An outer approximation bi-level framework for mixed categorical structural optimization problems

Pierre-Jean Barjhoux, Youssef Diouane, Stéphane Grihon, Joseph Morlier

An outer approximation bi-level framework for mixed categorical structural optimization problems

Pierre-Jean Barjhoux[1] · Youssef Diouane[2] · Stéphane Grihon[3] · Joseph Morlier[4]

Abstract

In this paper, mixed categorical structural optimization problems are investigated. The aim is to minimize the weight of a truss structure with respect to cross-section areas, materials, and cross-section type. The proposed methodology consists of using a bi-level decomposition involving two problems: master and slave. The master problem is formulated as a mixed-integer linear problem where the linear constraints are incrementally augmented using outer approximations of the slave problem solution. The slave problem addresses the continuous variables of the optimization problem. The proposed methodology is tested on three different structural optimization test cases with increasing complexity. The comparison to state-of-the-art algorithms emphasizes the efficiency of the proposed methodology in terms of the optimum quality, computation cost, as well as its scalability with respect to the problem dimension. A challenging 120-bar dome truss optimization problem with 90 categorical choices per bar is also tested. The obtained results showed that our method is able to solve efficiently large-scale mixed categorical structural optimization problems.

1 Introduction

In this paper, we investigate a class of structural optimization problems with a fixed topology (for the structure) but involving mixed categorical design variables (Barjhoux et al. 2020; Grihon 2012, 2018). Typically, in the context of structural optimization, the choices of material properties or cross-section types are depicted by categorical variables. The thicknesses or cross-section areas belong to the set of continuous design variables. Many optimization algorithms are designed to solve such problems. For example, metaphor-based metaheuristics and swarm intelligence algorithms (Liao et al. 2014; Goldberg 1989; Nouaouria and Boukadoum 2011) natively handle discrete variables. However, these methods are not suitable for solving large-scale optimization problems (Sigmund 2011; Stolpe 2011).

Various surrogate-based optimization strategies have been extended to solve mixed categorical structural optimization problems (Filomeno Coelho 2014; Müller et al. 2013; Herrera et al. 2014; Roy et al. 2017, 2019; Garrido-Merchán and Hernández-Lobato 2018; Pelamatti et al. 2019; Saves et al. 2022; Rufato et al. 2022). One of the main challenges of such approaches is related to their inefficiency when handling large dimension categorical design space. Other existing works propose new formulations of the original optimization problem by reducing the dimension of a structural optimization problem or by using continuous relaxation of the design variables (Gao et al. 2018; Stegmann and Lund 2005; Krogh et al. 2017). For all these existing approaches, there is no guarantee that the optimization will retrieve the best categorical choices.

By converting categorical variables to integers in the structural optimization problems, classical mixed-integer programming approaches can also be used to solve such problems. In this context, many existing approaches are based on branch-and-bound. For instance, Achtziger and Stolpe (2007), Stolpe (2007) proposed to rewrite the relaxed problem within their branch-and-bound algorithms as a convex problem which helps to reach easily the global minimum

Responsible Editor: Mathias Stolpe

✉ Youssef Diouane
youssef.diouane@polymtl.ca

[1] Airbus SAS, Institute of Technology IRT Saint Exupéry, Institut Clément Ader (ICA), Toulouse, France

[2] Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Montréal, QC, Canada

[3] Airbus SAS, Toulouse, France

[4] Institut Clément Ader (ICA),CNRS-ISAE SUPAERO-INSA-Mines Albi-UPS, Université de Toulouse, Toulouse, France

of the relaxed problem. Other existing approaches are based on decomposition strategies to transform the original problem into a sequence of easy-to-solve subproblems, e.g., Bender decomposition (Benders 1962; Geoffrion 1972) and outer approximation (Duran and Grossmann 1986; Fletcher and Leyffer 1994; Hijazi et al. 2014). Several variants based on outer approximation algorithm have been implemented (Stolpe 2015) and successfully applied on mixed-integer structural optimization problems. In the context of continuous structural optimization, decomposition schemes have been widely used, e.g., StiffOpt (Samuelides et al. 2009), Quasi Separable Decomposition (Haftka et al. 2006; Schutte et al. 2004).

In an industrial context, practical methodologies have emerged to tackle the curse of dimensionality when dealing with categorical variables in large-scale structural optimization. For instance, Grihon (2018) uses a bi-step strategy involving massively parallel element-wise optimizations. In fact, by assuming that the main optimization problem is separable with respect to design variables, the approach reduces to a set of optimization problems at the element (subsystem) level, with fixed internal loads. This, in particular, simplifies the impact of each categorical choice on the overall optimal internal loads distribution. This approach is industrially recognized at Airbus. The approach has a computational complexity that depends linearly with respect to the number of structural elements and categorical values. Although this existing approach is scalable, it cannot handle system-level behavior (optimum internal load distribution) nor system-level constraints (e.g., flutter, modal or displacement constraints). The absence of such constraints in the problem formulation is not representative of aircraft structure design problems, in a multidisciplinary design optimization for instance.

The proposed methodology in this paper relies on previous works in Barjhoux et al. (2018a, b, 2020) where a bi-level methodology was initially proposed. The framework is based on master and slave problems. In Barjhoux et al. (2018b), it has been shown that the hybrid branch-and-bound based approach (for the master problem) can be costly in terms of the number of calls to the finite elements model. The exploration cost was shown to grow exponentially with the number of elements and categorical choices, preventing from using this algorithm to solve large-scale problem instances. The computational cost of the Bi-level methodology as proposed in Barjhoux et al. (2020) is quasi-linear with respect to the number of structural elements. In particular it permits to solve medium to large-scale structural optimization problems (up to two hundred mixed variables). The latter approach offered an interesting compromise between the quality of the solution and the computational cost, provided the simplicity of the methodology.

In this study, we propose a new Bi-level methodology that leverages the use of linearizations of the subproblems in an Outer Approximation (OA) framework (Fletcher and Leyffer 1994). This leads to a more significant computational cost reduction. In our proposed formulation, the mixed categorical-continuous problem is first formulated as a mixed-integer-continuous problem with relaxable integer design variables. The continuous design variables are handled by the slave problem, while the integer variables are governed by the master problem. The latter consists of solving a mixed-integer linear problem built iteratively by concatenating linear approximations of the slave problem solutions. This approach is different from the OA framework originally presented in Fletcher and Leyffer (1994) in the sense that only a linear approximation of the slave problem is used to define the master problem approximation. The derivatives used to build the linear approximations of our subproblems are constructed using post-optimal sensitivities (Fiacco 1976). Under a convexity assumption, we will show that the proposed approach converges to the optimal solution. Although the convexity assumptions cannot be verified for structural optimization problems in general, the obtained numerical results show that our proposed method is performing very well compared to state-of-the-art methods.

This paper is organized as follows. In Sect. 2, the formulation of the mixed categorical-continuous optimization problem is presented. In Sect. 3, the proposed methodology is presented. The performance and the scalability of our approach are compared with state-of-the-art algorithms in Sect. 4. The section is concluded with the obtained results on a structural problem of 120 structural elements with 90 categorical choices for each of element. Conclusion and perspectives are drawn in Sect. 5.

2 Problem statement

2.1 Design space

In this work, our goal is to minimize the weight of a structure, at fixed topology, by exploring the internal geometry as well as material description of all the structural elements of the problem. Two kinds of design variables are thus involved when handling these kinds of problems.

Firstly, categorical choices are involved as design variables during the optimization process. Indeed, in this work, the possible choices of material and member profiles (e.g., “I”, “C”, “T”) for each element will be regarded as a part of the design variables and that have to be explored. The categorical choices will be described by a matrix $\mathbf{B}$ that has $n \times p$ binary coefficients, where p is the possible number of choices per structural element n. In this paper, we will often refer to the set possible choices by the set of catalogs. We specify here that the p available catalogs are the same for all the structural elements. In this context, during the optimization, we assign to each element a choice of material and cross-section type, all described by one categorical design variable. Let $\mathcal{C}^{n\times p}$ be the enumerated set that contains the p^n choices of materials and member profiles for each element of the structure, so that

$$\mathcal{C}^{n\times p} := \left\{ X \in \mathbb{R}^{n\times p} \ : \ X_{ij} \in \{0,1\} \ \text{and} \ \sum_{j=1}^{p} X_{ij} = 1 \right\},$$

where X_{ij} represents the element at the ith row and jth column of the matrix X. X_{ij} is a binary choice variable among the existing p choices per structural element, with $X_{ij} = 1$ if for the ith element the jth categorical option is chosen and $X_{ij} = 0$ otherwise. We note that, each row of the matrix $X \in \mathcal{C}^{n\times p}$ describes the catalog choices composition of a given element. For example, if $\mathbf{B} \in \mathcal{C}^{10\times 4}$ and $B_{42} = 1$, then the categorical choice associated to the 4th structural element is the choice 2, corresponding to a given combination of profile and material among the 4 available choices.

In a second time, we treat the member profiles areas as continuous design variables. Formally, the areas can be represented as a vector $\mathbf{a} \in \mathbb{R}^n$ where the number of components n corresponds to the number of structural elements. For a given choice of member profile, the areas scale the internal shape of the structural elements (Barjhoux et al. 2018a, b). The description of the internal cross-section parameters (with respect to areas $\mathbf{a}_i$) is given by

$$\mathbf{x}^{(i)}(\mathbf{a}_i) := \sqrt{\frac{\mathbf{a}_i}{\mathbf{a}_0(\mathbf{B}_{i,:})}}\mathbf{x}_0(\mathbf{B}_{i,:}), \tag{1}$$

where $\mathbf{B}_{i,:}$ represents the ith row of the matrix $\mathbf{B}$ and $\mathbf{x}_0(\mathbf{B}_{i,:})$ is the reference detailed geometry of the profile driven by the choice $\mathbf{B}_{i,:}$. This way, the parameters $\mathbf{x}^{(i)}$ are estimated as latent variables that depend on the areas $\mathbf{a}_i$. This description of the internal member profile geometry is inspired from existing approaches like, for example, the PRESTO methodology in Grihon (2012), Gao et al. (2018). Figure 1 shows how internal parameters (and so the area moments of inertia) can be scaled using the area of the cross-section. The proposed scaling allows to handle any profile type as far as we can add new member profiles depending on the design space we want to explore.

2.2 Objective and constraints functions

The objective function and the constraints as presented in Barjhoux et al. (2020) are reformulated in this article in particular for the sake of clarity. First, the categorical variable is coded as a binary variable. Second, we use continuous definitions of objective and constraints functions. The binary variables are introduced as continuous weighting factors in these functions. This means that, each of the objective and constraints functions can be evaluated at intermediate values of $\mathbf{B}$, even if the outputs have no physical meaning.

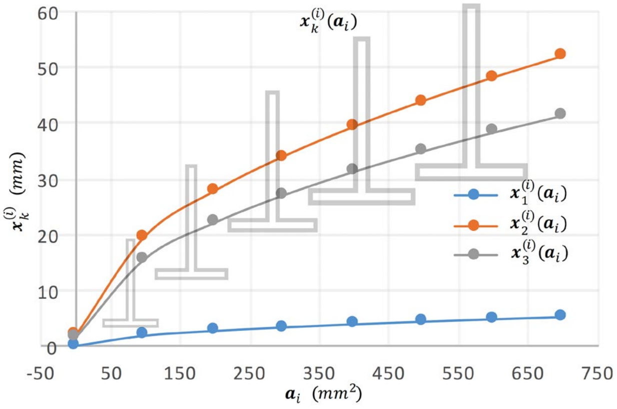


Fig. 1 Scaling of a bar section. Example with "T"-profile

First, we need to define the space on which the functions are defined. Let $\widetilde{\mathcal{C}}^{n\times p}$ be the set of matrices $\mathbf{B}$ of real coefficients such as

$$\widetilde{\mathcal{C}}^{n\times p} := \left\{ X \in \mathbb{R}^{n\times p} \ : \ X_{ij} \in [0,1] \ \text{and} \ \sum_{j=1}^{p} X_{ij} = 1 \right\}.$$

In other terms, $\widetilde{\mathcal{C}}^{n\times p}$ is the continuous relaxation of $\mathcal{C}^{n\times p}$ on $[0, 1]$. Of course, there is no underlying physical meaning when $\mathbf{B}$ takes intermediate (real) values in $[0, 1]$. The values of $\mathbf{B}$ in $\widetilde{\mathcal{C}}^{n\times p}$ will serve as weighting factors in the objective and constraints functions.

In this problem, the objective is the weight function, given by

$$\begin{aligned} \widetilde{w} \, : \, & \mathbb{R}^n \times \widetilde{\mathcal{C}}^{n\times p} \to \mathbb{R} \\ & (\mathbf{a}, \mathbf{B}) \mapsto \sum_{i=1}^{n}\sum_{c=1}^{p} \rho(c)\ell_i \mathbf{B}_{ic}\mathbf{a}_i, \end{aligned} \tag{2}$$

with $\rho(c)$ refers the density of the material that corresponds to the choice c. It is computed as the sum of the p available densities weighted by the continuous choices $\mathbf{B}$. The constant ℓ_i denotes the length of element i.

The constraints $\boldsymbol{\delta}$ on displacements $\mathbf{u}$ ensure that on d given nodes of the truss, the displacements will not exceed predefined upper bounds $\bar{\mathbf{u}} \in \mathbb{R}^d$. With $\boldsymbol{P}$ a projector that select the elements on which the displacement constraint will apply, the definition of $\boldsymbol{\delta}$ function is given as follows :

$$\begin{aligned}\widetilde{\boldsymbol{\delta}} : \mathbb{R}^n \times \widetilde{\mathcal{C}}^{n\times p} &\to \mathbb{R}^d \\ (\mathbf{a}, \mathbf{B}) &\mapsto \boldsymbol{P}\boldsymbol{u}(\mathbf{a}, \mathbf{B}) - \bar{\mathbf{u}}.\end{aligned} \tag{3}$$

The stress constraints $\widetilde{\mathbf{s}}_{ij}$ are defined such as

$$\begin{aligned}\widetilde{\mathbf{s}}_{ij} : \mathbb{R}^n \times \widetilde{\mathcal{C}}^{n\times p} &\to \mathbb{R} \\ (\mathbf{a}, \mathbf{B}) &\mapsto \sum_{c=1}^{p} \mathbf{B}_{ic}\mathbf{s}_{ij}(\mathbf{a}_i, c, \boldsymbol{\Phi}_i(\mathbf{a}, \mathbf{B}))\end{aligned} \tag{4}$$

where $\mathbf{s}_{ij}$ is given by

$$\boldsymbol{s} : \mathbb{R}^n \times \widetilde{\mathcal{C}}^{n\times p} \to \mathbb{R}^{n\times m}$$

and is of the form, for every choice c among the categorical set $\{1, \dots, p\}$

$$\begin{pmatrix} s_{11}(\mathbf{a}_1, c, \boldsymbol{\Phi}_1(\mathbf{a}, \mathbf{B})) & \dots & s_{1m}(\mathbf{a}_1, c, \boldsymbol{\Phi}_1(\mathbf{a}, \mathbf{B})) \\ s_{21}(\mathbf{a}_2, c, \boldsymbol{\Phi}_2(\mathbf{a}, \mathbf{B})) & \dots & s_{2m}(\mathbf{a}_2, c, \boldsymbol{\Phi}_2(\mathbf{a}, \mathbf{B})) \\ \vdots & \ddots & \vdots \\ s_{n1}(\mathbf{a}_n, c, \boldsymbol{\Phi}_n(\mathbf{a}, \mathbf{B})) & \dots & s_{nm}(\mathbf{a}_n, c, \boldsymbol{\Phi}_n(\mathbf{a}, \mathbf{B})) \end{pmatrix}.$$

The element constraint $\boldsymbol{s}_{ij}$ is defined as the difference between the structural member stress constraints value and a limit stress. In particular, if the members constraint stress exceeds the limit stress, then $\boldsymbol{s}_{ij}$ will take negative values and the constraints will be violated. Practical expressions of these optimization constraints are provided in the numerical section [see (11, 12, 13 and 14)].

In the context of this work, there is no change in the topology of the structure. Internal forces $\boldsymbol{\Phi}$ and displacements $\mathbf{u}$ will be computed using the direct stiffness method, introduced in Turner (1959), Turner et al. (1964). Structural elements are considered as truss elements with pin-jointed connections. This means that the bars will only carry axial forces. The cross-section shapes will be only involved through the Euler and local constraints definition (see Sect. 4). The stiffness matrix of the structure is not impacted by the cross-section shapes. At each node, displacements are allowed along the global axes. Each element i is defined by the elementary stiffness matrix $\boldsymbol{K}_i^e(\mathbf{a}_i, \mathbf{B}_{i,:}) \in \mathbb{R}^{q,q}$, with q the number of free nodes multiplied by the number of physical space dimensions. The global stiffness of the whole truss is given by the matrix $\boldsymbol{K}(\mathbf{a}, \mathbf{B}) \in \mathbb{R}^{q,q}$ in global coordinates. Such matrix can be computed as the sum of each element stiffness matrix expressed after its transformation with the i^{th} element rotation matrix $\mathbf{T}_i$, i.e., (Turner 1959; Turner et al. 1964):

$$\boldsymbol{K}(\mathbf{a}, \mathbf{B}) := \sum_{i=1}^{n} [\mathbf{T}_i^\top \boldsymbol{K}_i^e(\mathbf{a}_i, \mathbf{B}_{i,:})\mathbf{T}_i].$$

Given a vector $\boldsymbol{f} \in \mathbb{R}^q$ of external loads applied on each of the free nodes in the global coordinates, the vector of displacements $\boldsymbol{u} \in \mathbb{R}^q$ can be obtained by solving the following equation:

$$\boldsymbol{K}(\mathbf{a}, \mathbf{B})\boldsymbol{u}(\mathbf{a}, \mathbf{B}) = \boldsymbol{f}. \tag{5}$$

The vector of internal forces $\boldsymbol{\Phi} \in \mathbb{R}^n$ is then given by $\forall i \in \{1, \dots, n\}$,

$$\boldsymbol{\Phi}_i(\mathbf{a}, \mathbf{B}) := \boldsymbol{K}_i^e(\mathbf{a}_i, \mathbf{B}_{i,:})\mathbf{T}_i\boldsymbol{u}_i(\mathbf{a}, \mathbf{B}),$$

where $\boldsymbol{\Phi}_i$ is the axial force through element i and $\boldsymbol{u}_i$ its displacement vector.

2.3 Optimization problem

As formulated in Barjhoux et al. (2020), the problem involves categorical non-ordered and non-relaxable design variables. The formulation prevents from using algorithms that exploit the gradient of the functions with respect to all the design variables. This is why, objective and constraints have been introduced as continuous functions in Sect. 2.2. The categorical optimization problem is a mixed categorical-continuous optimization problem, formulated as a mixed-integer non-linear programming (MINLP) problem. The optimization problem consists of a structural weight minimization with respect to stress and displacements constraints :

$$\begin{aligned} \underset{(\mathbf{a},\mathbf{B})\in\mathbb{R}^n\times\mathcal{C}^{n\times p}}{\text{minimize}} \quad & \widetilde{w}(\mathbf{a}, \mathbf{B}) \\ \text{subject to} \quad & \widetilde{\mathbf{s}}(\mathbf{a}, \mathbf{B}) \le \mathbf{0}_{n,m} \\ & \widetilde{\boldsymbol{\delta}}(\mathbf{a}, \mathbf{B}) \le \mathbf{0}_d \\ & \mathbf{a} \le \mathbf{a} \le \bar{\mathbf{a}} \end{aligned} \tag{P}$$

where $\mathbf{a} \in \mathbb{R}^n$ and $\bar{\mathbf{a}} \in \mathbb{R}^n$ are the lower and upper bounds on areas, respectively. It is worth to note that the design space is $\mathcal{C}^{n\times p}$, such that the solution $\mathbf{B}$ is binary. The methodology presented in the Sect. 3 will take advantage of the continuous property of the objective and constraints functions. In the next section, we will describe our proposed methodology to solve (P).

3 Methodology

In what comes next, for the sake of clarity, we will consider $\mathbf{B}$ and $\widetilde{\mathbf{s}}$ as vectors instead of matrices. Namely, $\mathcal{C}^{n\times p} \subset \mathbb{R}^{np}$ and $\mathbb{R}^{n\times m} \sim \mathbb{R}^{nm}$. The identification between the vector and matrix space can be trivially obtained, for instance, for any $\mathbf{A} \in \mathbb{R}^{n\times p}$ one can use

$$[\mathbf{A}_{11}, \dots, \mathbf{A}_{1p}, \mathbf{A}_{21}, \dots, \mathbf{A}_{2p}, \dots, \mathbf{A}_{n1}, \dots, \mathbf{A}_{np}]^\top$$

to consider it as an element of $\mathbb{R}^{np}$ as well. We note that in our optimization setting, the topology will be kept unchanged. Handling the change in the topology of the structure will not be covered by our approach. The reason behind this restriction will be explained in Sect. 3.2.1.

3.1 A bi-level framework

The proposed bi-level decomposition of the problem (P) is presented in this section.

For a given $\mathbf{B} \in \mathcal{C}^{n\times p}$, let $\Omega(\mathbf{B}) \subset \mathbb{R}^n$ be the set of feasible constraints of the problem (P) given by

$$\Omega(\mathbf{B}) := \{\mathbf{a} \le \mathbf{a} \le \bar{\mathbf{a}} \ : \ \widetilde{s}(\mathbf{a},\mathbf{B}) \le \mathbf{0}_{mn} \text{ and } \widetilde{\delta}(\mathbf{a},\mathbf{B}) \le \mathbf{0}_d\}.$$

An efficient way to solve pure continuous optimization problems is by taking advantage of gradient-based algorithms. In the problem introduced in Section 2, it can be seen that by fixing (temporarily) design variables $\mathbf{B}$ in (P) at integer values, the optimization problem becomes a continuous one parameterized with $\mathbf{B}$, and where integrity constraints on $\mathbf{B}$ can be removed. This means that at a given $\mathbf{B}$, the weight $\widetilde{w}$ can be minimized with respect to the remaining continuous design variables that are the areas $\mathbf{a} \in \Omega(\mathbf{B})$. This leads to the following slave problem (sP($\mathbf{B}$)), that reduces to a structural sizing optimization problem:

$$\Psi(\mathbf{B}) := \underset{\mathbf{a}\in\Omega(\mathbf{B})}{\text{minimize}}\ \widetilde{w}(\mathbf{a},\mathbf{B}). \qquad (sP(\mathbf{B}))$$

The structure of the problem is such that this remaining optimization problem becomes more tractable. In fact, the decomposition leverages the use of the gradients (with respect to $\mathbf{a}$) of the objective and constraints to solve the problem (sP($\mathbf{B}$)). This is the main motivation in handling the continuous variables separately from the integer ones. In this approach, the integer (binary) variables will be handled by a master problem (mP) of the form

$$\underset{\mathbf{B}\in\mathcal{C}^{n\times p}}{\text{minimize}} \quad \Psi(\mathbf{B}), \tag{mP}$$

with $\Psi(\mathbf{B})$ is the result of the slave Problem (sP($\mathbf{B}$)). The slave problem (sP($\mathbf{B}$)) takes these complicating variables $\mathbf{B}$ as parameters, while optimizing with respect to continuous design variables. This means that during the slave optimization, the choices of materials and cross-section types for all elements remain fixed. This slave problem will be solved using a gradient-based method. The obtained solution can be seen as a function $\Psi(\mathbf{B})$ which is parameterized by the categorical choices through the continuous coding $\mathbf{B}$. Namely, $\Psi(\mathbf{B})$ corresponds to the optimal weight of the slave problem knowing the variables $\mathbf{B}$. This function is then taken as the objective of the master optimization problem (sP($\mathbf{B}$)). Although the slave problem can be easy to handle using gradient-based algorithms, the difficult part remains in the master problem. In fact, the problem (mP) is still a large-scale pure integer non-linear optimization problem, that usual combinatorial optimization solvers fail to solve efficiently. However, unlike the problem presented in Barjhoux et al. (2020), the integer variable $\mathbf{B}$ is relaxable and the functions are defined at intermediate non 0–1 values of $\mathbf{B}$. Moreover, all the functions of the optimization problem are continuously differentiable. This is a basic requirement to compute the sensitivity of the slave problem solution parameterized in $\mathbf{B}$.

3.2 On the minimization of Ψ

In this paper, we suggest to solve the master problem (mP) by means of outer approximation (OA) cuts that are built using the gradient information on Ψ. We propose to consider at the master level the minimization of an approximated problem $\mathcal{P}$ instead of (mP), so that the computational complexity of the master problem can be significantly reduced. For that, the following iterative scheme is implemented. Given an iteration k, the master problem (mP) of the bi-level formulation is reduced to a problem $\mathcal{P}^{(k)}$ easier to solve.

The slave optimization problem is defined by fixing the binary variables $\mathbf{B}^{(k)}$ in the problem (mP). Due to the set of constraints, the problem (mP) can be seen as a full integer optimization problem. The slave problem reduces to an evaluation of the objective $\Psi(\mathbf{B}^{(k)})$ which represents the optimal weight solution of (sP($\mathbf{B}$)) given for a fixed categorical choice $\mathbf{B}^{(k)}$. Assuming that there is at least one feasible solution depending on the fixed point $\mathbf{B}^{(k)}$, the optimal objective value of the slave problem is an upper bound of the solution to (P).

A definition of the master problem is given as follows. In fact, under the assumption that the function Ψ is convex, using (Fletcher and Leyffer 1994, Theorem 1), one deduces that solving the problem (mP) is equivalent to solving the following mixed-integer linear program (MILP) given by

$$\begin{aligned} &\underset{\mathbf{B}\in\mathcal{C}^{n\times p},\eta\in\mathbb{R}}{\text{minimize}} && \eta \\ &\text{s. t.} && \eta \ge \Psi(\widetilde{\mathbf{B}}) + \left.\frac{d\Psi}{d\mathbf{B}}\right|_{\widetilde{\mathbf{B}}}^{\top} (\mathbf{B} - \widetilde{\mathbf{B}}),\ \forall \widetilde{\mathbf{B}} \in \mathcal{C}^{n\times p} \end{aligned} \tag{6}$$

We note that in the problem (6), the function Ψ is replaced by an hyperplane that is also its linear support at $\widetilde{\mathbf{B}} \in \mathcal{C}^{n\times p}$.

Solving the MILP problem (6) directly may be out of reach as it would require p^n evaluations of Ψ corresponding to all integer vectors $\widetilde{\mathbf{B}} \in \mathcal{C}^{n\times p}$. In our case, it would require evaluations of the sizing problem (sP($\mathbf{B}$)) taken at every p^n combinations of materials and cross-section types available in $\{1, \dots, p\}^n$. The number of constraints p^n related to the problem (6) can be also extremely large for reasonable values of p and n. For this reason, instead of considering the problem (6), the OA algorithm involves a sequence of less expensive relaxed variant of the MILP problem (6), i.e., for a given iteration (k) one solves

$$\begin{aligned} \underset{\mathbf{B}\in\mathcal{C}^{n\times p},\eta\in\mathbb{R}}{\text{minimize}} \quad & \eta \\ \text{s. t.} \quad & \eta \geq \Psi(\widetilde{\mathbf{B}}) + \left.\frac{d\Psi}{d\mathbf{B}}\right|_{\widetilde{\mathbf{B}}}^{\top} (\mathbf{B}-\widetilde{\mathbf{B}}),\ \forall \widetilde{\mathbf{B}} \in K^{(k)} \end{aligned} \tag{7}$$

with $K^{(k)}$ a set of k elements in $\mathcal{C}^{n\times p}$, such that $K^{(k)} \subset \mathcal{C}^{n\times p}$. We note that, under the convexity assumption of Ψ, the problem (7) yields a lower bound to the solution of the Problem (6). At each iteration of the OA algorithm, the problem (7) can be geometrically interpreted as an exploration of the effects of the outer approximations (i.e., the linear supports) on the objective Ψ. The set $K^{(k)}$ will be updated recursively as follows $K^{(k)} \leftarrow K^{(k-1)} \bigcup \{\mathbf{B}^{(k)}\}$ where $\mathbf{B}^{(k)}$ is the solution of the MILP problem (7) at a fixed iteration (k). Hence, the MILP problem (7) for the iteration (k) is obtained just by adding to the problem (7) (related with the iteration $(k-1)$) the linear constraint

$$\eta \geq \Psi(\mathbf{B}^{(k)}) + \left.\frac{d\Psi}{d\mathbf{B}}\right|_{\mathbf{B}^{(k)}}^{\top} (\mathbf{B}-\mathbf{B}^{(k)}). \tag{8}$$

A key ingredient for setting the latter constraint is the estimation of the derivative $\left.\frac{d\Psi}{d\mathbf{B}}\right|_{\mathbf{B}^{(k)}}$. Under reasonable assumptions, the next subsection details how post-optimal sensitivities can be useful on estimating the gradient of Ψ at $\mathbf{B}^{(k)}$.

3.2.1 Computing the gradient of Ψ at $\mathbf{B}^{(k)}$: $\left.\frac{d\Psi}{d\mathbf{B}}\right|_{\mathbf{B}^{(k)}}$

The estimation of the gradient of Ψ with respect to the parameters $\mathbf{B}$ is a key ingredient in setting the constraint (8). On that way, the estimated gradient will provide information on the behavior of the optimal weight, solution of (sP($\mathbf{B}$)), subject to a small perturbation of $\mathbf{B}^{(k)}$. In this case, the gradient of Ψ is known as *post-optimal sensitivity* (Fiacco 1976). It can be noted that such perturbation has no physical meaning since $\mathbf{B}$ describes categorical choices in a continuous manner. The efficient computation of the gradient of Ψ at a given $\mathbf{B}^{(k)}$ will be a key feature of our proposed methodology. Indeed, if the gradient is estimated by finite differences, its computational cost is growing proportionally to np (i.e., the number of parameters $\mathbf{B}$). This would typically require to solve $n(p-1)$ optimization problems instances. Hence, for large-scale optimization problems, using finite differences may be out of reach. In the context of large-scale optimization problems, estimating the gradient using post-optimal sensitivity analysis can be very helpful. In Appendix 1, we give the details of deriving the derivatives $\left.\frac{d\Psi}{d\mathbf{B}}\right|_{\mathbf{B}^{(k)}}$.

The post-optimal sensitivity analysis is derived using the Karush–Kuhn–Tucker (KKT) conditions where, particularly, we require the constraint qualification (i.e., the gradients of the constraint functions of active constraints being linearly independent). Such assumption is reasonable in the context of structural optimization problems with a fixed topology. But, it might not be guaranteed for problems where a change in topology is allowed, mainly due to the presence of the so-called vanishing constraints. For this reason, in this work, we consider that the topology of the structure is unchanged during the optimization process.

3.2.2 An outer approximation bi-level framework

The proposed algorithm consists of solving an alternating sequence of slave and master problems, as defined previously. The post-optimal sensitivities of the slave problem (sizing) are involved in the definition of the master problem. Let (k) be the current outer iteration of the algorithm. The algorithm workflow is illustrated in Fig. 2.

First, the slave problem, reduced to an evaluation of Ψ, is solved at $\mathbf{B}^{(k)}$. This means that as a first step, the slave continuous optimization problem (sP($\mathbf{B}$)) aiming at minimizing the weight while satisfying stress and displacements constraints is solved. The problem (sP($\mathbf{B}^{(k)}$)) is solved and yields a solution $\mathbf{a}^{(k)}$ such that

$$\mathbf{a}^{(k)} := \underset{\mathbf{a}\in\Omega(\mathbf{B}^{(k)})}{\operatorname{argmin}}\ \widetilde{w}(\mathbf{a},\mathbf{B}^{(k)}). \tag{9}$$

An upper bound $U^{(k)}$ to the solution of (P) is defined by

$$U^{(k)} := \widetilde{w}(\mathbf{a}^{(k)},\mathbf{B}^{(k)}) = \Psi(\mathbf{B}^{(k)}),$$

The best current solution of the original problem (P) is thus given by the best upper bound returned during the (k) outer iterations :

$$U^{(k)}_{\text{min}} := \min\{U^{(1)},\dots,U^{(k)}\}. \tag{10}$$

Second, the relaxed MILP problem (7) can be set up. Its definition relies on the linearizations of Ψ taken at the solutions yielded during the (k) previous iterations. While the linearizations from the previous iterates $(l) < (k)$ remain unchanged, the linearization of Ψ at the current iteration (k) has to be computed. More precisely, the gradient of Ψ taken at $\mathbf{B}^{(k)}$ has to be evaluated.

Once the linearization of Ψ has been computed, it is added as constraint in the problem (7). Furthermore, since in practice the problem (P) does not need to be solved exactly, it is sufficient to generate the new $(\mathbf{B}^{(k+1)})$ by adding a tolerance ϵ on the upper bound $U^{(k)}_{\text{min}}$ as an additional constraint to the MILP master problem. The resulting mixed-integer linear integer problem (MILP(k)) is thus given by

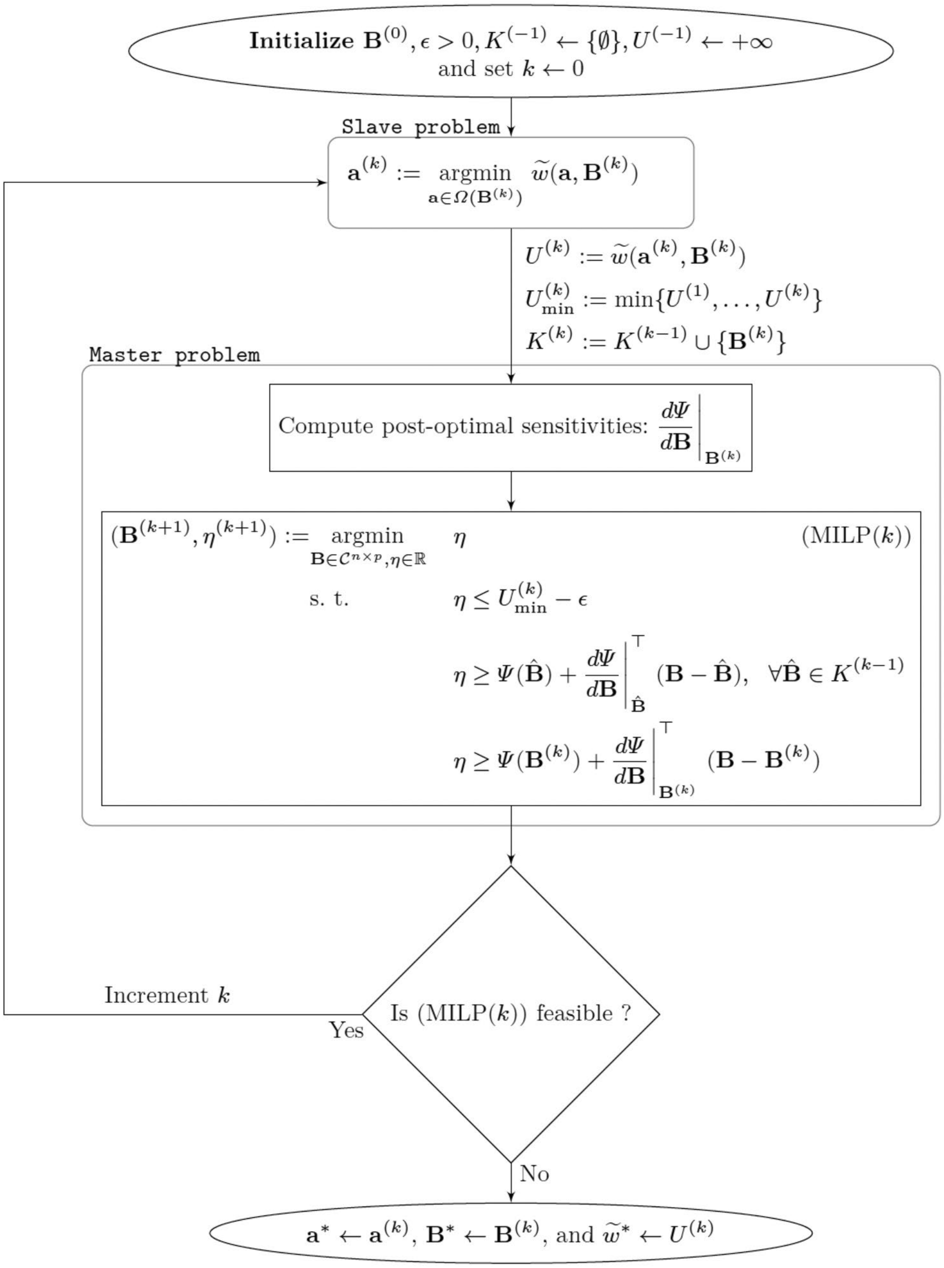

Initialize $\mathbf{B}^{(0)}, \epsilon > 0, K^{(-1)} \leftarrow \{\emptyset\}, U^{(-1)} \leftarrow +\infty$ and set $k \leftarrow 0$
Slave problem
$\mathbf{a}^{(k)} := \underset{\mathbf{a} \in \Omega(\mathbf{B}^{(k)})}{\operatorname{argmin}} \ \widetilde{w}(\mathbf{a}, \mathbf{B}^{(k)})$
$U^{(k)} := \widetilde{w}(\mathbf{a}^{(k)}, \mathbf{B}^{(k)})$
$U^{(k)}_{\min} := \min\{U^{(1)}, \dots, U^{(k)}\}$
$K^{(k)} := K^{(k-1)} \cup \{\mathbf{B}^{(k)}\}$
Master problem
Compute post-optimal sensitivities: $\left.\frac{d\Psi}{d\mathbf{B}}\right|_{\mathbf{B}^{(k)}}$
$(\mathbf{B}^{(k+1)}, \eta^{(k+1)}) := \underset{\mathbf{B} \in \mathcal{C}^{n \times p}, \eta \in \mathbb{R}}{\operatorname{argmin}} \ \eta$ (MILP(k))
s. t. $\eta \le U^{(k)}_{\min} - \epsilon$
$\eta \ge \Psi(\hat{\mathbf{B}}) + \left.\frac{d\Psi}{d\mathbf{B}}\right|_{\hat{\mathbf{B}}}^{\top} (\mathbf{B} - \hat{\mathbf{B}}), \ \forall \hat{\mathbf{B}} \in K^{(k-1)}$
$\eta \ge \Psi(\mathbf{B}^{(k)}) + \left.\frac{d\Psi}{d\mathbf{B}}\right|_{\mathbf{B}^{(k)}}^{\top} (\mathbf{B} - \mathbf{B}^{(k)})$
Increment k
Yes
Is (MILP(k)) feasible ?
No
$\mathbf{a}^* \leftarrow \mathbf{a}^{(k)}$, $\mathbf{B}^* \leftarrow \mathbf{B}^{(k)}$, and $\widetilde{w}^* \leftarrow U^{(k)}$

Fig. 2 Illustration of the proposed methodology (see Algorithm 1)

$$
\begin{aligned}
\underset{\mathbf{B} \in \mathcal{C}^{n \times p}, \eta \in \mathbb{R}}{\text{minimize}} \quad & \eta \\
\text{s. t.} \quad & \eta \le U^{(k)}_{\min} - \epsilon \\
& \eta \ge \Psi(\hat{\mathbf{B}}) + \left.\frac{d\Psi}{d\mathbf{B}}\right|_{\hat{\mathbf{B}}}^{\top} (\mathbf{B} - \hat{\mathbf{B}}), \ \forall \hat{\mathbf{B}} \in K^{(k-1)} \qquad \text{(MILP}(k)\text{)} \\
& \eta \ge \Psi(\mathbf{B}^{(k)}) + \left.\frac{d\Psi}{d\mathbf{B}}\right|_{\mathbf{B}^{(k)}}^{\top} (\mathbf{B} - \mathbf{B}^{(k)})
\end{aligned}
$$

with $K^{(k)}$ such that

$$K^{(k)} := K^{(k-1)} \cup \{\mathbf{B}^{(k)}\},$$

and $K^{(k-1)}$ the set of the $k-1$ previous $\mathbf{B}^{(k-1)}$. The problem (MILP(k)) is built iteration per iteration by adding, as constraints, linearization of the functions Ψ taken at the current solution ($\mathbf{B}^{(k)}$). The optimality of the algorithm relies on the convexity of Ψ, ensuring that the linearizations are underestimators of Ψ. Once built, the problem (MILP(k)) is solved and provides a lower bound of (P). Iteratively, the number of constraints within the problem (MILP(k)) is getting higher. This ensures a monotonic increase in the lower bound over the iterations (i.e., $\eta^{(k)} \le \eta^{(k+1)}$).

The algorithm will be declared as convergent when the feasible domain of the problem (MILP(k)) is getting empty. This particularly means that the numerical solutions $\mathbf{B}^{(k+1)}$ and $\eta^{(k+1)}$ of (MILP(k)) are getting unfeasible.

The bi-level procedure is detailed in Algorithm 1 (see Appendix 3), and illustrated as a workflow in Fig. 2.

The proposed algorithm leverages the use of post-optimal sensitivities by using them to define supporting hyperplanes of Ψ. These hyperplanes bound the convex hull of the slave problem. It is worth to note that the number of constraints involved in the master problem (mP) reduces to the k linearizations of Ψ from the (k) outer iterations, in addition to the (k) linear equality constraints involved in the definition of $C^{n \times p}$. Indeed, the OA algorithm is used to solve the master problem (mP), so that all the structural sizing constraints are handled by the slave problem (sP($\mathbf{B}$)). Hence, the MILP problem (MILP(k)) counts only $k+n$ linear constraints (including equality constraints from $C^{n \times p}$), compared to the $k \times (n \times m + d + n)$ (constraints $\mathbf{s}$, $\boldsymbol{\delta}$ and equality constraints from $C^{n \times p}$).

In industrial cases, where the number of structural elements n can reach 5000 elements (e.g., for a fuselage), and the number of constraints m per structural element is about 10. The problem (MILP(k)) can thus involve several millions of constraints. This could induce high computation time (Benson and Horst 1991; Stolpe and Sandal 2018) when solving the problem (MILP(k)).

Furthermore, two interesting properties about the OA algorithm efficiency have been introduced in Fletcher and Leyffer (1994). These properties also apply to the proposed methodology, that falls in the theoretical frame of the OA algorithm. The first property [see Fletcher and Leyffer (1994, Theorem 2)] states that if Ψ is convex, then Algorithm 1 converges, in a finite number of steps, at an optimal solution of (mP). If Ψ is linear, then Algorithm 1 trivially converges to the solution of (mP) in one iteration. We note that, although, the convexity assumption cannot be verified for general structural optimization, the proposed algorithm can be used independently of such assumption. In the next section, we will show the performance of the proposed method on practical structural optimization test cases.

4 Numerical results

In the present section, the proposed methodology will be applied to three different test cases: (i) the well-known 10-bar truss structure (Haftka and Gürdal 1992) adapted in Merval (2008), (ii) a 2D scalable cantilever structure (Shahabsafa et al. 2018), and (iii) a 120-bar dome truss structure (Saka and Ulker 1992). The third test case aims at demonstrating the efficiency of our methodology on complex structures with large number of categorical choices.

In this paper, we will consider problems with four different structural constraints per element (i.e., $m = 4$). In this case, one has two constraints in tension and compression, given by, respectively,

$$s_{i1}(\mathbf{a}_i, c, \boldsymbol{\Phi}_i(\mathbf{a}, \mathbf{B})) := \frac{\boldsymbol{\Phi}_i(\mathbf{a}, \mathbf{B})}{\mathbf{a}_i} - \sigma^t(c) \tag{11}$$

$$s_{i2}(\mathbf{a}_i, c, \boldsymbol{\Phi}_i(\mathbf{a}, \mathbf{B})) := -\frac{\boldsymbol{\Phi}_i(\mathbf{a}, \mathbf{B})}{\mathbf{a}_i} - \sigma^c(c) \tag{12}$$

with $\sigma^t(c) \in \mathbb{R}$ the stress limit in tension and $\sigma^c(c) \in \mathbb{R}$ the stress limit in compression, for a material choice $c \in \{1, \dots, p\}$. The two other constraints are the Euler and local buckling constraints, respectively, given by

$$s_{i3}(\mathbf{a}_i, c, \boldsymbol{\Phi}_i(\mathbf{a}, \mathbf{B})) := -\frac{\boldsymbol{\Phi}_i(\mathbf{a}, \mathbf{B})}{\mathbf{a}_i} - \frac{\pi^2 E(c) I(\mathbf{a}_i, c)}{\mathbf{a}_i \mathbf{L}_i^2} \tag{13}$$

$$s_{i4}(\mathbf{a}_i, c, \boldsymbol{\Phi}_i(\mathbf{a}, \mathbf{B})) := -\frac{\boldsymbol{\Phi}_i(\mathbf{a}, \mathbf{B})}{\mathbf{a}_i} - \frac{4\pi^2 E(c) \mathcal{K}^2(c)}{12(1 - \nu^2(c))}, \tag{14}$$

with $E(c)$, $I(\mathbf{a}_i, c)$, and $\nu(c)$ are, respectively, the Young's modulus, the quadratic moment of inertia, and the Poisson's ratio of the material for element i, given the choice $c \in \{1, \dots, p\}$. The ratio between cross-section internal sizes, depending on the stiffener profile, is given by $\mathcal{K}(c)$. ℓ_i denotes the length of element i. The local buckling constraint is introduced to prevent buckling of plate-like elements in the cross-section. It compares the stress value in the considered member with the elastic critical stress value for plate buckling. The derivatives of the weight function and the constraints (with respect to the areas $\mathbf{a}$) are obtained by applying the chain-rule theorem (see Appendix 2 for more details).

4.1 Implementation details

Algorithm 1 was implemented using the Generic Engine for MDO Scenarios (GEMSEO) (Gallard et al. 2018) in Python. The continuous non-linear optimization problems (i.e., evaluations of Ψ) are solved with the Method of Moving Asymptotes (MMA) (Svanberg 2002) as implemented in the non-linear-optimization (NLOPT) package (Johnson 2008). The MMA solver is capable of handling non-linear continuous optimization problems with inequality constraints. The mixed-integer linear optimization problems are solved with a branch and cut implemented as the coin-or branch and cut (coin-or/Cbc) in Forrest et al. (2018). All the default parameters are kept unchanged except the tolerance on the objective function which is set to 10^{-6} kg. In what comes next, the resulting implementation of Algorithm 1 will be called Bi-level OA.

Four solvers will be compared to Bi-level OA. The first solver is a baseline solver where we proceed with a full enumeration of continuous optimizations w.r.t. $\mathbf{a}$; see problem (sP($\mathbf{B}$)). At each iteration, all the available choice in $\mathcal{C}^{(n,p)}$ is tested. The resulting solution will be denoted as Baseline. The second solver in the comparison is a hybrid branch-and-bound (Barjhoux et al. 2018a) and will be noted h-B &B. This solver is based on the usual branch-and-bound algorithm where a specific bound method is adapted to tackle the mixed categorical problem. The procedure involves a continuous relaxation problem formulation to compute lower bounds. In the case where these problems are convex with respect to the sizing variables, the solvers Baseline and h-B &B are ensured to return the global optimum of the overall problem. The third solver used in the comparison (will be referred as Genetic) is a genetic algorithm (Deb and Goyal 1998) where we used the implementation given by Distributed Evolutionary Algorithms in Python (DEAP) toolbox (Fortin et al. 2012). Due to the stochastic nature of Genetic, we run ten times the optimizer and keep only the obtained results of the best run. The fourth solver will be noted Bi-level, it is the bi-level algorithm proposed by Barjhoux et al. (2020). The Bi-level solver is based on a similar bi-level paradigm as used in Bi-level OA. The main difference lies in the master problem formulation where, in the Bi-level, we minimize a first-order-like approximation. For all the solvers Bi-level, h-B &B and Bi-level OA, we use the MMA method from NLOPT to solve the slave problem.

The computation effort of a given solver will be measured by counting the number of structural analyses (noted #FEM) including those required by the computation of the gradients (when needed). The obtained optimal weights (by each solver) will be noted w^*, the latter will allow us to evaluate the quality of the optima found by each optimizer. We note also that in our setting, the Baseline solution can be seen as the best known categorical choices for the corresponding problem instance. We note that the quality of the Baseline solutions (being global optima or just local ones) is depending on the practical capabilities of the NLP solver to find a global optimal solution to the problem (sP($\mathbf{B}$)). For this reason, in cases where (sP($\mathbf{B}$)) is not convex, the Baseline solver may not guarantee to provide the global optimum. However, in all our numerical tests, we observe that the Baseline results (when available) give the best weight values. For that reason, we decided to evaluate how far the categorical choices are from the Baseline optimal choices. This information is displayed using the Hamming distance (noted d_h) where we will count the number of structural elements that have an optimal choice different to the Baseline categorical choices, i.e.,

$$d_h := \text{cardinal}\left\{ i \in \{1, \dots, n\} \mid [\mathbf{c}^*]_i \neq [\mathbf{c}_{\text{opt}}]_i \right\},$$

where $\mathbf{c}_{\text{opt}}$ is the the optimal Baseline catalogs and $\mathbf{c}^*$ is the optimal catalogs found by the other solvers.

4.2 An illustrative example: a 2-bar truss structure

To illustrate how the Bi-level method works, we will now describe in details its application to a 2-bar truss structure (see Fig. 3). For this problem, each element can take a value among three possible choices that, respectively, point to materials AL2139, TA6V and the same "I"-profile (see Fig. 4). The materials properties are listed in Appendix 4. For this simple case, one has $n = 2$, $p = 2$, and $\mathbf{B} \in C^{2\times 2}$. For all elements, the lower and upper bounds on areas are, respectively, fixed to 300 mm^2 and 2000 mm^2. A maximum downward displacement equal to $\bar{\mathbf{u}} = 7$ mm is allowed on the only free node of the structure:

$$\begin{aligned} \widetilde{\delta} : \mathbb{R}^2 \times \widetilde{\mathcal{C}}^{2,2} &\to \mathbb{R} \\ (\mathbf{a}, \mathbf{B}) &\mapsto \boldsymbol{Pu}(\mathbf{a}, \widetilde{\mathbf{E}}(\mathbf{B})) - \bar{\mathbf{u}}. \end{aligned}$$

The Bi-level OA method is initialized with

$$\mathbf{B}^{(0)} = \text{vec}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \text{and} \quad \epsilon = 1e{-}3\,kg.$$

The element 1 is thus made of TA6V, element 2 of AL2139.

- *First iteration* ($k = 0$)The first iteration $k = 0$ starts by solving the primal problem, that reduces to an evaluation of $\widetilde{\Psi}$ (by solving (sP($\mathbf{B}$))) at the current guess $\mathbf{B}^{(0)}$:

  $$U^{(0)} = 5.6\,kg, \ \mathbf{a}^{(0)} = [300.0, \ 942.8]\,\text{mm}^2$$

  Then, the gradient $\left.\frac{d\Psi}{d\mathbf{B}}\right|_{\mathbf{B}^{(0)}}$ is computed. To that purpose, the active constraints of the problem (sP($\mathbf{B}$)) at ($\mathbf{a}^{(0)}, \mathbf{B}^{(0)}$) are the lower bound constraint on the area of structural

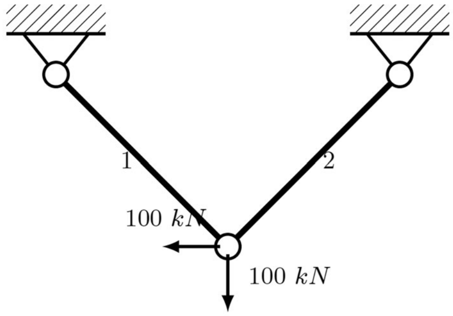
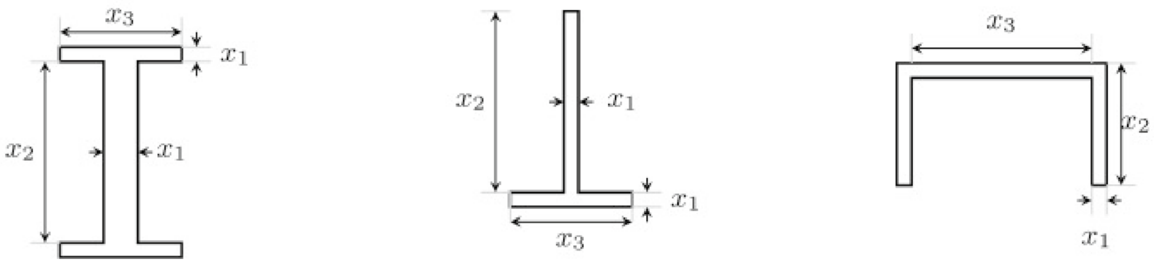


Fig. 3 A 2-bar truss structure where a downward and leftward load equal to 100 kN is applied on the free node



(a) Example of an “I”-profile described by 3 geometrical variables.

(b) Example of an “T”-profile, described by 3 geometrical variables.

(c) Example of an “C”-profile, described by 3 geometrical variables.

Fig. 4 Examples of commonly used member profiles in aircraft structural design. The internal geometrical variables are latent variables, scaled by the area of the cross-section

element 1, and the stress constraint in tension on the second structural element, i.e.,

$$\mathbf{a}_1 - \mathbf{a}_1^{(0)} = 0, \ \mathbf{s}_{11}(\mathbf{a}^{(0)}, \mathbf{B}^{(0)}) = 0.$$

Hence, the sets of active constraints indices are

$$\mathcal{A}_{\mathbf{a}}^{(0)} = \{1\}, \mathcal{A}_{\mathbf{s}}^{(0)} = \{5\}, \ \text{and} \ \mathcal{A}_{\delta}^{(0)} = \mathcal{A}_{\bar{\mathbf{a}}}^{(0)} = \{\emptyset\}.$$

The gradients of the weight and active constraints w.r.t. **a** are computed, respectively,

$$\left.\frac{\partial \widetilde{w}}{\partial \mathbf{a}}\right|_{\mathbf{z}^{(0)}} = \begin{pmatrix} 6.26e{-}3 \\ 3.96e{-}3 \end{pmatrix}^{\top}, \left.\frac{\partial \widetilde{\mathbf{s}}_{\mathcal{A}_{\mathbf{s}}^{(0)}}}{\partial \mathbf{a}}\right|_{\mathbf{z}^{(0)}} = \begin{pmatrix} 0. \\ -0.16 \end{pmatrix}^{\top},$$

where $\mathbf{z}^{(0)} = (\mathbf{a}^{(0)}, \mathbf{B}^{(0)})$ and $\mathbf{I}_{\mathcal{A}_{\mathbf{a}}^{(0)}} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}^{\top}$. One can see that the gradients of the active constraints are linearly independent. Equation (16) leads to the following linear system (with 2 equations and 2 unknown Lagrange multipliers):

$$\left.\frac{\partial \widetilde{w}}{\partial \mathbf{a}}\right|_{\mathbf{z}^{(0)}} + \left[\lambda_{\mathcal{A}_{\mathbf{s}}^{(0)}}^{(0)}\right] \left.\frac{\partial \widetilde{\mathbf{s}}_{\mathcal{A}_{\mathbf{s}}^{(0)}}}{\partial \mathbf{a}}\right|_{\mathbf{z}^{(0)}} + \left[\lambda_{\mathcal{A}_{a}^{(0)}}^{(0)}\right] \mathbf{I}_{\mathcal{A}_{\mathbf{a}}^{(0)}} = 0.$$

Then, as the gradients values are substituted by their value, one deduces the Lagrange multipliers value:

$$\lambda_{\mathcal{A}_{\mathbf{s}}^{(0)}}^{(0)} = 2.49\, m.s^2 \ \text{and} \ \lambda_{\mathcal{A}_{a}^{(0)}}^{(0)} = 6.26e{-}3\, kg/mm^2.$$

As a remark, these multipliers illustrate the optimal weight (of the slave problem (sP(**B**))) sensitivity with respect to a perturbation of the constraint on the area lower bound or stress constraint in tension, respectively.

The gradients of the weight and the stress constraints w.r.t. **B** are computed, respectively,

$$\left.\frac{\partial \widetilde{w}}{\partial \mathbf{B}}\right|_{\mathbf{z}^{(0)}} = [1.2, 1.9, 3.7, 5.9],$$

$$\left.\frac{\partial \widetilde{\mathbf{s}}_{\mathcal{A}_{\mathbf{s}}^{(0)}}}{\partial \mathbf{B}}\right|_{\mathbf{z}^{(0)}} = \left[0., 0., 0., -9.5e^2\right].$$

Thus, using Eq. 17, one deduces the gradient of Ψ:

$$\left.\frac{d\Psi}{d\mathbf{B}}\right|_{\mathbf{B}^{(0)}} = \left.\frac{\partial \widetilde{w}}{\partial \mathbf{B}}\right|_{\mathbf{z}^{(0)}} + \left[\lambda_{\mathcal{A}_{\mathbf{s}}^{(0)}}^{(0)}\right]^{\top} \left.\frac{\partial \widetilde{\mathbf{s}}_{\mathcal{A}_{\mathbf{s}}^{(0)}}}{\partial \mathbf{B}}\right|_{\mathbf{z}^{(0)}}$$
$$= [1.2, 1.9, 3.7, -17.7].$$

Physically, the values seem to indicate that the optimal weight (of the slave problem (sP(**B**))) is more sensitive to the choices of materials on the second structural element, when compared to the other one. Indeed, these sensitivities are only valid in a (close enough) neighborhood of $\mathbf{B}^{(0)}$. A change in the active constraint set could occur at intermediate values of **B**.

The history of the previous iterations is updated with $\mathbf{B}^{(0)}$ such that

$$K^{(0)} = \{\mathbf{B}^{(0)}\}.$$

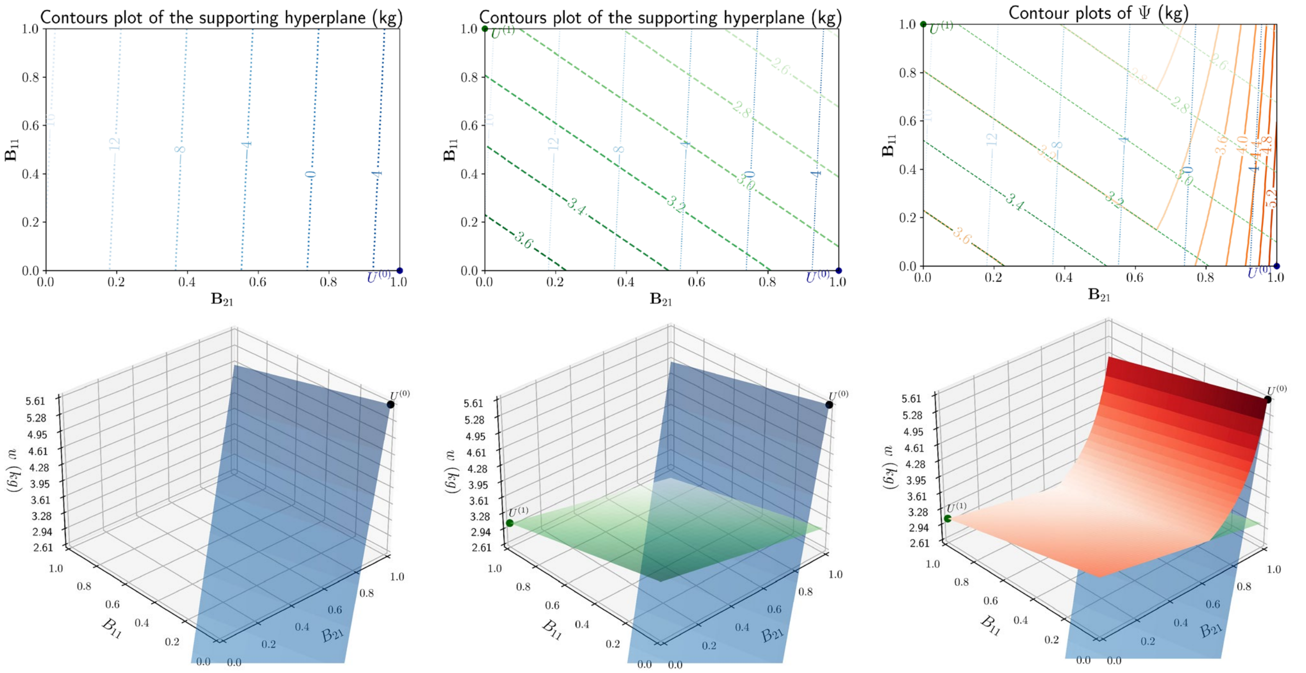
The MILP problem (MILP(k)) can now be set up. The solution of this problem provides the new integer candidate solution given by

$$\mathbf{B}^{(1)} = \text{vec}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

The optimal objective value is

$$\eta^{(0)} = -38.71,$$

meaning that the difference between the best known guess $U^{(0)}$ and the relaxed problem optimal objective value $\eta^{(0)}$ is lower than the given tolerance ϵ. Figure 5a shows the supporting hyperplane that provides the feasible set of the MILP problem at the first iteration. The



(a) A first supporting hyperplane of Ψ from the first iteration.

(b) A second supporting hyperplane of Ψ from the second iteration.

(c) Plot of the surface Ψ for every value of $\mathbf{B} \in \widetilde{C}^{2\times 2}$.

Fig. 5 Iterations of a 2-bar truss structure optimization example. The supporting hyperplanes at $\Psi(\mathbf{B}^{(0)})$ (resp., $\Psi(\mathbf{B}^{(1)})$) are built using the slope $\frac{d\Psi}{d\mathbf{B}}$ taken at $\mathbf{B}^{(0)}$ (resp., $\mathbf{B}^{(1)}$)

plotted supporting hyperplane, defined over $[0, 1] \times [0, 1]$, corresponds to the curve surface of the function

$$\begin{pmatrix} B_{11} \\ B_{21} \end{pmatrix} \to \Psi(\mathbf{B}^{(0)}) + \left.\frac{d\Psi}{d\mathbf{B}}\right|_{\mathbf{B}^{(0)}}^{\top} \left[\begin{pmatrix} B_{11} \\ 1 - B_{11} \\ B_{21} \\ 1 - B_{21} \end{pmatrix} - \mathbf{B}^{(0)} \right].$$

We note that one has $U^{(0)} = \Psi(\mathbf{B}^{(0)})$.

– *Second iteration* ($k = 1$)

The second iteration starts by solving primal problem, that reduces to an evaluation of Ψ (by solving (sP($\mathbf{B}$))) at the current guess $\mathbf{B}^{(1)}$:

$U^{(1)} = 3.07\,kg$, $\mathbf{a}^{(1)} = [300., 300.]\,\text{mm}^2$

Then, similarly to the first iteration of the algorithm, we estimate the gradient of Ψ with respect to $\mathbf{B}$:

$$\left.\frac{d\Psi}{d\mathbf{B}}\right|_{\mathbf{B}^{(1)}} = [1.19, 1.88, 1.19, 1.88].$$

The history of the previous iterations is updated with $\mathbf{B}^{(1)}$ i.e., $K^{(1)} = K^{(0)} \cup \{\mathbf{B}^{(1)}\}$ and the MILP problem (MILP(k)) can now be set up, as follows:

$$\begin{aligned}
\min_{\mathbf{B}\in C^{2\times 2}} \quad & \eta \\
\text{subject to} \quad & \eta \leq U^{(1)} - \epsilon \\
& \eta \geq \Psi(\mathbf{B}^{(1)}) + \left.\frac{d\Psi}{d\mathbf{B}}\right|_{\mathbf{B}^{(1)}}^{\top} (\mathbf{B} - \mathbf{B}^{(1)}) \\
& \eta \geq \Psi(\mathbf{B}^{(0)}) + \left.\frac{d\Psi}{d\mathbf{B}}\right|_{\mathbf{B}^{(0)}}^{\top} (\mathbf{B} - \mathbf{B}^{(0)})
\end{aligned} \tag{15}$$

Figure 5b depicts the result $U^{(1)}$ of the NLP problem (sP($\mathbf{B}$)) solved at $B^{(1)}$, and the associated new hyperplane behaves as an additional constraint for the new MILP problem. The optimal objective value is

$\eta^{(1)} = 3.07,$

that is equal to the best known guess $U^{(1)}$. This means that the current lower bound of the problem solution is now equal to its current upper bound. The problem (15) is thus infeasible, due to the first constraint violation. The solution found during this second iteration is the optimal solution.

The algorithm then stops, and the solution is such that

$$\widetilde{w}^* = U^{(1)} = 3.07\,kg$$
$$\mathbf{a}^* = \mathbf{a}^{(1)} = [300, 300]\,\text{mm}^2,$$
$$\mathbf{B}^* = \mathbf{B}^{(1)} = \text{vec}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

In other words, the optimal material for elements 1 and 2 is AL2139 and TA6V, respectively.

Figure 5c depicts the landscape of the function Ψ with respect to $\mathbf{B} \in \mathcal{C}^{2\times 2}$. One can see that the admissible solutions of the problem P applied to the 2-bar truss example are the four points at the boundary of Ψ where B_{11} and B_{21} take integer values. The optimal solution then corresponds to the point with the lowest value.

Remark 1 At the end of the optimization process, the optimal values for η coincides with the optimal value of Ψ. However, during the minimization process, the obtained values of η do not have necessarily a physical meaning; their values depends on the quality of the approximation provided by the convex hull based on the hyperplanes. For instance, the value for η is negative because the supporting hyperplane is not a good approximation for the function Ψ. The convex hull is then refined iteratively (by including new hyperplanes) until η corresponds to the value of Ψ at the final solution.

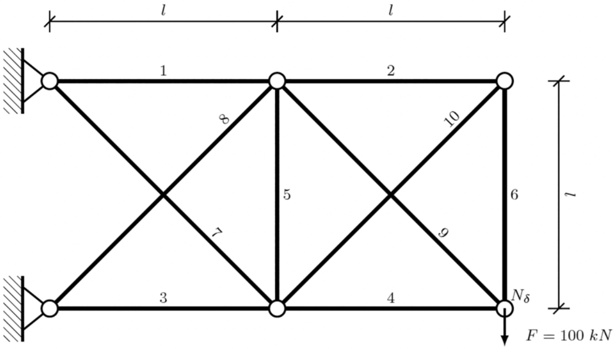


Fig. 6 10-bar truss seen as a scalable 2D cantilever problem with 2 blocks

Remark 2 For the Bi-level OA solver, the total computational cost is reduced to the computational effort required to solve 2 NLP problems and 2 MILP problems. Solving the same illustrative problem by enumeration (Baseline) would have require 2^2 NLP optimization problems. The Bi-level algorithm as proposed in Barjhoux et al. (2020) would required solving 6 NLP optimization problems.

4.3 A 10-bar truss structure

The 10-bar truss problem (Haftka and Gürdal 1992) is used to solve the mixed categorical-continuous optimization problem by enumeration, Bi-level or hybrid branch-and-bound (h-B &B) (Barjhoux et al. 2018b).

The 10-bar truss problem is illustrated in Fig. 6. A downward load $F = 100\,\text{kN}$ is applied vertically on node N_δ. A constraint on displacements is applied on the same node. Five cases with different bounds values $\bar{\mathbf{u}}$ on displacements are considered. For each of these cases, the displacements constraint is applied on node N_δ. Each structural element is also subjected to the stress constraints given by (11, 12, 13, and 14). The lower bounds, the upper bounds, and the initial areas are fixed to $100\,\text{mm}^2$, $1300\,\text{mm}^2$ and $1300\,\text{mm}^2$, respectively. Catalogs 1 and 2 point to materials AL2139 and TA6V, respectively. Materials properties are listed in Appendix 4. For this simple case, one has $n = 10$, $p = 2$, and $\mathbf{B} \in \mathcal{C}^{10,2}$.

Table 1 depicts the obtained results on a 10-bar truss mixed optimization using 5 different values of constraint on displacements. In all these cases, as shown by the Hamming distance d_h and the optimal weights w^*, the solutions found by `Baseline`, h-B &B, Bi-level, and Bi-level OA solvers are identical. The optimal solutions returned by the Genetic solver are not as good as the optimal weights found by the the rest of the solvers. In fact, although sometimes the Genetic solver is able to find the optimal catalogs (since $d_h = 0$), the continuous variables are not well handled (since

Table 1 Results of the 10-bar truss testcase with 5 different values of constraint on displacements

$\bar{\mathbf{u}}\,(mm)$	Baseline		h-B &B		Genetic		Bi-level		Bi-level OA	
	$\mathbf{c}^* = \mathbf{B}\boldsymbol{\gamma}$	w^*(kg)	d_h	w^*(kg)	d_h	w^*(kg)	d_h	w^*(kg)	d_h	w^*(kg)
-22	[2,2,1,1,1,2,2,1,2,1]	12.988	0	12.988	0	13.283	0	12.988	0	12.988
-20	[2,1,1,1,1,1,2,1,1,1]	13.996	0	13.996	0	14.423	0	13.996	0	13.996
-19	[2,1,1,1,1,1,2,1,1,1]	14.570	0	14.570	0	14.802	0	14.570	0	14.570
-18	[1,1,1,1,1,1,1,1,1,1]	15.175	0	15.175	2	15.642	0	15.174	0	15.174
-17	[1,1,1,1,1,1,1,1,1,1]	15.912	0	15.912	3	16.258	0	15.912	0	15.912

Comparison between the Bi-level OA, Bi-level, the Baseline solutions obtained by enumeration of the 2^{10} continuous optimizations, h-B &B, and the Genetic algorithm. The catalog 1 corresponds to material AL2139 and catalog 2 to TA6V

w^* is higher compared to the other solvers). The displacement constraint is active in all the cases; as far as the displacement constraint becomes more stringent, the material choice goes to the stiffest one despite of its high density. In our experiments, the optimal solutions of cases with maximum displacements equal to 18 mm and 17 mm contain indeed only TA6V material. Regarding the other constraints, the Euler buckling constraints were active on elements 10 and 8 for both test cases with a maximum displacement of 20 mm and 22 mm. The constraints were also active in all cases, but on different elements depending on the bound value on displacements. Namely, for the cases with a maximum displacement equals to 19 mm, 20 mm, and 22 mm, the stress constraints were active for the elements 2, 6, and 9. For the case 18 mm, the same constraint was active for the elements 1, 2, 6, 7, and 9; for the case 17 mm, the constraint was active for elements 1 and 7. Unlike the displacement constraints, in the provided examples, the local buckling constraints were not active at the solution. It seems that in our setting, such constraints do not have a significant role in the optimization process.

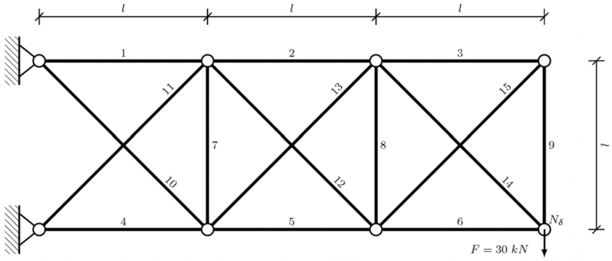


Fig. 7 An example of 2D cantilever problem with 3 blocks

4.4 Scalability of our algorithm

4.4.1 Scalability with respect to the number of elements

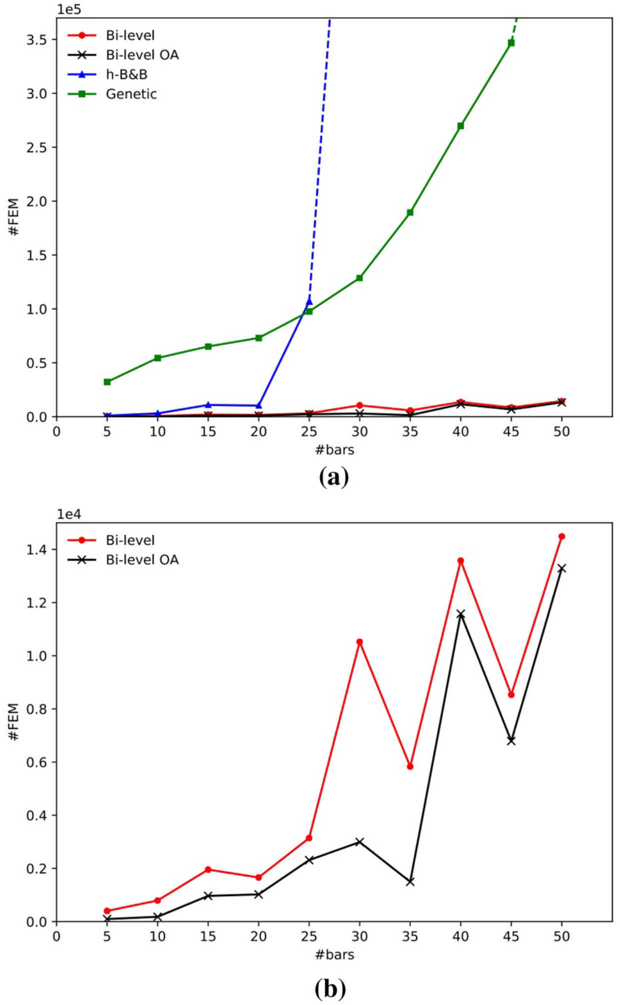
The objective of this test case is to describe the evolution of the computational cost with respect to the number of structural elements. It has been used in the literature to demonstrate the scalability of algorithms, see for instance Shahabsafa et al. (2018). Each block is composed of 4 nodes that are linked by 5 bars. An example of a parametric 2D cantilever structure with 3 blocks is given in Fig. 7. Table 2 presents the results obtained with structures composed of 1 to 10 blocks. In all cases, a force load $F = 30\,\text{kN}$ is applied on the node N_δ. The lower bounds, the upper bounds, and the initial areas are fixed to $100\,\text{mm}^2$, $2000\,\text{mm}^2$ and $2000\,\text{mm}^2$, respectively. No constraint on displacements is considered and each structural element is subjected to the stress constraints given by (11, 12, 13, and 14).

For each of the 10 cases, the results obtained by the Bi-level OA are compared to those obtained with reference solutions (Baseline, h-B &B) and Bi-level when available. First, for the three cases with 5 to 15 elements where a reference solution is available, it can be observed the global solution is found by the Bi-level OA. In these three cases, for all the tested solvers, the optimal categorical variable values are identical, excepted for the Genetic solver (based on the d_h values). For cases with more than 15 elements, the optima found by the Bi-level OA are slightly better than those obtained by the Genetic algorithm. The h-B &B solutions are noted with (*) since they are intermediate solutions: the solver was stopped after 24 h. The Bi-level OA solutions are very close (difference of 10^{-2} kg) to those obtained by the h-B &B. For cases with 40 and 45 elements, the Bi-level OA solutions are slightly lighter than

Table 2 A comparison of the obtained solutions for 10 instances of the scalable 2D cantilever problem, with a varying number of bars (from 5 to 50 bars)

#bars	Baseline	h-B &B				Genetic				Bi-level				Bi-level OA			
	w^*(kg)	d_h	w^*(kg)	#iter	#FEM	d_h	w^*(kg)	#iter	#FEM	d_h	w^*(kg)	#iter	#FEM	d_h	w^*(kg)	#iter	#FEM
5	2.56	0	2.56	10	1004	0	2.57	32	32300	0	2.56	2	400	0	2.56	2	96
10	6.06	0	6.06	26	3097	1	6.14	54	54500	0	6.06	2	792	0	6.06	2	181
15	10.23	0	10.23	95	10907	2	10.27	65	65200	0	10.23	4	1955	0	10.23	6	967
20	*	*	15.33	135	10315	*	15.59	73	73100	*	15.33	2	1659	*	15.33	7	1023
25	*	*	21.36	1199	610347	*	22.06	98	97700	*	21.36	3	3142	*	21.36	13	2312
30	*	*	28,30	4432	723388	*	28.84	129	128800	*	28.30	8	10522	*	28.30	13	2991
35	*	*	$36,17^{(*)}$	$5793^{(*)}$	$1096968^{(*)}$	*	37.00	189	189400	*	36.19	3	5830	*	36.19	6	1496
40	*	*	$44,97^{(*)}$	$5570^{(*)}$	$939726^{(*)}$	*	45.64	270	269800	*	44.97	7	13577	*	44.96	40	11578
45	*	*	$54,70^{(*)}$	$4181^{(*)}$	$818455^{(*)}$	*	55.98	347	346800	*	54.71	4	8531	*	54.67	20	6789
50	*	*	$65,35^{(*)}$	$4316^{(*)}$	$717627^{(*)}$	*	67.48	561	561200	*	65.34	6	14487	*	65.34	42	13290

We note that when optimizations last more than 24 h, the solver (Baseline, h-B &B) is stopped and the current solution (if exists) is marked by (*)



(a)

(b)

Fig. 8 Scalability of the Bi-level OA w.r.t. the number of structural elements. The computational cost's scaling of Bi-level OA and Bi-level with respect to the number of bars is almost linear, compared to the exponential computational cost of the h-B &B and Genetic solvers shown (**a**). The high computational cost of the h-B &B prevents from obtaining a solution for cases greater than 25 elements. The plot on (**b**) focuses on a comparison between the computational cost of Bi-level and Bi-level OA only. The computation cost is always lower than the Bi-level

the Bi-level. Furthermore, the number of analyses required by Bi-level OA is always lower than the number needed by the compared approaches, including Bi-level. The trends in terms of computational cost with respect to the number of elements are graphically represented in Fig. 8. The cost of the Genetic algorithm dominates the cost of h-B &B Bi-level and Bi-level OA. As with the Bi-level, the scaling of the Bi-level OA approach is nearly linear when compared to the h-B &B and Genetic approach. The trends in terms of Bi-level OA computational cost with respect to the number of elements are similar to the Bi-level computation cost. The observed efficiency makes the proposed approach relevant for higher dimensional problems.

4.4.2 Scalability with respect to the number of catalogs

The objective of this test case is to describe the computational cost scaling with respect to the number of categorical choices. The test case is the same 10-bar truss case presented in Sect. 4.3, with a constraint on displacements such that $\bar{\mathbf{u}} = 10\,\mathrm{mm}$. For this simple case, one has the number of structural elements fixed to $n = 10$, but p is varying from 5 to 90 catalogs. Each catalog is defined as a combination of different materials among AL2139, AL2024 and TA6V, with the member profiles I, T and C. For each member profile we consider using 10 different sizes. The material properties, the catalogs and the member profiles are listed in Appendix 4 (see, Tables 4, 6, and 7). Thus, by scaling the number of catalog choices, the full number of available categorical choices will range from 10^4 to 10^{90}.

Table 3 presents the results obtained by Bi-level OA and Bi-level. The h-B &B failed to solve the problem instances in 24 h. The optimal weight, the number of iterations (#ite), non-linear problems (#NLP) solved, and the number of individual calls to the structural solver (#FEM) are compared. First, in terms of #FEM and #NLP, the computational cost of Bi-level OA reveals to be almost linear with respect to the number of categorical values when compared to Bi-level. Furthermore, it is shown that for each case, the

Table 3 A comparison of the obtained solutions for 9 instances of the 10-bar truss problem, with a varying number of catalogs (from 4 to 90 catalogs)

#Catalogs	Bi-level				Bi-level OA			
	w^*(kg)	#iter	#NLP	#FEM	w^*(kg)	#iter	#NLP	#FEM
4	12.99	3	98	29245	12.99	84	84	8400
9	12.39	4	334	98871	12.39	89	89	3772
12	12.39	4	445	131358	12.39	61	61	2583
15	12.39	4	573	170407	12.39	45	45	1955
18	12.39	4	708	209201	12.39	65	65	2877
36	12.39	4	1404	416662	12.39	57	57	2232
45	12.15	3	1348	399172	12.15	69	69	2489
72	12.15	3	1864	551042	12.15	64	64	2898
90	12.15	3	2704	799166	12.15	86	86	3952

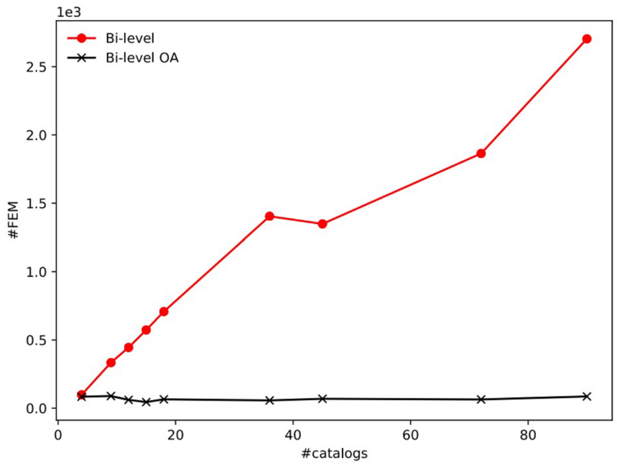


Fig. 9 Scalability of the Bi-level OA w.r.t. the number of catalogs. The computational cost's scaling of Bi-level OA with respect to the number of catalogs is nearly independent from the number of catalogs, compared to the quasi-linear computational cost of the Bi-level

optimal weights obtained by both solvers are close (the gap is less than 10^{-3}kg). This shows that the Bi-level is able to return good quality solutions even in cases with a large-scale categorical design space. Independently from the solver, it can be remarked that the optimal weights are identical from cases 4 to 36, and 45 to 90. This is due to the fact that the categorical values introduced in the design space from case 4 to 36 (or from 45 to 90) do not lead to any significant improvement for the optimal weight; the improvement is only observed from case 45 to 72. The computational cost with respect to the number of catalogs for all the tested solvers is depicted in Fig. 9.

4.5 120-bar truss

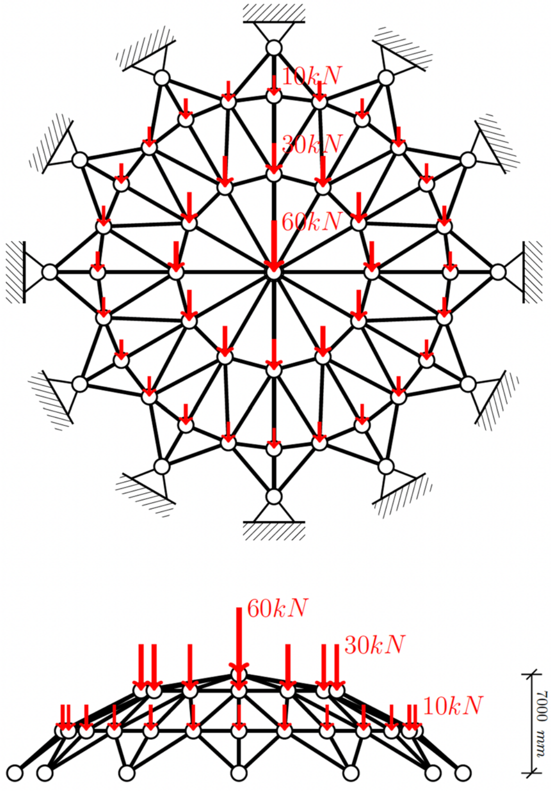


Fig. 10 Top and side view a 120-bar truss structure. Downward loads with three different magnitudes are applied

In this example, the structure of a 120-bar dome truss (Saka and Ulker 1992) detailed in Fig. 10 is considered. For each element, the categorical variable can take a value among 90 catalogs. With $n = 120$ and $p = 90$, the binary design space is $\mathcal{C}^{120\times90}$. The number of available categorical choices is thus equal to 90^{120}. Each catalog is defined as a combination of materials among AL2139, AL2024, and TA6V, with profiles I, T, and C (with 10 different sizes for each profile). The material properties are listed in Table 4 in Appendix 4. The catalogs are listed in Table 6 and the profiles in Table 7, in Appendix 4. The structure is subjected to a constraint on displacements: a maximum downward displacement of 10 mm is allowed on node 1 (i.e., the top of the dome). A downward load of 60 kN is applied on this same node, while 12 downward loads of 30 kN are applied on nodes 2 to 13 (i.e., inner ring) and 10 kN on nodes 14 to 37 (i.e., outer ring). For this test case, the lower bounds, the upper bounds, and the initial areas are fixed to 100 mm^2, 6000 mm^2 and 6000 mm^2, respectively. As for the previous tested problems, the areas of all the structural elements are handled as continuous design variables and each element is subjected to the stress constraints (11, 12, 13, and 14). The stress constraints will assess the structural integrity of the truss and avoid buckling in the members.

For this test problem, all the previously tested approaches (i.e., Genetic, h-B &B, and Bi-level) were unable to provide an optimum in a reasonable time (we could not converge to a competitive solution in 24 h). Only the Bi-level OA solver was able to converge to a competitive solution in approximately 2 h; the optimal weight returned by Bi-level OA is 1506 kg. Both the maximum displacement and the Euler buckling constraints interfere at the optimal solution. The Euler buckling constraints were active for all the structural elements, excepted the elements on the outer ring. Similarly to the previous test case, the local buckling constraints were not active at the solution. The optimal truss is pictured in Fig. 11. The categorical and continuous solution is provided in Table 5 in Appendix 4. We observe that only 3 choices have been

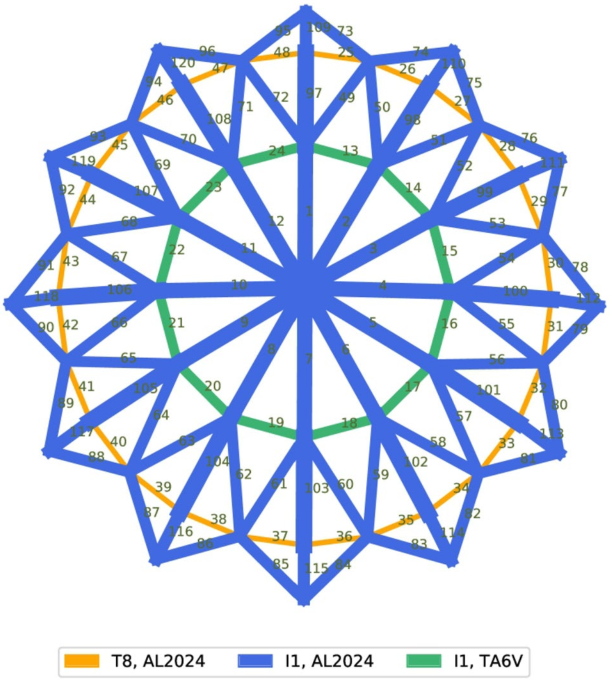


Fig. 11 Top view of the 120-bar truss mixed categorical-continuous optimization result

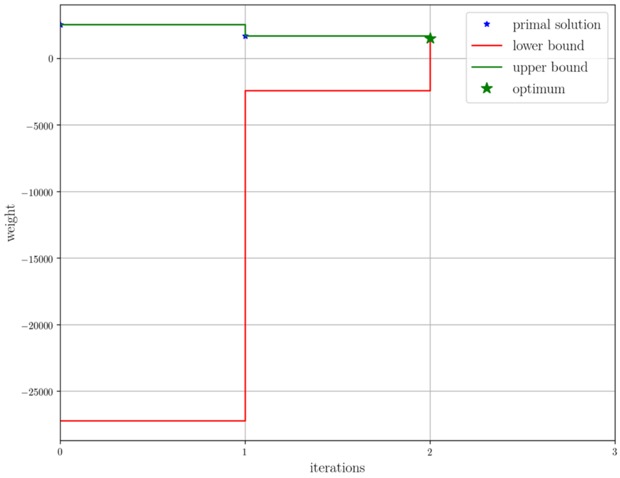


Fig. 12 History of the convergence of $\eta^{(k)}$ (i.e., the lower bound) and $U^{(k)}$ (i.e., the upper bound) during the application of the Bi-level OA method to solve the 120-bar truss problem

selected over a total of 90 . The material "TA6V" has been selected for members 13 to 24 (inner circle, in green), and "AL2024" for the rest of the structure. The profile "T8" has been selected for members 25 to 48 (outer circle, in orange), while "I1" is selected for the rest of the structure. The convergence history of $\eta^{(k)}$ (i.e., the lower bound) and $U^{(k)}$ (i.e., the upper bound) is depicted in Fig. 12. One can see that the optimization process is converging within 3 iterations. This means that it required to solve only 3 NLP (primal problems), within a total of 35339 calls to FEM.

5 Conclusion

In this paper, we solved a mixed categorical-continuous structural optimization problem with categorical variables, i.e., non-relaxable and non-ordered. The proposed algorithm used a bi-level decomposition of (P), and solved a sequence of master and slave problems. The resulting algorithm, named Bi-level OA, relied on the theory of the OA algorithm (Fletcher and Leyffer 1994; Bonami et al. 2008; Grossmann 2009) where the derivatives are estimated using a post-optimal sensitivity analysis (Fiacco 1976). Under a convexity assumption, we were able to guarantee the convergence of our proposed approach.

The numerical tests showed that the proposed method is capable of handling large-scale instances of the mixed categorical-continuous problem. The scalability in terms of computational cost has been tested with respect to the number of structural elements and number of categorical choices per element. Our convergence proof of the Bi-level OA relies on the convexity assumption with respect to the design variables, such assumption cannot be verified in the context of structural optimization problems. A further work could consist in studying convergence of the proposed strategy when non-convex cases occur.

Appendix 1. On the computation of $\frac{d\Psi}{d\mathbf{B}}\Big|_{\mathbf{B}^{(k)}}$ using post-optimal sensitivities

Gradient estimation, using post-optimal sensitivities, was introduced by (Fiacco 1976) using penalty approach. In the context of our structural optimization problem, the estimation of the gradient can be derived as follows. The Lagrangian of the problem (sP(**B**)) is given by

$$\begin{aligned}\mathcal{L}(\mathbf{a},\mathbf{B}) := {} & \widetilde{w}(\mathbf{a},\mathbf{B}) + \boldsymbol{\lambda}_{\mathbf{s}}^\top \widetilde{\mathbf{s}}(\mathbf{a},\mathbf{B}) + \boldsymbol{\lambda}_{\delta}^\top \widetilde{\boldsymbol{\delta}}(\mathbf{a},\mathbf{B}) \\ & + \boldsymbol{\lambda}_{\mathbf{a}}^\top(\mathbf{a}-\mathbf{a}) + \boldsymbol{\lambda}_{\bar{\mathbf{a}}}^\top(\mathbf{a}-\bar{\mathbf{a}}),\end{aligned}$$

where $\boldsymbol{\lambda}_{\mathbf{s}}$, $\boldsymbol{\lambda}_{\delta}$, $\boldsymbol{\lambda}_{\mathbf{a}}$, and $\boldsymbol{\lambda}_{\bar{\mathbf{a}}}$ are the Lagrange multipliers (column vectors) associated to the constraints $\widetilde{\mathbf{s}}$, $\widetilde{\boldsymbol{\delta}}$, $\mathbf{a}$, and $\bar{\mathbf{a}}$, respectively.

For a given (k) iteration, let $\mathbf{a}^{(k)}$ be the solution of the of problem (sP(**B**)) evaluated at $\mathbf{B}^{(k)}$ and define $\mathcal{A}_{\mathbf{s}}^{(k)}$, $\mathcal{A}_{\delta}^{(k)}$, and $\mathcal{A}_{\mathbf{a}}^{(k)}$, $\mathcal{A}_{\bar{\mathbf{a}}}^{(k)}$ as the sets of active constraints, i.e.,

$$\mathcal{A}_{\mathbf{s}}^{(k)} = \left\{\forall i \,|\, \widetilde{\mathbf{s}}_i\left(\mathbf{a}^{(k)}\right) = 0\right\}, \mathcal{A}_{\mathbf{a}}^{(k)} = \left\{\forall i \,|\, \mathbf{a}_i^{(k)} = \mathbf{a}_i\right\},$$
$$\mathcal{A}_{\boldsymbol{\delta}}^{(k)} = \left\{\forall i \,|\, \widetilde{\boldsymbol{\delta}}_i\left(\mathbf{a}^{(k)}\right) = 0\right\}, \mathcal{A}_{\bar{\mathbf{a}}}^{(k)} = \left\{\forall i \,|\, \mathbf{a}_i^{(k)} = \bar{\mathbf{a}}_i\right\}.$$

The active components of the constraints $\widetilde{\mathbf{s}}$ and $\widetilde{\boldsymbol{\delta}}$ will be noted by $\mathbf{s}^{(k)}_{\mathcal{A}_{\mathbf{s}}^{(k)}}$ and $\boldsymbol{\delta}^{(k)}_{\mathcal{A}_{\delta}^{(k)}}$, respectively. The components of $\mathbf{a}^{(k)}$ whose indices belong to $\mathcal{A}_{\mathbf{a}}^{(k)}$ (resp. $\mathcal{A}_{\bar{\mathbf{a}}}^{(k)}$) will be noted by $\mathbf{a}^{(k)}_{\underline{\mathcal{A}}}$ (resp. $\mathbf{a}^{(k)}_{\bar{\mathcal{A}}}$). Similarly, the Lagrange multipliers at the optimum will be denoted by $\boldsymbol{\lambda}_{\mathbf{s}}^{(k)}$, $\boldsymbol{\lambda}_{\boldsymbol{\delta}}^{(k)}$, $\boldsymbol{\lambda}_{\mathbf{a}}^{(k)}$, and $\boldsymbol{\lambda}_{\bar{\mathbf{a}}}^{(k)}$. Again, the Lagrange multipliers at the optimum related to the active constraints of $\widetilde{\mathbf{s}}$, $\widetilde{\boldsymbol{\delta}}$, and bounds constraints on $\mathbf{a}$ will be designated by noted $\boldsymbol{\lambda}^{(k)}_{\mathcal{A}_{\mathbf{s}}^{(k)}}$, $\boldsymbol{\lambda}^{(k)}_{\mathcal{A}_{\delta}^{(k)}}$, $\boldsymbol{\lambda}^{(k)}_{\mathcal{A}_{a}^{(k)}}$, and $\boldsymbol{\lambda}^{(k)}_{\mathcal{A}_{\bar{a}}^{(k)}}$, respectively.

Assuming that the objective and the constraints functions of (sP($\mathbf{B}$)) are continuously differentiable at $\mathbf{a}^{(k)}$ and that the gradients of active constraints at $\mathbf{a}^{(k)}$ are linearly independent. Then, by using the Karush–Kuhn–Tucker (KKT) optimality conditions applied to (sP($\mathbf{B}$)) at $\mathbf{B}^{(k)}$, one gets

$$\begin{aligned}&\left.\frac{\partial \widetilde{w}}{\partial \mathbf{a}}\right|_{\mathbf{z}^{(k)}} + \left[\boldsymbol{\lambda}^{(k)}_{\mathcal{A}_{\mathbf{s}}^{(k)}}\right]^\top \left.\frac{\partial \widetilde{\mathbf{s}}_{\mathcal{A}_{\mathbf{s}}^{(k)}}}{\partial \mathbf{a}}\right|_{\mathbf{z}^{(k)}} + \left[\boldsymbol{\lambda}^{(k)}_{\mathcal{A}_{\delta}^{(k)}}\right]^\top \left.\frac{\partial \widetilde{\boldsymbol{\delta}}_{\mathcal{A}_{\delta}^{(k)}}}{\partial \mathbf{a}}\right|_{\mathbf{z}^{(k)}} \\ &\quad - \left[\boldsymbol{\lambda}^{(k)}_{\mathcal{A}_{a}^{(k)}}\right]^\top \mathbf{I}_{\mathcal{A}_{\mathbf{a}}^{(k)}} + \left[\boldsymbol{\lambda}^{(k)}_{\mathcal{A}_{\bar{a}}^{(k)}}\right]^\top \mathbf{I}_{\mathcal{A}_{\bar{\mathbf{a}}}^{(k)}} = \mathbf{0}_n,\end{aligned} \tag{16}$$

where $\mathbf{z}^{(k)} := (\mathbf{a}^{(k)}, \mathbf{B}^{(k)})$ and the notation $g|_{\mathbf{z}}$ is used to denote the value of the function g at the point z. The matrices $I_{\mathcal{A}_{\bar{\mathbf{a}}}^{(k)}} \in \mathbb{R}^{|\mathcal{A}_{\bar{\mathbf{a}}}^{(k)}|\times n}$ and $I_{\mathcal{A}_{\mathbf{a}}^{(k)}} \in \mathbb{R}^{|\mathcal{A}_{\mathbf{a}}^{(k)}|\times n}$ are such that $\forall j \in [\![1, n]\!]$, one has

$$\forall i \in \mathcal{A}_{\mathbf{a}}^{(k)}, \quad \left(\mathbf{I}_{\mathcal{A}_{\mathbf{a}}^{(k)}}\right)_{ij} = \delta_{ij} \quad \text{and} \quad \forall i \in \mathcal{A}_{\bar{\mathbf{a}}}^{(k)}, \quad \left(\mathbf{I}_{\mathcal{A}_{\bar{\mathbf{a}}}^{(k)}}\right)_{ij} = \delta_{ij}$$

with δ_{ij} being the Kronecker symbol.

Consequently, once the problem (sP($\mathbf{B}$)) is solved for a given choice of $\mathbf{B}^{(k)}$, the Lagrange multipliers corresponding to active constraints can be obtained by solving the linear system given by (16). We note that, according to the KKT conditions, the computed values of the Lagrange multipliers have to be non-negative.

Now, under appropriate assumptions and by using (Fiacco 1976, Theorem 2.1), one can deduce that the function Ψ is continuously differentiable at $\mathbf{B}^{(k)}$. In fact, assuming that at each iteration (k) of our optimization process, one has

- The functions $\widetilde{w}$, $\widetilde{\mathbf{s}}$, and $\widetilde{\boldsymbol{\delta}}$ are twice continuously differentiable w.r.t. $\mathbf{a}$.
- $\frac{\partial \widetilde{w}}{\partial \mathbf{a}}, \frac{\partial \widetilde{\mathbf{s}}}{\partial \mathbf{a}}, \frac{\partial \widetilde{\boldsymbol{\delta}}}{\partial \mathbf{a}}$ are once continuously differentiable w.r.t. $\mathbf{B}$ in a neighborhood of $\mathbf{z}^{(k)}$,
- The second-order sufficient KKT conditions related with the problem (sP($\mathbf{B}$)) hold at $\mathbf{z}^{(k)}$,
- A strict complementary holds, i.e., in the example of the displacements constraints :

$$(\lambda_{\boldsymbol{\delta}}^{(k)})_i = 0 \iff \boldsymbol{\delta}_i(\mathbf{z}^{(k)}) < 0 \quad \forall i \in \mathcal{A}_{\boldsymbol{\delta}}^{(k)}.$$

In this case, by using (Fiacco 1976, Theorem 2.1), we conclude that the function Ψ is continuously differentiable, and its derivative taken in $\mathbf{B}^{(k)}$ is given by

$$\begin{aligned}\left.\frac{d\Psi}{d\mathbf{B}}\right|_{\mathbf{B}^{(k)}} =& \left.\frac{\partial \widetilde{w}}{\partial \mathbf{B}}\right|_{\mathbf{z}^{(k)}} + \left[\boldsymbol{\lambda}^{(k)}_{\mathcal{A}_{\mathbf{s}}^{(k)}}\right]^\top \left.\frac{\partial \widetilde{\mathbf{s}}_{\mathcal{A}_{\mathbf{s}}^{(k)}}}{\partial \mathbf{B}}\right|_{\mathbf{z}^{(k)}} \\ &+ \left[\boldsymbol{\lambda}^{(k)}_{\mathcal{A}_{\delta}^{(k)}}\right]^\top \left.\frac{\partial \widetilde{\boldsymbol{\delta}}_{\mathcal{A}_{\delta}^{(k)}}}{\partial \mathbf{B}}\right|_{\mathbf{z}^{(k)}},\end{aligned} \tag{17}$$

where we used the fact that bound constraints on the areas do not depend on $\mathbf{B}$ in order to eliminate the terms related to the bounds in the right hand side of Eq. (17).

In this section, we have detailed the mathematical theory of the post-optimality sensitivity analysis as stated in Fiacco (1976), and applied to the problem (sP($\mathbf{B}$)). Hence, for a given $\mathbf{a}^{(k)}$, the gradient of the function Ψ taken at $\mathbf{B}^{(k)}$ can be estimated in five main steps :

- Build the set of active constraints $\mathcal{A}_{\mathbf{s}}^{(k)}$, $\mathcal{A}_{\boldsymbol{\delta}}^{(k)}$, $\mathcal{A}_{\mathbf{a}}^{(k)}$, and $\mathcal{A}_{\bar{\mathbf{a}}}^{(k)}$.
- Compute the gradients of the objective and active constraints w.r.t. $\mathbf{a}$ at the point $(\mathbf{a}^{(k)}, \mathbf{B}^{(k)})$.
- Compute the Lagrange multipliers $\boldsymbol{\lambda}^{(k)}_{\mathcal{A}_{\mathbf{s}}^{(k)}}$ and $\boldsymbol{\lambda}^{(k)}_{\mathcal{A}_{\delta}^{(k)}}$ by solving the linear system (16).
- Compute the gradients of the objective and active constraints w.r.t. $\mathbf{B}$ at the point $(\mathbf{a}^{(k)}, \mathbf{B}^{(k)})$.
- Compute the post-optimal sensitivity $\frac{d\Psi}{d\mathbf{B}}$ at $\mathbf{B}^{(k)}$ using Eq. (17).

For all our optimization test cases, we did not get any numerical issue while deriving the post-optimality sensitivities. We thus believe that those assumptions are not strong on practical truss optimization.

Appendix 2. Derivatives of the weight and the constraint functions

The derivatives of the weight function and the constraints (with respect to the areas a) are obtained by applying the chain-rule theorem. Namely, the gradient analytic expression of the weight function (2) with respect to the areas a is given by

$$\frac{\partial \widetilde{w}}{\partial \mathbf{a}} = \left(\sum_{c=1}^{p} \rho(c)\ell_1 \mathbf{B}_{1c}, \ldots, \sum_{c=1}^{p} \rho(c)\ell_n \mathbf{B}_{nc} \right).$$

The gradient of the constraints are obtained as follows

$$\frac{\partial \widetilde{\mathbf{s}}_{ij}}{\partial \mathbf{a}} = \sum_{c=1}^{p} \mathbf{B}_{ic} \frac{\partial \mathbf{s}_{ij}}{\partial \mathbf{a}}(\mathbf{a}_i, c, \mathbf{\Phi}_i(\mathbf{a}, \mathbf{B})) \frac{\partial \mathbf{\Phi}_i}{\partial \mathbf{a}}(\mathbf{a}, \mathbf{B}).$$

The derivative of the internal axial force in each member of the structure $\frac{\partial \mathbf{\Phi}_i}{\partial \mathbf{a}}(\mathbf{a}, \mathbf{B})$ is given by

$$\frac{\partial \mathbf{\Phi}_i}{\partial \mathbf{a}}(\mathbf{a}, \mathbf{B}) = \widetilde{\boldsymbol{K}}_i^e(\mathbf{a}_i, \mathbf{B}_{i,:})\mathbf{T}_i \boldsymbol{u}_i(\mathbf{a}, \mathbf{B}) + \boldsymbol{K}_i^e(\mathbf{a}_i, \mathbf{B}_{i,:})\mathbf{T}_i \frac{\partial \boldsymbol{u}_i}{\partial \mathbf{a}}(\mathbf{a}, \mathbf{B}).$$

where $\widetilde{\boldsymbol{K}}_i^e(\mathbf{a}_i, \mathbf{B}_{i,:}) = \frac{\partial \boldsymbol{K}_i^e}{\partial \mathbf{a}}(\mathbf{a}_i, \mathbf{B}_{i,:})$. The derivative of the displacements $\frac{\partial \boldsymbol{u}_i}{\partial \mathbf{a}}(\mathbf{a}, \mathbf{B})$ is obtained, after derivation of (5), by

$$\frac{\partial \boldsymbol{u}_i}{\partial \mathbf{a}}(\mathbf{a}, \mathbf{B}) = -\left[\boldsymbol{K}_i^e(\mathbf{a}_i, \mathbf{B}_{i,:})\right]^{-1} \widetilde{\boldsymbol{K}}_i^e(\mathbf{a}_i, \mathbf{B}_{i,:}) u_i(\mathbf{a}, \mathbf{B}).$$

Appendix 3. The proposed Bi-level algorithm

Algorithm 1 A Bi-level framework using OA cuts.

```
initialize $\mathbf{B}^{(0)}, \epsilon > 0, K^{(-1)} \leftarrow \{\emptyset\}, U^{(-1)} \leftarrow +\infty$. Set Feasible $\leftarrow 1$ and $k \leftarrow 0$.
while Feasible $= 1$ do
    Compute $\Psi(\mathbf{B}^{(k)})$, let $\mathbf{a}^{(k)}$ be the approximate solution to the problem (sP($\mathbf{B}$)) evaluated at $\mathbf{B}^{(k)}$.
    if $\mathbf{a}^{(k)} \in \Omega(\mathbf{B}^{(k)})$ and $\Psi(\mathbf{B}^{(k)}) < U^{(k)}$ then
        $U^{(k)} \leftarrow \Psi(\mathbf{B}^{(k)})$
    else
        $U^{(k)} \leftarrow U^{(k-1)}$
    end if
    $K^{(k)} \leftarrow K^{(k-1)} \bigcup \{\mathbf{B}^{(k)}\}$
    Estimate $\frac{d\Psi}{d\mathbf{B}}$ at $\mathbf{B}^{(k)}$ as given in Section 3.2.1.
    Let $\mathbf{B}^{(k+1)}$ and $\eta^{(k+1)}$ be the approximate solution of (MILP($k$))
    if $\eta^{(k+1)}$ satisfies the constraints of (MILP($k$)) then
        Feasible $\leftarrow 1$
    else
        Feasible $\leftarrow 0$
    end if
    $k \leftarrow k + 1$
end while
return $\mathbf{a}^* \leftarrow \mathbf{a}^{(k-1)}$, $\mathbf{B}^* \leftarrow \mathbf{B}^{(k-1)}$, and $\widetilde{w}^* \leftarrow U^{(k-1)}$.
```

Appendix 4. Test cases data

See Tables 4, 5, 6, and 7.

Table 4 Numerical details on materials attributes for the test cases

	AL2139	AL2024	TA6V
Density [kg/mm^3]	$2.8e^{-6}$	$2.77e^{-6}$	$4.43e^{-6}$
Young modulus [MPa]	$7.1e^4$	$7.4e^4$	$11.0e^4$
Poisson coefficient [−]	0.3	0.33	0.33
Tension allowable [MPa]	$1.5e^2$	$1.6e^2$	$11.0e^2$
Compression allowable [MPa]	$2.0e^2$	$2.1e^2$	$8.6e^2$

Table 5 The obtained solution for the 120-bar truss problem

Elements	Catalog	$\mathbf{a}[mm^2]$
$1 \cdots 12$	62	1100
$13 \cdots 24$	32	695
$25 \cdots 47$	89	379
$48 \cdots 72$	62	773
$73 \cdots 96$	62	799
$97 \cdots 108$	62	799
$109 \cdots 120$	62	1195

Table 6 A description of the categorical design space related to the 120-bar truss problem ($c \in \{1, \ldots, 90\}$)

c	Cross section	Material
$1 \cdots 10$	$I_1 \ldots I_{10}$	AL2139
$11 \cdots 20$	$C_1 \ldots C_{10}$	AL2139
$21 \cdots 30$	$T_1 \ldots T_{10}$	AL2139
$31 \cdots 40$	$I_1 \ldots I_{10}$	TA6V
$41 \cdots 50$	$C_1 \ldots C_{10}$	TA6V
$51 \cdots 60$	$T_1 \ldots T_{10}$	TA6V
$61 \cdots 70$	$I_1 \ldots I_{10}$	AL2024
$71 \cdots 80$	$C_1 \ldots C_{10}$	AL2024
$81 \cdots 90$	$T_1 \ldots T_{10}$	AL2024

Table 7 Definition of the profiles I, C and T reference detailed geometry of the 120-bar truss problem

	$\mathbf{x}_0$[1] [mm]	$\mathbf{x}_0$[2] [mm]	$\mathbf{x}_0$[3] [mm]
I_1, C_1, T_1	5	50	40
I_2, C_2, T_2	10	110	40
I_3, C_3, T_3	10	90	40
I_4, C_4, T_4	10	100	40
I_5, C_5, T_5	5	100	40
I_6, C_6, T_6	10	60	40
I_7, C_7, T_7	15	100	40
I_8, C_8, T_8	10	70	35
I_9, C_9, T_9	10	80	40
I_{10}, C_{10}, T_{10}	10	90	45

Acknowledgements This work is part of the MDA-MDO project of the French Institute of Technology IRT Saint Exupery. We would lke to thank D. Bettebghor for his help and advices. We are also very thankful for two anonymous reviewers; their insightful comments improved significantly the content of the paper. We wish to acknowledge the PIA framework (CGI, ANR) and the project industrial members for their support, financial backing and/or own knowledge: Airbus, Altran Technologies, Capgemini DEMS France and CERFACS. The support of F. Gallard (IRT Saint Exupery) for his scientific and technical assistance on multilevel schemes implementation is gratefully acknowledged.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Replication of results All the results in this paper are obtained using homemade Python code relying on the GEMS library, that will be released under open source license in 2021. All the necessary data and equations are available in this paper to reproduce the results. The geometries are depicted on Fig. 7 and Fig. 10, material data are provided in Table 4, and the solution of the 120-bar truss is given in Table 5. The NLP and MINLP optimizations are performed thanks to the NLOPT (Johnson 2008) package and Coin-or Branch and Cut wrapped into the Google ortools suite (Perron and Furnon 2019), respectively. Both libraries are under open source license.

References

Achtziger W, Stolpe M (2007) Truss topology optimization with discrete design variables–guaranteed global optimality and benchmark examples. Struct Multidisc Optim 34(1):1–20

Barjhoux PJ, Diouane Y, Grihon S, Bettebghor D, Morlier J (2018a) A Bilevel methodology for solving a structural optimization problem with both continuous and categorical variables. In: 2018 Multidisciplinary Analysis and Optimization Conference. American Institute of Aeronautics and Astronautics, Atlanta, Georgia, pp 1–16

Barjhoux PJ, Diouane Y, Grihon S, Bettebghor D, Morlier J (2018b) Mixed Variable structural optimization: toward an efficient hybrid algorithm. In: Advances in structural and multidisciplinary optimization. Springer, Cham, pp 1880–1896

Barjhoux PJ, Diouane Y, Grihon S, Bettebghor D, Morlier J (2020) A bi-level methodology for solving large-scale mixed categorical structural optimization. Struct Multidisc Optim 62(1):337–351

Benders JF (1962) Partitioning procedures for solving mixed-variables programming problems. Numer Math 4(1):238–252

Benson HP, Horst R (1991) A branch and bound-outer approximation algorithm for concave minimization over a convex set. Comput Math Appl 21(6–7):67–76

Bonami P, Biegler LT, Conn AR, Cornuéjols G, Grossmann IE, Laird CD, Lee J, Lodi A, Margot F, Sawaya N, Wächter A (2008) An algorithmic framework for convex mixed integer nonlinear programs. Discret Optim 5(2):186–204

Deb K, Goyal M (1998) A flexible optimization procedure for mechanical component design based on genetic adaptive search. J Mech Des 120(2):162

Duran MA, Grossmann IE (1986) An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Math Program 36(3):307–339

Fiacco AV (1976) Sensitivity analysis for nonlinear programming using penalty methods. Math Program 10:287

Filomeno Coelho R (2014) Metamodels for mixed variables based on moving least squares: application to the structural analysis of a rigid frame. Optim Eng 15(2):311–329

Fletcher R, Leyffer S (1994) Solving mixed integer nonlinear programs by outer approximation. Math Program 66(1–3):327–349

Forrest J, Ralphs T, Vigerske S, LouHafer, Kristjansson B, jpfasano, EdwinStraver, Lubin M, Santos HG, rlougee, Saltzman M (2018) coin-or/cbc: Version 2.9.9

Fortin FA, De Rainville FM, Gardner M, Parizeau M, Gagné C (2012) DEAP: evolutionary algorithms made easy. J Mach Learn Res 13:2171–2175

Gallard F, Vanaret C, Guenot D, Gachelin V, Lafage R, Pauwels B, Barjhoux PJ, Gazaix A (2018) GEMS: a Python Library for Automation of Multidisciplinary Design Optimization Process Generation. Technical Report January, IRT Saint-Exupéry

Gao H, Breitkopf P, Coelho RF, Xiao M (2018) Categorical structural optimization using discrete manifold learning approach and custom-built evolutionary operators. Struct Multidisc Optim 58(1):215–228

Garrido-Merchán EC, Hernández-Lobato D (2018) Dealing with categorical and integer-valued variables in Bayesian Optimization with Gaussian Processes. Neurocomputing 380:20

Geoffrion AM (1972) Generalized benders decomposition. J Optim Theory Appl 10(4):237–260

Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning, vol Addison-We. Addison-Wesley Longman Publishing Co., Inc., Boston

Grihon S (2012) PRESTO: A rapid sizing tool for airframe conceptual design studies. In: LMS European Aeronautical Conference, Toulouse

Grihon S (2018) Structure sizing optimization capabilities at airbus. In: Schumacher A, Vietor T, Fiebig S, Bletzinger KU, Maute K (eds) Advances in structural and multidisciplinary optimization. Springer, Cham, pp 719–737

Grossmann IE (2009) MINLP: Outer approximation algorithm. In: Floudas CA, Pardalos PM (eds) Encyclopedia of optimization. Springer, Boston, MA, pp 2179–2183

Haftka RT, Gürdal Z (1992) Elements of structural optimization, volume 11 of solid mechanics and its applications. Springer, Dordrecht

Haftka RT, Watson LT, Haftka RT, Watson LT (2006) Decomposition theory for multidisciplinary design optimization problems with mixed integer Quasiseparable subsystems. Optim Eng 7(7):135–149

Herrera M, Guglielmetti A, Xiao M, Filomeno Coelho R (2014) Metamodel-assisted optimization based on multiple kernel regression for mixed variables. Struct Multidisc Optim 49(6):979–991

Hijazi H, Bonami P, Ouorou A (2014) An outer-inner approximation for separable mixed-integer nonlinear programs. INFORMS J Comput 26(1):31–44

Johnson SG (2008) The NLopt nonlinear-optimization package. http://ab-initio.mit.edu/nlopt

Krogh C, Jungersen MH, Lund E, Lindgaard E (2017) Gradient-based selection of cross sections: a novel approach for optimal frame structure design. Struct Multidisc Optim 56(5):959–972

Liao T, Socha K, Montes de Oca MA, Stutzle T, Dorigo M (2014) Ant colony optimization for mixed-variable optimization problems. IEEE Trans Evol Comput 18(4):503–518

Merval A (2008) L'Optimisation Multiniveaux D'une Structure. PhD Thesis

Müller J, Shoemaker CA, Piché R (2013) SO-MI: a surrogate model algorithm for computationally expensive nonlinear mixed-integer black-box global optimization problems. Comput Opera Res 40(5):1383–1400

Nouaouria N, Boukadoum M (2011) A particle swarm optimization approach to mixed attribute data-set classification. In: 2011 IEEE Symposium on Swarm Intelligence. IEEE, pp 1–8

Pelamatti J, Brevault L, Balesdent M, Talbi EG, Guerin Y (2019) Efficient global optimization of constrained mixed variable problems. J Global Optim 73(3):583–613

Perron L, Furnon V (2019) OR-Tools, Version 7.2, Google

Roy S, Moorey K, Hwang J, Gray J, Crossley W, Martins J (2017) A mixed integer efficient global optimization algorithm for the simultaneous aircraft allocation-mission-design problem. In: 58th AIAA/ASCE/AHS/ASC Structures. Structural Dynamics, and Materials Conference 2017, pp 1–18

Roy S, Crossley WA, Stanford B, Moore KT, Gray JS (2019) A mixed integer efficient global optimization algorithm with multiple infill strategy—applied to a wing topology optimization problem. In: AIAA Scitech 2019 Forum

Rufato RC, Diouane Y, Henry J, Ahlfeld R, Morlier J (2022) A mixed-categorical data-driven approach for prediction and optimization of hybrid discontinuous composites performance. In: AIAA AVIATION 2022 Forum

Saka M, Ulker M (1992) Optimum design of geometrically nonlinear space trusses. Comput Struct 42(3):289–299

Samuelides M, Bettebghor D, Grihon S, Merval A, Morlier J (2009) Modèles réduits en optimisation multiniveau de structures aéronautiques. In: 9e Colloque national en calcul des structures. CSMA, Giens, France

Saves P, Bartoli N, Diouane Y, Lefebvre T, Morlier J, David C, Nguyen Van E, Defoort S (2022) Multidisciplinary design optimization with mixed categorical variables for aircraft design. In: AIAA Scitech 2022 Forum

Schutte JF, Haftka RT, Watson LT (2004) Decomposition and two-level optimization of structures with discrete sizing variables. Constraints. https://doi.org/10.2514/6.2004-1541

Shahabsafa M, Mohammad-Nezhad A, Terlaky T, Zuluaga L, He S, Hwang JT, Martins JRRA (2018) A novel approach to discrete truss design problems using mixed integer neighborhood search. Struct Multidisc Optim 58(6):2411–2429

Sigmund O (2011) On the usefulness of non-gradient approaches in topology optimization. Struct Multidisc Optim 43(5):589–596

Stegmann J, Lund E (2005) Discrete material optimization of general composite shell structures. Int J Numer Methods Eng 62(14):2009–2027

Stolpe M (2007) On the reformulation of topology optimization problems as linear or convex quadratic mixed 0–1 programs. Optim Eng 8(2):163–192

Stolpe M (2011) To bee or not to bee–comments on Discrete optimum design of truss structures using artificial bee colony algorithm. Struct Multidisc Optim 44(5):707–711

Stolpe M (2015) Truss topology optimization with discrete design variables by outer approximation. J Global Optim 61(1):139–163

Stolpe M, Sandal K (2018) Structural optimization with several discrete design variables per part by outer approximation. Struct Multidisc Optim 57(5):2061–2073

Svanberg K (2002) A class of globally convergent optimization methods based on conservative convex separable approximations. SIAM J Optim 12(2):555–573

Turner M (1959) The direct stiffness method of structural analysis. In: Structural and Materials Panel Paper, AGARD Meeting. Aachen

Turner MJ, Martin HC, Leible RC (1964) Further Development and Applications of Stiffness Methods. In: Colman JF (ed) Matrix methods of structural analysis, vol 1, 1st edn. Macmillian, New York, pp 203–266

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.