Defence Technology 40 (2024) 125–137

Contents lists available at ScienceDirect

Defence Technology

journal homepage: www.keaipublishing.com/en/journals/defence-technology

Topological optimization of ballistic protective structures through genetic algorithms in a vulnerability-driven environment

Salvatore Annunziata, Luca Lomazzi*, Marco Giglio, Andrea Manes

Politecnico di Milano, Department of Mechanical Engineering, Via Giuseppe La Masa, 1, 20156, Italy

ARTICLE INFO

Article history:
Received 14 February 2024
Received in revised form
2 May 2024
Accepted 22 May 2024
Available online 25 May 2024

Keywords:
Topological optimization
Protective structure
Genetic algorithm
Survivability
Vulnerability

ABSTRACT

Reducing the vulnerability of a platform, i.e., the risk of being affected by hostile objects, is of paramount importance in the design process of vehicles, especially aircraft. A simple and effective way to decrease vulnerability is to introduce protective structures to intercept and possibly stop threats. However, this type of solution can lead to a significant increase in weight, affecting the performance of the aircraft. For this reason, it is crucial to study possible solutions that allow reducing the vulnerability of the aircraft while containing the increase in structural weight. One possible strategy is to optimize the topology of protective solutions to find the optimal balance between vulnerability and the weight of the added structures. Among the many optimization techniques available in the literature for this purpose, multi-objective genetic algorithms stand out as promising tools. In this context, this work proposes the use of a in-house software for vulnerability calculation to guide the process of topology optimization through multi-objective genetic algorithms, aiming to simultaneously minimize the weight of protective structures and vulnerability. In addition to the use of the in-house software, which itself represents a novelty in the field of topology optimization of structures, the method incorporates a custom mutation function within the genetic algorithm, specifically developed using a graph-based approach to ensure the continuity of the generated structures. The tool developed for this work is capable of generating protections with optimized layouts considering two different types of impacting objects, namely bullets and fragments from detonating objects. The software outputs a set of non-dominated solutions describing different topologies that the user can choose from.

1. Introduction

The concept of survivability, defined as the ability of a platform to avoid or withstand a man-made hostile environment, plays a crucial role in platform design. Increasing survivability is usually performed by reducing susceptibility and vulnerability. The former refers to the inability of a platform to avoid the elements of an enemy's defense, while the latter identifies its inability to withstand such elements.

Survivability-driven design of structures has started to play a crucial role in World War I [1]. In that period, measures to enhance survivability were limited to incorporating rudimentary protections to stop threats directed towards the pilot, consequently increasing the platform's ability to withstand man-made threats [1]. Significant advancements in combat survivability emerged during the Cold War conflict in Southeast Asia in the 1970s when the high-threat environment led to specifically designing platforms for reduced vulnerability [2,3]. The '80s marked a further increase in the importance of survivability and vulnerability, culminating in the establishment of the Department of Defense Survivability/Vulnerability Information Analysis Center (SURVIAC) in 1984.

Nowadays, the design of military and civil platforms is centered around enhancing their survivability features, ensuring the protection of occupants and mission success. This is exemplified by initiatives like the Joint Strike Fighter program [1].

The principles of survivability and vulnerability were born in the aeronautical field thanks to the seminal work written by R. Ball in 1985 [1]. Here, the concept of survivability for aerial platforms was formalized, and design principles for fixed and rotary-wing aircraft to improve survivability features were presented. A comprehensive methodology for conducting detailed single-hit vulnerability

* Corresponding author.
E-mail address: luca.lomazzi@polimi.it (L. Lomazzi).
Peer review under responsibility of China Ordnance Society

https://doi.org/10.1016/j.dt.2024.05.011

assessments during the conceptual design stage of aircraft was presented in Ref. [4], based on Ball's earlier research. The methodology, similar to more recently proposed approaches, was based on a statistical framework, requiring a substantial amount of data to evaluate the impact of objects from various perspectives. To handle this data, computer programs like the FASTGEN and COVART software packages were developed [5,6]. The FASTGEN software implemented a shot-line-based procedure, which, when taken in by the COVART software, enabled the determination of crucial indices in the vulnerability assessment framework, extensively explained in Refs. [7,8]. It is worth noting that the shot-line-based procedure employed in FASTGEN software has roots in consolidated ray-tracing algorithms. The vulnerability concepts pioneered by Ball were further explored in Ref. [9], where a computable calculation model based on the shot-line scanning method was introduced. This model allowed accounting for the threat's state of motion during the penetration process in a simplified manner. Subsequently, the authors refined the shot-line method and introduced the evaluation of platform kill state to assess the overall kill probability of aircraft in a single-hit scenario, considering the potential overlap between components [9]. In Ref. [10], a direct simulation method was introduced to assess the multiple-hit vulnerability of aircraft with overlapping components. This approach was based on the statistical framework of Monte Carlo simulations proposed in Ref. [11].

Although the principles of survivability and vulnerability were born in the aeronautical field, they extend far beyond such field and are commonly applied to the design of any platform susceptible to threats. For instance, in Ref. [12], Kincheleo et al. set the fundamentals for ground combat system vulnerability. Based on this work, Yoo and Park, in Ref. [13], introduced a unified process to assess the vulnerability of ground vehicles using target functional modeling and fault tree analyses. Regarding the naval framework, in Ref. [14] a simplified vulnerability analysis program to evaluate the vulnerability of the warship at the conceptual design stage was presented. In Ref. [15], Brownlow et al. proposed an adaptable network-based approach that allows the development of a novel vulnerability assessment method for early-stage naval ship design. Always concerning naval ships, in Ref. [16] Sungkyun et al. developed a vulnerability assessment model by using the M&S technique. Moreover, quantitative risk and resilience analysis of passive protection measures installed on critical assets were introduced in Ref. [17]. Additionally, a management methodology was reported to reduce the vulnerability of individuals within and near these assets.

To date, most of the works proposed in the literature have only focused on methods for assessing the vulnerability of platforms, while few contributions have studied novel solutions for reducing vulnerability. Moreover, considering that the traditional approach adopted to reduce vulnerability involves installing protective measures on the structure, particularly in the aerospace field, the added weight can result in diminished flight performance. Therefore, it is important not only to explore novel protective solutions but also to investigate methods for mitigating excessive structural weight gain. In this context, when the objective shifts towards reducing platform vulnerability without incurring significant weight increases, the adoption of an optimization algorithm may offer a viable solution. A promising approach would be that of optimizing protective structures to protect the critical components identified through preliminary vulnerability assessments. Optimizing protective structures requires performing topological optimization. Various techniques for topological optimization, including solid isotropic material with penalization (SIMP) [18,19], level set [20], particle swarm [21], evolutionary structural optimization (ESO) [22,23], and genetic algorithms (GAs) [24–26] have been proposed in the literature. Particularly promising capabilities characterize GAs. Such algorithms are powerful and robust stochastic global search methods [24,27,28]. They belong to an emerging family of optimization methods that simulate natural biological evolution based on the survival-of-the-fittest principle [29]. Ref. [30] presented a procedure to develop near-optimal topologies of load-bearing truss structures using genetic algorithms. Furthermore, Ref. [31] solved a discrete problem through GAs by using a bit-array representation method, also presenting a way of suppressing possible checkerboard pattern outcomes.

In this context, this work builds on the general vulnerability framework proposed by Lomazzi et al. [32] in 2021, and proposes a method for developing a vulnerability-driven topological optimization tool to design protective components. Two independent modules were developed, each aimed to optimize protections against a different threat. That is, two threats were considered: (i) projectiles and (ii) fragments resulting from a detonating object. Multi-objective topological optimization of protections was performed through GAs to simultaneously reduce (i) weight increase and (ii) the vulnerability of identified critical components. The approach was demonstrated through a case study involving a representative portion of an aerial platform.

This work is organized as follows. Section 2 presents two different approaches to vulnerability analysis, protection generation, and optimization: one specifically developed for designing protections against projectiles, and the other specialized for fragments from detonating objects. Section 3 demonstrates the two approaches through a case study involving projectiles and fragments impacting on a representative portion of an aerial platform. Section 4 draws out the conclusions and presents some possible future developments.

2. Vulnerability-driven optimization methodology

The algorithm presented in this work comprises two distinct modules tailored to address specific types of threats. The first module is designed to deal with projectiles, while the second module focuses on fragments resulting from a detonating object. The modules build on the vulnerability approach described in Ref. [32] and aim to generate and optimize protective solutions.

2.1. Optimization of protections against ballistic impacts

The module dedicated to generating ballistic protections against projectiles operates within a bi-dimensional framework. This design choice is driven by the nature of the vulnerability assessment methodology employed. Although the framework has already been described in Ref. [33], it is briefly recalled here in the interest of clarity. The workflow is shown in Fig. 1, and it involves the following steps.

(1) *Initial vulnerability assessment.* The initial vulnerability assessment consists of the vulnerability assessment of the platform without protections. The analysis makes use of a ray tracing algorithm where straight shot-lines are generated from a plane perpendicular to the projectile trajectory of interest. The distance of the plane from the target platform does not affect the vulnerability assessment process. The results obtained in this phase are used to identify the vulnerable parts so that the protection surfaces can be properly placed to protect them. This part of the algorithm is detailed in subsection 2.1.1.
(2) *Grid initialization.* During the grid initialization phase, protective elements are initialized on the shot-lines plane so to protect the most vulnerable components selected through

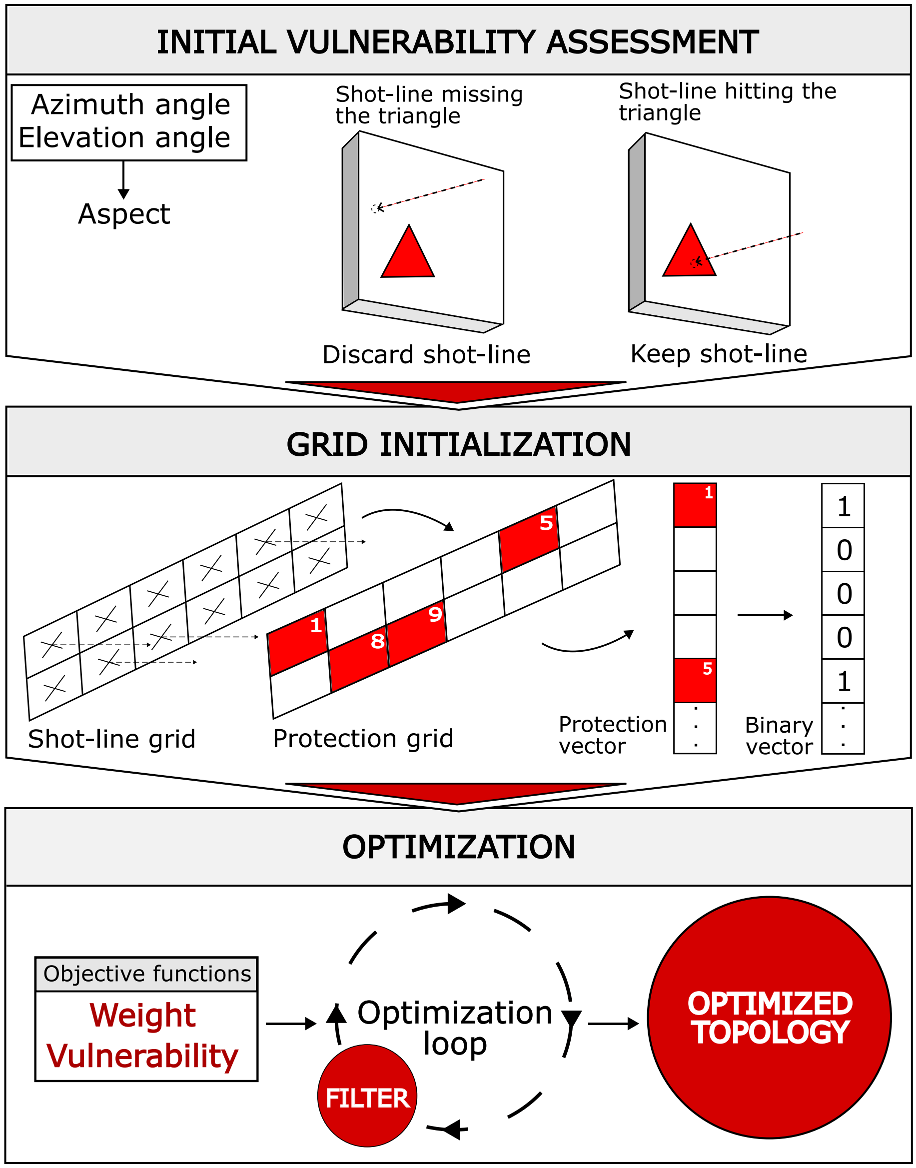

Fig. 1. Optimization framework for designing protections against ballistic impacts.

the information coming from the previous step. This phase is detailed in subsection 2.1.2.

(3) *Optimization.* The initialized protections are the input of the optimization process. Multi-objective GAs are used to generate protective solutions with optimized topology. Two objective functions are used: the weight of the protections and the vulnerability of the target components. To guarantee convergence to continuous solutions, a spatial filter is also considered during the optimization process, penalizing protections with spread and isolated elements. The optimization phase is further described in subsection 2.1.3.

2.1.1. Initial vulnerability assessment

The algorithm implemented to assess the platform vulnerability is detailed in Ref. [32], and it is here summarized in the interest of clarity.

The first step of the vulnerability assessment consists of pre-processing the platform. That is, the critical components are identified and modeled through simple hollow geometric shapes that preserve the mass and volume of the real components. Then, the components are projected onto the plane perpendicular to the projectile direction, which is identified using azimuth and elevation angles, as shown in Fig. 2. The assessment takes into account the position and geometry of all the components in the platform. Multiple aspects can be considered for optimization.

Each aspect is then analyzed separately. The plane perpendicular to the projectile direction is divided into regular square cells, so to cover the whole platform projection. The grid of square cells is referred to as *shot-lines grid*. A shot line perpendicular to the plane is generated at the center of each cell, and the platform components it intersects are identified through a ray-tracing algorithm. All the shot-lines that do not intersect any components are discarded. A penetration algorithm is then used to account for the projectile velocity loss as it penetrates through the intersected components. For the sake of brevity, the details regarding the penetration algorithm will not be included in the paper. However, readers are encouraged to consult the work presented in Ref. [32] for additional

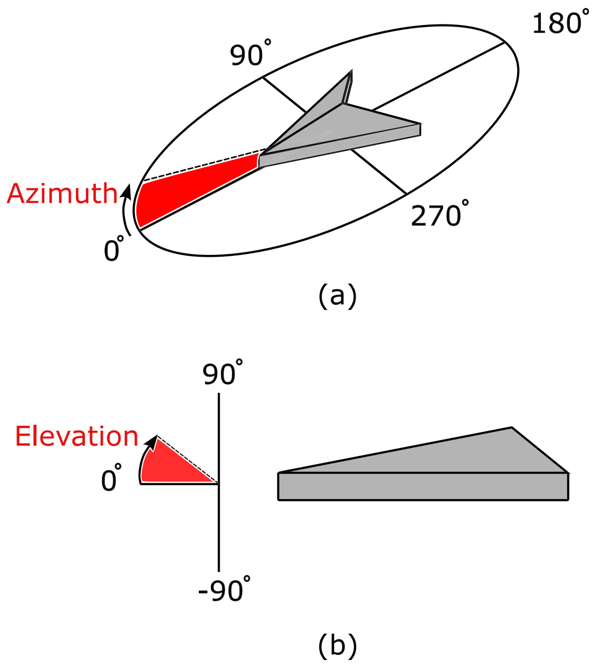

Fig. 2. (a) Aspect definition - Azimuth angle; (b) Elevation angle. Reprint from Ref. [33].

insights. Finally, kill probabilities are computed by means of a statistical framework. The i-th component kill probability P^i_{k} is determined according to Eq. (1).

$$P^i_{\mathrm{k}} = P^i_{\mathrm{H}} \cdot P^i_{\mathrm{k|H}} = P_{\mathrm{H}} \frac{A^i_{\mathrm{v}}}{A^{\mathrm{Ptf}}_{\mathrm{p}}} \tag{1}$$

where P^i_{H} is the component susceptibility, $P^i_{\mathrm{k|H}}$ is its kill probability given that the platform has been hit, P_{H} is the platform susceptibility, while A^i_{v} and $A^{\mathrm{Ptf}}_{\mathrm{p}}$ are the i-th component vulnerable area and the platform presented area, respectively. Presented and vulnerable areas are computed using Eqs. (2) and (3).

$$A^i_{\mathrm{p}} = N^i \cdot s^2 \tag{2}$$

$$A^i_{\mathrm{v}} = A^i_{\mathrm{p}} \cdot P^i_{\mathrm{k|h}} \tag{3}$$

where N^i is the number of shot-lines hitting the i-th component in the considered aspect, s^2 is the area of each square cell with length s, and $P_{\mathrm{k|h}}$ is the i-th component vulnerability, i.e., its kill probability given that the component has been hit. Similarly to Eq. (2), the platform presented area is computed through Eq. (4).

$$A^{\mathrm{Ptf}}_{\mathrm{p}} = N^{\mathrm{Ptf}} \cdot s^2 \tag{4}$$

where N^{Ptf} is the number of shot-lines hitting the platform in the considered aspect.

2.1.2. Grid initialization

After computing the vulnerability of the components and considering all the aspects of interest, the most critical components and aspects are selected. For simplicity, in this work, the algorithm is described assuming that only one component and aspect are considered. In the selected aspect, an *optimization grid* is superimposed on the *shot-lines grid*. The *shot-lines grid*, and consequently the *optimization grid*, is placed over the projection of the whole platform and is not limited to the selected components only. The protection layout is initialized by placing elements in the cells where shot-lines intersect the selected critical component. The *optimization grid* and the initialized elements are shown in Fig. 3. Here, dashed lines represent the shot-lines that hit the selected component, and red square cells are used to indicate the initial protection layout.

A vector $\boldsymbol{x}$ is used to identify where protection elements have been initialized in the optimization grid. The vector has N_{cells} elements, where N_{cells} is the number of cells in the optimization grid. Each element can assume a value of 0 or 1, indicating the absence or presence of the protection, respectively. In the interest of clarity, the vector initialization is shown in Fig. 4. Note that the elements of the protection are initialized so to stop all the shot-lines that hit the target component. Consequently, the initialized protective structure is discontinuous and composed of a scattered layout.

2.1.3. Optimization

A multi-objective GA is used to optimize the protection layout. Optimization is performed using the MATLAB© built-in function *gamultiobj*.

The vector that identifies the protection layout is taken in by the algorithm. The optimization is constrained so to only consider integer values of the design variables. Consequently, each vector element can either assume values 0 or 1. Furthermore, the lower

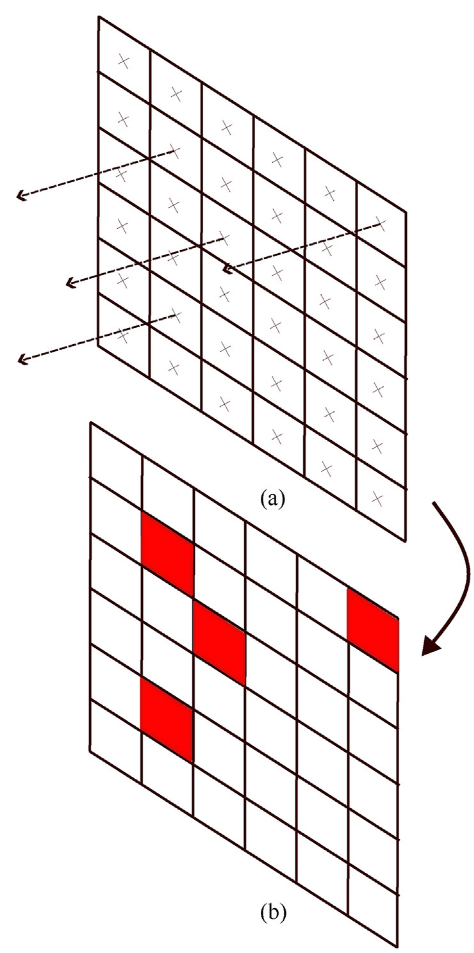

Fig. 3. Shot-lines grid and optimization grid: (a) Shot-lines grid; (b) Optimization grid.

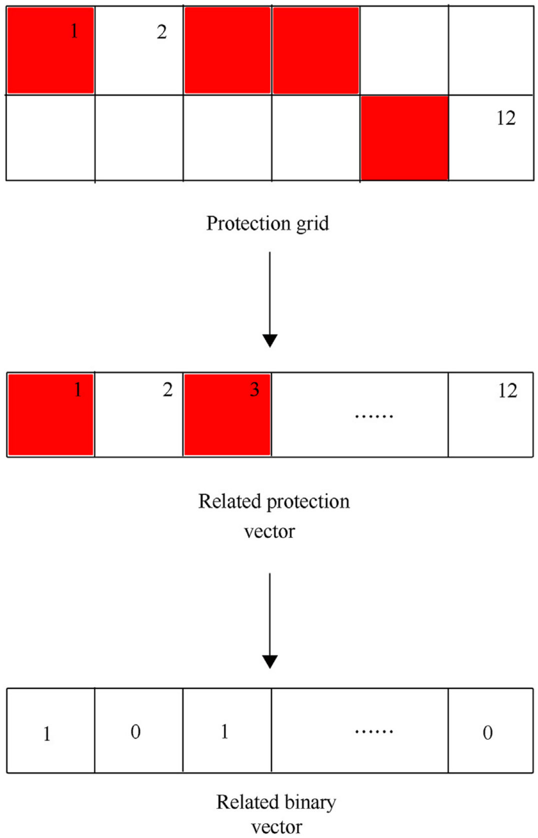

Fig. 4. Generation of the binary vector representing the protection. Reprint from Ref. [33].

bound for the optimization process is represented by the null vector, while a vector of ones represents the upper bound. Two objective functions are considered to optimize the design: component vulnerability and normalized protection mass. To ensure the proper functioning of the genetic algorithm, the two fitness functions need to be defined based on the vector representing the protection layout. Vulnerability is computed according to the procedure already described in subsection 2.1.1, which is modified to account for the presence of protective elements. More specifically, the shot-lines grid is utilized to calculate the kill probability using the given formulas, considering that when a protective element intercepts a shot-line it is fully blocked. The assumption that the shotlines are stopped when they hit protective elements needs to be verified after coming up with the optimized topology. In fact, the optimizer only defines a non-dominated optimal protection layout, leaving penetration considerations to the user. Instead, weight is defined by summing up the number of ones in the protection vector, and dividing the sum by the total number of elements, as shown in Eq. (5).

$$\overline{mass} = \frac{\sum_{i}^{N} x(i)}{N_{\text{cells}}} \tag{5}$$

where $x(i)$ identifies the i-th element of the vector defining the protection layout.

The objective functions mentioned above are not sufficient to converge to continuous protection layouts. To address this issue, a spatial filter is introduced within the optimization loop. The filtering mechanism acts through a penalty factor applied to the fitness functions. That is, the number of isolated clusters in the i-th protection layout N^i_{clusters} is counted, and the objective functions of the i-th layout are multiplied by the penalty factor P^i_{f} shown in Eq. (6).

$$P^i_{\text{f}} = 10N^i_{\text{clusters}} \tag{6}$$

2.2. Optimization of protections against fragments from detonating objects

The module dedicated to the generation of protections against fragments operates within a three-dimensional framework. Also, in this case, the design choice is driven by the nature of the vulnerability assessment methodology employed. Working in this framework, the algorithm generates the protective structure directly on the outermost part of the platform being analyzed. The workflow is shown in Fig. 5, and it involves the following steps.

(1) *Initial vulnerability assessment.* The initial vulnerability assessment consists of the vulnerability assessment of the platform without protections. The analysis makes use of a ray tracing-based algorithm, considering the trajectories of the generated fragments. The results obtained in this phase are used to identify the vulnerable parts so that the protection surfaces can be properly placed to protect them. This phase of the algorithm is detailed in subsection 2.2.1.
(2) *Protection initialization.* During the protection initialization phase, protection elements are placed on the outermost layer of the platform to intercept all the fragments intersecting the target components identified in the previous step. This part of the algorithm is detailed in subsection 2.2.2.
(3) *Optimization.* The initialized protections are the input of the optimization process. Multi-objective GAs are used to generate protective solutions with optimized topology. Protection weight and the vulnerability of the target components are minimized by the optimization algorithm. Furthermore, a novel spatial filter is proposed in this work to guarantee convergence to continuous solutions, penalizing protections with spread and isolated elements. Such a filter works on the graph representation of the protective structure. This phase is detailed in subsection 2.2.3.

2.2.1. Initial vulnerability assessment

The algorithm implemented to assess the platform vulnerability to fragments is as an extension of the algorithm described in subsection 2.1.1. No changes are applied to the pre-processing phase described above. After all the components have been modeled through simplified mass- and volume-preserving shapes, a 3D hemispherical *detonation grid* is generated around the platform. The grid is made of detonation points, each representing the position where a detonating object would detonate and generate fragments. The velocity vector defining the threat motion is defined according to azimuth and elevation angles. In reference to the global reference frame (x_g, y_g, z_g) shown in Fig. 6, azimuth represents the rotation around z_g, and elevation is the rotation around $-y_g$. All the threats share the same velocity vector. The hemispherical grid is centered to the platform geometrical center and its radius identifies the threat *detonation distance*. Multiple detonation distances can be considered in the vulnerability assessment. Generating a detonation grid requires defining the detonation *reference frame* shown in Fig. 6. Its origin coincides

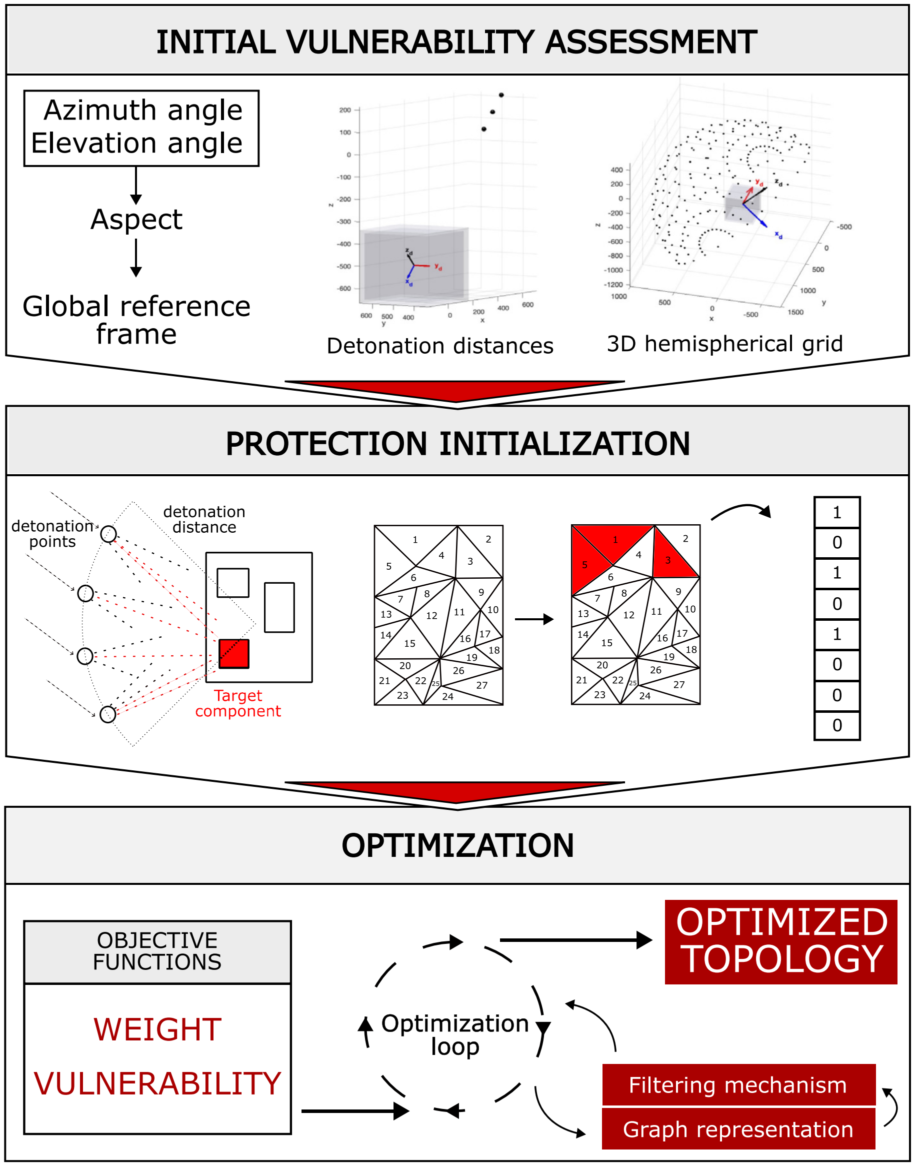

Fig. 5. Optimization framework for designing protections against fragments impacts.

with the geometric center of the platform, the x-axis (x_d) is aligned with the detonating object velocity, and the z-axis (z_d) is perpendicular to this velocity and described by the same azimuth (i.e., it is rotated by 90° around $-y_\mathrm{g}$). The y-axis is defined by the cross-product between x_d and z_d. The number of detonation points within each *detonation grid* is determined through two parameters:

(1) The number of equally spaced points in the semi-circumference projected onto the xy plane of the detonation frame.
(2) The number of equally spaced points in the semi-circumference projected onto the xz plane of the detonation frame.

Shot-lines are defined according to each fragment velocity vector and start from the respective detonation point. The fragmentation of a representative detonation object is shown in Fig. 7, where the detonating object velocity vector is shown by a dashed black arrow, and the shot-lines are identified through black dashed lines. Fragment generation and their velocity vector are described through the Gurney formula. The interested reader is referred to Refs. [34,35] for additional details.

Kill probabilities at a specific detonation distance are defined as the mean kill probabilities due to the detonation of threats at each point within the hemispherical grid at that distance. Considering for simplicity the i-th component of the platform, its kill probability is computed by means of a statistical framework embedding a multi-hit approach. That is, for each detonating object, the number of shot-lines hitting the component is computed, and i-th component kill probability P_k^i is defined according to Eq. (7)

$$P_\mathrm{k}^i = 1 - \prod_{1}^{n_i}\left(1 - P_\mathrm{H}^i \cdot P_\mathrm{k|h}^i\right) \tag{7}$$

where n_i is the number of shot-lines intercepting the component. Eq. (7) assumes that the fragments represent n_i independent hits.

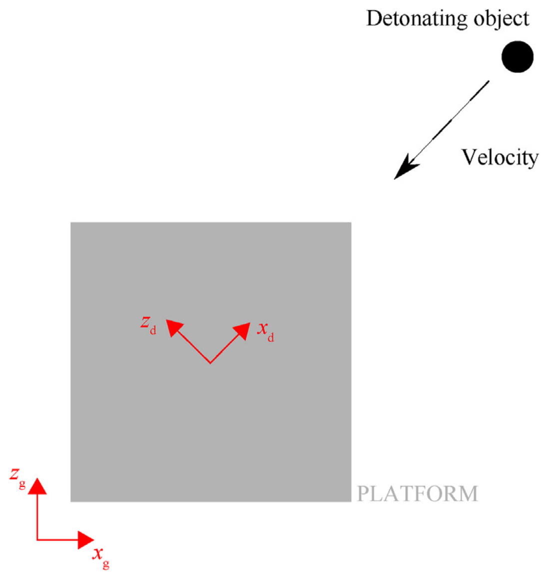

Fig. 6. Detonation point, global and detonation reference frame.

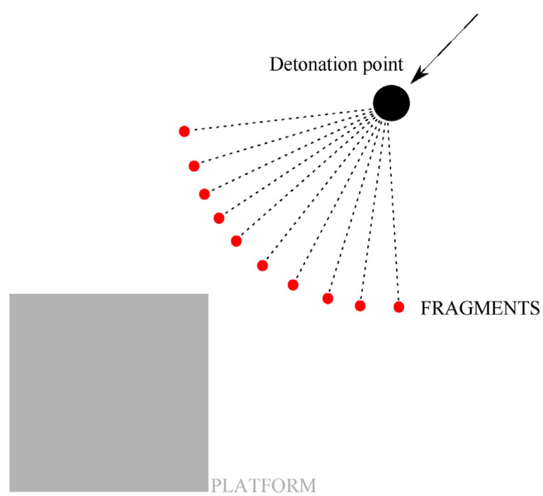

Fig. 7. Detonating object fragmentation.

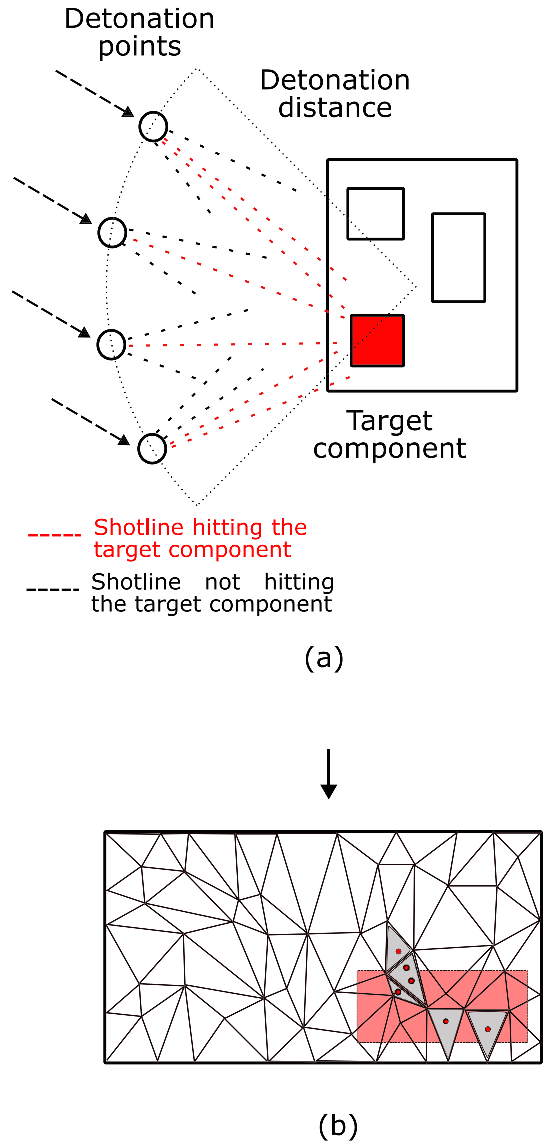

Fig. 8. Initialization of the protective structure: (a) Side view; (b) Front view.

2.2.2. Protection initialization

Several critical components identified through the vulnerability analysis described above can be selected for designing the protection. For simplicity, in this work, the algorithm is described by assuming that only one component is considered.

Initializing the protection requires generating an optimization mesh over the outermost layer of the platform model, where protection elements will be placed by the algorithm. The relevant pieces of information carried by the mesh are the elements' position and their connectivity. The intersection points between the shot-lines generated by all the detonation points in all the detonation grids and the mesh elements are tracked. Only the shot-lines intersecting the selected critical component are retained in the analysis. The protection is initialized at all the intercepted mesh elements, as shown in Fig. 8. Note that shot-lines may intersect more than one mesh element, especially if complex geometries of the platform outermost layer are considered. However, considering that protections are assumed to be capable of completely stopping incoming threats, protection elements are initialized only at the first mesh elements intercepted by the fragments. The resulting protection can stop every fragment directed toward the selected critical component, thereby reducing its kill probability to zero. At this phase of the algorithm, the initialized protective structure is discontinuous and composed of a scattered layout.

Similarly to what has already been described in subsection 2.1.2, the initialized protection layout is encoded into a binary vector. The vector length is the number of elements of the mesh of the platform's outermost layer.

2.2.3. Optimization

A multi-objective GA is used to optimize the protection layout. Optimization is performed using the MATLAB© built-in function *gamultiobj*. The vector that identifies the protection layout is taken in by the algorithm. Each vector element can assume a value of 0 or 1, indicating the absence or presence of protection elements.

Two objective functions are considered to optimize the design: component vulnerability and normalized protection mass. The former is computed according to the procedure already described in subsection 2.2.1, assuming that the shot-lines intercepting protection elements are stopped. The latter is computed according to Eq. (5).

The objective functions are not sufficient to converge to a continuous protection layout. To address this issue, a novel, specifically developed mutation function is presented. This way, besides minimizing the weight of the protection and increasing the chance that shot-lines intercept protection elements, the algorithm tries to enforce a cohesive and continuous structure by promoting the formation of a single cluster of protective material. The custom mutation function is described in Algorithm 1. Here, Population is the $m \times n$ matrix representing the m individuals in the population described by the n design variables, and $\boldsymbol{A}_g$ is the matrix taking into account the connectivity of the current protection. Matrix $\boldsymbol{A}_g$ is sparse and contains ones at all positions (i,j) where the i-th element is connected to the j-th element. The custom mutation function is based on Bernoulli trials. That is, for each element in the vector defining the protection layout, a random number r_i is extracted from a uniform distribution ranging from 0 to 1 and compared to a mutation probability P_m defined according to Algorithm 1. Mutation occurs if the extracted number r_i is smaller or equal to the mutation probability P_m. The values of the vector elements selected for mutation are then replaced by their ones' complement.

Algorithm 1 Custom mutation function for topological optimization of protections against fragments

for $i = 1: m$ **do**
for $j = 1: n$ **do**
$N(i,j) = \sum \boldsymbol{Population}(i,:)\boldsymbol{A}_g(j,:)$
end for
end for
Mutation probability matrix: $P_m = 0.3N/\max(N)$
Perform Bernoulli trials for selecting mutating genes

Furthermore, to increase the chance of converging to continuous optimized solutions, a spatial filter is introduced into the optimization loop. The number of material clusters is counted, and discontinuous solutions are penalized. Counting the number of clusters the protection is made of is performed through the graph representation of the protection vector. In graph theory, graphs are representations of networks and are made of nodes and edges. Each node represents an entity, while the edges represent the connections between the nodes. Node connectivity is represented through the adjacency matrix $\boldsymbol{A}$. Assuming that a graph is made of n_g nodes, its matrix $\boldsymbol{A}$ has dimension $n_g \times n_g$. A nonzero element at position (i,j) identifies that the i-th and the j-th nodes are connected. Note that $\boldsymbol{A}$ is symmetric and that its principal diagonal is zero, considering that nodes cannot be self-connected. Furthermore, it is important to specify that graphs do not inherently convey spatial information. Given the characteristics just described, graphs seem to provide an effective way to represent meshes. In this work, graph nodes correspond to the outermost platform layer mesh elements, and graph edges represent the connectivity between the mesh elements. Graph nodes with value 1 correspond to mesh elements where protective material is placed, as shown in Fig. 9. The graph representation of the protection layout is then updated to count the number of isolated clusters N_i. Graph update is performed according to Algorithm 2 and aims to remove all the graph edges that start or end at nodes with no material. In Algorithm 2, V_1^m and V_2^m are vectors whose i-th element identifies whether protective material is present at the node where the i-th edge starts or ends, respectively. Index m is used to identify the m-th protection layout, and index k scans through the elements of vector $V_3^m = V_1^m V_2^m$.

The number of isolated clusters is then used to penalize the fitness functions according to Eq. (8)

Algorithm 2 Graph update procedure

for $i = 1: m$ **do**
$V_3^m = V_1^m \cdot V_2^m$
for $i = 1: k$ **do**
if $V_3^m(k) = 0$ **then**
Remove k-th edge from the graph
end if
end for
end for

$$\left[\overline{mass}, P_k^i\right] = (N_i - 1)\left[\overline{mass}, P_k^i\right] \tag{8}$$

This process is repeated for each protection layout generated by the optimization algorithm. At the end of the optimization process, a set of optimized solutions with different topologies is obtained.

3. Case study

The optimization framework described in Section 2 is demonstrated through a case study involving a representative portion of the aerial platform. It includes the outermost layer, named skin, and three representative critical components modeled by hollow cubes, named C1, C2, and C3, respectively, as shown in Fig. 10. The hollow cubes are simplified models of some representative platform's components. These models must guarantee the same encumbrance and mass of the original object they represent. This is achieved through a two-step procedure. The first step consists of modeling the box so that it wraps the original object. This way, encumbrance conservation is guaranteed. Then, the wall thickness is computed to fulfill the mass conservation requirement. The procedure relies on Eqs. (9)–(11).

$$M_{\text{original}} = M_{\text{simplified}} \tag{9}$$

$$M_{\text{original}} = \rho_{\text{simplified}} \cdot V_{\text{simplified}} \tag{10}$$

$$M_{\text{original}} = \rho_{\text{simplified}}[abc - (a - 2t)(b - 2t)(c - 2t)] \tag{11}$$

where M is the mass of the object, ρ is the density of the material the object is made of, V is the volume of the object, a, b and c is the box dimensions and t is the cube wall thickness. Solving Eq. (11) for t leads to the determination of the required box thickness. The kill probability of the critical components given a hit on the components themselves are $P_{k|h}^{C1} = 0.3$, $P_{k|h}^{C2} = 0.2$, $P_{k|h}^{C3} = 0.3$, and $P_{k|h}^{skin} = 0$. Those values depend on many factors, and the striking velocity is among them. In particular, experiments should be carried out at different striking velocities to come up with a variable $P_{k|h}$ value [1]. This work takes for granted the $P_{k|h}$ values and builds up on them to compute the vulnerability of the components and drive the optimization algorithm.

3.1. Optimized protections against ballistic impacts

The results of the initial vulnerability assessment are not reported here in the interest of brevity. Component C2 was identified as the target component for the optimization of the protection. Two different aspects were considered: (i) *Azimuth* = 90°, *Elevation* = 0°, and (ii) *Azimuth* = 50°, *Elevation* = 30°. The single-point crossover function was adopted within the optimization scheme. The hyperparameters of the optimization are the size of the

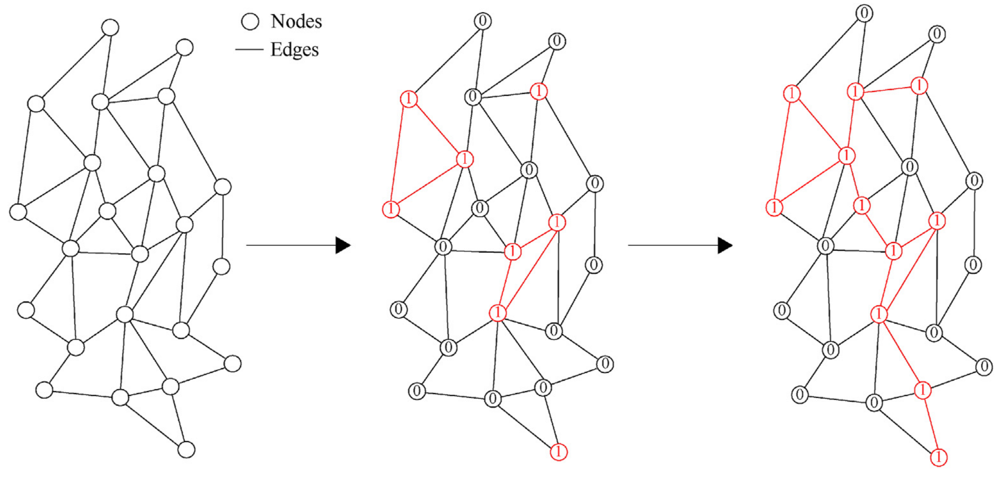

Fig. 9. Example of graph representation: (a) Graph representing the mesh; (b) Graph representing the protection with several cluster; (c) Graph representing the optimized protection with a single cluster.

population, the tolerance, the maximum number of generations, the crossover fraction, and the mutation parameter. The values adopted in this work are reported below. They were tuned to get results in a reasonable time, without sacrificing the quality of the results.

- Population size: 200
- Tolerance: 10^{-5}.
- Max. Number of generations: 100
- Crossover fraction: 0.8
- Mutation parameter: 0.01

The results of the optimization process in regard to the 90°–0° aspect are shown in Fig. 11. The solutions in the Pareto front represent different trade-offs between mass and kill probability, allowing for flexibility in selecting the most suitable protection configuration based on specific requirements and constraints. The protection layouts designed by the optimization algorithm are shown in Fig. 12. Here, shot-lines are represented by black dots and the generated protection is shown in red.

In Fig. 13, further observations can be made regarding the solutions presented in Fig. 12, particularly focusing on the shot lines that intersect the component and the resulting protections. Here, the black dots indicate the centers of the grid elements from which the shot lines originate, while the shot lines themselves, striking the component, are represented by dashed arrows. The generated protections are once again highlighted in red for clarity. Solution 1, as can be observed in Fig. 11, represents the heaviest configuration. Since four shot-lines intersect the target component, the software placed material at the corresponding four points on the grid to nullify the kill probability. This solution offers the lowest kill probability of the target component, i.e., $P_{\mathrm{k}}^{\mathrm{C2}} = 0\%$, at the expense of the normalized mass $\overline{mass} = 0.0076$. Solution 2 is described by reduced mass ($\overline{mass} = 0.0057$) and slightly higher kill probability ($P_{\mathrm{k}}^{\mathrm{C2}} = 0.005\%$) compared to the previous configuration. However, it still provides a high level of protection to the target component, prioritizing survivability over mass. Note that two different protection topologies with the same objective function values were generated by the software. Solution 2 stopped three shot-lines out of the four. Solution 3 is lighter than the second solution ($\overline{mass} = 0.0038$), but offers reduced protection, i.e., $P_{\mathrm{k}}^{\mathrm{C2}} = 0.011\%$. Protective elements were placed to stop only 2 shot-lines out of 4. Also in this case the optimizer generated two different topologies

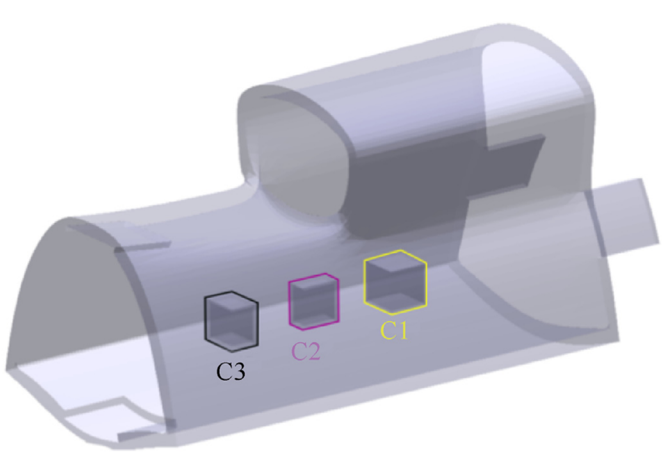

Fig. 10. Representative portion of the demonstrator considered in the case study.

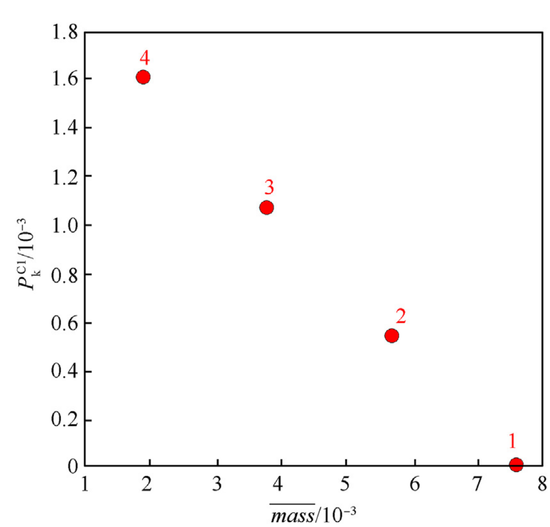

Fig. 11. 90°–0° aspect: Pareto front.

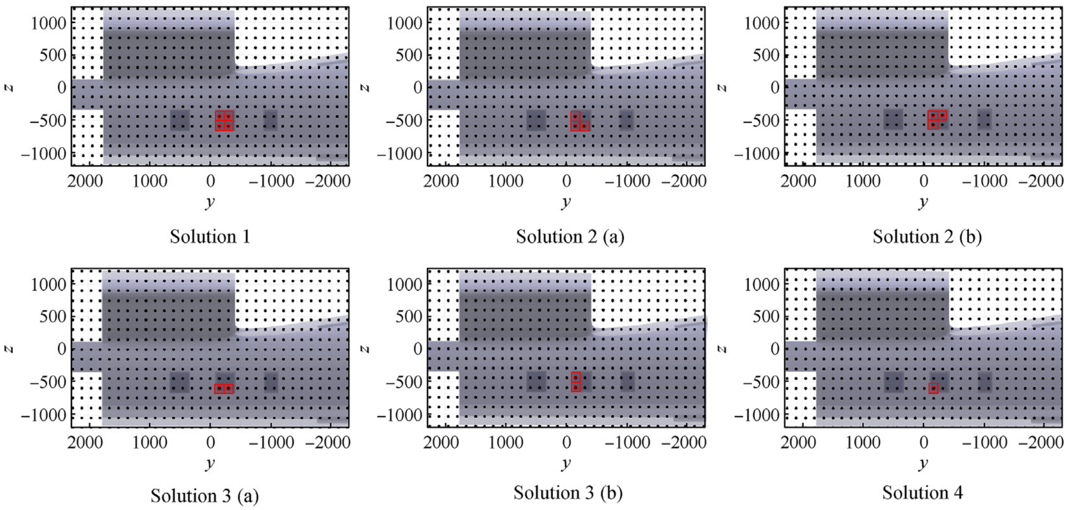

Fig. 12. 90°−0° aspect: optimized protection layouts.

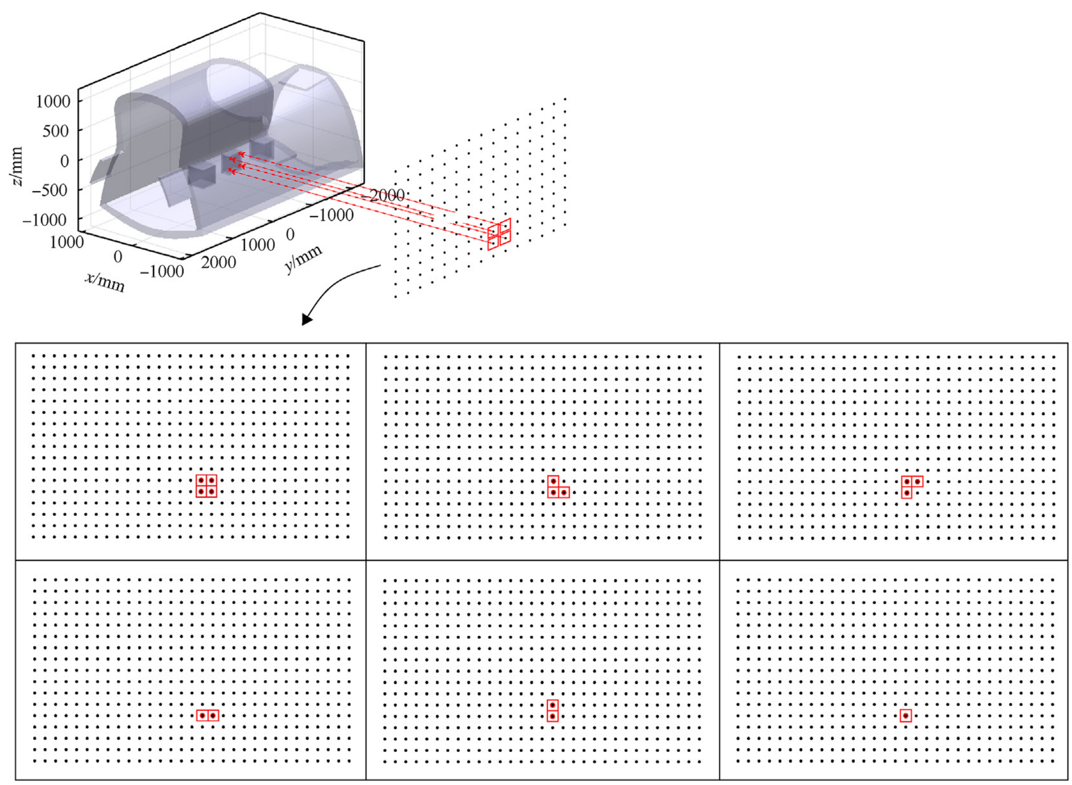

Fig. 13. 90°−0° aspect: optimized protection layouts, alternative visualization.

corresponding to the same objective function values. Solution 4 represents the lightest configuration ($\overline{mass} = 0.0019$) and provides limited protection to the target component, i.e., $P_{\mathrm{k}}^{\mathrm{C2}} = 0.16\%$. In this case, the protection layout allows stopping only 1 shot-line out of 4.

The solutions concerning the 90°−0° *aspect* are trivial, for this reason, the software is also tested with azimuth and elevation angles that would generate a shot grid not parallel to any face of the target component or any plane defined by the axes of the reference system of the analyzed platform. Consequently, also the 50°−30° *aspect* is analyzed, with the resulting Pareto plot displayed in Fig. 14. The protection layouts designed by the optimization algorithm are shown in Fig. 15. Solution 1 proposes a protection that reduces the kill probability of the target component to zero, blocking all the shot-lines, i.e. $P_{\mathrm{k}}^{\mathrm{C2}} = 0\%$, at the expense of the normalized mass $\overline{mass} = 0.0063$. Solution 2 consists of lighter protections, resulting in an increased kill probability of the target component. The solution proposes two different possible topologies that differ in terms of the arrangement of protective elements only, both with the same values of normalized mass ($\overline{mass} = 0.005$) and kill probability ($P_{\mathrm{k}}^{\mathrm{C2}} = 0.045\%$).

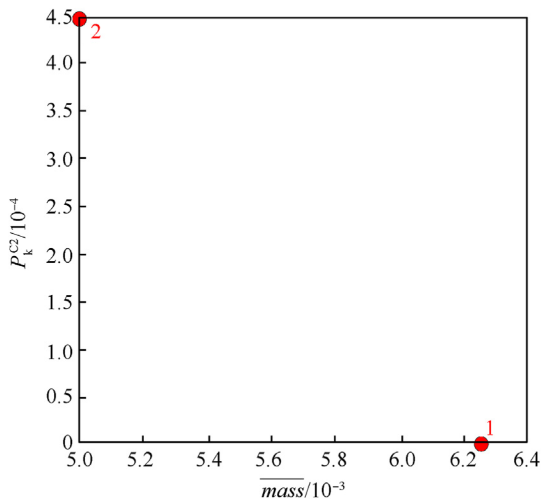

Fig. 14. 50°−30° aspect: Pareto front.

3.2. Optimized protections against fragments from detonating objects

Component C_3 was selected to optimize the protection in case of damage from fragments generated by detonating objects. The fragments impacting the model are obtained considering three detonation distances (i.e., 4 m, 4.5 m, 5 m) and ten detonation points along the *xy* and *xz* planes, assuming a single aspect for the detonating object (30°−45°). As in the case of the optimization of protections against ballistic impacts, the same hyperparameters already mentioned in subsection 3.1 must be provided to the optimizer. The values considered in this work are shown below:

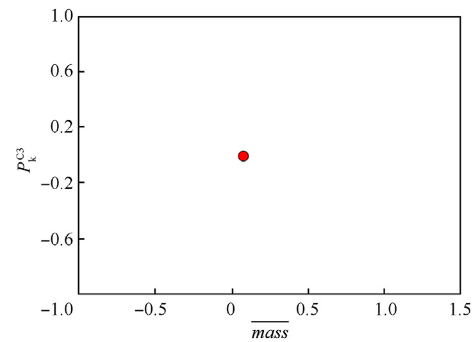

Fig. 16. Three-dimensional framework: Pareto front.

- Population size: 400
- Tolerance: 10^{-5}.
- Max. Number of generations: 1000
- Crossover fraction: 0.3
- Custom function (subsection 2.2.3)

In this case, the algorithm converged to a single solution, generating only one point in the Pareto front shown in Fig. 16. The optimized protection layout nullified the kill probability of the target component while providing a limited increased mass. That is, the solution was described by $\overline{mass} = 0.0736$ and $P_k^{C3} = 0.0$. The optimized protection layout is shown in Fig. 17. Here, the elements of the skin mesh where the protective material was placed are highlighted in red, and dashed lines are used to represent the shot-lines that would hit the target component if no protection was considered. The optimized protection allowed for intercepting all the shot-lines. Furthermore, the filter managed to generate a continuous protection layout.

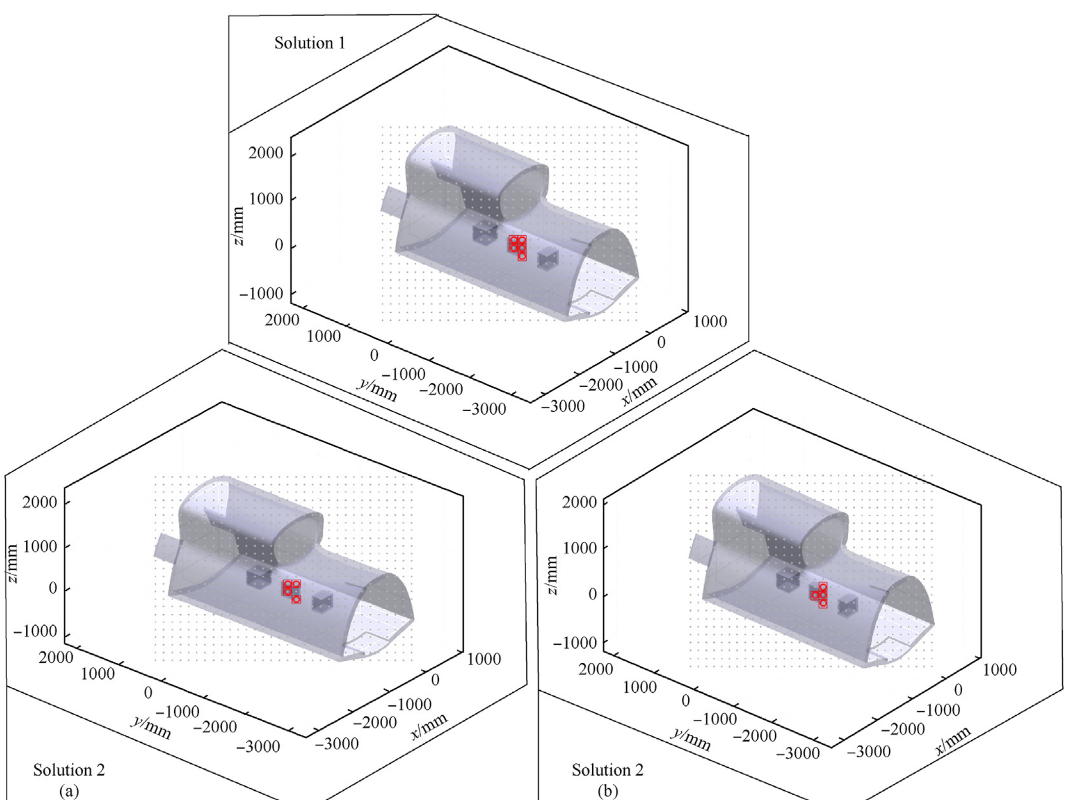

Fig. 15. 50°−30° aspect: optimized protection layouts.

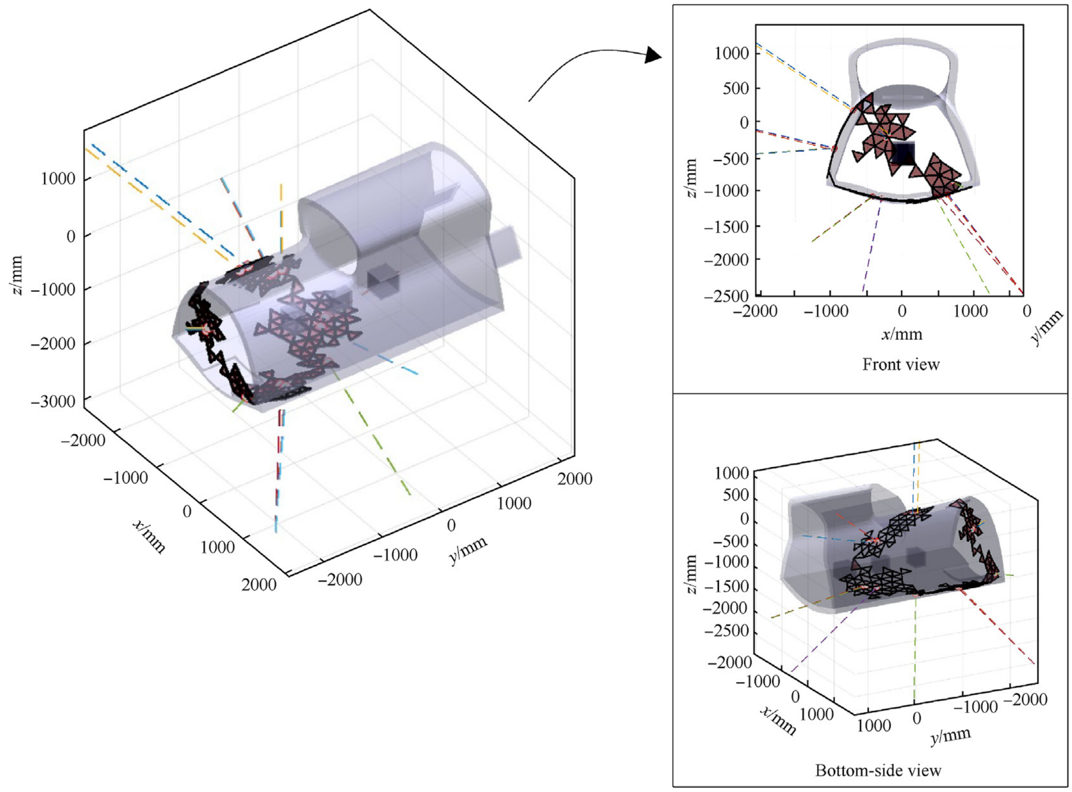

Fig. 17. Three-dimensional framework: optimized protection topology.

As can be seen from Fig. 17, although the skin exhibits an opening in the frontal area, the optimizer generates protective elements also in this region. This was expected since the skin mesh modeled to generate protective elements did not account for the opening. That is, in the interest of simplicity, the optimization grid was modeled through fully connected elements without borders. This limitation could be overcome by simply retaining openings in the optimization mesh.

4. Conclusions

In the last decades, the significance of survivability has grown considerably, emerging as a key design factor for modern platforms. Consequently, there has been a pressing need to develop tools aimed at enhancing the survivability characteristics of platforms. In this context, this work presents a general framework for generating topologically optimized protections for mitigating platform vulnerabilities. Multi-objective genetic algorithms have been used to generate protection layouts that simultaneously minimize (i) the vulnerabilities identified in the platform and (ii) the weight increase caused by adding protective elements. Two threats have been considered through two independent software packages. That is, protections against ballistic impact have been optimized using a vulnerability-driven 2D optimization framework, while an approach working in the 3D domain has been developed for protection against fragments generated by detonating threats. The integration of the vulnerability-driven tool with a genetic algorithm for the generation of optimized topology protections represents a novel approach in the field of topology optimization algorithms. However, using genetic algorithms within the vulnerability framework turned out not to be enough to converge to continuous topologies. In fact, the employed optimization process naturally converges to scattered protective structures, placing protective elements to diminish vulnerability and mass without considering structural continuity. For this reason, spatial filtering techniques have been implemented into both modules to drive the optimizer towards continuous protection layouts. Moreover, a custom mutation function based on Bernoulli trials and on the graph representation of the protection has also been proposed for the 3D framework to make convergence to continuous solutions easier. The implementation of this custom graph-based mutation function, which not only accelerates the convergence of the algorithm, but also enables the generation of continuous and optimized solutions, represents another innovative aspect of the article. A case study involving a representative portion of an aerial platform with three critical components has been considered to demonstrate the capabilities of the proposed optimization methods. The vulnerability algorithm with the genetic algorithm is applied first to the "simple" case, which involves the bullet as the impacting object, to verify its correct functionality in this scenario. Then, it is applied to a much more complex case study involving a 3D domain and fragments from detonating objects. In this more complex case study, different methods from the genetic algorithms, like for example the enumeration method, cannot be used to come up with a set of non-dominated solutions. In that case, guaranteeing continuous layouts that would simultaneously minimize two fitness functions would be impossible. Protections for projectiles flying in two different directions were independently generated through the 2D optimization framework. Several protection layouts were identified within the Pareto front, each characterized by a different ratio between the two objective functions. This feature of the proposed optimization tool allows designers to consider various solutions and choose the best one depending on design strategies and constraints. The optimized protection layouts present a limited mass for both the considered aspects, as shown in Figs. 11 and 14. The topology is optimized based on the assumption that the protection can fully block the impacting object. For this reason, if the user wants to determine the stiffness of the protection, should be considered that the stiffness is influenced by the thickness of the protection. Consequently, the appropriate thickness should be computed. Instead, protections against fragments were optimized considering several detonation points and distances around the platform, assuming a single flight direction for the detonating object. A single solution was generated by the multi-objective genetic algorithm. Increasing, eventually, the number of detonation points and distances and considering multi-angle detonations does not lead to protections that cover the entire surface of the aircraft since this would maximize the weight fitness function. Instead, in such a complex case, the optimal protection generated by the algorithm would cover most of the portion of the skin that makes fragments pass and intercept the target component(s) while reducing the amount of protection in other, non-critical, areas to the bare minimum. Thanks to the capability of the proposed method of generating continuous optimized solutions, able to adapt to any complex shape of a platform skin, it is possible to eventually install this protection layout coming out of the optimization framework in real-life applications.

However, it is important to acknowledge certain limitations of the proposed approach. For instance, when applied to more complex demonstrators, the algorithm exhibits lengthy computational times. This limitation could potentially be mitigated by porting the code to a different computing environment, for example Python©, where specialized libraries could be leveraged to execute the code on the GPU, resulting in a significant reduction of the computational time. Furthermore, it should be noted that the current software does not account for the weight of all the additional structures required for installing the protections. Future upgrades should address these limitations. Further work will consist of improving the optimization framework dedicated to ballistic impacts so to design protections that protect the identified critical components against projectiles coming from different directions. This would require generalizing the algorithm to the 3D domain and adopting a scheme similar to the one already implemented for fragments. Additionally, the proposed optimization method will be extended to also include damage from blast waves.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] B. RE.. The fundamentals of aircraft combat survivability: analysis and design. AIAA Education Series, AIAA 1985.
[2] K. C.. Are helicopters vulnerable ?, vol. 59. Australian Aviation; March 2005.
[3] A. JA. Survivability design features incorporated in the a-10a close air support weapon system. 1978.
[4] N. J. RE.. A case study of a combat aircraft's single hit vulnerability. 1986.
[5] Aitken Ed DA, Jones SL. A guide to fastgen target geometric modeling. Oct 1993.
[6] for Munitions JTCG. Covart 3.0-a simulation program for computation of vulnerable areas and repair time. 1989.
[7] Pei Y CT. Importance measure method for ranking the aircraft component vulnerability. 2014.
[8] Pei Y LL, Wang W. Ranking the vulnerable components of aircraft by considering performance degradations. 2016.
[9] Pei Y HQ, Song B. Shotline geometric description method for aircraft vulnerability calculation. 2009.
[10] P. Y, S. B, H. Q, O. B.. A direct simulation method for calculating multiple-hit vulnerability of aircraft with overlapping components. 2009.
[11] P. Y, S. B, H. QO. B.. Solving the combinatorial explosion problem when calculating the multiple-hit vulnerability of aircraft. 2006.
[12] Kincheloe W, et al. Fundamentals of ground combat system ballistic vulnerability/lethality. American Institute of Aeronautics and Astronautics, Inc.; 2009.
[13] Yoo C, Park K, Choi SY. The vulnerability assessment of ground combat vehicles using target functional modeling and fta. Int J Precis Eng Manuf 2016;17:651–8.
[14] Shin Y-H, Kwon J-I, Chung J-H. Development of a simplified vulnerability analysis program for naval vessel. Journal of the Society of Naval Architects of Korea 2013;50(6):383–9.
[15] Brownlow LC, Goodrum CJ, Sypniewski MJ, Coller JA, Singer DJ. A multilayer network approach to vulnerability assessment for early-stage naval ship design programs. Ocean Eng 2021;225:108731.
[16] Lee S, Go J, Kim C, You S. Development of a vulnerability assessment model for naval ships on a theater engagement analysis. Journal of the Korea Society for Simulation 2021;30(1):1–9.
[17] H. I, P. M, V. G, S. A, R. EM, R. K, R. MV. Risk and resilience analysis of public civil buildings against shelling with explosive sources in urban contexts. October 2020.
[18] O S. Materials with prescribed constitutive parameters: an inverse homogenization problem. 1994. p. 2313–29.
[19] O S. On the design of compliant mechanisms using topology optimization. 1997. p. 495–526.
[20] Sethian J WA. Structural boundary design via level set and immersed interface methods. 2000. p. 489–528.
[21] Kennedy J ER. Particle swarm optimization. 1995. p. 1942–8.
[22] Xie Y SG. A simple evolutionary procedure for structural optimization. 1993. p. 885–96.
[23] Xie Y SG. Evolutionary structural optimization. 1997.
[24] Kane MSC. Topological optimum design using genetic algorithms. 1996. p. 1059–88.
[25] Holland J. Adaptation in natural and artificial systems. 1975.
[26] Goldberg D. Genetic algorithms in search, optimization and machine learning. 1989.
[27] Rozvany G. Aims, scope, methods, history and unified terminology of computer-aided topology optimization in structural mechanics. 2001.
[28] Jenkins W. Structural optimization with the genetic algorithm. 1991. p. 418–22.
[29] Wang KTSY. Structural topology design optimization using genetic algorithms with a bit-array representation. 2005. p. 3749–70.
[30] Hajela P, Lee E, Lin C-Y. Genetic algorithms in structural topology optimization. Topology design of structures 1993:117–33.
[31] Wang S, Tai K, Wang MY. An enhanced genetic algorithm for structural topology optimization. Int J Numer Methods Eng 2006;65(1):18–44.
[32] Lomazzi L, Cadini F, Giglio M, Manes A. Vulnerability assessment to projectiles: approach definition and application to helicopter platforms. Defence Technology 2022;18(9):1523–37. https://www.sciencedirect.com/science/article/pii/S2214914721001550.
[33] Annunziata S, Lomazzi L, Giglio M, Manes A. Vulnerabilitydriven multi-objective topological optimization of aircraft protections. AIAS; 2023.
[34] Gurney RW. The initial velocities of fragments from bombs, shell, grenades. MD: Ballistic Research Laboratories Aberdeen Proving Ground; 1943.
[35] Charron YJ. Estimation of velocity distribution of fragmenting warheads using a modified gurney method. Air Force Institute of Technology; 1979. Ph.D. thesis.