POLITECNICO DI TORINO
Repository ISTITUZIONALE

Nonpenalty Machine Learning Constraint Handling Using {PSO}-{SVM} for Structural Optimization

(Article begins on next page)

20 April 2024

Research Article

Nonpenalty Machine Learning Constraint Handling Using PSO-SVM for Structural Optimization

Marco M. Rosso ⓘD, **Raffaele Cucuzza** ⓘD, **Fabio Di Trapani** ⓘD, and **Giuseppe C. Marano** ⓘD

Politecnico di Torino, DISEG, Dipartimento di Ingegneria Strutturale, Edile e Geotecnica, Corso Duca Degli Abruzzi, 24, Turin 10128, Italy

Correspondence should be addressed to Fabio Di Trapani; fabio.ditrapani@polito.it

Firstly formulated to solve unconstrained optimization problems, the common way to solve constrained ones with the meta-heuristic particle swarm optimization algorithm (PSO) is represented by adopting some penalty functions. In this paper, a new nonpenalty-based constraint handling approach for PSO is implemented, adopting a supervised classification machine learning method, the support vector machine (SVM). Because of its generality, constraint handling with SVM appears more adaptive both to nonlinear and discontinuous boundary. To preserve the feasibility of the population, a simple bisection algorithm is also implemented. To improve the search performances of the algorithm, a relaxation function of the constraints is also adopted. In the end part of this paper, two numerical literature benchmark examples and two structural examples are discussed. The first structural example refers to a homogeneous constant cross-section simply supported beam. The second one refers to the optimization of a plane simply supported Warren truss beam. The obtained results in terms of objective function demonstrate that this new approach represents a valid alternative to solve constrained optimization problems even in structural optimization field. Furthermore, as demonstrated by the Warren truss beam problem, this new algorithm provides an optimal structural design which represents also a good solution from the technical point of view with a trivial rounding-off that does not jeopardize significantly the optimization design process.

1. Introduction

Many optimum design engineering problems can be modelled as continuous nonlinear optimization problems, in which the search space is reduced due to the existence of various constraints:

$$
\begin{aligned}
&\min_{\mathbf{x} \in \Omega} \{f(\mathbf{x})\}, \\
&\text{s.t.} \quad g_q(\mathbf{x}) \le 0 \quad \forall q = 1, \ldots, n_q, \\
&\qquad h_r(\mathbf{x}) = 0 \quad \forall r = 1, \ldots, n_r,
\end{aligned}
\tag{1}
$$

in which $\mathbf{x} = [x_1, \ldots, x_n]^T$ is the design variable vector, $f(\mathbf{x})$ is the objective function (OF) to be minimized, and Ω is a box-type search space, defined as Cartesian products among the admissible intervals within a lower and an upper bound for each design variable $\Omega = [x_1^l, x_1^u] \times \cdots \times [x_n^l, x_n^u]$. The equality constraints can also be rewritten in order to have inequality constraints only, i.e., $g_p(\mathbf{x}) \le 0$, for $p = 1, \ldots, n_q, n_{q+1}, \ldots, n_p$, with $n_p = n_q + 2n_r$. The modern approaches to solve them are related to the Soft Computing techniques; in particular, the particle swarm optimization (PSO) algorithm [1] is adopted in this paper. The swarm exhibits an intelligent search behavior because of the social and cognitive interactions among the individuals of the population. This mechanism is inspired by natural aggregations such as bird flocking, fish schooling, or swarming of insects in nature. The PSO was successfully adopted in structural optimization fields and one can refer to, e.g., [2–6] for truss problems, to, e.g., [7, 8] for composite structures, to, e.g., [9], for shape optimization of arch dams under earthquake loading, or to, e.g., [10] for optimization of steel frames. The starting positions and velocities are randomly sampled using the Latin Hypercube Sampling (LHS) technique to generate

an initial population with minimum correlation between samples [11]. In the Newtonian dynamics-based algorithm, each ith particle (with $i = 1, \ldots, N$, in which N is the population size) is completely defined by the position ${}^{k}\mathbf{x}_i$ and the velocity ${}^{k}\mathbf{v}_i$ at the kth generation. The update rules for the next generation are the following:

$$
{}^{(k+1)}\mathbf{v}_i = {}^{k}\mathbf{v}_i + c_1^{(k+1)}\mathbf{r}_{1i} * \left[{}^{k}\mathbf{x}_i^{Pb} - {}^{k}\mathbf{x}_i \right] + c_2^{(k+1)}\mathbf{r}_{2i}^* \left[{}^{k}\mathbf{x}_i^{Pb} - {}^{k}\mathbf{x}_i \right],
\tag{2}
$$

$$
{}^{(k+1)}\mathbf{x}_i = {}^{k}\mathbf{x}_i + \tau^{(k+1)}\mathbf{v}_i \quad (\tau = 1),
\tag{3}
$$

in which the symbol $*$ denotes the term-by-term vector multiplication (Hadamard product, [12]). Each particle is an intelligent agent with memory of its best visited position until the current iteration in terms of minimal OF ${}^{k}\mathbf{x}_i^{Pb}$ (denoted as *pbest*), whereas the term ${}^{k}\mathbf{x}^{Gb}$ (denoted as *gbest*) is usually the best individual of the entire swarm in terms of minimal OF (fully connected neighborhood topology). It is possible to adopt also a local strategy limiting the sharing of information about the gbest to a certain number of neighbor particles [13–15]. The velocity update rule (2) depends on a inertia term related to the previous velocity vector, on a cognitive term related to the pbest attraction, and on a social term related to the gbest, which are randomly weighted by ${}^{(k+1)}\mathbf{r}_{1i}, {}^{(k+1)}\mathbf{r}_{2i} = \mathrm{rand}[0, 1]$. The positive scalar acceleration factors c_1 and c_2 are, respectively, called *cognitive* and *social* parameters. To protect the cohesion of the swarm, each velocity component is limited by a maximum value typically assumed as $\mathbf{v}^{\max} = \gamma (\mathbf{x}^u - \mathbf{x}^l)/\tau$, where $\tau = 1$ is a time-related parameter, whereas $\gamma \in [0.1, 1]$ (generally set to 0.5) defines how far a particle can move starting from its current position [13]. After the update phase, the algorithm checks if the particles are inside the box-search space. A stopping criterion is usually set through a maximum number of iterations L; however, the strictly required ones to reach the optimum are not know a priori; therefore, one can refer to the suggestions of [16]. Later, to improve the exploration capacity of the swarm, Shi and Eberhart [17, 18] introduced an inertia weight term ${}^{k}w$ multiplied to the previous velocity in (2), which changes during iterations from an initial value ${}^{0}w$ to a final one ${}^{L}w$ with a linearly decreasing law. Some other inertia weight laws were proposed during the years [14]. The performance of the algorithm is strongly affected by the choice of the parameters such as the swarm size N, usually set in a range of $[20, 100]$ with $n \le 30$, or the acceleration factors, which are usually assumed statically fixed to $c_1 = c_2 = 2$ [13]. In this study, it is assumed that all of them are constant values equal to $c_1 = c_2 = 2$, ${}^{0}w = 0.90$, and ${}^{L}w = 0.40$ [2].

Despite several studies (e.g., [19]) demonstrating that this optimizer has a good convergence rate, the PSO, such as many other Soft Computing techniques, was originally proposed to solve unconstrained optimization problems. Therefore, the implementation of some effective constraint-handling mechanisms is considered a crucial issue for all biological inspired optimizers [20–23]. These methods have been classified by different authors into certain categories (see, for instance, the state-of-the-art review by [21, 24, 25]): penalty

functions-based methods; methods based on special operators and representations; methods based on repair algorithms; methods based on the separation between OF and constraints; hybrid methods. The most adopted method due to its simplicity is the exterior penalty approach which allows to convert the problem in an unconstrained one [26, 27]. Many different approaches such as the death, static, dynamic, or adaptive penalty functions have been proposed in time, e.g., see [26]. A proper choice of the constraint-handling mechanism affects the performance of the algorithm, and one of the critical issues to take into account is the preservation of the diversity of the population [28]. The brutal elimination of the unfeasible particles, such as in the death penalty rule, can jeopardize the exploration performances due to a loss of information [21, 29]. Moreover, it has been verified that several traditional penalty-based approaches may not be adequate to deal with highly complex search spaces, especially for problems in which several constraints are actively close to the optimum [30]. Specific constraint handling operators have been implemented in PSO, as presented in [26], and can be classified in four main groups: penalty-based mechanisms; separatist mechanisms; hybrid mechanisms; other constraint handling mechanisms (such as Del Valle's ranking approach). Many variant of the PSO have been proposed in the years, e.g., the ILS-PSO by [31] which adopts a local search operator to deal with equality constraints or the PSO + by [27] which is based on the preservation of the feasibility. In the last decades, to overcome the drawbacks of a single heuristic approach, many hybridization have been proposed such as with the simplex method [32], with sequential quadratic programming [6], with radial basis function approximations [33], with fuzzy logic [9, 34], with neural networks [35], with parallel computing techniques [36], with genetic algorithm GA-PSO [12], with differential evolution algorithm DEPSO [37], with ant colony optimization [5], and with many other algorithms such as exposed in [14]. In the following, a new nonpenalty constraint handling approach based on the support vector machine (SVM) is reported in detail, and, after that, two numerical problems are discussed to verify the convergence properties of the new PSO-SVM algorithm. Subsequently, the proposed PSO-SVM is adopted into the structural optimization field analyzing two examples, respectively, related to the size optimization of a simply supported beam and to the size and shape optimization of a Warren truss beam.

2. Constraint Handling in the Proposed Framework: The PSO-SVM Approach

In this paper, a new nonpenalty-based constraint handling approach is implemented using the SVM. This machine learning approach is based on statistical learning theory [38]. During the learning process, the machine learns from examples contained in a known training dataset which could be composed of both input and output data (supervised) or only input data (unsupervised) [39]. The SVM is able to map the input data to another space (feature space) usually with a higher dimension [40] where the data are linearly separable and it will search the optimal separating hyperplane according to the principle of the maximal margin [41]. The

transformation is performed by a Kernel Function which represents the inner product in the feature space (the so-called kernel trick procedure [39]). Then, after the remapping of the optimal separating hyperplane in the starting design space, the result of this process is the definition of a predictive model that could map the output of other new input data. If the output is a variable from a finite set, which represents the class (or category or label) of the input data, the problem is called classification (or pattern recognition) problem [39, 42]. After a training phase on the current positions of the swarm, the new trial positions are labelled as feasible or unfeasible using the trained predictive model. In this case, since the algorithm has to separate feasible positions from unfeasible ones, it is performing a binary classification problem. The SVM is based on the inner product in feature space, so it is possible to avoid explicitly writing the transformation of training data into the feature space, but it is possible to directly operate defining a proper kernel function, as stated in [39]. This approach is called in [40] as implicit mapping or implicit embedding. According to [43], typical widely spread kernel functions are polynomial kernels and Gaussian kernel [39, 40] which is the isotropic form of the radial basis kernels, RBF [42]. In the proposed approach, the Gaussian kernel is adopted. It can occur that training data are nonseparable, i.e., which cannot be linearly divided without committing a certain error [41]. In this case, the SVM operates with a soft margin defining a hyperplane minimizing the error. This is governed by a penalty coefficient $C > 0$ also know as regularization parameter which controls the tolerance on training classification errors and the margin position related to the complexity of the model [39, 42, 43]. For the proposed algorithm, it was found that a good trade-off between accuracy and computational effort is obtained using C between 100 and 1000.

The new proposed approach is implemented in Matlab ® code, and its aim is to integrate the PSO algorithm with SVM in order to separate feasible positions from the unfeasible ones and reduce the search space.

In Figure 1(a), the flowchart of the proposed algorithm is shown. The initial random positions, which are obtained by the LHS technique, are labelled with respect to the constraints, and the SVM is able to construct the nonlinear boundary of the feasible region, thanks to the kernel trick. In order to get very accurate results still from the beginning, it would be necessary to sample a very huge swarm size. However, since the particles move in another configuration at each iteration, new available positions can be used to train a new SVM model at each iteration; therefore, this incremental approach is adopted in the PSO-SVM. Since the constraints are defined as $g_q \leq 0$, if at least one of them is greater than zero the entire n-tuple (which is a single individual candidate solution) is labelled as unfeasible ($y_i = -1$), otherwise it is labelled as feasible ($y_i = +1$). Unfortunately, sometimes, the SVM really struggles to work properly, e.g., with a very narrow search space. In fact, the preservation of the feasible points only, into a wide unfeasible space, leads to instabilities such as overfitting or underfitting problems of the SVM. To improve the performance of the algorithm, a relax constraint function $\psi_i(k)$

is defined, wherein the subscript i refers to ith constraint and the k refers to the current iteration. This approach leads to enlarge the real feasible space to a fictitious wider one because of a relaxation of the actual constraints. This means that the real constraints are "moved" from their actual position to a fictitious one through a proper choice of $\psi_i(k)$ which is not trivial to generalize. This procedure acts as substituting the original inequality constraints with the following relaxed constraints:

$$g_{i,k}(\mathbf{x}) \leq \psi_i(k) \longrightarrow g_{i,k}(\mathbf{x}) - \psi_i(k) \leq 0. \qquad (4)$$

The $\psi_i(k)$ factor is tuned on the standard deviation $\sigma_{u,k}$ of the amount of all unfeasible points detected in all generations until the current one. This approach allows a better exploration during the first generations and, subsequently, it could be forced to zero implementing a decreasing function, which directly depends on the current generation k and the number of maximum generations $k_{\max}$. Defining a further reduction factor λ of the standard deviation directly chosen by the user, the following relaxation functions are implemented:

(i) Constant relax: $\psi_i(k) = \lambda \sigma_{u,k}$

(ii) Piecewise constant relax: $\psi_i(k) = \begin{cases} \lambda \sigma_{u,k}, & \text{if } k \leq k_{\max}/20, \text{otherwise} \end{cases}$

(iii) Linear decreasing relax: $\psi_i(k) = \lambda \sigma_{u,k} - \lambda \sigma_{u,k}/k_{\max} k$

(iv) Piecewise linear decreasing relax:
$\psi_i(k) = \begin{cases} \lambda \sigma_{u,k} - 2\lambda \sigma_{u,k}/k_{\max} k, & \text{if } k \leq k_{\max}/2 \\ 0, & \text{otherwise} \end{cases}$

(v) Parabolic decreasing relax: $\psi_i(k) = \lambda \sigma_{u,k} - \lambda \sigma_{u,k}/k_{\max}^2 k^2$

(vi) Piecewise parabolic decreasing relax:
$\psi_i(k) = \begin{cases} \lambda \sigma_{u,k} - 4\lambda \sigma_{u,k}/k_{\max}^2 k^2, & \text{if } k \leq /20, \text{otherwise} \end{cases}$

The proposed algorithm implements the SVM with soft margin with regularization parameter $C = 100$. Indeed, with respect to a hard margin, to speed up the algorithm, it is numerically convenient to adopt a high box-constraint value, e.g., between 100 and 1000, allowing a certain misclassification rate but assuring all the possible solutions lie in the feasible space boundary respecting the constraints with a certain engineering tolerance. After the training phase, the unfeasible points of the initial population have to be randomly resampled in the feasible region adopting the LHS technique, and, therefore, they are discarded and sampled again until their label becomes $y_i = +1$. At this point, the initial velocity of the particles is randomly sampled always using LHS, and then, the evolutionary phase of the algorithm can begin. The velocity for the next generation is calculated as in equation (2), but it is necessary to check if it respects the allowable maximum velocity ("Max Velocities Correction" block in Figure 1(a)). The next block "Max Position Correction" is a fundamental step because it is the check of the feasibility of the position given by (3), and, for greater clarity, it is expanded in Figure 1(b). First of all, a temporary final position is computed with equation (3) for each particle, and the algorithm checks if it lie inside the box-search space or moves it to the nearest boundary of Ω if it lies outside. If the
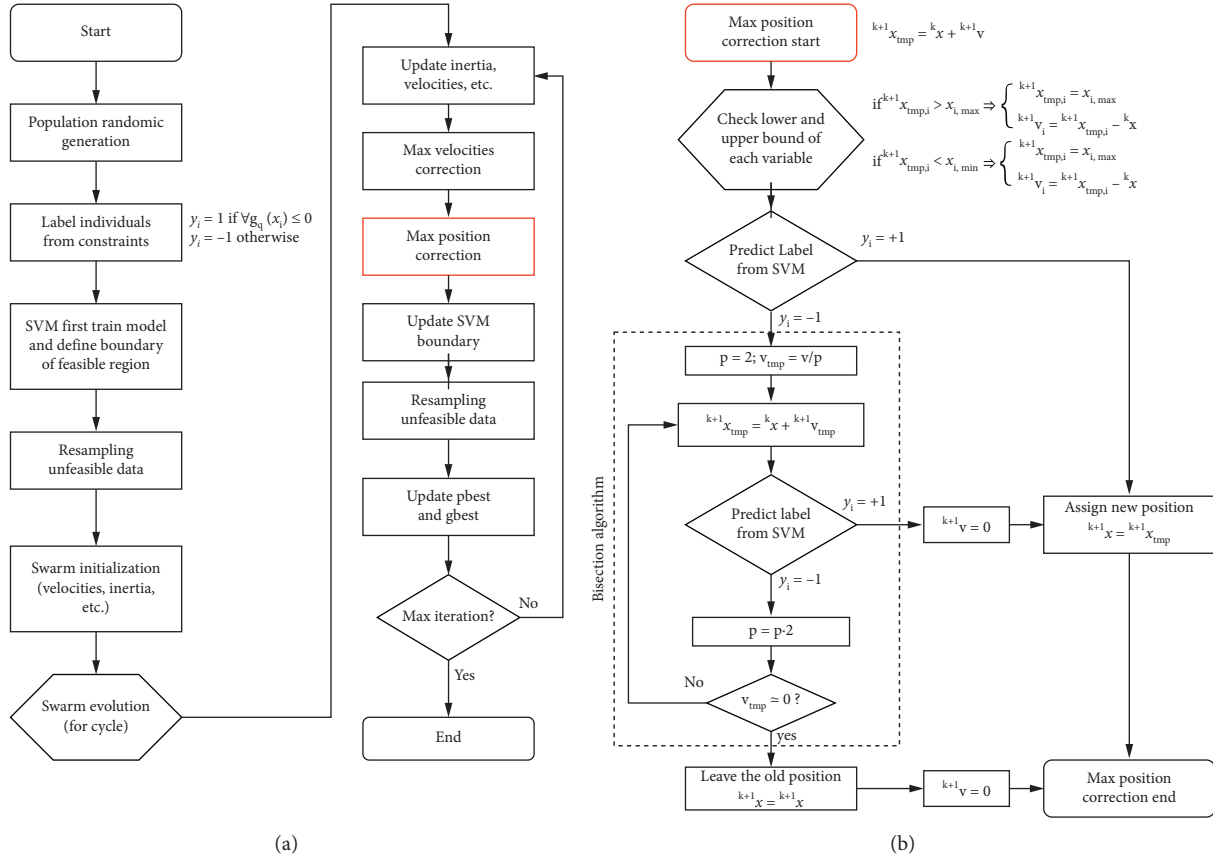
FIGURE 1: Flowchart of the proposed PSO-SVM algorithm. (a) General flowchart of the proposed algorithm; (b) flowchart of "Max Position Correction" block.

temporary position label is $y_i = +1$, the new temporary position lies in the feasible region and it is stored, otherwise a simply bisection approach is performed. The velocity vector which leaded to the temporary position is firstly divided by $p = 2$ and a new temporary final position is computed. If the label of this new temporary position becomes positive, so this new position is accepted and stored for the next generation. Instead, if it is still negative, the algorithm will redouble p, getting a new velocity vector and a new temporary position to label and so on. When all the elements of temporary velocity vector tend to zero, it is possible to leave the original position for the next generation. In this way, the particle can move only if the landing position is into the feasible region, otherwise it stands still. After that, new positions are available to increase the database set and train a new refined SVM model getting a new boundary of the feasible region. It is possible that some of the points which were inside the previous boundary may lie outside of the new boundary. In this case, it is necessary to resample them as before using LHS until their label becomes positive. Despite this, the latter naive approach may slow the algorithm, and it is beneficial for the diversity of the population, allowing new starting search positions. The update of the gbest is performed considering only the real feasible points respect to the actual constraints and not the relaxed ones. This procedure ensures that the algorithm gets good and feasible results also using a constant relax constraint function during

the generations. The stopping criterion can be set to the achievement of a maximum number of iterations k_{max}. A further strategy to improve the behaviour of the PSO is reducing the maximum possible velocity range of the particles, governed by γ, updating this latter during generations. For example, it is possible to set $\gamma/2$ starting from $k_{max}/3$ and to the minimum value ($\gamma = 0.1$) in the last generations to improve the exploitation. In the following, the PSO-SVM is firstly tested on some numerical mathematical literature problems and statements are reported in Appendix. To make some comparison of the results, in these last examples, the OF value is compared with the genetic algorithm of Matlab ® and the PSO with penalty approach provided from the code proposed by [44].

3. Numerical Example 1: Sickle Problem [45]

The first numerical example (from literature [45]) is the benchmark test 1 stated in the Appendix also known as sickle function. Since this optimization problem has two design variables, it is possible to follow the optimization process of the PSO-SVM in the design variables' plane. In Figure 2(a), the objective function and the constraints are graphically represented as a three-dimensional graph. The projection on the design variables' plane of the feasible region is really narrow because it is formed by the space between the two constraints' parabolas. For this problem, a population size of
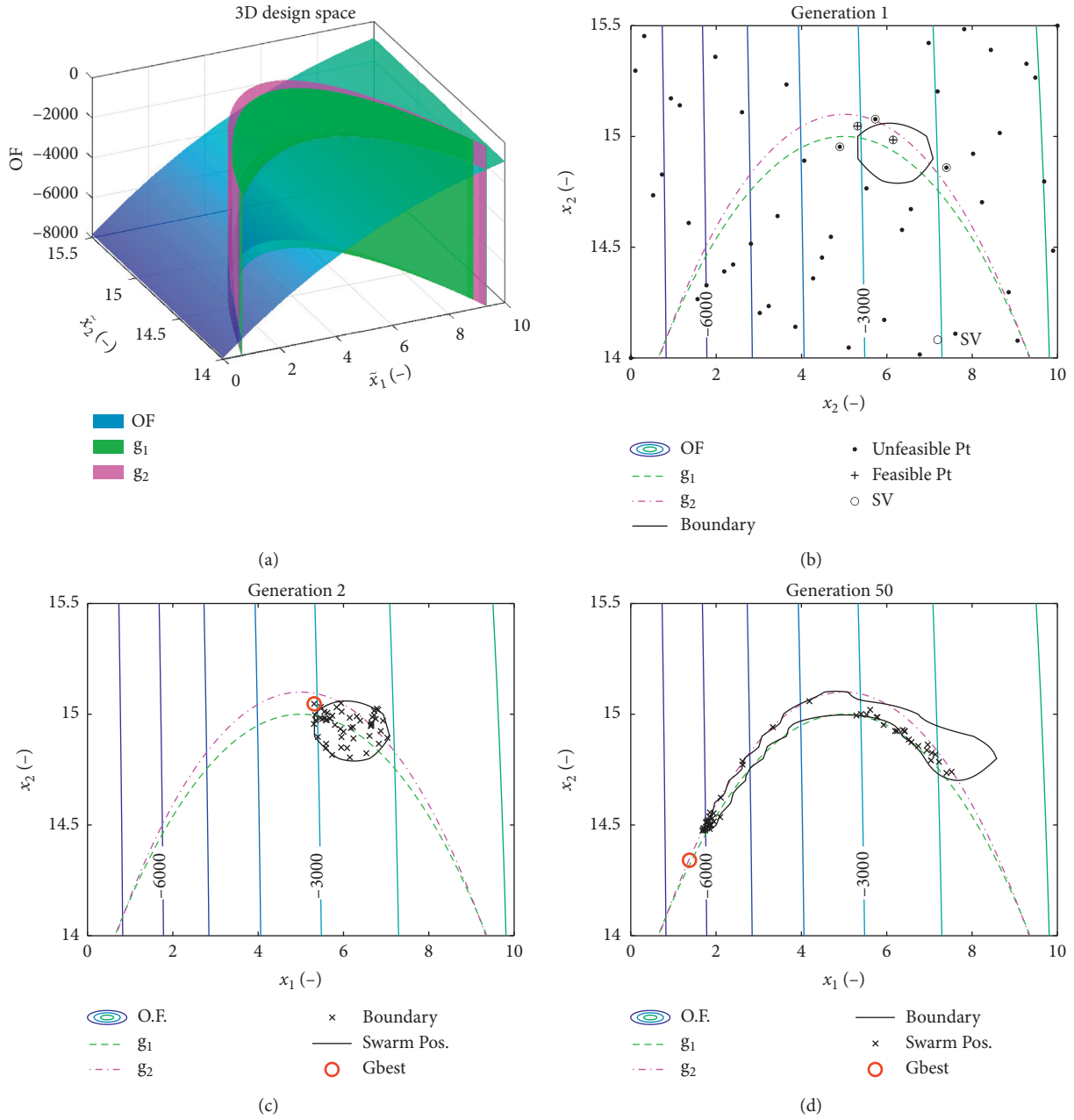
(a)
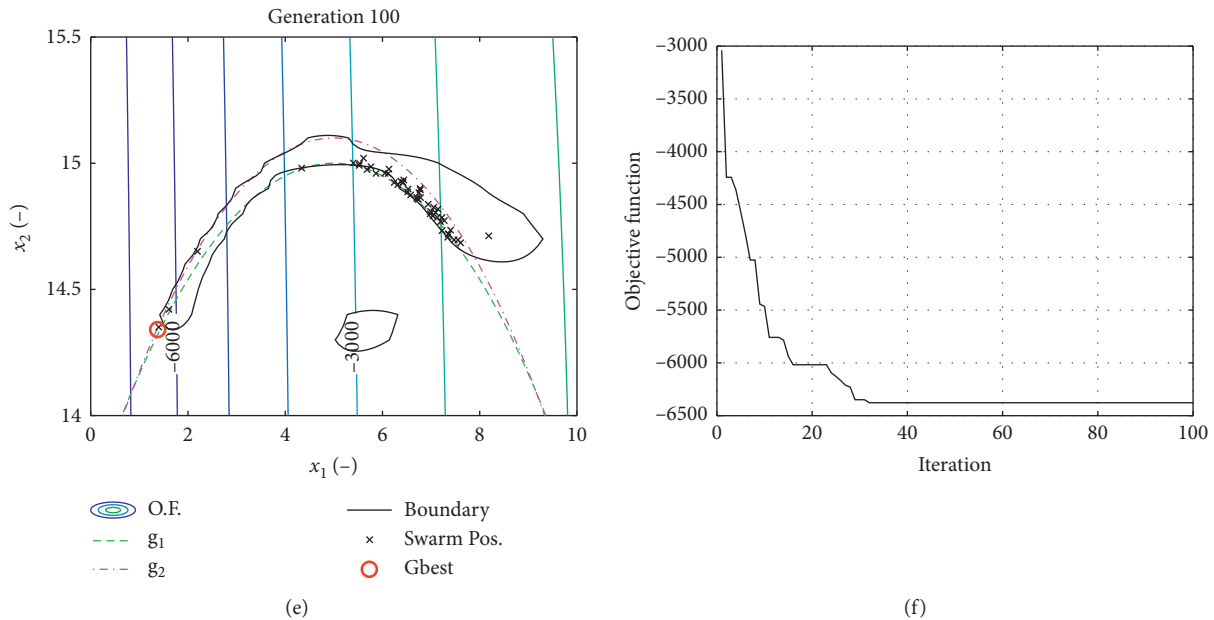
(b)

(c)

(d)

FIGURE 2: Continued.

FIGURE 2: Example 1 (sickle problem [45]): case no relax constraints' function. (a) 3D graph of sickle problem design space. (b) Generation 1. (c) Generation 2. (d) Generation 50. (e) Generation 100. (f) Objective function history.

50 individuals is set, and in Figure 2(b), the first random generation is shown and the support vectors are emphasized. After that, the unfeasible points are resampled until all the population falls inside the SVM-based boundary. At each generation, new points are added to the SVM training data and the boundary is improved. Since the search space is really narrow, running the algorithm with no relax constraint function results in poor performance in defining the SVM boundary (black solid line) with respect to the actual one (dashed lines), as shown in Figure 2. It is possible to improve the performance in terms of the OF decreasing history adopting the relax of constraints, e.g., the constant relax function, as shown in Figure 3. In this latter example, the reduction factor of the standard deviation of the unfeasible point is chosen as $\lambda = 0.5$. The PSO-SVM performances using constant relax function is compared with other algorithms in Table 1. The comparison is carried out in terms of the objective function value to the optimal solution after 50 runs with $k_{max} = 100$ each run and collecting mean value μ, standard deviations σ, best OF, and worst OF. As one can check, the convergence of the new proposed method is satisfied since the objective function value is getting close to the global optimum with a little standard deviation as the other existing methods.

4. Numerical Example 2: Five Design Variables' Optimization Problem

The second numerical example, in which the statement is reported in Appendix, is a literature benchmark test optimization problem with five design variables and six constraints. In this example, a comparison between different histories of the objective function using different relax constraint functions is performed. As shown in Figure 4, all

the examples give a good result and tend to converge to the exact solution with different decreasing rates. In general, one can notice that, in piecewise functions, in the first iterations, the algorithm generally boosts the exploration instead of the exploitation, which is usually enhanced in the second half with a zero relax coefficient. This feature is important because it can affect the performance of the proposed algorithm with different kinds of problems and the user needs to try different relax functions in order to find the most suitable for the specific problem to solve. In Tables 2 and 3, a comparison with the GA and the PSO-Penalty is performed in terms of OF value and optimal design points, running the codes 50 times with $k_{max} = 100$ each run and collecting mean value μ, standard deviation σ, worst, and best. Also, with this more complex optimization problem, the convergence of the new PSO-SVM method is satisfied, getting an OF value close to the global optimum with a little standard deviation as the other existing methods.

5. Structural Example 1: Simply Supported Beam

In Figure 5, it is considered as an ideal simply supported beam of length L with a constant cross-section $A = b \cdot h$ loaded with a distributed constant load q which is supposed to be much greater than the self-weight for the sake of simplicity. The aim is to minimize the weight of this structure respecting the tensional constraints and maximum deflection constraint due to only the q load. The self-weight is proportional to the volume V of the material, as stated by [47]. The objective function is $f(\mathbf{d}) = \rho V = \rho AL$, where $\mathbf{d}$ is the design vector and ρ is the material density which is supposed to be constant. In this case, only stress constraints on normal stress σ, tangential stress τ, and maximum deflection $v(z)$ are considered. This is a typical
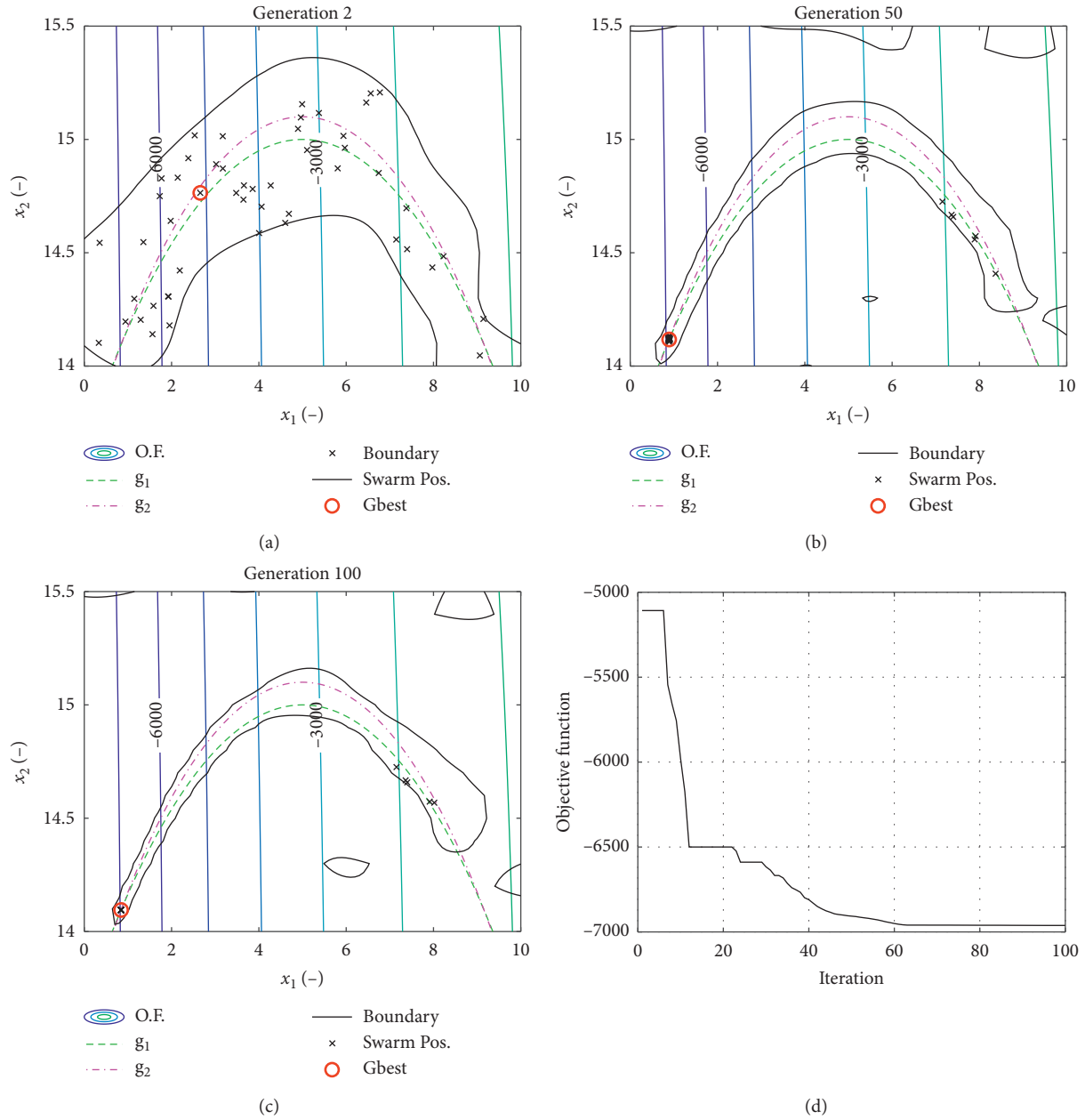
FIGURE 3: Example 1 (sickle problem [45]): case constant relax constraints' function. (a) Generation 2. (b) Generation 50. (c) Generation 100. (d) Objective function history.

TABLE 1: Numerical Example 1 (sickle problem [45]): comparison of PSO-SVM, GA, and PSO-penalty.

	PSO-SVM	PSO-Penalty	GA
μ_{x_1}	0.8433	0.8721	0.8502
σ_{x_1}	0.0004	$1.121e-15$	0.0172
μ_{x_2}	14.0952	14.1091	14.0986
σ_{x_2}	0.0002	$8.972e-15$	0.0087
μ_{OF}	$-6.9614e+03$	$-6.9291e+03$	$-6.9537e+03$
σ_{OF}	0.4609	$4.59e-12$	19.2968
Best OF	$-6.9595e+03$	$-6.93e+03$	$-6.9618e+03$
Worst OF	$-6.96e+03$	$-6.93e+03$	$-6.8547e+03$

sizing optimization problem. The design vector $\mathbf{d} = [d_1, d_2]^T$ contains the design variables which are changed during the optimization process, i.e., in this case, $d_1 = b$ and $d_2 = h$. Since the cross-section as well as the cross-sectional dimensions must be greater than zero, it implies the presence of a new constraint to satisfy. Performing an elastic analysis, the maximum allowable stress is the yielding stress σ_y, and it is possible to use the Navier Formula and the Jourawsky Formula for the normal and tangential stress, respectively. The maximum moment is in the middle span, whereas the maximum shear force is in correspondence of the supports $z = 0$ and $z = L$:
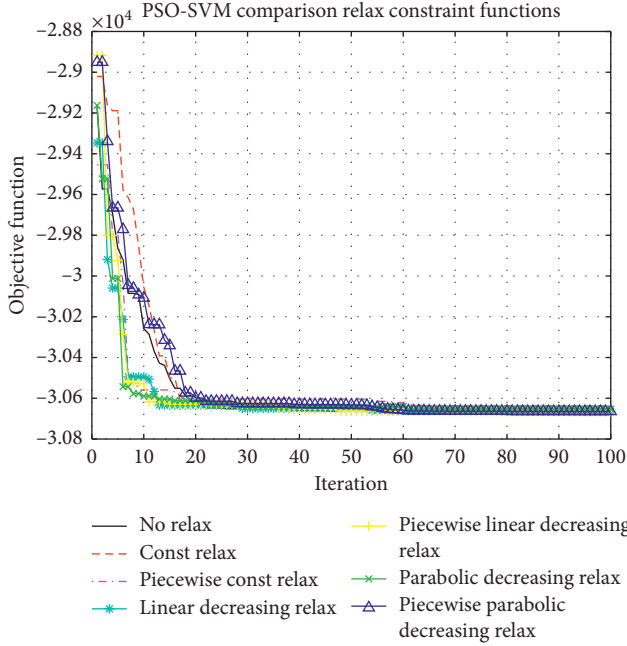
FIGURE 4: Numerical example 2: objective value history comparison among different relax constraint functions for a single run.

$$M_{Ed}\left(z = \frac{L}{2}\right) = \frac{qL^2}{8}, V_{Ed}(z = 0, L) = \frac{qL}{2}. \qquad (5)$$

Recalling the elastic resistance modulus for a rectangular section and using the Navier formula, it is possible to write the maximum normal stress in the middle span as

$$W_{el} = \frac{bh^2}{6} = \frac{d_1 d_2^2}{6}, \sigma\left(z = \frac{L}{2}\right) = \frac{M_{Ed}}{W_{el}} = \frac{3}{4}\frac{qL^2}{d_1 d_2^2}. \qquad (6)$$

Using the Jourawsky formula, it is possible to write the maximum tangential stress in the middle of cross-section $y = 0$ (parabolic tangential stress diagram on a rectangular section) as

$$\tau(z = 0, z = L) = \frac{V_{Ed} S_x^*(y = 0)}{I_x b} = \frac{3}{2}\frac{V_{Ed}}{bh} = \frac{3}{4}\frac{qL}{d_1 d_2}. \qquad (7)$$

In order to take into account both normal and tangential stresses, it is necessary to refer to a yield criterion. In this case, the Von Mises yield criterion is adopted:

$$\sqrt{\sigma^2(z) + 3\tau^2(z)} \le \sigma_{id}. \qquad (8)$$

Substituting (6) and (7) into (8), respectively, it is possible to obtain the two expressions of the stress constraints in middle span $z = L/2$ (pure moment) and $z = 0, L$ (pure shear):

$$\frac{3}{4}\frac{qL^2}{d_1 d_2^2} \le \sigma_{id}, \frac{3}{4}\frac{qL}{d_1 d_2} \le \frac{\sigma_{id}}{\sqrt{3}}. \qquad (9)$$

The maximum deflection $v(z = L/2)$ can be calculated using the virtual work principle obtaining

$$v\left(z = \frac{L}{2}\right) = \frac{5}{384}\frac{qL^4}{EI_x} = \frac{5}{32}\frac{qL^4}{Ed_1 d_2^3}. \qquad (10)$$

The complete statement of the optimization problem is the following:

$$\begin{aligned} &\min\ f(d_1, d_2) = d_1 d_2, \\ &\quad d_1 > 0, d_2 > 0, \\ &\quad \frac{3}{4}\frac{qL^2}{d_1 d_2^2} - \sigma_{id} \le 0, \\ &\text{s.t.}\ \ \frac{3}{4}\frac{qL}{d_1 d_2} - \frac{\sigma_{id}}{\sqrt{3}} \le 0, \\ &\quad \frac{5}{32}\frac{qL^4}{Ed_1 d_2^3} - v_{\max} \le 0, \end{aligned} \qquad (11)$$

where the constant ρL is dropped by the objective function as stated in similar problems analyzed in [48], σ_{id} is the ideal Von Mises normal stress, and $v_{\max}$ is the maximum deflection admissible by reference design codes, i.e., in this case, it is fixed to $v_{\max} = L/250$. It is possible to define the fixed variables' vector $\mathbf{b} = [b_1, b_2, b_3, b_4, b_5]^T$ which contains problem data which does not change during the optimization process, i.e., in this case, $b_1 = q$, $b_2 = L$, $b_3 = \sigma_{id}$, $b_4 = E$, and $b_5 = v_{\max}$. Because of the amount of fixed parameters, it is more convenient to work with a dimensionless form of the same problem. Posing $\tilde{d}_1 = b/L$ and $\tilde{d}_2 = h/L$, the new dimensionless objective function become $\tilde{f}(\tilde{d}_1, \tilde{d}_2) = f(d_1, d_2)/L^2 = \tilde{d}_1 \tilde{d}_2$, whilst the normal stress constraint (9), the shear constraint (9), and the deflection constraint (10) become, respectively, as

$$\frac{3}{4}\left(\frac{q}{L\sigma_{id}}\right)\frac{1}{\tilde{d}_1 \tilde{d}_2^2} \le 1, \frac{3\sqrt{3}}{4}\left(\frac{q}{L\sigma_{id}}\right)\frac{1}{\tilde{d}_1 \tilde{d}_2} \\ \le 1, \frac{5 \cdot 125}{16}\left(\frac{q}{EL}\right)\frac{1}{\tilde{d}_1 \tilde{d}_2^3} \le 1. \qquad (12)$$

It is useful to define two dimensionless nonnegative parameters, collected in $\tilde{b}$, which completely characterize the fixed variables of the problem:

$$\psi_\sigma = \frac{q}{L\sigma_{id}}, \psi_E = \frac{q}{EL}. \qquad (13)$$

TABLE 2: Numerical example 2: comparison among PSO-SVM with different relax constraint functions (check Appendix and [46]).

	Comparison among constraint relax functions in PSO-SVM						
Design Var.	No relax	Const. relax	Piecewise Const.	Lin. relax	Piecewise lin.	Parabolic relax	Piecewise Par.
μ_{x_1}	78.0004	78.0086	78.0026	78.0009	78.0006	78.0027	78.0077
σ_{x_1}	0.0013	0.0243	0.0120	0.0038	0.0041	0.0100	0.0526
μ_{x_2}	33.0062	33.0040	33.0056	33.0104	33.0107	33.0740	33.0052
σ_{x_2}	0.0158	0.0183	0.0143	0.0438	0.0322	0.2274	0.0158
μ_{x_3}	30.0027	29.9985	30.0037	30.0009	30.0038	30.0366	30.0027
σ_{x_3}	0.0101	0.0105	0.0125	0.0235	0.0178	0.1217	0.0160
μ_{x_4}	44.6569	44.2360	44.3643	44.9858	44.8002	44.0814	44.6662
σ_{x_4}	1.5986	2.9630	2.0027	0.0617	1.0734	3.2862	1.3233
μ_{x_5}	36.8981	37.0755	37.0136	36.7671	36.8360	37.0446	36.8931
σ_{x_5}	0.6444	1.1956	0.8036	0.0652	0.4310	1.3843	0.5324
μ_{OF}	$-3.0655e+04$	$-3.0644e+04$	$-3.0647e+04$	$-3.0664e+04$	$-3.0659e+04$	$-3.0634e+04$	$-3.0655e+04$
σ_{OF}	43.0187	78.9205	53.7773	3.9918	28.9931	87.9967	36.2059
Best OF	$-3.0666e+04$	$-3.0666e+04$	$-3.0666e+04$	$-3.0666e+04$	$-3.0666e+04$	$-3.0666e+04$	$-3.0666e+04$
Worst OF	$-3.0375e+04$	$-3.0185e+04$	$-3.0449e+04$	$-3.0643e+04$	$-3.0468e+04$	$-3.0186e+04$	$-3.0452e+04$

TABLE 3: Numerical example 2: results from PSO-SVM without relax constraints, PSO-Penalty, and GA (check Appendix and [46]).

	PSO-SVM	PSO-Penalty	GA
μ_{x_1}	78.0004	78	78.0004
σ_{x_1}	0.0013	0	0.0024
μ_{x_2}	33.0062	33	34.2398
σ_{x_2}	0.0158	0	0.7052
μ_{x_3}	30.0027	29.9967	30.801
σ_{x_3}	0.0101	$2.15E-14$	0.378
μ_{x_4}	44.6569	45	45
σ_{x_4}	1.5986	0	0
μ_{x_5}	36.8981	36.7736	34.8023
σ_{x_5}	0.6444	$0.00e+00$	0.9057
μ_{OF}	$-3.0655e+04$	$-3.0665e+04$	$-3.0531e+04$
σ_{OF}	43.0187	$1.47E-11$	65.8031
Best OF	$-3.0666e+04$	$-3.0665e+04$	$-3.0660e+04$
Worst OF	$-3.0375e+04$	$-3.0665e+04$	$-3.0378e+04$

FIGURE 5: Problem formulation: simply supported beam with constant cross-section.

Finally, the dimensionless version of problem (11) is given by

$$\min \tilde{f}(\tilde{d}_1, \tilde{d}_2) = \tilde{d}_1 \tilde{d}_2,$$

$$\text{s.t. } \tilde{d}_1 > 0, \tilde{d}_2 > 0,$$

$$\frac{3}{4} \psi_\sigma \frac{1}{\tilde{d}_1 \tilde{d}_2^2} \le 1,$$

$$\frac{3\sqrt{3}}{4} \psi_\sigma \frac{1}{\tilde{d}_1 \tilde{d}_2} \le 1, \quad (14)$$

$$\frac{5 \cdot 125}{16} \psi_E \frac{1}{\tilde{d}_1 \tilde{d}_2^3} \le 1.$$

To solve this problem, the PSO-SVM is adopted with the piecewise linear decreasing relax constraint function with a user coefficient λ for standard deviation of the unfeasible po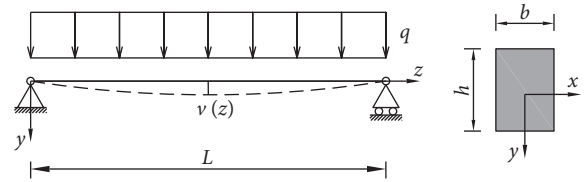ints fixed to $\lambda = 0.05$. For academic purposes, in order to graphically analyze the behavior of the constraint handling, the dimensionless parameters are fixed as $\psi_\sigma = \psi_E = 0.2$, looking for an optimal solution in the design domain for the dimensionless design variables as $0 \le \tilde{d}_1 \le 1.5$ and $0 \le \tilde{d}_2 \le 1$. In this way, with this particular choice of the dimensionless parameters, the constraints intersect each other creating discontinuous nonlinear boundary of the feasible region. The population size is always 50 individuals and the maximum iterations are 100. After 50 runs, the results show a quite great variability of the design variables but always at the same objective function value. This fact enlightens the presence of a front of possible optimal solutions. In fact, as shown in the graphical representations in Figure 6, for this specific choice of $\tilde{b}$, only two constraints are active, and there exists a region of optimal solutions on the τ constraint boundary line. The comparison in Table 4 among PSO-SVM, PSO-Penalty, and GA shows the mean value and the standard deviation of the best objective function value obtained after 50 runs.

In order to find the entire front of the all optimal possible solutions for this specific problem, it is necessary to find all the pairs $(\tilde{d}_{1,opt}, \tilde{d}_{2,opt})$ posing the objective function as $f(\tilde{d}_1, \tilde{d}_2) = \tilde{d}_1 \cdot \tilde{d}_2 \approx 0.2598$. As one can see in Figure 6, this optimal front corresponds to a part of the τ constraint (12) posed as an equality. Referring to the $\tilde{d}_1$ optimal possible values, the optimum front is upper bounded from the σ constraint (12) posed as an equality and lower bounded from the box search space limits. Then, to calculate the optimal
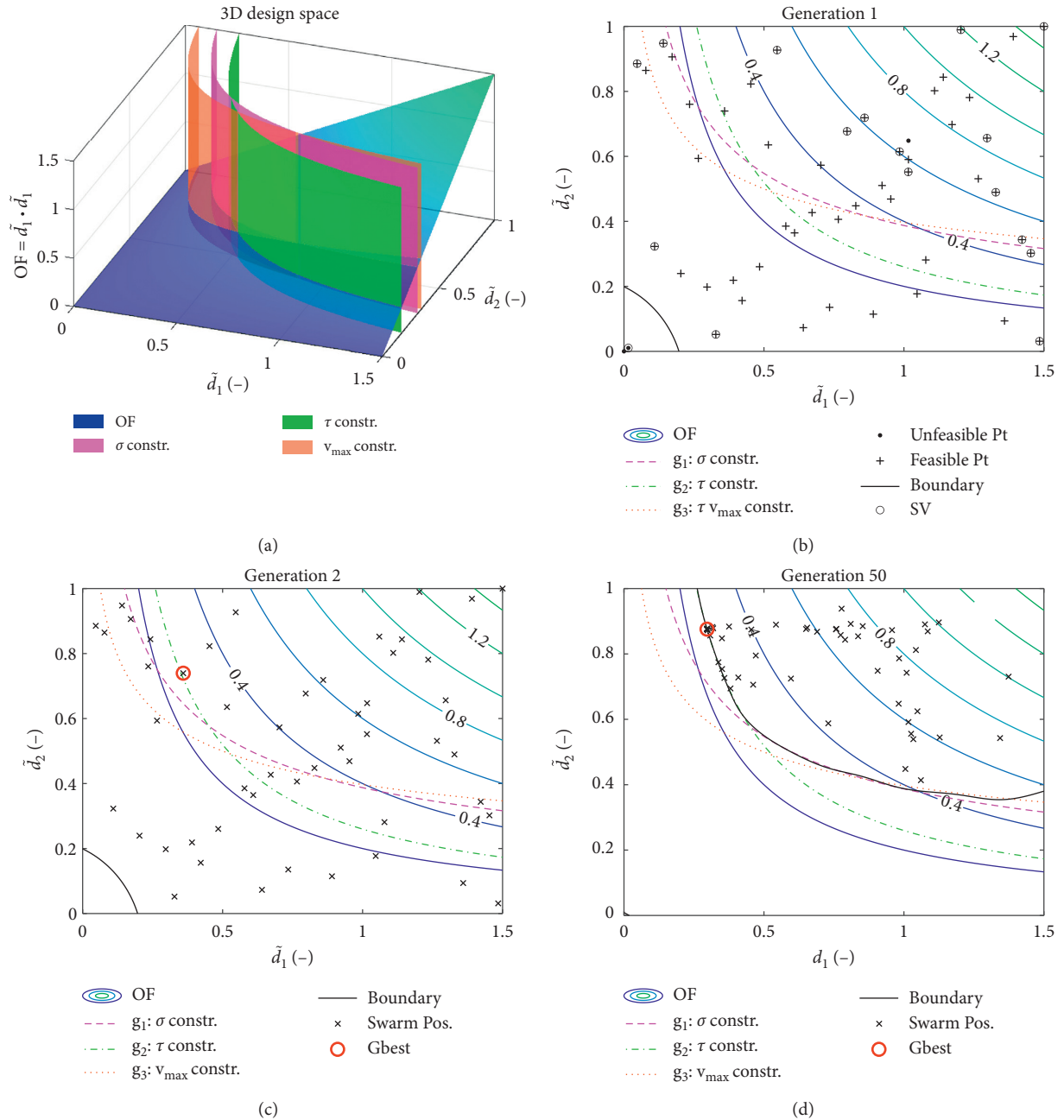
(a)
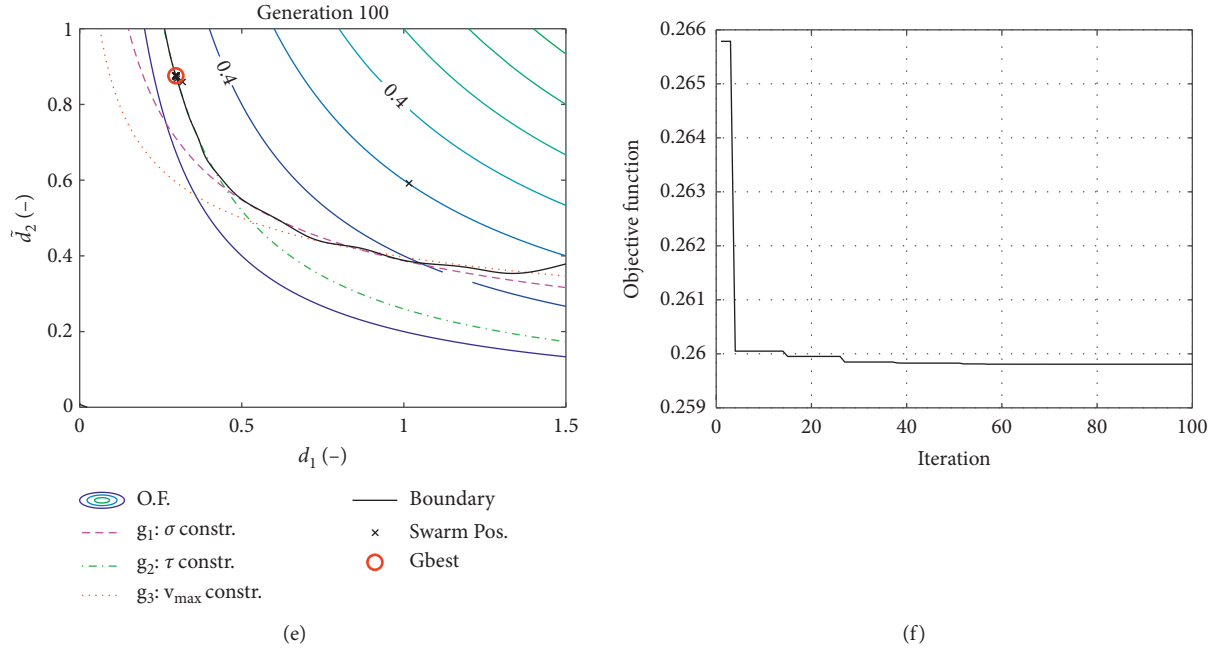
(b)

(c)

(d)

FIGURE 6: Continued.

(e)

(f)

FIGURE 6: Structural example 1: simply supported beam, case piecewise linear decreasing relax function. (a) 3D graph of simply supported beam problem design space. (b) Generation 1. (c) Generation 2. (d) Generation 50. (e) Generation 100. (f) Objective function history.

TABLE 4: Structural Example 1: results from PSO-SVM piecewise linear decreasing relax constraints, PSO-Penalty, and GA.

	PSO-SVM	PSO-Penalty	GA
μ_{OF}	0.2598076	0.2598076	0.2610035
σ_{OF}	2.04e-09	5.61e-17	2.58e-03
Best OF	0.2598076	0.259808	0.2598076
Worst OF	0.2598076	0.259808	0.2664507

upper bound $\tilde{d}_{1,\text{opt,UB}}$, it is necessary to calculate the intersection between the two aforementioned constraints:

$$\left.\begin{array}{l} \dfrac{3}{4}\psi_\sigma \dfrac{1}{\tilde{d}_1\,\tilde{d}_2^2} = 1, \\[3mm] \dfrac{3\sqrt{3}}{4}\psi_\sigma \dfrac{1}{\tilde{d}_1\,\tilde{d}_2} = 1, \end{array}\right\} \longrightarrow \sqrt{\dfrac{3}{4}\psi_\sigma \dfrac{1}{\tilde{d}_1}} = \dfrac{3\sqrt{3}}{4}\psi_\sigma \dfrac{1}{\tilde{d}_1} \longrightarrow \tilde{d}_{1,\text{opt,UB}} = \dfrac{9}{4}\psi_\sigma = 0.45.$$

(15)

As before, to calculate the optimal lower bound $\tilde{d}_{1,\text{opt,LB}}$, it is necessary to calculate the intersection between the equality τ constraint and the horizontal line $\tilde{d}_2 = 1$, obtaining

$$\left.\begin{array}{l} \dfrac{3\sqrt{3}}{4}\psi_\sigma \dfrac{1}{\tilde{d}_1\,\tilde{d}_2} = 1, \\[3mm] \tilde{d}_2 = 1, \end{array}\right\} \longrightarrow \tilde{d}_{1,\text{opt,LB}} = \dfrac{3\sqrt{3}}{4}\psi_\sigma = 0.2598.$$

(16)

Finally, considering (15) and (16), it is possible to obtain all the optimal pairs $(\tilde{d}_{1,\text{opt}}, \tilde{d}_{2,\text{opt}})$ using the following equation:

$$\tilde{d}_{2,\text{opt}} = \dfrac{0.2598}{\tilde{d}_{1,\text{opt}}}, \quad \text{with } 0.2598 \le \tilde{d}_{1,\text{opt}} \le 0.45.$$

(17)

Since the algorithm works with dimensionless parameters, in order to find the physical dimensions of the optimized cross-section, it is sufficient to multiply the obtained values $(\tilde{d}_{1,\text{opt}}, \tilde{d}_{2,\text{opt}})$ by L.

To show a technical possible application coming from this simple example, only a concrete beam with span length $L = 3$ m is now considered. Disregarding for the moment the self-weight load, the q load set to 15 kN/m represents only a live load. The concrete modulus is set to $E = 25$ GPa and the Von Mises ideal stress is related to the tensile stress of concrete set to $\sigma_{id} = 3$ MPa. Considering the box search space as $0 \le b \le 40$ cm and $0 \le h \le 45$ cm, the algorithm found the minimum weight respecting the constraints with $b = 16.67$ cm and $h = 42$ cm. Rounding-off these values, the self-weight associated to a concrete beam with $b = 18$ cm and and $h = 45$ cm is equal to

$$G = \gamma_{\text{concrete}} \cdot b \cdot h = 24 \cdot 0.18 \cdot 0.45 = 1.944 \text{ kN/m}.$$

(18)

For sake of simplicity, adding G to q a new load equal to 16.944 kN/m which takes into account also the self-weight is defined. Launching again the algorithm, new optimal exact dimensions are now obtained: $b = 18.83$ cm and $h = 45$ cm. Rounding-up the exact solution, a new self-weight equal to $G = 2.16$ kN/m is coming from a section with $b = 20$ cm and $h = 45$ cm. Now, the convergence is reached because the new optimal exact solution is $b = 19.07$ cm and $h = 45$ cm. Finally, the optimal cross-section for this concrete beam which minimizes the self-weight is given by $b = 20$ cm and $h = 45$ cm.

6. Structural Example 2: Optimization of a Warren Truss Beam

The second structural example comes from [49]. In that work, the weight optimization of an in-plane Warren truss simply supported beam, depicted in Figure 7, is performed with Differential Evolutionary Algorithm (DEA). The steel profile used for truss members is a square hollow core section, as shown in Figure 7. This kind of profile ensures good stability against buckling, and it represents a good solution for this type of structure because of its high strength-to-weight ratio [49]. On the contrary, joint connections are usually welded, and so, in order to reduce the total cost, it is important to reduce the size of sections to be welded. Regarding to the size optimization problem, as shown in Figure 7, this kind of sections are completely described by only two independent design variables: the outer dimension of the cross-section B and the thickness of the webs s. In this problem, the truss has m members belonging to four different types of cross-sections, as reported in Figure 7: lower chord (B_1, s_1), upper chord (B_2, s_2), internal webs (B_3, s_3), and external webs (B_4, s_4). To perform the shape optimization further, two design variables are considered: vertical height of the external webs H_{min} and the maximum height H_{max}. The design vector is therefore defined as

$$\mathbf{x} = (B_1, B_2, B_3, B_4, s_1, s_2, s_3, s_4, H_{min}, H_{max}), \quad (19)$$

and the box search space domain Ω is defined by $60 \leq B_i \leq 360$ mm, $4 \leq s_i \leq 30$ mm, and $50 \leq H \leq L$ mm, where $i = 1, 2, 3,$ and 4 and L is the total span length. Considering the maximum value of thickness $s_{max} = 30$mm, the minimum value of dimension B cannot be assumed less than $2s_{max}$ due to geometric limits. The objective function is represented by the total weight of the structure [49]:

$$W(\mathbf{x}) = \sum_{i=1}^{m} \rho_i l_i A_i, \quad (20)$$

where $\rho_i = 7.85$ t/m^3 is the steel density supposed equal for all members, l_i is the length, and A_i is the cross-section of the ith member. The structural steel used in this example is a S275, and the modulus of elasticity of the steel is 210 GPa. Regarding topology optimization, in [49], for fixed length $L = 20$m, the optimal number of bays which divides the lower chord is 20. The external load is as a uniformly distributed load $q = 100$kN/m applied on the lower chord acting as point loads in the nodes of the truss. The constraints are represented by the strength verifications about tensile stress (without any holes), compression stress, and buckling instability according to Eurocode 3 (EN 1993-1 2005 and EN 1993-2 2006). Despite the Eurocode $\gamma_{M0} = 1$ and $\gamma_{M1} = 1.1$ for bridges being recommended, to be more safe, the partial safety factors are set both equal to $\gamma_{M_0} = \gamma_{M_1} = 1.1$. Another constraint to satisfy is the maximum deflection which is usually set to $u_{lim} = L/500$ for bridges like that. For the current case, model uncertainty was not considered; however, the eventual adoption of specific model uncertainty coefficients (e.g., Castaldo et al., [50]) can be easily included in resistance or deformability checks. In order to make a comparison with the results of [49], for academic reasons, there is no distinction of the load combination for the strength verifications and for the deformability checks. The verification equations do not need deeper examination because this is beyond the scope of the present document. Therefore, the optimization problem statement is the following [49]: *Find* $\mathbf{x} \in \Omega$ *such that*

$$\min \ f(x) = W(x),$$

$$\text{s.t.} \quad \begin{aligned} \frac{N_{Ed}}{N_{t,Rd}} &\leq 1, \\[2mm] \frac{N_{Ed}}{N_{c,Rd}} &\leq 1, \\[2mm] \frac{N_{Ed}}{N_{b,Rd}} &\leq 1, \\[2mm] u_{max} &\leq u_{lim}. \end{aligned} \quad (21)$$

In order to solve (21), the FEM structural analysis was performed in the Matlab® CALFEM and PSO-SVM was adopted for the optimization process. In the PSO-SVM, a population size of 100 individuals is set with $k_{max} = 200$ iterations and a constant relax function with user parameter $\lambda = 3$ is applied to standard deviation of unfeasible points. It is performed 50 time runs, and the best-obtained solutions are collected in Figure 8. As one can see in Figure 8, due to the complexity of the problem, sometimes the algorithm does not reach the optimum and stack in a local minimum. As shown in the graph, the optimum solution is around 3.1 t; then, it is possible to cut the graph considering only the 21 runs over the total 50 which are characterized by a best OF solution lower than 3.1 t (dashed line). In this way, the possible outliers are excluded and now it is possible to perform the postprocessing searching for the real best solution.

Considering the abovementioned solutions, the obtained results shown in Table 5 have a quite large standard deviation in terms of design variables but very low in terms of objective function. This means that it is possible to find many combinations of design variables which are giving always the almost same objective function. The best design value of the 50 runs, also reported in Table 5, is taken into consideration. It is possible to compare this latter objective function (3.074t) with the optimal exact solution given by the original DEA code output [49]. The DEA optimal solution was characterized by the weight of 2.95t, so this is in the right order of magnitude. The comparison of the design variables is based only on general observation in accordance with the literature, as affirmed in [49]. In fact, for instance, it is expected that, mainly due to instability problems, upper chord and external diagonals would be bigger than the lower chord and internal diagonals. Finally, the best exact solution may be trivially rounded-up to get an industrialized more realistic design. The new design variables are reported in 5,
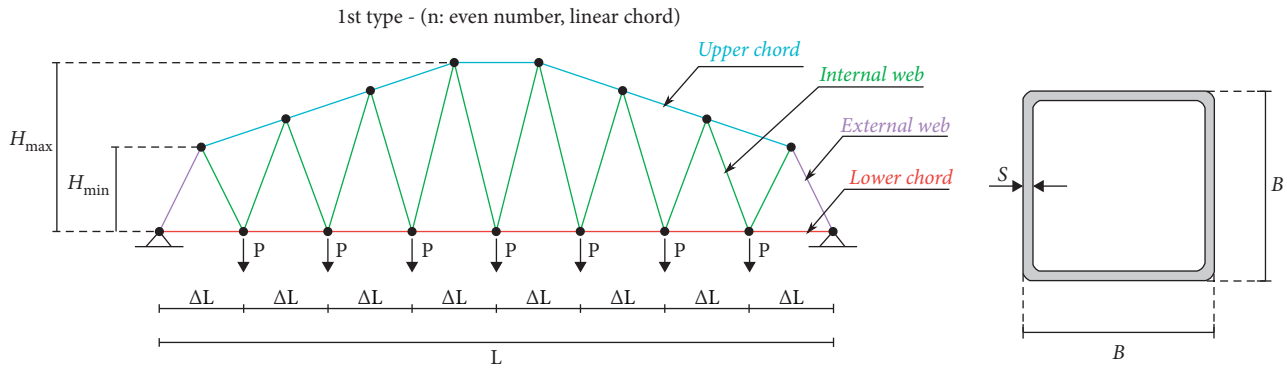
1st type - (n: even number, linear chord)

FIGURE 7: On the left, problem formulation of simply supported truss Warren beam. On the right, square hollow core tubular section with the indication of the design variables.
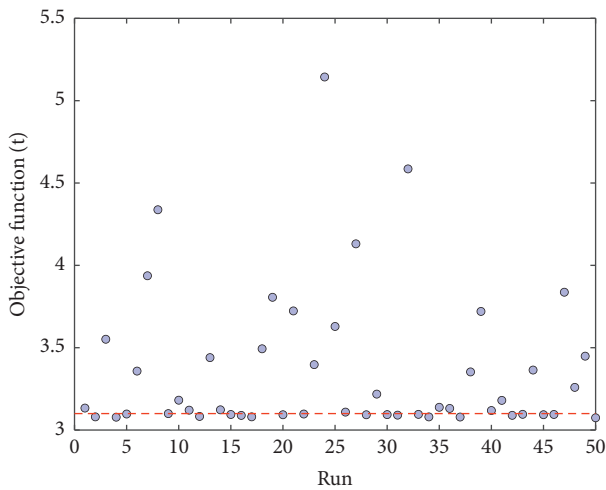

FIGURE 8: Structural example 2: Warren truss. Results from 50 times run PSO-SVM.

and as one can see, this solution is more conservative, and it leads to an increased total weight (3.189t).

If one wants to find a more accurate industrial solution, it is necessary to perform a more accurate analysis of the obtained results. Since the topology optimization was already taken into account in [49], one has to remember that, in the design variables, the algorithm is performing the size optimization and the shape optimization. This latter is regarding to the definition of the H_{min} and H_{max} values. Considering only the abovementioned 21 solutions and the standard deviations of H_{min} and H_{max}, it is possible to assume that the rounded-up values of the best exact solution can represent a good result for shape optimal parameters values: $H_{min} \approx 410mm$ and $H_{max} \approx 4145mm$.

Once the shape optimization is solved, regarding the size optimization, firstly the best exact solution as the optimal one, one has to remember that B_i and s_i were chosen as design parameters because of their independence, but in the optimization process, they are connected. In fact, in both objective function evaluation and constraints evaluation, these two parameters are combined into the resisting cross-section value. It is possible to obtain almost the same value of cross-section with different combinations of the design

parameters. In particular, one can refer to the optimal exact solution in terms of resisting cross-sections which represent the best solution in terms of both strength verification and minimization of the weight. As one can check, for the best exact solution, the section class of all members is 1, but other optimal solutions within the 21 considered are characterized by class 4 profile. In this case, to get the strength verification satisfied, it is necessary to refer to the resisting effective area. Usually in the design, when it is possible, it is preferred to avoid class 4 profiles, and the best condition is to find an optimal solution with class 1 profile. Therefore, the best industrial solution which respects all the constraint is given by all the pairs (B_i, s_i) with $i = 1, 2, 3$, and 4 which gives class 1 profiles and the minimum value of area greater or equal to the effective areas requested by the best solution in Table 6. This procedure allows us to find the best solution which respects the strength verification only. For the instability verification, it is necessary to take into account also the second moment of inertia which conditions Euler's critical load N_{cr} and consequently the dimensionless slenderness $\bar{\lambda}$ which influences the reduction factor χ. Fixing the thickness s_i to discrete values (rounded with 1 mm of precision), starting from the best-found solution and making an iterative discrete research to find B_i (rounded with 5 mm of precision) respectful of our abovementioned design rule, the best optimal industrial solution is found and reported in the last column of Table 5. As one can check, the minimum cross area of internal webs could be assured by the pair $(B_3, s_3) = (105; 5)mm$, but, due to instability problems, it is necessary to take into account the inertia and choose a profile that ensures both strength and instability verifications. The best industrial structure is verified by all strength and instability constraints. Making a comparison between the two last columns of Table 5, the trivial rounded-up solution represents a good optimal solution in terms of objective function. In fact, minimizing the weight is important, but the total cost is also affected by other aspects, e.g., welding and detailing and labour cost. So, the solution obtained by trivial rounded-up the exact one; it can be considered as an acceptable optimal result. In Figure 9(a), the undeformed and the deformed shape are depicted. It is possible to appreciate that the node 7 and, due to symmetry, the node 15 are the nodes that undergo the most deflection

TABLE 5: Structural example 2: mean values μ and standard deviations σ of best results from 21 solution over 50 runs of PSO-SVM with OF less than 3.1 t. Last three columns: best exact solution, trivial rounded-up solution, and refined industrial solution.

(mm)	μ exact sol.	σ exact sol.	Best sol.	Trivial industrial sol.	Best industrial sol.
B_1	72.5	15.7	94.2	95	95
B_2	190.2	70.2	128.1	130	105
B_3	128.9	1.4	128.8	130	130
B_4	211.7	107.9	132.1	135	110
s_1	6.0	1.5	4	4	4
s_2	14.7	7.4	18.8	20	26
s_3	4.0	0.03	4	4	4
s_4	14.0	8.4	14.7	15	19
$H_{\min}$	399.4	21.5	410.5	410	410
$H_{\max}$	4064.7	113.5	4145.0	4145	4145
OF (t)	3.0898	0.0071	3.074	3.189	3.092

TABLE 6: Structural example 2: comparison between best exact solution and best industrial solution and relative requested and provided areas.

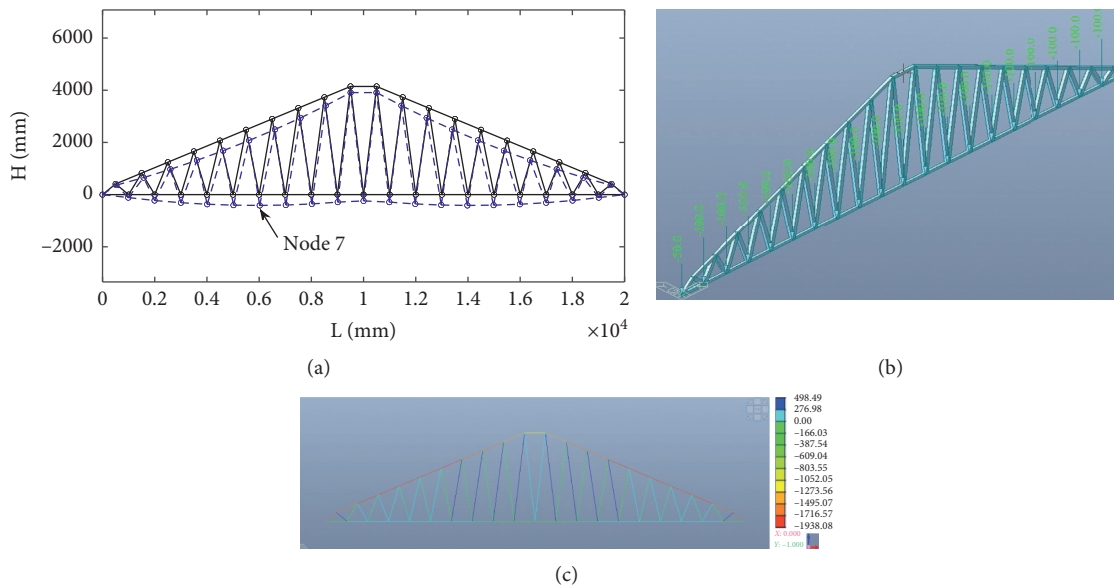| (mm, mm^2)	$(B; s)_{\text{exact}}$	Requested area	$(B; s)_{\text{industrial}}^{\text{best}}$	Provided area
Lower chord	(94.2; 4)	1442.46	(95; 4)	1456
Upper chord	(128.1; 18.8)	8206.18	(105; 26)	8216
Internal webs	(128.8; 4)	1996.81	(130; 4)	2016
External webs	(132.1; 14.7)	6907.21	(110; 19)	6916

FIGURE 9: Structural example 2. (a) Optimal Warren truss, solid Line: undeformed shape; dashed line: deformed shape. (b) View of the model of the Warren truss beam on Midas Gen®. (c) Planar view of the Warren truss beam on Midas Gen® with the axial force values.

$u_{\max} = 39.8$mm; however, it is respectfully of the service limit $L/500 = 40$ mm. In order to assess the validity of the results of the optimization process, the Warren truss beam is modelled with FEM professional software MIDAS Gen®. The assessment is made not only in terms of axial force for each member, but in particular, in terms of the performance ratio. This latter represents an efficiency percentage of the usage of the steel and is given by the strength ratio between the demand and the capacity. The simply supported Warren truss is modelled through truss elements in order to guarantee pure axial behaviour of each member. The section properties of the trivial best industrial results from Table 5

are assigned to each element. The adopted steel is always a S275 and the uniformly distributed load $q = 100$ kN/m acting on the lower chord is reconducted as a hanged load directly applied at lower chord nodes acting as concentrated forces, as depicted in Figure 9(b). As already remarked, in order to get results which are directly comparable with the Matlab code and with DEA code from [49] and for the sake of simplicity, no load combination is considered. The aim is to demonstrate that the proposed algorithm provides comparable results with DEA code which is used as a benchmark and not to make a perfect design completely respectful of the current codes. In order to take into account

this latter issue, it is sufficient to consider the correct the load combination at ULS for the strength verification and SLS for the maximum allowable displacement.

As reported in Figure 9(c), the results obtained from the Matlab code are equal to the FEM software. The overall behaviour shows that the upper chord is entirely compressed, whereas the lower chord is entirely tensed. The internal webs are alternatively compressed and tensed as usual for truss beam of this typology. Calculating the strength ratio, it is worth noting that a performance ratio between 75% and 98% is obtained for the members. These remarkable results show the importance of the optimization process during the design phase which can strongly support the decision process of the designer.

7. Conclusions

After an introductory part on PSO and the state of the art of constraint handling techniques, the paper presented a new valid alternative not-penalty method to solve constrained optimization problems. The main advantage with respect to the most-used nowadays penalty approach is represented by the generality of the machine learning SVM algorithm. Since it depends intrinsically on the inner product of the data, it is more adaptive even with discontinuous and nonlinear boundary of the feasible region in the design space. In order to improve the behavior of the proposed algorithm to deal with very sharp and narrow feasible regions, a relax constraint function is also implemented. From the computational point of view, the trade-off has been found adopting a population size not too large and using an incremental boundary update. It would be also possible sampling a huge initial random data and leaving the boundary fixed during the generations; however, this procedure does not lead to a good result in terms of objective function. Finally, the two numerical benchmark examples demonstrated the convergence of the new method in comparison with another penalty approach and with a GA. The last two examples highlighted the adaptability of this new method even into the structural optimization field. In particular, in the Warren truss beam problem, the optimization algorithm provided a numerical exact solution which can be easily industrialized by the designer with a trivial rounding-off without

jeopardizing the optimization process. Although the Warren truss beam example is performed under simplified assumption in order to make comparisons with DEA code from [49], from the technical point of view, the new optimization algorithm becomes a really useful and powerful support for the designer during the design and the decision process. It is important to stress that working with metaheuristic algorithms always involves the definition of the value of many arbitrary parameters. Starting from literature suggestions for these values, it is always strongly suggested to perform a fine-tuning of some of these parameters also with a trial-and-error approach for each specific problem in order to find the best optimal results in terms of objective function convergence, computational effort, and elaboration time.

Future applications of the proposed PSO-SVM framework in structural engineering are different. Among the different possibilities, an effective exploitation could address reliability-based calibration of partial safety factors or semiempirical coefficients included in design expressions (e.g., Mancini et al. [51]). Other applications could include seismic reliability or expected annual loss optimization for structural problems which could benefit (e.g., Di Trapani et al. [52] and Cavaleri et al. [53, 54]).

Appendix

Test Functions' Constrained Problems

The following mathematical problems were tested for the proposed PSO-SVM algorithm.

The following problem is taken by [45], and it is called sickle function:

$$
\begin{aligned}
\min \quad & f(\mathbf{x}) = (x_1 - 20)^3 + (x_2 - 10)^3 \\
\text{s.t.} \quad & g_1(\mathbf{x}) = (x_1 - 5)^2 + (x_2 - 5)^2 - 100 \geq 0 \\
& g_2(\mathbf{x}) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 82.81 \geq 0,
\end{aligned} \tag{A.1}
$$

where the search space is defined as $0 \leq x_1 \leq 10$ and $14 \leq x_2 \leq 15.5$. The global optimum is located at $\mathbf{x}^* = [14.095, 0.84296]$, where $f(\mathbf{x}) = -6961.8139$. The following problem is taken from [46], and it is a multivariable problem with five design variables and six constraints:

$$
\begin{aligned}
\min \quad & f(\mathbf{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141, \\
& g_1(\mathbf{x}) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \leq 0, \\
& g_2(\mathbf{x}) = -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \leq 0, \\
& g_3(\mathbf{x}) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \leq 0, \\
\text{s.t.} \quad & g_4(\mathbf{x}) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \leq 0, \\
& g_5(\mathbf{x}) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \leq 0, \\
& g_6(\mathbf{x}) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \leq 0,
\end{aligned} \tag{A.2}
$$

where the search space is defined as $78 \leq x_1 \leq 102$, $33 \leq x_2 \leq 45$, and $27 \leq x_3, x_4, x_5 \leq 45$. The optimum is located at $\mathbf{x}^* =$ [78, 33, 29.995256025682, 45, 36.775812905788], where $f(\mathbf{x}) = -30,665.539$.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Piscataway, NJ, USA, Jully 1995.

[2] R. E. Perez and K. Behdinan, "Particle swarm approach for structural design optimization," *Computers and Structures*, vol. 85, no. 19-20, pp. 1579–1588, 2007.

[3] L. J. Li, Z. B. Huang, and F. Liu, "A heuristic particle swarm optimization method for truss structures with discrete variables," *Computers and Structures*, vol. 87, no. 7-8, pp. 435–443, 2009.

[4] O. Hasançebi, S. Çarbaş, E. Doğan, F. Erdal, and M. P. Saka, "Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures," *Computers and Structures*, vol. 87, no. 5-6, pp. 284–302, 2009.

[5] A. Kaveh and S. Talatahari, "Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures," *Computers and Structures*, vol. 87, no. 5-6, pp. 267–283, 2009.

[6] V. Plevris and M. Papadrakakis, "A hybrid particle swarm-gradient algorithm for global structural optimization," *Computer-Aided Civil and Infrastructure Engineering*, 2010.

[7] S. N. Omkar, D. Mudigere, G. N. Naik, and S. Gopalakrishnan, "Vector evaluated particle swarm optimization (VEPSO) for multi-objective design optimization of composite structures," *Computers and Structures*, vol. 86, no. 1-2, pp. 1–14, 2008.

[8] M. W. Bloomfield, J. E. Herencia, and P. M. Weaver, "Analysis and benchmarking of meta-heuristic techniques for lay-up optimization," *Computers & Structures*, vol. 88, no. 5-6, pp. 272–282, 2010.

[9] S. M. Seyedpoor, J. Salajegheh, E. Salajegheh, and S. Gholizadeh, "Optimum shape design of arch dams for earthquake loading using a fuzzy inference system and wavelet neural networks," *Engineering Optimization*, vol. 41, no. 5, pp. 473–493, 2009.

[10] S. Gholizadeh and E. Salajegheh, "Optimal design of structures subjected to time history loading by swarm intelligence and an advanced metamodel," *Computer Methods in Applied Mechanics and Engineering*, vol. 198, no. 37-40, pp. 2936–2949, 2009.

[11] G. Monti, G. Quaranta, and G. C. Marano, "Genetic-algorithm-based strategies for dynamic identification of nonlinear systems with noise-corrupted response," *Journal of Computing in Civil Engineering*, vol. 24, no. 2, pp. 173–187, 2010.

[12] V. Plevris, *Innovative Computational Techniques for the Optimum Structural Design Considering Uncertainties*, National Technical University of Athens, Athens, Europe, 2009.

[13] G. Quaranta, W. Lacarbonara, and S. F. Masri, "A review on computational intelligence for identification of nonlinear dynamical systems," *Nonlinear Dynamics*, vol. 99, no. 2, pp. 1709–1761, 2020.

[14] S. Sengupta, S. Basak, and R. A. Peters, "II. "*Particle swarm optimization: a Survey of Historical and recent Developments with hybridization perspectives*," *Machine Learning and Knowledge Extraction*, vol. 1, pp. 157–191, 2019.

[15] T.-Y. Chen and T.-M. Chi, "On the improvements of the particle swarm optimization algorithm," *Advances in Engineering Software*, vol. 41, no. 2, pp. 229–239, 2010.

[16] B. Li and R. Y. Xiao, "The particle swarm optimization algorithm: how to select the number of iteration," in *Proceedings of the Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing IIHMSP 2007*, Kaohsiung, Taiwan, November 2007.

[17] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE World Congress on Computational Intelligence*, pp. 69–73, Anchorage, AK, USA, May 1998.

[18] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the Congress on evolutionary computation*, Washington, DC, USA, July 1999.

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

[20] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2-4, pp. 311–338, 2000.

[21] C. A. C. Coello, "Theoretical and numerical constraint handling techniques used with evolutionary algorithms: a survey of the state of the art," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 11-12, pp. 1245–1287, 2002.

[22] Y. Wang, Z. Cai, Y. Zhou, and Z. Fan, "Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique," *Structural and Multidisciplinary Optimization*, vol. 37, no. 4, pp. 395–413, 2009.

[23] E. Mezura-Montes, *Constraint-Handling In Evolutionary Optimization*, Springer, Berlin, Germany, 2009.

[24] S. Koziel and Z. Michalewicz, "Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization," *Evolutionary Computation*, vol. 7, no. 1, pp. 19–44, 1999.

[25] Z. Michalewicz and D. B. Fogel, *How to Solve it: Modern Heuristics*, Springer, Berlin, Germany, 2000.

[26] A. R. Jordehi, "A review on constraint handling strategies in particle swarm optimisation," *Neural Computing and Applications*, vol. 26, no. 6, pp. 1265–1275, 2015.

[27] M. Kohler, L. Forero, M. Vellasco, R. Tanscheit, and M. A. Pacheco, "PSO+: A nonlinear constraints-handling particle swarm optimization," in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 2518–2523, Vancouver, BC, USA, December 2016.

[28] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.

[29] E. Mezura-Montes and C. A. C. Coello, "A simple multi-membered evolution strategy to solve constrained

optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 1, pp. 1–17, 2005.

[30] T. P. Runanarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284–294, 2000.

[31] F. Mota, V. Almeida, E. F. Wanner, and G. Moreira, "Hybrid PSO algorithm with iterated local search operator for equality constraints problems," in *Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–6, Rio de Janeiro, Brazil, July 2018.

[32] O. Begambre and J. E. Laier, "A hybrid Particle Swarm Optimization - simplex algorithm (PSOS) for structural damage identification," *Advances in Engineering Software*, vol. 40, no. 9, pp. 883–891, 2009.

[33] C. Praveen and R. Duvigneau, "Low cost PSO using meta-models and inexact pre-evaluation: application to aerodynamic shape design," *Computer Methods in Applied Mechanics and Engineering*, vol. 198, no. 9-12, pp. 1087–1096, 2009.

[34] A. R. M. Rao and K. Sivasubramanian, "Multi-objective optimal design of fuzzy logic controller using a self configurable swarm intelligence algorithm," *Computers and Structures*, vol. 86, no. 23-24, pp. 2141–2154, 2008.

[35] H.-W. Ge, Y.-C. Liang, and M. Marchese, "A modified particle swarm optimization-based dynamic recurrent neural network for identifying and controlling nonlinear systems," *Computers and Structures*, vol. 85, no. 21-22, pp. 1611–1622, 2007.

[36] V. Kalivarapu, J.-L. Foo, and E. Winer, "Synchronous parallelization of particle swarm optimization with digital pheromones," *Advances in Engineering Software*, vol. 40, no. 10, pp. 975–985, 2009.

[37] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[38] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, Inc, New York, NY, USA, 1998.

[39] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer Science, Berlin, Germany, 2006.

[40] N. Cristianini and B. Schölkopf, "Support vector machines and kernel methods," *AI Magazine*, vol. 23, no. 3, 2002.

[41] C. Cortes and V. Vapnik, *Support-Vector Networks*, Kluwer Academic Publishers, Boston, MA, USA, 1995.

[42] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, Massachusetts Institute of Technology Press, Cambridge, MA, USA, 2012.

[43] LI. Hong-shuang, LÜ Zhen-zhou, and Y. Zhu-Feng, *Support Vector Machine for Structural Reliability Analysis*, Applied Mathematics and Mechanics English Ed, Berlin, Germany, 2006.

[44] M. N. Alam, *Codes in MATLAB for Particle Swarm Optimization*, https://doi.org/10.13140/RG.2.1.1078.7608, March 2016.

[45] P. A. Simionescu, D. G. Beale, and G. V. Dozier, "Constrained optimization problem solving using estimation of distribution algorithms," in *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, vol. 1, pp. 296–302, Portland, OR, USA, June 2004.

[46] T. P. Runarsson and X. Yao, "Search biases in constrained evolutionary optimization," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 35, no. 2, pp. 233–243, May 2005.

[47] W. R. Spillers, *Structural Optimization*, pp. 1–5, Springer-Verlag US, Berlin, Germany, 2009.

[48] P. W. Christensen et al., *An Introduction to Structural Optimization*, p. 44, Springer ScienceBusiness Media B.V., Berlin, Germany, 2009.

[49] A. Fiore, G. C. Marano, R. Greco, and E. Mastromarino, "Structural optimization of hollow-section steel trusses by differential evolution algorithm," *International Journal of Steel Structures*, vol. 16, no. 2, pp. 411–423, 2016.

[50] P. Castaldo, D. Gino, G. Bertagnoli, and G. Mancini, "Resistance model uncertainty in non-linear finite element analyses of cyclically loaded reinforced concrete systems," *Engineering Structures*, vol. 211, p. 110496, 2020.

[51] G. Mancini, V. I. Carbone, G. Bertagnoli, and D. Gino, "Reliability-based evaluation of bond strength for tensed lapped joints and anchorages in new and existing reinforced concrete structures," *Structural Concrete*, pp. 1–14, 2017.

[52] F. Di Trapani, V. Bolis, F. Basone, and M. Preti, "Seismic reliability and loss assessment of RC frame structures with traditional and innovative masonry infills," *Engineering Structures*, vol. 208, p. 110306, 2020.

[53] L. Cavaleri, F. Di Trapani, and M. F. Ferrotto, "A new hybrid procedure for the definition of seismic vulnerability in Mediterranean cross-border urban areas," *Natural Hazards*, vol. 86, no. S2, pp. 517–541, 2017.

[54] L. Cavaleri, M. Fossetti, and M. Papia, "Modeling of out-of-plane behavior of masonry walls," *Journal of Structural Engineering*, vol. 135, no. 12, pp. 1522–1532, 2009.