A Bilevel Methodology for solving a Structural Optimization Problem with both Continuous and Categorical Variables

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

A Bilevel Methodology for Solving a Structural Optimization Problem with both Continuous and Categorical Variables

Pierre-Jean Barjhoux*
French Institute of Technology IRT Saint Exupery, Toulouse, 31432, France

Youssef Diouane†
ISAE-SUPAERO, Toulouse, 31400, France

Stéphane Grihon‡
Airbus Operations SAS, Toulouse, 31300, France

Dimitri Bettebghor§
ONERA The French Aerospace Lab, Chatillon, 92320, France

Joseph Morlier¶
Universite de Toulouse, Institut Clement Ader (ICA), CNRS-ISAE SUPAERO-INSA-Mines Albi-UPS, Toulouse, 31400, France

In industrial structural optimization problems, two kinds of variables are involved : continuous variables, like geometrical parameters, and categorical variables like choices of material and cross-sectional shapes. A key difficulty of this kind of problem is the sensitivity of the computation cost with respect to the number of categorical design variables. In this article, a new methodology is proposed to address this mixed continuous-categorical structural optimization problem. For this purpose, a 10-bar truss test case representative for industrial design is introduced. Promising results have been obtained in terms of computation cost, compared to approaches based on combinatorial algorithms.

Nomenclature

n	=	number of structural elements or bars in the structure
p	=	number of catalogs available for each element of the structure
$\mathcal{M}_{n,p}(\mathbb{R})$	=	set of matrices of size $n \times p$ with real coefficients
$\mathcal{B}_{n,p}$	=	set of matrices of size $n \times p$ with binary coefficients that describe choices
$0_{n,p}$	=	set of matrices of size $n \times p$ with null coefficients
Γ	=	a set of catalogs, i.e.; the list of available combination of choices for each element, $\Gamma = \{1, \ldots, p\}$
a	=	vector of areas of all elements or bars, $a \in \mathbb{R}^n$
t	=	vector of thicknesses of all elements or bars, $a \in \mathbb{R}^n$
c	=	categorical variables, ie a choice of catalog for each bar, $c \in \Gamma^n$
B	=	matrix of binary variables, $B \in \mathcal{B}_{n,p}$
B_i	=	i^{th} line of the matrix of binary variables B, $B_i \in \mathcal{B}_{1,p}$
$\mathbf{\Phi}$	=	vector of internal forces in the structure, $\mathbf{\Phi}(a, t, c) \in \mathbb{R}^n$
m	=	number of stress constraints per element
$\boldsymbol{s}$	=	stress constraints in initial problem description, $\boldsymbol{s}(a, t, c) \in \mathcal{M}_{n,m}(\mathbb{R})$
$\boldsymbol{g}$	=	stress constraints after problem reformulation, $\boldsymbol{g}(a, t, B) \in \mathcal{M}_{n,m}(\mathbb{R})$
$\boldsymbol{w}$	=	weight of the whole structure, $\boldsymbol{w}(a, c) \in \mathbb{R}$

*PhD student, Embedded Systems Department, pierre-jean.barjhoux@irt-saintexupery.com
†Associate Professor, Complex Systems Engineering Department, youssef.diouane@isae.fr
‡Expert in Structural Optimization, Engineering Strucrures, stephane.grihon@airbus.com
§Research Scientist, Aeroelasticity and Structural Dynamics department, dimitri.bettebghor@onera.fr
¶Professor, Department of Mechanics Structures and Materials, joseph.morlier@isae.fr

$\boldsymbol{d}$	=	constraint on displacement of one node, $\boldsymbol{d}(a, t, c) \in \mathbb{R}$
$\lambda^{ub_a}, \lambda^{ub_t}, \lambda^{lb_a}, \lambda^{lb_t}$	=	lagrange multiplier of lower bounds on areas and thicknesses, and then of upper bounds, $\lambda \in \mathbb{R}^n$

I. Introduction

In the field of structural optimization, two kinds of design variables are involved. Continuous or sizing variables describe the size of aircraft structural parts: in the case of thin-sheet stiffened sizing, they represent panel thicknesses and stiffening cross-sectional areas. On the other hand, technological choices (e.g. material choices, composite stacking sequences choices, cross-section shapes) or even computation assumptions (e.g. buckling margin policy) are depicted by categorical variables. These categorical variables are non-ordered. For example there is no ordering between combinations of material and cross-section shapes choices.

The challenge of this kind of optimization problem lies in the mixed continuous-categorical nature of the design space. More precisely, the large dimension of the categorical design space makes this kind of problem very hard to solve. Because of the curse of dimensionality, existing generic method solving Mixed Integer Non-Linear Programming (MINLP) are not efficient enough to face this problem in industrial context.

In this paper, details on related work are provided in a first place. Then the formulation of the optimization problem, as encountered in the industry, is described. It is followed by a description of the new approach proposed in this article. This approach is based on the decomposition of the mixed structure optimization problem in two levels: one where the sizing variables are optimized at given categorical choices, and a master problem that handles the categorical variables formulated as integers. Finally, an adapted 10-bar truss mixed structural optimization test case is presented and the results obtained are detailed. Promising results have been obtained compared to an hybrid branch and bound algorithm [1] in terms of computation cost. Indeed, the number of optimizations is proportional to the number of elements times the number of available choices per element, instead of a computation cost proportional to the number of elements at the power of the number of choices.

II. Related work

To handle this kind of problem, mixed-variable optimization algorithms have been proposed, such as Genetic Algorithms [2], Differential Evolution [3], Particle Swarm Optimization [4], Ant Colony Optimization [5], and Pattern Search [6]. Some algorithms rely on continuous relaxation, or consider discrete variables as ordered [7–9]. In other cases, categorical variables are handled through a native mixed-variable optimization approach for both categorical and continuous variables without relaxation [10, 11]. Other strategies consist in turning categorical variables into continuous ones, allowing the use of continuous optimization algorithms. For example, continuous indices containing shape and material information are built [12], meta-models are constructed through Multiple Kernel Regression [13]. Other approaches rely on the structure of the problem, and suggest to handle discrete and continuous variables in two different problems, for example in [14, 15]. Decomposition theory for multidisciplinary optimization has been adapted to mixed integer quasiseparable subsystems [16] and then applied to structural optimization with discrete variables [17]. However, in most of the existing approaches that can handle MINLP problems, the algorithms are not able to solve a mixed structural optimization problem at an industrial scale. Furthermore, in many cases the algorithm relies on an ordering or relaxation of the discrete variables, that is not straightforward in the case presented in this article.

In industry, structural optimization software solutions handle continuous variables while categorical ones are often ignored. In fact, discrete variables always introduce a combinatorial behavior leading to an exponential number of configurations to be explored. This makes difficult to maintain the optimization performance while managing all variables types simultaneously. For instance, at Airbus, two structural optimization software were developed to split up the problem. The first one is inspired by classical continuous optimization approaches using gradients. The second solver, $PRESTO$ [18], focuses on discrete and categorical variables in a preliminary phase of the design process where trade-offs are evaluated [19]. In $PRESTO$, the overall optimization process alternates between local (element-wise) optimizations at given internal loads, and an internal loads update at given choices by Finite Element Model (FEM) analysis. The advantage of this method lies in the fact that the categorical variables are considered independent from each other during the element-wise optimization process. This makes this approach very efficient, and affordable in terms of computation costs even for industrial applications. However, one of the drawbacks is that since the optimizations are not performed on the whole structure, it is not possible to take into account global constraints like rigidity constraints. Furthermore, this optimization process is driven by local choices. In [1], the problem is decomposed in two levels. In a master level, the categorical variables are handled. In the slave problem, the continuous sizing variables are optimized, the choices being fixed. The categorical variables, at the upper level, are handled through a branch and bound algorithm. Usually, branch and bound lower bound computations rely on convex relaxation of the discrete variable. Since the non-ordered nature of the categorical variables does not allow to relax the categorical variable, an innovative formulation of a sub-problem for lower bound computations has been proposed. Thus, in this approach, the optimization is performed on the whole structure ; the FEM computations are nested into the optimizations. This approach can thus handle displacements constraints. However, the number of optimizations remains prohibitive for industrial applications.

III. Problem description

Let us name a the areas where $a \in \mathbb{R}^n$, and t the thicknesses where $t \in \mathbb{R}^n$. The set Γ is an enumerated set that contains all combinations of choices, for example the combination of choices of a material and the stiffener type of an element. To each combination of these choices is attributed an indice, that is the value of the catalog belonging to Γ. If p is the number of catalogs, we can note $\Gamma = \{1, \dots, p\}$. The categorical variables c can take values of the set Γ^n (number of categorical choices per element at the power of the number of elements). Each value of this set is known as a catalog. In the industrial case, the structural model counts around 100 elements and from 10 up to 100 catalogs. This means that the continuous design space counts 200 continuous variables, and the categorical design space counts 100 variables. Furthermore, the categorical design space Γ^n counts 100^{10} up to 100^{100} combinations describing the choices for the whole structure. This high combinatory dimension enforces the need for a methodology to solve efficiently such problems. The optimization problem is formulated as follows:

$$\begin{aligned} \underset{c \in \Gamma^n, a \in \mathbb{R}^n, t \in \mathbb{R}^n}{\text{minimize}} \quad & \boldsymbol{w}(a, t, c) \\ \text{subject to} \quad & \boldsymbol{s}(a, t, c) \leq 0 \\ & \boldsymbol{d}(a, t, c) \leq 0 \\ & \underline{a} \leq a \leq \bar{a} \\ & \underline{t} \leq t \leq \bar{t} \end{aligned} \tag{1}$$

with $\underline{a}$, $\underline{t}$, the lower bounds on areas and thicknesses, and $\bar{a}$, $\bar{t}$ their upper bounds. In this formulation, the constraints $\boldsymbol{s}$ ensure the structural strength element per element. It can be noted that $\boldsymbol{s}(a, t, c)$ is a matrix with n rows and m columns corresponding to the type of constraints per element. The topology of the structure to optimize is assumed to be fixed.

Furthermore, it is essential to describe the dependency of the constraints to state variables like internal loads $\boldsymbol{\Phi}$, and design parameters (a,t,c):

$$\boldsymbol{s}(a,t,c) = \begin{array}{c} \\ elt_1 \\ elt_2 \\ \vdots \\ elt_n \end{array} \begin{array}{c} \begin{array}{cccc} \text{Constraint type 1} & \text{Constraint type 2} & \dots & \text{Constraint type } m \end{array} \\ \begin{pmatrix} \boldsymbol{s}_{11}(a_1, t_1, c_1, \boldsymbol{\Phi}(a,t,c)) & \boldsymbol{s}_{12}(a_1, t_1, c_1, \boldsymbol{\Phi}(a,t,c)) & \dots & \boldsymbol{s}_{1m}(a_1, t_1, c_1, \boldsymbol{\Phi}(a,t,c)) \\ \boldsymbol{s}_{21}(a_2, t_2, c_2, \boldsymbol{\Phi}(a,t,c)) & \boldsymbol{s}_{22}(a_2, t_2, c_2, \boldsymbol{\Phi}(a,t,c)) & \dots & \boldsymbol{s}_{2m}(a_2, t_2, c_2, \boldsymbol{\Phi}(a,t,c)) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{s}_{n1}(a_n, t_n, c_n, \boldsymbol{\Phi}(a,t,c)) & \boldsymbol{s}_{n2}(a_n, t_n, c_n, \boldsymbol{\Phi}(a,t,c)) & \dots & \boldsymbol{s}_{nm}(a_n, t_n, c_n, \boldsymbol{\Phi}(a,t,c)) \end{pmatrix} \end{array} \tag{2}$$

Note that the continuous and categorical variables (i.e.; areas a, thicknesses t and categorical variables c hiding material or stiffening principle choices) have a significant role in this problem. The categorical variables affect the weight $\boldsymbol{w}$, internal forces $\boldsymbol{\Phi}$, rigidity constraints $\boldsymbol{d}$, stress constraints $\boldsymbol{s}$. On the other hand, continuous variables affect the weight, internal loads, stress constraints and rigidity constraints. It is worth to note that a change in a categorical variable or area will modify the loads distribution $\boldsymbol{\Phi}$ along the structure. Since the stresses $\boldsymbol{s}$ require the value of $\boldsymbol{\Phi}$, each component of $\boldsymbol{s}$ vector depends on the whole structure description. Internal forces $\boldsymbol{\Phi}$ are computed using a finite element model (FEM). During this process the linear elasticity equation is solved:

$$Ku = F \tag{3}$$

with $K \in \mathcal{M}_{q,q}(\mathbb{R})$ the stiffness matrix of the considered structure, $u \in \mathbb{R}^q$ the vector of displacements, $F \in \mathbb{R}^q$ the vector of forces applied on the nodes. q is the dimension of the space times the number of nodes. Then, the strain is computed from the displacements u, and the constraints $\boldsymbol{s}$ from the Hooke's law.

The rigidity constraint $\boldsymbol{d}$ on displacements ensures that at one node of the truss, the displacement will not exceed a predefined value.

IV. A bi-level approach

To tackle the problem (1) at an industrial scale, a reformulation is proposed. Let $\mathcal{B}_{n,p}$ describe a set of matrices whose coefficients have integer values. Each line of the matrices has only one binary coefficient equal to 1, the other coefficients are 0. Physically, each line of the matrix describes a choice of catalog applied to a given element, and only one choice can be made for each element.

$$\mathcal{B}_{n,p} = \left\{ X = (x_{ij})_{\substack{1\leq i\leq n \\ 1\leq j\leq p}} \Big/ \; x_{ij} \in \{0,1\}, \text{ and } \sum_{j=1}^{p} x_{ij} = 1, \;\; \forall i \in \{1,\ldots,n\} \right\}$$

Let be $\gamma \in \mathbb{Z}^p$ a vector of all catalogs in Γ. If $c \in \Gamma$ and a matrix $B \in \mathcal{B}_{n,p}$ describe the same choices, the relation between c and B is then given by:

$$c = B\gamma$$

A numerical example of equivalent values for c and B is described in Fig. 1.

$$c = \begin{pmatrix} 4 \\ 2 \\ 1 \\ 3 \\ 2 \end{pmatrix}$$

(a) Categorical variables : $c \in \Gamma^5$

$$B = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

(b) Binary coding of categorical variables $B \in \mathcal{B}_{5,4}$

Fig. 1 Specific coding of the categorical design variable in the case of a 5 elements structure : $n = 4$**. In this case** Γ **is a set of 4 catalogs :** $p = 4$**. Both** B **and** c **describe the same choices : choice of catalog 4 for bar 1, catalog 2 for bar 2, catalog 1 for bar 3, etc. Fig. a shows the categorical variables description used in the approach presented in Section III :** $c \in \Gamma^5$**. Fig. b illustrates the binary coding** $B \in \mathcal{B}_{5,4}$ **of choices** c**.**

Let be $\boldsymbol{g}$ the function that outputs a matrix $\boldsymbol{g}(a,t,B) \in M_{n,m}(\mathbb{R})$ of stress constraints. B_i is the i^{th} line of the matrix B. The expression of the coefficients of this matrix is thus:

$$g_{ij}(a,t,B) = \sum_{k=1}^{p} B_{ik} s_{ij}(a_i, t_i, B_i\gamma, \Phi(a,t,B\gamma)) \;\; \forall (i,j) \in [\![1,n]\!] \times [\![1,m]\!] \tag{4}$$

The mixed variable optimization problem can be written as follows :

$$\begin{aligned} \underset{B\in\mathcal{B}_{n,p}\; a\in\mathbb{R}^n \; t\in\mathbb{R}^n}{\text{minimize}} \quad & \boldsymbol{w}(a,t,B) \\ \text{subject to} \quad & \boldsymbol{g}(a,t,B) \leq 0 \\ & \boldsymbol{d}(a,t,B) \leq 0 \\ & \underline{a} \leq a \leq \bar{a} \\ & \underline{t} \leq t \leq \bar{t} \end{aligned} \tag{5}$$

The weight $\boldsymbol{w}$ is defined as:

$$\boldsymbol{w}(a,t,B) = \sum_{i=1}^{n}\sum_{j=1}^{p} B_{ij}\rho_j a_i L_i \tag{6}$$

with the vector of densities $\rho \in \mathbb{R}^p$, corresponding to the density associated to each catalog described in γ, in the same ordering.

A. Algorithm 1

This problem is based on a decomposition in two optimization problems. The slave problem consists in a continuous optimization with respect to continuous variables and relaxed components of the categorical variables. The optimal value of the objective of this problem is then linearized with respect to the categorical variables, and optimized in a master problem that computes the optimal categorical choices. This work is still in progress (see Appendix) and it is preferred to provide in this paper a detailed Bi-level methodology with new results.

B. Algorithm 2

In this approach, the post-optimal sensitivity analysis of the slave problem in the previous algorithm is replaced by a computation of the slope of the optimal Lagrangian between two discrete choices.

The problem (5) is decomposed in two problems, $(P_{a,t})$ and (P_B). The choices are optimized at the master level (P_B), relying on optimization with respect to a,t in the slave problem $(P_{a,t})$, where B is a parameter. The slave problem $(P_{a,t})$ consists thus in an optimization with respect to continuous design variables that are thicknesses and areas (resp. t, a), while the binary variable is fixed at a given B:

$$
\begin{aligned}
(P_{a,t})\big|_B \quad \underset{a,t\in\mathbb{R}^n}{\text{minimize}} \quad & \boldsymbol{w}(a,t,B) \\
\text{subject to} \quad & \boldsymbol{g}(a,t,B) \le 0 \\
& \boldsymbol{d}(a,t,B) \le 0 \\
& \underline{a} \le a \le \bar{a} \\
& \underline{t} \le t \le \bar{t}
\end{aligned}
\tag{7}
$$

This corresponds to a sizing step, where the choices of materials and shape are fixed. At the first step (0) of the algorithm, the problem $(P_{a,t})\big|_{B^{(0)}}$ is solved at a given $B^{(0)}$. Let be $w^*(B)$ the optimal weight of $(P_{a,t})\big|_B$ taken at optimal areas and thicknesses, respectively $a^*(B)$ and $t^*(B)$.

Let be $\mathcal{L}(B)$ the value of the Lagrangian of the problem $(P_{a,t})\big|_B$, i.e.;

$$
\begin{aligned}
\mathcal{L}(B) = \boldsymbol{w}(a(B),t(B),B) + \sum_{i=1}^{n}\sum_{j=1}^{m} \lambda_{ij}^{\boldsymbol{g}}(B)\, \boldsymbol{g}_{ij}(a(B),t(B),B) + \lambda^{\boldsymbol{d}}(B)\boldsymbol{d}(a(B),t(B),B) \\
+ \sum_{i=1}^{n}\left(\lambda_i^{ub_a}(B)(a_i-\underline{a}_i) + \lambda_i^{lb_a}(B)(\underline{a}_i - a_i) + \lambda_i^{ub_t}(B)(t_i-\underline{t}_i) + \lambda_i^{lb_t}(B)(\underline{t}_i - t_i)\right)
\end{aligned}
\tag{8}
$$

Let be $\boldsymbol{g}^*(B)$ and $\boldsymbol{d}^*(B)$ the constraints (on stress and displacement, resp.) at the optimum, and $\boldsymbol{w}^*(B)$ the optimal weight taken at the optimum of $(P_{(a,t)})$, such that:

$$
\begin{aligned}
\boldsymbol{g}^*(B) &= \boldsymbol{g}(a^*(B),t^*(B),B), \\
\boldsymbol{d}^*(B) &= \boldsymbol{d}(a^*(B),t^*(B),B), \\
\boldsymbol{w}^*(B) &= \boldsymbol{w}(a^*(B),t^*(B),B).
\end{aligned}
\tag{9}
$$

The associated optimal Lagrangian $\mathcal{L}^*(B)$ is then given by:

$$
\begin{aligned}
\mathcal{L}^*(B) = \boldsymbol{w}^*(B) + \sum_{i=1}^{n}\sum_{j=1}^{m} \lambda_{ij}^{\boldsymbol{g}}(B)\, \boldsymbol{g}_{ij}^*(B) + \lambda^{\boldsymbol{d}}(B)\boldsymbol{d}^*(B) \\
+ \sum_{i=1}^{n}\left(\lambda_i^{ub_a}(B)(a_i^*(B)-\underline{a}_i) + \lambda_i^{lb_a}(B)(\underline{a}_i - a_i^*(B)) + \lambda_i^{ub_t}(B)(t_i^*(B)-\underline{t}_i) + \lambda_i^{lb_t}(B)(\underline{t}_i - t_i^*(B))\right).
\end{aligned}
\tag{10}
$$

A linearization of $\mathcal{L}^*$ around $B^{(0)}$ is given by:

$$\hat{\mathcal{L}}^*(B) = \mathcal{L}^*(B^{(0)}) + \sum_{i=1}^{n}\sum_{j=1}^{p}\left[\frac{\Delta\mathcal{L}^*(B^{(0)})}{\Delta B^{ij}}\right]_{ij}(B_{ij} - B_{ij}^{(0)}) \tag{11}$$

with :

- ΔB^{ij} a matrix of perturbation of $B^{(0)}$ on the coefficient $B_{ij}^{(0)}$,
- $B_{ij}^{(0)}$ the binary design variable that drives the choice of the j^{th} catalog on the i^{th} element, at a given step (0) of the methodology,
- $\left[\frac{\Delta\mathcal{L}(B^{(0)})}{\Delta B^{ij}}\right]_{ij}$ the coefficient ij of the matrix $\frac{\Delta\mathcal{L}(B^{(0)})}{\Delta B^{ij}}$.

At the master level, the linearization of the Lagrangian (11) of $(P_{a,t})\big|_B$ is minimized with respect to B :

$$(P_B) \quad \underset{B\in\mathcal{B}_n}{\text{minimize}} \quad \hat{\mathcal{L}}^*(B)\ . \tag{12}$$

Considering equation (11), solving problem (12) is equivalent to find, for each element i, the binary coefficient B_{ij} such that the discrete slope of the Lagrangian with respect to B_{ij} is minimal. The problem (12) is thus equivalent to compute $\hat{\mathcal{L}}^*(B)$ so that :

$$\begin{aligned} B_{ij}^{(1)} &= \begin{cases} 1 \ \text{ if } \ j = l_i \ (\forall i \in \{1,\dots,n\}) \\ 0 \ \text{ else} \end{cases} \\ \text{with } l_i &= \underset{j\in[\![1,p]\!]}{\text{argmin}} \left[\frac{\Delta\mathcal{L}^*(B^{(0)})}{\Delta B^{ij}}\right]_{ij} \quad (\forall i \in \{1,\dots,n\})\ . \end{aligned} \tag{13}$$

The discrete slope of the Lagrangian $\left[\frac{\Delta\mathcal{L}^*(B^{(0)})}{\Delta B^{ij}}\right]_{kl}$ is given by :

$$\left[\frac{\Delta\mathcal{L}^*(B^{(0)})}{\Delta B^{ij}}\right]_{kl} = \begin{cases} \dfrac{\mathcal{L}^*(B^{(0)} + \Delta B^{ij}) - \mathcal{L}^*(B^{(0)})}{\Delta B^{ij}_{kl}} & \text{if} \quad B_{ij}^{(0)} \neq 1 \\ 0 & \text{else} \end{cases} \tag{14}$$

with $\Delta B^{ij} \in \mathcal{M}_{n,p}(\mathbb{R})$, so that $\Delta B^{ij} = 0$ if $B_{ij}^{(0)} \neq 1$, otherwise $k \in \{1,\dots,n\}$ and $l \in \{1,\dots,p\}$:

$$\Delta B^{ij}_{kl} = \begin{cases} 1 \ \text{ if } \ (k,l) = (i,j) \\ -1 \ \text{ if } \ k = i \text{ and } B^{(0)}_{k,l} = 1\ . \\ 0 \ \text{ else} \end{cases} \tag{15}$$

A numerical example of the computation of the perturbation ΔB^{ij}, for a given $B^{(0)}$ is depicted in Fig. 2.

$$B^0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \qquad \Delta B^{11} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad \Delta B^{12} = \begin{pmatrix} -1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad \Delta B^{11} = \begin{pmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

Fig. 2 A numerical example of the computation of ΔB^{1j} ($\forall j \in [\![1,p]\!]$) with respect to B^0, with $n = 2$, $p = 3$.

According to KKT conditions, the Lagrangian at the optimum is strictly equal to the optimal value of the objective. Assuming that $\mathcal{L}^*(B)$ is a good approximation of $\mathcal{L}(B)$ around $B^{(0)}$:

$$\mathcal{L}^*(B) = \boldsymbol{w}^*(B)\ . \tag{16}$$

According to equations (14) and (15), solving problem (13) is thus equivalent to find the binary variable such that, for each element, it minimizes the comparison between the optimal weight computed at a perturbed choice $B^{(0)} + \Delta B^{ij}$, and the optimal weight at $B^{(0)}$.

The problem (13) is thus equivalent to compute $\hat{\mathcal{L}}^*(B)$ so that :

$$
\begin{aligned}
&B_{ij}^{(1)} = \begin{cases} 1 \ \text{ if } \ j = l_i \ (\forall i \in \{1, \ldots, n\}) \\ 0 \ \text{ else} \end{cases} \\
&\text{with } l_i = \underset{j \in [\![1,p]\!]}{\operatorname{argmin}} \ \boldsymbol{w}(B^{(0)} + \Delta B^{ij}) - \boldsymbol{w}(B^{(0)}) \ \ (\forall i \in \{1, \ldots, n\}).
\end{aligned} \tag{17}
$$

Once $B^{(1)}$ has been computed, the next iteration of the algorithm starts again with the computation of the solution of the sizing problem $(P_{a,t})\big|_{B^{(1)}}$ taken at $B^{(1)}$. The full process of the algorithm is depicted in Algorithm 1.

Algorithm 1 Bi-level algorithm

$B^{(0)} \longleftarrow$ initialize B ▹ initialize B
$w_{hist}, t_{hist}, a_{hist}, B_{hist} \longleftarrow list(), list(), list(), list()$ ▹ initialize lists that store results history
$w^{(0)}, t^{(0)}, a^{(0)} \longleftarrow$ solve $(P_{a,t})\big|_{B^{(0)}}$
add $w^{(0)}, t^{(0)}, a^{(0)}, B^{(0)}$ to $w_{hist}, t_{hist}, a_{hist}, B_{hist}$
$k \longleftarrow 1$
while $|w^{(k)} - w^{(k-1)}| < \epsilon$ **do** ▹ check convergence on weight
  for each bar i from 1 to n **do**
    $r \in \mathbb{R}^p$
    for each catalog j from 1 to p **do**
      if $B_{ij}^{(k-1)} = 1$ **then** ▹ avoids to repeat optimization at $B^{(k-1)}$
        $r_j \longleftarrow w^{(k-1)}$
      else
        $r_j \longleftarrow$ solve $(P_{a,t})\big|_{B^{(k-1)}+\Delta B^{ij}}$ ▹ solve sizing taken at perturbed choice, r_j is the optimal weight
    $l \longleftarrow \underset{j \in [\![1,p]\!]}{\operatorname{argmin}} \ r_j$ ▹ find choice that minimizes weight
  $B_{ij}^{(k)} = 1$ if $j = l_i$ $(\forall i \in \{1, \ldots, n\})$, 0 otherwise ▹ build binary solution of iteration $^{(k)}$
  $w^{(k)}, t^{(k)}, a^{(k)} \longleftarrow$ solve $(P_{a,t})\big|_{B^{(k)}}$ ▹ sizing problem taken at current optimal choices (for all elements)
  add $w^{(k)}, t^{(k)}, a^{(k)}, B^{(k)}$ to $w_{hist}, t_{hist}, a_{hist}, B_{hist}$ ▹ store results history of iteration $^{(k)}$
  $k \longleftarrow k + 1$
$w^*, indice \longleftarrow \min(w_{hist}), \operatorname{argmin}(w_{hist})$ ▹ retrieve best solution in case of oscillations
$t^*, a^*, B^* \longleftarrow t_{hist}(indice), a_{hist}(indice), B_{hist}(indice)$
return w^*, t^*, a^*, B^* ▹ returns optimal weight, thicknesses, areas, and catalogs

V. Test Case

In order to test the new methodology, the well-known 10-bar truss optimization test-case often introduced as a continuous optimization problem [20, 21], has been adapted [1] to match the needs of a mixed optimization problem. The truss is illustrated in Fig. 3.

In this case, the sizing variables are the areas a of the bars cross-sections. Each $s_{ij}(a, B\gamma)$ is computed as the difference between the stress in the current element i and allowable in tension, compression, and the computed values of local and Euler buckling. It is worth noting that the internal forces computation requires all the components of the variable B, and all areas a, i.e.;

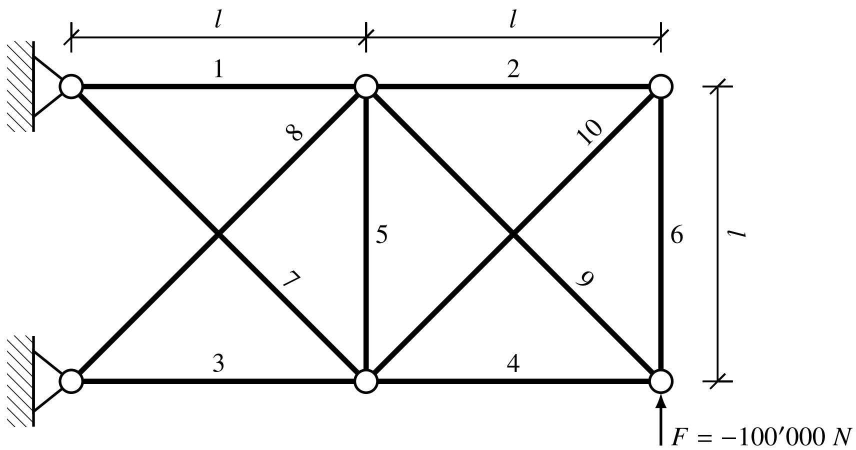


Fig. 3 Illustration of a 10-bar truss structure. Each member works in tension-compression only. A vertical load is applied on node 4. The length of horizontal and vertical bars is $l = 1m$.

$$
\boldsymbol{s}(a, B\gamma) = \begin{array}{c} \\ elt_1 \\ elt_2 \\ \vdots \\ elt_n \end{array} \begin{array}{c} \text{Allowable Tension} \quad\quad \text{Allowable Compression} \quad\quad \text{Euler Buckling} \quad\quad \text{Local Buckling} \\ \begin{pmatrix} \boldsymbol{s}_{11}(a_1, B_1\gamma, \boldsymbol{\Phi}(a, B\gamma)) & \boldsymbol{s}_{12}(a_1, B_1\gamma, \boldsymbol{\Phi}(a, B\gamma)) & \boldsymbol{s}_{13}(a_1, B_1\gamma, \boldsymbol{\Phi}(a, B\gamma)) & \boldsymbol{s}_{14}(a_1, B_1\gamma, \boldsymbol{\Phi}(a, B\gamma)) \\ \boldsymbol{s}_{21}(a_2, B_2\gamma, \boldsymbol{\Phi}(a, B\gamma)) & \boldsymbol{s}_{22}(a_2, B_2\gamma, \boldsymbol{\Phi}(a, B\gamma)) & \boldsymbol{s}_{23}(a_2, B_2\gamma, \boldsymbol{\Phi}(a, B\gamma)) & \boldsymbol{s}_{24}(a_2, B_2\gamma, \boldsymbol{\Phi}(a, B\gamma)) \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{s}_{n1}(a_n, B_n\gamma, \boldsymbol{\Phi}(a, B\gamma)) & \boldsymbol{s}_{n2}(a_n, B_n\gamma, \boldsymbol{\Phi}(a, B\gamma)) & \boldsymbol{s}_{n3}(a_n, B_n\gamma, \boldsymbol{\Phi}(a, B\gamma)) & \boldsymbol{s}_{n4}(a_n, B_n\gamma, \boldsymbol{\Phi}(a, B\gamma)) \end{pmatrix} \end{array} \tag{18}
$$

The definition of the constraint $\boldsymbol{g}$ becomes, with B_k the k^{th} row of the matrix B:

$$
g_{ij}(a, B) = \sum_{k=1}^{p} B_{ik} s_{ij}(a_i, B_k\gamma, \Phi(a, B\gamma)) \;\; \forall (i, j) \in [\![1, n]\!] \times [\![1, m]\!] \tag{19}
$$

Thus, in this test case the number of elements n is 10 and the number of stress constraints m is 4. The constraints $\boldsymbol{g} \in \mathcal{M}_{10,p}$ are defined using equation (4). Here is the problem formulation :

$$
\begin{aligned}
\underset{B\in\mathcal{B}_{10,p} \;\; a\in\mathbb{R}^{10}}{\text{minimize}} \quad & \boldsymbol{w}(a, B) = \sum_{i=1}^{10} \sum_{j=1}^{p} B_{ij} \rho_j a_i L_i \\
\text{subject to} \quad & \boldsymbol{g}(a, B) \leq 0 \\
& \boldsymbol{d}(a, B) \leq 0 \\
& 100 \leq a \leq 1300 \;\; (mm)
\end{aligned} \tag{20}
$$

The sizing problem (P_a) is thus :

$$
\begin{aligned}
(P_a) \quad \underset{a\in\mathbb{R}^n}{\text{minimize}} \quad & \boldsymbol{w}(a, B) \\
\text{subject to} \quad & \boldsymbol{g}(a, B) \leq 0 \\
& \boldsymbol{d}(a, B) \leq 0 \\
& 100 \leq a \leq 1300 \;\; (mm)
\end{aligned} \tag{21}
$$

The implementation of this test-case relies on GEMS, a Python library for programing MDO processes [22]. Each elementary function like binary variable encoding, weight, internal forces and constraints computations has been

catalog	material	shape
0	M_1	**I**
1	M_2	**I**
2	M_1	**C**
3	M_2	**C**

Table 1 Catalogs definition with respect to case numbers, $c \in \Gamma^4$

	M_1	M_2
density (kg/mm^3)	$2.8\ 10^{-6}$	$2.7\ 10^{-6}$
young modulus (MPa)	$7.1\ 10^4$	$7.4\ 10^4$
poisson coefficient $(-)$	0.3	0.33
tension allowable (MPa)	$1.6\ 10^2$	$1.5\ 10^2$
compression allowable (MPa)	$2.1\ 10^2$	$2.0\ 10^2$

Table 2 Numerical details on Material definitions

implemented as a GEMS MDO discipline object. GEMS automatically chains all of them, depending on the design variables, objective and constraints declared to the GEMS MDO scenario object. The jacobian of each MDO discipline has been implemented, so that GEMS automatically builds the jacobian of objective and constraints. The gradients of objective and constraints have been checked with finite differences. Due to the quasi-separable structure of the problem, the implementation of the Method of Moving Asymptotes (MMA) [23] from the Python package NLopt has been used to perform the continuous optimizations.

VI. Results

The catalogs definition, with respect to the case number, is detailed in Table 1. The characteristics of materials M_1 and M_2 are presented in Table 2.

The results obtained by the bi-level optimization can be compared to results obtained by 3 other approaches : $PRESTO$ [19], hybrid branch and bound [1] (h-B&B), and a complete enumeration. Each case is thus solved 4 times. All cases are different in terms of categorical design space Γ and bound on displacements. The complete enumeration is an enumeration of all continuous optimizations taken at all possible discrete choices. This is why for $p = 2$ and $n = 10$, it required $2^{10} = 1024$ optimizations on areas. This means that the complete enumeration that would solve problem (21) requires 1024 different matrices B fixed. All the results are exposed in Table 3.

In cases 1 to 3 and 7 to 9, the bound constraint on displacements is active. These cases can not be solved by the $PRESTO$ approach. On the other hand, cases 4 to 6 and 10 can be solved by a $PRESTO$ approach since this constraint is not active (and can be omitted in the optimization problem). It is worth to note that an optimization during the $PRESTO$ approach is less costly than an optimization with the other 3 methods. Indeed, with $PRESTO$ the FEM is evaluated at each global iteration during the internal loads computation, i.e. outside element-wise optimizations. For example the case 4 required a total 200 optimizations but only 10 evaluations to the FEM, one per global iteration. First, it can be observed that the optimal choices and weight (and thus areas) computed by the 4 approaches in all cases are equal. Furthermore, the number of optimizations required by the Bi-level approach seems promizing. Indeed, it can be remarked that the total number of optimizations required to solve this problem with a complete enumeration is 1024 for $p = 2$. The number of optimizations required by the h-B&B approach varies from 72 to 252, i.e. around 7% to 25% of the computation cost of a complete enumeration. The Bi-level approach requires 33 optimizations (around 3% of the computation cost), and is converged after 3 global iterations. The efficiency of the Bi-level approach becomes more interesting when the number of catalogs is increased. Indeed, for cases 7 to 10, the Bi-level requires from 1.7% to 6.8% of the computations required by the h-B&B approach. The computation cost reaches 0.002% when it is compared to the complete enumeration. Thus, according to these experiments, the h-B&B computation cost is proportional to the number of catalogs p at the power of the number of n structural elements (p^n). The computation cost of the Bi-level is proportional to the number of catalogs times the number of structural elements ($p \times n$). Furthermore, all the $p \times n$ optimizations performed at each global iteration of the Bi-level approach can be performed in parallel. With the h-B&B approach, p optimizations per branching step can be launched in parallel without modifying the overall methodology. The optimal weights obtained by the Bi-level approach are very close to those obtained with h-B&B and complete enumeration approaches. In particular for cases 4 to 6, the optimal catalog 0 is chosen at the 4^{th} bar instead of choice 1, leading to a higher weight. For cases 7 to 10, it can be remarked that even if optimal choices seem very different from the results obtained by the h-B&B approach, the optimal weights are very close. In the other hand, it is worth to note that when the number of catalogs p is increased, convergence difficulties can be observed with the Bi-level approach.

Cases			**Number of optimizations**			
n°	**p**	**displ. bound** (mm)	**PRESTO**	**Complete enumeration**	**h-B&B**	**Bi-Level**
1	2	-17	$^{(*)}$	1024	72	33
2	2	-18	$^{(*)}$	1024	102	33
3	2	-19	$^{(*)}$	1024	252	33
4	2	-19,96	200 + 10 $^{(**)}$	1024	110	33
5	2	-22	220 + 11 $^{(**)}$	1024	108	33
6	2	-30	180 + 9 $^{(**)}$	1024	108	33
7	4	-17	$^{(*)}$	1048576 $^{(***)}$	7592	217
8	4	-18	$^{(*)}$	1048576 $^{(***)}$	12676	217
9	4	-19	$^{(*)}$	1048576 $^{(***)}$	9228	217
10	4	-22	440 + 11 $^{(**)}$	1048576 $^{(***)}$	3212	217
Cases			**Optimal weight** (kg)			
n°	**p**	**displ. bound** (mm)	**PRESTO**	**Complete enumeration**	**h-B&B**	**Bi-Level**
1	2	-17	$^{(*)}$	15,180	15,180	15,180
2	2	-18	$^{(*)}$	14,522	14,522	14,522
3	2	-19	$^{(*)}$	14,069	14,069	14,11
4	2	-19,96	13.946	13,898	13,898	13,914
5	2	-22	13.946	13,898	13,898	13,914
6	2	-30	13.946	13,898	13,898	13,914
7	4	-17	$^{(*)}$	$^{(***)}$	15.180	15.181
8	4	-18	$^{(*)}$	$^{(***)}$	14.522	14.522
9	4	-19	$^{(*)}$	$^{(***)}$	14.067	14.11
10	4	-22	13.946	$^{(***)}$	13.898	13.898
Cases			**Optimal catalogs** $c = B\gamma$			
n°	**p**	**displ. bound** (mm)	**PRESTO**	**Complete enumeration**	**h-B&B**	**Bi-Level**
1	2	-17	$^{(*)}$	1111111111	1111111111	1111111111
2	2	-18	$^{(*)}$	1111110111	1111110111	1111110111
3	2	-19	$^{(*)}$	0011100111	0011100111	0011100111
4	2	-19,96	0000100101	0001100101	0001100101	0000100101
5	2	-22	0000100101	0001100101	0001100101	0000100101
6	2	-30	0000100101	0001100101	0001100101	0000100101
7	4	-17	$^{(*)}$	$^{(***)}$	3131113131	1313131111
8	4	-18	$^{(*)}$	$^{(***)}$	3111112111	2200003222
9	4	-19	$^{(*)}$	$^{(***)}$	0011102111	0300033000
10	4	-22	0000100101	$^{(***)}$	0001300101	0300033000

Table 3 Number of optimizations, optimal weights, and optimal catalogs obtained for 10 different cases. Each case takes different values of bounds on displacements and/or number of catalogs. $^{(*)}$ It is not possible to handle constraints on displacements with $PRESTO$. $^{(**)}$ <number of local optimizations> + <number of calls to FEM computations>. $^{(***)}$ Computation not performed because of computation cost (complete enumeration)

VII. Conclusion

In this work, a new methodology is proposed to solve a mixed categorical-continuous structural optimization problem. This methodology relies on the separation of the continuous variables from the categorical ones, that are treated in two different optimization problems. Compared to existing approaches like the hybrid branch and bound, the industrial approach $PRESTO$ and a complete enumeration, the computation cost seems promising, and similar optimal weights are obtained. Furthermore, the whole process could benefit from the parallelization of all optimizations performed during the overall iterations. This observed efficiency is due to the linearization of the Lagrangian of the sizing optimization problem, making the choices of each structural element independent from each other. However, in a future work it will be interesting to assess the drawbacks of this linearization. In addition, another way to compute the discrete Lagrangian slope, based on post-optimal sensitivity definition, is under investigation. Furthermore, it would be interesting to scale up the model and increase the design space dimension in order to check the behavior of the convergence. Finally, it has been observed that in some cases, this current formulation of the Bi-level algorithm retrieves a higher optimal weight than a complete enumeration. This is why another approach, based on the framework AMIEGO [24], is under investigation. The main idea of this framework is to leverage the efficiency of a gradient-based optimizer while globally exploring the integer design space with the help of the Efficient Global Optimization (EGO) [25] algorithm.

Appendix

In this first version of the bi-level approach, it is proposed to apply a relaxation on binary variables in problem (5). The relaxed binary variable is afterwards ensured to be discrete thanks to a particular formulation of the master optimization problem. Let be $C_{n,p}$ such as :

$$C_{n,p} = \left\{ X = (x_{ij})_{\substack{1 \leq i \leq n \\ 1 \leq j \leq p}} \Big/ \; x_{ij} \in [0,1], \text{ and } \sum_{j=1}^{p} x_{ij} = 1, \;\; \forall i \in [\![1, n]\!] \right\}$$

The overall problem (5) becomes, with relaxed binary variables U belonging to $C_{n,p}$:

$$\begin{aligned} \underset{U \in C_{n,p} \; (a,t) \in \mathbb{R}^{2n}}{\text{minimize}} \quad & \boldsymbol{w}(a,t,U) = \sum_{i=1}^{n} \sum_{j=1}^{p} U_{ij} \rho_j a_i L_i \\ \text{subject to} \quad & \boldsymbol{g}(a,t,U) \leq 0 \\ & \underline{a} \leq a \leq \bar{a} \\ & \underline{t} \leq t \leq \bar{t} \end{aligned} \tag{22}$$

It can be remarked that solving problem (5) is equivalent to solve problem (22) on the edges of $C_{n,p}$: Problem (5) is the restriction of problem (22) on $\mathcal{B}_{n,p}$.The constraint on displacements $\boldsymbol{d}$ is not considered in this first approach.

Now that the problem has been defined, the bi-level algorithm can be described. First, the variable U is decomposed into two variables. A variable α drives the local choices through the limit stress computation and the choice of density, while β drives the global effects through the choices involved in the internal loads computations. Both α and β belong to $C_{n,p}$. Let be $E \in \mathcal{M}_{p,f}$ a matrix of material properties. On each column are listed the p values of one material property. There are f material properties, including for example Young modulus and Poisson coefficient. Since the algorithm will ensure that α never takes intermediate values, only β can take intermediate values between 0 and 1. This is why β can not directly multiply the vector of choices γ, but the matrix of continuous properties E. An example of definition of E is given in Fig. 4.

$$E = \begin{pmatrix} e_1 & \nu_1 \\ e_2 & \nu_2 \\ e_3 & \nu_3 \\ e_4 & \nu_4 \end{pmatrix}$$

Fig. 4 Example of matrix of material properties, with $p = 4$ and $f = 2$. It provides the values of Young modulus (e) and Poisson coefficient (ν) in the same order than the list of choices γ.

The constraint formulation (4) is thus rewritten as:

$$g_{ij}(a,t,\alpha,\beta) = \sum_{k=1}^{p} \alpha_{ik} s_{ij}(a_i, t_i, \gamma_k, \Phi(a,t,\beta E)) \;\; \forall (i,j) \in [\![1,n]\!] \times [\![1,m]\!] \tag{23}$$

and the weight definition is given by :

$$\boldsymbol{w}(a,t,\alpha) = \sum_{i=1}^{n} \sum_{j=1}^{p} \alpha_{ij} \rho_j a_i L_i \tag{24}$$

The slave problem ($P_{a,t,\beta}$), consists in an optimization with respect to the sizing variables a and t, and the variable β. The variable α is fixed.

$$
\begin{aligned}
(P_{a,t,\beta}) \quad \text{Given } \alpha: \quad & \underset{(a,t)\in\mathbb{R}^{2n} \quad \beta\in C_{n,p}}{\text{minimize}} && \boldsymbol{w}(a,t,\alpha) \\
& \text{subject to} && \boldsymbol{g}(a,t,\alpha,\beta) \le 0 \\
& && \boldsymbol{d}(a,t,\alpha,\beta) \le 0 \\
& && \underline{a} \le a \le \bar{a} \\
& && \underline{t} \le t \le \bar{t} \\
& && \epsilon \le \alpha - \beta \le \epsilon
\end{aligned} \tag{25}
$$

This corresponds to a sizing step, where the choices of materials and shape are fixed. Let be $\mathcal{L}(\alpha)$ the value of the Lagrangian of the problem $P_{(a,t,\beta)}$, i.e.;

$$
\begin{aligned}
\mathcal{L}(\alpha) = \; & \boldsymbol{w}(a(\alpha),t(\alpha),\alpha) + \sum_{i=1}^{n}\sum_{j=1}^{m} \lambda_{ij}^{\boldsymbol{g}}(\alpha)\, \boldsymbol{g}_{ij}(a(\alpha),t(\alpha),\alpha,\beta(\alpha)) \\
& + \sum_{i=1}^{n} \left(\lambda_i^{ub_a}(\alpha)(a_i - \underline{a}_i) + \lambda_i^{lb_a}(\alpha)(\underline{a}_i - a_i) + \lambda_i^{ub_t}(\alpha)(t - \underline{t}_i) + \lambda_i^{lb_t}(\alpha)(\underline{t}_i - t_i) \right) \\
& + \sum_{i=1}^{n}\sum_{j=1}^{p} \left(\lambda_{ij}^{ub_{\alpha-\beta}}(\alpha)(\alpha_{ij} - \underline{\alpha}_{ij}) - \sum_{i=1}^{n} \lambda_i^{lb_{\alpha-\beta}}(\alpha)(\underline{\alpha}_{ij} - \alpha_{ij}) \right)
\end{aligned} \tag{26}
$$

Let be $\mathcal{L}^*$ the Lagrangian at the optimum of problem 25. A linearization of $\mathcal{L}^*$ around α^0 is given by:

$$
\hat{\mathcal{L}}^*(\alpha) = \mathcal{L}^*(\alpha^0) + \sum_{i=1}^{n}\sum_{j=1}^{p} \left[\frac{d\mathcal{L}^*(\alpha^0)}{d\alpha} \right]_{ij} (\alpha_{ij} - \alpha_{ij}^0) \tag{27}
$$

with :

- $\left[\frac{d\mathcal{L}^*(\alpha^0)}{d\alpha} \right]_{ij}$ the coefficient ij of the jacobian of the Lagrangian taken at α_0,
- α_{ij}^0 the relaxed binary design variable that drives the choice of the j^{th} catalog on the i^{th} element, at a given step 0 of the methodology.

At the master level, the Lagrangian (27) of $(P_{a,t,\beta})$ is minimized with respect to α :

$$
(P_\alpha) \quad \text{Given } a,t,\beta: \quad \underset{\alpha\in C_{n,p}}{\text{minimize}} \quad \hat{\mathcal{L}}^*(\alpha) \tag{28}
$$

Using equation (27), it can be demonstrated that solving problem (28) is equivalent to find, for each element i, the binary coefficient $\alpha_{ij} = 1$ such that the slope of the Lagrangian is minimal. The problem (28) becomes :

$$
\begin{aligned}
& (\forall i \in [\![1,n]\!]), \; l_i = \underset{j\in[\![1,p]\!]}{\text{argmin}} \left[\frac{d\mathcal{L}^*(\alpha^0)}{d\alpha} \right]_{ij} \\
& \text{with } \alpha^* \text{ given by } \alpha_{ij}^* = \begin{cases} 1 & \text{if } \; j = l_i \; (\forall i \in [\![1,n]\!]) \\ 0 & \text{else} \end{cases}
\end{aligned} \tag{29}
$$

The slope of the Lagrangian $\left[\frac{d\mathcal{L}^*(\alpha^0)}{d\alpha} \right]_{ij}$ taken at α_0 can be computed using post-optimal sensitivity definition of problem (25), i.e.;

$$
\begin{aligned}
\frac{d\mathcal{L}^*(\alpha^0)}{d\alpha_{ij}} = & \sum_{k=1}^{n} \frac{\partial\mathcal{L}^*(\alpha^0)}{\partial a_k}\frac{da_k}{d\alpha_{ij}} + \sum_{k=1}^{n} \frac{\partial\mathcal{L}^*(\alpha^0)}{\partial t_k}\frac{dt_k}{d\alpha_{ij}} + \sum_{k=1}^{n} \frac{\partial\mathcal{L}^*(\alpha^0)}{\partial \beta_k}\frac{d\beta_k}{d\alpha_{ij}} + \sum_{k=1}^{n}\sum_{l=1}^{m} \frac{\partial\mathcal{L}^*(\alpha^0)}{\partial \lambda_{kl}^{\boldsymbol{g}}}\frac{d\lambda_{kl}^{\boldsymbol{g}}}{d\alpha_{ij}} + \\
& \frac{\partial\mathcal{L}^*(\alpha^0)}{\partial \lambda^{\boldsymbol{d}}}\frac{d\lambda^{\boldsymbol{d}}}{d\alpha_{ij}} + \frac{\partial\mathcal{L}^*(\alpha^0)}{\partial \alpha_{ij}}
\end{aligned} \tag{30}
$$

Furthermore,

- the KKT conditions of problem (25) give

$$\forall i \in [\![1,n]\!] \quad \frac{\partial \mathcal{L}^*(\alpha^0)}{\partial a_i} = 0, \quad \frac{\partial \mathcal{L}^*(\alpha^0)}{\partial t_i} = 0, \quad \frac{\partial \mathcal{L}^*(\alpha^0)}{\partial \beta_i} = 0 \tag{31}$$

- under hypothesis of Fiacco's theorem, $a^*(\alpha)$, $t^*(\alpha)$ and $\beta^*(\alpha)$ are local solutions in the vicinity of α. Thus, a strong assumption is made: in a neighborhood of α the set of active constraints $\mathcal{A}(\alpha)$ does not change, in particular no inactive constraint is activated (or the contrary) with a change of α :

$$\forall (i,j) \in [\![1,n]\!] \times [\![1,p]\!], \ \forall (k,l) \in [\![1,n]\!] \times [\![1,m]\!] \quad \frac{d\lambda^{\boldsymbol{g}}_{kj}}{d\alpha_{ij}} = 0 \tag{32}$$

Thus, under hypothesis of Fiacco's theorem, the expression 30 becomes:

$$\frac{d\mathcal{L}^*(\alpha^0)}{d\alpha_{ij}} = \frac{\partial \mathcal{L}^*(\alpha^0)}{\partial \alpha_{ij}} \tag{33}$$

and the derivative of the Lagrangian, at the optimum, is:

$$\begin{aligned}
\frac{d\mathcal{L}^*(\alpha^0)}{d\alpha_{ij}} &= \frac{\partial \boldsymbol{w}(a^*(\alpha^0), t^*(\alpha^0), \alpha^0)}{\partial \alpha_{ij}} + \sum_{k=1}^{n}\sum_{l=1}^{m} \lambda^{\boldsymbol{g}}_{kl} \frac{\partial \boldsymbol{g}^*_{kl}(a(\alpha^0), t(\alpha^0), \alpha^0, \beta(\alpha^0))}{\partial \alpha_{ij}} - \lambda^{lb_{\alpha-\beta}}_{ij}(\alpha^0) + \lambda^{ub_{\alpha-\beta}}_{ij}(\alpha^0) \\
&= \rho_j a^*_i(\alpha^0) L_i + \sum_{k=1}^{n}\sum_{l=1}^{m} \lambda^{\boldsymbol{g}}_{kl} s_{kj}(a^*_k(\alpha^0), t^*_k(\alpha^0), \gamma_j, \Phi(a^*(\alpha^0), t^*(\alpha^0), \beta^*(\alpha^0)\ E)) - \lambda^{lb_{\alpha-\beta}}_{ij}(\alpha^0) + \lambda^{ub_{\alpha-\beta}}_{ij}(\alpha^0)
\end{aligned} \tag{34}$$

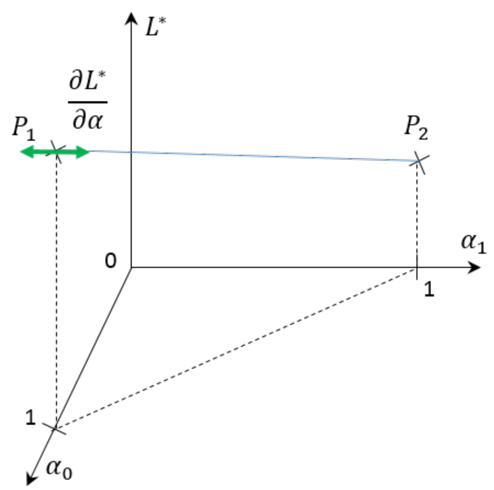
The algorithm procedure is described in Algorithm 2:

Algorithm 2 Bi-level algorithm with post-optimal sensitivities

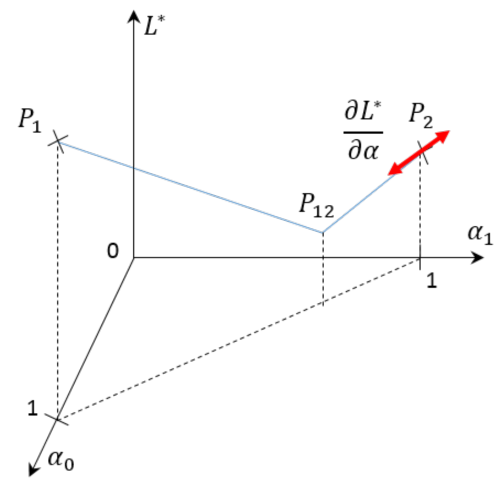
$\alpha^{(0)} \leftarrow$ initialize α ▷ initialize α
$w^{(0)}, t^{(0)}, a^{(0)}, \beta^{(0)} \leftarrow$ solve $(P_{a,t,\beta})\big|_{\alpha^{(0)}}$
$k \leftarrow 1$
while $|w^{(k)} - w^{(k-1)}| < \delta$ **do** ▷ check convergence on weight
  compute $\dfrac{\partial \mathcal{L}^*(\alpha^{(k)})}{\partial \alpha}$
  for each bar i from 1 to n **do**
    $l_i = \underset{j \in [\![1,p]\!]}{\operatorname{argmin}} \left[\dfrac{\partial \mathcal{L}(\alpha^{(k)})}{\partial \alpha} \right]_{ij}$
    $\alpha^{(k)} \leftarrow \alpha^{(k-1)} + \Delta\alpha^{il}$ ▷ build binary solution
  $w^{(k)}, t^{(k)}, a^{(k)}, \beta^{(k)} \leftarrow$ solve $(P_{a,t,\beta})\big|_{\alpha^{(k)}}$
  $k \leftarrow k+1$
  $\epsilon^k \leftarrow \epsilon^{k-1}/2$ ▷ Decrease the tolerancy between α and β
return w^*, t^*, a^*, α^* ▷ returns optimal weight, thicknesses, areas, and catalogs

Thus in this algorithm, although α belongs to $C_{n,p}$, the linearization of the Lagrangian leads to pure binary values. Theoretically, α belongs to $C_{n,p}$, but the algorithm (in particular problem (29)) ensures that α coefficients are such that α belongs to $\mathcal{B}_{n,p}$. Later in the process, when problem (25) is solved, the relaxed variable β is forced to stay in a vicinity ϵ around α. This vicinity ϵ is decreased iteration per iteration, so that at the end $\alpha = \beta$, with $(\alpha, \beta) \in \mathcal{B}^2_{n,p}$.

However, numerical experiments show that the assumptions leading to equation (33) are not verified. Indeed, in practice, a change of the binary variable α induces a change in the set of active constraints. This is responsible of a non-linearity as illustrated in Fig. 5b.



(a) The Lagrangian is linear w.r.t. α and thus the post-optimal sensitivity returns a correct Lagrangian slope.



(b) The misleading post-optimal sensitivity computed in P_2 makes the algorithm choose $\alpha_0 = 1$, however the optimal weight in P_1 is greater than the one in P_2. A change of constraints is responsible of a non-linearity in P_{12}.

Fig. 5 Optimization with $n = 1$ and $p = 2$. This simplified case illustrates the importance of the behaviour of the Lagrangian between two solutions P_1 and P_2. The post-optimal sensitivity can lead to wrong choices if, for example, there is a change in the set of active constraints (in Fig. 5b). In both situations, the optimal solution is $\alpha_1 = 1$ (and thus $\alpha_0 = 0$).

Acknowledgments

This work is part of the MDA-MDO project of the French Institute of Technology IRT Saint Exupery. We wish to acknowledge the PIA framework (CGI, ANR) and the project industrial members for their support, financial backing and/or own knowledge: Airbus, Airbus Group Innovations, SOGETI High Tech, Altran Technologies, CERFACS.

References

[1] Barjhoux, P.-J., Diouane, Y., Grihon, S., Bettebghor, D., and Morlier, J., "Mixed Variable Structural Optimization: toward an Efficient Hybrid Algorithm," *WCSMO12*, 2017.

[2] Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Vol. Addison-We, 1989.

[3] Storn, R., and Price, K., "Differential Evolution A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *Journal of Global Optimization*, Vol. 11, No. 4, 1997, pp. 341–359.

[4] Kennedy, J., Eberhart, R. C., and Shi, Y., "Swarm Intelligence," *Scholarpedia*, Vol. 2, No. 9, 2001, p. 1462.

[5] Liao, T., Socha, K., Montes de Oca, M. A., Stutzle, T., and Dorigo, M., "Ant Colony Optimization for Mixed-Variable Optimization Problems," *IEEE Transactions on Evolutionary Computation*, Vol. 18, No. 4, 2014, pp. 503–518.

[6] Torczon, V., "On the convergence of pattern search algorithms," *SIAM Journal on Optimization*, Vol. 7, 1997, pp. 1–25.

[7] Lampinen, J. A., and Zelinka, I., "Mixed Integer-Discrete-Continuous Optimization, by Differential Evolution, Part 1: the optimization method," *Proceedings of MENDEL'99, 5th International Mendel Conference on Soft Computing*, 1999, pp. 71–76.

[8] Guo, C.-X., Hu, J.-S., Ye, B., and Cao, Y.-J., "Swarm intelligence for mixed-variable design optimization." *Journal of Zhejiang University. Science*, Vol. 5, No. 7, 2004, pp. 851–60.

[9] Dimopoulos, G. G., "Mixed-variable engineering optimization based on evolutionary and social metaphors," *Computer Methods in Applied Mechanics and Engineering*, Vol. 196, No. 4-6, 2007, pp. 803–817.

[10] Deb, K., and Goyal, M., "A Flexible Optimization Procedure for Mechanical Component Design Based on Genetic Adaptive Search," *Journal of Mechanical Design*, Vol. 120, No. 2, 1998, p. 162.

[11] Audet, C., and Dennis, J. E., "Pattern Search Algorithms for Mixed Variable Programming," *SIAM Journal on Optimization*, Vol. 11, No. 3, 2001, pp. 573–594.

[12] Rakshit, S., and Ananthasuresh, G. K., "Simultaneous material selection and geometry design of statically determinate trusses using continuous optimization," *Structural and Multidisciplinary Optimization*, Vol. 35, No. 1, 2008, pp. 55–68.

[13] Herrera, M., Guglielmetti, A., Xiao, M., and Filomeno Coelho, R., "Metamodel-assisted optimization based on multiple kernel regression for mixed variables," *Structural and Multidisciplinary Optimization*, Vol. 49, No. 6, 2014, pp. 979–991.

[14] Geoffrion, A. M., "Generalized Benders decomposition," *Journal of Optimization Theory and Applications*, Vol. 10, No. 4, 1972, pp. 237–260.

[15] Duran, M. A., and Grossmann, I. E., "An outer-approximation algorithm for a class of mixed-integer nonlinear programs," *Mathematical Programming*, Vol. 36, No. 3, 1986, pp. 307–339.

[16] Haftka, R. T., Watson, L. T., Haftka, R. T., and Watson, L. T., "Decomposition theory for multidisciplinary design optimization problems with mixed integer quasiseparable subsystems," *Optim Eng*, Vol. 7, No. 7, 2006, pp. 135–149.

[17] Schutte, J. F., Haftka, R. T., and Watson, L. T., "Decomposition and Two-level Optimization of Structures with Discrete Sizing Variables," *Constraints*, Vol. 0, No. 1, 2004, pp. 1–5.

[18] Grihon, S., "PRESTO: A rapid sizing tool for airframe conceptual design studies," *LMS European Aeronautical Conference*, Toulouse, 2012.

[19] Grihon, S., "Structure Sizing Optimization Capabilities at AIRBUS," *WCSMO12*, 2017.

[20] Merval, A., "L'Optimisation Multiniveaux D'une Structure," *PhD Thesis*, 2008.

[21] Samuelides, M., Bettebghor, D., Grihon, S., Merval, A., and Morlier, J., "Modèles réduits en optimisation multiniveau de structures aéronautiques," *9e Colloque national en calcul des structures*, CSMA, Giens, France, 2009.

[22] Gallard, F., Vanaret, C., Guenot, D., Gachelin, V., Lafage, R., Pauwels, B., Barjhoux, P.-J., and Gazaix, A., "GEMS: A Python Library for Automation of Multidisciplinary Design Optimization Process Generation," *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2018.

[23] Svanberg, K., "A class of globally convergent optimization methods based on conservative convex separable approximations," *SIAM Journal on Optimization*, Vol. 12, No. 2, 2002, pp. 555–573.

[24] Roy, S., Crossley, W. A., Moore, K. T., Gray, J. S., and Martins, J., "Next generation aircraft design considering airline operations and economics," *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2018.

[25] Jones, D. R., Schonlau, M., and Welch, W. J., "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, Vol. 13, No. 4, 1998, pp. 455–492.