Proceedings of the Fifteenth International Conference on
Computational Structures Technology
Edited by: P. Iványi, J. Kruis and B.H.V. Topping
Civil-Comp Conferences, Volume 9, Paper 10.4
Civil-Comp Press, Edinburgh, United Kingdom, 2024
ISSN: 2753-3239, doi: 10.4203/ccc.9.10.4
©Civil-Comp Ltd, Edinburgh, UK, 2024

Size Optimisation of 2D Frame Structures using Inexact Restoration

T.Světlík,[1] M.Mrovec,[2] L.Pospíšil[1]and M.Cermak[1]

[1]Department of Mathematics,
VSB – Technical University of Ostrava, Czech Republic
[2]School of Chemistry, University of Sydney,
Camperdown, Australia

Abstract

In this paper, we focus on the size optimization of frame structures with the aim of minimizing the structural weight under the necessary conditions for serviceability and usability, such as deformation and stress constraints. Using the Finite Element Method, this approach leads to the area of nonlinear programming with a nonlinear cost function as well as nonlinear inequality constraints. We tackle this issue using Inexact Restoration (IR). Unlike traditional optimization methods, the IR algorithm separates the feasibility restoration from the objective function improvement, allowing for a more efficient search for the optimal solution. This method is particularly advantageous in dealing with the complex, non-linear behavior of beam structures with various constraints. We review the theory, present our implementation and compare results using a benchmark.

Keywords: frame structure, structural optimisation, finite element method, inexact restoration, gradient method, nonlinear programming

1 Introduction

The optimization of building structures has been a widely extended topic dating back to the beginning of the last century and remains one of the most widespread topics not only in structural mechanics [7]. The current direction of the issue closely focuses on the optimization of the topology of 2D and 3D structures [16], aiming to remove excess material from a predefined area or volume so that the boundary conditions of the task are met. These conditions, in most cases, focus on maximizing the stiffness of the structure while ensuring force equilibrium and keeping the maximum volume of the structure bellow predefined level [14]. A less widespread, but for practical application much more attractive direction of optimization of building structures, is moving towards the size optimization of the building elements, which is mainly extended for beam structures [5]. The main difference compared to topology optimization is the predefined geometry of the structure with unknown defining dimensions of the cross-sections of individual beams. The optimization criterion is then to minimize the weight of the structure while meeting the equilibrium of force balance and other constraints related to deformations, stress, buckling capacity, and others [13], [9]. In truss structures, deformations and stresses are solely determined by the cross-sectional area, which means that size optimisation with stress constraints results in the use of Sequential Linear Programming (SLP) [15]. However, in frame structures, both the cross-sectional area and the moment of inertia influence deformations and stresses. This complexity turns the optimization of general cross-sectional shapes into a nonlinear optimization problem, characterized by nonlinear inequality constraints.

For solving size optimization tasks, two directions are currently extended, namely solutions using gradient methods and evolutionary algorithms. Gradient methods such as Sequential Quadratic Programming (SQP) [3] and the Method of Moving Asymptotes (MMA) [12] are based on replacing the original optimization task with a locally approximated one of simpler formulation, whose solution approaches the optimum of the original task. These approaches typically only lead to a local maximum or minimum, which might not align with the overall best solution for the given problem. However, they offer the advantage of faster convergence by leveraging gradient information. On the other hand, evolutionary algorithms like Differential Evolution (DE) [1], or Particle Swarm Optimization (PSO) [6] are metaheuristic algorithms that uses purely the information of the function values and are used for their ability to find a global optimum. A significant limitation of these algorithms is their need for numerous function evaluations. This problem is especially pronounced in structural optimization tasks that employ the finite element method, as they require the computation of the deformation vector through the resolution of a system of linear equations. The evaluation of the cost function is directly tied to the time-consuming process of solving these equations. When combined with the high frequency of evaluations, this significantly escalates the computational effort, thus limiting the method's applicability in practical scenarios. In this paper we focus on gradient method of Inexact Restoration (IR) [8], which is particularly designed for engineering problems characterized by a multitude of inequality constraints, ensuring the solutions retain their physical significance.

The paper is organized as follows: We begin with a brief review of the mathematical background; see Section 2 for the finite element method and Section 3 for the size optimization problem. The main part of the paper, Section 4, introduces the Inexact Restoration method for the size optimization problem. Section 5 presents the considered benchmarks along with our numerical results. The concluding Section 6 summarizes the paper and outlines our future work.

2 Structure weight optimisation

The goal of structural optimization is to minimize the weight of the structure while ensuring structural integrity and serviceability. A common approach is to use the Finite Element Method (FEM) [2] to solve the discretized form of potential elastic energy, leading to a minimization problem

$$\min_{\mathbf{r}} \quad \frac{1}{2}\mathbf{r}^T\mathbf{K}\mathbf{r} - \mathbf{r}^T\mathbf{f}$$

which is equivalent to the solution of a system of linear equations

$$\mathbf{K}\mathbf{r} = \mathbf{f}, \tag{1}$$

where $\mathbf{K} \in \mathbb{R}^{n_d \times n_d}$ is a symmetric, positive definite stiffness matrix, $\mathbf{f} \in \mathbb{R}^{n_d}$ is the load vector, $\mathbf{r} \in \mathbb{R}^{n_d}$ is the vector of deformations, and n_d is the degree of freedom of the structure. The stiffness matrix is composed of the contributions from the stiffness matrices of individual elements

$$\mathbf{K} = \sum_{i=1}^{n_e} \mathbf{L}_i^T \mathbf{K}_{\mathbf{e},i} \mathbf{L}_i,$$

with $\mathbf{L}_i \in \mathbb{R}^{6 \times n_d}$ being the allocation matrix that maps the entries of $\mathbf{K}_{\mathbf{e},i}$ to their global positions in $\mathbf{K}$, $\mathbf{K}_{\mathbf{e},i}$ is a local stiffness matrix of the element, and n_e is the number of elements. In this paper, we consider $\mathbf{K}_{\mathbf{e},i} \in \mathbb{R}^{6\times 6}$ to be the elastic stiffness matrix of a 2D Euler beam element

$$\mathbf{K}_{\mathbf{e},i} = \begin{bmatrix} \frac{E_iA_i}{l_i} & 0 & 0 & -\frac{E_iA_i}{l_i} & 0 & 0 \\ 0 & \frac{12E_iI_{y,i}}{l_i^3} & \frac{6E_iI_{y,i}}{l_i^2} & 0 & -\frac{12E_iI_{y,i}}{l_i^3} & \frac{6E_iI_{y,i}}{l_i^2} \\ 0 & \frac{6E_iI_{y,i}}{l_i^2} & \frac{4E_iI_{y,i}}{l_i} & 0 & -\frac{6E_iI_{y,i}}{l_i^2} & \frac{2E_iI_{y,i}}{l_i} \\ -\frac{E_iA_i}{l_i} & 0 & 0 & \frac{E_iA_i}{l_i} & 0 & 0 \\ 0 & -\frac{12E_iI_{y,i}}{l_i^3} & -\frac{6E_iI_{y,i}}{l_i^2} & 0 & \frac{12E_iI_{y,i}}{l_i^3} & -\frac{6E_iI_{y,i}}{l_i^2} \\ 0 & \frac{6E_iI_{y,i}}{l_i^2} & \frac{2E_iI_{y,i}}{l_i} & 0 & -\frac{6E_iI_{y,i}}{l_i^2} & \frac{4E_iI_{y,i}}{l_i} \end{bmatrix},$$

where E_i is Young's modulus, l_i is the length of the element, A_i is the area of the cross-section, and $I_{y,i}$ is the moment of inertia of the cross-section, which may vary for each element $i \in \{1, 2, \ldots, n_e\}$.

Given the nature of the Euler stiffness matrix, it is possible to decompose $\mathbf{K}_{\mathbf{e},i}$ into a sum of two separate matrices corresponding to axial and bending stiffness, denoted as stiffness $\mathbf{K_T} \in \mathbb{R}^{6\times 6}$ and $\mathbf{K_E} \in \mathbb{R}^{6\times 6}$, respectively

$$\mathbf{K}_{\mathbf{e},i} = A_i \mathbf{K}_{\mathbf{T},i} + I_{y,i} \mathbf{K}_{\mathbf{E},i}$$

$$\mathbf{K}_{\mathbf{E},i} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \dfrac{12E_i}{l_i^3} & \dfrac{6E_i}{l_i^2} & 0 & -\dfrac{12E_i}{l_i^3} & \dfrac{6E_i}{l_i^2} \\ 0 & \dfrac{6E_i}{l_i^2} & \dfrac{4E_i}{l_i} & 0 & -\dfrac{6E_i}{l_i^2} & \dfrac{2E_i}{l_i} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\dfrac{12E_i}{l_i^3} & -\dfrac{6E_i}{l_i^2} & 0 & \dfrac{12E_i}{l_i^3} & -\dfrac{6}{l_i^2} \\ 0 & \dfrac{6E_i}{l_i^2} & \dfrac{2E_i}{l_i} & 0 & -\dfrac{6E_i}{l_i^2} & \dfrac{4E_i}{l_i} \end{bmatrix},$$

$$\mathbf{K}_{\mathbf{T},i} = \begin{bmatrix} \dfrac{E_i}{l_i} & 0 & 0 & -\dfrac{E_i}{l_i} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\dfrac{E_i}{l_i} & 0 & 0 & \dfrac{E_i}{l_i} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Notice that for fixed lengths of the elements l_i and Young's modulus E_i, matrices $\mathbf{K}_{\mathbf{T},i}$ and $\mathbf{K}_{\mathbf{E},i}$ are constant for each element, and the $\mathbf{K}_{\mathbf{e},i}$ depends purely on the cross-sectional parameters $A_i, I_{y,i}$ of each element. This property is especially useful for size optimization and will be exploited in Section 4.

3 Size optimisation problem

Size optimization is a specific type of structural optimization in which the geometry of the structure is fixed, but its individual parts are subject to optimization. In the context of beam structures, size optimization involves optimizing the cross-sectional dimensions of each beam element. The goal of the optimization is to find the mass distribution that minimizes the weight of the structure while maintaining the serviceability and ultimate limit states, such as maximal deformations, internal forces, and stresses. Let us define a size optimization problem as the minimization of the structure's weight:

$$m(\mathbf{a}) = \sum_{i=1}^{n_e} A_i(\mathbf{a}) L_i \rho_i$$

This is done while maintaining force equilibrium:

$$\begin{aligned} &\mathbf{K}(\mathbf{a})\mathbf{r} = \mathbf{f} \\ &\mathbf{a}_{min} \leq \mathbf{a} \end{aligned} \tag{2}$$

and ensuring the structure meets serviceability and ultimate limit states:

$$\hat{\mathbf{c}}(\mathbf{r}, \mathbf{a}) \leq \mathbf{0}$$

Thus, the optimization problem can be formulated as:

$$\begin{aligned} \min_{\mathbf{r},\mathbf{a}} \quad & m(\mathbf{a}) \\ \text{s.t.} \quad & \mathbf{K}(\mathbf{a})\mathbf{r} = \mathbf{f} \\ & \hat{\mathbf{c}}(\mathbf{r}, \mathbf{a}) \leq \mathbf{0} \\ & \mathbf{a}_{min} \leq \mathbf{a} \end{aligned} \tag{3}$$

The $m(\mathbf{a})$ is a simple summation of the masses of each element, where $\rho_i > 0$ is the material density of an element, and $A(\mathbf{a})_i > 0$ is the cross-sectional area of an element, dependent on the geometric parameters of the cross-section, $\mathbf{a} \in \mathbb{R}^{n_p}$.

The expression (2) can be understood as an extension of equation (1), where the stiffness matrix $\mathbf{K}(\mathbf{a})$ is now a function of the cross-sectional parameters

$$\mathbf{K}(\mathbf{a}) = \sum_{i=1}^{n_e} \mathbf{L}_i^T \mathbf{K}_{\mathbf{e},i}(\mathbf{a}) \mathbf{L}_i,$$

$$\mathbf{K}_{\mathbf{e},i}(\mathbf{a}) = A_i(\mathbf{a})\mathbf{K}_{\mathbf{T},i} + I_{y,i}(\mathbf{a})\mathbf{K}_{\mathbf{E},i}.$$

These parameters are constrained by their minimum values, $\mathbf{a}_{min} \in \mathbb{R}^{n_p}$. This ensures the physical validity of each geometrical parameter, such as preventing a rectangle from having negative side lengths. From a mathematical perspective, these constraints also ensure the positive definiteness of $\mathbf{K}(\mathbf{a})$. The number of parameters, n_p, depends on the parametric equations which define the cross-sections of individual elements, and this may vary from one element to another. For example, the number of geometric parameters is only one for square and circular cross-sections, two for rectangular and pipe cross-sections, and four to six for I-shaped cross-sections.

Lastly, $\hat{\mathbf{c}}(\mathbf{r}, \mathbf{a})$ represents the set of inequality constraints that ensure the structure's serviceability and ultimate limit states, such as maximal deformations, strains, and stresses. The number of inequality constraints depends on the applications and may vary. In this paper, we consider deformation and stress constraints as

$$\hat{\mathbf{c}}(\mathbf{r}, \mathbf{a}) = \begin{bmatrix} \mathbf{r} - \mathbf{r}_{max} \\ \mathbf{B}_\sigma(\mathbf{a})\mathbf{r} - \boldsymbol{\sigma}_{max} \end{bmatrix}$$

where $\mathbf{r}_{max} \in \mathbb{R}^{n_d}$ is the vector of limit deformations, $\mathbf{B}_\sigma(\mathbf{a}) \in \mathbb{R}^{n_\sigma \times n_p}$ is the matrix mapping the deformations to element stresses dependent on geometrical parameters $\mathbf{a}$, and $\boldsymbol{\sigma}_{max} \in \mathbb{R}^{n\sigma}$ is the vector of limit stresses. Here, n_σ indicates the count of stress constraints, leading to a total constraint count of $n_c = n_d + n_\sigma$.

Since the stiffness matrix $\mathbf{K}(\mathbf{a})$ is positive definite, the deformation vector has a unique solution

$$\mathbf{r}(\mathbf{a}) = \mathbf{K}(\mathbf{a})^{-1}\mathbf{f},$$

which, when substituted back into the minimisation (3), we obtain a nonlinear optimisation problem with nonlinear inequality constraints of purely geometric parameters

$$\begin{aligned} \min_{\mathbf{a}} \quad & m(\mathbf{a}) \\ \text{s.t.} \quad & \mathbf{a} \in \boldsymbol{\Omega} \end{aligned}$$

$$\Omega = \{\mathbf{a} \in \mathbb{R}^{n_p} \mid \mathbf{c}(\mathbf{a}) \leq \mathbf{0}, \mathbf{a}_{min} \leq \mathbf{a}\}$$

with a set of inequality constraints

$$\mathbf{c}(\mathbf{a}) = \begin{bmatrix} \mathbf{r}(\mathbf{a}) - \mathbf{r}_{max} \\ \mathbf{B}_\sigma(\mathbf{a})\mathbf{r}(\mathbf{a}) - \boldsymbol{\sigma}_{max} \end{bmatrix}.$$

4 Inexact restoration

An inexact restoration algorithm is designed to solve optimization problems characterized by nonlinear cost functions and inequality constraints. The prerequisites for applying this algorithm include the smoothness of the cost function and inequality constraints, as well as the convexity and compactness of the feasible set of optimized variables.

The algorithm operates through two alternating phases: restoration and minimization. Initially, a starting point, denoted by $\mathbf{a}_\mathbf{x}$, is selected. From there, the process advances to the restoration phase.

4.1 Restoration

In the restoration step, the objective is to find a point $\mathbf{a}_\mathbf{y}$ that is more feasible than the starting point $\mathbf{a}_\mathbf{x}$ in terms of constraint violations

$$h(\mathbf{a}_\mathbf{y}) \leq p \cdot h(\mathbf{a}_\mathbf{x}), \tag{4}$$

where $h(\mathbf{a})$ is a constraint violation function and $p \in (0, 1)$ is an algorithm-specific constant that ensures a reduction in constraint violations. To simplify the notation, let us denote $c_j(\mathbf{a})$ as the j-th constraint of $\mathbf{c}(\mathbf{a})$ and the constraint violation function as

$$h(\mathbf{a}) = |\mathbf{c}^+(\mathbf{a})|_2$$

$$\mathbf{c}^+(\mathbf{a}) = \begin{bmatrix} \max(c_1(\mathbf{a}), 0) \\ \vdots \\ \max(c_{n_c}(\mathbf{a}), 0) \end{bmatrix}$$

where $|.|_2$ denotes the Euclidean norm and $\max(c_j(\mathbf{a}), 0)$ acts as a filter that returns only the violated constraints. It's important to note that the required inequality is of the form $\mathbf{c}(\mathbf{a}) \leq \mathbf{0}$. Thus, function values below zero, which satisfy the constraints, are mapped to zero, and only the positive function values, indicating constraint violations, are considered.

To find $\mathbf{a_y}$, we compute the gradient of the constrain violation function $h(\mathbf{a})$ and perform a line search along the descent direction, transforming the inequality (4) into one dimensional problem of finding the multiple of gradient $\alpha \in \mathbb{R}$ such that

To find $\mathbf{a_y}$, we compute the gradient of the constraint violation function $h(\mathbf{a})$ and perform a line search along the descent direction, transforming the inequality (4) into a one-dimensional problem of finding the multiple of the gradient $\alpha \in \mathbb{R}$ such that

$$h(\mathbf{a_x} - \alpha \nabla h(\mathbf{a_x})) \leq p \cdot h(\mathbf{a_x}),$$

where $\nabla h(\mathbf{a_x})$ is the gradient of violated constraints at $\mathbf{a_x}$. It turns out that one step of the Newton method [10] in the direction of $\nabla h(\mathbf{a_x})$ provides a sufficient decrease in the constraint violation function with an explicit solution for α

$$\alpha = \frac{h(\mathbf{a_x})}{\nabla h(\mathbf{a_x})^T \nabla h(\mathbf{a_x})}$$

and the new point $\mathbf{a_y}$

$$\mathbf{a_y} = \mathbf{a_x} - \alpha \nabla h(\mathbf{a_x}), \tag{5}$$

which not only eliminates the need to define the parameter p in inequality (4) but is also computationally efficient since both $h(\mathbf{a_x})$ and $\nabla h(\mathbf{a_x})$ are already known at this stage. The only missing part is an evaluation of the gradient of violated constraints [11]

$$\nabla h(\mathbf{a}) = \frac{1}{h(\mathbf{a})} \nabla \mathbf{c}(\mathbf{a})^T \mathbf{c}^+(\mathbf{a}),$$

where $\nabla \mathbf{c}(\mathbf{a}) \in \mathbb{R}^{n_c \times n_p}$ is a gradient of inequality constraints

$$\nabla \mathbf{c}(\mathbf{a}) = \begin{bmatrix} \nabla \mathbf{r}(\mathbf{a}) \\ \nabla(\mathbf{B}_\sigma(\mathbf{a})\mathbf{r}(\mathbf{a})) \end{bmatrix}$$

composed of the gradient of the deformation vector $\nabla \mathbf{r}(\mathbf{a})$ and the gradient of element stresses $\nabla(\mathbf{B}_\sigma(\mathbf{a})\mathbf{r}(\mathbf{a}))$. It turns out that the gradient of the deformation vector $\mathbf{r}(\mathbf{a})$ leads to a system of linear equations

$$\nabla \mathbf{r}(\mathbf{a}) = \mathbf{K}(\mathbf{a})^{-1} \nabla \mathbf{F}(\mathbf{a})$$

$$\nabla \mathbf{F}(\mathbf{a}) = \left[\frac{\partial \mathbf{K}(\mathbf{a})}{\partial a_1} \mathbf{r}(\mathbf{a}), \ \frac{\partial \mathbf{K}(\mathbf{a})}{\partial a_2} \mathbf{r}(\mathbf{a}), \ \cdots, \ \frac{\partial \mathbf{K}(\mathbf{a})}{\partial a_{n_p}} \mathbf{r}(\mathbf{a}) \right],$$

where $\dfrac{\partial \mathbf{K}(\mathbf{a})}{\partial a_k}$ is the partial derivative of the stiffness matrix. Since the $\mathbf{K}(\mathbf{a})$ is positive definite, the deformation gradient $\nabla \mathbf{r}(\mathbf{a})$ leads to the same system of linear equations as (1) with multiple right-hand sides. To compute the k-th partial derivatives of the stiffness matrix, where $k \in \{1, 2, \ldots, n_p\}$, we derive from the sum relation of the stiffness matrix $\mathbf{K}(\mathbf{a})$ and element stiffness matrix decomposition $\mathbf{K}_{\mathbf{e},i}(\mathbf{a})$. Since the cross-section parameters in $\mathbf{K}_{\mathbf{e},i}(\mathbf{a})$ act as a scalar multipliers, we get

$$\frac{\partial \mathbf{K}(\mathbf{a})}{\partial a_k} = \sum_{i=1}^{n_e} \mathbf{L}_i^T \left(\frac{\partial A_i(\mathbf{a})}{\partial a_k} \mathbf{K}_{\mathbf{T},i} + \frac{\partial I_{y,i}(\mathbf{a})}{\partial a_k} \mathbf{K}_{\mathbf{E},i} \right) \mathbf{L}_i,$$

which, with pre-computed $\mathbf{K}_{\mathbf{T},i}, \mathbf{K}_{\mathbf{E},i}$, depends on the derivation of the cross-section area and moment of inertia that typically results in the derivation of a polynomial. In the case of the gradient of element stress $\nabla(\mathbf{B}_\sigma(\mathbf{a})\mathbf{r}(\mathbf{a}))$, evaluation is slightly more complex than the gradient of the deformation vector, but using simple chain rule we are able to derive an explicit solution

$$\nabla(\mathbf{B}_\sigma(\mathbf{a})\mathbf{r}(\mathbf{a})) = \nabla \mathbf{B}_\sigma(\mathbf{a})\mathbf{r}(\mathbf{a}) + \mathbf{B}_\sigma(\mathbf{a})\nabla \mathbf{r}(\mathbf{a}),$$

where $\nabla \mathbf{B}_\sigma(\mathbf{a})$ is the gradient of the stress mapping matrix. Thus, while constraining deformations, it is rather simple to add an additional stress constraint. With the gradient of the constraint violation function, we are then able to find the point $\mathbf{a}_\mathbf{y}$ using (5) and proceed to the minimisation step.

4.2 Minimisation step

With the new $\mathbf{a}_\mathbf{y}$, we compute the modified linearly approximated feasible set of inequality constraints by evaluating the constraints $\mathbf{c}(\mathbf{a})$ at $\mathbf{a}_\mathbf{y}$ while keeping the gradient of constraints $\nabla \mathbf{c}(\mathbf{a})$ from the minimization step, which was evaluated at $\mathbf{a}_\mathbf{x}$. The modified linearly approximated feasible set of inequality constraints is then given by

$$\pi = \left\{ \mathbf{a} \in \mathbb{R}^{n_p} \mid c_j(\mathbf{a}_\mathbf{y}) + \nabla c_j(\mathbf{a}_\mathbf{x})(\mathbf{a} - \mathbf{a}_\mathbf{y}) \leq c_j^+(\mathbf{a}_\mathbf{y}) \right\}.$$

This simple modification removes the requirement for the second evaluation of constraint gradients, the most expensive part of the algorithm to only one in each outer loop iteration. Next we define a trust region with radius $\delta > 0$

$$\mathbb{T} = \{ \mathbf{a} \in \mathbb{R}^{n_p} \mid \|\mathbf{a} - \mathbf{a}_\mathbf{y}\|_2 \leq \delta \}.$$

In the intersection of these two sets, we minimize objective function $m(\mathbf{a})$, i.e

$$\begin{aligned} \min_{\mathbf{a}} \quad & m(\mathbf{a}) \\ \text{s.t.} \quad & \mathbf{a} \in \pi \cap \mathbb{T}. \end{aligned} \tag{6}$$

To minimize (6), we compute a projected gradient of $m(\mathbf{a})$ onto the approximated feasible set

$$\mathbf{d} = P(\mathbf{a}_\mathbf{y} - \eta \nabla m(\mathbf{a}_\mathbf{y})) - \mathbf{a}_\mathbf{y},$$

where $\nabla m(\mathbf{a_y})$ is the gradient of $m(\mathbf{a})$, $\eta \in (0,1]$ is a scaling parameter, and P is the projection onto $\pi \cap \mathbb{T}$, given as an solution of optimization problem

$$\begin{aligned} P(\mathbf{a_y} - \eta \nabla m(\mathbf{a_y})) = \arg\min_{\mathbf{a}} \quad & \|\mathbf{a} - (\mathbf{a_y} - \eta \nabla m(\mathbf{a_y}))\|_2, \\ \text{s.t.} \quad & \mathbf{a} \in \pi \cap \mathbb{T}. \end{aligned} \tag{7}$$

The problem of projecting onto an arbitrary set can be interpreted as finding the closest point in the set to the projected point. Conveniently, the minimization (7) leads to quadratic programming with linear inequality constraints, which can be very efficiently solved using the active set method approach [4].

With the projected descent direction of $m(\mathbf{a})$, we perform a backtracking line search to find β satisfying the Armijo rule

$$m(\mathbf{a_y} + \beta \mathbf{d}) \leq m(\mathbf{a_x}) + c_1 \beta (\Delta m(\mathbf{a_y})^T \mathbf{d}) \tag{8}$$

where $\beta \in (0, \min(\frac{\delta}{|\mathbf{d}|_2}, 1)]$ is a scalar multiplier of the gradient and $c_1 = 0.1$ is the Armijo rule constant. The new point $\mathbf{a_z}$ is then evaluated as

$$\mathbf{a_z} = \mathbf{a_y} + \beta \mathbf{d}.$$

4.3 Acceptance and rejection

Since the minimization of (6) deals only with the linearly approximated feasible set $\pi \cap \mathbb{T}$, the new point $\mathbf{a_z}$ might not necessarily belong to the feasible set Ω. Thus, in addition to the Armijo rule, we define a merit function that simultaneously assesses both the decrease in the objective function and the constraint violations.

The proposed merit function $\psi(\mathbf{a}, \theta)$ is a convex combination of the objective function and the constraint violation function

$$\psi(\mathbf{a}, \theta) = \theta m(\mathbf{a}) + (1 - \theta) h(\mathbf{a}),$$

where $\theta \in (0,1]$ is the penalty parameter satisfying

$$P_{red}(\theta) \geq \frac{1}{2}(h(\mathbf{a_x}) - h(\mathbf{a_y})) \tag{9}$$

and $P_{red}(\theta)$ is the predicted decrease of the merit function

$$P_{red}(\theta) = \theta(m(\mathbf{a_x}) - m(\mathbf{a_z})) + (1 - \theta)(h(\mathbf{a_x}) - h(\mathbf{a_y})).$$

Since the P_{red} is linear in θ, by rewriting inequality (9) as an equation, we can derive an explicit solution for θ as

$$\theta = \begin{cases} \max(\theta_{eq}, \theta) & \text{if} g_\theta \geq 0 \\ \max(0, \theta_{eq}) & \text{otherwise} \end{cases}$$

where g_θ represents the derivative of $P_{red}(\theta)$, and θ_{eq} is the penalty parameter value that satisfies the inequality (9) as an equality

$$g_\theta = (m(\mathbf{a_x}) - m(\mathbf{a_z})) - (h(\mathbf{a_x}) - h(\mathbf{a_y})),$$

$$\theta_{eq} = -\frac{1}{2}\left(\frac{m(\mathbf{a_x}) - m(\mathbf{a_z})}{h(\mathbf{a_x}) - h(\mathbf{a_y})} - 1\right)^{-1}.$$

With the updated penalty parameter θ, we can then evaluate the reduction in the merit function between $\mathbf{a_x}$ and $\mathbf{a_z}$

$$A_{red} = \theta(m(\mathbf{a_x}) - m(\mathbf{a_z})) + (1-\theta)(h(\mathbf{a_x}) - h(\mathbf{a_z})),$$

and compare it with the predicted decrease P_{red}

$$A_{red} \geq 0.1 P_{red}. \tag{10}$$

If the inequality (10) holds, we set $\mathbf{a_z}$ as the new $\mathbf{a_x}$ and return to the restoration step (5). Otherwise, we reduce the trust region δ and return to (8).

5 Numerical benchmark

For the numerical benchmark, we consider a cantilever beam with square cross section of length $l = 10\mathrm{m}$ with fixed support on one end and $f_z = 10\mathrm{kN}$ vertical force on the other, see Figure 1. The beam is made out of steel with material density $\rho = 7850\mathrm{kg/m}^3$ and Young modulus $E = 210\mathrm{GPa}$. For the constrains, we consider only maximal stress for each element equal to yield strength $f_y = 235\mathrm{MPa}$. The presented benchmark has an analytical solution for optimal geometrical parameter at given point and the the minimal weight of beam as

$$a(x) = \left(\frac{6 f_z x}{f_y}\right)^{\frac{1}{3}} \qquad m_{opt} = 1.981 \rho L \left(\frac{f_z L}{f_y}\right)^{\frac{2}{3}}$$

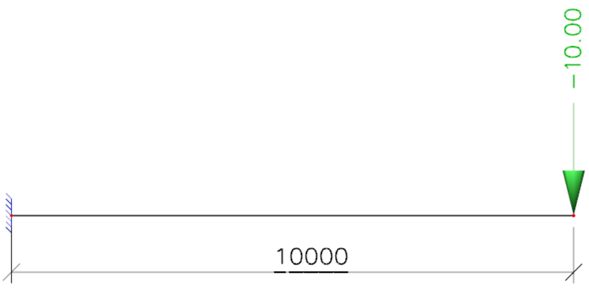

Figure 1: Numerical benchmark: Cantilever beam

The beam is segmented into n_e elements, each connected by $n_e + 1$ nodes, resulting in $3(n_e+1)$ degrees of freedom. For a series of element counts $n_e \in \{2, 4, 8, 16, 32, 64, 128\}$, we generate 400 random initial configurations within the range $a_x \in [0.01, 0.2]$. The algorithm is executed for each configuration, and the outcomes are averaged for every set of divisions.

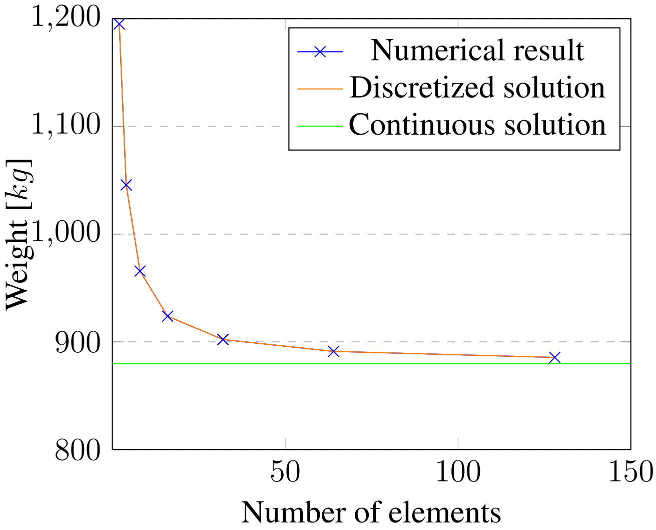

Figure 2: Objective function (weight) dependent on the number of elements

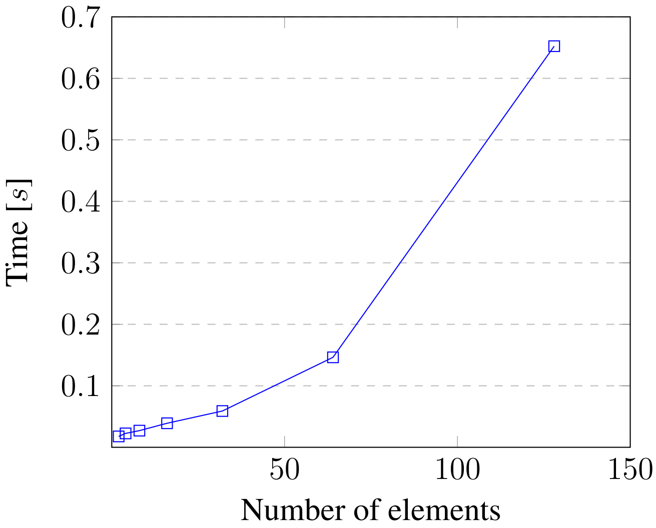

Figure 3: Execution time per run of the algorithm dependent on the number of elements

Figure 2 illustrates a comparison between the results obtained from the Inexact Restoration algorithm (depicted in blue), the optimal weight derived from the discretized problem (shown in red), and the analytical solution for a beam with a continuously distributed mass (represented in green). The numerical results align precisely with those of the discretized solution and demonstrate an asymptotic approach towards the analytical solution as the number of elements increases. Additionally, Figure 3 presents the execution time for a single run of the algorithm, measured in seconds, which exhibits an cubically increase with the rising number of elements. This trend is primarily attributed to the computation involved in the gradient of deformations $\nabla \mathbf{r}(\mathbf{a})$, which has a numerical complexity of $O(n_d^3)$. In the context of this straightforward benchmark, it's possible to express the degrees of freedom as a function of the number of elements, leading to a complexity of $O(n_e^3)$. Consequently, the computational time required for each algorithm run escalates cubically with an increase in the number of elements.

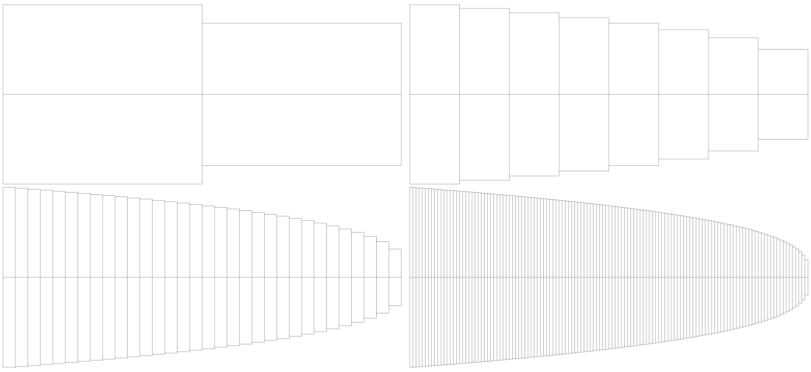

Figure 4: Optimised shape of the benchmark cantilever beam for 2 (top left), 8 (top right), 32 (bottom left) and 128 (bottom right) elements

6 Conclusion and future work

In the paper, we have presented the results of our implementation of the inexact restoration algorithm for solving the constrained weight optimisation problem for a 2D elastic frame structure discretized by FEM with deformation and stress constraints. The results from our benchmark, as seen in Fig. 2, show an exact match with the analytical discretized solution as well as asymptotic convergence to the analytical solution for the beam with continuously distributed mass. The clock time for the algorithms, as illustrated in Fig. 3, shows cubical growth with the number of elements, but remains within a practically acceptable range of 0.7 seconds for 128 elements.

In our future work, we will extend the presented approach to include a variety of cross sections and multiple load cases, which are essential for the practical use of the algorithms. Additionally, in practical applications, it would be highly inefficient to assign each beam a different cross section. Thus, adding modularity to the system would not only decrease the complexity of the system but also bring the algorithms one step closer to practical application.

References

[1] M.F. Ahmad, N.A.M. Isa, W.H. Lim, K.M. Ang, Differential evolution: A recent review based on state-of-the-art works. *Alexandria Engineering Journal*, 61(5):3831–3872, May 2022.

[2] K.J. Bathe, *Finite Element Procedures*. Prentice Hall, 2006.

[3] Z. Dostál, *Optimal Quadratic Programming Algorithms, with Applications to Variational Inequalities*, volume23. SOIA, Springer, New York, US, 2009.

[4] A. Forsgren, P.E. Gill, E. Wong, Primal and dual active-set methods for convex quadratic programming, *Mathematical Programming*, 159(1–2):469–508, December 2015.

[5] H. Fredricson, T. Johansen, A. Klarbring, J.Petersson, Topology optimization of frame structures with flexible joints, *Structural and Multidisciplinary Optimization*, 25(3):199–214, August 2003.

[6] A.G. Gad, Particle swarm optimization algorithm and its applications: A systematic review, *Archives of Computational Methods in Engineering*, 29(5):2531–2561, April 2022.

[7] J. Logo, H. Ismail, Milestones in the 150-year history of topology optimization: A review, *Computer Assisted Methods in Engineering and Science*, 27(2–3):97–132, 2020.

[8] J.M. Martínez E.A. Pilotta, Inexact-restoration algorithm for constrained optimization1, *Journal of Optimization Theory and Applications*, 104(1):135–163, January 2000.

[9] F. Mitjana, S. Cafieri, F. Bugarin, C. Gogu, F. Castanie, Optimization of structures under buckling constraints using frame elements, *Engineering Optimization*, 51(1):140–159, March 2018.

[10] J. Nocedal, S.J. Wright, *Numerical Optimization*, Springer, New York, NY, USA, 2e edition, 2006.

[11] K.B. Petersen, M.S. Pedersen, The matrix cookbook, October 2008. Version 20081110.

[12] K. Svanberg, The method of moving asymptotes—a new method for structural optimization, *International Journal for Numerical Methods in Engineering*, 24(2):359–373, February 1987.

[13] G.G. Tejani, V.J. Savsani, S. Bureerat, V.K. Patel, P. Savsani, Topology optimization of truss subjected to static and dynamic constraints by integrating simulated annealing into passing vehicle search algorithms, *Engineering with Computers*, 35(2):499–517, May 2018.

[14] B. Torstenfelt, A. Klarbring, Structural optimization of modular product families with application to car space frame structures, *Structural and Multidisciplinary Optimization*, 32(2):133–140, March 2006.

[15] K.Trustrum, *Linear Programming*, Springer Netherlands, 1971.

[16] Z. Zhao, S. Zhou, K. Cai, Y.M. Xie, A direct approach to controlling the topology in structural optimization, Computers & Structures, 227:106141, January 2020.