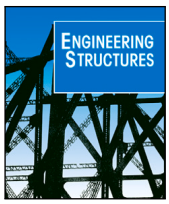
Engineering Structures 266 (2022) 114607

Contents lists available at ScienceDirect

Engineering Structures

journal homepage: www.elsevier.com/locate/engstruct



Discrete swarm intelligence optimization algorithms applied to steel–concrete composite bridges

D. Martínez-Muñoz [a,*], J. García [b,*], J.V. Martí [a], V. Yepes [a]

[a] Institute of Concrete Science and Technology (ICITECH), Universitat Politècnica de València, València, 46022, Spain
[b] Escuela de Ingeniería de Construcción y Transporte, Pontificia Universidad Católica de Valparaíso, Valparaíso, 2362807, Chile

ARTICLE INFO

Keywords:
Combinatorial optimization
Bridge
Metaheuristics
Composite structures
K-means

ABSTRACT

Composite bridge optimization might be challenging because of the significant number of variables involved in the problem. The optimization of a box-girder steel–concrete composite bridge was done in this study with cost and CO_2 emissions as objective functions. Given this challenge, this study proposes a hybrid algorithm that integrates the unsupervised learning technique of k-means with continuous swarm intelligence metaheuristics to strengthen the latter's performance. In particular, the metaheuristics sine-cosine and cuckoo search are discretized. The contribution of the k-means operator regarding the quality of the solutions obtained is studied. First, random operators are designed to use transfer functions later to evaluate and compare the performances. Additionally, to have another point of comparison, a version of simulated annealing was adapted, which has solved related optimization problems efficiently. The results show that our hybrid proposal outperforms the different algorithms designed.

1. Introduction

Bridge optimization is an interesting problem to address both because of the technical challenges that the problem presents and because of the potential applications in reducing costs, CO_2 emissions, and energy consumption. The technical difficulties are related to the large number of discrete variables required in its design and the complex objective functions and restrictions that these must satisfy [1]. Due to the more significant number of variables necessary for their design, Steel–Concrete Composite Bridges (SCCB) present a considerable challenge. According to the SCCB literature, they can be classified into three groups according to their cross-section: Plate-beam, Twin-Girders, and Box-Girder [2], and their behavior are different between these types.

Due to the type of objective function and the constraints used in structural design problems, metaheuristics (MH) have had good results in optimizing structures. These techniques have been applied in steel structures [3], arch bridges [4,5] or reinforced concrete columns [6] among others. In some studies, metaheuristic has been applied as first optimization step for topological optimization [7] due to the computational cost of this last method [8]. In particular, metaheuristic techniques have performed well in addressing complex SCCBs optimization [9]. For example, in [10] a discrete harmony search algorithm was proposed and applied to the design of multi-span composite box girder bridges. In the article, the authors obtained a 15% reduction of materials when compared to a traditional design. In [11], a two-stage based optimization methodology is developed for the design of simply supported steel–concrete composite I-girder bridges. In the first stage, a simplified structural model based on expert criteria is developed and used with the aim of providing a starting point for the local search. In the second stage, a search group algorithm is chosen based on statistical analysis. The proposed method was shown to reduce the structural cost by up to 9.17%. Three metaheuristic algorithms were studied in [12] and used for reaching the optimal design of steel–concrete composite I-girder bridges. The algorithms used were Collision Bodies (CBO), Collision Body Enhanced Optimization (ECBO), and Vibration Particle System (VPS). Among the results, it was obtained that the final optimized design does not need longitudinal stiffeners.

Despite the excellent performance of metaheuristics and the large size of many combinatorial problems, the strengthening of metaheuristics is also necessary. Among the different strategies to strengthen the metaheuristics, the hybrid methods have stood out. Several of the most frequently utilized hybridization techniques, include *hybrid heuristic*, [13], where different metaheuristic algorithms are combined to enhance their capabilities. *Mateheuristics*, [14], where mathematical programming methods are integrated with metaheuristics techniques. *Simheuristics*, [15], which encompass a combination of simulations with metaheuristic modeling.

* Corresponding authors.
E-mail addresses: damarmu1@cam.upv.es (D. Martínez-Muñoz), jose.garcia@pucv.cl (J. García), jvmartia@cst.upv.es (J.V. Martí), vyepesp@cst.upv.es (V. Yepes).

https://doi.org/10.1016/j.engstruct.2022.114607
Received 24 January 2022; Received in revised form 24 May 2022; Accepted 29 June 2022
Available online 7 July 2022

Metaheuristics generate important ancillary data in the solution search process, which can be exploited by machine learning methods. This opens a line of research in which machine learning techniques can be integrated into metaheuristic algorithms to strengthen the performance of the latter.

When searching for integration types in which machine learning techniques improve the performance of metaheuristic algorithms, three main categories are found. Low-level integrations, high-level integrations, and optimization problems [16,17]. In the case of the hybrid algorithm proposed in this article, the type of integration designed is a low-level integration. The low-level integrations involve local search operators, population initiation, binarization, parameter control, in which ML approaches improve particular operators of the MH algorithm. For example in parameter tuning, in [18], an iterated racing method was employed; a reset mechanism was combined with an elitist procedure (to assure the optimal configuration), and the use of a truncated sampling distribution to allow for automated parameter setting. Decision trees were utilized in [19] to modify particular parameters in the traveling salesman problem. Compared to fixed parameter values, the decision tree technique improved the quality of the solutions and the computing time. Another case of low-level integration is used to initialize solutions. For the shop scheduling problem [20], decision trees were utilized to produce early solutions for new instances, in conjunction with Opposition Based Learning (OBL), to begin complementary solutions. Reinforcement Learning (RL) and other approaches can also be utilized to create solutions. In these situations, the solution building may be thought of as a series of additions of decisions, for which an RL algorithm can be trained. For example, a Deep Q-network was constructed and utilized for the optimization of the Job-Shop Scheduling Problem [21]. Similarly, in [22], transfer learning approaches were utilized to generate initial solutions using three evolutionary multi-objective algorithms and applied to 12 benchmark functions. Finally, another successful application of low-level integration has been used to generate binary versions of algorithms that work on continuous spaces.

Another essential field of study is the creation of binary versions of algorithms that function in continuous areas naturally. There are some examples of ML and metaheuristics working together in this field. The K-means approach was utilized to build binary versions of the cuckoo search algorithm (CS) and used to the matrix covering problem in [23]. For the multidimensional knapsack problem, in [24] a hybrid algorithm using k-means as the binarization method and KNN as the local search operator is proposed. The hybridization between metaheuristic techniques with the aim of improving the convergence or quality of the solutions is another interesting line of low-level integration. In [25], the hybridization of hybrid metaheuristic algorithms was proposed with the aim of addressing the optimal dimensioning of steel beam structures with numerous discrete variables. The numerical results indicate that the hybrid algorithm of adaptive dimensional search and exponential big bang-big crunch is the most promising of the techniques investigated. In [26], it is integrated the convergence curve of each subsequent execution of the algorithm in relation to the information gained from prior executions. It is monitored at specified times during each subsequent execution, referred to as the solution monitoring period. The solution monitoring period is chosen in such a way that each run allows the algorithm to explore the search space in order to increase the quality of the solution, while also periodically forcing the algorithm to return to the most promising prior visit. If it is unable to improve the solution after a specified number of iterations, it will terminate. Numerical investigations with tough test examples containing up to 354 design variables reveal that, in general, the proposed approach improves the solution quality and the robustness or stability of the outcomes in metaheuristic structure optimization.

Following this last line of generating binary versions to efficiently solve binary optimization problems. In this article, the integration method has been adapted to address discrete problems. In particular, a discrete hybrid algorithm is proposed. This algorithm incorporates the k-means technique into the discretization solution phase of continuous swarm intelligence metaheuristics. The contribution of this work includes:

- In this study, a cost, and CO_2 emissions optimization of a 60-100-60 three-span single box-girder steel–concrete composite bridge has been performed.
- It should be noted that it considers 35 design variables on average 55 possible choices for each variable, which implies a computationally demanding structure.
- A discrete hybrid k-means swarm intelligence algorithm is proposed, and the contribution of the k-means technique to the robustness of the algorithm is studied. In particular, it should be noted that in previous works [24,27], k-means has been used to solve binary optimization problems, that is, whether the variable is present or not. In the optimization developed in this article, the technique was adapted to allow more than two states for each of the variables.
- The results of the hybrid algorithm are compared with discrete simulated annealing that has been efficient in solving related problems [28], in addition to considering the comparison with algorithms that perform discretization through transfer functions that are frequently used to binarize or discretize solutions [29].

A brief content structure of the following sections: In Section 2, the box-girder steel–concrete composite bridge problem is detailed. Later, in Section 3, the discrete k-means swarm intelligence algorithm is developed. Our numerical experiments and comparisons are detailed in Section 4. Finally, in Section 5, the conclusions and future lines of research are discussed.

2. The optimization problem and computational model

This section aims to define and detail the optimization problem. In the case of bridges, there are different objective functions to be minimized, among which there is a particular interest in the costs of the bridge and the CO_2 emissions released in the manufacture of its materials. In this work, two mono-objective functions are defined. The first shows the bridge's overall cost, which is formalized in Eq. (1) and is calculated by multiplying the unit cost of each material c_i, multiplied by the units used, m_i. In the case of CO_2 emissions, the calculation is similar to the previous one, with the difference that instead of considering cost, the emissions e_i considers cradle-to-gate analysis for each unit of material i multiplied by the amount of material i used. The emissions calculation is formalized in Eq. (2). For the cost function, the values from Table 1 are used, which were obtained from the Construction Technology Institute from Catalonia by the BEDEC database [30]. Finally, the bridge design process is subject to constraints imposed by expert recommendations and regulations related to the standard. Generically, these are shown in Eq. (3).

$$C(\vec{x}) = \sum_{i=1}^{n} c_i \cdot m_i(\vec{x}) \tag{1}$$

$$E(\vec{x}) = \sum_{i=1}^{n} e_i \cdot m_i(\vec{x}) \tag{2}$$

$$G(\vec{x}) \leq 0 \tag{3}$$

For the description of the problem, the variables, parameters, and constraints of the problem will be considered. In the case of variables, they correspond to the values modified in the optimization procedure to achieve the optimum. In the case of parameters, they are values that are considered fixed in the optimization, and that usually represent boundary conditions. Finally, the constraints are imposed by regulations [31–33] and recommendations of specialists [2,34].

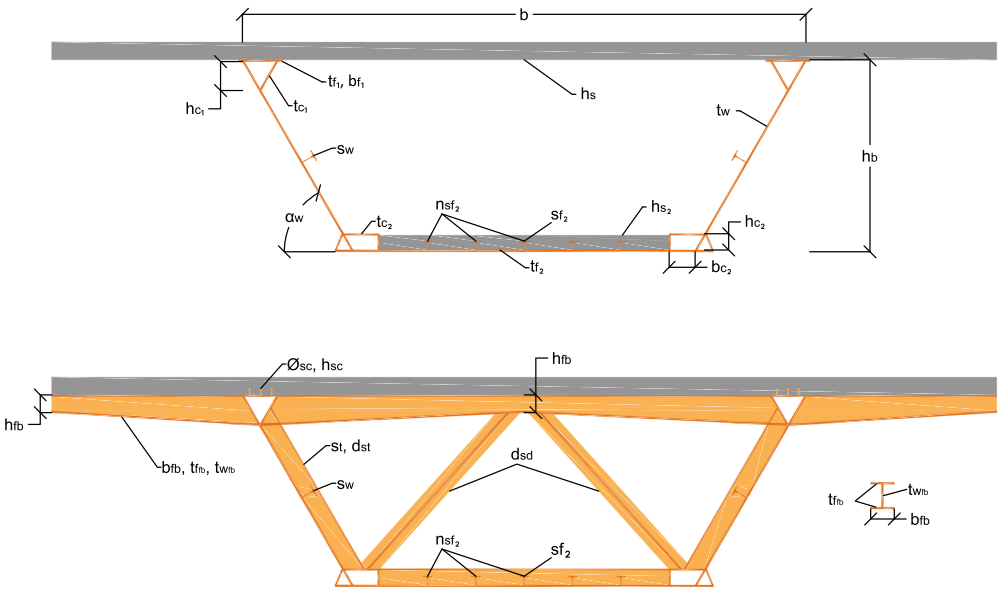


Fig. 1. Transverse section variables for SCC bridge.

Table 1
Cost and CO_2 emission values.

Unit	Cost (€)	Emissions (kg of CO_2)
m^3 of concrete C25/30	88.86	256.66
m^3 of concrete C30/37	97.80	277.72
m^3 of concrete C35/45	101.03	278.04
m^3 of concrete C40/50	104.08	278.04
m^2 of precast pre-slab	27.10	54.98
kg of steel B400S	1.40	0.70
kg of steel B500S	1.42	0.70
kg of rolled steel S275	1.72	4.33
kg of rolled steel S355	1.85	4.33
kg of rolled steel S460	2.01	4.33
kg of shear-connector steel	1.70	2.8

2.1. Variables of the problem

Our optimization problem is a Steel–Concrete Composite Bridge (SCCB) with a box-girder cross section divided in three spans of 60-100-60 m. The problem variables can be grouped into two groups. The first group correspond to the geometric variables of the bridge and the second group with grades of steel and concrete. In order to design bridges that are feasible to build, these variables cannot take any value, but only allowed values, with which our search space corresponds to a discrete space. Variables are shown in Table 2. These variables, depending on their characteristics, are grouped into five categories. The first category correspond to geometric variables of the transverse section. Upper distance between wings (b), wings and cells angle (α_w), top slab thickness (h_s), beam high (h_b), floor beam minimum high (h_{fb}), top flange thickness (t_{f_1}), top flange width (b_{f_1}), top cells high (h_{c_1}) and thickness (t_{c_1}), wing thickness (t_w), bottom cells high (h_{c_2}), thickness (t_{c_2}), and width (b_{c_2}) and bottom slab thickness (h_{s_2}). For clarity, these variables are outlined in Fig. 1.

The second category of variables corresponds to the diameters of the base reinforcement, first reinforcement and second reinforcement bar diameters (ϕ_{base}, ϕ_{r_1}, ϕ_{r_2}), and the corresponding bar number of the reinforcement areas (n_{r_1}, n_{r_2}). These variables are intended to improve the bridge transverse section. To optimize the top slab reinforcement, it has been divided into a base reinforcement that is the minimum required by regulations [31–33] and two more areas, in negative bending sections, where the reinforcement is increased. The lower slab and the lengths of the increasing area of reinforcement will be described in Section 2.2.

Table 2
Design variables and boundaries.

Variables	Unit	Lower bound	Increment	Upper bound	Values number
b	m	7	0.01	10	301
α_w	deg	45	1	90	46
h_s	mm	200	10	400	21
h_b	cm	250	1	400	151
h_{fb}	mm	400	100	700	31
t_{f_1}	mm	25	1	80	56
b_{f_1}	mm	300	10	1000	71
h_{c_1}	mm	0	1	1000	101
t_{c_1}	mm	16	1	25	10
t_w	mm	16	1	25	10
h_{c_2}	mm	0	10	1000	101
t_{c_2}	mm	16	1	25	10
b_{c_2}	mm	300	10	1000	71
t_{f_2}	mm	25	1	80	56
h_{s_2}	mm	150	10	400	26
$n_{s_{f_2}}$	u	0	1	10	11
d_{st}	m	1	0.1	5	41
d_{sd}	m	4	0.1	10	61
b_{fb}	mm	200	100	1000	9
$t_{f_{fb}}$	mm	25	1	35	11
$t_{w_{fb}}$	mm	25	1	35	11
n_{r_1}	u	200	1	500	301
n_{r_2}	u	200	1	500	301
ϕ_{base}	mm	6, 8, 10, 12, 16, 20, 25, 32			8
ϕ_{r_1}	mm	6, 8, 10, 12, 16, 20, 25, 32			8
ϕ_{r_2}	mm	6, 8, 10, 12, 16, 20, 25, 32			8
s_{f_2}	mm	From IPE 200 to IPE 600[a]			12
s_w	mm	From IPE 200 to IPE 600[a]			12
s_t	mm	From IPE 200 to IPE 600[a]			12
h_{sc}	mm	100, 150, 175, 200			4
ϕ_{sc}	mm	16, 19, 22			3
f_{ck}	MPa	25, 30, 35, 40			4
f_{yk}	MPa	275, 355, 460			3
f_{sk}	MPa	400, 500			2

[a]Following the standard series of IPE profiles.

The stiffeners correspond to the third category of variables. In this design, half IPE profiles for wings (s_w), bottom flange (s_{f_2}) and the transverse ones (s_t) are considered variable stiffeners. For bottom flange stiffeners, the number of stiffeners ($n_{s_{f_2}}$) has also been considered as a variable. As can be seen in Fig. 1, there are two more variables that define the distance between diaphragms (d_{sd}) and transverse stiffeners (d_{st}).

Table 3
Optimization problem parameters.

Geometrical parameters		
Bridge deck width (W)	16	m
Span number	3	
Central span length	100	m
External span length	60	m
Minimum web thickness ($t_{w_{min}}$)	15	mm
Minimum flange thickness ($t_{f_{2min}}$)	25	mm
Reinforcement cover	45	mm
Material parameters		
Maximum aggregate size	20	mm
Concrete longitudinal strain modulus (E_{cm})	$22 \cdot ((f_{ck}+8)/10)^3$	MPa
Concrete transverse strain modulus (G_{cm})	$E_{cm}/(2 \cdot (1+0.2))$	MPa
Steel longitudinal strain modulus (E_s)	210 000	MPa
Steel transverse strain modulus (G_s)	80 769	MPa
Regulation requirement parameters		
Regulations	Eurocodes [31–33,35], IAP-11 [36]	
Exposure environment	XD2	
Structural class	S5	
Service life	100	years
Loading parameters		
Reinforced concrete density	25	kN/m^3
Steel density	78.5	kN/m^3
Asphalt density	24	kN/m^3
Asphalt layer thickness	100	mm
Bridge traffic protections	5.6	kN/m
Traffic load	According to the codes	
Thermal load	According to the codes	
Wind load	According to the codes	

The last two categories correspond to the geometry of the shear connector's characteristics, and the floor beam variables. The floor beam variables are defined by the width of the floor beam (b_{fb}) and the thicknesses of the flanges ($t_{f_{fb}}$) and webs ($t_{w_{fb}}$). Shear connectors have been defined by their height (h_{sc}) and diameter (ϕ_{sc}). Finally, the elastic limit of rolled steel (f_{yk}), the strength of concrete (f_{ck}) and the elastic limit of reinforcing steel bars (f_{sk}) complete the definition of the variables.

2.2. Parameters of the problem

In each optimization problem, it is necessary to set some variables, typical of the conditions to which the structure will be subjected, defined by some regulation, environmental conditions, or some geometric definition that has no need or the possibility of changing. These fixed attributes are called parameters and remain unchanged throughout the optimization process. The parameters and values defined in the bridge design are summarized in Table 3.

Considering the Eurocode regulations [31,32], that our bridge is a 60-100-60 m three-span box-girder steel–concrete composite bridge with a deck width (B) of 16 m without height variation, for cells have been defined in the transverse section for improving the resistant behavior. These cells are shown in Fig. 1. Two cells are at the upper side of the wings, and the other two are at the bottom. The minimum height of these cells, (h_{c_1}, h_{c_2}), has been set to zero in order to identify if they contribute to the reduction of costs or emissions of CO_2 additionally for set the upper limit of these, the bridge design rules defined in [34] has been considered.

The base reinforcement for the upper and lower concrete slab is set according to the minimum need for reinforcement defined in Eurocode 2 [33]. The connection between the concrete slab and the steel beam is dimensioned to resist the shear lag stresses produced in the top flanges. For bending resistance the effective width given by Eurocode 4 [31] have been considered. Also, because the only width considered resistant is effective, the defined steel bar reinforcements are placed only in that width.

Finally, steel bar reinforcement increase and lower slab areas are defined. The lower slab is placed in negative bending sections to mobilize the composite dual-action. To define lengths where negative bending can be produced, we have considered the distance defined by Eurocode 4 [31] for shear lag stresses that correspond with one-third of the span length. As stated earlier, it is necessary to increase the upper slab reinforcement to resist the tension stresses produced. In this case study, we have considered two reinforcement areas. The first is placed in zones where the section can be subjected to negative bending, and base reinforcement cannot resist the stresses. The second is placed on top of supports, corresponding to one-third of the distance between the support and the point of change of sign of the bending of the theoretical law.

2.3. Constraints of the problem

In designing a structure, the constraints imposed by regulations and specific conditions to the structure, such as safety factors, must be considered. Mainly, in the optimization of this bridge, the necessary constraints to consider are defined in the regulations, [31–33] and additionally, recommendations defined in, [2,34] have been incorporated.

When analyzing the regulations, it is found that the constraints imposed make up two groups: The first one corresponds to the Ultimate Limit States (ULS) and the second group, to the Service Limit States (SLS). In the case of ULS, the restrictions are related to the structural resistance of the bridge elements subjected to the stresses caused by the acting loads. In the case of SLS, the restrictions are intended to ensure the serviceability of the structure during its useful life. All applied loads and their combination are defined in the [35] regulation. The Table 3 summarizes the load cases considered.

In order to verify the ULS restrictions in all the elements of the bridge, both the global and the local analyses have been considered. In the case of the global analysis, the checks consider shear, flexure, torsion, and flexure–shear interaction. To obtain stresses and deflections, a linear elastic analysis has been considered. When considering [31,32], these indicate that the resistance of the section must be included, in our design, the effective section has been selected and applying it both the shear lag reductions and the reduction of the section of steel plates classified as class 4. To achieve the above, a precision of 10^{-6} m has been considered for the iterative process. To obtain the value of the mechanical characteristics of the homogenized section, the relationship (n) between the longitudinal deformation modulus of concrete (E_{cm}) and steel (E_s) has been obtained according to Eq. (4). For the case of concrete creep and shrinkage, they were defined following the [31,33] standard. Likewise, a local model was developed to verify the beams, reinforcements, and diaphragms in the ULS floor, considering controls for shear, flexure, buckling, and minimum mechanical characteristics.

$$n = \frac{E_s}{E_{cm}} \tag{4}$$

The SLS considered for the analysis is the tension limit for materials, fatigue, and deflection. There is no apparent limit for deflection in Eurocodes, but the IAP-11 Spanish regulation [36] gives a maximum of $L/1000$ for the frequent combination of live loads deflection value, with L representing the span length. This has been considered as the maximum value of the deflection. In addition, we have considered geometrical and constructibility requirements.

Additionally, a numerical model has been implemented to obtain the stresses and carry out all the ULS, SLS, and geometric and constructibility checks. In the case of deflections and stresses, the model applies the perfect embedding forces method, taking the 34 bridge variables we selected as input data. The model divides each span of bridge into a defined number of bars. In this case, the total number of bars is 44, distributed in 12-20-12 corresponding to the three spans of the bridge; thus, discretize the bridge into bars of 5 m in length. Once

the stresses have been obtained, the program performs structural checks and returns the measurement results, cost, emissions, and verification coefficients. These verification coefficients correspond to the quotient between the design values of the effects of the actions (E_d) and their corresponding resistance value (R_d), as shown in Eq. (5). If these coefficient values are greater than or equal to one, then the Section complies with the imposed restriction.

$$\frac{R_d}{E_d} \geq 1 \tag{5}$$

2.4. Structure computational model description

The procedure used to obtain the deflections and stresses has been the perfect embedding forces method. This method consists in solving Eq. (6).

$$\mathbf{f} = \mathbf{K} \cdot \mathbf{d} + \mathbf{f_0} \tag{6}$$

In this equation, $\mathbf{f_0}$ correspond to the perfect embedding forces vector. These forces would be obtained if each of the system bars had all the degrees of freedom constrained. $\mathbf{K}$ is the stiffness matrix of the system, generated by assembling the stiffness matrices of all bar elements. To get the stiffness matrix of each element, the average between both frontal and dorsal nodes' mechanical properties has been calculated. The complete section without considering the shear lag and panel reduction has been considered to obtain these mechanical properties. Finally, $\mathbf{d}$ and $\mathbf{f}$ are the deflections and stress vectors, respectively.

This procedure is repeated with all load cases. The following load cases have been considered loading the entire bridge length as a single load case: Self Weight, Dead Loads, Thermal Heating, Thermal Cooling, and Wind. In order to consider the different positions of traffic loads, every 5-m bar has been loaded separately, considering two separated loading cases, the punctual load and the distributed. This gives, as a result, 88 load cases for traffic load and a total of 93 if all load cases are considered. The results obtained from loading each bar have been combined to consider all loading possibilities regarding traffic load. After this, the load case envelope has been calculated to consider each section's maximum and minimum results.

Regarding combinations and envelopes, the envelope of all persistent and transitory situations combinations have been obtained for ULS. These combinations have been considered dominant action all live loads in different combinations. The envelope of all characteristic combinations has been considered for SLS regarding stress limitation.

3. Optimization algorithms

The detail of the discretization algorithms will be explained in this Section. First, the metaheuristics used to perform the optimization will be detailed in Section 3.1. Then the proposed hybrid algorithm, Section 3.2, which uses k-means as the discretization method will be explained. Later in Section 3.3, the algorithm that uses a transfer function as a discretization method is detailed. The following reference is recommended for a greater depth of transfer functions and their applications in combinatorial optimization [29]. We must emphasize that the k-means discretization method takes all the solutions, groups them, and later assigns the probabilities. In the case of transfer functions, each probability is assigned individually, without looking at the other solutions.

Fig. 2 shows the flowchart used to perform the optimization using cuckoo search and the sine cosine algorithms (SCA). As a starting point, the set of solutions is initialized, this set corresponds to a valid set, that is, it complies with the constraints imposed by the problem. Once the solutions have been initialized, it is asked whether the stopping criteria of the algorithm are met. In this case, the stopping criterion of the algorithm corresponds to the maximum number of defined iterations. In the event that the maximum number of iterations has not been met, the hybrid algorithm is executed. As a first stage of the hybrid algorithm, the set of velocities for the different generated solutions is obtained. Subsequently to the set of solutions, the k-means clustering technique is applied in order to group the solutions and assign a transition probability to each group in the transition probabilities stage. The detail of these three stages will be explained in Section 3.2. Finally, a solution update criterion is established, in which it is evaluated if each of the variables or dimensions associated with a solution is updated. This is intended to balance the exploitation and exploitation of the search space. Solutions with good values of the fitness function will have few updates to be able to exploit the space. The detail of this update will also be made in Section 3.2

3.1. Swarm intelligence algorithms: SCA and CS

This section details the swarm intelligence algorithms used to address optimization. Specifically, the cuckoo search was chosen as it has successfully solved a large number of optimization problems, particularly in the area of civil engineering. Additionally, the parameterization of the original algorithm is quite simple. In the case of the sine cosine algorithm, the type of move it executes is based on the sin and cosine functions and is completely different from the move of cuckoo search. On the other hand, this last metaheuristic does not require proper parameter tuning.

3.1.1. Sine Cosine Algorithm (SCA)

Sine Cosine Algorithm (SCA) was proposed in [37] and corresponded to a swarm intelligence algorithm that considers the sine and cosine functions to carry out the process of exploring and exploiting the search space. To carry out the movement of the solutions, P_j^t is additionally used, which corresponds to the position of the destination solution for iteration t and dimension j, and typically uses the best solution obtained so far. In addition to P_j^t, the algorithm uses four random numbers r_1, r_2, r_3, r_4. As the algorithm starts to iterate, r_1 decreases. It starts at 2 and converges to 0 at the end of the optimization. On the other hand, r_2 takes values between 0, 2π. r_3 considers values between 0 and 2, and finally r_4 is used to select Eqs. (7) and (8) taking values between 0 and 1 and a threshold of 0.5.

The update method used is shown in Eqs. (7) and (8).

$$x_{i,j}^{t+1} = x_{i,j}^{t} + r_1 \times sin(r_2) \times \mid r_3 P_j^t - x_{i,j}^t \mid \tag{7}$$

$$x_{i,j}^{t+1} = x_{i,j}^{t} + r_1 \times cos(r_2) \times \mid r_3 P_j^t - x_{i,j}^t \mid \tag{8}$$

3.1.2. Cuckoo search algorithm (SC)

The reproductive strategy phenomena observed in cuckoo species, which lay their eggs in the nests of other bird species, has inspired the CS algorithm. Such is the level of sophistication of cuckoo birds that in some cases even the colors and patterns of the eggs of the chosen host species are mimicked. In the analogy, an egg corresponds to a solution. The concept behind the analogy is to use the best solutions (cuckoos) to replace those that do not perform well. The CS algorithm uses three basic rules:

1. Each cuckoo lays one egg at a time and deposits its egg in a randomly chosen nest.
2. The nests with the best results, that is, with high-quality eggs, will be considered in the next generation.
3. The number of nests available is a fixed parameter. The egg laid by a cuckoo can be discovered by the host bird with a probability $p_a \in (0, 1)$

In Eq. (9) the movement of CS is defined. The symbol $\oplus$, denotes entry-wise multiplication, whereas $\alpha > 0$ denotes the step size. This step size specifies the maximum distance that a particle can travel in

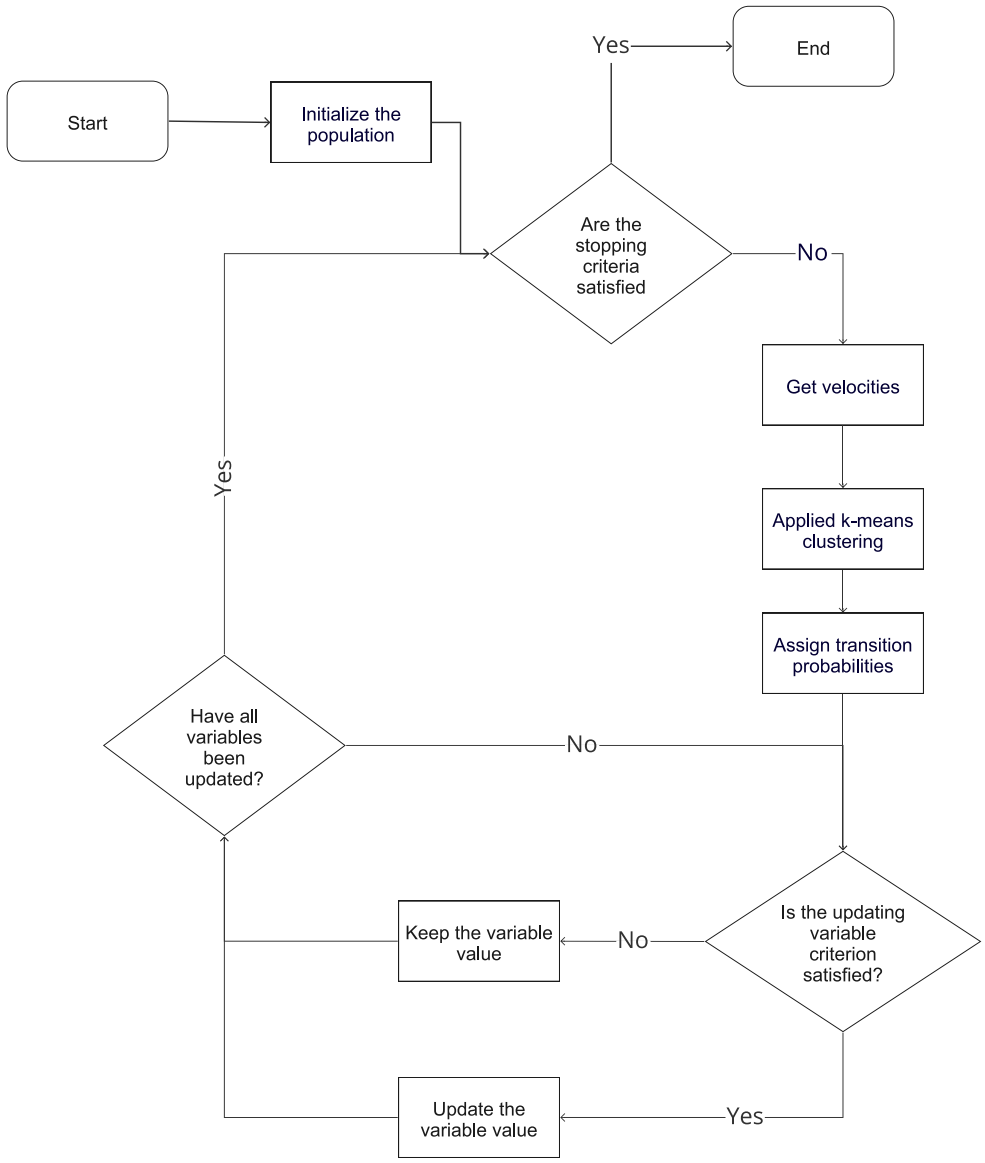


Fig. 2. Discrete hybrid k-means algorithm flow chart.

a random walk over a set number of iterations. The Lévy distribution modulates the transition probability of the Lévy flights in Eq. (10).

$$x_{i,j}^{t+1} = x_{i,j}^{t} \alpha \oplus levy(\lambda) \tag{9}$$

$$levy(\lambda) \sim g^{-\lambda}, (1 < \lambda < 3) \tag{10}$$

3.2. K-means discrete algorithm

In this subsection, the detail of the algorithm that allows discretizing the SCA and CS metaheuristics is explained, these MH, naturally work in continuous search spaces. The k-means discrete algorithm (KMDA), uses the unsupervised learning technique k-means to cluster the solutions. As input parameters, KMDA considers the list of solutions $lSol$, the metaheuristic MH to be discretized, and the list of transition probabilities $transitionProbs$, where each group obtained by applying k-means is associated with a value of transition probability. In line 4, KMDA uses the metaheuristic that is being discretized, for this case, SCA or CS, however, it can be any continuous swarm intelligence metaheuristic, and together with the list of solutions $lSol$ obtained in the previous iteration, the movement of the metaheuristic is applied, obtaining the velocity for each solution in the list of solutions. The calculation of the velocity for each dimension of the solution vector is done using the Eqs. (7) and (9). The velocity in the metaheuristics is considered by $lSol_{i,j} = |x_{i,j}^{t+1} - x_{i,j}^{t}|$. $vlSol$ identifies the list of velocities associated with the list of solutions $lSol$.

Once the list of solutions $vlSol$ has been obtained, the next step corresponds to grouping the velocities using the k-means technique for a number K of clusters. This applies in line 5 of the algorithm. The objective of applying k-means is to generate groups where the elements of each group have similar characteristics. In this case the groups that have velocities with small values and that therefore in continuous space would move very little, will be related to small transition values in our discrete space. As a result of clustering, $lSolClustered$ is obtained, where the cluster is stored for each of the components of each speed associated with the list of solutions $lSol$.

Subsequently, each component's $dimSol$ cluster is considered for each solution, and a transition probability is assigned. For the case studied here, a $K = 5$ was used and with transition probabilities [0.1, 0.2, 0.4, 0.8, 0.9]. The smallest value of the transition probability is

Table 4
Parameter setting for the hybrid cuckoo search algorithm.

Parameters	Description	Value	Scanned range
N	Number of nest	10	[5, 10, 20]
γ	Step length	0.01	0.01
κ	Levy distribution parameter	1.5	1.5
K	Number of clusters	5	[4, 5]
Iteration number	Maximum iterations	600	[200, 400, 600]

assigned to the cluster with the lowest average velocity, and in that order, the probabilities are assigned. The assignment of the transition probabilities was established intuitively and supported by previous experiences when solving binary problems. Following the idea that the first two clusters have the smallest velocities, small transition probabilities are assigned, with the aim of favoring the exploitation of space. The last two clusters that have the highest velocities, are assigned high probabilities to favor exploration of the search space, and the middle cluster has a probability of 0.4.

Finally, using a random number r_1 it is determined if the component $lSol_{i,j}$, is updated or stays the same. The higher the probability of transition, the greater the probability of change. Additionally, it is used a random number r_2, making the update be considering the best value or randomly. In our case $\beta = 0.8$. The KMDA pseudocode is shown in algorithm 1.

Algorithm 1 k-means discrete algorithm (KMDA)

1: **Function** KMDA($lSol$, MH,$transitionProbs$)
2: **Input** $lSol$, MH, $transitionProbs$
3: **Output** $lSol$
4: $vlSol \leftarrow$ getVelocities($lSol$, MH)
5: $lSolClustered \leftarrow$ appliedKmeansClustering($vlSol$, K)
6: **for** (each Sol_i in $lSolClustered$) **do**
7: **for** (each $dimSol_{i,j}l$ in Sol_i) **do**
8: $dimSolProb_{i,j}$ = getClusterProbability($dimSol$, $transitionProbs$)
9: **if** $dimSolProb_{i,j} > r_1$ **then**
10: **if** $beta > r_2$ **then**
11: Update $lSol_{i,j}$ considering the best.
12: **else**
13: Update $lSol_{i,j}$ with a random value allowed.
14: **end if**
15: **else**
16: Don't update the element in $lSol_{i,j}$
17: **end if**
18: **end for**
19: **end for**
20: **return** $lSol$

3.3. Transfer function discrete algorithm

In the case of the algorithm that uses transfer functions, the structure is very similar to KMDA. Specifically, a transfer function is applied that aims to bring the velocity values, which can take values in $\mathbb{R}$, to values between [0, 1). In this case, a v-shape transfer function, $|\tanh(v)| = |\frac{e^x - e^{-x}}{e^x + e^{-x}}|$, was used. The fundamental difference is that KMDA calculates the speeds and then applies clustering by analyzing all the values of the solutions. This can be seen in lines 4 and 5 of algorithm 1. In the case of transfer functions, this is done individually within the for loops on lines 6 and 7 of algorithm 2.

Algorithm 2 Transfer function discrete algorithm

1: **Function** Discretization($lSol$, MH)
2: **Input** $lSol$
3: **Output** $lSol$
4: **for** (each Sol in $lSol$) **do**
5: **for** (each $dimSol$ in Sol) **do**
6: $vdimSol \leftarrow$ getVelocity($dimSol$, MH)
7: $dimSolProb \leftarrow$ appliedTransferFunction($vdimSol$)
8: **if** $dimSolProb > r_1$ **then**
9: **if** $beta > r_2$ **then**
10: Update $lSol_{i,j}$ considering the best.
11: **else**
12: Update $lSol_{i,j}$ with a random value allowed.
13: **end if**
14: **else**
15: Don't update the element in $lSol_{i,j}$
16: **end if**
17: **end for**
18: **end for**
19: **return** $lSol$

4. Results

In this Section, the experiments developed to evaluate the behavior of the discrete hybrid algorithm are detailed, in addition to analyzing the findings found when optimizing the bridge. The results are divided into four subsections. In Section 4.1 it is explained how the selection of the hyperparameters used by the algorithm was made. Later in Section 4.2, the results of the experiments that identify the contribution of the algorithm in the optimization result are detailed. Later in Section 4.3, the results are compared with other implementations.

Python 3.6 was used to create the algorithm, along with a PC running Windows 10, a core i7 processor, and 32 GB of RAM. To see if the difference is statistically significant, the Wilcoxon signed-rank [38]method was used. The 0.05 p-value was chosen. The methods described in [39] was used to choose the test. The Shapiro–Wilk normality test is used initially in this process. The Wilcoxon signed-rank is recommended to check the difference if one of the populations is not normal and has the same number of points.

4.1. Parameter setting

The methodology used to select the correct parameters was adapted from the procedure defined [27]. To obtain an adequate selection of the parameters, we used three measures defined by the Eqs. (11) to (13). For the generation of values, each combination of parameters was executed five times. The set of parameters explored and selected for CS is shown in Table 4. For the calculation of the best performance, each of the indicators is constructed to have values between 0 and 1. The closer to 1, the better the performance. These values are plotted on a radar chart and the area under the curve is calculated. The set of indicators that takes the largest area, corresponds to the best performance.

1. The percentage deviation of the best value obtained in the specific execution, compared to the best value obtained of all the runs:

$$bSolution = 1 - \frac{BestTotalValue - BestValue}{BestTotalValue} \tag{11}$$

2. The percentage deviation of the worst value obtained in the specific execution, compared to the best value obtained of all the runs:

$$wSolution = 1 - \frac{BestTotalValue - WorstValue}{BestTotalValue} \tag{12}$$

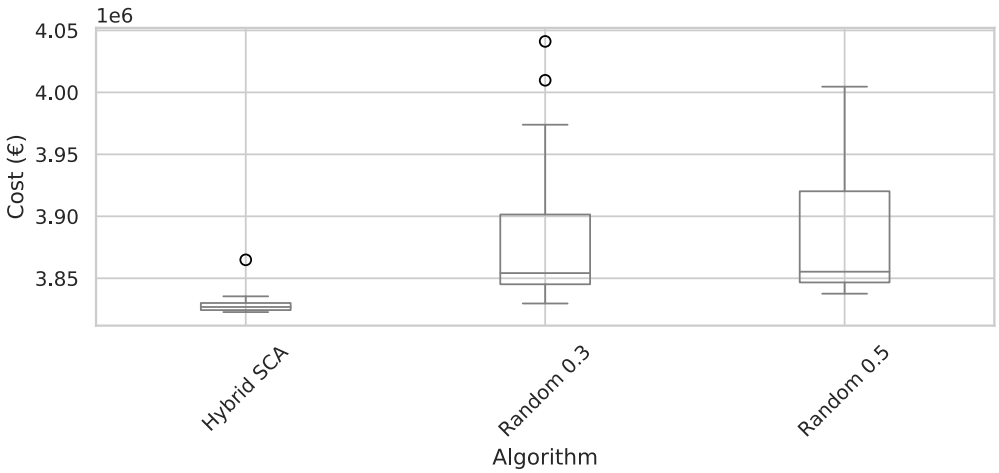


Fig. 3. Cost boxplots for Random0.5, Random0.3, and SCA Algorithms.

3. The percentage deviation of the average value obtained in the specific execution, compared to the best value obtained of all the runs:

$$aSolution = 1 - \frac{BestTotalValue - AverageValue}{BestTotalValue} \quad (13)$$

4.2. Insight into discrete algorithm

This Section aims to evaluate the contribution of the KMDA operator in the result of the optimization of the bridge. Two random discretization operators were designed, $Random0.5$ and $Random0.3$. Specifically, these operators do not execute the clustering in line 5 of the algorithm 1 or the probability assignment in line 8. The value of $dimSolProb_{i,j}$ is replaced by 0.5 in the case of $Random0.5$ and 0.7 in the case of $Random0.3$ (30% probability of transition). The rest of the code remains unchanged. These operators were applied to the SCA metaheuristics and the bridge cost optimization problem. The results are shown in Table 5 and Fig. 3.

In Table 5, the result of 30 executions for each of the operators mentioned above, together with the descriptive statistics, are shown. In this experiment, the objective function corresponds to cost optimization. From each of the optimizations, the bridge that obtained the best cost is registered, the emissions obtained for that bridge, and the time it took for the optimization. In the case of the cost results, we see that Hybrid SCA obtains the best values and smaller dispersion of the results. When emissions are analyzed, we see that the Hybrid SCA case is more robust in all indicators. Additionally, the latter suggests an essential correlation between optimizing the cost of the bridge and reducing its emissions. The Wilcoxon test shows that the results are statistically significant. When the execution times are analyzed, the results are similar in the three evaluated operators. When comparing the cost boxplots, Fig. 3, we visually observe the robustness of Hybrid SCA against random operators.

4.3. Algorithm comparisons

This Section will evaluate KMDA's performance against other implementations that have effectively solved combinatorial problems. The first algorithm used was an implementation of simulated annealing (SA) proposed in [40] and used to optimize prestressed concrete precast road bridges and later applied to other structural design problems [28,41]. For the second comparison, the algorithm detailed in Section 3.3 is used, which performs the discretization procedure using the v-shape $|tanh(v)|$ function. To make the comparison, all the algorithms are executed 30 times to optimize costs and 30 times to optimize CO_2 emissions. The results are recorded in Tables 6, and 7. Additionally, the results are shown through box diagrams in Figs. 4, and 5.

In Table 6, the cost optimization for SA, transfer SCA, hybrid SCA, and hybrid CS is shown. When analyzing the descriptive statistics, we see that the cost results obtained by the hybrid algorithms are very similar and superior to those for SA and transfer CSA in the minimum, maximum, average, and deviation obtained. When applying the Wilcoxon test, it indicates that the difference is not significant between the hybrid algorithms and if it is significant between hybrid SCA with respect to SA and transfer SCA. When applying the Wilcoxon test, it indicates that the difference is not significant between the hybrid algorithms and if it is significant between hybrid SCA with respect to SA, transfer SCA and transfer CS. In the case of the emissions obtained in cost optimization, we observe that both hybrid algorithms consistently obtain very similar results and are more robust than SA and transfer SCA. Again, the strong relationship between optimizing costs and reducing emissions of CO_2 is observed in the hybrid algorithms. When analyzing the times, we see that SA is 21.5% slower than Hybrid SCA, being this very similar to transfer SCA and hybrid CS. When comparing the boxplots, Fig. 4, the similarity of the hybrid methods is visually observed, and the robustness of the results concerning the other methods.

Table 7 shows the comparison between discrete algorithms that use k-means (Section 3.2), SA, and transfer functions (Section 3.3), optimizing CO_2 emissions. When analyzing the amount of CO_2 produced during the emissions optimization process, we see that the results of the hybrid method are better than the one that uses transfer functions and SA in all indicators. This is also visually verified by comparing the different boxplots in Fig. 5. However, when analyzing the costs obtained in the optimization of emissions, it does not behave equivalent to the optimization of costs. In this case, a significant dispersion is observed in all algorithms. The range between Max and Min is significant in all five algorithms. This correlation between cost optimization and emission reduction of CO_2 identified in Table 5, is related to the fact that the different grades of steel obtained from the BEDEC [30] database have the same amount of emissions. In the case of the CO_2 optimization, different elastic limit values are obtained for structural steel without producing essential variations in its objective emission function, but the highest in terms of cost.

In Table 8 the design variables results have been shown for cost and CO_2 optimization objectives. As it can be seen, there are some differences between cost and CO_2 optimization design results. The first one is the yield stress (f_{yk}) obtained for the best individual. It is observed that cost optimization results from this variable 275 MPa while CO_2 optimization gets 460 MPa. This difference is due to the difference in structural steel's cost and emission values. As shown in Table 1 as the cost increase, as the value of the yield stress increases, the value is the same for emissions for all yield stress values. Thus,

Table 5
Cost minimization results for 30 executions of Random0.5, Random0.3, and discrete hybrid SCA algorithms.

Run	Random0.5			Random0.3			Hybrid SCA		
	Cost (€)	CO_2 (kg)	Time (s)	Cost (€)	CO_2 (kg)	Time (s)	Cost (€)	CO_2 (kg)	Time (s)
1	3 841 685.5	9 423 182.3	7545.1	3 854 631.0	9 441 992.7	7434.5	3 830 092.8	9 390 104.1	7835.5
2	3 838 056.5	9 417 710.5	8121.4	3 841 685.5	9 423 182.3	7892.6	3 864 886.9	9 480 536.6	7945.0
3	3 856 001.6	9 455 145.0	6978.9	3 868 347.8	9 487 298.0	7112.7	3 826 395.0	9 388 407.3	7873.4
4	4 004 603.5	9 837 622.5	7984.7	4 041 117.6	9 915 536.1	8001.2	3 825 919.0	9 385 589.0	7929.9
5	3 837 584.6	9 406 572.5	6921.5	3 863 494.3	9 467 939.5	6893.2	3 823 801.1	9 385 004.6	7916.1
6	3 920 211.2	9 618 810.5	8021.3	4 009 757.4	9 837 067.3	8021.3	3 835 442.1	9 391 386.1	7911.4
7	3 863 494.3	9 467 939.5	7214.8	3 835 377.4	9 395 269.5	7324.6	3 826 324.6	9 389 542.7	7920.3
8	4 004 603.5	9 837 622.5	7498.1	3 973 917.2	9 747 159.4	7568.3	3 826 206.4	9 385 730.7	7935.5
9	3 920 211.2	9 618 810.5	8210.4	3 844 805.5	9 422 679.4	7901.4	3 830 234.3	9 387 716.3	7858.2
10	3 867 325.2	9 485 202.2	7645.7	3 938 023.8	9 657 116.7	7923.5	3 825 188.7	9 385 235.2	7931.7
11	3 920 211.2	9 618 810.5	7645.2	3 912 499.4	9 593 267.2	7234.8	3 828 878.5	9 387 047.9	7748.9
12	3 847 797.6	9 432 071.6	8024.1	3 840 298.2	9 419 023.8	8024.1	3 831 864.4	9 388 519.9	7719.6
13	3 844 078.0	9 432 582.0	7643.7	3 847 990.2	9 432 380.4	7701.4	3 823 462.5	9 384 378.1	7637.7
14	3 848 079.2	9 419 256.1	7891.4	3 844 078.0	9 432 582.0	7903.2	3 828 178.8	9 386 856.2	7819.4
15	3 927 551.4	9 631 163.6	7798.4	3 920 211.2	9 618 810.5	7923.2	3 826 847.5	9 386 123.4	7933.9
16	3 853 756.2	9 458 527.3	7234.1	3 847 713.1	9 431 886.6	8001.5	3 824 311.6	9 384 796.6	7687.4
17	3 854 631.0	9 441 992.7	8102.3	3 851 331.6	9 451 618.9	8114.7	3 822 723.1	9 384 013.6	6165.3
18	4 004 603.5	9 837 622.5	7743.6	3 829 666.1	9 398 360.6	6902.6	3 824 024.1	9 384 655.0	7767.6
19	3 844 695.3	9 425 683.0	7893.9	3 844 407.2	9 425 168.7	7745.2	3 824 115.0	9 384 852.9	7918.0
20	3 840 156.4	9 402 992.6	7745.1	3 853 756.2	9 458 527.3	7801.4	3 829 979.1	9 387 820.3	7891.1
21	3 858 728.1	9 455 868.2	7874.5	3 846 266.3	9 424 806.5	7931.6	3 823 245.0	9 384 270.9	7870.4
22	3 846 266.3	9 424 806.5	7534.2	3 856 001.6	9 455 145.0	7345.2	3 828 654.6	9 388 464.0	7945.4
23	3 868 347.8	9 487 298.0	7654.9	3 858 728.1	9 455 868.2	7791.5	3 827 333.5	9 386 286.3	7895.7
24	3 853 062.4	9 444 842.4	7943.4	3 930 520.1	9 638 238.9	8002.3	3 824 394.8	9 384 837.7	7876.2
25	4 004 603.5	9 481 380.6	7653.2	3 866 161.5	9 481 380.6	7754.8	3 830 913.2	9 388 051.0	7855.7
26	3 920 211.2	9 618 810.5	7896.7	3 853 062.4	9 444 842.4	7931.9	3 829 366.5	9 387 824.7	7668.7
27	3 844 407.2	9 425 168.7	7745.7	3 867 165.6	9 474 659.0	7742.5	3 833 463.0	9 391 767.1	7731.3
28	3 847 873.6	9 432 179.5	7653.3	3 847 714.8	9 447 150.9	7509.8	3 824 394.8	9 384 837.7	7845.4
29	3 851 331.6	9 451 618.9	7694.9	3 844 695.3	9 425 683.0	7654.7	3 823 562.7	9 384 427.5	7696.0
30	3 867 165.6	9 474 659.0	7893.5	3 938 023.8	9 657 116.7	8032.4	3 830 124.4	9 388 127.9	7947.6
Average	3 883 377.8	9 512 198.4	7713.6	3 879 048.3	9 512 058.6	7704.1	3 828 477.6	9 389 907.0	7789.3
Max	4 004 603.5	9 837 622.5	8210.4	4 041 117.6	9 915 536.1	8114.7	3 864 886.9	9 480 536.6	7947.6
Min	3 837 584.6	9 402 992.6	6921.5	3 829 666.1	9 395 269.5	6893.2	3 822 723.1	9 384 013.6	6165.3
Std	55 639.3	130 762.0	312.2	54 164.3	134 417.5	340.2	7 622.2	17 251.2	320.7
Wilcoxon p-value	4.1e−4		1.7e−4						

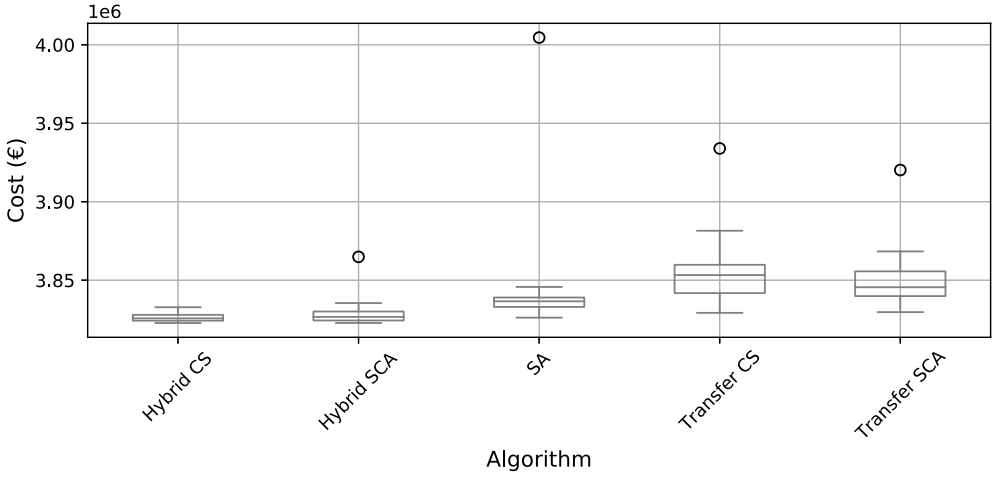


Fig. 4. Cost boxplots for SA, discrete transfer SCA, discrete hybrid SCA, and discrete hybrid CS Algorithms.

the CO_2 optimization takes a higher yield stress value because the strength capacity of these steels is higher and is capable of resisting more stresses with less material. Regarding cells dimensions it can be seen that CO_2 best design gets higher values for both upper (h_{c_1}) an lower (h_{c_1}) cells heights. In the case of the cost objective function, the best individual gives a null value to the height of the upper cell. The objective of these cells is to reduce the distance between stiffened zones to reduce the web plate's reduction. For cost optimization, the upper cell does not accomplish this function or is almost not enough to be relevant for the design, while for CO_2 emissions objective function, it allows a better cross-section behavior and takes a positive value. Regarding bottom flange stiffeners ($n_{s_{f_2}}$), both optimization objectives remove this elements for the optimum design. This is because, in negative bending moments, sections exist a concrete slab in the bottom flanges that do not allow the instability of the bottom flange plate, while in positive bending moment sections, this plate is in tension and, consequently, cannot buckle, and these elements are not necessary. Regarding the other variables, both optimization designs give a similar value being the higher difference in the beam height value (h_b), where CO_2 optimization takes a lower value due to the increase in the yield stress (f_{yk}) that allow reducing the Section by increasing the structural steel resistance.

Table 6
Cost minimization comparison for 30 executions of SA, discrete transfer function SCA and CS, discrete hybrid SCA, and CS algorithms. Time is measured in seconds.

Run	SA			Transfer SCA			Transfer CS			Hybrid SCA			Hybrid CS		
	Cost (€)	CO_2 (kg)	Time	Cost (€)	CO_2 (kg)	Time	Cost (€)	CO_2 (kg)	Time	Cost (€)	CO_2 (kg)	Time	Cost (€)	CO_2 (kg)	Time
1	3 829 112	9 393 007	9196	3 854 631	9 441 992	7497	3 853 274	9 447 811	7621	3 830 092	9 390 104	7835	3 825 644	9 385 453	7975
2	3 845 663	9 442 138	7589	3 841 685	9 423 182	7822	3 853 926	9 449 454	7741	3 864 886	9 480 536	7945	3 825 115	9 385 192	7976
3	3 829 827	9 390 569	9687	3 868 347	9 487 298	7889	3 861 618	9 468 696	7812	3 826 395	9 388 407	7873	3 825 644	9 385 453	7891
4	3 834 439	9 395 041	9719	3 837 467	9 411 814	7635	3 837 015	9 413 229	7635	3 825 919	9 385 589	7929	3 830 529	9 387 861	7959
5	3 836 720	9 393 995	9430	3 863 494	9 467 939	7786	3 859 823	9 464 319	7653	3 823 801	9 385 004	7916	3 822 875	9 384 095	7930
6	3 832 832	9 394 394	9198	3 838 032	9 396 760	7795	3 933 952	9 636 170	7563	3 835 442	9 391 386	7911	3 827 681	9 386 457	7983
7	3 837 598	9 398 873	9291	3 835 377	9 395 269	7317	3 844 299	9 425 351	7752	3 826 324	9 389 542	7920	3 824 141	9 384 712	8008
8	3 841 417	9 408 629	9271	3 839 077	9 400 419	7876	3 832 605	9 397 118	7976	3 826 206	9 385 730	7935	3 827 522	9 386 379	7971
9	3 826 259	9 391 263	9225	3 844 805	9 422 679	7832	3 853 274	9 447 811	7698	3 830 234	9 387 716	7858	3 827 541	9 386 388	7949
10	3 837 246	9 398 956	9691	3 867 325	9 485 202	7880	3 857 615	9 458 840	8043	3 825 188	9 385 235	7931	3 825 756	9 387 883	8055
11	3 838 964	9 399 136	9507	3 833 501	9 406 118	7556	3 829 202	9 389 102	7467	3 828 878	9 387 047	7748	3 824 519	9 384 899	8267
12	3 844 258	9 420 045	9668	3 840 298	9 419 023	7903	3 861 618	9 468 696	7894	3 831 864	9 388 519	7719	3 831 847	9 388 511	8292
13	3 840 202	9 408 438	9557	3 844 078	9 432 582	7509	3 854 516	9 450 954	7735	3 823 462	9 384 378	7637	3 827 980	9 386 681	8291
14	4 701 903	11 582 022	9856	3 848 079	9 419 256	7789	3 836 478	9 406 458	7642	3 828 178	9 386 856	7819	3 823 891	9 384 589	8267
15	4 004 603	9 837 622	9956	3 920 211	9 618 810	7820	3 853 274	9 447 811	7756	3 826 847	9 386 123	7933	3 825 444	9 385 765	8148
16	3 837 030	9 407 814	9504	3 840 156	9 402 992	7886	3 860 073	9 450 935	7985	3 824 311	9 384 796	7687	3 823 063	9 384 870	8210
17	3 838 077	9 398 394	9705	3 851 331	9 451 618	7740	3 851 195	9 427 987	7463	3 822 723	9 384 013	6165	3 832 782	9 401 328	8308
18	3 826 142	9 389 610	9793	3 829 666	9 398 360	7905	3 848 626	9 422 617	7683	3 824 024	9 384 655	7767	3 828 246	9 386 736	8370
19	3 836 306	9 393 541	9326	3 844 407	9 425 168	7902	3 839 130	9 400 681	7843	3 824 115	9 384 852	7918	3 831 724	9 388 450	8249
20	3 829 965	9 397 333	9912	3 853 756	9 458 527	7736	3 839 701	9 402 112	7722	3 829 979	9 387 820	7891	3 824 459	9 384 869	8236
21	3 834 063	9 395 196	9590	3 846 266	9 424 806	7921	3 851 195	9 427 987	7463	3 823 245	9 384 270	7870	3 830 466	9 387 831	7897
22	3 838 868	9 397 515	9535	3 856 001	9 455 145	7502	3 879 874	9 500 752	7985	3 828 654	9 388 464	7945	3 825 593	9 385 428	7645
23	3 840 493	9 410 516	9238	3 858 728	9 455 868	7583	3 861 537	9 457 936	7583	3 827 333	9 386 286	7895	3 826 446	9 385 925	7912
24	3 836 563	9 399 930	9617	3 839 779	9 410 778	7903	3 881 525	9 505 995	7793	3 824 394	9 384 837	7876	3 827 796	9 386 514	7914
25	3 833 027	9 394 227	9494	3 866 161	9 481 380	7729	3 849 452	9 428 558	7642	3 830 913	9 388 051	7855	3 822 766	9 384 035	7896
26	3 834 233	9 397 503	9412	3 853 062	9 444 842	7790	3 841 782	9 409 898	7856	3 829 366	9 387 824	7668	3 822 723	9 384 013	7024
27	3 845 712	9 417 868	9565	3 867 165	9 474 659	7781	3 854 384	9 440 750	7843	3 833 463	9 391 767	7731	3 822 723	9 384 013	5218
28	3 832 969	9 403 292	9984	3 847 714	9 447 150	7552	3 858 984	9 467 252	7748	3 824 394	9 384 837	7845	3 825 907	9 385 583	7943
29	3 829 559	9 389 435	8800	3 844 695	9 425 683	7660	3 851 331	9 448 449	7654	3 823 562	9 384 427	7696	3 823 593	9 384 442	7923
30	3 834 992	9 398 075	9775	3 838 056	9 417 710	7891	3 841 344	9 423 805	7962	3 830 124	9 388 127	7947	3 830 083	9 388 051	7914
Average	3 870 301	9 487 479	9470	3 850 445	9 440 101	7746	3 854 421	9 446 251	7740	3 828 477	9 389 907	7789	3 826 483	9 386 414	7921
Max	4 701 903	11 582 022	9984	3 920 211	9 618 810	7921	3 933 952	9 636 170	8043	3 864 886	9 480 536	7947	3 832 782	9 401 328	8370
Min	3 826 142	9 389 435	7589	3 829 666	9 395 269	7317	3 829 202	9 389 102	7463	3 822 723	9 384 013	6165	3 822 723	9 384 013	5218
Std	160 135	403 661	443	17 047	43 542	159	19 179	45 846	158	7 622	17 251	320	2 954	3 137	570
Wilcoxon p-value	3.6e−4			1.4e−4											

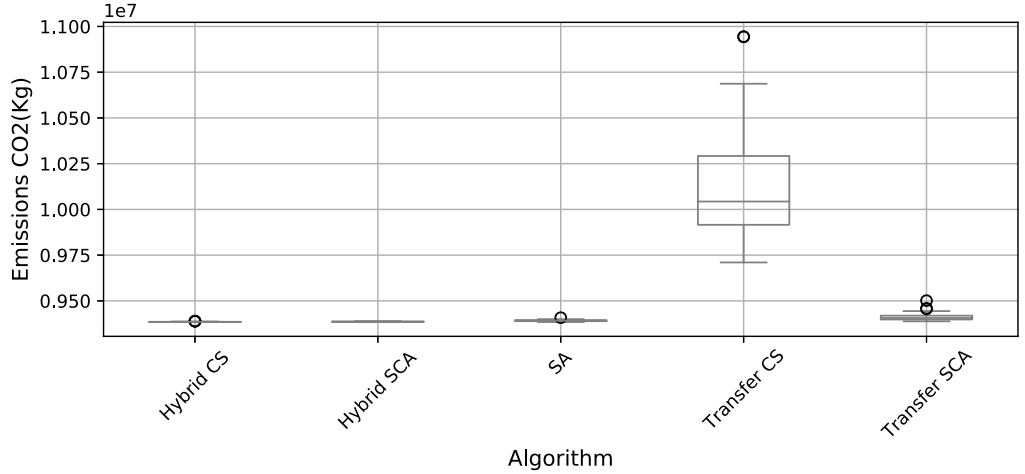


Fig. 5. Emissions boxplots for SA, discrete transfer SCA, discrete transfer CS, discrete hybrid SCA, and discrete hybrid CS Algorithms.

5. Conclusions

The present work proposes a discrete hybrid KMDA method that uses the k-means unsupervised learning technique in conjunction with SCA and SC to solve a discrete optimization issue. It was specifically applied in this work to the design of a Steel–Concrete Composite Bridge. Two experiments were designed to demonstrate the superiority of KMDA in bridge optimization. The first experiment aims to compare the proposed approach to a baseline algorithm in which the discretization stage is executed by a random operator with fixed transition probabilities. $Random0.3$ and $Random0.5$. The results showed that the incorporation of the k-means operator allows obtaining better results, as well as the reduction of the dispersion of the objective values. In the comparison, it was obtained for cost optimization that Hybrid SCA reduced the value by 1.4% compared to $Random0.5$ and by 1.3% compared to $Random0.3$. In the comparison of the emissions of CO_2, considering cost optimization, the hybrid SCA managed to reduce emissions by 1.3% in both cases. The Wilcoxon statistical test showed that the difference is significant. Regarding the execution times, these were similar in the different algorithms.

In the second experiment, the result of the proposed hybrid technique was compared with another frequently used method to discretize

Table 7
CO_2 minimization comparison for 30 executions of the discrete transfer function and discrete hybrid SCA and CS algorithms.

Run	SA		Transfer SCA		Transfer CS		Hybrid SCA		Hybrid CS	
	Cost (€)	CO_2 (kg)	Cost (€)	CO_2 (kg)	Cost (€)	CO_2 (kg)	Cost (€)	CO_2 (kg)	Cost (€)	CO_2 (kg)
1	4 103 199.1	9 394 472.7	3 826 062.4	9 388 245.3	4 531 487.4	10 115 112.3	4 420 699.5	9 384 170.2	4 421 107.9	9 384 371.6
2	4 104 830.5	9 393 422.6	4 107 363.5	9 398 200	4 931 000.7	10 230 565.0	3 828 339.5	9 386 782.2	4 091 626.2	9 384 500.2
3	4 095 750.7	9 386 615.5	4 106 480.8	9 419 624.4	4 789 317.2	10 033 694.7	4 092 720.6	9 385 866.9	4 093 543.7	9 385 445.5
4	4 101 735.1	9 390 010.9	3 830 420.8	9 390 027.3	4 607 448.4	9 710 063.3	4 092 681.4	9 385 020.4	4 421 714.6	9 384 676.8
5	4 431 949.9	9 390 386.3	4 423 829.8	9 388 224.6	4 724 373.8	9 870 644.4	4 428 155.9	9 387 845.9	3 823 916.4	9 384 601.8
6	4 428 021.6	9 387 786.7	4 104 429.3	9 413 019.9	4 735 134.9	9 912 923.2	3 829 500.6	9 387 354.6	3 823 680	9 384 485.3
7	4 428 243.3	9 393 282.6	4 444 861.3	9 427 586	4 730 550.6	9 924 267.8	4 427 399.4	9 387 473	4 091 769.9	9 384 571.1
8	4 424 257.7	9 387 914.4	3 845 183.8	9 417 923.6	4 503 658.7	9 765 664.9	4 093 430.2	9 385 389.5	3 823 245	9 384 270.9
9	4 099 351.4	9 395 896.2	3 828 004.3	9 391 991.1	4 212 280.9	10 052 411.1	3 831 646.9	9 388 412.7	4 421 527.7	9 384 578.5
10	3 836 602.7	9 407 896.6	4 434 043.8	9 399 596.9	4 666 593.6	9 725 656.3	3 827 295.3	9 386 273.6	4 093 543.7	9 385 445.5
11	4 436 583.5	9 399 830.4	4 440 365	9 405 285.5	4 841 631.0	10 686 873.0	4 091 126.9	9 384 254.1	3 824 016.6	9 384 651.2
12	4 099 070.8	9 391 395.7	4 098 731.9	9 398 154.5	5 058 456.2	10 525 430.7	3 824 275.7	9 384 778.9	3 823 245	9 384 270.9
13	3 837 692.9	9 394 543.3	4 097 483.5	9 395 141.9	4 257 869.8	9 838 989.8	4 093 666.6	9 385 506.1	3 826 336.9	9 385 795.0
14	4 098 859.5	9 394 851.7	4 447 519.4	9 406 304.8	4 957 796.6	10 944 461.4	4 091 859.4	9 385 667.9	3 827 656.8	9 386 445.7
15	4 092 562.3	9 385 573.4	4 149 564.8	9 501 199.2	4 721 028.2	9 992 277.7	3 828 232.5	9 386 739.1	4 426 817	9 387 185.9
16	4 437 303.8	9 398 323.1	4 111 137.6	9 411 014.2	4 925 090.5	10 312 531.2	4 091 796.4	9 384 584.1	4 096 921.1	9 387 110.4
17	4 100 494.5	9 392 570.7	4 103 984.7	9 401 026.2	4 977 891.6	10 121 176.4	3 829 720	9 387 462.8	4 101 662	9 389 447.4
18	4 424 307.7	9 386 325.9	3 852 755.9	9 455 679.3	4 552 468.0	10 031 915.7	4 090 883.3	9 384 585.2	3 824 016.6	9 384 651.2
19	4 102 823.5	9 396 581.4	3 831 846.9	9 398 755	4 611 778.6	10 355 530.8	4 100 425.2	9 388 837.8	4 093 460.5	9 385 404.4
20	4 428 702.7	9 390 314.8	3 853 687	9 444 022.8	4 928 246.3	10 324 582.5	3 823 782.1	9 384 535.6	4 423 239.1	9 385 422.2
21	4 430 019.3	9 392 768.3	3 834 232.5	9 393 079.5	4 894 711.7	10 129 787.8	3 825 979.5	9 385 618.9	3 825 050.6	9 385 167.1
22	4 102 389.5	9 391 525.7	3 843 360.9	9 430 497.2	4 878 828.2	10 058 610.4	4 096 206.3	9 386 758	4 568 165.1	9 384 308.4
23	3 829 423.6	9 390 490.5	4 132 163.3	9 459 204.2	4 891 768.4	10 229 330.1	4 094 423	9 385 878.9	3 823 948.5	9 384 617.7
24	4 106 712.4	9 395 756.6	4 111 394.3	9 413 642.1	4 598 949.9	10 406 844.4	3 826 340.7	9 385 796.9	4 092 556.6	9 384 958.9
25	3 833 087.7	9 391 660.1	3 835 567.5	9 409 985.7	4 763 740.6	9 977 672.0	4 095 292.9	9 386 307.8	3 826 660.3	9 385 954.4
26	3 823 206.8	9 384 338.0	4 109 800.5	9 405 586.3	4 614 532.1	9 738 456.8	3 828 734.7	9 386 977.1	3 823 948.5	9 384 617.7
27	4 428 242.5	9 392 728.7	4 442 164.1	9 424 594.9	5 249 949.1	10 943 039.5	3 823 782.1	9 384 535.6	3 832 507.4	9 388 836.8
28	3 832 154.7	9 388 663.3	4 108 189.4	9 406 482.7	4 371 744.8	9 740 572.8	4 100 432.8	9 388 841.5	3 825 740.1	9 388 369.9
29	4 428 372.5	9 397 013.8	4 445 608.5	9 410 269.7	4 872 910.1	10 028 070.8	3 828 738.5	9 386 978.9	4 090 843.3	9 384 114.8
30	4 096 391.3	9 387 611.7	3 837 284.5	9 401 492.1	4 576 118.0	9 997 514.0	3 832 291.8	9 388 730.6	4 092 556.6	9 384 958.9
Average	4 167 411	9 392 617.2	4 087 917.4	9 413 528.6	4 732 578.5	10 124 155.7	4 002 995.3	9 386 265.5	4 039 167.5	9 385 469.5
Max	4 437 303	9 407 896.6	4 447 519.4	9 501 199.2	5 249 949.1	10 944 461.4	4 428 155.9	9 388 841.5	4 568 165.1	9 389 447.4
Min	3 823 206	9 384 338.0	3 826 062.4	9 388 224.6	4 212 280.9	9 710 063.3	3 823 782.1	9 384 170.2	3 823 245.0	9 384 114.8
Std	226 692	4878	232 227.7	24 665.3	231 256.6	328 416.3	192 413.8	1423.3	241 158.4	1414.8.9
Wilcoxon p-value		2.7e−4		1.2e−4		2.7e−6				

algorithms, transfer functions. Additionally, a version of SA adapted to solve civil engineering optimization problems was included. In the case of transfer functions, the solution only uses its information to discretize, unlike k-means which first analyzes the group of solutions and then discretizes. From the results of the experiments, it is once again observed that the proposed hybrid technique is superior in the results obtained. Particularly in the case of cost optimization, it is observed that Hybrid SCA reduced costs on average compared to SA by 1.1% and compared to transfer SCA by 0.57%. In the case of Hybrid CS, the result was very similar to that of Hybrid SCA, the latter being 0.05% higher on average. In the case of CO_2 optimization, we again observed that the hybrid algorithms were superior to those using transfer function and SA. Hybrid CS outperformed Transfer CS by 7.8%, and up to 0.07% compared to SA. In the case of Hybrid CSA, it outperforms Transfer CSA by 0.3% and SA by 0.07%.

Finally, an analysis of the optimum obtained was conducted, observing that outstanding results are also obtained for CO_2 emissions when costs are optimized. However, the reciprocal, that is, when emissions are optimized, does not imply that costs are optimized. This was related to the fact that different steel grades have different costs but generate the same emissions.

As a new line of research, we identified that optimization consumes a significant amount of time, near to 8000 (s), with which developing algorithms that allow reducing optimization times allows exploring a more significant number of configurations and more complex situations. When an analysis of the execution times is carried out, we observe that the primary time is consumed to evaluate the constraints. It is hypothesized that by incorporating a deep learning model that does not have essential execution times, the constraints could be modeled and replaced, thereby improving the times used in optimization. Along the same lines, to try to reduce the number of calculations, another idea to explore is to use upper bound strategies (UBS) [42] to reduce the total number of structural analyzes in design optimization.

CRediT authorship contribution statement

D. Martínez-Muñoz: Conceptualization, Software, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **J. García:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Visualization, Project administration. **J.V. Martí:** Validation, Writing – review & editing, Supervision. **V. Yepes:** Validation, Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition.

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.

Acknowledgments

The authors gratefully acknowledge the funding received from the following research projects:

- Grant PID2020-117056RB-I00 funded by MCIN/AEI/10.13039/501100011033 and by "ERDF A way of making Europe".
- Grant FPU-18/01592 funded by MCIN/AEI/10.13039/501100011033 and by "ESF invests in your future"
- Grant CONICYT/FONDECYT/INICIACION/11180056

Table 8
Design variables results for best, mean, minimum, maximum, and standard deviation values.

Variables	Unit	Cost optimization				CO_2 optimization			
		Best	Mean	Min	Max	Best	Mean	Min	Max
b	m	7	7	7	7	7	7	7	7
$alpha_w$	deg	55	63	45	87	71	62.3	45	90
h_s	mm	200	200	200	200	200	200	200	200
h_b	cm	298	304	250	381	286	311	250	388
h_{fb}	mm	410	438	400	610	620	480	400	680
t_{f1}	mm	25	27	25	57	25	26	25	39
b_{f1}	mm	300	309	300	480	300	300	300	300
h_{c_1}	mm	0	264	0	960	9	333	0	830
t_{c_1}	mm	16	16	16	17	16	16	16	16
t_w	mm	16	16	16	16	16	16	16	16
h_{c_2}	mm	33	38.8	00	90	41	32.8	0	86
t_{c_2}	mm	18	19	16	25	18	19	16	25
b_{c_2}	mm	300	302	300	370	300	300	300	300
t_{f_2}	mm	25	25	25	29	25	25	25	25
h_{s_2}	mm	150	150	150	150	150	150	150	150
ϕ_{base}	mm	6	6	6	6	6	6	6	6
ϕ_{r_1}	mm	6	6	6	6	6	6	6	6
ϕ_{r_2}	mm	6	6	6	6	6	6	6	6
n_{r_1}	u	200	278.4	200	436	200	282	200	425
n_{r_2}	u	200	280	200	403	228	270.8	200	418
s_{f_2}	mm	300	322.1	200	550	200	328	200	550
$n_{s_{f_2}}$	u	0	0	0	0	0	0	0	0
s_w	mm	450	327.5	200	600	200	347.6	200	600
s_t	mm	240	306	200	600	200	322.6	200	600
d_{st}	m	3.3	2.407	1	4.2	2.52	1.2	1	5
d_{sd}	cm	6	6.5	4.1	9.7	4.8	6.3	4	9.9
b_{fb}	mm	1000	450	200	1000	200	430	200	1000
$t_{f_{fb}}$	mm	33	29	25	35	29	29	25	34
$t_{w_{fb}}$	mm	27	28	25	35	26	29	25	35
f_{ck}	MPa	25	25	25	25	25	25	25	25
f_{yk}	MPa	275	275	275	275	460	328	275	460
f_{sk}	Mpa	500	500	500	500	500	500	500	500
h_{sc}	mm	100	100	100	100	100	100	100	100
ϕ_{sc}	mm	16	17	16	22	16	16.3	16	22

References

[1] Yepes V, Dasi-Gil M, Martínez-Muñoz D, López-Desfilis VJ, Martí JV. Heuristic techniques for the design of steel-concrete composite pedestrian bridges. Appl Sci 2019;9(16):3253.

[2] Vayas I, Iliopoulos A. Design of steel-concrete composite bridges to eurocodes. Boca Raton: CRC Press; 2017.

[3] Dillen W, Lombaert G, Schevenels M. A hybrid gradient-based/metaheuristic method for eurocode-compliant size, shape and topology optimization of steel structures. Eng Struct 2021;239:112137.

[4] Korus K, Salamak M, Jasiński M. Optimization of geometric parameters of arch bridges using visual programming FEM components and genetic algorithm. Eng Struct 2021;241:112465.

[5] Belevičius R, Juozapaitis A, Rusakevičius D, Žilėnaitė S. Parametric study on mass minimization of radial network arch pedestrian bridges. Eng Struct 2021;237:112182.

[6] Zhao X, Lu Y, Liang H, Wang Y, Yan Y. Optimal design of reinforced concrete columns strengthened with square steel tubes and sandwiched concrete. Eng Struct 2021;244:112723.

[7] Han Y, Xu B, Wang Q, Liu Y. Bi-directional evolutionary topology optimization of continuum structures subjected to inertial loads. Adv Eng Softw 2021;155:102897.

[8] Briseghella B, Fenu L, Lan C, Mazzarolo E, Zordan T. Application of topological optimization to bridge design. J Bridge Eng 2013;18:790–800.

[9] Martínez-Muñoz D, Martí JV, Yepes V. Steel-concrete composite bridges: Design, life cycle assessment, maintenance and decision-making. Adv Civ Eng 2020;2020:8823370.

[10] Kaveh A, Bakhshpoori T, Barkhori M. Optimum design of multi-span composite box girder bridges using cuckoo search algorithm. Steel Compos Struct 2014;17(5):703–17.

[11] Pedro RL, Demarche J, Miguel LFF, Lopez RH. An efficient approach for the optimization of simply supported steel-concrete composite i-girder bridges. Adv Eng Softw 2017;112:31–45.

[12] Kaveh A, Zarandi MMM. Optimal design of steel-concrete composite i-girder bridges using three meta-heuristic algorithms. Periodica Polytech Civ Eng 2019;63(2):317–37.

[13] Talbi E-G. Combining metaheuristics with mathematical programming, constraint programming and machine learning. Ann Oper Res 2016;240(1):171–215.

[14] Caserta M, Voß S. Metaheuristics: intelligent problem solving. In: Matheuristics. Springer; 2009, p. 1–38.

[15] Juan AA, Faulin J, Grasman SE, Rabe M, Figueira G. A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. Oper Res Persp 2015;2:62–72.

[16] Calvet L, de Armas J, Masip D, Juan AA. Learnheuristics: hybridizing metaheuristics with machine learning for optimization with dynamic inputs. Open Math 2017;15(1):261–80.

[17] Talbi E-G. Machine learning into metaheuristics: A survey and taxonomy. ACM Comput Surv 2021;54(6):1–32.

[18] López-Ibáñez M, Dubois-Lacoste J, Cáceres LP, Birattari M, Stützle T. The irace package: Iterated racing for automatic algorithm configuration. Oper Res Persp 2016;3:43–58.

[19] Ries J, Beullens P. A semi-automated design of instance-based fuzzy parameter tuning for metaheuristics based on decision tree induction. J Oper Res Soc 2015;66(5):782–93.

[20] Shahzad A, Mebarki N. Data mining based job dispatching using hybrid simulation-optimization approach for shop scheduling problem. Eng Appl Artif Intell 2012;25(6):1173–81.

[21] Waschneck B, Reichstaller A, Belzner L, Altenmüller T, Bauernhansl T, Knapp A, Kyek A. Optimization of global production scheduling with deep reinforcement learning. Procedia CIRP 2018;72(1):1264–9.

[22] Jiang M, Huang Z, Qiu L, Huang W, Yen GG. Transfer learning-based dynamic multiobjective optimization algorithms. IEEE Trans Evol Comput 2017;22(4):501–14.

[23] García J, Crawford B, Soto R, Astorga G. A clustering algorithm applied to the binarization of swarm intelligence continuous metaheuristics. Swarm Evol Comput 2019;44:646–64.

[24] García J, Lalla-Ruiz E, Voß S, Droguett EL. Enhancing a machine learning binarization framework by perturbation operators: analysis on the multidimensional knapsack problem. Int J Mach Learn Cybern 2020;11(9):1951–70.

[25] Kazemzadeh Azad S. Enhanced hybrid metaheuristic algorithms for optimal sizing of steel truss structures with numerous discrete variables. Struct Multidiscip Optim 2017;55(6):2159–80.

[26] Azad SK. Monitored convergence curve: a new framework for metaheuristic structural optimization algorithms. Struct Multidiscip Optim 2019;60(2):481–99.

[27] García J, Crawford B, Soto R, Castro C, Paredes F. A k-means binarization framework applied to multidimensional knapsack problem. Appl Intell 2018;48(2):357–80.

[28] Martínez-Muñoz D, Martí JV, García J, Yepes V. Embodied energy optimization of buttressed earth-retaining walls with hybrid simulated annealing. Appl Sci 2021;11(4):1800.

[29] Crawford B, Soto R, Astorga G, García J, Castro C, Paredes F. Putting continuous metaheuristics to work in binary search spaces. Complexity 2017;2017.

[30] BEDEC ITEC materials database. Catalonia Institute of Construction Technology; 2021, https://metabase.itec.cat/vide/es/bedec [online] (accessed on January 2021).

[31] CEN. Eurocode 2: Design of concrete structures. Brussels, Belgium: European Committee for Standardization; 2013.

[32] CEN. Eurocode 3: Design of steel structures. Brussels, Belgium: European Committee for Standardization; 2013.

[33] CEN. Eurocode 4: Design of composite steel and concrete structures. Brussels, Belgium: European Committee for Standardization; 2013.

[34] Monleón S. Diseño estructural de puentes (in Spanish). València: Universitat Politècnica de València; 2017.

[35] CEN. Eurocode 1: Actions on structures. Brussels, Belgium: European Committee for Standardization; 2019.

[36] MFOM. IAP-11: Code on the actions for the design of road bridges. Madrid: Ministerio de Fomento; 2011.

[37] Mirjalili S. Sca: a sine cosine algorithm for solving optimization problems. Knowl-Based Syst 2016;96:120–33.

[38] Wilcoxon F. Individual comparisons by ranking methods. In: Breakthroughs in statistics. Springer; 1992, p. 196–202.

[39] Hays WL, Winkler RL. Statistics: Probability, inference, and decision. Tech. rep., 1970.

[40] Martí JV, Gonzalez-Vidosa F, Yepes V, Alcalá J. Design of prestressed concrete precast road bridges with hybrid simulated annealing. Eng Struct 2013;48:342–52.

[41] Payá-Zaforteza I, Yepes V, González-Vidosa F, Hospitaler A. On the Weibull cost estimation of building frames designed by simulated annealing. Meccanica 2010;45(5):693–704.

[42] Azad SK, Hasançebi O. Upper bound strategy for metaheuristic based design optimization of steel frames. Adv Eng Softw 2013;57:19–32.