Structural and Multidisciplinary Optimization (2022) 65:312
https://doi.org/10.1007/s00158-022-03393-9

Optimal design of steel–concrete composite bridge based on a transfer function discrete swarm intelligence algorithm

David Martínez-Muñoz[1] · Jose García[2] · Jose V. Martí[1] · Víctor Yepes[1]

Received: 10 February 2022 / Revised: 31 August 2022 / Accepted: 1 September 2022

Abstract

Bridge optimization can be complex because of the large number of variables involved in the problem. In this paper, two box-girder steel–concrete composite bridge single objective optimizations have been carried out considering cost and CO_2 emissions as objective functions. Taking CO_2 emissions as an objective function allows to add sustainable criteria to compare the results with cost. SAMO2, SCA, and Jaya metaheuristics have been applied to reach this goal. Transfer functions have been implemented to fit SCA and Jaya to the discontinuous nature of the bridge optimization problem. Furthermore, a Design of Experiments has been carried out to tune the algorithm to set its parameters. Consequently, it has been observed that SCA shows similar values for objective cost function as SAMO2 but improves computational time by 18% while also getting lower values for the objective function result deviation. From a cost and CO_2 optimization analysis, it has been observed that a reduction of 2.51 kg CO_2 is obtained by each euro reduced using metaheuristic techniques. Moreover, for both optimization objectives, it is observed that adding cells to bridge cross-sections improves not only the section behavior but also the optimization results. Finally, it is observed that the proposed design of double composite action in the supports allows to remove continuous longitudinal stiffeners in the bottom flange in this study.

Keywords Swarm intelligence · Steel–concrete composite structures · Bridges · Optimization · Metaheuristics · Sustainability

1 Introduction

Traditionally, structural design processes depend on methods based on common practice. Once the analysis of this first design is done, the geometry of the sections and the grade of the materials are modified based on the experience of the technician (Yepes et al. 2008). Researchers have implemented optimization methods to obtain structural designs through automated processes to reduce this need for expertise. Optimization techniques can be classified into two large groups, the first of complete techniques and the second of approximate or incomplete methods. The exact or complete approaches are the ones that produce the best result regardless of the processing time. The most commonly used strategies in integer programming are branch-and-cut and branch-and-bound. Many combinatorial optimization problems can be expressed as mixed-integer linear programming problems (Otsuki et al. 2021). These exact algorithms have had good results solving complex problems, however, when the type of constraints does not meet certain conditions or the size of the problem is very large, these algorithms do not necessarily work well. On the other hand, incomplete techniques are those that find a suitable solution that is not always the best but does so in a reasonable amount of time. Among these incomplete techniques are heuristic and metaheuristic algorithms.

These methods use heuristic or metaheuristic algorithms that allows to explore the space of possible solutions while considering both rules and randomness. A peculiarity of structural design problems is that the variables on which the problem depends are discrete, making the optimization

Responsible Editor: Makoto Ohsaki

✉ David Martínez-Muñoz
damarmu1@cam.upv.es

[1] Institute of Concrete Science and Technology (ICITECH), Universitat Politècnica de València, Camino de Vera, s/n, 46022 Valencia, Valencian Community, Spain

[2] Escuela de Ingeniería en Construcción, Pontificia Universidad Católica de Valparaíso, Avenida Brasil, 2950, 2362807 Valparaiso, Valparaiso, Chile

problem more complex. Optimization methods have been used extensively in structural problems, as can be seen in some of the literature reviews (Sarma and Adeli 1998; Hare et al. 2013; Afzal et al. 2020). These structures include reinforced concrete (RC) building frames (Liu et al. 2020), wind turbine foundations (Mathern et al. 2022), or bridge decks (Jaouadi et al. 2020). These methods have also been applied to beam (Camacho et al. 2020; García-Segura et al. 2017) and cable-stayed (Martins et al. 2020) bridges among others.

In bridges, some very complex structural optimization problems can arise due to the high number of variables. This complexity can be even greater in composite bridges, where the number of possible solutions increases due to a large number of variables (Payá-Zaforteza et al. 2010). Furthermore, steel–concrete composite bridges (SCCB) can be divided into three groups according to the cross-section: plate-girder, twin-girders, and box-girder (Vayas and Iliopoulos 2017), and its behavior differs between these types. Consequently, literature review have collected the techniques used in SCCBs' optimization (Martínez-Muñoz et al. 2020). In simplified problems, an Excel solver (Musa and Diaz 2007) or the fmincom MATLAB® function (Lv and Fan 2014) have been applied. Meanwhile, other methods have been used for more complex SCCBs, such as set-based parametric design (Rempling et al. 2019), Harmony Search (HS; Kaveh et al. 2014), Genetic Algorithm (GA), or the Imperialist competitive algorithm (Pedro et al. 2017). In the optimization algorithms, there is a family that uses swarm intelligence methods. These algorithms have also been applied to SCCB, such as Cuckoo Search (CS), Particle Swarm Optimization (PSO; Kaveh et al. 2014), Colliding Bodies Optimization (CBO), Enhanced CBO (ECBO), or Vibration Particle System (VPS; Kaveh and Zarandi 2019). Methods such as GA or Simulated Annealing (SA) have been widely used in structural optimization problems due to their easy adaptation to discrete optimization problems. On the other hand, swarm intelligence methods are usually built to optimize on continuous spaces, such as the sine cosine algorithm (SCA; Mirjalili 2016) or Jaya (Venkata Rao 2016). Recent optimization research has applied transfer functions to these algorithms to adapt them to binary (Hussien et al. 2020; Ghosh et al. 2021) problems, which is common in engineering optimization problems. These latest algorithms, under certain conditions, have made it possible to exceed the results of algorithms such as GA or SA.

To get an optimum, it is first necessary to define one objective function. In bridges, this objective function has traditionally been related to the cost or weight reduction. In SCCB optimization, the research objective function has cost in all studies (Martínez-Muñoz et al. 2020). Considering only cost as an optimization objective function means that other criteria, such as the environmental or social impact, have not been considered. In concrete bridges, many authors have applied objective functions to get more sustainable solutions, such as embodied energy (Penadés-Plà et al. 2019) or the bridge lifetime reliability (García-Segura et al. 2017).

In this study, as a first contribution, a bridge composed of steel and concrete with three sections and a single box-girder of 60–100–60 m has been modeled and optimization of costs and emissions CO_2 has been carried out. Both optimization criteria have been considered as single-goal optimizations to compare the results. By incorporating CO_2 emissions, the impact has been analyzed from the point of view of economic resources and the sustainability of the infrastructure. Additionally, three optimization algorithms have been considered: Simulated Annealing with a Mutation Operator (SAMO2), Sinus Cosinus Algorithm (SCA), and Jaya. The first is a traditional trajectory-based algorithm that has efficiently solved structural optimization problems (Payá-Zaforteza et al. 2010). The other two algorithms implemented in this study are SCA and Jaya, these correspond to swarm intelligence algorithms and naturally work in continuous search spaces. As a second contribution, a discretization method based on transfer functions (used to solve binary problems) has been proposed to adapt SCA and Jaya algorithms in order to solve the discrete optimization problem of the bridge. To evaluate the results of the discretizations, they were compared with SAMO2, which has efficiently solved structural design problems. We should also point out that this discretization method can be extended to solve other types of discrete problems. Finally, to perform the cost and emissions analysis, the SCA is used, which was the one that obtained the best result.

2 Optimization: problem description

Optimization maximizes or minimizes one objective function. This search can be done by considering the objective functions separately or together; if the criteria are considered separate, the process is called single objective optimization. On the contrary, if all criteria are considered together it is known as multi-objective optimization. In this research, the optimization objective functions are cost and CO_2 emissions considered as two different single objective optimizations. In Eq. 1, the cost objective function is defined by multiplying the unit cost of every material in the bridge by its measurement. The CO_2 emissions target function is formulated in Eq. 2. The data for CO_2 emissions consider cradle-to-gate analysis. Thus, it is necessary to consider the emissions of every process to get bridge materials on-site and execute the project. The data of prices and CO_2 emissions that are shown in Table 1 have been obtained from the Construction Technology Institute from Catalonia by the BEDEC database (BEDEC 2021). Both optimization expressions need to fulfill, throughout the entire process, the constraints imposed

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

by the regulations or recommendations represented by Eq. 3 in a general manner. The specific constraints for this optimization problem are defined in Sect. 2.3 and more concretely by Eq. 5 and Table 4 of the aforementioned section.

$$C(\mathbf{x}) = \sum_{i=1}^{n} p_i \cdot m_i(\mathbf{x}), \tag{1}$$

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

$$G(\mathbf{x}) \geq 1. \tag{3}$$

2.1 Variables

A 220 m continuous steel–concrete composite box-girder three-span bridge is proposed for optimization. The problem variables correspond to each bridge element's geometry, reinforcement, and concrete and steel grades. To reach a buildable solution, all of these variables have been discretized, configuring a discrete optimization problem. The variables discretization has been defined in Table 2. Considering this variable discretization, the number of combinations for the optimization problem corresponds to 1.38×10^{46}. Due to many possible combinations, metaheuristic techniques are justified to obtain the optimum. In total, 34 variables are considered for the global definition of this bridge optimization problem. These bridge variables have been represented in Fig. 1. According to the nature of the variables, they can be grouped into six categories. The first correspond to cross-section geometric variables, which are upper distance between wings (b), wings and cells angle (α_w), top slab thickness (h_s), beam depth (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}). Beam depth bounds correspond to $L/40$ and $L/25$, being L, the largest span length.

SCCB can take advantage of materials to a greater extent because each material that makes it up is subjected to the stresses that best resist. This would be true in an SCCB working as an statically determinate girder. In this case, the upper concrete slab would be compressed along the entire length of the bridge. This upper slab is connected to the top flanges by shear connectors. This would also stiffen the flanges plate, which avoids buckling. Moreover, in the isostatics case, the lower flanges would be subjected to tensile stress, avoiding buckling instability phenomena. However, in the present case and with the usual loads to which the bridges are subjected (mostly gravitational), negative bending stresses will occur in supported areas. This will result in reversing the forces and tensile stresses in the upper concrete slab and the compression in the lower flange. In this case, to improve the behavior of the bridge cross-section, it has been decided to materialize a concrete bottom slab in these areas in addition to the usual increase of the top slab reinforcement. To optimize the top slab reinforcement, it has been divided into a base reinforcement that is the minimum required by regulations (CEN 2013a, b, c) and two more areas, in negative bending sections, where the reinforcement is increased. The bottom slab and reinforcement increasing area lengths are described in Sect. 2.2. Accordingly, the second group of variables corresponds to 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}).

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

The last categories correspond to floor beam variables geometry, the shear connector's characteristics, and the materials' grades. Floor beam variables are defined by the floor beam width (b_{fb}), and the flanges ($t_{f_{fb}}$) and wing ($t_{w_{fb}}$) thicknesses. The shear connectors have been defined by their height (h_{sc}) and diameter (ϕ_{sc}). Finally, the yield stress from rolled steel (f_{yk}), concrete strength (f_{ck}), and reinforcement steel bars yield stress (f_{sk}) complete the variable definition. The variables are the same for all the spans of the bridge.

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 (L/40)	1	400 (L/25)	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
$n_{s_{f_2}}$	μ	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}	μ	200	1	500	301
n_{r_2}	μ	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*			12
s_w	mm	From IPE 200 to IPE 600*			12
s_t	mm	From IPE 200 to IPE 600*			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

*Following the standard series of IPE profiles (CEN 2017)

2.2 Parameters

To narrow down the problem, some variables or properties need to be fixed in every optimization problem. These fixed variables are named parameters, and they remain invariant during the whole optimization process. In this case, these parameters correspond to boundaries defined to some bridge elements, including dimension, thicknesses, reinforcement distributions, external ambient conditions, or density (among others). The values of these parameters are summarized in Table 3.

The bridge deck width (B) corresponds to 16 m, and the depth does not vary over the entire length of the bridge. In the cross-section, it has been defined by four cells: two on the upper side of the wings and two more on the bottom, as can be seen in Fig. 1. These cells allow these parts of the wing to be stiffened, creating a sheet of class one to three that does not need to be reduced according to Eurocodes (CEN 2013a, c). To allow the optimization process to define if these cells improve the structural behavior of the cross-section (and consequently are relevant to obtain a minimum of the objective function), the minimum height of these cells is fixed to zero. The boundaries of all of the variables, including the cells heights (h_{c_1}, h_{c_2}), can be seen in Table 2. The variable's boundaries have been defined following Monleón bridge design publication (Monleón 2017). The cell height (h_{c_1}, h_{c_2}) defines the floor beam depth in the zone of contact with the wings. If the cell height is smaller than the

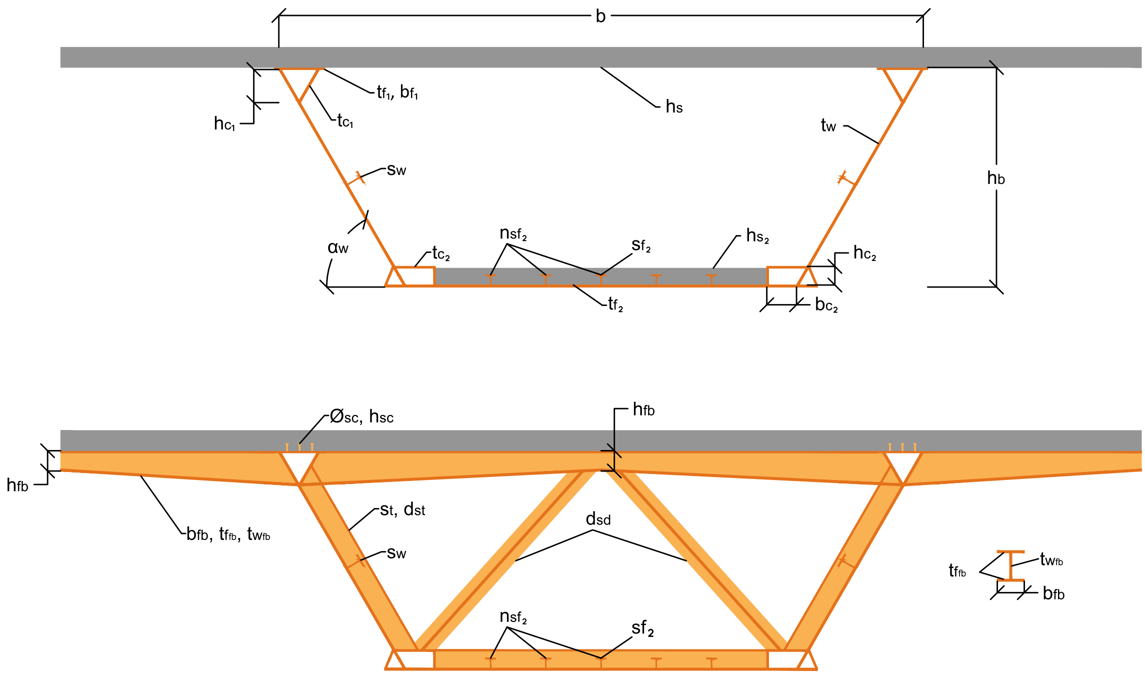


Fig. 1 Cross-section variables for SCC bridge

Table 3 Optimization problem main 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 ($t_{f_2\,min}$)	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 (CEN 2013a, b, c, 2019), IAP-11 (MFOM 2011)	
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

floor beam minimum depth (h_{fb}), then it takes that minimum value for beam depth in that zone. Profiles placed to materialize the diaphragm sections are 2 L 150 × 15. Furthermore, pre-slabs have been considered for use as a formwork. It should be noted that this element is designed to be part of the resistant section. Therefore, the measurement module of the software subtracts it from the total amount of concrete.

Base reinforcement for both the upper and the lower concrete slabs is obtained according to the minimum need for reinforcement defined in Eurocode 2 (CEN 2013a). The connection between the steel beam and concrete slab is designed to resist the whole stress of the concrete slab considering the effective width that is given by Eurocode 4 (CEN 2013c) due to shear lag. Because the only width considered as resistant (both in the concrete slab and in the lower flange) is effective, the defined steel bar reinforcement is placed only in that width.

To optimize some materials in SCCB, it is usual to modify the thicknesses of webs and flanges to reduce their amount. In this work, the variation of thicknesses has been programmed by considering a theoretical bending and shear law for a distributed load over the entire surface of the bridge. In Fig. 1, the lower flange thickness is modified along the bridge, varying from a minimum value $t_{\text{f}_2\text{min}}$ to the one defined as t_{f_2}. This variation corresponds to the theoretical bending law. In contrast, the wing's thickness varies according to the shear law from $t_{\text{w}_{\text{min}}}$ to t_{w}. The minimum value of these thicknesses has been defined according to recommendations in Monleón (2017).

Finally, steel bar reinforcements 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, it has been considered the distance defined by Eurocode 4 (CEN 2013c) for shear lag stresses that correspond with one-third of the span length. It is necessary to increase the upper slab reinforcement to resist the tension stresses produced. In this case study, it has been 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. This decision is related to the position of the center of gravity of the parabola, which is at one-third of its total length. Figure 2 shows the top slab's reinforcement distributions.

2.3 Constraints

As mentioned in Sect. 2, optimization procedures must comply about the constraints imposed on the problem. In bridge optimization, these constraints are set by the regulations

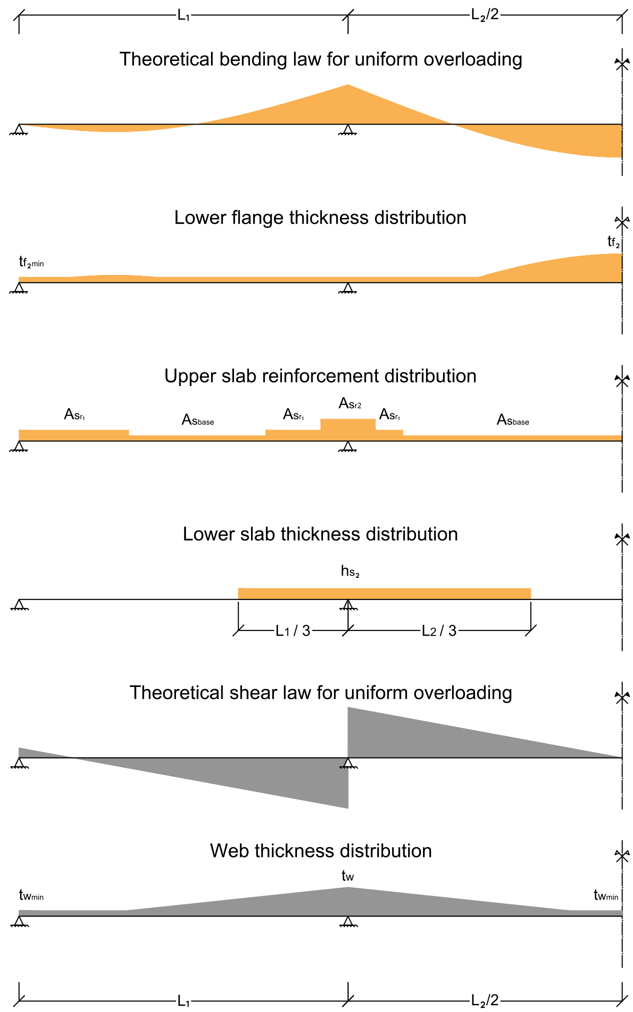


Fig. 2 Longitudinal distribution of thicknesses and steel bar reinforcements

(CEN 2013a, b, c) and recommendations (Vayas and Iliopoulos 2017; Monleón 2017).

Constraints imposed by regulations can be divided into two main groups: the Ultimate Limit States (ULS) and Serviceability Limit States (SLS). All of the loads applied and their combination are defined in regulations (CEN 2019). Table 3 summarizes the structural checks and load values that have been considered.

To check ULS for all bridge elements, it has been considered both global and local analysis. The checks considered for global analysis include flexure, shear, torsion, and flexure–shear interaction as defined in Table 4. A linear elastic analysis has been used to obtain the deflections and stresses. To get section resistance, the effective area has been considered by applying both reductions due to shear lag (CEN 2013c) and section reduction of the steel plates classified as class 4 (CEN 2013b). This last reduction is carried out by an iterative process. This procedure produces a variation of

Table 4 Structural checks and load values

Checkings		
ULS	Flexure	$M_{\mathrm{Ed}} \leq M_{\mathrm{Rd}} = \frac{W_{\mathrm{el,min}} \mathrm{f}_y}{1.05}$
	Shear	$V_{\mathrm{Ed}} \leq V_{\mathrm{Rd}} = \frac{A_{\mathrm{v}}\left(f_y/\sqrt{3}\right)}{1.05}$
	Torsion	$M_{\mathrm{T,Ed}} \leq M_{\mathrm{T,Rd}} = \frac{A_{\mathrm{T}}\left(f_y/\sqrt{3}\right)}{1.05}$
	Flexure–shear interaction	$M_{\mathrm{Ed}} \leq M_{\mathrm{Rd}} = \frac{W_{\mathrm{el,min}}\left(1-\left(\frac{2V_{\mathrm{Ed}}}{V_{\mathrm{pl,Rd}}}-1\right)^2\right)f_{\mathrm{y}}}{1.05}$
	Stiffeners	$Ist \geq \frac{\sigma_{\mathrm{m}}}{E}\left(\frac{b}{\pi}\right)^4\left(1 + w_0\frac{300}{b}u\right)$
SLS	Stress limitation	$\sigma_{\mathrm{y}} \leq fyk\sigma_{\mathrm{c}} \leq 0.6fck\sigma_{\mathrm{s}} \leq 0.8fsk$
	Fatigue	$\frac{\gamma_{\mathrm{Ff}}\ \Delta\sigma_{\mathrm{E,2}}}{\Delta\sigma_{\mathrm{C}}/\gamma_{\mathrm{Mf}}} \leq 1 \frac{\gamma_{\mathrm{Ff}}\ \Delta\tau_{\mathrm{E,2}}}{\Delta\tau_{\mathrm{C}}/\gamma_{\mathrm{Mf}}} \leq 1$
	Deflection	$L/1000$
Loads		
Dead	Self-weight	Depends on the geometry
	Dead loads	46.72 kN/m
Live	Traffic concentrated	(300, 200, 100) kN
	Traffic distributed	(9, 2.5, 2.5) kN/m^2
	Thermal heating	18° C
	Thermal cooling	− 10° C
	Win	$F_{\mathrm{wz}} = 60.84$ kN/m F_{wy} =10.78 kN/m $F_{\mathrm{wx}} = 43.12$ kN/m

the neutral fiber of the section due to the area reduction. This process must be repeated until the difference between the neutral fiber obtained between iterations is null or negligible. To attain this, a difference of 10^{-6} m has been imposed as termination criteria for the iterative process. To obtain the value of the mechanical characteristics of the homogenized section, the relationship (n) between the modulus of longitudinal deformation of concrete (E_{cm}) and steel (E_{s}) has been obtained according to Eq. 4. Concrete creep and shrinkage have been considered according to regulations (CEN 2013a, c). The procedure used for the time-dependent effects evaluation of concrete is the Ageing coefficient method defined in the annex KK of EN 1992-2:2013 (CEN 2013a). Furthermore, a local model has been considered to check ULS in-floor beams, stiffeners, and diaphragms by considering flexure, shear, buckling, and minimum mechanical characteristics checks.

$$n = \frac{E_{\mathrm{s}}}{E_{\mathrm{cm}}}. \tag{4}$$

The SLS considered for the analysis are the stress limit for materials, fatigue, and deflection as defined in Table 4. There is no explicit limit for deflection in Eurocodes. Still, the IAP-11 Spanish road bridges regulation (MFOM 2011) gives a maximum of $L/1000$ for the frequent value of live loads deflection value, with L representing the span length. This frequent value is defined in the IAP-11 as $\psi_1 Q_k$, where ψ_1 is the simultaneity factor and Q_k are the values of each live load. This loads value corresponds to the actions associated with a 1-week return period. The values of this ψ_1 coefficients are: 0.75 for the concentrated traffic load 0.40 for the distributed traffic load, 0.2 for wind load, and 0.6 for the thermal loads MFOM (2011). This has been considered as the maximum value of the deflection. In addition, geometrical and constructability requirements have been deemed.

A numerical model has been implemented in the Python (Van Rossum and Drake 2009) programming language to get the stresses and carry out all ULS, SLS, and geometrical and constructability checks defined in regulations (CEN 2013a, b, c, 2019) and recommendations (Monleón 2017; Vayas and Iliopoulos 2017) as defined in Table 4. To calculate the deflections and stresses, this software applies the displacement method considering the vertical displacements (U_z) and the spins in y and x-axes (θ_y, θ_x), taking as input data the 34 bridge variables defined in Sect. 2.1 and the loads specified in regulations. To obtain the effects due to the moving loads, all possible load combinations have been considered to get their envelope as defined in Sect. 2.3.1. This software divides every bridge span 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, discretizing the bridge into 5-m length bars. Once the stresses have been obtained, the program performs structural checks and returns the measurements, cost, CO_2 emissions, and checking coefficients. These checking coefficients correspond to the quotient between the design values of the effects of actions (E_d) and its corresponding resistance value (R_d), as shown in Eq. 5. If these coefficient values are greater or equal to one, then the section complies with the imposed restriction defined in Table 4.

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

2.3.1 Computational model description

The procedure used to obtain the deflections and stresses has been the displacement method. This method consists in solving Eq. 6.

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

In this equation, $\mathbf{f_0}$ corresponds 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. The computational model process flowchart for stresses is shown in Fig. 3.

This procedure is repeated with all load cases defined in Table 4. 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 concentrated 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 Methodology

In this section, the algorithms used are detailed. SAMO2 and a discrete version of the SCA and Jaya Algorithms were used to develop the experiments. The algorithms were chosen due to the differences in their movement methods and the ease of parameterization in the case of Jaya and SCA.

3.1 Trajectory-based algorithm: SAMO2

Simulated Annealing was developed by Kirkpatrick et al. (1983). This algorithm is an analogy based on the thermodynamic behavior of a group of atoms forming a crystal.

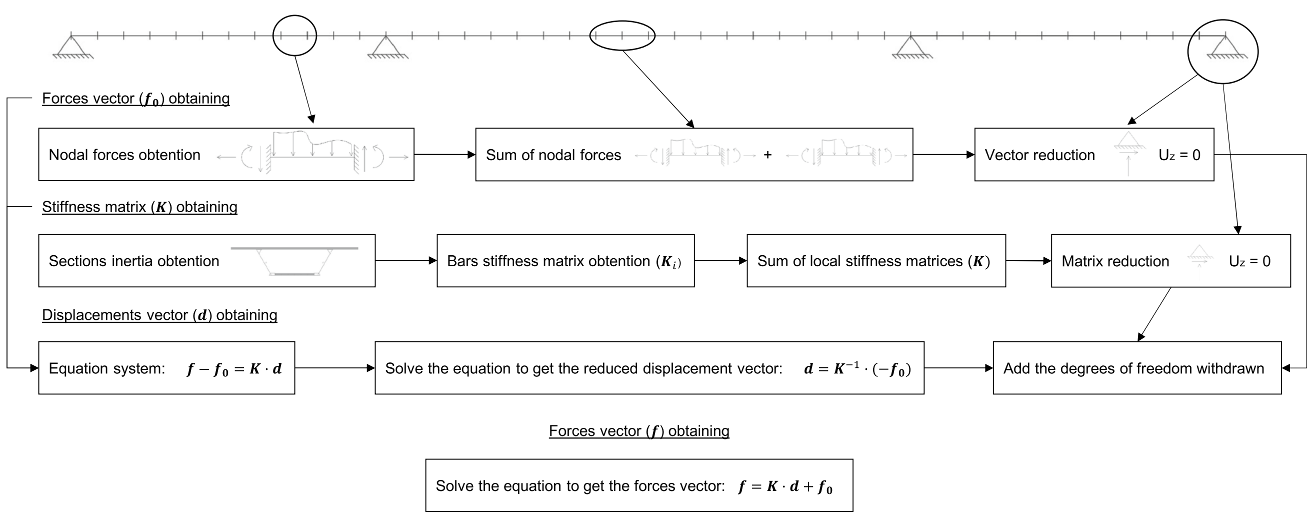


Fig. 3 Computational model process flowchart

"Annealing" refers to the chemical process of heating and cooling materials in a controlled manner. This study has chosen a variant to carry out the optimization, which includes the benefits of GAs. The GA seeks the best solution through selection, crossover, and mutation operators. To include these strategies, SAMO2 has been used. This metaheuristic introduces the probabilistic acceptance of the poorer quality solutions to flee from local optimums and directs the search towards better objective function values. For this reason, it accepts inadequate solutions with probability P_a. The expression is given by the expression of Glauber (7), where T is a parameter that decreases with time. Consequently, the probability of accepting a poor solution is reduced from the initial value, T_0. Furthermore, it includes a mutation operator that allows the algorithm to change some variables to explore the optimization process.

$$P_a = \frac{1}{1 + e^{\frac{-\Delta E}{T}}}. \tag{7}$$

The initial temperature is set according to the method proposed by Medina (2001). This algorithm depends on several parameters: Markov Chain Length (MCL), which defines the number of iterations before temperature decreases, and the Cooling Coefficient (CC), which is always less than one and represents the temperature variation. Furthermore, the mutation operator depends on the Variables Number (VN) and the Standard Deviation (SD). To fix the end of the optimization, two termination criteria have been defined for this metaheuristic: the first is the Unimproved Chains (UC) that limit the number of Markov Chains allowed without any improvement before finishing the optimization, and the second ends the process if the temperature reaches 5% of the initial (T_0). This algorithm has been chosen as it has achieved good results in other bridge optimization problems (Penadés-Plà et al. 2019).

3.2 Swarm intelligence algorithms: SCA and Jaya

3.2.1 Sine cosine algorithm (SCA)

SCA was proposed in Mirjalili (2016) 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 three random numbers r_1, r_2, r_3, which take values between 0 and 1. The update method used is shown in Eqs. 8 and 9.

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

$$x_{i,j}^{t+1} = x_{i,j}^t + r_1 \times \cos(r_2) \times \mid r_3 P_j^t - x_{i,j}^t \mid. \tag{9}$$

3.2.2 Jaya

Jaya is a swarm intelligence algorithm that allows to tackle continuous optimization problems, with and without constraints naturally. Jaya was proposed in Rao (2016) to solve benchmark problems. However, it has been used to solve complex optimization problems in different areas. The peculiar distinctive feature of Jaya from the other swarm intelligence algorithms is that it updates agents' positions in the population by considering the best and worst individuals. Additionally, binary versions of Jaya have been developed. For example, in Aslan et al. (2019) an XOR operator was integrated to be able to tackle binary problems. Another attractive quality of Jaya is that it does not have specific control parameters, and only the size of the population and the number of generations need to be defined. In Fig. 4 and Eq. 10, the flowchart and the movement of Jaya are shown, respectively.

$$x_{i,j}^{t+1} = x_{i,j}^t + r_1 (x_{best,j}^t - \mid x_{i,j}^t \mid - r_2 (x_{worst,j}^t - \mid x_{i,j}^t \mid)). \tag{10}$$

3.2.3 Discretization algorithm

The discretization algorithm is applied in the case of swarm intelligence metaheuristics because both metaheuristics work naturally in continuous spaces. As input parameters, it uses the metaheuristic, MH, and the list of discrete solutions obtained in the previous iteration, lSol. As an output, it returns a new list of discrete solutions, lSol. As the first case, the discretization algorithm obtains the velocities of the MH. This specifically corresponds to the component that modifies $x_{i,j}^t$ in Eqs. 8 to 10 . For example, in the case of Jaya, it corresponds to what is obtained from the operation $r_1 (x_{best,j}^t - \mid x_{i,j}^t \mid - r_2 (x_{worst,j}^t - \mid x_{i,j}^t \mid))$.

Subsequently, a transfer function is applied that aims to bring the velocity values, which can take values in $\mathbb{R}$, to values between [0, 1). A v-shaped transfer function has been used in this study case, $\mid \tanh(v) \mid$. With the obtained values lSolProbability, when applying the transfer function, each solution and dimension are considered, and the value is compared with a random number r_1 between [0,1). If the value of lSolProbability is greater than the random number, an update occurs in that dimension; otherwise, it is not modified. The update procedure has two possibilities: a β value is considered, and a random number r_2 is generated. If this r_2 is less than β, the value is replaced by the value of the best obtained so far for that dimension. Otherwise, a random update is performed. This last option is intended to improve the exploration of the search space.

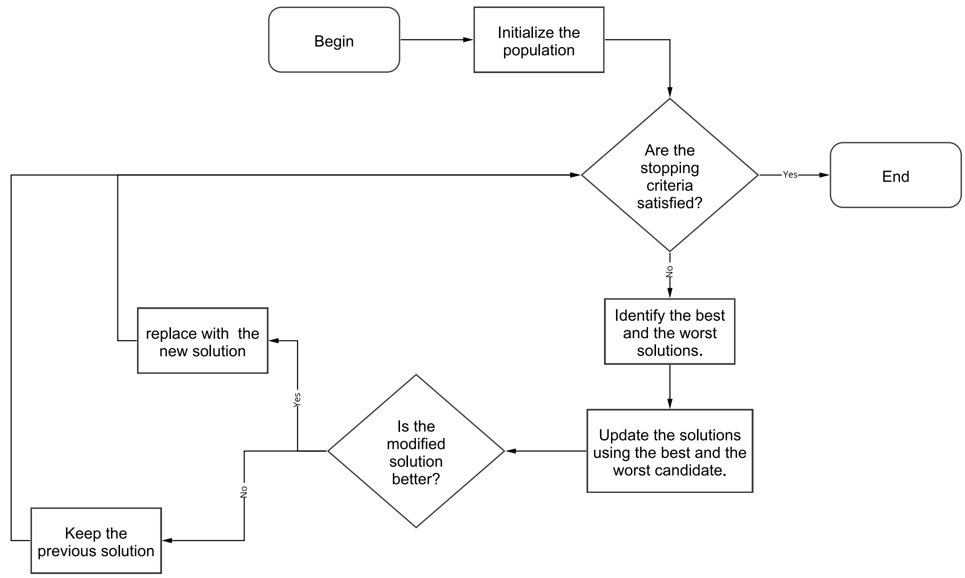


Fig. 4 The standard Jaya algorithm flowchart

Algorithm 1 Discretization algorithm

Function Discretization($lSol$, MH)
Input $lSol$
Output $lSol$
$vlSol \leftarrow$ getVelocities($Lsol$, MH)
$lSolProbability \leftarrow$ appliedTransferFunction($vlSol$)
for (each $SolProbability$ in $lSolProbability$) **do**
    for (each $dimSolProbability$ in $SolProbability$) **do**
        if $dimSolProbability > r_1$ **then**
            if $beta > r_2$ **then**
                Update $lSol_{i,j}$ considering the best.
            else
                Update $lSol_{i,j}$ with a random value allowed.
            end if
        else
            Don't update the element in $lSol_{i,j}$
        end if
    end for
end for
return $lSol$

3.3 Parameter tuning

The results obtained from the metaheuristics depend on their parameter values. Consequently, a parameter selection process is needed to choose those that give the best results for the objective function. This depends strongly on the optimization problem. Therefore, different optimization problems will result in different parameter values. The search for parameters that best fit the optimization problem is called parameter tuning.

3.3.1 SAMO2 tuning

Depending on the metaheuristic, the parameter number varies. There are algorithms with more parameters, such as SAMO2 than others with a smaller number. First, searching for the best fitting ones can become a complex problem.

Table 5 SAMO2 variables bound for DoE

Parameter	Lower bound (−)	Upper bound (+)
MCL	100	1000
SD	0%	30%
VN	1	5
CC	0.80	0.95
UC	1	5

Consequently, existing procedures allow the researcher to get the most statistically significant parameters to focus the search on the variation of these. These procedures are called *Design of Experiment s (DoE)*. In this case, a 2^k *fractional factorial design* has been carried out to get the SAMO2 parameter tuning.

In factorial designs, each factor level's possible combinations are studied in each trial or replication. This makes it possible to evaluate the change in response when the level of the factor is varied. This variation is called the effect of

Table 6 Parameter values combination and results for DoE

	MCL	SD	VN	CC	UC	Cost (€)	Time (s)	%Desv (%)	%Min (%)
1	−	−	−	−	+	4,620,844.60	1055.93	5.80	20.61
2	+	−	−	−	−	4,033,264.82	9833.70	7.46	5.27
3	−	+	−	−	−	5,109,179.62	989.63	3.37	33.35
4	+	+	−	−	+	3,831,318.29	9810.90	0.10	0.00
5	−	−	+	−	−	4,609,783.20	804.36	13.33	20.32
6	+	−	+	−	+	4,088,143.85	7819.85	7.90	6.70
7	−	+	+	−	+	4,694,176.90	787.77	11.78	22.52
8	+	+	+	−	−	4,622,308.95	7846.10	6.32	20.65
9	−	−	−	+	−	4,164,394.64	3043.06	11.89	8.69
10	+	−	−	+	+	3,831,268.79	28,688.65	0.11	0.00
11	−	+	−	+	+	4,430,917.82	3252.46	10.75	15.65
12	+	+	−	+	−	3,831,788.82	28,818.08	0.07	0.01
13	−	−	+	+	+	4,743,449.44	3182.93	9.49	23.81
14	+	−	+	+	−	3,851,070.22	30519.37	0.50	0.52
15	−	+	+	+	−	4,463,121.26	2977.47	13.46	16.49
16	+	+	+	+	+	3,839,681.73	26664.77	0.11	0.22

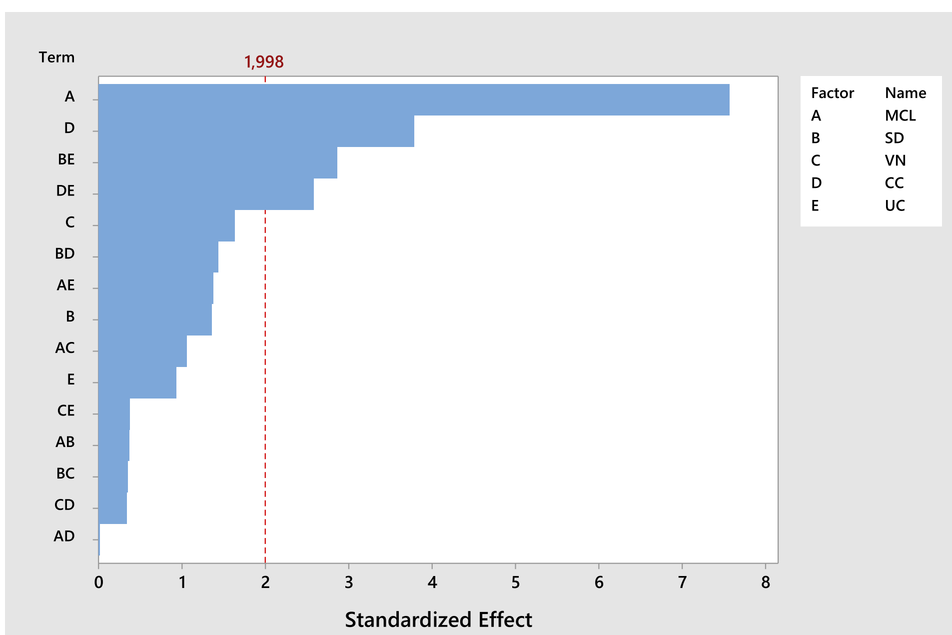


Fig. 5 Pareto chart of the standardized effects

Table 7 Parameter chosen for SAMO2 algorithm

MCL	SD	VN	CC	UC
1000	30%	1	0.8	5

Table 8 Scanned parameters for swarm metaheuristics

Parameters	Description	Value	Range
N	Number of solutions	10	[10, 20]
Iteration Number	Maximum iterations	600	[600, 800]
β	Exploration–exploitation	0.8	[0.7, 0.8]

the factor and is related to its statistical significance (Montgomery 2013). Two levels need to be assigned to the studied algorithm parameters to carry out this procedure. The studied parameters and the levels are chosen are shown in Table 5.

Because two levels are defined for each variable, 32 (2^5) runs are needed to get a complete factorial design. Furthermore, five replications need to be considered to get the average and the deviation for each experiment, obtaining 160 runs. To reduce the number of runs, it has been decided to carry out a fractional factorial DoE of resolution V. This reduces the number of runs to 80 because of the reduction of combinations to 16. A summary of the parameter value combinations is given in Table 6.

DoE Minitab (Minitab 2019) software has been used to carry out the statistical analysis. For the statistical analysis, the first-order interaction has also been considered. Accordingly, in Fig. 5, it can be seen that the parameters with more effect are MCL and CC. In addition, the interaction between UC with SD and UC is also significant. The average results of the five replicates for each of the 16 experiments are shown in Table 6.

As can be seen in Table 6, the best results correspond to experiment number 10. However, considering the cost and the optimization time, it can be observed that with a worsening of 0.001% in the objective function, the result can be got in 34.28% less time if the parameters of experiment four are used. Furthermore, the deviation between experiments ten and four is similar, 0.11% and 0.10%, respectively. Due to the improvement in computation time and slight difference in deviation and objective function value, the parameters chosen for the SAMO2 optimization correspond to experiment four, as shown in Table 7.

3.3.2 Swarm intelligence metaheuristics tuning

The methodology proposed in García et al. (2018) was used in the selection of the parameters. To obtain an adequate selection of the parameters, this methodology uses four measures defined by Eqs. (11) to (14). GBestValue corresponds to the best value obtained from all executions considering all of the parameter settings. BestValue and WorstValue correspond to the best and the worst value obtained for a given parameter setting. The parameters and explored values are shown in Table 8. In the Range column, the explored values are displayed for each parameter. The Value column corresponds to the selected value. For the generation of values, each combination of parameters was executed five times. 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. To determine the number of iterations, 600 and 800 iterations were considered. In the latter case, there were no significant differences in the optimal, but it did have an important impact on the time used.

1. The percentage deviation of the best value obtained compared to the best known value:

$$\text{bSolution} = 1 - \text{abs}\left(\frac{\text{GBestValue} - \text{BestValue}}{\text{GBestValue}}\right). \tag{11}$$

2. The percentage deviation of the worst value obtained compared to the best known value:

$$\text{wSolution} = 1 - \text{abs}\left(\frac{\text{GBestValue} - \text{WorstValue}}{\text{GBestValue}}\right). \tag{12}$$

3. The percentage deviation of the average value obtained compared to the best known value:

$$\text{aSolution} = 1 - \text{abs}\left(\frac{\text{GBestValue} - \text{AverageValue}}{\text{GBestValue}}\right). \tag{13}$$

4. The convergence time for the best value:

$$\text{nTime} = 1 - \text{abs}\left(\frac{\text{convergenceTime} - \text{minTime}}{\text{maxTime} - \text{minTime}}\right). \tag{14}$$

4 Results

4.1 Parameter tuning

In this section, the results obtained from the parameterization of the metaheuristics are shown. It should be noted that SCA and Jaya have no necessary parameters for their movements. In Fig. 6, the results of the first four configurations

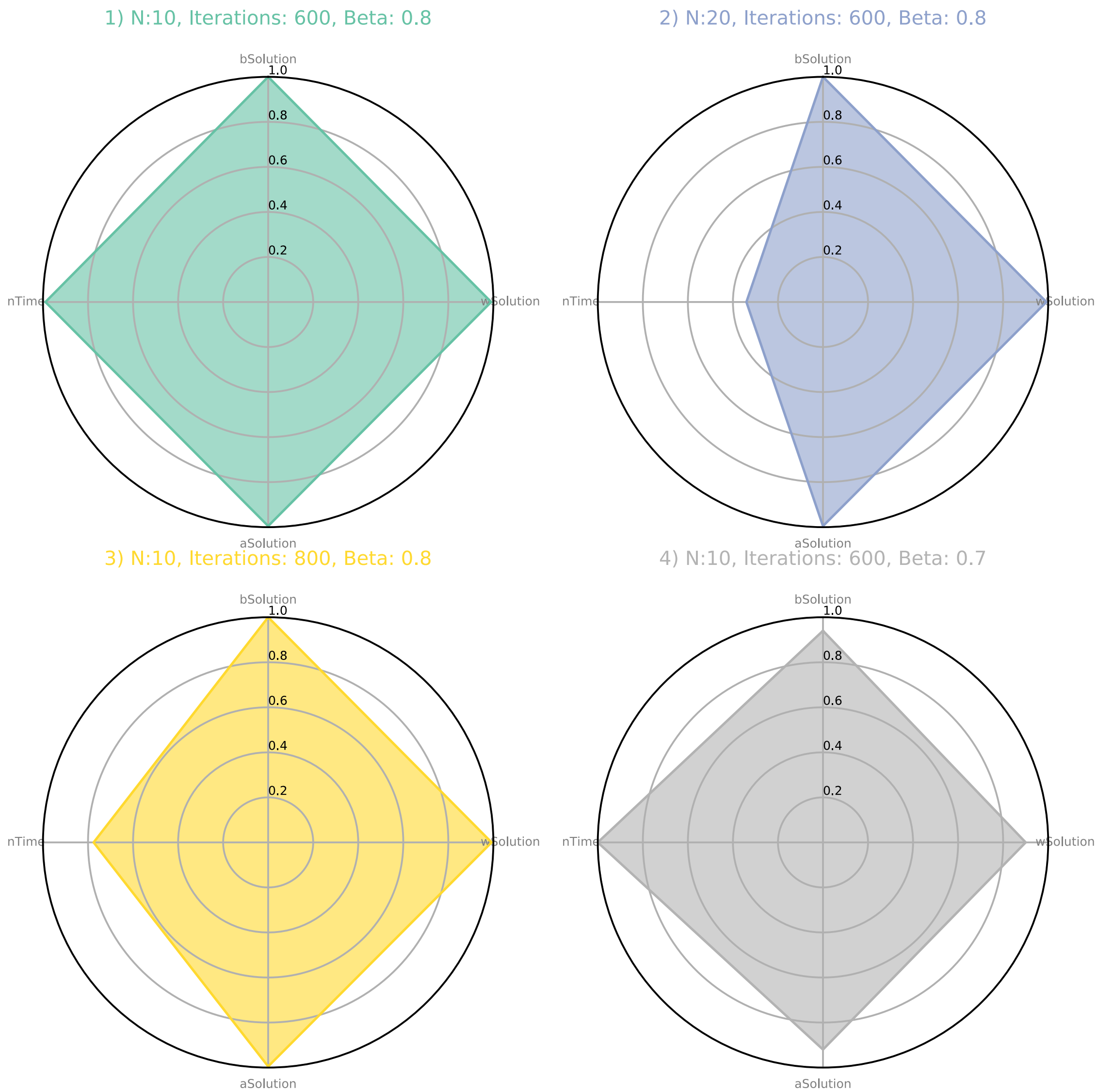
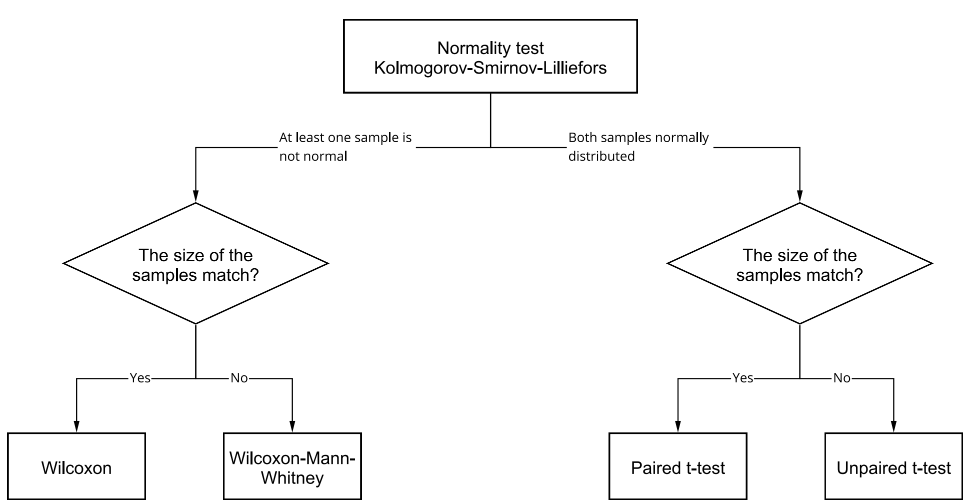


Fig. 6 Adjustment of swarm parameters by means of radar chart

Fig. 7 Statistical methodology (Hays and Winkler 1970; Lanza-Gutierrez et al. 2017)



are shown. Of the four configurations, chart 2 and chart 3 have considerably worse nTime indicators than the other two configurations. Graphs 1, 2, and 3 have similar values for aSolution, wSolution, and bSolution. Therefore, 1 has a better performance than the other two. When comparing 1 with 4, we see that nTime is similar, however, 1 is superior in the other indicators, with which the configuration $N = 10$, iteration = 600, and $\beta = 0.8$ was chosen.

4.2 Cost minimization metaheuristic comparison

This section aims to describe and analyze the results obtained by the SAMO2, discrete Jaya, and discrete SCA algorithms. For an adequate analysis, descriptive statistics are used together with boxplot visualizations. Additionally, the Kolmogorov–Smirnov–Lilliefors and the signed-rank Wilcoxon statistical tests are used to determine the statistical significance of the results. These tests were chosen according to the statistical methodology shown in Fig. 7 (Hays and Winkler 1970; Lanza-Gutierrez et al. 2017).

In this research work, 30 executions were used. The choice of 30 cases is related to the conditions for the statistical methods to be reliably applicable. Particularly according to Richardson (2010), in the case of the parametric statistical test $n > 30$ is suggested. On the other hand, in the case of the Wilcoxon test, the minimum value is 15 (Mundry and Fischer 1998). However, the value of 30, in the case of non-parametric tests, is widely used in cases of comparison of

Table 9 Cost minimization results for 30 executions of SAMO2, discrete Jaya, and discrete SCA algorithms

Run	SAMO2			Discrete Jaya			Discrete SCA		
	Cost (€)	CO_2 (kg)	Time (s)	Cost (€)	CO_2 (kg)	Time (s)	Cost (€)	CO_2 (kg)	Time (s)
1	3,829,112	9,393,007	9196	4,143,961	10,114,677	7842	3,854,631	9,441,993	7497
2	3,845,663	9,422,139	7590	4,768,396	11,451,567	7010	3,841,685	9,423,182	7822
3	3,829,828	9,390,570	9687	4,274,386	9,681,541	5682	3,868,348	9,487,298	7890
4	3,834,439	9,395,042	9719	4,167,039	9,494,257	7700	3,837,468	9,411,814	7635
5	3,836,721	9,393,995	9431	4,296,276	9,832,819	7863	3,863,494	9,467,940	7786
6	3,832,833	9,394,394	9198	3,966,049	9,664,702	6634	3,838,032	9,396,761	7795
7	3,837,599	9,398,873	9291	3,867,355	9,439,086	6690	3,835,377	9,395,270	7318
8	3,841,418	9,408,629	9271	3,923,888	9,536,557	7945	3,839,078	9,400,419	7876
9	3,826,260	9,391,263	9226	3,942,003	9,495,904	7704	3,844,805	9,422,679	7832
10	3,837,246	9,398,956	9691	3,862,458	9,465,666	5887	3,867,325	9,485,202	7880
11	3,838,964	9,399,137	9507	4,193,812	9,480,247	6545	3,833,502	9,406,118	7557
12	3,844,258	942,0046	9669	4,507,870	10,273,813	7231	3,840,298	9,419,024	7904
13	3,840,202	9,408,438	9557	3,900,545	9,469,228	6024	3,844,078	9,432,582	7509
14	4,701,903	11,582,022	9857	3,919,121	9,538,184	7932	3,848,079	9,419,256	7790
15	4,004,603	9,837,622	9957	4,191,451	10,219,877	7916	3,920,211	9,618,810	7821
16	3,837,030	9,407,815	9504	4,426,445	10,272,926	7568	3,840,156	9,402,993	7886
17	3,838,077	9,398,395	9706	3,988,854	9,625,319	7738	3,851,332	9,451,619	7740
18	3,826,143	9,389,610	9794	4,628,723	10,388,171	7639	3,829,666	9,398,361	7905
19	3,836,306	9,393,541	9326	3,884,798	9,425,667	7015	3,844,407	9,425,169	7902
20	3,829,965	9,397,333	9913	4,260,373	9,525,527	7995	3,853,756	9,458,527	7736
21	3,834,064	9,395,196	9591	4,704,005	11,472,333	6842	3,846,266	9,424,806	7922
22	3,838,869	9,397,516	9535	3,953,660	9,626,126	7902	3,856,002	9,455,145	7502
23	3,840,493	9,410,517	9239	4,363,499	9,990,653	7275	3,858,728	9,455,868	7583
24	3,836,563	9,399,930	9618	4,589,899	10,380,952	7753	3,839,780	9,410,778	7904
25	3,833,027	9,394,227	9495	4,025,218	9,846,483	3307	3,866,162	9,481,381	7730
26	3,834,233	9,397,504	9413	4,133,015	9,433,096	7848	3,853,062	9,444,842	7790
27	3,845,712	9,417,868	9566	3,889,122	9,532,552	7550	3,867,166	9,474,659	7781
28	3,832,970	9,403,292	9985	4,116,324	9,418,032	7931	3,847,715	9,447,151	7552
29	3,829,559	9,389,435	8800	4,482,044	10,843,764	7022	3,844,695	9,425,683	7660
30	3,834,992	9,398,076	9775	3,909,149	9,511,426	6425	3,838,057	9,417,710	7891
Average	3,870,302	9,487,480	9470	4,175,991	9,881,705	7147	3,850,445	9,440,101	7747
Wilcoxon *p* value	0.012	0.0012		1.92e−06	4.29e−06				

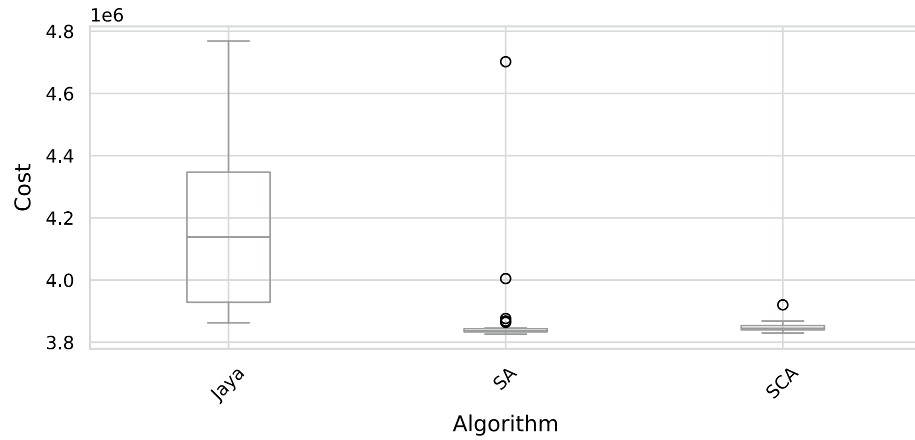


Fig. 8 Cost boxplots for SAMO2, Jaya, and SCA Algorithms

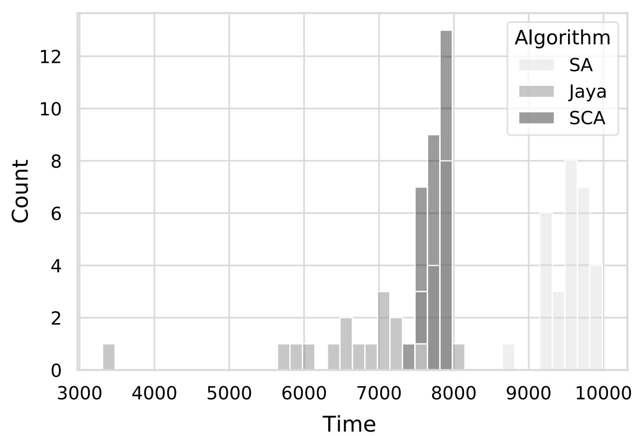


Fig. 9 Time histogram for SAMO2, Jaya, and SCA Algorithms

algorithms in the area of computer science and operations research.

The results of the 30 executions of each of the algorithms are shown in Table 9, with the settings selected for the problem of minimizing the cost of the structure. The Cost column corresponds to the minimum value obtained in the execution. Column CO_2 corresponds to the value of emissions of CO_2 for the minimum cost structure obtained. The time corresponds to the time required to obtain the minimum.

When analyzing the table, it can be observed that the best value was obtained with the SAMO2 algorithm with a cost of 3,826,142 €, followed by the minimum obtained by SCA of 3,829,666 €. However, on average, SCA is systematically higher than SAMO2, obtaining an average value for the 30 executions of 3,850,445 €, whereas SAMO2 got 3,870,302 €. The Jaya algorithm was quite a bit further away with an average of 4,175,991 €. When applying the Kolmogorov–Smirnov–Lillefors test and later the Wilcoxon test, it can be seen that the difference between SCA and SAMO2 is significant. Figure 8 shows the comparison of the cost minimization boxplots obtained by the different algorithms. It has been observed that in the case of SAMO2 and SCA, the interquartile range is very similar; however, SAMO2 has a significant number of outliers. The latter observation reinforces the robustness of SCA concerning SAMO2.

The computational time required by each algorithm to find the minimum is another interesting variable to analyze. In this case, the best time was obtained by Jaya with a value of 3306, but with very bad values (probably due to the fast convergence of the algorithm). In a comparison between SAMO2 and SCA, it is seen that it consistently performs better in all SCA executions. SCA gets an average time of 7747 s and SAMO2 9470 s. Additionally, Fig. 9 shows the histograms of the convergence times for the three algorithms. The SAMO2 histogram is shifted towards higher values, getting the worst performance. In the case of Jaya, a much more dispersed histogram reinforces the possibility of a fast convergence that implies bad results in the optimization. In the case of SCA, a much less dispersed histogram is obtained than the previous ones, with values mainly between 7500 and 7900. Finally, when the emission values associated to cost optimization results are analyzed, a clear correlation is founded between cost and CO_2 optimization. Therefore, the designs minimized by SCA also obtain minimum emission values of CO_2. On average, SCA got emissions of 9,440,101 and SAMO2 of 9,487,480 kg of CO_2.

The results obtained from cost optimization show that SCA gets the best results in cost and computation time compared with Jaya and SAMO2. Accordingly, SCA results have been considered for the cost optimization analysis. Furthermore, the correlation between cost and CO_2 optimization in these algorithms is consistent with the results obtained in other bridge optimization works. Because of this relationship between both targets, the same algorithm parameters have been chosen to get the results for CO_2 optimization.

Table 10 Cost minimization results for 30 executions of Random 0.3, Random 0.5, and discrete SCA algorithms

Run	Random 0.5		Random 0.3		Discrete SCA	
	Cost	Time (s)	Cost	Time (s)	Cost	Time (s)
1	3,841,686	7545	3,854,631	7435	3,854,631	7497
2	3,838,057	8121	3,841,686	7893	3,841,685	7822
3	3,856,002	6979	3,868,348	7113	3,868,348	7890
4	4,004,604	7985	4,041,118	8001	3,837,468	7635
5	3,837,585	6922	3,863,494	6893	3,863,494	7786
6	3,920,211	8021	4,009,757	8021	3,838,032	7795
7	3,863,494	7215	3,835,377	7325	3,835,377	7318
8	4,004,604	7498	3,973,917	7568	3,839,078	7876
9	3,920,211	8210	3,844,806	7901	3,844,805	7832
10	3,867,325	7646	3,938,024	7924	3,867,325	7880
11	3,920,211	7645	3,912,499	7235	3,833,502	7557
12	3,847,798	8024	3,840,298	8024	3,840,298	7904
13	3,844,078	7644	3,847,990	7701	3,844,078	7509
14	3,848,079	7891	3,844,078	7903	3,848,079	7790
15	3,927,551	7798	3,920,211	7923	3,920,211	7821
16	3,853,756	7234	3,847,713	8002	3,840,156	7886
17	3,854,631	8102	3,851,332	8115	3,851,332	7740
18	4,004,604	7744	3,829,666	6903	3,829,666	7905
19	3,844,695	7894	3,844,407	7745	3,844,407	7902
20	3,840,156	7745	3,853,756	7801	3,853,756	7736
21	3,858,728	7875	3,846,266	7932	3,846,266	7922
22	3,846,266	7534	3,856,002	7345	3,856,002	7502
23	3,868,348	7655	3,858,728	7792	3,858,728	7583
24	3,853,062	7943	3,930,520	8002	3,839,780	7904
25	4,004,604	7653	3,866,162	7755	3,866,162	7730
26	3,920,211	7897	3,853,062	7932	3,853,062	7790
27	3,844,407	7746	3,867,166	7743	3,867,166	7781
28	3,847,874	7653	3,847,715	7510	3,847,715	7552
29	3,851,332	7695	3,844,695	7655	3,844,695	7660
30	3,867,166	7894	3,938,024	8032	3,838,057	7891
Average	3,883,378	7714	3,879,048	7704	3,850,445	7747
Max	4,004,604	8210	4,041,118	8115	3,920,211	7922
Min	3,837,585	6922	3,829,666	6893	3,829,666	7318
std	55,639	312	54,164	340	17,048	159

4.3 Insight into the discrete algorithm

This section aims to investigate some features of the procedure given in Algorithm 1. The first attribute to investigate relates to the transfer function application in line 5 of the algorithm. Particularly, it is desired to determine whether the transfer function contributes to the discretization procedure. This is accomplished by replacing the transfer function with a uniform random operator that generates values between 0 and 1. In addition, line 8 of the algorithm configures two values for dimSolProbability. The first value is set to 0.5 (Random 0.5), corresponding to a 50% chance of executing a transition. The second value is set to 0.7 (Random 0.3), corresponding to a 30% chance of executing a transition. The results are presented in Table 10. The table shows that, on average, the values obtained by Discrete SCA are higher than those obtained by the random operator for its different parameters. In particular, it was 0.75% higher than Random 0.3 and 0.86% higher than Random 0.5. The same situation occurs when analyzing the maximums; in the case of the Random operator, these are greater than in the case of SCA. The standard deviation also shows a considerable difference, where the dispersion of the random operator has values close to 55,000, and in the case of SCA, it is 17,048. Finally, the execution times are quite similar in all cases.

A second experiment involves the parameter beta used in line 9 of the Algorithm 1. This parameter has to do with exploration and exploitation. If the criterion is met, the

Table 11 Cost minimization results for 30 executions of Discrete SCA 0.8, Discrete SCA 0.5, and Discrete SCA 0.3 algorithms

Run	Discrete SCA 0.8		Discrete SCA 0.5		Discrete SCA 0.3	
	Cost	Time (s)	Cost	Time (s)	Cost	Time (s)
1	3,854,631	7497	3,843,524	5676	3,852,498	5621
2	3,841,685	7822	3,845,599	5894	3,851,261	4947
3	3,868,348	7890	3,832,283	6109	3,859,721	5076
4	3,837,468	7635	3,829,373	5964	3,854,194	5938
5	3,863,494	7786	3,840,952	5826	3,870,990	5367
6	3,838,032	7795	3,839,057	6028	3,849,196	3840
7	3,835,377	7318	3,849,643	5444	3,839,412	4984
8	3,839,078	7876	3,839,271	5470	3,850,429	3134
9	3,844,805	7832	3,839,664	6062	3,857,432	5189
10	3,867,325	7880	3,840,787	5122	3,847,051	5963
11	3,833,502	7557	3,845,540	2349	3,851,258	5992
12	3,840,298	7904	3,844,938	5631	3,876,165	5515
13	3,844,078	7509	3,834,878	5901	3,861,690	5767
14	3,848,079	7790	3,846,365	5554	3,852,585	4235
15	3,920,211	7821	3,833,527	4493	3,859,730	3051
16	3,840,156	7886	3,827,056	5701	4,056,478	4554
17	3,851,332	7740	4,029,735	6070	3,840,542	6077
18	3,829,666	7905	3,846,775	5445	3,855,213	5681
19	3,844,407	7902	3,834,013	5946	3,859,171	5079
20	3,853,756	7736	3,838,546	5027	3,841,768	5056
21	3,846,266	7922	3,838,290	5659	3,853,698	4084
22	3,856,002	7502	3,833,359	4896	3,875,181	4084
23	3,858,728	7583	3,846,665	5643	3,844,489	2744
24	3,839,780	7904	3,849,429	5611	3,858,573	5245
25	3,866,162	7730	3,841,566	5998	3,845,703	6115
26	3,853,062	7790	3,830,238	6047	3,842,629	4220
27	3,867,166	7781	3,856,881	6057	3,840,573	5631
28	3,847,715	7552	4,158,713	6045	3,862,079	5616
29	3,844,695	7660	3,832,127	5833	3,843,622	3737
30	3,838,057	7891	3,848,890	4531	3,864,325	4549
Average	3,850,445	7747	3,862,589	5834	3,865,589	5170
Max	3,920,211	7922	4,158,713	6109	4,056,478	6115
Min	3,829,666	7318	3,827,056	2349	3,839,412	2744
std	17,048	159	66,973	746	38,284	997

update considers the best solution; otherwise, a random update is carried out. In addition to the value used (0.8), the values 0.5 and 0.3 were also investigated. The outcomes are shown in Table 11. According to the averages, the parameter with the best outcome was 0.8. This holds true when examining the maximum. In the event of the minimum, SCA 0.3 earned the best value, but SCA 0.8 was not far behind. Another notable result is the value of the standard deviation, which is significantly lower for SCA 0.8, indicating higher stability in locating the optimal ones. This is also associated with the convergence times. In the case of SCA 0.5 and 0.3 are considerably less than 0.8, but their dispersion is greater. All of the above points to a decrease in the stability of the algorithm when using these parameters.

4.4 Optimization results

This work has compared both cost and CO_2 single objective optimizations of a continuous box-girder SCCB of 220 m with three spans divided in 60, 100, and 60 m length. As stated earlier, and backed by data obtained from the algorithm comparison, the results correspond to SCA optimization. In total, 30 algorithm runs have been carried out to perform a statistical analysis of the results obtained. To get results from CO_2 emission, the same procedure as in cost optimization has been used while considering CO_2 emissions as the objective function. Because the optimization problem is similar, the same algorithm parameters have been applied for the CO_2 target.

Table 12 Design variables results for best individual and minimum and maximum values

Variables	Unit	Cost optimization			CO_2 optimization		
		Best	Min	Max	Best	Min	Max
b	m	7	7	7.16	7	7	7
α_w	deg	63	46	86	65	45	84
h_s	mm	200	200	200	200	200	200
h_b	cm	312	250	388	298	255	384
h_{fb}	mm	430	400	610	400	400	610
t_{f_1}	mm	70	25	74	34	25	79
b_{f_1}	mm	780	300	780	350	300	780
h_{c_1}	mm	440	70	820	420	0	800
t_{c_1}	mm	21	16	23	16	16	24
t_w	mm	16	16	25	16	16	28
h_{c_2}	mm	80	0	860	630	10	800
t_{c_2}	mm	16	16	25	20	16	25
b_{c_2}	mm	310	300	700	300	300	610
t_{f_2}	mm	25	25	70	27	25	60
h_{s_2}	mm	150	150	180	150	150	240
$n_{s_{f_2}}$	u	0	0	0	0	0	0
d_{st}	m	1	1	4.3	1.6	1	5
d_{sd}	m	4.3	4	9.3	4.7	4	9.5
b_{fb}	mm	300	200	900	200	200	1000
$t_{f_{fb}}$	mm	28	25	35	28	25	34
$t_{w_{fb}}$	mm	27	25	35	31	25	34
n_{r_1}	µ	200	200	439	259	200	446
n_{r_2}	µ	337	200	431	403	200	424
ϕ_{base}	mm	6	6	8	6	6	6
ϕ_{r_1}	mm	6	6	6	6	6	6
ϕ_{r_2}	mm	6	6	6	6	6	6
s_{f_2}*	mm	270	200	600	330	200	600
s_w*	mm	400	200	600	200	200	550
s_t*	mm	360	200	550	500	200	600
h_{sc}	mm	100	100	100	100	100	100
ϕ_{sc}	mm	16	16	22	16	16	22
f_{ck}	MPa	25	25	25	25	25	25
f_{yk}	MPa	275	275	275	355	275	460
f_{sk}	MPa	500	500	500	500	500	500

*Values of the standard series of IPE profiles (CEN 2017). Min and Max correspond to the maximum and minimum values obtained. Best correspond to the value obtained for the best individual

This section gives the bridge variables values obtained considering cost and CO_2 as two single objective optimizations while briefly comparing both results. Furthermore, cost and CO_2 relation for both optimizations is shown in Fig. 16, while in Fig. 14 structural and reinforcement steel amounts have been shown for both cost and CO_2 optimizations best results. In Sect. 5, a more extensive discussion of these results is provided.

The first results are related to the material's resistance, reinforcement, and shear connector diameter. For cost optimization results, concrete compressive strength (f_{ck}) and yield stress for structural steel (f_{yk}) correspond to 25 and 275 MPa for all individuals. However, for CO_2 optimization, the value of steel yield (f_{yk}) shows greater dispersion. The best individual has a 355 MPa value, as can be seen in Table 12. Reinforcement diameters (ϕ_{base}, ϕ_{r_1} , and ϕ_{r_2}) obtained from optimization correspond to 6 mm for both base and reinforcement layers. Consequently, optimization gets three reinforcement layers on the top slab. Regarding shear connectors, as in reinforcement bars, the optimization gets both lowest diameter (ϕ_{sc}) and connector length (h_{sc}). For CO_2 the optimization results show the same results.

Once the materials have been defined, the results from the geometrical variables are obtained. Steel beam depth (h_b),

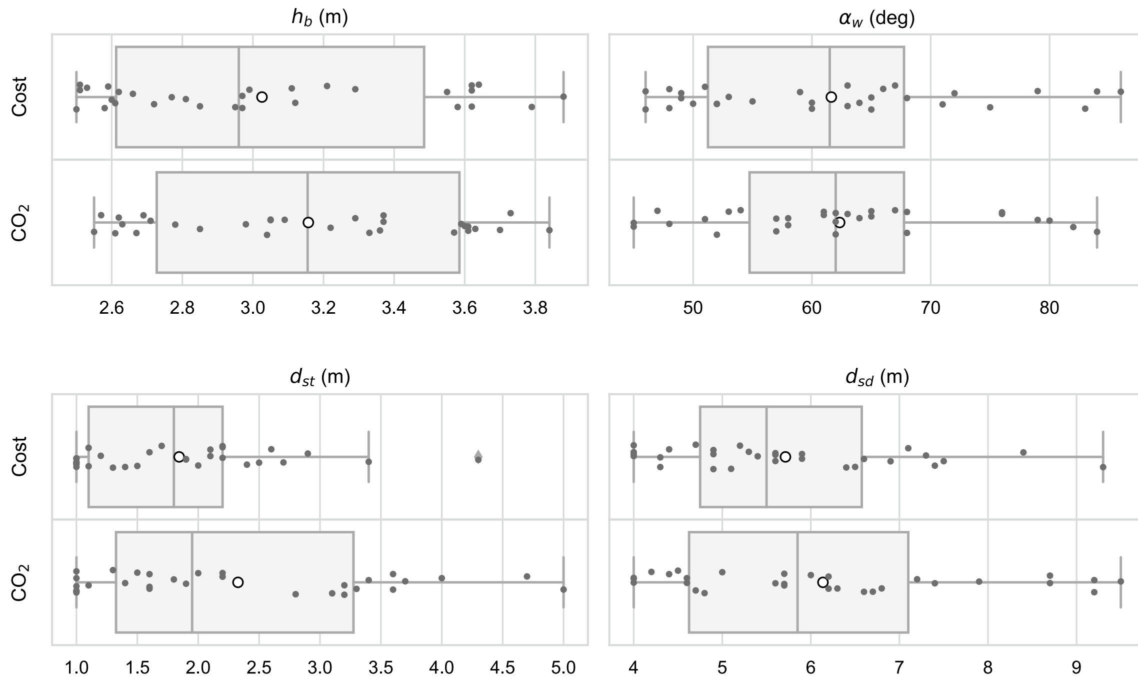


Fig. 10 Cross-section geometrical variables for cost and CO_2 optimization

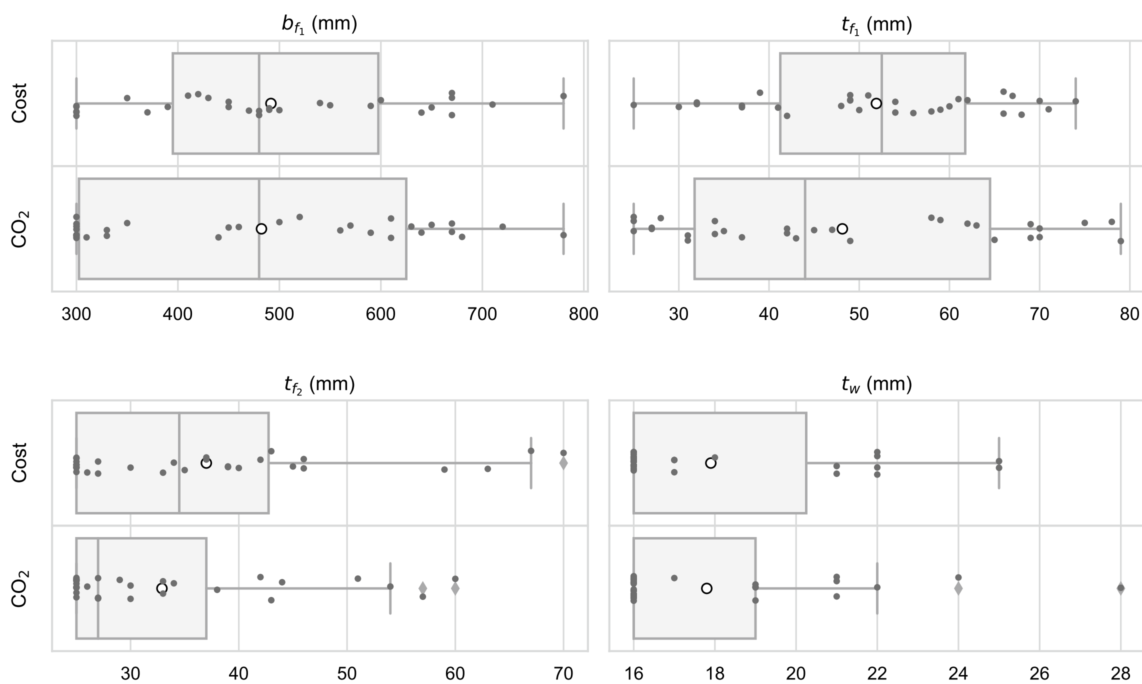


Fig. 11 Flanges and web variables for cost and CO_2 optimization

web angle (α_w), and distances between transverse stiffeners (d_{st}) and diaphragms (d_{sd}) are shown in Fig. 10. It should be emphasized that the thickness of the upper (h_s) and lower (h_{s_2}) concrete slabs gives the same result for both optimizations and takes the minimum possible value of 0.20 and 0.15 m, respectively. Meanwhile, CO_2 optimization gets higher beam depths (h_b), and stiffener (d_{st}) and diaphragm (d_{sd}) distance values than cost. The next variable values are related to the webs and flanges of the cross-section. As can be seen in Fig. 11, CO_2 takes a higher range of values than cost for both

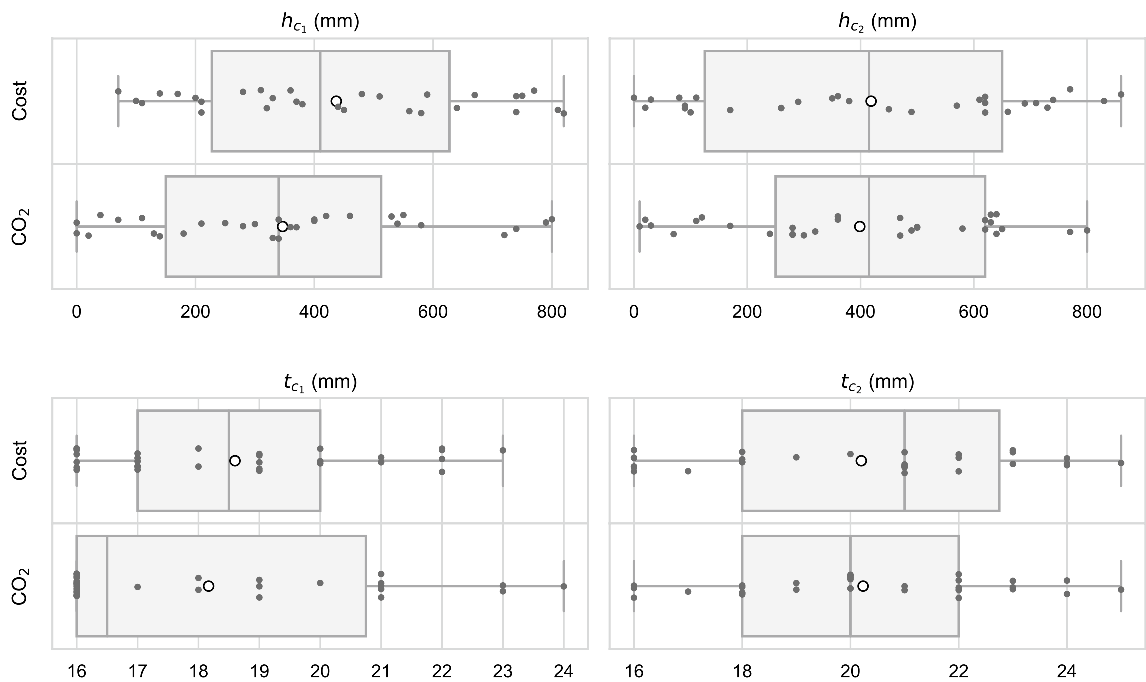


Fig. 12 Cell variables results for cost and CO_2 optimization

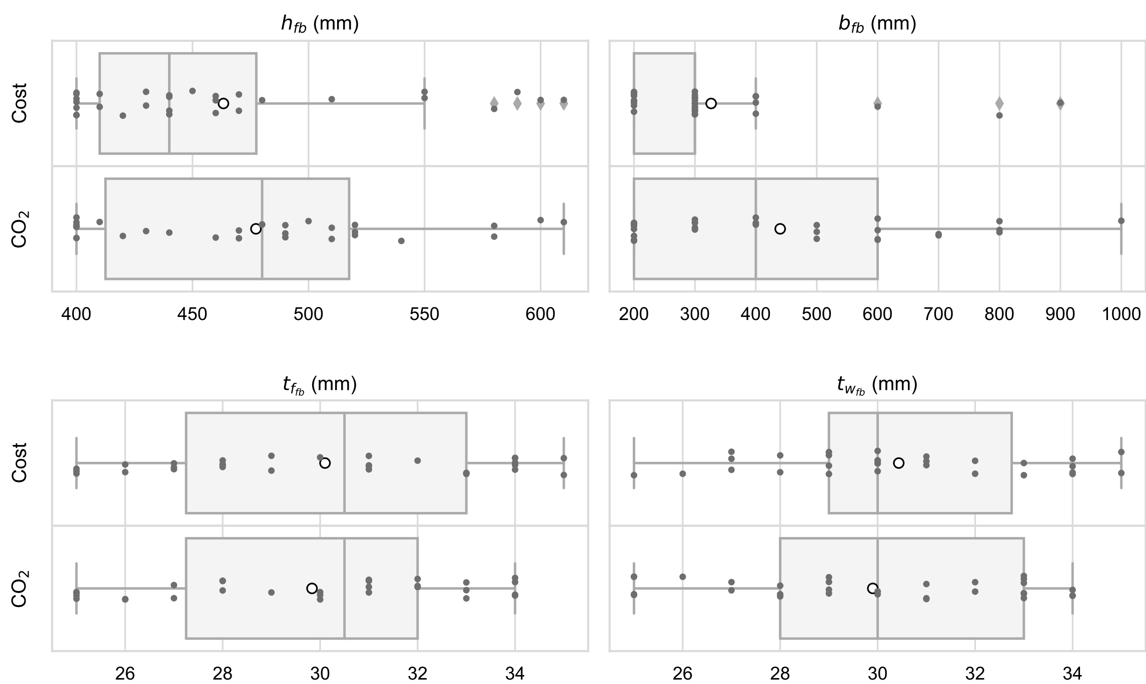


Fig. 13 Floor beam variables results for cost and CO_2 optimization

width (b_{f_1}) and thickness (t_{f_1}) of the upper flanges, while for webs (t_w) and lower flange (t_{f_2}), thickness gets lower values.

As stated in Sect. 2.2, and in accordance with Fig. 1, the cross-section of this optimization problem involves the inclusion of four cells: two uppers and two lower. The aim of these cells is to improve structural cross-section behavior, which allows better values of the objective function to be obtained. Figure 12 shows the results obtained for cell variables (h_{c_1}, t_{c_1}, h_{c_2}, t_{c_2}, b_{c_2}). It should be noted that the algorithm is left to eliminate these cells by allowing them to

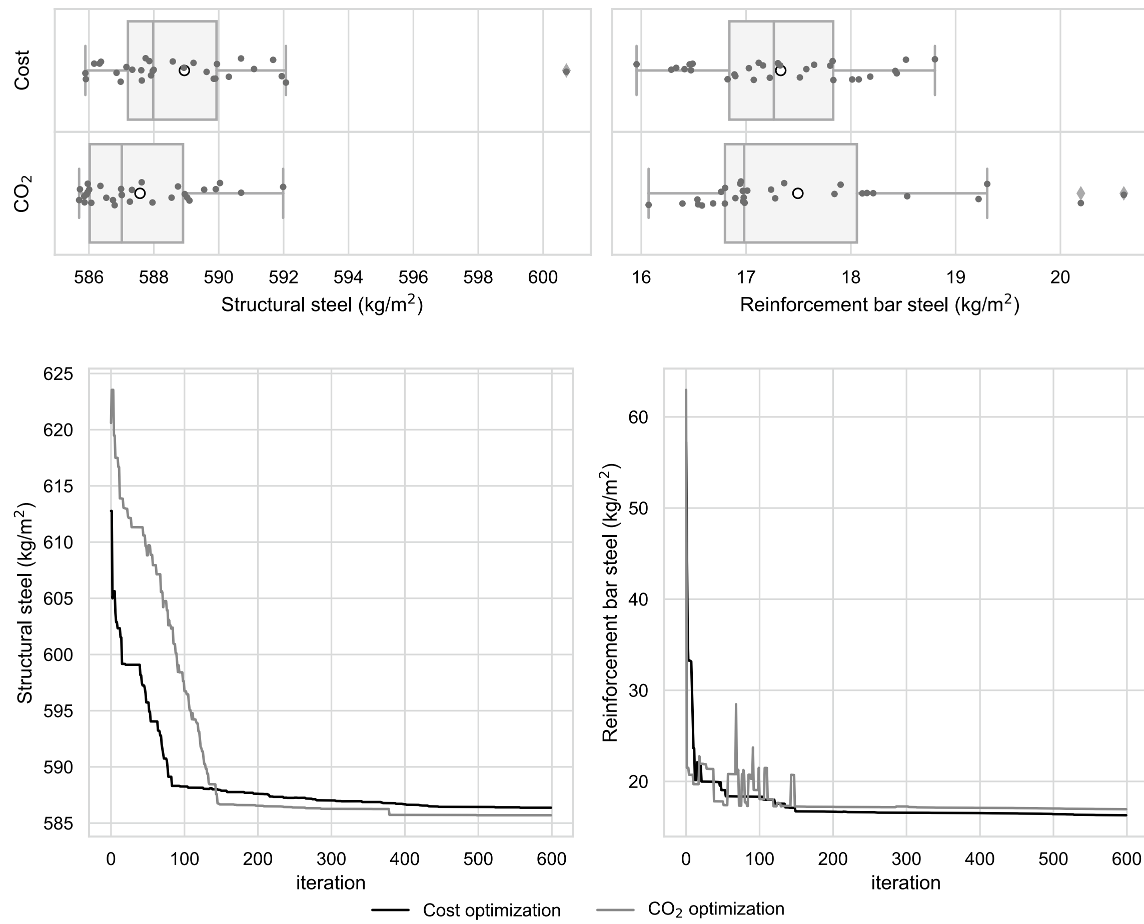


Fig. 14 Reinforcement bars and structural steel amounts for both optimization objectives

take a null value in variables that define its geometry. As can be seen in Fig. 12, both optimization objectives get values larger than zero for cell variables. It can be observed that CO_2 optimization gives in average lower values for upper cell height (h_{c_1}) and thickness (t_{c_1}). Meanwhile, for lower cells, although the average value of the results obtained is similar, the cost optimization gives a wider range of values for variables of this element (h_{c_2}, t_{c_2}, b_{c_2}).

Figure 13 gives the floor beam variables results. As can be seen in this figure, and consequently with results in Fig. 10, CO_2 optimization gives higher values of depths (h_{fb}) and widths (b_{fb}) due to the higher distances between diaphragm sections, where these floor beams are materialized. Against that, thicknesses ($t_{f_{fb}}$, $t_{w_{fb}}$) values are similar in both optimizations.

Finally, the results from material amounts and cost are represented in Figs. 14 and 16, respectively. The first figure shows that the cost target function gives higher values for rolled steel and lower values for reinforcement steel in slabs. However, CO_2 optimization gives the opposite result. The first part of Fig. 14 gives boxplots that show the values reached by the 30 individuals obtained from the algorithm runs. In the second part, the trajectory of steel amounts has been represented for the best individuals obtained from cost and CO_2 optimization. Regarding the relationship between cost and CO_2 obtained in Fig. 16, it can be seen that there is a clear relationship between both criteria for cost optimization. For this case, a straight line with equation $CO_2 = 2.5144 \cdot \text{Cost} - 241,642$ with a $R^2 = 0.98$ expresses a good fit of the straight line. By applying cost optimization for each euro reduced, a reduction of 2.5144 kg of CO_2 is obtained by applying heuristic optimization techniques. In contrast, for CO_2 optimization for the same cost, there is a large dispersion between the CO_2 values obtained. This difference between cost and CO_2 objective functions optimization is shown in Fig. 15. In this figure, cost and CO_2 trajectories have been plotted for the best individual of both optimization objectives. It can be seen that when optimizing cost, both cost and CO_2 amounts decreases following the same trend. However, when the objective function is CO_2, cost has a high variation during the optimization getting a clear difference in terms of cost at the end. Furthermore,

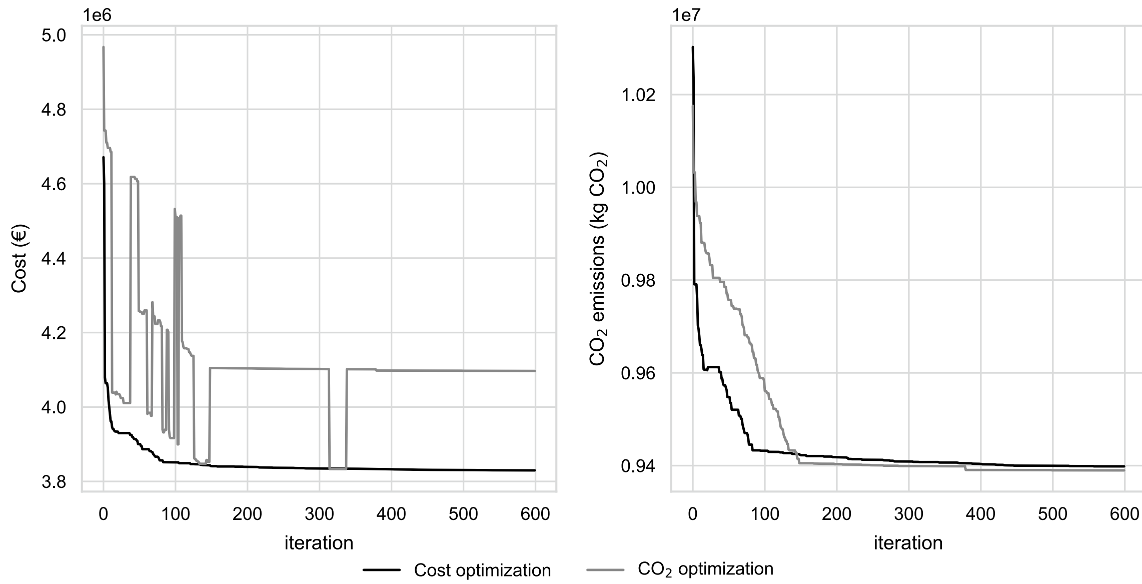


Fig. 15 Cost and CO_2 variation during the optimization process for both optimization objective functions

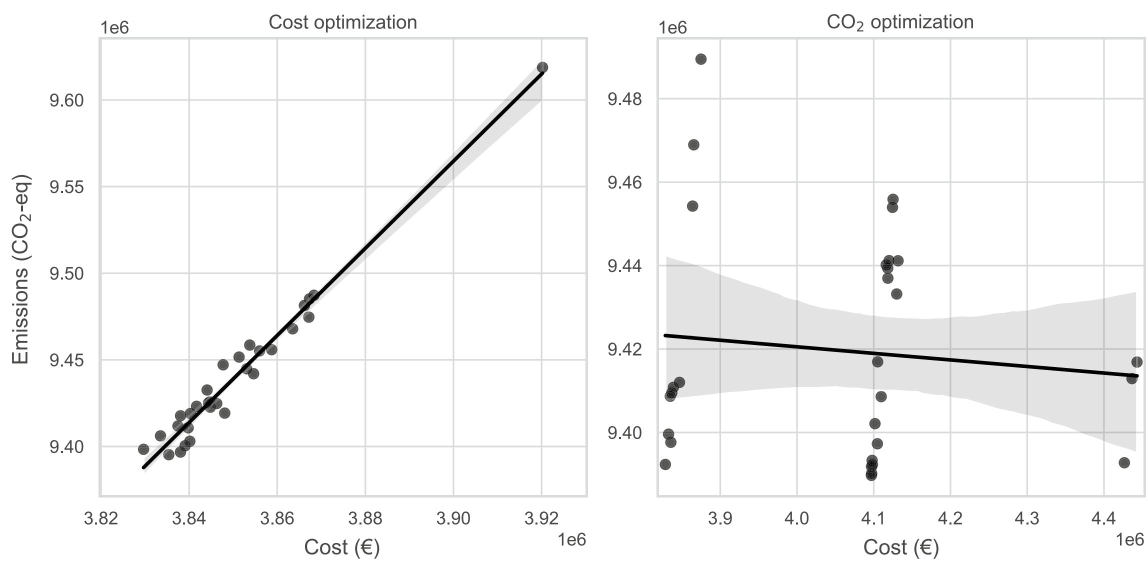


Fig. 16 Cost and CO_2 correlation considering both optimization objectives

Table 13 Lower checking coefficients values obtained from for both cost and CO_2 optimization best individuals

Constraints	Cost	CO_2
ULS		
Flexure	1.132	1.476
Flexure–shear interaction	1.155	–
SLS		
Stress in steel	1.768	1.831
Stress in concrete	1.817	1.793

This checking coefficients correspond to the expression that compares the design stresses and the resistant ones defined in Eq. 3

in Table 13 the lowest values for ULS and SLS constraints are shown.

5 Discussion

In this section, the results shown in Sect. 4.4 will be discussed. These results have been compared with earlier optimization studies of Briseghella et al. (2013) and Kaveh et al. (2014), where box-girder SCCB has been optimized. As can be seen in Table 12, the concrete strength (f_{ck}) obtained from both cost, and CO_2 optimizations is 25 MPa. This concrete strength value is a result of the high inertia of the resistant section in compressed zones that make the concrete compression lower than the strength limit defined by regulations (CEN 2013a, c). For steel, the value obtained by cost optimization is unusual. For structural steel in bridges, the expected value is 355 MPa, as in Briseghella et al. (2013). This reduction in yield stress (f_{yk}) makes a difference between a traditional design between a cost optimization design. Meanwhile, Kaveh et al. (2014) used 275 MPa steel for the bridge solution. Moreover, if the CO_2 optimization is analyzed, it can be observed that the best individual takes a 355 MPa value for yield stress (f_{yk}). This is produced because there is no difference in CO_2 emissions between different yield stress; consequently, taking higher resistance steel does not increase the value of the objective function. This higher value allows to use less material due to the higher steel resistance. Regarding reinforcing bars steel, it can be seen that the results given for both optimization objectives are yield stress (f_{sk}) of 500 MPa, which is the usual value for concrete structures (Monleón 2017; Vayas and Iliopoulos 2017). Continuing with reinforcing bar analysis, it can be seen that the optimization always gets a 6 mm diameter. The program can add up to three layers of reinforcement to the top slab. The optimization algorithm uses this possibility to adjust as far as possible the reinforcing needs, decreasing the bar diameter as a consequence. For shear connectors, it can be seen that the program takes the lowest boundary values for both heights (h_{sc}) and diameter (ϕ_{sc}) in cost and CO_2 emissions optimization of best individuals.

Next is the analysis of the main cross-section variables. It can be seen in Sect. 4.4 that cost optimization, in general, gets lower deck depth values compared with CO_2 optimization. Moreover, the CO_2 best optimization individual gets a greater web angle (α_w), which leads to a higher value of the bottom flange, obtaining a higher value of steel amount. Regarding top flanges, the results in Fig. 11 are confirmed in Table 12. CO_2 obtains lower values of width (b_{f_1}), and it is observed that this plate thickness (t_{f_1}) also takes a lower value than cost.

One of the aims of this study has been to analyze if cells added to the cross-section help reduce costs and emissions. It can be stated that this is true. The values from the cell variables show that their values are not zero in every case. Therefore, cells improve the structural behavior of the cross-section because they allow buckling of the plates to be controlled by reducing the distances between elements without stiffening. These elements allow to add a more resistant section and become longitudinal stiffening elements. The opposite occurs for bottom flange longitudinal stiffeners. If the values shown in Table 12 are observed, then, in every case the value of this element's number ($n_{s_{f_2}}$) takes the value of zero. This may lead to a contradiction because these elements prevent the lower flange from buckling when compressed (i.e., in the support areas on piles). But if the results of this research are compared with Kaveh et al. (2014), then it can be seen that in his study, he obtains the same result. In this optimization case, it is logical to obtain this result because, in sections subjected to sagging, a lower slab materializes that works in compression and do not allow the

Table 14 Material amount summary for both optimization objectives

Material	Unit	Cost optimization			CO_2 optimization		
		Best	Min	Max	Best	Min	Max
Concrete	m^3	528	528	528	528	528	528
Structural steel	kg	2,064,029	2,062,333	2,114,520	2,061,655	2,061,656	2,083,789
Reinforcement steel	kg	57,328	56,161	66,184	59,668	56,566	72,530
Cost	€	3,829,666	3,829,666	3,920,211	4,096,922	3,828,450	4,443,057
CO_2	kg	9,398,360	9,395,269	9,618,810	9,389,721	9,389,721	9489469

Min and Max correspond to the maximum and minimum values obtained. Best correspond to the value obtained for the best individual

Table 15 Cost and emissions for the best individual of both optimization objectives

Material	Unit	Cost optimizatión			CO_2 optimization		
		Measurement	Cost (€)	CO_2 (kg)	Measurement	Cost	CO_2
Concrete	m^3	528	46,918 (1.2%)	135,516 (1.4%)	528	46,918 (1.1%)	135,516 (1.4%)
Structural steel	kg	2,064,029	3,550,130 (92.7%)	8,937,246 (95.1%)	2,061,655	3,814,062 (93.1%)	8,926,966 (95.1%)
Reinforcement steel	kg	57328	80,259 (2.1%)	40,130 (0.4%)	59,668	83,535 (2.0%)	41,768 (0.4%)
		Total	3,829,666	9,398,360	Total	4,096,922	9,389,721

plate's buckling. Furthermore, in hogging sections (i.e., in span centers), this plate's main effort is tension and, therefore, buckling will not occur. Moreover, the center part of the bottom flanges is not taken into account for the strength calculation of the section due to the shear lag reductions imposed by the standards (CEN 2013c) used for the calculation. In Briseghella et al. (2013), where a topological optimization is carried out, the material in these bottom flange areas is removed because it exceeds the maximum working stress.

Finally, material amounts and objective function values obtained have been analyzed. The material summary results are shown in Table 14, while cost and CO_2 emissions are in Table 15. The relation between both objective functions has been represented in Fig. 16. As stated in Sect. 4, there is a clear relationship between cost and CO_2 optimization when choosing cost as the objective function, while on the contrary, it is not. This is due to the equality between different steel grade emissions in data obtained from BEDEC database (2021). This allows the CO_2 optimization process to obtain different yield stress values for structural steel without producing major variations in its target function, but on the higher ones in terms of cost. This contrast with related traditional concrete bridges optimization works (Yepes et al. 2012, 2015) where it is found that both cost and CO_2 optimization leads to the optimization of the other.

6 Conclusions

In this article, the design of a SCCB has been considered. This design has considered the analysis of costs and emissions of CO_2 . The proposed bridge considers 34 discrete variables that correspond to 1.38×10^{46} combinations. A discretization method was proposed through the use of transfer functions, which was applied to the SCA and Jaya metaheuristics. To evaluate the method, they were compared with SAMO2, which has previously solved structural problems efficiently. The results showed that discrete SCA was the one that obtained the best results both in the optimization values and in the execution times. SCA was 24.5% faster than SAMO2 and in the case of cost optimization, considering the average, SCA obtained 0.5% lower values than SAMO2.

Subsequently, SCA was used to compare cost and CO_2 optimizations. Regarding the results obtained, it was observed that in both optimizations bottom flange stiffeners has been removed due to the double composite action of concrete slabs on supports. Furthermore, the use of inner cells in the bridge cross-section has been considered. These cells improve the section stress resistances and reduce the distance between non-stiffened areas in steel plates. In addition, there is a clear relationship between cost and CO_2 optimization. In this case, it can be observed that one euro decrease in cost translates into 2.5144 kg of CO_2 reduction when applying heuristic optimization techniques.

Acknowledgements 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" and Grant CONICYT/FONDECYT/INICIACION/11180056.

Author contribution **D. Martínez-Muñoz**: Conceptualization, Methodology, 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**: Conceptualization, Validation, Writing – review & editing, Supervision. **V. Yepes**: Conceptualization, Validation, Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. This research has been made possible thanks to 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" and Grant CONICYT/FONDECYT/INICIACION/11180056.

Declarations

Conflict of interest The authors have no competing interests to declare relevant to this article's content.

Replication of results The data presented in this work are available on request from the corresponding author. The data are not publicly available because it is part of an ongoing study.

References

Afzal M, Liu Y, Cheng JC, Gan VJ (2020) Reinforced concrete structural design optimization: a critical review. J Clean Prod 260:120623

Aslan M, Gunduz M, Kiran MS (2019) JayaX: Jaya algorithm with XOR operator for binary optimization. Appl Soft Comput 82:105576

BEDEC (N.D.) BEDEC ITEC materials database. Catalonia Institute of Construction Technology. https://metabase.itec.cat/vide/es/bedec. Accessed Jan 2021

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

Camacho VT, Horta N, Lopes M, Oliveira CS (2020) Optimizing earthquake design of reinforced concrete bridge infrastructures based on evolutionary computation techniques. Struct Multidisc Optim 61(3):1087–1105

CEN (2013a) Eurocode 2: design of concrete structures. European Committee for Standardization, Brussels

CEN (2013b) Eurocode 3: design of steel structures. European Committee for Standardization, Brussels

CEN (2013c) Eurocode 4: design of composite steel and concrete structures. European Committee for Standardization, Brussels

CEN (2017) EN 10365:2017: hot rolled steel channels. I and H sections, dimensions and masses. European Committee for Standardization, Brussels

CEN (2019) Eurocode 1: actions on structures. European Committee for Standardization, Brussels

García J, Crawford B, Soto R, Castro C, Paredes F (2018) A k-means binarization framework applied to multidimensional knapsack problem. Appl Intell 48(2):357–380

García-Segura T, Yepes V, Frangopol DM (2017) Multi-objective design of post-tensioned concrete road bridges using artificial neural networks. Struct Multidisc Optim 56:139–150

Ghosh KK, Guha R, Bera SK, Kumar N, Sarkar R (2021) S-shaped versus V-shaped transfer functions for binary manta ray foraging optimization in feature selection problem. Neural Comput Appl 33:11027–11041

Hare W, Nutini J, Tesfamariam S (2013) A survey of non-gradient optimization methods in structural engineering. Adv Eng Softw 59:19–28

Hays WL, Winkler RL (1970) Statistics: probability, inference, and decision. Technical report

Hussien AG, Hassanien AE, Houssein EH, Amin M, Azar AT (2020) New binary whale optimization algorithm for discrete optimization problems. Eng Optim 52(6):945–959

Jaouadi Z, Abbas T, Morgenthal G, Lahmer T (2020) Single and multi-objective shape optimization of streamlined bridge decks. Struct Multidisc Optim 61(4):1495–1514

Kaveh A, Zarandi MMM (2019) Optimal design of steel–concrete composite I-girder bridges using three meta-heuristic algorithms. Period Polytech Civ Eng 63(2):317–337

Kaveh A, Bakhshpoori T, Barkhori M (2014) Optimum design of multi-span composite box girder bridges using cuckoo search algorithm. Steel Compos Struct 17(5):703–717

Kirkpatrick S, Gelatt CDJ, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680

Lanza-Gutierrez JM, Crawford B, Soto R, Berrios N, Gomez-Pulido JA, Paredes F (2017) Analyzing the effects of binarization techniques when solving the set covering problem through swarm optimization. Expert Syst Appl 70:67–82

Liu J, Liu P, Feng L, Wu W, Li D, Chen YF (2020) Automated clash resolution for reinforcement steel design in concrete frames via Q-learning and building information modeling. Autom Constr 112:103062

Lv N, Fan L (2014) Optimization of quickly assembled steel–concrete composite bridge used in temporary. Mod Appl Sci 8(4):134–143

Martínez-Muñoz D, Martí JV, Yepes V (2020) Steel–concrete composite bridges: design, life cycle assessment, maintenance, and decision-making. Adv Civ Eng 2020:8823370

Martins AM, Simões LM, Negrão JH (2020) Optimization of cable-stayed bridges: a literature survey. Adv Eng Softw 149:102829

Mathern A, Penadés-Plà V, Armesto Barros J, Yepes V (2022) Practical metamodel-assisted multi-objective design optimization for improved sustainability and buildability of wind turbine foundations. Struct Multidisc Optim 65(2):46

Medina JR (2001) Estimation of incident and reflected waves using simulated annealing. J Waterw Port Coast Ocean Eng 127(4):213–221

MFOM (2011) IAP-11: code on the actions for the design of road bridges. Ministerio de Fomento, Madrid

Minitab (2019) Minitab 19 statistical software. Minitab, State College

Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. Knowl Based Syst 96:120–133

Monleón S (2017) Diseño estructural de puentes. Universitat Politècnica de València, València (**in Spanish**)

Montgomery DC (2013) Design and analysis of experiments. Wiley, Hoboken

Mundry R, Fischer J (1998) Use of statistical programs for nonparametric tests of small samples often leads to incorrect p values: examples from animal behaviour. Anim Behav 56(1):256–259

Musa YI, Diaz MA (2007) Design optimization of composite steel box girder in flexure. Pract Period Struct Des Constr 12(3):146–152

Otsuki Y, Li D, Dey SS, Kurata M, Wang Y (2021) Finite element model updating of an 18-story structure using branch-and-bound algorithm with epsilon-constraint. J Civ Struct Health Monit 11(3):575–592

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

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

Penadés-Plà V, García-Segura T, Yepes V (2019) Accelerated optimization method for low-embodied energy concrete box-girder bridge design. Eng Struct 179:556–565

Rao R (2016) Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int J Ind Eng Comput 7(1):19–34

Rempling R, Mathern A, Tarazona Ramos D, Luis Fernández S (2019) Automatic structural design by a set-based parametric design method. Autom Constr 108:102936

Richardson A (2010). In: Corder GW, Foreman DI (eds) Nonparametric statistics for non-statisticians: a step-by-step approach. Wiley, Hoboken

Sarma KC, Adeli H (1998) Cost optimization of concrete structures. J Struct Eng 124(5):570–578

Van Rossum G, Drake FL (2009) Python 3 Reference Manual. CreateSpace, Scotts Valley

Vayas I, Iliopoulos A (2017) Design of steel–concrete composite bridges to Eurocodes. CRC Press, Boca Raton

Venkata Rao R (2016) Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int J Ind Eng Comput 7:19–34

Yepes V, Alcala J, Perea C, González-Vidosa F (2008) A parametric study of optimum earth-retaining walls by simulated annealing. Eng Struct 30(3):821–830

Yepes V, Gonzalez-Vidosa F, Alcala J, Villalba P (2012) CO_2-optimization design of reinforced concrete retaining walls based on a VNS-threshold acceptance strategy. J Comput Civ Eng 26(3):378–386

Yepes V, Martí JV, García-Segura T (2015) Cost and CO^2 emission optimization of precast-prestressed concrete U-beam road bridges by a hybrid glowworm swarm algorithm. Autom Constr 49:123–134