



SIXTH FRAMEWORK PROGRAMME

FP-6 STREP 30717 PLATO-N (Aeronautics and Space)

PLATO-N

**A PLAtform for Topology Optimisation incorporating Novel, Large-Scale,
Free-Material Optimisation and Mixed Integer Programming Methods**

Convex approximations for complex non-convex constraints

PLATO-N Public Report PU-R-6-2007

October 13, 2007

Authors:

A. Beck

A. Ben-Tal

L. Tetrushvili

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)

Dissemination level: Public

Convex approximations for complex non-convex constraints

Amir Beck , Aharon Ben-Tal and Luba Tetrushvili *

October 13, 2007

Abstract

We consider three algorithmic approaches to deal with displacement and/or stress constraints in free material optimization (FMO) models. The first approach is a simple sequential linear programming (SLP) approach and we describe how to adopt it to FMO-type problems with semidefinite constraints. We then provide another approximation scheme based on [1] to a general form of nonconvex constraints that includes (among other) stress or displacement constraints. At each iteration the constraint function is approximated by another overestimate function and thus the approximated solution is guaranteed to be feasible. Finally, we briefly review a sequential convex semidefinite programming method recently developed by Stingl, Kočvara and Leugering [4].

*MINERVA Optimization Center, Faculty of Industrial Engineering, Technion - Israel Institute of Technology, Haifa 32000, Israel.

e-mails: becka@ie.technion.ac.il, abental@ie.technion.ac.il, lubate@tx.technion.ac.il

1 Sequential Linear Programming

Sequential Linear Programming (SLP) is the a very simple and popular approach for finding an approximate solution of (possibly nonconvex) optimization problems. It is also used in structural optimization problems, see e.g., [2]. In this section we briefly describe the approach and its adaptation to FMO-type problems. Given an optimization problem of the form

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && g_j(x) \leq 0, \quad j = 1, \dots, n, \end{aligned} \tag{1.1}$$

the SLP method begins with an initial point x_0 and replaces the objective function and constraints by their corresponding linear approximations around x_0 :

$$\begin{aligned} & \text{minimize} && f(x_0) + (x - x_0)^T \nabla f(x_0) \\ & \text{subject to} && g_j(x_0) + (x - x_0)^T \nabla g_j(x_0) \leq 0 \quad j = 1, \dots, n, \\ & && lb \leq x - x_0 \leq ub. \end{aligned} \tag{1.2}$$

The last set of constraints are called *move limits*, with vectors lb and ub being the lower and upper bounds on the allowed change of the variable x .

Denote by x_L the optimal point of problem (1.2). We replace x_0 by the point x_L and solve problem (1.1) again, this time linearized around the new point x_L . This process should be repeated until we get sufficiently close to the optimal solution of (1.1). Hence, in the SLP approach we solve at each iteration a linear programming (LP) problem.

We will now show how to adopt the SLP method to solve FMO-type problems. Recall one of the basic models of the FMO problem. Consider a collection $t = \{t_i\}_{i=1}^N$ of positive definite matrices $t_i \in \mathbb{S}_{++}^d$; t_i is the rigidity tensor of the shape in finite element cell i multiplied by the area/volume of the cell. The twice compliance of the shape w.r.t. a load f is

$$C_f(t) = f^T (A(t))^{-1} f, \quad A(t) = \sum_{i,s} b_{is}^T t_i b_{is}.$$

The multi-load design problem is

$$\min_{t \in \mathcal{T}} \max_{1 \leq k \leq K} \ln(f_k^T (A(t))^{-1} f_k), \tag{1.3}$$

where \mathcal{T} is a given closed convex set $\{t = \{t_i\}_{i=1}^N : t_i \succeq 0, \sum_i \text{Tr}(t_i) = 1\}$. Consider the following smoothing approximation of this problem:

$$\min_{t \in \mathcal{T}} F_\beta(t), \quad F_\beta(t) \equiv \beta^{-1} \ln \left(\sum_{k=1}^K (f_k^T (A(t))^{-1} f_k)^\beta \right), \tag{1.4}$$

where $F_\beta(t)$ is convex and smooth function, twice continuously differentiable on X . Observe that

$$\max_k \ln(f_k^T (A(t))^{-1} f_k) \leq F_\beta(t) \leq \max_k \ln(f_k^T (A(t))^{-1} f_k) + \beta^{-1} \ln K.$$

It follows that for sufficiently big value of β we obtain a good approximation of the objective function in (1.3) and thus we can substitute the original objective function in (1.3) by the function $F_\beta(t)$. We shall use $F_\beta(t)$ as our objective function. The above choice of F_β is

flexible. For example, we can use an objective function which is a weighted sum of the compliances ($\sum_{k=1}^K \omega_k (f_k^T (A(t))^{-1} f_k)$). The convex FMO problem we consider is therefore:

$$\begin{aligned} \min \quad & \beta^{-1} \ln \left(\sum_{k=1}^K (f_k^T (A(t))^{-1} f_k)^\beta \right) \\ \text{s.t.} \quad & \sum_i \text{Tr}(t_i) = 1, \\ & t_i \succeq 0, \quad i = 1, \dots, N, \end{aligned} \tag{1.5}$$

where t is a collection of N $d \times d$ positive definite symmetric matrices ($d = 3$ for planar shapes and $d = 6$ for spatial shapes, N is the number of final elements); $f_k, k = 1, \dots, K$, are given load M -dimensional vectors (K -the number of load cases, M -number of nodes); $A(t) = \sum_{i=1}^N \sum_{s=1}^S b_{is}^T t_i b_{is}$ is the global stiffness matrix where b_{is} are given $d \times M$ matrices.

We would also like to introduce nonconvex constraints. To illustrate the approach, we consider a more involved model in which displacement constraints are added:

$$\begin{aligned} \min \quad & \beta^{-1} \ln \left(\sum_{k=1}^K (f_k^T (A(t))^{-1} f_k)^\beta \right) \\ \text{s.t.} \quad & \sum_i \text{Tr}(t_i) = 1, \\ & e_j^T (A(t))^{-1} f_k \leq d_j, \quad k = 1, \dots, K, \quad j \in J, \\ & t_i \succeq 0, \quad i = 1, \dots, N. \end{aligned} \tag{1.6}$$

Here e_j is the j -th vector in the canonical base and J is the set of the coordinates of nodes for which displacement constraints are to be imposed. In what follows we describe how the SLP method can be applied to the problem (1.6).

1.1 Solving FMO problem with displacement constraints by SLP method

At each stage of the SLP algorithm we solve the following linearized model of the problem (1.6):

$$\begin{aligned} \min \quad & \sum_{i=1}^N \text{Tr}(u_i^0 t_i) \\ \text{s.t.} \quad & \sum_{i=1}^N \text{Tr}(t_i) = 1, \\ & \sum_{i=1}^N \text{Tr}(u_i^j t_i) \leq \tilde{d}_j, \quad j \in J, \\ & \alpha^{-1} \bar{t}_i \preceq t_i \preceq \alpha \bar{t}_i, \end{aligned} \tag{1.7}$$

where u_i^j are given symmetric matrix derived from $F'_\beta(\bar{t})$ and $g_j^i(\bar{t}) = (e_j^T (A(\bar{t}))^{-1} f_k - d_j)'$; (\bar{t} is the last generated solution of the linear approximation problem, i.e., collection of positive definite matrices).

Consider the Cholesky factorization of \bar{t}_i (recall that \bar{t}_i is positive definite), $\bar{t}_i = R_i^T R_i$. Define the following matrices:

$$\begin{aligned} q_i^j &= R_i w_i^j R_i^T, \quad j \in J \cup \{0\}, \\ h_i &= R_i R_i^T, \end{aligned}$$

and the new matrix variable

$$s_i = (R_i^T)^{-1} t_i R_i^{-1}.$$

We can now rewrite problem (1.7) as follows:

$$\begin{aligned} \min \quad & \sum_{i=1}^N \text{Tr}(q_i^0 s_i) \\ \text{s.t.} \quad & \sum_{i=1}^N \text{Tr}(h_i s_i) = 1, \\ & \sum_{i=1}^N \text{Tr}(q_i^j s_i) \leq \tilde{d}_j, \quad j \in J, \\ & \alpha^{-1} I \preceq s_i \preceq \alpha I. \end{aligned} \tag{1.8}$$

To solve the above problem, we will apply a dual approach. The Lagrangian dual objective function is

$$q(\lambda, \mu) \equiv \min_{\alpha^{-1} I \preceq s_i \preceq \alpha I} \sum_{i=1}^N \text{Tr}(q_i^0 s_i) + \sum_{j \in J} \lambda_j \left(\sum_{i=1}^N \text{Tr}(q_i^j s_i) - \tilde{d}_j \right) + \mu \left(\sum_{i=1}^N \text{Tr}(h_i s_i) - 1 \right).$$

The dual problem is given by

$$\max_{(\lambda, \mu) \in \mathbb{R}_{|J|}^+ \times \mathbb{R}} q(\lambda, \mu) \equiv \max_{(\lambda, \mu) \in \mathbb{R}_{|J|}^+ \times \mathbb{R}} \min_{(\alpha^{-1} I \preceq s_i \preceq \alpha I)} \left(\sum_{i=1}^N \text{Tr}((q_i^0 + \sum_{j \in J} \lambda_j q_i^j + \mu h_i) s_i) - \tilde{d}^T \lambda - \mu \right). \tag{1.9}$$

This is a convex problem (maximization of a concave function) and we can therefore apply a bundle level method or any other suitable algorithm to solve it. To use the bundle level method we need to be able to find the value and derivative of the function $L(\lambda, \mu)$. To compute a function value $q(\lambda, \mu)$ for a given (λ, μ) we need to solve the following problem for every $i = 1, \dots, N$:

$$\begin{aligned} \min \quad & \text{Tr}((q_i^0 + \sum_{j \in J} \lambda_j q_i^j + \mu h_i) s_i) \\ \text{s.t.} \quad & \alpha^{-1} I \preceq s_i \preceq \alpha I. \end{aligned} \tag{1.10}$$

whose solution is denoted by $s_i(\lambda, \mu)$

Remark. Note that (1.10) can be easily transformed into a simple LP. To see that, observe that the matrix $D(\lambda, \mu) = q_i^0 + \sum_{j \in J} \lambda_j q_i^j + \mu h_i$ is symmetric, a spectral decomposition can be computed $d(\lambda, \mu) = V^T D(\lambda, \mu) V$, where V is orthogonal and D is diagonal. Define the new variable $x = V^T s_i V$. The problem (1.10) can be rewritten as the following program:

$$\begin{aligned} \min \quad & \text{Tr}(d(\lambda, \mu)x) \\ \text{s.t.} \quad & \alpha^{-1}I \preceq x \preceq \alpha I \end{aligned} \tag{1.11}$$

It is easy to show that among the optimal solutions of (1.11) there exists diagonal matrix solution. Thus we obtain the simple linear problem:

$$\begin{aligned} \min \quad & \text{Tr}(d(\lambda, \mu)x) \\ \text{s.t.} \quad & \alpha^{-1}I \leq x \leq \alpha I \end{aligned} \tag{1.12}$$

where x is a diagonal matrix.

After we find the matrices $s_i(\lambda, \mu)$ we can compute the gradient of the function $q(\lambda, \mu)$ for given point (λ, μ) as follows:

$$q'(\lambda, \mu)|_{\lambda_j} = \sum_{i=1}^N \text{Tr}(q_i^j s_i(\lambda_*, \mu_*)) - \tilde{d}_j, \quad j \in J,$$

$$q'(\lambda, \mu)|_{\mu} = \sum_{i=1}^N \text{Tr}(h_i s_i(\lambda_*, \mu_*)) - 1.$$

Denote by (λ_*, μ_*) the optimal solution of the problem (1.9) and by $s_i(\lambda_*, \mu_*)$ the optimal solution of problem (1.8). The optimal solution of the original problem (1.7) is given by $t_i(\lambda_*, \mu_*) = R_i^T s_i(\lambda_*, \mu_*) R_i$.

2 Convex approximation for displacement and stress constraints

In this section we propose an approximation methodology for a class of nonconvex constraints appearing in FMO problems, in particular displacement and stress constraints. The approach is based on the technique advocated in [1] for truss design with a global buckling constraint.

Consider the function

$$H(x, y, z) = |e^T Y X^{-1} Z f|, \tag{2.1}$$

where X, Y, Z are matrix-valued functions depending on a variable $t \in \mathcal{T}$ such that $X \succ 0$. We begin by showing the relevance of this class of functions to constraints arising in FMO problems:

- i. **Displacement constraints.** Suppose that we are given an l_1 norm displacement constraint:

$$\|A(t)^{-1} f\|_1 \leq \rho$$

This constraint can be transformed into a system of constraints:

$$\begin{aligned} |e_j^T A(t)^{-1} f| &\leq \tau_j, \quad j = 1, \dots, N, \\ \sum \tau_j &\leq \rho. \end{aligned}$$

The nonlinear constraints in the above system are of the required structure ($Y = Z = I$ and $X = A(t)$). A more general form of displacements constraints is given by

$$CA(t)^{-1}f \leq d,$$

where the matrix C and vector d are of suitable sizes (see e.g.,[3]). It can be shown that these constraints can also be rewritten in terms of functions of the form (2.1).

ii. **Stress constraints.** Stress constraints can be modelled as [3]:

$$\sum_{s=1}^S \|tB_{i,s}A(t)^{-1}f\|^2 \leq \nu, \quad i = 1, \dots, N$$

where $B_{i,s}$ are given matrices. Note that $e_j^T tB_{i,s}A(t)^{-1}f$ is the j -th coordinate of the vector $q_{i,s} = tB_{i,s}A(t)^{-1}f$ so we can rewrite the stress constraint as

$$\sum_{s=1}^S \sum_j |e_j^T tB_{i,s}A(t)^{-1}f|^2 \leq \nu, \quad i = 1, \dots, N.$$

Setting $tB_{i,s} = Y_{i,s}(T), X = A(t)$ and $Z = I$, the above can be equivalently rewritten as the following set of convex inequalities

$$|e_j^T Y_{i,s}(t)X(t)^{-1}Z(t)f| \leq \tau_{j,i,s}$$

$$\sum_s \sum_j \tau_{j,i,s}^2 \leq \nu, \quad i = 1, \dots, N.$$

As in the case of displacement constraint, we observe that the nonlinear constraint comprise functions of the form (2.1).

We have thus established the fact that the function H given by (2.1) appears both in displacement and stress constraints. We will next show that the function

$$F_{\lambda,h}(X, Y, Z) = \frac{\lambda}{2} e^T Y X^{-1} Y^T e + \frac{1}{2\lambda} (Zf + Xh)^T X^{-1} (Zf + Xh) \quad (2.2)$$

is an upper bound of $H(X, Y, Z)$ for every $\lambda > 0$ and h satisfying $e^T Y h = 0$.

Theorem 2.1 *Let X, Y, Z be $n \times n, k \times m$ and $n \times r$ matrix-valued functions depending on variable $t \in \mathcal{T}$, suppose $X \succ 0$ is positive definite matrix for all $t \in \mathcal{T}$ and let e and f be given vectors of lengths k and r respectively. Let λ be some positive number and h a vector satisfying $e^T Y h = 0$. Then*

$$H(X, Y, Z) \leq F_{\lambda,h}(X, Y, Z) \text{ for every } t \in \mathcal{T}.$$

Before proving Theorem 2.1 we state in the following lemma some basic facts (without a proof) that will be used in the proof of Theorem 2.1.

Lemma 2.2 i. Let Q be an $n \times n$ symmetric positive definite matrix, $a \in \mathbb{R}^n$, $b \in \mathbb{R}^n$. Then

$$|a^T b| \leq (a^T Q a)^{1/2} (b^T Q^{-1} b)^{1/2} \quad (2.3)$$

and equality holds if and only if a and $Q^{-1}b$ are linearly dependent.

ii. For given scalars $\alpha \geq 0, \beta \geq 0$, the optimal solution of

$$\min_{\lambda \geq 0} \left\{ \frac{\lambda}{2} \alpha + \frac{1}{2\lambda} \beta \right\}$$

is $\sqrt{\frac{\beta}{\alpha}}$ with a corresponding optimal value $\sqrt{\alpha\beta}$.

We are now ready to prove Theorem 2.1.

Proof of Theorem 2.1: First note that since h satisfies $e^T Y h = 0$ we have

$$H(X, Y, Z) = |e^T Y X^{-1} Z f| = |e^T Y X^{-1} (Z f + X h)|.$$

Applying (2.3) with $a = Y^T e$, $Q = X^{-1}$ and $b = X^{-1}(Z f + X h)$ we thus obtain

$$H(X, Y, Z) \leq \left(\underbrace{e^T Y X^{-1} Y^T e}_{\alpha} \right)^{1/2} \left(\underbrace{(Z f + X h)^T X^{-1} (Z f + X h)}_{\beta} \right)^{1/2}$$

Invoking part ii of Lemma 2.2 on α and β defined above, we thus obtain that

$$H(X, Y, Z) \leq \frac{\lambda}{2} e^T Y X^{-1} Y^T e + \frac{1}{2\lambda} (Z f + X h)^T X^{-1} (Z f + X h) = F_{\lambda, h}(X, Y, Z),$$

proving the result. \square

It is important to note that $F_{\lambda, h}(X, Y, Z)$, as opposed to $H(X, Y, Z)$, is convex.

Proposition 2.3 For given λ and vector h the function $F_{\lambda, h}(X, Y, Z)$ is convex.

Proof: Denote

$$F_{\lambda, h} = \frac{\lambda}{2} f_1(X, Y, Z) + \frac{1}{2\lambda} f_2(X, Y, Z).$$

It is enough to show that both f_1 and f_2 are convex. The epigraph of f_1 is given by

$$\{(X, Y, Z, u_1) | u_1 \geq e^T Y X^{-1} Y^T e\},$$

which by Schur's complement can be cast an LMI:

$$\begin{pmatrix} u_1 & e^T Y \\ Y^T e & X \end{pmatrix} \succeq 0.$$

Similarly, the epigraph of the function f_2 is given by

$$\begin{pmatrix} u_2 & (Z f + X h)^T \\ Z f + X h & X \end{pmatrix} \succeq 0,$$

proving the convexity of $F_{\lambda, h}(X, Y, Z)$ \square

Important remarks

- Note that $F_{\lambda,h}(X(t), Y(t), Z(t))$ is also convex function of the variable t whenever $X(t), Y(t), Z(t)$ depend affinely on t .
- It follows from the above proof that the inequality $F_{\lambda,h}(X, Y, Z) \leq \rho$ has the following semidefinite representation (SDR):
$$\left\{ \begin{array}{l} \left(\begin{array}{cc} u_1 & e^T Y \\ Y^T e & X \end{array} \right) \succeq 0, \left(\begin{array}{cc} u_2 & (Zf + Xh)^T \\ Zf + Xh & X \end{array} \right) \succeq 0, \quad \frac{\lambda}{2}u_1 + \frac{1}{2\lambda}u_2 \leq \rho, \quad X \succ 0 \end{array} \right\}$$
Note that if we set $X = A(t)$ we get linear matrix inequality system (LMI) in the variable t .

It is also important to study the conditions on λ and h under which H and $F_{\lambda,h}$ are equal.

Proposition 2.4 For given (X, Y, Z) there is a point $(\bar{\lambda}, \bar{h})$ such that $e^T Y \bar{h} = 0, \bar{\lambda} > 0$ and

$$H(X, Y, Z) = F_{\bar{\lambda}, \bar{h}}(X, Y, Z). \quad (2.4)$$

Proof: A straightforward computation shows that (2.4) is satisfied with

$$\bar{h} = X^{-1}(\theta Y^T e - Zf), \bar{\lambda} = |\theta|, \quad (2.5)$$

where $\theta = \frac{e^T Y X^{-1} Z f}{e^T Y X^{-1} Y^T e}$. \square

Now we are ready to describe the iterative procedure for solving approximated problem (1.6).

Algorithm

Step 0 Choose an arbitrary vector (λ, h) and replace the nonconvex constraint in (1.6) by convex one $F_{\lambda,h}(X(t), Y(t), Z(t)) \leq \rho$ or by its SDR system and solve the obtained approximated problem (1.6) in variable t .

Step 1 Let t_k be the solution of the approximated problem (1.6) at iteration k . Compute

$$\begin{aligned} \theta_k &= \frac{e^T Y(t_k) X(t_k)^{-1} Z(t_k) f}{e^T Y(t_k) X(t_k)^{-1} Y(t_k)^T e}, \\ h_k &= X(t_k)^{-1} (\theta_k Y(t_k)^T e - Z(t_k) f), \\ \lambda_k &= |\theta_k| \end{aligned}$$

Step 2 Solve the approximated problem (1.6) with (λ_k, h_k) and denote by t_{k+1} its optimal solution. Stop if $\|t_{k+1} - t_k\| \leq \varepsilon$. Otherwise go to Step 1.

3 A Sequential Convex Semidefinite Programming Method [4]

In this section we briefly review a sequential convex semidefinite programming algorithm recently proposed by Stingl M.Kočvara and Leugering [4] for multiple load FMO problems. The main idea of the new method is to approximate the original problem by a sequence of semidefinite sub-problems, in which nonlinear functions (defined in matrix variable) are approximated by convex, block-separable functions. The algorithm is defined on a convex formulation of the FMO problem but can be extended to cover state constraints, such as

displacement constraints, strain or stress constraints in a straight forward way.

Definition. Let $f : \mathbb{S}^{d_1} \times \mathbb{S}^{d_2} \times \dots \times \mathbb{S}^{d_m} \rightarrow \mathbb{R}$ be continuously differentiable on a subset $B \subset \mathbb{S} = \mathbb{S}^{d_1} \times \mathbb{S}^{d_2} \times \dots \times \mathbb{S}^{d_m}$ and denote $\bar{Y} = (\bar{Y}_1, \dots, \bar{Y}_m) \in B$ to be a vector of matrices. Moreover let asymptotes $L = (L_1, L_2, \dots, L_m)^T$, $U = (U_1, U_2, \dots, U_m)^T$ be given such that

$$L_i \prec \bar{Y}_i \prec U_i, \quad i = 1, \dots, m$$

and $\tau := \{\tau_1, \tau_2, \dots, \tau_m\}$ be set of non-negative real parameters. Then we define the hyperbolic convex approximation $f_{\bar{Y}}^{L,U,\tau}$ of f at \bar{Y} as

$$\begin{aligned} f_{\bar{Y}}^{L,U,\tau}(Y) &:= f(\bar{Y}) + \\ &\sum_{i=1}^m \langle \nabla_+^i f(\bar{Y}), (U_i - \bar{Y}_i)(U_i - Y_i)^{-1}(U_i - \bar{Y}_i) - (U_i - \bar{Y}_i) \rangle_{\mathbb{S}^{d_i}} - \\ &\sum_{i=1}^m \langle \nabla_-^i f(\bar{Y}), (\bar{Y}_i - L_i)(Y_i - L_i)^{-1}(\bar{Y}_i - L_i) - (\bar{Y}_i - L_i) \rangle_{\mathbb{S}^{d_i}} + \\ &\sum_{i=1}^m \langle (Y_i - \bar{Y}_i)^2, (U_i - Y_i)^{-1} + (Y_i - L_i)^{-1} \rangle_{\mathbb{S}^{d_i}} \end{aligned}$$

where $\langle A, B \rangle_{\mathbb{S}^{d_i}} := \text{Tr}(AB)$ and $\langle \nabla_+^i f(\bar{Y}), \nabla_-^i f(\bar{Y})$ denote the projections of $\frac{\partial f}{\partial Y_i}$ onto $\mathbb{S}_+^{d_i}$ and $\mathbb{S}_-^{d_i}$ respectively.

Note that $f_{\bar{Y}}^{L,U,\tau}(Y)$ is separable w.r.t. the matrix variables Y_1, Y_2, \dots, Y_m .

Algorithm

Consider the following problem:

$$\begin{aligned} \min_{Y \in \mathbb{S}} \quad & f(Y) && (\mathcal{P}) \\ \text{s.t.} \quad & g_k(Y) \leq 0, \quad k = 1, 2, \dots, K, \\ & \underline{Y}_i \preceq Y_i \preceq \bar{Y}_i, \quad i = 1, 2, \dots, m. \end{aligned}$$

Throughout this section we make the following assumptions:

A1 $f : \mathbb{S} \rightarrow \mathbb{R}$ is convex and is the maximum over a finite set of twice continuously differentiable functions

$$f(Y) = \max_l f_l(Y).$$

A2 the functions $g_k : \mathbb{S} \rightarrow \mathbb{R}$ ($k = 1, 2, \dots, K$) are continuously differentiable and convex.

A3 The interior of the feasible domain is non-empty.

A4 The compact sets of asymptotes \mathcal{L} and \mathcal{U} satisfy the following condition: Let \mathcal{F} be a feasible domain of the problem (\mathcal{P}) then $\forall L \in \mathcal{L}$ and $\forall U \in \mathcal{U} \exists \mu > 0 : \forall i = 1, \dots, m, \forall Y \in \mathcal{F} : Y_i - L_i \succeq \mu I$ and $U_i - Y_i \succeq \mu I$.

For a given feasible point Y^j of problem (\mathcal{P}) , we define a local hyperbolic approximation of f as

$$f^j(Y) := \max_l f_l^j(Y) := \max_l (f_l)_{Y^j}^{L^j, U^j, \tau^j}(Y).$$

Using this function, we define a local approximation of (\mathcal{P}) close to Y^j as follows:

$$\begin{aligned} \min_{Y \in \mathbb{S}} \quad & f^j(Y) && (\mathcal{P}^j) \\ \text{s.t.} \quad & g_k(Y) \leq 0, \quad k = 1, 2, \dots, K, \\ & \underline{Y}_i^j \preceq Y_i \preceq \overline{Y}_i^j, \quad i = 1, 2, \dots, m. \end{aligned}$$

Here the bounds $\underline{Y}_i^j, \overline{Y}_i^j$ should satisfy the following assumption:

A5 $\underline{Y}_i \preceq \underline{Y}_i^j \preceq Y_i^j \preceq \overline{Y}_i^j \preceq \overline{Y}_i$ for all $i = 1, \dots, m$.

Proposition 3.1 *Each sub-problem (\mathcal{P}^j) has a unique solution Y^{j+1} . Associated with Y^{j+1} there exist Lagrangian multipliers (v^{j+1}, V^{j+1}) such that $(Y^{j+1}, v^{j+1}, V^{j+1})$ is a KKT-point of (\mathcal{P}^j) .*

Description of the algorithm

1. Put $j=1$.
2. Choose asymptotes $L^j \in \mathcal{L}, U^j \in \mathcal{U}$ and $\bar{\tau} \geq \tau_1^j, \tau_2^j, \dots, \tau_m^j \geq \underline{\tau} > 0$.
3. Solve problem (\mathcal{P}^j) . Denote the solution by $Y^+ \in \mathbb{S}$ and the associated Lagrangian multipliers by $(v^+, V^+) \in \mathbb{R}_+^k \times \mathbb{S}_+$.
4. Choose $\alpha^j = \min\{1, \hat{\alpha}\}$, where $\hat{\alpha} = \min_{\alpha \in \mathbb{R}_+} f(Y^j + \alpha(Y^+ - Y^j))$.
5. Set $(Y^{j+1}, v^{j+1}, V^{j+1}) = (Y^j, v^j, V^j) + \alpha^j((Y^+, v^+, V^+) - (Y^j, v^j, V^j))$.
6. Stop if Y^{j+1} is stationary for problem (\mathcal{P}) ; otherwise put $j = j + 1$ and goto (2).

Theorem 3.2 *The Algorithm stops at a global minimizer of (\mathcal{P}) , or the sequence $\{Y^j\}$ generated by the Algorithm has at least one accumulation point and each accumulation point is a global minimizer of (\mathcal{P}) .*

The algorithm used to solve sub-problem (\mathcal{P}^j) is based on a generalized augmented Lagrangian method for the solution of nonlinear semidefinite programs.

The authors of the described Algorithm are currently investigating the possibilities to extend the method for the problems with non-convex constraints, when not only the objective function, but also the constraints can be replaced by sequence of hyperbolic approximations.

References

- [1] A. Ben-Tal, F. Jarre, M. Kočvara, A. Nemirovski, and J. Zowe. Optimal design of trusses under a nonconvex global buckling constraint. *Optim. Eng.*, 1(2):189–213, 2000.
- [2] R. T. Haftka and Z. Gürdal. *Elements of structural optimization*, volume 11 of *Solid Mechanics and its Applications*. Kluwer Academic Publishers Group, Dordrecht, third edition, 1992.
- [3] M. Kočvara, A. Beck, A. Ben-Tal, and M. Stingl. PLATO-N task 2.2: Software specification, selection of FMO problem formulations.
- [4] M. Stingl, M. Kočvara, and G. Leugering. A sequential convex semidefinite programming algorithm for multiple-load free material optimization. 2007. preprint.