearly shows the circular nature of the in-plane initial wave. The first reflections start to be visible at around $t = 92$ µs, from the interaction with the longer edge of the rectangular plate. The first reflection at the shorter edge of the plate is visible at approximately $t = 98$ µs, and complex wave interactions and interference are visible from there onward, as can be seen in the snapshot at $t = 128$ µs in Figure 8d.

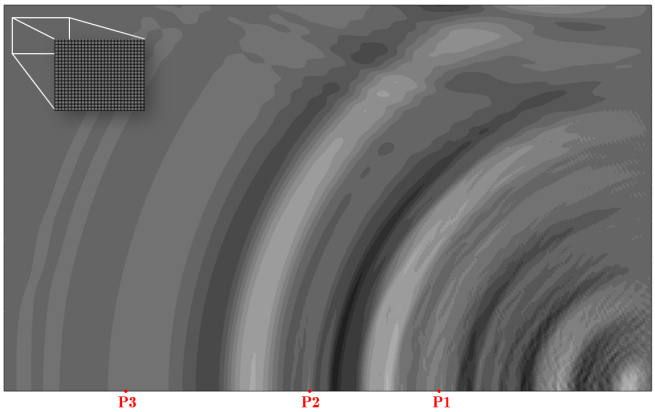


Figure 6.: Location of control nodes P1, P2 and P3 on the symmetry plane of the plate, at 82, 132 and 203 mm from the impact location, respectively.

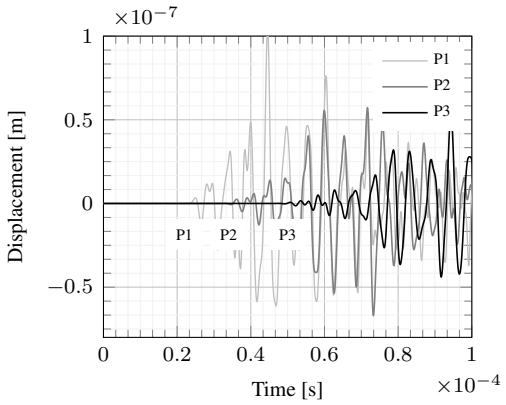


Figure 7.: Displacement history of the control nodes P1, P2 and P3. The time of arrival of the wave to each control node is used to calculate the elastic wave speed.

3.3. The interlayer

This section addresses the mechanical response of the energy absorbing interlayer and how this depends on its material and geometrical parameters. Three different parameters are analysed: (i) the interlayer thickness, (ii) the interlayer elastic material parameters and (iii) the projectile impact velocity. The mechanical properties of the tested interlayer materials are listed in Table 1 and the bar chart in Figure 9 compares their mechanical impedance.

The results in Figure 10a show the correlation between the peak stress at the rear face of the plate for an impact velocity $v_p = 20$ m/s, for the four interlayer materials. The interlayer thickness is a dominant factor that dictates the propagation of stress. The thickness of the layer in which the stress wave propagates is directly proportional to the time of arrival of the wave at its rear face [14]. This is supported by the results in Figure 10a, where it can be observed that the stress decreases with the increase of the interlayer thickness, regardless of the material. This relation is, however, not strictly linear. From all the tested materials, it can be seen that the material that led to a lower peak stress at the rear face, regardless of the thicknesses, was the aluminium foam, which presented a significantly lower stress compared to the other materials. This is counter-intuitive because EPDM has lower impedance, which should result in a higher impedance mismatch at the interface, noting that a higher impedance

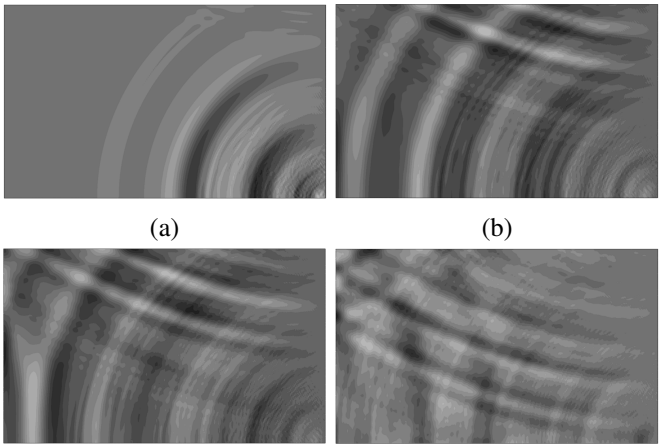


Figure 8.: Impact and reflected wave propagation on the surface of the plate at (a) $t = 54$ μs, (b) $t = 92$ μs, (c) $t = 98$ μs and (d) $t = 128$ μs.

Table 1.: Mechanical properties of the interlayer materials [13]

Material	**Young's modulus** [MPa]	**Density** [kg/m^3]	**Yield stress** [MPa]	**Poisson's ratio** [–]	**Impedance** [kg/(m^{2}s)] $\times 10^4$
EPDM	2.5	960	16.8	0.499	4.9
Aluminium foam	103.08	410	1.24	0.05	20.6
Cork	9000	293	1.0	0.30	162.4
Nylon-6	3000	1140	82	0.35	184.9
Aluminium (plate)	70×10^3	2700	276	0.33	1374.8

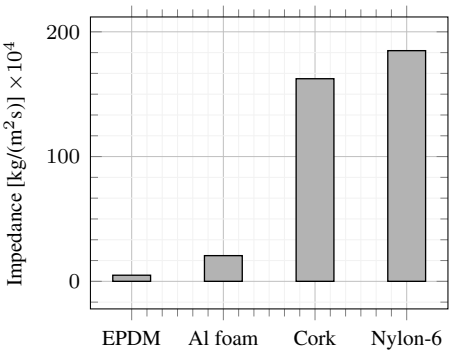


Figure 9.: Mechanical impedance of the interlayer materials.

mismatch leads to a more effective stress attenuation [1]. However, the cellular structure of the aluminium foam (and its consequent low density) play an important role in the wave propagation. It can thus be concluded that the impedance mismatch between layers is not sufficient to ensure effective stress attenuation. Different variables, such as the material's internal structure (which might also affect its density) and Poisson's ratio are also important parameters to consider.

The results in Figure 10b show the effect of the projectile velocity on the peak stress at the rear face of the plate. These results suggest that the increase in the projectile's velocity leads to an almost linear increase on the stress, regardless on the interlayer material. It should be noted, however, that plastic strains at the interlayer are visible for velocities above $v_\mathrm{p} = 10$ m/s for Nylon-6 and above $v_\mathrm{p} = 5$ m/s for cork. This leads to energy dissipation which is a possible explanation for the deviation from a linear relation between the impact velocity and the stress.

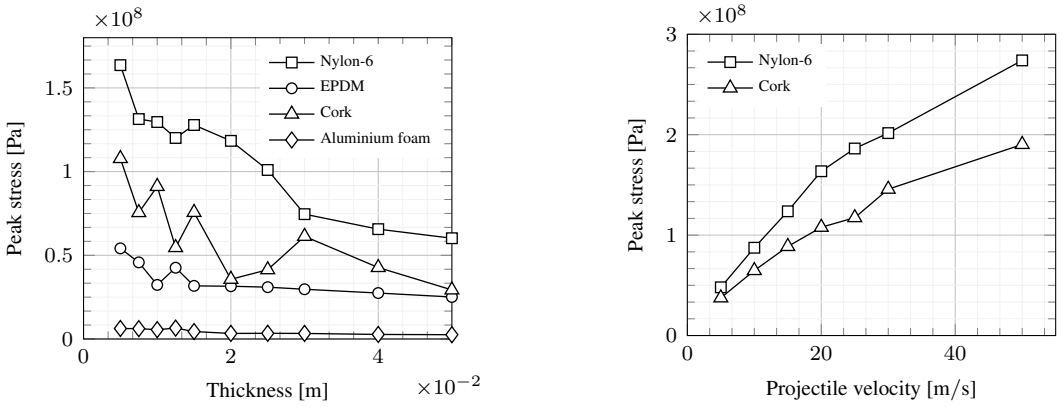


Figure 10.: Peak stress at the rear face of the target as a function of (a) the thickness of the interlayer and (b) the projectile impact velocity.

4. Optimisation framework

Optimisation is an increasingly important field on the development of engineering systems, going beyond trial and error approaches when seeking either to minimise resources or to maximise the desired benefit [15]. The general formal mathematical statement of an optimisation

problem can be expressed as

$$\begin{aligned} \text{minimise} \quad & f(\mathbf{x}) \\ \text{subjected to} \quad & g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \ldots, m \\ & h_k(\mathbf{x}) = 0, \quad k = 1, 2, \ldots, l \\ & x_i^{\min} \leq x_i \leq x_i^{\max}, \quad i = 1, 2, \ldots, n, \end{aligned} \tag{8}$$

where $f(\mathbf{x})$ is the objective function, $g(\mathbf{x})$ are m inequality constraints, $h(\mathbf{x})$ are l equality constraints and x are the n project variables within the lower and upper boundaries.

In this work, the Exterior Penalty Function Method (EPFM) is used to replace the originally constrained optimisation problems with a series of unconstrained problems. This updates the general formulation in Equation 8 by adding a penalty function P to the objective function as [16]

$$\begin{aligned} \text{minimise} \quad & F(\mathbf{x}, r_{\text{h}}, r_{\text{g}}) = f(\mathbf{x}) + P(\mathbf{x}, r_{\text{h}}, r_{\text{g}}) \\ & x_i^{\min} \leq x_i \leq x_i^{\max}, \quad i = 1, 2, \ldots, n, \end{aligned} \tag{9}$$

where r_{h} and r_{g} are constant penalty parameters and F is the augmented objective function. The penalty function is used for both equality and inequality constraints, and is equal to zero within the feasible region.

4.1. Optimisation algorithms

This section describes the three non-linear optimisation algorithms used in this work: (i) Particle Swarm Optimisation (PSO), (ii) Genetic Algorithm (GA) and (iii) Simulated Annealing (SA). All these are evolutionary algorithms [17] within meta-heuristics, which rely on probabilistic techniques.

4.1.1. Particle swarm optimisation

Particle Swarm Optimisation (PSO) was originally introduced by Kennedy and Eberhart [18], and Shi and Eberhart [19], and it was initially intended to simulate social behaviour. It was only when this algorithm was simplified that its optimisation performance and abilities became evident [19]. PSO aims to mimic the movement and behaviour of organisms (e.g. birds, fish or insects) by what is described as swarm intelligence [20].

PSO generates a swarm of n individual particles and, as suggested by [19], each particle $i \in \{1, \ldots, n\}$ is given a random position vector within the search domain S ($x_i^k \in$ S) and a random velocity vector v_i^k. The initial position ($k = 0$) for each particle is

$$x_i^{k=0} = b_{\text{min}} + r\left(b_{\text{max}} - b_{\text{min}}\right), \tag{10}$$

where $r \in [0, 1]$ is a constant, and b_{min} and b_{max} are the lower and upper bounds of the design variable search domain, respectively. Similarly, the initial velocity is

$$v_i^{k=0} = U\left(-\left|b_{\text{max}} - b_{\text{min}}\right|, \left|b_{\text{max}} - b_{\text{min}}\right|\right), \tag{11}$$

where U is a random value within the uniformly distributed range $\left(-\left|b_{\text{max}} - b_{\text{min}}\right|, \left|b_{\text{max}} - b_{\text{min}}\right|\right)$. Each particle i is then evaluated by the objective func-

tion and their velocity v_i^k updated according to

$$v_i^k \leftarrow \omega v_i^{k-1} + \phi_{\mathrm{p}} r_{\mathrm{p}} \left(p_i^{k-1} - x_i^{k-1} \right) + \phi_{\mathrm{g}} r_{\mathrm{g}} \left(g^{k-1} - x_i^{k-1} \right). \tag{12}$$

ω is the inertia weight, a specific PSO parameter, and ϕ_{p} and ϕ_{g} are the cognitive and social parameters, respectively. The cognitive parameter (p) represents the best current position of the particle, and the social parameter (g) is the global best current position. $r_{\mathrm{p}} \in [0, 1]$ and $r_{\mathrm{g}} \in [0, 1]$ are constants. Subsequently, for each new iteration the position x_i^k is updated according to the previous position x_i^{k-1} and the updated velocity v_i^k, as

$$x_i^k = x_i^{k-1} + v_i^k. \tag{13}$$

4.1.2. Genetic algorithm

Genetic Algorithms (GA) are based on natural selection. These are based on the concept of Darwin's theory of evolution and were originally introduced by John Holland in 1960 and later further developed by his student David E. Goldberg [21]. As an evolutionary algorithms, a genetic algorithms imply the definition of an initial population of individual solutions. GA then relies on operators such as mutation, crossover and selection to improve the individual's solution at each generation. Due to their powerful and efficient operators, GA can be used to solve a broad range of optimisation problems, including highly nonlinear, stochastic and non-differentiable functions. This versatility led to their frequent implementation in design and engineering optimisation problems. GA are based on an iterative process where a given population of candidate solutions — the individuals — is evolved toward better solutions. Each individual has a specific set of properties — the chromosomes —, which are usually encoded in a binary array. This unique characteristic makes each individual vulnerable to changes in their chromosome, either through crossover with another individual or through other operators. Each individual's chromosome is then decoded and tested by a *fitness function* [22]. The fittest individuals, which represent the best solutions to the optimisation problem, ensure their legacy in further generations and have a higher probability of being selected for crossover.

4.1.3. Simulated annealing

Simulated Annealing (SA) mimics the procedure of annealing with the purpose of finding an approximation for a global minimum for a function with a large number of variables [23,24]. The inspiration for this algorithm comes from the annealing heat treatment in metallurgy. The concept of slow cooling implemented in the SA algorithm corresponds to a gradual decrease of the probability of keeping worse solutions as the search domain is explored. This notion of accepting worse solutions is part of the intrinsic nature of the SA algorithm and is a fundamental property of meta-heuristics as a whole as it allows a broader search of the global optimum solution and, ultimately, avoids local minima.

The optimisation process starts by initialising a random position x_{curr} for the variable within the upper and lower boundaries $U(b_{\mathrm{low}}, b_{\mathrm{up}})$. Tthe main iterative loop is then initiated and a neighbour position x_{prop} is defined based of the current position x_{curr} and a constant k that defines the maximum distance the neighbour can be from the current position. The quality of these two positions is evaluated and, if the neighbour corresponds to a better solution, the algorithm moves towards it by replacing the current with the neighbour position and solution. If the neighbour solution is worse than the current one, it may still be accepted depending on

the acceptance probability

$$P = \left\{ \exp\left[\frac{C(x_{\text{prop}}) - C(x_{\text{best}})}{T_{\text{curr}}} \right] \right\}^{-1}, \tag{14}$$

where T and C are the temperature and the objective function solutions, respectively. Subscripts prop and best designate the current proposed solution and the best solution, respectively. If the acceptance probability P is lower than $\texttt{random}[0, 1]$, the neighbour solution is accepted, even if worse. The last step of each iteration is to update the current temperature T_{curr}, which decreases depending on the cooling parameter α, updated for each iteration as $T^*_{\text{curr}} = \alpha T_{\text{curr}}$. As the temperature decreases, so does the probability of acceptance P. This results in a gradually lower probability of accepting worse solutions as the iterative process and convergence evolves.

5. Optimisation studies

The optimisation algorithms are divided into four main benchmarks, labelled B#, with increasing complexity, studying their behaviour for different design variables, objective functions and constraints, strategies and algorithm parameters, listed in Table 2. Optimisation algorithms are tested and compared according to different objective functions (weight and stress minimisation) and constraints (stress and weight upper limits), using both continuous (thickness) and discrete (material index) variables. BD5 is an extension of BD3, where the addition of a displacement constraint is tested as a way to control the solution and verification of the robustness of the implemented tools. This is expanded in Section 5.6), where the best algorithm is selected to perform a multi-objective optimisation as a combination of the previous analyses.

Table 2.: Characteristics of the optimisation studies, including variables, constraints and algorithms.

B#	**Minimisation of**	**Variables**	**Constraints**	**Algorithms**
BC1 (Sec. 5.1)	Weight (W)	thickness (t_{int}, continuous)	$\sigma_{\text{z, max}}$	PSO, GA, SA
BC2 (Sec. 5.2)	Rear stress (S)	thickness (t_{int}, continuous)	w_{max}	PSO, GA, SA
BD3 (Sec. 5.3)	Weight (W)	material (**x**, discrete)	$\sigma_{\text{z, max}}$	PSO, GA, SA
BD4 (Sec. 5.4)	Rear stress (S)	material (**x**, discrete)	w_{max}	PSO, GA, SA
BD5 (Sec. 5.5)	Weight (W)	material (**x**, discrete)	$\sigma_{\text{z, max}}$, $u_{\text{z,max}}$	PSO

5.1. BC1 (continuous variable — Weight minimisation with stress constraint)

Benchmark BC1 aim to minimise $F(t_{\text{int}}, r_{\text{g}}, \beta)$, which is the the plate's weight function $W(t_{\text{int}})$ augmented with the penalty function $P(t_{\text{int}}, r_{\text{g}}, \beta)$. The design variable is the interlayer thickness, t_{int}, with the search domain $[1, 50] \times 10^{-3}$ m. The constraint is the maximum

stress $\sigma_{\text{z, max}}$ at the rear surface. The formulation of this benchmark is then to

$$\begin{aligned} \text{minimise} \quad & F(t_{\text{int}}, r_{\text{g}}, \beta) = W(t_{\text{int}}) \\ & + P(t_{\text{int}}, r_{\text{g}}, \beta) \\ \text{subjected to} \quad & t_{\text{int, min}} \leq t_{\text{int}} \leq t_{\text{int, max}} \\ & g(x) = \sigma_{\text{z}} - \sigma_{\text{z, max}} \leq 0 \end{aligned} \tag{15}$$

with

$$W(t_{\text{int}}) = L \times B \left[t_{\text{front}} \rho_{\text{front}} + t_{\text{int}} \rho_{\text{int}} + t_{\text{rear}} \rho_{\text{rear}} \right], \tag{16}$$

where L and B are the length and breadth of each layer, respectively, and ρ is the density of the material. The penalty function, with $r_{\text{g}} = 2$ and $\beta = 2$ penalises the objective function if the measured peak stress $\sigma_{\text{z, max}}$ at the rear face of the target (see Section 2) is above $\sigma_{\text{z, max}} = 50$ MPa. For this benchmark, the front and rear plates, as well as the interlayer are made of aluminium, steel and Nylon-6, respectively, with the material properties listed in Table 3 [13]. The projectile impact velocity is 5 m/s.

Table 3.: Material properties.

ID	**Material**	**Density** [kg/m^3]	**Young's modulus** [MPa]	**Poisson's ratio** [$-$]	**Yield stress** [MPa]
0	Aluminium	2700	70×10^3	0.33	276
1	Nylon-6	1140	3×10^3	0.35	82
2	Steel	7850	200×10^3	0.25	350
3	EPDM	960	2.5	0.499	16.8
4	Cork	293	9×10^3	0.3	1
5	Aluminium Foam	410	103.08	0.05	1.24
6	Polycarbonate	1300	1.8×10^3	0.3182	63
7	Epoxy	1540	3.5×10^3	0.33	15
8	Titanium Ti-6Al-4V	4430	113.8×10^3	0.342	880

The optimisation variable t_{int} is continuous and limited to the search domain $[1, 50] \times 10^{-3}$ m, which is normalised to $[0, 1]$ to increase implementation flexibility. While not strictly necessary in this case, this is a useful approach when dealing with multiple problems, as is the case of the multi-objective optimisation procedures. This is generally applied to objective functions, which are normalised using a feature scaling approach [25] taking the first feasible solution, f_0, as reference, *i.e.*

$$f_{\text{norm}} = \frac{f_i}{f_0}, \tag{17}$$

where f_{norm} is the normalised solution.

5.1.1. BC1 — PSO

PSO parameters are sensitive to each particular optimisation problem [26] and require calibration. The parameters used for BC1 are listed in Table 4 [27–29].

Table 4.: Particle Swarm Optimisation (PSO) operational parameters [27–29].

Particles	**Iter.**	ω	ϕ_p	ϕ_g	v_i^0
15	30	0.5	1	2	$\mathtt{ran}(-1{,}1)\times 0.2$

The behaviour of the PSO algorithm for benchmark BC1 is represented in Figure 11. Immediately after the first iteration, the PSO converges to a solution which is only $\sim 0.0011\%$ different from the final and optimal solution of 40.186907 kg. It should be noted, however, that strictly meta-heuristic methods can not ensure a global optimum solution [30] and, thus, this solution might simply be a local optimum. The average and worst solution of each population at every iteration is also taken into consideration, as shown in Figure 11a. From these normalised results, it is clear that this is a stochastic but also evolutionary algorithm. In the first few iterations, a decent discrepancy between the best, average and worst solutions of a population becomes noticeable. This is due to the random positions and velocities attributed to the initial individuals, highlighting the stochastic nature. However, due to social and cognitive parameters, the position of each particle tends to evolve. This is illustrated by the monotonous convergence of both the average and worst solutions to the best global solution for increasing number of iterations.

The evolution of particle positions, culminating in solutions that converge to the overall best solution is shown in Figure 11b. The initial solutions are intentionally dispersed to cover the search space. This strategy reduces the possibility of converging to a local optimum. Nevertheless, if parameters such as the number of particles are not high enough, the objective function is likely to converge to a local optimum. Besides, due its stochastic nature, the results of several attempts using the same PSO parameters may differ even if only slightly, as will be discussed later.

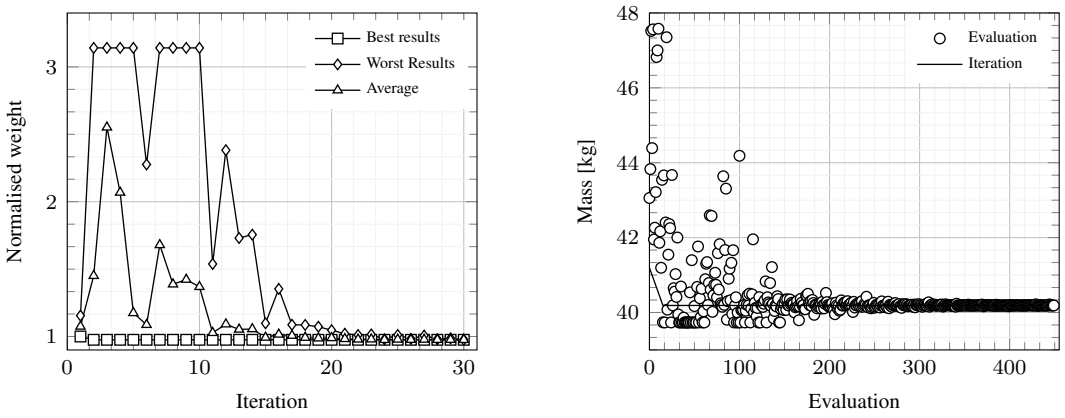
5.1.2. BC1 — GA

The genetic operators adopted in the first benchmark, listed in Table 5, were adjusted empirically [7,8,31,32].

Table 5.: Genetic Algorithm (GA) operational parameters for benchmark BC1 [7,8,31,32].

Population	20
Individuals	30
Crossover	80%
Mutation	20%
Chromosome Length	32

The evolution of the objective function is shown in Figure 12a, by comparing the average, worst and best solution for each evaluation. This emphasises the influence of the penalty func-

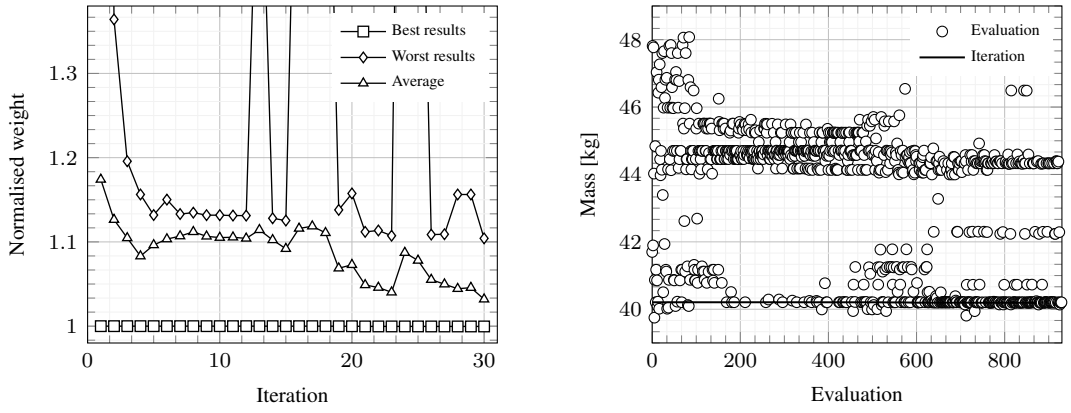


(a) (b)

Figure 11.: (a) Evolution of the best, worst and mean values of the objective function with the number of iterations, for the PSO in BC1. (b) Comparison of the solution at each evaluation without penalisation with the overall best solution at each iteration, for the PSO in BC1.

tion as is noticeable that some evaluations surpass the constraint threshold and are penalised. Both the worst and average curves show sporadic evaluations that correspond to abnormal high results, indicating that they were penalised. Nevertheless, the evolution of the average and worst solution curves towards the overall best solution is considered adequate. Comparing with the results from the PSO algorithm in the previous section, however, the GA algorithm shows a slower rate of convergence to the overall best solution. This method also shows a wider dispersion of solutions over its population, with varying levels of performance of each individual.

The underlying structure of the GA and its evolution are evident in the results in Figure 12b. The final solution and the rate at which the objective function converges are heavily dependent on the first individual's genome. Thus, the number of generations required to converge to the optimum solution and the optimum solution itself may diverge every time the GA algorithm is tested.



(a) (b)

Figure 12.: (a) Evolution of the best, worst and mean values of the objective function with the number of iterations, for the GA in BC1. (b) Comparison of the solution at each evaluation without penalisation with the overall best solution at each iteration, for the GA in BC1.

5.1.3. BC1 — SA

Although belonging to the same field of evolutionary algorithms as the GA and PSO, the SA has a peculiar search strategy (*vd.* Sec. 4.1.3). The main operational parameter, the temperature, controls the progression of the objective function. This parameter depends on the cooling factor, which directly influences how quickly the objective function converges. If this parameter is low, it may lead to a premature convergence that results in stagnation on a local optimum. If instead the cooling parameter is high, the algorithm will need more time to converge to the optimum solution and, thus becoming computationally more expensive. The parameters used in this benchmark for the SA algorithm are listed in Table 6 [33–35].

Table 6.: Simulated Annealing (SA) operational parameters [33–35].

Iterations	40
Evaluations per iteration	10
T	1000
α	0.65
k_{step}	0.1

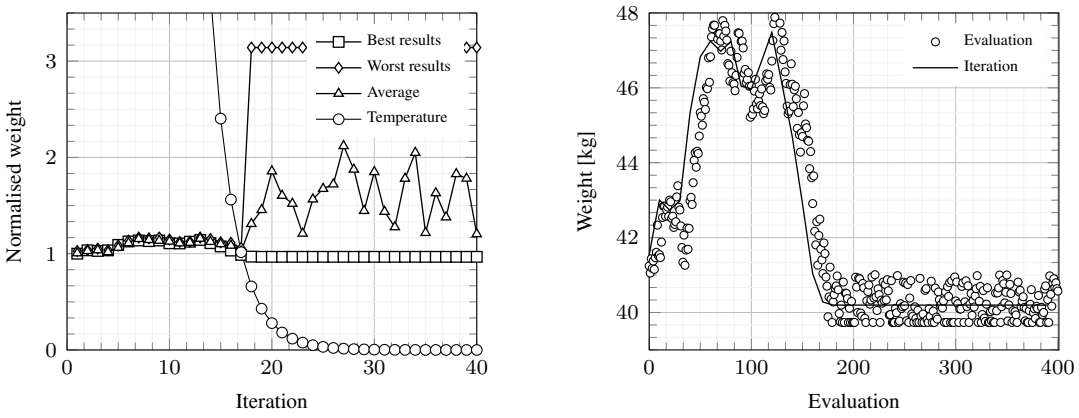
A distinct evolution strategy is visible from the evolution of the objective function shown in Figure 13, when comparing to the more linear results from the PSO and GA. The objective function suggests to be evolving to continuously worst solutions until iteration 13, after which it starts to converge to the optimal solution at iteration 19. This peculiar search strategy is intrinsic to SA. As the temperature is still high, there is a correlated probability (*vd.* Sec. 4.1.3) for the current evaluated solution to be accepted, even if that solution is worse than the previous one. As the temperature decreases (due to the cooling parameter (α) the acceptance probability also decreases resulting in the acceptance of new better solutions.

The evolution of best, average and worst solution for each iteration, as well as the curve of the temperature evolution are shown in Figure 13a. The average and worst curves increase drastically after iteration 17. This happens simultaneously with the temperature curve reaching a sufficiently low threshold (set at $T < 1$), allowing the SA formulation to reduce the probability of accepting worse solutions. After this, the SA algorithm will only accept better solutions and the position of each evaluation will steadily approach the optimum position. Because the optimum position leads to a solution that is at the limit of being penalised, the chance of the algorithm generating a position beyond this limit is high. The result is then an immediate increase of the worst and average curves.

The comparison of the evaluations to the best solution in each iteration is plotted in Figure 13b, showing that evaluations have a reduced spread over each iteration. This is caused by the k_{step} parameter controlling the maximum and minimum ranges by which the new position can exceed the previous. Consequently, results obtained from the SA algorithm do not intrinsically evolve towards a better solution based on previous solutions, as is the case with the GA and PSO. This also supports the fact that, although the SA is an evolutionary algorithm, it is more stochastic in nature than the PSO and GA.

5.1.4. BC1 — Summary

The output of each algorithm is listed in Table 7 showing that the algorithm with the best performance in this benchmark is the PSO as it was the one that reached the lowest weight and interlayer thickness of 40.18691 kg and 0.00365 m, respectively. Although the SA reached



(a) (b)

Figure 13.: (a) Evolution of the best, worst and mean values of the objective function with the number of iterations, for the SA in BC1. (b) Comparison of the solution at each evaluation without penalisation with the overall best solution at each iteration, for the SA in BC1.

a local optimum in 188 evaluations the PSO reached a better solution than the SA at iteration 2 (15 evaluations), corresponding to 40.187359 kg. The GA produced the second best solution, and the best average and standard deviation. These results are, however, relative because the PSO had a considerably bad first iteration (41.194223 kg). Not considering this first iteration, the average and standard deviation drops to values lower than those of the GA, at 40.187257 kg and 0.000135652, respectively. In order to investigate how the stochastic nature the algorithms affects the results, this benchmark was executed five times, with the corresponding results listed in Table 8. It can be seen that for every execution the results from the PSO algorithm are the lowest, rendering this algorithm the most efficient.

Table 7.: BC1 final results for the different algorithms.

	BC1 PSO	**BC1 GA**	**BC1 SA**
Best Solution [kg]	40.18691	40.19316	40.19709
Average [kg]	40.22198	40.20398	44.49111
Standard Deviation	0.183737	0.005402	2.464782
Best Variable [m]	0.00365	0.003688	0.00371
Evaluations until best solution	428	807	188

5.2. BC2 (Continuous variable — Rear stress minimisation with weight constraint)

The objective function, $F(t_{\text{int}}, r_{\text{g}}, \beta)$, for this second benchmark is used to minimise the stress on the rear face of the plate, $S(t_{\text{int}})$. The design variable is again the interlayer thickness t_{int}

Table 8.: BC1 final results after running five times for each algorithm.

Run	**BC1 PSO**	**BC1 GA**	**BC1 SA**
1	40.186896	40.193157	40.19709
2	40.186907	40.201195	40.19290
3	40.186896	40.266657	40.18923
4	40.186932	40.265890	40.20305
5	40.186897	40.260264	40.18873
Average [kg]	40.1869	40.2374	40.1942

with the same limit constraints $[1, 50] \times 10^{-3}$ m. The compact formulation is

$$\begin{aligned} \text{minimise} \quad & F(t_{\text{int}}, r_{\text{g}}, \beta) = S(t) + P(t, r_{\text{g}}, \beta), \\ \text{subjected to} \quad & t_{\text{min}} \leq t \leq t_{\text{max}}, \\ & g(x) = w - w_{\text{max}} \leq 0\,. \end{aligned} \tag{18}$$

The maximum allowed weight of the plate is $w_{\text{max}} = 40$ kg. Accordingly, $r_g = 1 \times 10^6$ and $\beta = 2$ were used in the penalty function. Results for $S(t_{\text{int}})$ are obtained with a black box approach (provided by Abaqus).

This benchmark uses the same methodologies, strategies and operational parameters as the previous benchmark. The behaviour and evolution of each algorithm throughout the iterations are similar to the previous benchmark. The performance of each algorithm is illustrated in Table 9. The best final solution, average, standard deviation and final position we all produced by the PSO. The SA algorithm found a local minimum, although higher than that of the PSO, at evaluation 243. However, the performance of the SA demonstrated a low level of accuracy as the nature of this algorithm relies on an explicitly random parameter.

Table 9.: BC2 final results for the different algorithms.

	BC2 PSO	**BC2 GA**	**BC2 SA**
Best Solution [MPa]	63.392584	63.486055	64.433575
Average [MPa]	67.489943	85.609726	72.319194
Standard Deviation [MPa]	18.564183	32.971243	41.626055
Best Variable [m]	0.0025	0.002497	0.00247
Evaluations until best solution	434	526	243

5.3. BD3 (Discrete variable — Weight minimisation with stress constraint)

Benchmark BD3 tests how the algorithms perform when the variable of study has a discrete nature. Hence, the objective is to minimise the total weight of the plate, similarly to the first benchmark, while varying the material properties of each layer. As such, the variables of study

correspond to the indexes of a matrix (database) that contains the mechanical properties of 9 materials (see Table 3), This optimisation benchmark takes the form

$$\begin{aligned} \text{minimise} \quad & F(\mathbf{x}, r_{\mathrm{g}}, \beta) = W(\mathbf{x}) \\ & +P(\mathbf{x}, r_{\mathrm{g}}, \beta), \\ \text{subjected to} \quad & x_i \in \{0, 1, \ldots, a\}, \\ & i = \{\text{front}, \text{interlayer}, \text{rear}\}, \\ & g(\mathbf{x}) = \sigma_{\mathrm{z}} - \sigma_{\text{z,max}} \leq 0. \end{aligned} \tag{19}$$

where

$$\begin{aligned} W(\mathbf{x}) &= L \times B \left[t_{\text{front}} \rho_{\text{front}} + t_{\text{int}} \rho_{\text{int}} + t_{\text{rear}} \rho_{\text{rear}} \right] \\ &= 0.07^2 \left[0.025 (\rho_{\text{front}} + \rho_{\text{rear}}) \right. \\ &\quad \left. + 0.002 \times \rho_{\text{int}} \right] . \end{aligned} \tag{20}$$

A penalty function with $r_g = 2$ and $\beta = 2$ is implemented to penalise the objective function every time the measured peak stress $\sigma_{\text{z,max}}$ at the rear face of the plate (Section 2) surpasses $\sigma_{\text{z,max}} = 5$ MPa. The variables in this benchmark, $\mathbf{x}_i$, are discrete ($\{0, 1, \ldots, 8\}$). The code implementation used in this work uses a continuous search domain within $[0, 1]$ and converts the solutions back into discrete values for the evaluations.

5.3.1. BD3 — PSO

As the study variables are discrete, the PSO operational parameters are adapted as shown in Table 10. The range for the initial velocity of each particle, v_i^0, was increased as well as its weight ω to ensure enough inertia to escape from a potential local minimum.

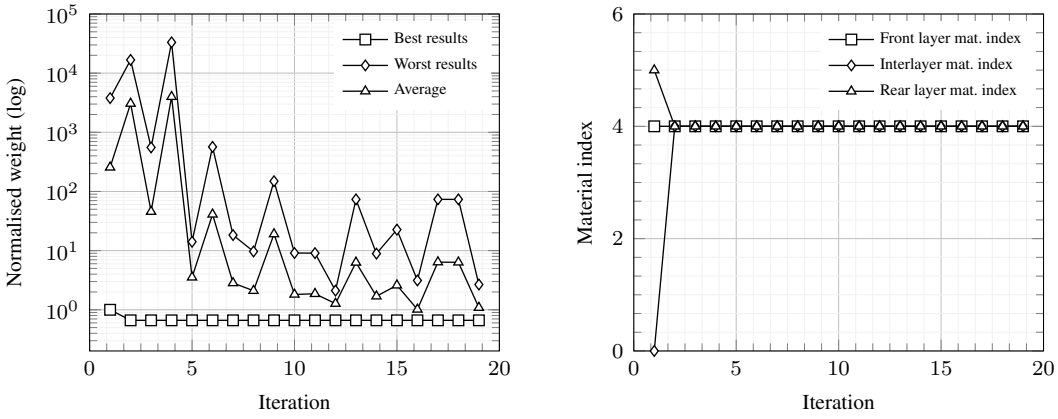
Table 10.: PSO operational parameters for BD3.

Particles	**Iter.**	ω	ϕ_p	ϕ_g	v_i^0
15	20	0.7	1	2	`ran`$(-1,1)\times 0.8$

A fast convergence rate from the objective function is visible from the results in Figure 14a as it converged to the final optimum solution at the second iteration. A similar trend is followed by the average and worst curves. The material properties of the three layers converge at the second iteration as shown in Figure 14b. The result is the index 4 for the three layers, which corresponds to cork. Although in an optimisation paradigm the PSO performed well, within an engineering view this result is not satisfactory and demonstrates a premature choice and implementation of constraints. As such, the same problem is addressed later in Section 5.5 with an additional constraint associated with the projectile displacement.

5.3.2. BD3 — GA

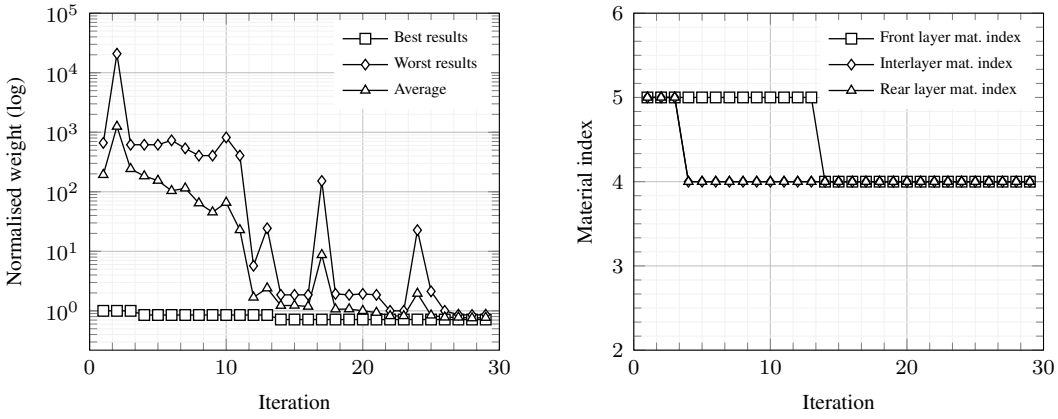
As the number of variables increases from 1 to 3, so does the chromosome length (from 32 to 64). This ensures a good performance of the GA. The objective function and both the average



(a) (b)

Figure 14.: (a) Evolution of the best, worst and mean values of the objective function with the number of iterations, for the PSO in BD3. (b) Combination of material indexes for the PSO in BD3.

and worst solutions converge at a good rate towards the optimum solution (*vd.* Fig. 15). The final position is the same as the PSO's, which is a cork plate.



(a) (b)

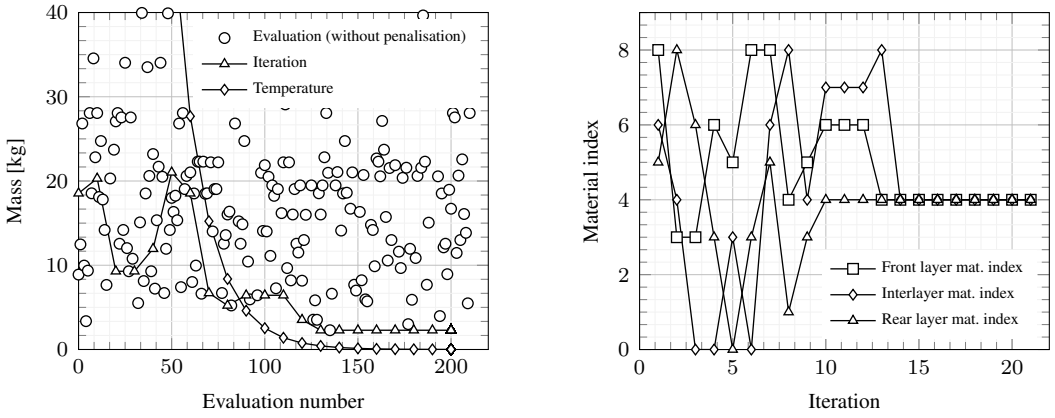
Figure 15.: (a) Evolution of the best, worst and mean values of the objective function against the number of iterations, for the GA in BD3. (b) Combination of material indexes for the GA in BD3.

5.3.3. BD3 — SA

The operation parameters used in this case are the same as in the previous benchmarks, with the exception of the cooling parameter, which is now $\alpha = 0.55$. This parameter was decreased due to the excessive number of evaluations it needed to converge.

Similarly to how SA behaved in the previous benchmarks, a high rate of randomness in each evaluation is noticeable (see Figure 16a). Nevertheless, the objective function shows a consistent evolution towards the final optimal solution, stopping to improve when the temperature reaches values close to zero. This behaviour is similar to that discussed in previous

benchmarks. The material's indexes for each layer appear to have a random evolution until they simultaneously converge, as shown in Figure 16b. The final solution is the same as that of the previous algorithms.



(a) (b)

Figure 16.: (a) Comparison of the solution at each evaluation without penalisation with the overall best solution at each iteration, for the SA in BD3. (b) Combination of material indexes for the GA in BD3.

5.3.4. BD3 — Summary

All three algorithms reached the same combination of design variables, as can be seen in Table 11. Consequently, the efficiency of the algorithms has to be assessed by comparing how quickly they reach the optimum solution. The PSO algorithm has the lowest average, standard deviation and, by far, number of evaluations needed to find the best solution and is thus the most efficient of the three algorithms.

Table 11.: BD3 final results for the different algorithms.

	BD3 PSO	**BD3 GA**	**BD3 SA**
Best solution [kg]	2.2854	2.2854	2.2854
Average [kg]	2.3465	2.5311	7.7129
Standard deviation	0.2592	0.3036	6.3928
Final indexes	[4,4,4]	[4,4,4]	[4,4,4]
Evaluations until best solution	21	291	135

5.4. BD4 (Discrete variable — Rear stress minimisation with weight constraint)

The aim of benchmark BD4 is to compare the efficiency of the algorithms when required to minimise the stress at the rear face of the plate, similarly to benchmark BD2, but using the same discrete variables, material mechanical properties and strategies as in benchmark BD3.

The formulation for this benchmark is

$$\begin{aligned} \text{minimise} \quad & F(\mathbf{x}, r_{\mathrm{g}}, \beta) = S(\mathbf{x}) + P(\mathbf{x}, r_{\mathrm{g}}, \beta), \\ \text{subjected to} \quad & x_i \in \{0, 1, \ldots, a\}, \\ & i = \{\text{front}, \text{interlayer}, \text{rear}\}, \\ & g(\mathbf{x}) = w - w_{\text{max}} \leq 0. \end{aligned} \tag{21}$$

The algorithm that resulted in the optimal layer combination (i.e. the one with the lowest stress at the rear) is once again PSO, as shown in Table 12. PSO also has the lowest average and the second lowest number of evaluations required to find the optimal solution. The resultant combination of materials were aluminium foam as the top and rear layer and titanium as the interlayer. The GA, besides having the worst solution and average of the three algorithms, could not avoid a local minimum. Its final configuration was a front, interlayer and rear layer made of, respectively, nylon, cork and aluminium foam. The SA has had the second best solution and average, with a final configuration of aluminium foam as the front and rear layer and aluminium as the interlayer.

Table 12.: BD4 final results for the different algorithms.

	BD4 PSO	**BD4 GA**	**BD4 SA**
Best evaluation [Pa]	640657.5691	986006.602	716679.511
Average [Pa]	734298.7146	986006.602	842247.4266
Standard deviation	168602.5592	0	165028.5726
Final indexes	[5,8,5]	[1,4,5]	[5,0,5]
Evaluations until best solution	84	8	120

5.5. BD5 — Alternative approach (Discrete variable — Weight minimisation with stress constraint)

The previous benchmarks were proposed to analyse the three tested algorithms and select the most efficient and robust one, aiming to do so in both a strict optimisation approach and with an engineering application in mind. The focus was also to test and train different algorithm programming strategies and different problem formulations to perform the multi-objective optimisation in Section 5.6. The results from each benchmark were analysed by emphasising optimisation performance rather than engineering integrity. The most evident example is benchmark BD3 where all three algorithms pointed to the same optimal solution: a plate entirely made of cork. According to the initial formulation and constraints, the three algorithms reached the global minimum of the problem. From an engineering standpoint, however, a plate entirely made of cork might not be an efficient protection against ballistic impacts, even if for impacts generating elastic stresses only. This solution was selected, however, because cork has a low yield strength and density (see Table 3). The low yield strength resulted in a premature material plastification and, as a consequence, a significant fraction of the impact energy is absorbed in this processes, resulting in a lower average stress at the rear face of the plate. Consequently, the rear stress was never penalised during the optimisation process. Cork also has the lowest density of all materials tested, which resulted in being the best solution

for minimising the weight of the plate. To avoid such premature solutions (*i.e.* from an engineering standpoint) a new constraint was added to penalise solutions where the maximum displacement of the projectile surpassed a predefined threshold. By limiting the projectile's maximum displacement, the algorithm gained indirect control over solutions that point towards materials that endured excessive deformation.

In order to perform a proper validation, the PSO algorithm was used in a new version of benchmark BD3, now formulated as

$$\begin{aligned} \text{minimise} \quad & F(\mathbf{x}, r_{\mathrm{g}}, \beta, r_{\mathrm{h}}, \gamma) = W(\mathbf{x}) \\ & +P(\mathbf{x}, r_{\mathrm{g}}, \beta, r_{\mathrm{h}}, \gamma), \\ \text{subjected to} \quad & x_i \in \{0, 1, \ldots, a\}, \\ & i = \{\text{front}, \text{interlayer}, \text{rear}\}, \\ & g(\mathbf{x}) = \sigma_{\mathrm{z}} - \sigma_{\mathrm{z,max}} \leq 0, \\ & h_{\mathrm{k}}(\mathbf{x}) = u_{\mathrm{z}} - u_{\mathrm{z,max}} \leq 0, \end{aligned} \tag{22}$$

where

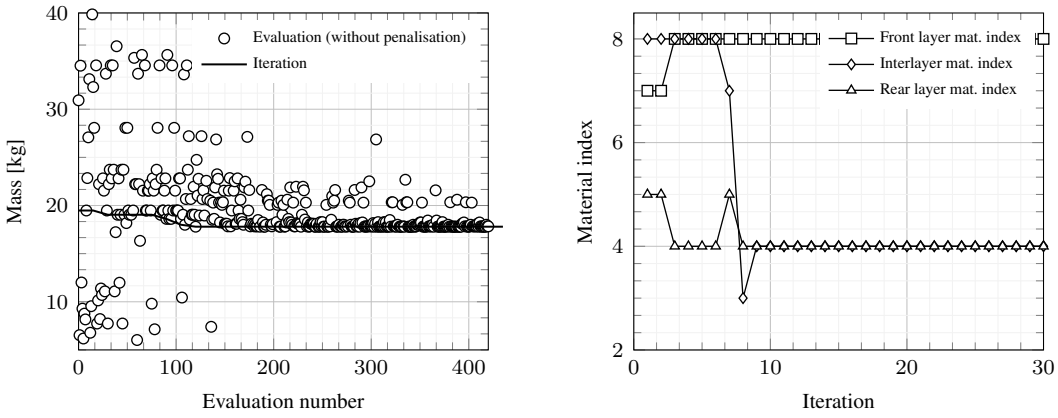
$$\begin{aligned} W(\mathbf{x}) &= L \times B \left[t_{\text{front}} \rho_{\text{front}} \right. \\ &\quad \left. + t_{\text{int}} \rho_{\text{int}} + t_{\text{rear}} \rho_{\text{rear}} \right] \\ &= 0.07^2 \left[0.025 (\rho_{\text{front}} + \rho_{\text{rear}}) \right. \\ &\quad \left. + 0.002 \times \rho_{\text{int}} \right] \end{aligned} \tag{23}$$

and u and σ are the projectile displacement and stress at the rear surface, respectively. The penalty function parameters are $r_g = 2$, $\beta = 2$, $r_h = 1 \times 10^6$, $\gamma = 2$, $\sigma_{\text{z,max}} = 10$ MPa and $u_{\text{z,max}} = 1.5 \times 10^{-4}$ m. The PSO operational parameters used in the scope of this optimisation process are the same as in benchmark BD3, also listed in Table 10.

The particles evolve towards the best solution starting from solutions at the upper and lower extremes, as shown in Figure 17a. Comparing with the evolutions from benchmark BD3 (*vd.* Figs. 17, 15 and 16), where solutions start converging only from inside the admissible domain, it is possible to see the influence of the new displacement constraint (Equation 23). The penalty function is now active and the PSO need to manage also have particles in violation of this constraint, rendering the previous solution invalid. The final solution is a front layer of titanium, with interlayer and rear layer made of cork, as shown in Figure 17b. This solution has a projectile displacement of 0.78×10^{-4} m, which is significantly lower than the predefined limit.

5.6. Multi-objective optimisation

The procedures developed in previous sections are now incorporated in one multi-objective optimisation problem. PSO is used as it was demonstrated to be the most efficient algorithm. The multi-objective optimisation problem in this section is a combination of the previous



(a) (b)

Figure 17.: (a) Comparison of the solution at each evaluation without penalisation with the overall best solution at each iteration, for the alternative approach to benchmark BD3. (b) Combination of material indexes for the alternative approach to benchmark BD3.

individual single-objective optimisation problems. Its compact formulation can be defined as

$$\begin{aligned} \text{minimise} \quad & F(\mathbf{x}) = (y) f_1(\mathbf{x}) + (1-y) f_2(\mathbf{x}), \\ \text{subjected to} \quad & x_i \in \{0, 1, \ldots, a\}, \\ & i = \{\text{front}, \text{interlayer}, \text{rear}\}, \\ & t_{\text{min}} \leq t \leq t_{\text{max}}, \\ & g(\mathbf{x}) = \sigma_{\text{z}} - \sigma_{\text{z,max}} \leq 0, \\ & h(\mathbf{x}) = W - W_{\text{max}} \leq 0, \\ & d(\mathbf{x}) = u_{\text{z}} - u_{\text{z,max}} \leq 0, \end{aligned} \tag{24}$$

where t is the interlayer thickness in $[1, 25] \times 10^{-3}$ and x_i is the layer's material index from Table 3. $\mathbf{x}$ is a vector containing the four design variables, $\mathbf{x} = [x_i, t]$. $g(\mathbf{x})$ is the constraint function related to the maximum stress at the rear face of the plate, $h(\mathbf{x})$ is the plate's weight constraint function and $d(\mathbf{x})$ is the projectile's displacement constraint. Five weights were used in the scope of this multi-objective-optimisation, $y \in \{0, 0.25, 0.5, 0.75, 1\}$, corresponding to five different and individual runs of the multi-objective function. The single-objective function f_1 aims to minimise the weight of the plate and f_2 its rear stress, as

$$F_1(\mathbf{x}) = W(\mathbf{x}) + P_1(\mathbf{x}, r_{\text{g}}, \beta, r_{\text{h}}, \gamma), \tag{25}$$

$$F_2(\mathbf{x}) = S(\mathbf{x}) + P_2(\mathbf{x}, r_{\text{g}}, \beta, r_{\text{h}}, \gamma). \tag{26}$$

Both f_1 and f_2 have a penalty function assigned, P_1 and P_2, respectively, *i.e.*

$$P_1(\mathbf{x}, r_\mathrm{g}, \beta, r_\mathrm{f}, \gamma) = r_\mathrm{g}\left[\max\left\{0, g(x)\right\}\right]^\beta + r_\mathrm{d}\left[\max\left\{0, d(x)\right\}\right]^\gamma, \tag{27}$$

$$P_2(\mathbf{x}, r_\mathrm{h}, \beta, r_\mathrm{f}, \gamma) = r_\mathrm{h}\left[\max\left\{0, h(x)\right\}\right]^\beta + r_\mathrm{d}\left[\max\left\{0, d(x)\right\}\right]^\gamma. \tag{28}$$

The maximum weight (mass) is $W_\mathrm{max} = 30$ kg, the maximum admissible rear stress is $\sigma_\mathrm{z,max} = 40$ MPa and the maximum admissible projectile displacement is $u_\mathrm{z,max} = 1.5 \times 10^{-4}$ m. For the penalty functions described in Equations 28 and 29, the following parameters were used: $r_\mathrm{g} = r_\mathrm{h} = r_\mathrm{d} = 10$, $\beta = \gamma = 3$.

The multi-objective function in Equation 24 combines two different single-objective functions: f_1 (Equation 25), which evaluates the weight of the plate and f_2 (Equation 26), which evaluates the stress at the rear face. Both f_1 and f_2 have different units and significantly different orders of magnitude, leading to common but robust approach regarding multiple objectives [36], using normalised objective functions as

$$f_q^\mathrm{norm} = \frac{f_q(\mathbf{x}) - f_q^\circ}{f_q^\mathrm{max} - f_q^\circ}, \quad q = \{1, 2\}, \tag{29}$$

where f_q° is the utopia point and f_q^max is the maximum or worst point for function q. For both functions f_1 and f_2, neither the utopia nor the maximum points are known or, at least, not immediate to obtain. Thus, one approach to determine these points is to minimise each function individually. Depending on the accuracy of f_q^max and f_q°, f_q^norm typically has values between 0 and 1. By following this approach, the normalisation parameters to be used for the scope of the multi-objective optimisation are listed in Table 13.

Table 13.: Normalisation parameters used in the multi-objective optimisation problem.

f_1 [kg]		f_2 [MPa]	
f_1°	f_1^max	f_2°	f_2^max
11	22	1.2	8

The PSO algorithm used in the current multi-objective problem was adapted in such way that it differentiates the nature of each variable and assigns different parameters accordingly. By doing this, the PSO performance verified in Section 5 is ensured and is independent of the number and nature of the variables of study. The operational parameters used for the current multi-objective problem are listed in Table 14, where the subscripts ω_c and ω_d are the PSO weight parameters used in the case of a continuous or discrete variable, respectively. Parameters $v_{i,\mathrm{c}}^0$ and $v_{i,\mathrm{d}}^0$ are the assigned particle initial velocity for a continuous and discrete variable, respectively.

The results obtained from a multi-objective optimisation (vd. 15) are intrinsically more complex and require a more meticulous approach than the results from a typical single-objective optimisation. In order to obtain results, the multi-objective optimisation process

Table 14.: PSO operational parameters used in the multi-objective optimisation problem.

Particles	**Iterations**	ω_c	ω_d	ϕ_p	ϕ_g	$v^0_{i,c}$	$v^0_{i,d}$
15	30	0.5	0.7	1	2	`random`(-1,1)×0.2	`random`(-1,1)×0.8

consisted in five consecutive runs, in which the weight y from Equation 24 assume different values ($y \in \{0, 0.25, 0.5, 0.75, 1\}$).

The two-dimensional Pareto curve obtained by plotting the optimum results of f_1 and f_2 for every weight y is shown in Figure 18. From Equation 24, when $y = 0$, the global objective function neglects the contribution of f_1 and only considers the results from f_2. This point corresponds to the highest possible value for f_1, which is 1.5705, and the lowest value for f_2, 0.0073. As y increases and gets closer to 1, the opposite is observed: f_2 is expected to gradually increase as f_1 decreases. Finally, when $y = 1$, the global objective function neglects the effect of f_2, which rises to its maximum value in the Pareto curve whilst f_1 is minimised. The results obtained from the multi-objective optimisation procedure for each of the weights are shown in Figure 18. As expected, the global objective function optimum, f, is lower at the extremes ($y = 0$ and $y = 1$). A multi-objective optimisation has an infinite number of solutions. Therefore, the Pareto's curve illustrated in Figure 18 is the ultimate solution for such a problem as it indicates a region that contains all the optimal solutions. This region is known as the Pareto optimal region, which corresponds to the region within the utopia line and the Pareto's curve itself [16].

Table 15.: Final results obtained from the multi-objective optimisation for each weight y.

Weight of f_1 (y)			**Design variables**		**Results**
$y = 0.00$	f	0.0027	Material Index $[x_{\text{front}}, x_{\text{interlayer}}, x_{\text{rear}}]$	$[0, 8, 5]$	$W = 20.57$ kg
	f_1	0.8700			$\sigma_z = 1.22$ MPa
	f_2	0.0027	Thickness, t [m]	0.0134	$u_z = 0.057$ mm
$y = 0.25$	f	0.1856	Material Index $[x_{\text{front}}, x_{\text{interlayer}}, x_{\text{rear}}]$	$[8, 4, 5]$	$W = 18.194$ kg
	f_1	0.6540			$\sigma_z = 1.40$ MPa
	f_2	0.0294	Thickness, t [m]	0.0100	$u_z = 0.096$ mm
$y = 0.50$	f	0.0564	Material Index $[x_{\text{front}}, x_{\text{interlayer}}, x_{\text{rear}}]$	$[0, 4, 5]$	$W = 11.71$ kg
	f_1	0.0642			$\sigma_z = 1.53$ MPa
	f_2	0.0486	Thickness, t [m]	0.0010	$u_z = 0.143$ mm
$y = 0.75$	f	0.0849	Material Index $[x_{\text{front}}, x_{\text{interlayer}}, x_{\text{rear}}]$	$[0, 0, 5]$	$W = 12.07$ kg
	f_1	0.0970			$\sigma_z = 1.53$ MPa
	f_2	0.0485	Thickness, t [m]	0.0010	$u_z = 0.135$ mm
$y = 1.00$	f	0.0243	Material Index $[x_{\text{front}}, x_{\text{interlayer}}, x_{\text{rear}}]$	$[0, 4, 4]$	$W = 11.27$ kg
	f_1	0.0243			$\sigma_z = 7.91$ MPa
	f_2	0.9877	Thickness, t [m]	0.0010	$u_z = 0.113$ mm

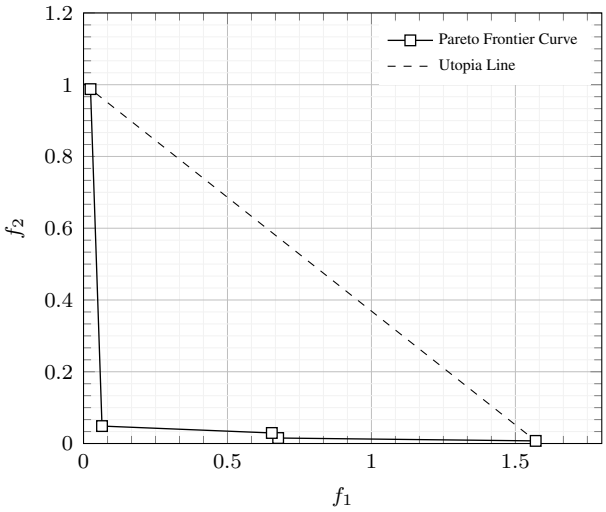


Figure 18.: Pareto curve of the multi-objective optimisation.

6. Final remarks

This work analyses the behaviour of three metaheuristic optimisation algorithms over four benchmarks and a multi-objective approach, applied to a generic protective armour plate design problem. This is not a common application for this type of methodology, namely due to of the high computational cost of each evaluation. Abaqus Python scripting and design parametrisation techniques are used to provide the necessary automation and efficiency for the iterative process.

The paper includes a set of model validation studies, focuses on the mechanical response of the interlayer in an energy absorption and elastic stress wave propagation problem in multilayer/multimaterial armour systems. The proposed optimisation procedures and approaches, including algorithms, methods and strategies, define a platform in which to access both the performance and the behaviour of each metaheuristic on this type of application. All algorithms reach comparable optimal solutions, but the provided detailed analyses of each benchmark led to the choice of Particle Swarm Optimisation (PSO) over the Genetic Algorithm (GA) or the Simulated Annealing (SA) methods as the best of the proposed implementations.

The final multi-objective optimisation problem combined several features from the benchmark, namely continuous and discrete variables, and two conflicting weight minimisation and stress minimisation objectives. Taking as a reference a balanced weight objective of $y = 0.5$, the optimal solution pointed to an interlayer thickness of 1 mm and a system where the front, interlayer and rear layers are made of aluminium, cork and aluminium foam, respectively. The developed models successfully achieved the study purposes both in the simulation of generic ballistic impacts and in the quality of the optimised solutions. This demonstrated the high potential for this type of optimisation method on terminal ballistic applications, serving as a standpoint for further studies into higher energy impacts and material non-linearities.

Acknowledgements

This work was also supported by the projects UIDB/00481/2020 and UIDP/00481/2020 – FCT – Fundação para a Ciência e a Tecnologia; and CENTRO-01-0145-FEDER-022083

– Centro Portugal Regional Operational Programme (Centro2020), under the PORTUGAL 2020 Partnership Agreement, through the European Regional Development Fund.

References

[1] P. J. Hazell, *Materials, Theory, and Design*. CRC Press, 1st ed., 2015.

[2] J. G. Hetherington, "The optimization of two component composite armours," *International Journal of Impact Engineering*, vol. 12, no. 3, pp. 409–414, 1992.

[3] M. L. Wilkins, "Mechanics of penetration and perforation," *International Journal of Engineering Science*, vol. 16, no. 11, pp. 793–807, 1978.

[4] M. F. Ashby, "Hybrids to fill holes in material property space," *Philosophical Magazine*, vol. 85, pp. 3235–3257, 2005.

[5] Y. Gu and V. F. Nesterenko, "Design and ballistic testing of Ti-6Al-4V matrix composites," *Journal of Composite Materials*, vol. 41, no. 19, pp. 2313–2323, 2007.

[6] M. Park, J. Yoo, and D. T. Chung, "An optimization of a multi-layered plate under ballistic impact," *International Journal of Solids and Structures*, vol. 42, no. 1, pp. 123–137, 2005.

[7] M. Yong, B. G. Falzon, and L. Iannucci, "On the application of genetic algorithms for optimising composites against impact loading," *International Journal of Impact Engineering*, vol. 35, no. 11, pp. 1293–1302, 2008.

[8] M. Yong, L. Iannucci, and B. G. Falzon, "Efficient modelling and optimisation of hybrid multi-layered plates subject to ballistic impact," *International Journal of Impact Engineering*, vol. 37, no. 6, pp. 605–624, 2010.

[9] R. Davies, "Stress waves in solids," *Science and Technology in Industry*, vol. 7, pp. 203–209, 1956.

[10] K. F. Graff, *Book: Wave Motion in Elastic Solids*. New York, United States: Dover Publications Inc., new editio ed., 1991.

[11] N. Y. Gnedin, V. A. Semenov, and A. V. Kravtsov, "Enforcing the Courant–Friedrichs–Lewy condition in explicitly conservative local time stepping schemes," *Journal of Computational Physics*, vol. 359, pp. 93–105, 2018.

[12] J. Plešek, R. Kolman, and D. Gabriel, "Estimation of the critical time step for explicit integration," *Engineering mechanics*, pp. 1001–1010, 2012.

[13] Matweb, "Online Materials Information Resource," http://www.matweb.com/, 2019. (last accessed September 2019).

[14] X. Luo, A. J. Aref, and G. F. Dargush, "Optimal Design of Bundled Layered Elastic Stress Wave Attenuators," *Journal of Computing in Civil Engineering*, vol. 26, no. 3, pp. 387–395, 2011.

[15] S. Rao, *Engineering optimization: theory and practice*. John Wiley & Sons, 4th ed., 2009.

[16] A.Andrade-Campos, J. Dias-de-Oliveira, and J. Pinho-da-Cruz, *Otimização Não-Linear em Engenharia*. ETEP, 2015.

[17] J. Car, "An Introduction to Genetic Algorithms.," *Artificial Life*, vol. 3, no. 1, pp. 63–65, 2014.

[18] J. Kennedy and R. Eberhart, "Particle Swarm Optimization James," *IEEE international conference on neural networks,*, pp. 1942–1948, 1995.

[19] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360)*, pp. 69–73, IEEE, 1998.

[20] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001.

[21] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998.

[22] M. Mitchell, "Genetic algorithms: An overview," *Complexity*, vol. 1, no. 1, pp. 31–39, 2013.

[23] A. Khachaturyan, S. Semenovskaya, and B. Vainstein, "Statistical-thermodynamic approach to determination of structure amplitude phases," *Soviet physics, crystallography*, vol. 24, no. 5, pp. 519–524, 1979.

[24] A. Khachaturyan, S. Semenovsovskaya, and B. Vainshtein, "The thermodynamic approach to the

structure analysis of crystals," *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 37, no. 5, pp. 742–754, 1981.

[25] J. Grus, *Data Science from Scratch: First Principles with Python*. O'Reilly, 1st ed., 2015.

[26] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," *Proceedings of the 1999 Congress on Evolutionary Computation*, vol. 3, no. February, pp. 1945–1950, 1999.

[27] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proceedings of the 2000 Congress on Evolutionary Computation, CEC 2000*, vol. 1, pp. 84–88, 2000.

[28] R. E. Perez and K. Behdinan, "Particle swarm optimization in structural design," in *Swarm Intelligence, Focus on Ant and Particle Swarm Optimization*, IntechOpen, 2007.

[29] Y. Shi and R. C. Eberhart, "Parameter Selection in Particle Swarm Optimization," in *Proceedings of the 7th International Conference on Evolutionary Programming VII*, EP '98, (London, UK), pp. 591–600, Springer-Verlag, 1998.

[30] S. Luke, *Essentials of Metaheuristics*. Lulu, second ed., 2013. Available for free at http://cs.gmu.edu/∼sean/book/metaheuristics/.

[31] D. Chakraborty and A. Dutta, "Optimization of FRP composites against impact induced failure using island model parallel genetic algorithm," *Composites Science and Technology*, vol. 65, no. 2005, pp. 2003–2013, 2013.

[32] S. Y. Chen, "An approach for impact structure optimization using the robust genetic algorithm," *Finite Elements in Analysis and Design*, vol. 37, no. 5, pp. 431–446, 2001.

[33] Y. Nourani and B. Andresen, "A comparison of simulated annealing cooling strategies," vol. 31, pp. 8373–8385, 1998.

[34] A. K. Peprah, S. K. Appiah, and S. K. Amponsah, "An Optimal Cooling Schedule Using a Simulated Annealing Based Approach," *Applied Mathematics*, vol. 8, no. 08, p. 1195, 2017.

[35] H. Youssef, S. M. Sait, and H. Adiche, "Evolutionary algorithms, simulated annealing and tabu search: A comparative study," *Engineering Applications of Artificial Intelligence*, vol. 14, no. 2, pp. 167–181, 2001.

[36] S. Arora, *Introduction to Optimum Design*. Academic Press, 2012.