Nonlinear programming in structural design

Bram de Jager
Technische Universiteit Eindhoven, Dept. of Mechanical Engineering WH 0.132
PO Box 513, NL-5600 MB Eindhoven, The Netherlands
A.G.de.Jager@wfw.wtb.tue.nl
WWW home page: http://yp.wtb.tue.nl/showemp.php/900
Phone: +31 40 2472784, FAX: +31 40 2461418

Abstract. The design of structural entities is formulated as a nonlinear program. In the nonlinear program the objective is the compliance (or stiffness under load) of a structure. Constraints are due to equilibrium conditions, signature of forces, failure conditions like yield and buckling, minimum length requirements and available material. Nonlinear constraints make this problem difficult to solve for realistic design problems using standard tools (Sequential Quadratic Programming: SQP) for nonlinear programming. Using techniques like scaling of design variables and constraints to bring the singular values of the constraint Jacobian in a narrower range and closer to 1, introducing additional design variables to make the constraints more linear, and employing sparsity of the Jacobian of the constraints, the problem can be solved without intervention for planar (small) problems, but not readily for spatial (large) problems. Some results in the design of a special class of structures, namely tensegrities, illustrate the problems. This bad scalability and lack of robustness are disadvantages of using standard tools, like SQP, to solve large nonlinear programs, which we would like to see redressed in the current tools, or alleviated in future ones. Using constraint satisfaction programs, it may be possible to more closely follow the curved manifolds of the constraints. A multi-phase procedure, including a global optimization phase, may be a feasible approach for large scale problems, because it may avoid badly conditioned area’s in the design space.

Keywords: Nonlinear programming, structural optimization, topology optimization, SQP, tensegrity structures

1 Introduction

We want to express our concerns about the techniques available to solve large nonlinear optimization problems that have value in practice. Problems were encountered during the design of structures (topology, geometry and material optimization, see Bendsøe (1995)), when too much simplifying assumptions were avoided. By introducing simplifying assumptions and using elimination techniques, topology optimization problems for discrete structures like trusses can be formulated as quadratic of even linear programs (Jarre et al. 1998), and are therefore readily solved. The simplifications makes these formulations useful to gain insight, but the resulting designs are of no practical use.
In our case, the focus is on a special class of structures called tensegrities (Adhikari et al. 1998). In these structures compressive and tensile members have pre-assigned roles and compressive members are not allowed to touch each other, giving a structure with excellent possibilities for changing shape dynamically. Differences with classical truss structures are the requirement that tensile members can carry tensile loads only, so changes in the sign of the force are not allowed, and compressive members are not allowed to connect to each other. These differences prevent the use of standard design tools.

Being forced to adapt the design tools, the goal was to do away with most of the simplifying assumptions also, and to try to develop a (combination) of techniques that would give insight and would also result in designs that are useful in practice.

In our formulation the objective is the compliance (or stiffness under load) of a structure. Constraints are a limited amount of material available, equilibrium conditions, signature of forces, minimum length of members, and failure conditions like yield and buckling. A tensegrity structure is characterized by its topology, geometry, and distribution of material. The topology describes how the members of the structure are connected, via a incidence matrix, while the geometry is specified by the positions of nodal points. Using as design parameters the geometry of the structure, the pre-stress, distribution of material over the members of the structure, and nodal displacements under load, the topology, geometry and material are specified. The goal is to find optimized values for these parameters. This formulation is not a standard way to approach this problem – it is quite challenging – and existing tools appear to be not really suitable. So our goal has not been reached yet.

The purpose of this paper is therefore to document our approach, highlight the shortcomings of, and probe for changes to, the formulation of the optimization problem, or to the choice and quality of tools used, in order to bring progress to the current state of the art, and enable the design of optimal structures that can be realized in practice.

The following sections give details of the mechanics of tensegrities, state the optimization problem, give the solution techniques, present some design examples, and finish with a discussion.

2 Mechanics of tensegrities

The equilibrium conditions for a frame structure with nodal point coordinates $p$ under a load $f$ acting on those nodes and causing a displacement $u$ of these nodes can be posed as

$$C^T \text{kron}(\Lambda, I_{\dim})C(p + u) = f,$$

which is just the balance of element forces under load at each of the nodes. Boundary conditions, e.g., for a support, are handled by removing the equilibrium equations for the relevant nodes from (1). Equation (1) is a classical result in the analysis of equilibria for mechanical structures.

Here, $p \in \mathbb{R}^{\dim \times n_n}$ is a column containing the nodal coordinates ($n_n$ is the number of nodes and $\dim$ is the dimensionality of the problem, either 2 or 3, note that $f$ and $u$ are elements of the same sized space as $p$). The matrix $C \in \mathbb{N}^{\dim(n_m \times n_n)}$ represents the connectivity of the frame, with $n_m$ the number of elements or members. Matrix $C$ is a
sparse block matrix whose $i, j$-th block is $I_{\text{dim}}$ or $-I_{\text{dim}}$ if the element $i$ ends at or emanates from node $j$, otherwise it is $0_{\text{dim}}$. It is a member-node incidence matrix. By using this formulation, it is assumed that a maximum set of allowed element connections of a tensegrity structure and its associated oriented graph have been adopted.

The diagonal matrix $\Lambda \in \mathbb{R}^{n_m \times n_m}$ contains the force coefficients (note that the member force itself is the force coefficient (a scaling factor) times the element vector $g_i$). The sign convention is that $\Lambda$ is positive for tensile and negative for compressive forces. The functions $\Lambda$ depend on the displacement $u$, so the equations are nonlinear, and they also depend on member volume and material properties. Setting member volume to zero eliminates a member from the structure, allowing topology optimization by an elimination approach.

Pre-stress is needed to stabilize the structure. So without load the member forces do not vanish in general and $\Lambda$ is not necessarily zero in the unloaded case, with $f = 0$. By using the Kronecker product we expand $\Lambda$ by a factor dim, so it matches in the equation.

The vector $g \in \mathbb{R}^{\text{dim} \times n_m}$, representing the orientation vectors of the elements, here in the loaded equilibrium, is computed as

$$g = C(p + u)$$

while the length of member $i$ is derived from

$$l_i = \|g_i\|_2.$$

Depending on the material model chosen, the relationship between force coefficients, $\Lambda$, and physical parameters of the structure may be different. For this analysis the linear elastic material model is used. Then the relation for $\Lambda$ is smooth, algebraic, and monotonous. It depends on material properties like Young’s modulus because this determines the changes in length of the tendons and bars due to changes in force. Expressed in terms of initial length, pre-stress and elongation of member $i$, it holds that

$$\Lambda_i l_i - \Lambda_0 l_0 = E_i v_i \frac{l_i - l_0}{l_i l_0}$$

with $E_i$ the modulus of elasticity of member $i$ and $v_i$ its volume. The subscript 0 indicates the values of force coefficients and lengths for the unloaded ($f = 0, u = 0$) case, or for any other case.

3 Formulation of optimization problem

The objective of this analysis is to design a tensegrity structure that, for a given mass of the material available, has an optimal stiffness. Assuming that all the elements are made of the same material, fixing the mass available is equivalent to specifying total volume, $S_v$, of the material used.

The optimization algorithm to a tensegrity structure whose number of nodes and number of members available are $n_n, n_m$ respectively, assigns structural parameters
collected in vectors of the nodal positions, \( p \), pre-stress of the elements, \( \Lambda_0 \), and volumes of the elements, \( v \in \mathbb{R}^{nm} \). For a given vector of applied external nodal forces \( f \), this set of parameters defines a structure, whose static response, defined in the vector of nodal displacements, \( u \), yields a compliance energy

\[
\frac{1}{2} f^T u,
\]

that is guaranteed to be improved from the value corresponding to an initial design. Note that compliance is used as a measure of the stiffness of the structure.

Our approach is based on nonlinear programming (NLP), in which we can embed pre-stress, failure conditions, and changes in geometry due to displacements. For the optimization of topology/geometry we consider the following set of design or optimization variables, as sketched above,

\[
x = \begin{bmatrix} p \\ \Lambda_0 \\ v \\ u \end{bmatrix}.
\]

Displacement \( u \) is included to be able to enforce the equilibrium conditions. We could also choose an approach were we solve the nonlinear set of equations (1) explicitly, but it is more convenient to let the optimization process handle this issue.

All columns stacked together give the design vector \( x \in \mathbb{R}^n \) with \( n = 2 \cdot \text{dim} \cdot n_n + 2n_m \). Appropriate modifications are made when some of the variables are not supposed to change, e.g., for the position of the nodes where the force is applied, or no deformations occur, e.g., for the position and displacement of the nodes that are connected to the support, so the size of the design vector is slightly less than indicated above.

It is clear that with this design vector the geometry can be influenced. Also the topology can be determined, when we allow a starting grid of nodal points and members that is more detailed than required, e.g., for accuracy of shape control. Members are allowed to vanish, when their volume approaches zero, making a change in topology, so \( v \) is included in \( x \).

To make the structure stiff, our objective is to minimize compliance

\[
\min_x f^T u,
\]

the inner product between the load \( f \) and the displacement \( u \) of the nodal points under load, in the presence of a set of (nonlinear) constraints. The composition of the load vector \( f \) is given, so the objective is linear in \( x \). The nonlinear equality and inequality constraints are:

1. equilibrium constraint

\[
C^T \kron(\Lambda(x), I_{\text{dim}})C(p + u) - f = 0
\]

employing the equations for a static equilibrium,
2. sign restricted force coefficients constraint

\[-z_i \Lambda_i \leq 0\]

which enforces tensile forces in the tendons (for which \(z_i = 1\)) and compressive forces in the bars (for which \(z_i = -1\)),

3. yield constraint

\[|\Lambda_i| l_i^2 - v_i \sigma_i \leq 0\]

so the stress in member \(i\) is always bounded by the yield stress \(\sigma_i\),

4. buckling constraint for the bars

\[|\Lambda_i| l_i^5 - \frac{\pi}{4} E_i v_i^2 \leq 0\]

which considers Euler buckling of a round cross section bar,

5. minimum length constraint for the unloaded case

\[-l_0 + a_i \leq 0, \quad a_i > a_{min}\]

imposed because the member length cannot be too short, otherwise there is no place for the joint construction or for the device that is needed to actively change the tendon lengths, or, when using memory type alloys, a short tendon allows only a limited range of length changes.

When only one load case is considered, the constraints 1–4 have to hold for both unloaded \((f = 0, u = 0)\) and loaded case. Several load cases can be handled simultaneously by extending the vector of design variables \(x\) with the nodal displacements \(u\) for the other loads, by using a linear combination of compliances as criterion, and by requiring constraints 1–4 to hold for the additional load cases also.

All nonlinear constraints are incorporated in a constraint vector

\[c \in \mathbb{R}^{(n_l+1)(\dim a_{n}+2a_{m}+n_{n}/2)},\]

where \(n_{n}/2\) is the number of bars and \(n_l\) the number of loaded cases. The part of \(c\) related to inequality constraints is required to be non-positive, and the part related to equality constraints is required to be zero. The actual number of constraints contained in \(c\) is slightly less than indicated above, due to the boundary conditions.

There are also a number of linear constraints and bounds. The sign requirement on \(\Lambda_0\) is implemented as a simple bound. A fixed volume, or mass, for the system is obtained with the linear equality constraint

\[\sum_i v_i = S_v, \quad v_i \geq 0\]

where we sum over all members of the structure. The linear constraints are specified directly by their coefficients and are not included in \(c\).
4 Solution tools and techniques

This nonlinear programming problem is solved with SNOPT 6.1-1(5) (Gill et al. 1999). This program employs a sequential quadratic programming (SQP) approach with active set strategy to solve the problem. The main reasons to choose SNOPT are its readily availability, its efficient implementation in Matlab, the advantage that SNOPT takes of sparsity, the continuing development of the program, and the excellent support.

The SNOPT software is extended with MAD (Forth 2001), a library of functions in Matlab, using a class library and operator overloading, for automatic differentiation, to compute the Jacobian $J$ of the constraint vector $c$. This is advantageous from a numerical point of view, because finite differencing is not needed anymore. The choice for MAD is based on an efficient sparse implementation, kind support, continuing development, and availability of source code (which has been slightly modified to fit our purposes).

To be able to solve the problem efficiently and reliably, several modifications are carried out. First, we discuss scaling issues to avoid ill-conditioning. Because scaling is done once, at the start of the optimization, and the problem is quite nonlinear, account has to be taken of the most appropriate scaling for the complete optimization run. Our approach was to make changes in the following order:

1. Use values of material parameters ($E$ and $\sigma$) that are not representative of physical values, e.g., choose $E = 100$ and $\sigma = 5$. When values representative of, e.g., steel are used, no convergence at all can be expected. This change brings the displacement $u$ closer to the values for the other design variables and should be accompanied by an appropriate increase in available volume $S_v$. The optimal values for other design variables are not influenced much, unless the ratio of $E$ and $\sigma$ is changed and the yield constraint is binding. As long as the ratio $E/\sigma$ does not change, the minimizer obtained, after scaling back the volumes $v$ and deformations $u$, represents the optimal physical solution when geometric nonlinearities are neglected.

2. Scale the objective so it is around 1.

3. Multiply the force coefficient constraint by $l^2$. This can be regarded as trying to express the constraints all in the same physical dimensions, in this case the same as the yield constraints.

4. Scale the constraints, so they have a magnitude of at most 1.

5. Scale constraints and design variables so the elements in the Jacobian are of the same order of magnitude, preferably around 1.

6. Balance the equilibrium constraints so they are slightly more important than the inequality constraints. When the equality constraints are not satisfied, the optimization criterion can be reduced drastically, and this is not what we want.

All these changes try to get a Jacobian whose singular values are not spread over too large a range and are close to 1, while giving some equations more importance than others. The difficulty lies in the fact that the appropriate scaling changes during the optimization due to nonlinearity. The optimal structure is also potentially ill-conditioned, e.g., because $v_i$ may approach zero for some members, leading to ill-conditioning being introduced during the process of optimization.
Other changes to the problem or solver are made.

1. Adding sets of slack variables. This can make some constraints linear, adds sparsity to the Jacobian, while the dimension of $c$ can be kept the same due to the use of simple bounds. The size of the vector $x$ of design variables is increased, however. This has been done for the constraint on the force coefficient $\Lambda$ in the loaded case.
2. To avoid convergence problems, the nodal positions $p$ are often constrained to lie in a box enclosing the support and load application nodes. This adds a set of simple bounds.
3. Changes to the default options of the solver, to improve the accuracy of the QP subproblem and of the line search.

A more fundamental problem arises from the nonlinearity of the equilibrium conditions. Under certain conditions, several solutions for these equations exist, and the optimization may drift away to an unwanted solution. The solution is normally unwanted when it cannot be reached following a path of stable equilibria during gradual load increases until the design load is reached. Normally this is solved by using path following algorithms, but these are expensive, certainly when embedded in an optimization. The solution chosen is to use two different amplitudes for the load, and force the solutions for $u$ to lie in a cone, a path constraint strategy. This heuristic prevents the problem in some cases, but gives no guarantees. Its advantage is the ease of incorporation in the problem formulation, which is done as follows. The vector $x$ is expanded with the displacements for the second load case, the constraints in $c$ are enforced for both load cases, and the cone is enforced by linear constraints on the displacements $u$. The objective still considers only the design load.

5 Design examples

We present several design examples where the solution is obtained straightforwardly, these are planar designs where $\dim$ is equal to 2, and a case that until now is not solved to satisfaction, a spatial design where $\dim$ is equal to 3.

5.1 Planar design

The basic design problem used to illustrate our approach is stiffness optimization for the tensegrity beam in Fig. 1. This beam structure

- is built up from 3 planar tensegrity crosses
- with an aspect ratio of 7
- while the support at the left side removes the 3 degrees-of-freedom of a rigid body for $\dim = 2$
- and is loaded by a unit vertical load at the top/right node.

For this example $n_m = 32$ and $n_n = 12$. Using the NLP formulation, we obtain the results in Fig. 2, see also (De Jager et al. 2002). The color coding in the figures is as follows.
Fig. 1. Basic tensegrity beam system (not optimized, in loaded state, showing deformation under load)

- unstressed member: light gray (green),
- pre-stressed bars: dark gray (red),
- pre-stressed tendons: black (blue).

Fig. 2. Optimal topology/geometry, showing deformation under load

In the shallow figure we do not allow the positions of the nodal points $p$ to move outside the horizontal lines at $y = 0$ and $y = 1$, so the $y$-components of $p$ are removed from $x$, while in the other figure the nodal points $p$ can move freely in the plane, except for the nodes at the support and for the node with the load. The size of $J$ is on the order of $200 \times 100$. Generating solutions for this problem only takes a few minutes, and convergence problems are hardly encountered.
We note the following

- optimizing topology/geometry improves the objective by more than 50%, and leaving the nodal points free to move is quite advantageous,
- for the optimized structures the sum of force coefficients is increased considerably, due to more short members, but all forces are within the failure constraints,
- in the generated optima some nodes move close to each other, this also causes members to be close to each other, so some members are hidden from view in the figures, and may introduce ill-conditioning, which is normally associated with small angles between members meeting at a node,
- approximately the same optimum value of the objective is obtained consistently when starting from different initial conditions, the resulting designs may differ quite a bit with respect to topology and geometry, showing the appearance of local optima.

We can compare these results with a theoretically optimal solution of Michell type (Michell 1904), obtained by dropping most of the constraints and only keeping those for the unloaded case, enabling a solution without pre-stress. This provides a comparison in Fig. 3.

Fig. 3. Optimal topology for an adjustable grid: comparison with Michell type truss (unstressed)
From this figure it appears that the additional pre-stress and constraint are, in this example, not so limiting to impose a severe performance penalty, because the difference in the objective (the compliance is better when lower), is only a meager 2%. It also provides confidence in the problem formulation and solution techniques.

5.2 Spatial design

In Fig. 4 a spatial design example is presented. This tensegrity structure is made of four 6-bar tensegrity units. It is supported at nodes 1, 6, 10, 11 (left-front) and loaded at the opposite end at nodes 43, 44, 45 (right-back-top) with a vertical force of magnitude $-0.1$. In the optimization the buckling constraints are not enforced and the $z$-coordinate of the nodal positions $p$ was removed from the design variables, because reasonable results for the full problem could not be obtained yet. In this case the cone constraint was not needed. The objective for the unoptimized case is computed by a finite element code assuming a constant stiffness matrix at an intermediate deformation state, for $p + u/2$, so it has been computed iteratively to capture some of the nonlinear effects.

The sparsity structure for the constraint Jacobian of this problem is given in Fig. 5. The matrix is quite sparse.

We note the following:

- the optimum is likely of local nature,
- the geometry of the optimal design tends to mimic the planar design, by concentrating most of the elements in the plane of the load vector,
- the structure will be sensitive to out-of-plane forces, which can be handled by taking account of multiple load cases,
- the structure becomes sensitive to (global) buckling, which is not accounted for in the present design,
- all these sensitivities make the problem potentially ill-conditioned,
- the solution was cumbersome to achieve, due to convergence problems,
- convergence was sometimes hampered due to the small step sizes resulting from the line search, which could be related to inaccurate line search directions generated from the QP subproblems,
- the QP subproblems were ill-conditioned in some subsets of the design space, requiring a large number of iterations to get a QP solution meeting the accuracy requirements,
- on the order of $10^5$ QP iterations were needed to converge,
- obtaining a solution may take a few hours and may need repeated human intervention,
- employing sparsity was a big advantage with respect to computing time, as can be concluded from Fig. 5,
- using automatic differentiation enabled a more reliable solution process, with finite differences more convergence problems occurred, although small errors in the Jacobian sometimes seemed to get around some local optima.
6 Discussion

There are several disadvantages in the tools that are used to solve the optimization problem used in this paper and in the formulation of the problem.

First, SQP type techniques are useful when the constraints are almost linear and the objective is quite nonlinear. For the objective, second order information is employed, the Hessian, for the constraints only first order information, the Jacobian. In the current formulation, however, the objective is linear in $x$, but the constraints are quite nonlinear, making the problem potentially difficult to solve. This can also be attributed to problems with closely following the curved manifolds of the constraints. Perhaps constraint satisfaction programs can improve this. For an SQP approach it may be of advantage to incorporate second order information for the constraints in the optimization procedure, or to reformulate the problem so constraints are more linear and nonlinearities are moved to the objective.
Second, ill-conditioning in finite element type models, such as the one employed here, are normally attributed to members that almost overlap, implying small angles between members that meet at a node. When approaching an optimum, sometimes members are folded together in the solution, implying small angles, essentially indicating that they can be removed. As long as their volumes do not go to zero, they will effect the conditioning of the model. Using a multi-phase approach, with global optimization carried out first, which may be crude, a subsequent local optimization may start with those folded part already removed, improving the possibility for speedy convergence.

Third, scaling issues in nonlinear programming are a nuisance, because during the optimization the scalings need to be adapted, and this requires too much human intervention. Also, when the design specifications change the scaling has to be carried out again. Adaptive scaling could better be handled internal to the optimization procedure.

Furthermore, the formulation of the problem still leads to solutions which are not attractive. This is due to, e.g., manufacturing constraints, that require a limited set of dimensions, i.e., bar diameters, making the problem of mixed-integer type and still harder to solve in general.

Other issues are the need of introducing symmetry and fractality in the design, to make it more esthetically pleasing, and easier to control the shape of the structure. This may be achieved by a sub-structuring technique, which can also be used to reduce the
dimensionality of the problem, but requires a solution for several levels of abstraction on the model of the structure. In hindsight, it is questionable if the proposed formulation is well suited. Feedback from different fields is needed, asking for multidisciplinarity in setting up and solving the design problem.

Acknowledgment

Kenneth Holmström and Michael A. Saunders provided useful observations and suggested and tested modifications of the optimization problem.

References