Generation of Articulated Mechanisms by Optimization Techniques

Kawamoto, Atsushi

Publication date:
2004

Document Version
Publisher's PDF, also known as Version of record

Generation of Articulated Mechanisms by Optimization Techniques

by

Atsushi Kawamoto

Department of Mathematics

Technical University of Denmark

Title of Thesis:

Generation of Articulated Mechanisms by Optimization Techniques

Ph.D. Student:

Atsushi Kawamoto

Department of Mathematics, Technical University of Denmark

Address: Matematiktorvet B303, DK-2800 Kgs. Lyngby, Denmark

Email: A.Kawamoto@mat.dtu.dk

Supervisors:

Martin P. Bendsøe

Department of Mathematics, Technical University of Denmark

Address: Matematiktorvet B303, DK-2800 Kgs. Lyngby, Denmark

E-mail: M.P.Bendsoe@mat.dtu.dk

Ole Sigmund

Department of Mechanical Engineering, Technical University of Denmark

Address: Nils Koppels Allè B404, DK-2800 Kgs. Lyngby, Denmark

E-mail: Sigmund@mek.dtu.dk

Preface

This thesis is submitted in partial fulfillment of the requirements for obtaining the degree of Ph.D. The research work has been carried out at the Department of Mathematics, Technical University of Denmark in the period from September 2001 to November 2004 under the supervision of Professor dr.techn. Martin P. Bendsøe and Professor dr.techn. Ole Sigmund. I am very grateful to my two supervisors for their kind help and valuable advice.

I wish to acknowledge that the Ph.D. study has been financially supported by Toyota Central R&D Labs., Inc. I would like to thank President Norikatsu Ishikawa for accepting the project and Director Noboru Kikuchi for encouraging me to come to Denmark. I would also like to thank Director Akinori Saito and my superior officers Nobuyuki Mori and Mizuho Inagaki for their official and private support.

I would like to thank Mathias Stolpe and the rest of TopOpt group at Technical University of Denmark (www.topopt.dtu.dk) for many valuable discussions on the subjects presented in the thesis. A thank also goes to Krister Svanberg at Royal Institute of Technology, Sweden for permission to use his software implementation of the MMA algorithm.

The period from December 2002 to January 2003 was spent at University of Heidelberg, Germany. A special thank goes to Jens Starke who was my host. Conversations with him and his students contributed to this project.

I wish to gratefully acknowledge many stimulating and instructive conversations on mathematics with Carsten Thomassen, Jens Gravesen, Tom Høholdt, Michael Pedersen, Vagn L. Hansen and the rest of my fellow colleagues at the Department of Mathematics. I truly appreciate the friendly and cooperative atmosphere of the department.

I would like to express my special gratitude to Anders Bisgaard. It has been *hyggelig* to share the office with him. Finally, I would like to thank my family for all their support during the period.

Kgs. Lyngby, November 2004

Atsushi Kawamoto

Abstract

This thesis deals with computer-based methodologies for designing articulated mechanisms. An articulated mechanism consists of several links (or bars) connected by joints and it gains all of its mobility from the joints. This is in contrast to a compliant mechanism which is a different type of mechanism that gains some or all of the mobility from the elasticity of its components.

A truss ground-structure approach is taken for the optimization of such mechanisms. This allows for an efficient analysis of the properties of the mechanism and is a suitable basis for an optimization procedure that handles topological variations. In this thesis the technique is also extended so as to design the shape as well as the topology of the truss using cross-sectional areas and nodal positions as design variables. This leads to a technique for simultaneous type and dimensional synthesis of articulated mechanisms.

A critical issue for designing articulated mechanisms is geometric non-linearity in the kinematics. This should be considered for the analysis of mechanisms since the displacements of mechanisms are intrinsically large. The consideration of geometric non-linearity, however, causes numerical difficulties relating to the non-convexity of the potential energy defining the equilibrium equations. It is thus essential to implement a numerical method that can – in a consistent way – detect a stable equilibrium point; the quality of the equilibrium analysis not only is important for itself but also directly affects the result of the associated sensitivity analysis. Another critical issue for mechanism design is the concept of mechanical degrees of freedom and this should be also considered for obtaining a proper articulated mechanism. The thesis treats this inherently discrete criterion in some detail and various ways of handling constraints related to mechanical degrees of freedom are suggested.

The thesis consists of the following four parts corresponding to the four thesis papers that are the main material of the thesis:

1. Graph-theoretical enumeration [Paper 1]
2. Gradient-based local optimization [Paper 2]
3. Branch and bound global optimization [Paper 3]
4. Path-generation problems [Paper 4]

In terms of the objective of the articulated mechanism design problems, the first to third papers deal with maximization of output displacement, while the fourth paper solves prescribed path generation problems. From a mathematical programming point of view, the methods proposed in the first and third papers are categorized as deterministic global optimization, while those of the second and fourth papers are categorized as gradient-based local optimization. With respect to design variables, only cross-sectional areas are used in the first and second papers, whereas both cross-sectional areas and nodal positions are simultaneously optimized in the third and fourth papers.

Abstrakt (in Danish)

Denne afhandling omhandler computer-baserede metoder til at designe mekanismer, der udfører deres funktion alene gennem bevægelige led og uden udnyttelse af en eventuel fleksibilitet af elementerne. En sådan mekanisme består således af flere ikke-fleksible elementer (stænger) forbundet af led, og mekanismen opnår al bevægelse fra dens bevægelige led. I modsætning hertil opnår såkaldte fleksible mekanismer en mulig bevægelse fra elasticiteten af dens komponenter.

I denne afhandling behandles designoptimeringproblemet gennem en beskrivelse af de mulige mekanismer via en gitterkonstruktion, som udgør en grundstruktur ud fra hvilken den optimale topologi kan findes; grundstrukturen definerer rummet af mulige mekanismer. Teknikken er i afhandlingen udvidet til at kunne håndtere såvel form som topologi, idet både elementernes tværsnitsarealer og knudepunkternes positioner benyttes som designvariable. Dette resulterer i en metodik, som tillader samtidig syntese af type og dimensioner for bevægelige led-mekanismer.

Et centralt emne for design af bevægelige led-mekanismer er hensyntagen til geometrisk ikke-linearitet. Dette er et nødvendigt aspekt af analysen af mekanismer idet forskydningerne under bevægelse er store. Den geometriske ikke-linearitet er imidlertid årsag til visse numeriske vanskeligheder relateret til ikke-konveksitet af den potentielle energi, der bestemmer ligevægtsligningerne. Derfor er det essentielt at implementere en numerisk metode, som sikkert kan detektere en stabil ligevægt; kvaliteten af ligevægtsanalysen er ikke kun i sig selv vigtig, men er ligeledes af stor vigtighed for den associerede følsomhedsanalyse som er central for optimeringsprocessen. Et andet kritisk koncept for mekanismer er antallet af frihedsgrader og en angivelse af dette er et vigtigt element i at opnå en ren ledmekanisme. Afhandlingen foreslår en række muligheder for at man i optimeringsproblemet kan tage hensyn til dette kriterie, der i sin natur er heltalligt.

Afhandlingen består af en oversigtsartikel og fire delafhandlinger (artikler), der behandler følgende emne:

1. Grafteorisk karakterisering [Artikel 1]
2. Gradient-baseret lokal optimering [Artikel 2]
3. Branch og bound global optimering [Artikel 3]
4. Kurvegenereringsproblemer [Artikel 4]

Hvad angår objektfunktionen for mekanismedesignproblemet, omhandler den første til tredje artikel maximering af forskydningen ved en udgangsknude, mens den fjerde artikel behandler problemer hvor en given kurve er foreskrevet for en udgangsknude. Fra et matematisk programmerings synspunkt, falder metoderne foreslået i den første og tredje artikel i kategorien “deterministisk global optimering”, mens den anden og fjerde artikel benytter gradient-baseret lokal optimering. Med hensyn til designvariable benyttes i den første og anden artikel kun stangtværsnittets arealer, mens dette kombineres med brug af knudepunktspositionerne i en samtidig optimering i den tredje og fjerde artikel.

Publications

International journals:

[J1] A. Kawamoto, M.P. Bendsøe, and O. Sigmund. Planar articulated mechanism design by graph theoretical enumeration. *Structural and Multidisciplinary Optimization*, 27(4):295–299, 2004. [Paper 1]

[J2] A. Kawamoto, M.P. Bendsøe, and O. Sigmund. Articulated mechanism design with a degree of freedom constraint. *International Journal for Numerical Methods in Engineering*. 61(9):1520–1545, 2004. [Paper 2]

[J3] M. Stolpe and A. Kawamoto. Design of planar articulated mechanisms using branch and bound. *Mathematical Programming*, Accepted, subject to minor revisions. [Paper 3]

[J4] A. Kawamoto. Path-generation of articulated mechanisms by shape and topology variations in non-linear truss representation. Submitted. [Paper 4]

Conference proceedings:

[C1] A. Kawamoto, M.P. Bendsøe, and O. Sigmund. Articulated mechanism design by an enumeration approach. 15th Nordic Seminar on Computational Mechanics (NSCM15), Aalborg, Denmark, 18–19 October 2002.

[C2] A. Kawamoto, M.P. Bendsøe, and O. Sigmund. Articulated mechanism design—introduction of DOF constraint. 5th World Congress of Structural and Multidisciplinary Optimization (WCSMO5), Venice, Italy, 19–23 May 2003.

[C3] Atsushi Kawamoto and Mathias Stolpe. Design of articulated mechanisms with a degree of freedom constraint using global optimization. 21st International Congress of Theoretical and Applied Mechanics (ICTAM2004), Warsaw, Poland, 15–21 August 2004.

[C4] A. Kawamoto. Simultaneous type and dimensional synthesis of articulated mechanisms for path-generation, 17th Nordic Seminar on Computational Mechanics (NSCM17), Stockholm, Sweden, 15–16 October 2004.

Contents

[Paper 1] Planar articulated mechanism design by graph theoretical enumeration. *Structural and Multidisciplinary Optimization*, 27(4):295–299, 2004.

[Paper 2] Articulated mechanism design with a degree of freedom constraint. *International Journal for Numerical Methods in Engineering*, 61(9):1520–1545, 2004.

[Paper 3] Design of planar articulated mechanisms using branch and bound. *Mathematical Programming*. Accepted, subject to minor revisions.

[Paper 4] Path-generation of articulated mechanisms by shape and topology variations in non-linear truss representation. Submitted.

1 Introduction

The history of machine mechanism design probably dates back as early as the beginning of our machine civilization; by *mechanism* we here mean a mechanical device that can transfer motion and/or force from a source to an output [22]. The patenting of the Watt steam engine in 1769 is viewed as a pivotal event for the ensuing Industrial Revolution [36]. Although James Watt [34] is sometimes incorrectly credited as the inventor of the steam engine, by the time of his birth vast numbers of Newcomen engines* were already pumping water from deep mines all over Great Britain. James Watt was an artisan and made miniature models of steam engines for scientists. In 1764 he was given a model of a Newcomen engine to repair and realized that it was hopelessly inefficient. Shortly afterward he came up with his own revolutionary design for a steam engine. As the story goes, a good design improves our life and an innovative design may change the world.

Design activities are creative efforts through which designers pursue higher quality and value of products. Therefore, the activities intrinsically involve a process of iterative improvement and adjustment, namely optimization. Visiting an automobile museum, it is interesting to see how vehicles evolved through history reflecting the change of the world affairs such as the oil crisis and the regulation of exhaust gas. And it is all the more interesting to know how the designers came across the designs. In a sense the history of our machine civilization is the history of design optimization. The optimization was carried out by individuals or groups within a short period or between generations. For the nature of design activities, it seems quite natural that computer-based optimization techniques are playing a more and more important role in industrial design processes.

Let us shift our focus to more modern circumstances in the automobile industry. Figure 1 shows an outline of an automobile development process from the initial product planning to the final mass production. The development process requires collaboration among several different divisions: body styling division, body structure division, power train division, drive train division, etc. There are typically two major design stages, namely basic design and detailed design. In the basic design, a main layout of components is determined according to the preceding product planning and conceptualization. For detailed design, each component is precisely designed by the corresponding design divisions. By the end of this stage, a complete drawing should be finished for the following experimental prototyping. The loop depicted by *L1* indicates feedback from the experimental results to the detailed design. This loop is normally repeated more than once until the final design is settled. In the worst case, the basic design should be reviewed in the loop *L2*. It is a costly loop that should essentially be precluded by maturing the initial basic design in advance because the change of the basic design affects every relating division and may eventually delay the whole schedule. The loop *L3* is a rather new design loop for virtual prototyping due to the growth of computer simulation technologies. Under the pressure to reduce the total development time, the number of loops *L1* is reduced or replaced by the loop *L3*. This trend seems to get accelerated year by year.

Commercial computer-aided design (CAD) systems such as AutoCAD® and CATIA® have been successfully used for so long in the aerospace and automobile industries that almost all mechanical components are designed using CAD systems. On the other hand, computer-aided engineering (often abbreviated as CAE) is a broad term describing the use of computer technology to assist engineers in their design activities. In the narrow sense CAE indicates computer simulation technologies such as finite element analysis and multibody system analysis. Nowadays, CAD and CAE (also CAM: computer-aided manufacturing) are integrated into a software suite that merges many different aspects of the product life-cycle management (PLM) including design, production planning, product testing, visualization and product documentation etc. Several advanced CAE packages now also provide optimization tools. With reference to the current status of optimization tools in commercial CAE packages, structural optimization using finite element formulation is at a much more advanced stage than mechanism optimization using multibody system formulation. This is because the finite element formulation is suitable for the analytical sensitivity analysis which is essential for efficient implementation of optimization. In contrast, for multibody system formulations that typically end up with differential algebraic equations (DAE) [33] it is extremely difficult to generalize the sensitivity analysis (see, e.g., [32, 61]). In practice one must quite often rely on finite difference sensitivity analysis, thus repeating the analysis as many times as the number of the design variables for the forward (or backward) difference sensitivities and twice as many times for the central difference sensitivities. Unfortunately

*The first commercially successful steam engines manufactured for more than a hundred years: the engines were rugged and reliable and worked day and night, but were less than one per cent efficient, used a lot of coal, and consequently were first installed in coal-mines [35].

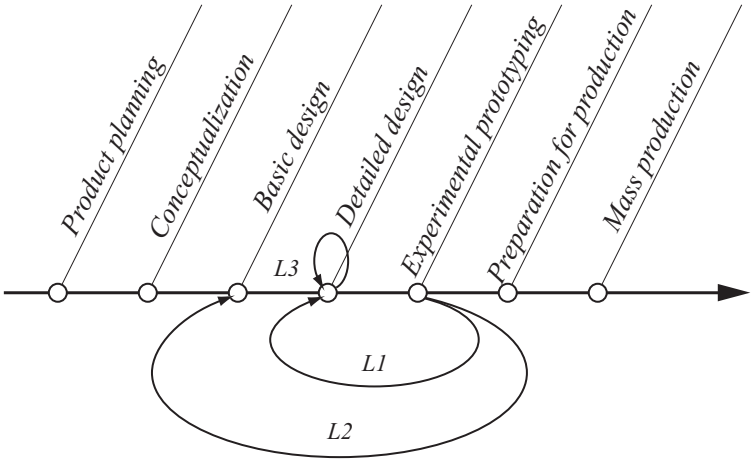


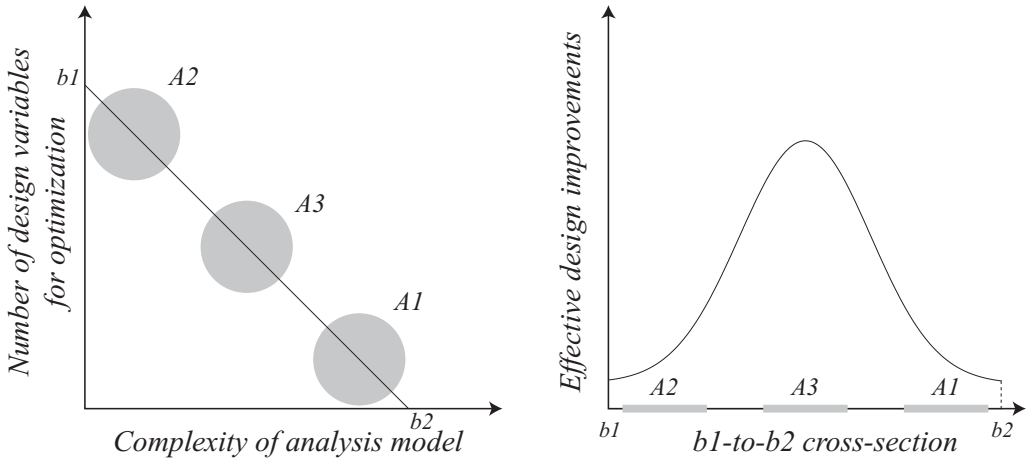
Fig. 1 A automobile development process

multibody system analysis is not a cheap calculation since it involves expensive time integration. It is thus not surprising that, compared to structural optimization, the optimization of multibody systems is rather new. There are several inspiring pilot research papers [12, 13, 19, 29, 30, 49] on this challenging area but the subject seems to be still open and there is much work to be done.

From a practical optimization point of view, one aspect we must keep in mind is that we have a time limit for designing a product and that computer resources are not unlimited, however faster computers become. Figure 2(a) shows a relationship between the complexity of analysis models in the horizontal axis and the number of design variables for optimization in the vertical axis. In general, if we consider more details of a phenomenon, the model complexity increases, and the analysis takes more time. Details are, of course, important especially when we are finalizing the design in the last stage. The area *A1* indicates the extreme case. But if the analysis takes so long, we cannot use many design variables. By contrast, the area *A2* indicates the other extreme case where the analysis model is very simple. If the model is so simple that one function evaluation is almost for nothing, we can apply more aggressive optimization techniques such as global optimization that allows for a more drastic search, possibly to global optimality. However, if it is too simple to maintain the realistic solution, we have no effective feedback from the calculation. That is why it is very important to find the decisive factors for obtaining a meaningful solution to the real problem. In other words, the model should be as simple as possible but not simpler. In that sense, this issue should be discussed from another dimension, namely *effective design improvements*; as shown in Figure 2(b) there must be a trade-off for exploiting the computer resources by properly distributing the computer resources. The area *A3* indicates such an ideal situation.

This thesis specifically deals with computer-based methodologies for designing *articulated mechanisms*. An articulated mechanism consists of several links (or bars) connected by joints and gains all the mobility from its joints. This is in contrast to a *compliant mechanism* which is a different type of mechanism that gains some or all of the mobility from the elasticity of its components. Figure 3 shows examples of these two types of inverter mechanisms. Apart from the apparent difference of shapes, there is another subtle difference between them, namely the support in the circle of the articulated inverter mechanism. Without this support the articulated mechanism would collapse, suggesting that for obtaining a proper articulated mechanism a crucial factor is the number of mechanical degrees of freedom (DOF). This is defined as the number of independent inputs required to determine the position of all links of the mechanism with respect to a global coordinate system [22]. It is, however, an insignificant feature for compliant mechanism design as such mechanisms do not have real joints.

The targeted design phase of work of this thesis is somewhere between conceptualization and basic design, namely before the design is completely parameterized. The use of optimization techniques in this stage of mechanism design is very rare. However, it is clear that we have more freedom for designing mechanisms in the earlier stages than the latter. In this early stage, furthermore, designers must sometimes



(a) Distribution of computer resources

(b) Effective design improvements

Fig. 2 Trade-off for exploiting computer resources

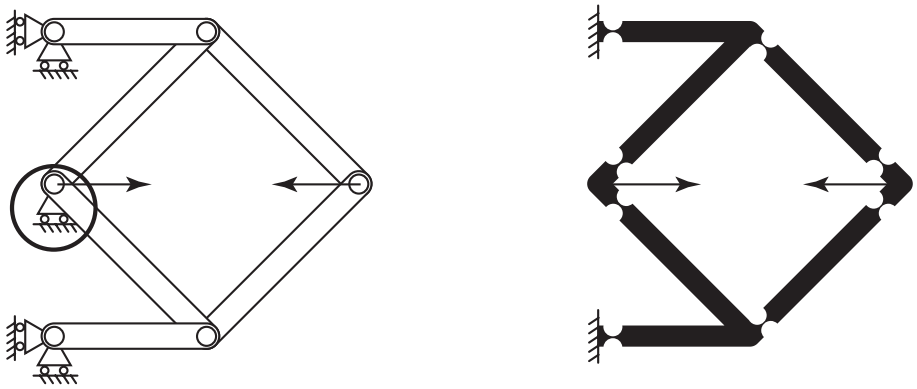


Fig. 3 Articulated (left) and compliant (right) inverter mechanisms

determine important and decisive design factors which may be difficult to change later such as the basic layout of mechanism components. But this does not mean that the optimization in the latter stage is less important. On the contrary, it is also very important and difficult to optimize complicated detailed models while observing many design constraints (sometimes contradictory between different components) and help designers determine exact dimensions. Designers or engineers have to be very careful about changing design in the latter part of the development in order not to interfere with other design requirements. Quite often it ends up with a very conservative design change that has a surprisingly modest effect.

The central issue in the first step of mechanism design is kinematics. Many basic ideas in kinematics, which we almost take for granted, originate in the work done by Franz Reuleaux, often called the father of kinematics [46]. He noticed the importance of the topological aspect of mechanisms and he conceived a way of categorizing a plethora of mechanism inventions in the 19th century. One of those inventions was the straight line mechanism shown in Figure 4 [1]. (At that time, it was believed impossible to draw an exact straight line using a limited number of links. But in 1864 a French engineer Charles-Nicolas Peaucellier showed that the mechanism could transform the circular motion at the middle joint to an exact straight-line motion at the right end joint. In the fourth thesis paper this problem is treated as an example but in slightly different proportion.) Reuleaux was also the first to attempt to establish a mathematical framework on kinematic synthesis of mechanism design. Before Reuleaux, mechanism design was something engineers did by trial and error, but it was not a subject that scientists could talk about. His work was influential in

Fig. 4 Peaucellier straight-line mechanism (credit line: Jon Reis/www.jonreis.com)

those days and was carried on by his students and successors.

Nowadays, focusing on kinematic synthesis, a standard and systematic mechanism design procedure is divided into the following three steps, see, e.g., [21, 22]. Firstly, an objective is set for the mechanism to be designed. The objective may, for example, be the maximization of output displacement or generation of a prescribed output path for a given input. The first step may correspond to the conceptualization in Figure 1. Secondly, mechanism properties such as the number of bars and joints and their connections are determined. This step is called *type synthesis* and is mainly concerned with topological questions. The type synthesis should be done between the conceptualization and the basic design stages. An elaborate type synthesis technique based on number synthesis is developed in [59, 60]. In that approach much effort is devoted to generating a complete list of kinematic chains by eliminating isomorphic graphs (a significant unsolved problem in number synthesis). Finally, the exact dimensions of all the mechanical components are determined. This step is called *dimensional synthesis* and is mostly related to geometrical questions. The dimensional synthesis typically starts in the latter part of the basic design and is repeated until the detailed design is completed. The above mentioned multibody-system-based optimization should play a key role in this step. Reviews on dimensional synthesis using optimization techniques are found in [28, 45]. Although type and dimensional syntheses are normally separated in the engineering context, they are actually so closely connected that they should preferably be treated at the same time.

The optimization of geometry and topology of a structural layout has a great impact on the performance of a structure. In the structural optimization society, a methodology called *topology optimization* has been attracting many researchers since it was shown that this can be handled numerically by a material distribution method [10]; this is a generalization of the ground-structure approach for truss topology optimization as pioneered by, e.g., Dorn and Gomory [18], while the layout theory for continua as originally developed by Michell [44], and extended by, e.g., Prager and Rozvany [52]. It goes without saying that the significant feature of the technology is that, unlike other shape and sizing optimization methods, the method can optimize simultaneously both the topology and geometry of a structural layout. The topology optimization has been extended and developed into many directions. One of the most successful areas is design of compliant mechanisms. Research on design of compliant mechanisms [3, 14, 50, 55] using topology optimization techniques in continuum structures suggests that it should also be possible to obtain results for articulated mechanisms by applying such techniques [25, 54] and that the techniques have good potential for devel-

oping into a systematic method that can design articulated mechanisms combining type and dimensional syntheses.

A common critical factor for designing both compliant and articulated mechanisms is geometric non-linearity. This should be considered for the analysis of mechanisms since the displacements of mechanisms are intrinsically large. The consideration of geometric non-linearity, however, causes numerical difficulties relating to the non-convexity of the equilibrium equations. It is here essential to implement a numerical method that can consistently detect a stable equilibrium point because the quality of the equilibrium analysis directly affects the result of the associated sensitivity analysis. Another critical factor specific to articulated mechanisms is the number of *mechanical degrees of freedom* as is already argued above.

This introductory note to the four thesis papers is organized as follows. Section 2 presents a collection of theoretical aspects included in the accompanying thesis papers. The section consists of seven different subjects. Among them, the first five subjects are commonly or repeatedly used throughout the thesis papers. The sixth and seventh subjects are specific to the topics in [Paper 3] and [Paper 4], respectively. Section 3 is intended to give a perspective of the whole research work by summarizing the main results of the thesis papers. The section contains four topics that are presented in (almost) chronological order in which the research work developed. In order to bridge the gaps between the independent topics, some additional results and further discussions are included. Section 4, finally, contains conclusions and possible directions for future research in this area.

As regards the notations, I must beg your forgiveness and patience in advance about the discrepancies between the thesis papers. In order to keep the overall coherence within this introductory note, some notation used here might be somewhat different from the thesis papers.

2 Theoretical aspects

This section presents the following seven different theoretical subjects that constitute the basis for the mechanism design methods proposed in the thesis papers:

1. Truss representation
2. Ground-structure approach
3. Equilibrium analysis
4. Stability analysis
5. Degree of freedom analysis
6. Branch and bound algorithm
7. Configuration space

The first to fifth subjects are all common throughout the thesis papers. The sixth and seventh subjects are specific to the topics in [Paper 3] and [Paper 4], respectively. The first two subjects constitute a basic approach to the design of articulated mechanisms within the framework of truss topology design. The third to fifth subjects deal with three different types of analysis problems that should be considered for obtaining proper mechanism designs.

2.1 Truss representation

Kinematic diagrams (or skeleton diagrams) are the schematic drawings commonly used for conventional mechanism designs. They are especially helpful for the understanding of complicated mechanism topologies [22]. It seems natural to represent kinematic diagrams by truss elements and pin joints as one obtains the following advantages:

- One can directly interpret the truss topology from the kinematic diagram.
- It is possible to accommodate extremely large displacement considering geometric non-linearity without any element distortion problems.

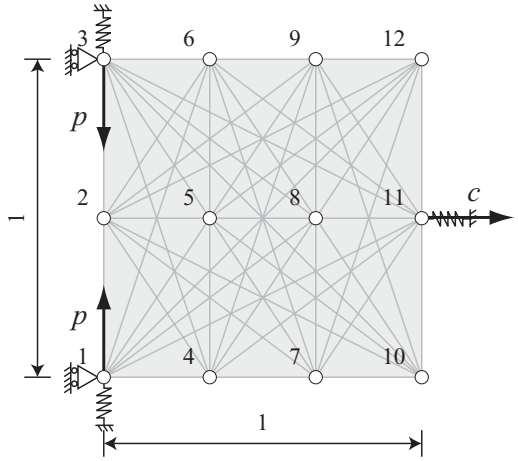


Fig. 5 Ground-structure for the benchmark example

- With this interpretation one can formulate the optimization problems in terms of the so-called ground-structure approach, discussed in Section 2.2, for truss topology design [11].

Throughout the thesis, geometric non-linearity is considered using the Green-Lagrange strain measure. This is essential for the analysis of articulated mechanisms since the displacements of an articulated mechanism are intrinsically large. The geometric non-linear analysis problem is discussed in Section 2.3.

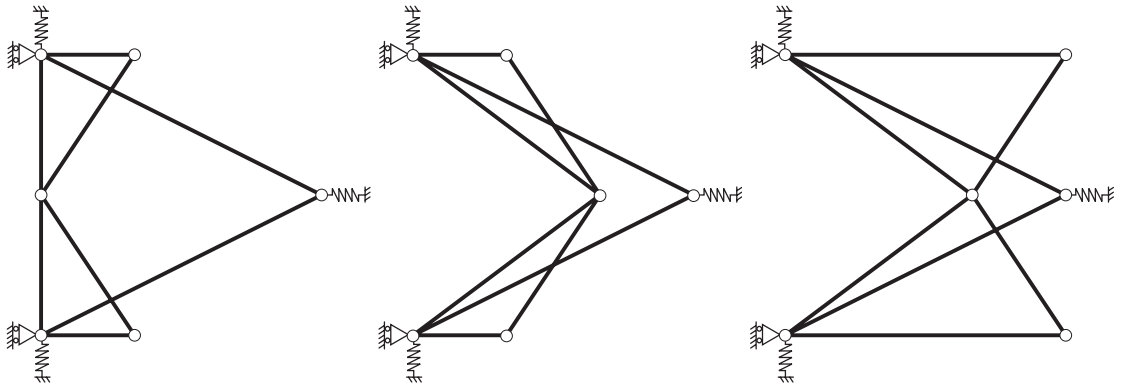
Normally articulated mechanisms are modeled by rigid bars and revolute joints. Nevertheless, a truss representation is used for articulated mechanisms in the thesis papers. But, by introducing the degree of freedom constraint discussed in Section 2.5, the articulated mechanisms are not supposed to rely on the elasticity in the final designs. In other words, the function of the articulated mechanisms does not change after all the truss elements are replaced by rigid bars. This is the critical difference between articulated and compliant mechanisms.

2.2 Ground-structure approach

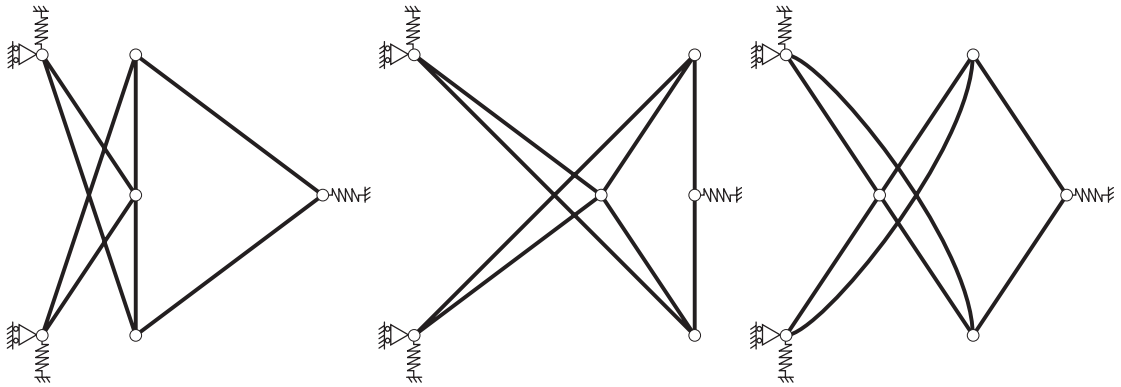
In order to formulate the mechanism design problem, the so-called *ground-structure approach* is adopted. The idea of the truss ground-structure approach is: first, give all potential elements in the design domain; then, eliminate unnecessary elements according to the requirements (or, equivalently, choose the necessary elements). The ground-structure approach was originally introduced in 1960s [18, 23]. Since then, it has been applied to numerous truss topology design problems, see, e.g., [2, 5, 7, 8, 9].

Figure 5 shows a truss ground-structure for a benchmark example which is intensively used for the development of the mechanism design methods throughout the thesis papers. The truss ground-structure contains possible nodes and connections, boundary conditions, external loads, and dimensions. The ground-structure consists of $N = 12$ nodes and $n = 66$ potential bars. The design domain is depicted by the grey unit square area. The external load denoted by p is applied at node 1 and 3 simultaneously. The nodes at which the external load is applied are called the input ports. The node at which the desired output should take place is named the output port; in this example it corresponds to node 11. The direction of the desired output is denoted by c. The springs and forces at the input ports model strain based actuators (e.g. shape memory alloys or piezoelectric devices) and the output spring simulates resistance of movement. This approach was originally introduced in [55].

The benchmark mechanism design problem is the maximization of the output displacement $c^\top u$ for the given input forces p by picking eight bars out of the $n = 66$ possible connections, so that the resulting mechanism has exactly one mechanical degree of freedom (discussed later in Section 2.5) and is symmetric with respect to the horizontal line between nodes 2 and 11. The displacement vector u is a solution to the non-linear equilibrium equations discussed in Section 2.3. For the sake of some insight into this mechanism design problem, six different candidate mechanisms with eight bars and one mechanical degree of freedom are displayed in Figure 6. The curved lines in Figure 6(f) represent overlapping bars.



(a) Unstable (0.118) (b) Stable (0.118) (c) Stable (0.118)



(d) Stable (0.791) (e) Unstable (0.820 or −0.05) (f) Stable (0.778)

Fig. 6 Some candidate mechanism designs for the benchmark example

A very important concept in the mechanism design problem is elastic stability. In the benchmark example instability can be characterized by buckling of sub-structures. This is a global phenomenon in contrast to the local Euler buckling of long thin bars. See, for example, the two left-most vertical bars in Figure 6(a). If a small force is applied as indicated in the ground-structure, these two bars will be compressed. In this situation their common node, labeled 2 in Figure 5, will initially not move. As the applied force is increased, the node will suddenly move (or snap) either to the right or left; the direction is unpredictable since it depends on imperfections in the design.

In Figure 6 there are indications of "stable" or "unstable" for each mechanism together with its potential output displacement. The mechanism shown in Figure 6(e) is the other unstable mechanism since the two right-most vertical bars must buckle before movement of the output port is possible; in this case, the output displacement becomes either positive (0.820) or negative (−0.05). This situation is undesirable because we want the designed mechanism to function with certainty. The condition on the truss stability can be modeled by a matrix inequality discussed in Section 2.4.

Even though six different topologies with completely different behavior under loading are depicted in Figure 6, the connectivity of the mechanisms shown in Figure 6(a) - 6(c) and Figure 6(d) - 6(f) respectively are very similar. In fact, if the nodal positions of the three active nodes where no springs are attached are allowed to change, the topologies in Figure 6(a) - 6(c) (and 6(d) - 6(f) respectively) can be made exactly the same (they are topologically equivalent). This feature is fully utilized in the graph-theoretical enumeration summarized in Section 3.1 that enables us to work out all possible symmetric mechanisms consisting of eight bars with one degree of freedom.

2.3 Equilibrium analysis

For the analysis of mechanisms, geometric non-linearity should be taken into account since the displacements of mechanisms are intrinsically large. When considering geometric non-linearity, however, it becomes a non-trivial task to obtain a reasonable equilibrium point due to the non-convexity of the equilibrium equations. It is essential to implement a numerical method that can consistently detect a stable equilibrium point because the quality of the equilibrium analysis directly affects the result of the associated sensitivity analysis.

We begin with an energy form in order to obtain a stable equilibrium point. In the articulated mechanism design problems, the truss topology is expressed by a binary truss connectivity vector which is defined as a collection of normalized cross-sectional areas $a \in \mathbb{B}^n$, where n is the number of potential bars in the ground-structure. (In this thesis $\mathbb{B}^n$ denotes the set of n-dimensional binary vectors, namely $\{0,1\}^n$.) The nodal positions are denoted by $x \in \mathbb{R}^{2N}$, where N is the number of nodes (frictionless joints) in the ground-structure. Furthermore, the state variables in the model are collected in the vector $u \in \mathbb{R}^d$ containing the nodal displacements of the design under the external load. The dimension of the displacement vector d is the number of degrees of freedom (in the finite element method sense) of the structure which for a two-dimensional truss $d = 2N - d_f$, where d_f is the number of prescribed fixed displacements. The total potential energy $\Pi(a,u,x)$ of a truss subjected to the external static nodal force vector $p \in \mathbb{R}^d$ is given by

$$\Pi(a,u,x) = \frac{1}{2}u^\top K_0 u + \frac{1}{2}\sum_{j=1}^{n} E_j l_j(x) a_j \epsilon_j(u,x)^2 - p^\top u, \tag{1}$$

where $K_0 \in \mathbb{R}^{d\times d}$ is the constant stiffness matrix due to linear springs attached to some of the nodes in the truss, $E_j > 0$ is the Young's modulus in the j-th bar, $l_j(x)$ is the undeformed length, and $\epsilon_j(u,x)$ is the strain (defined below) of the j-th bar, respectively.

For a given topology $a \in \mathbb{B}^n$ and geometry $x \in \mathbb{R}^{2N}$, the displacement vector $u \in \mathbb{R}^d$ should be a minimizer of the potential energy. In order to find a stable equilibrium, we need to find a local minimizer or at least a stationary point of $\Pi(a,u,x)$ such that the Hessian of the potential energy with respect to u is positive semidefinite. The problem of finding such a u is called the analysis problem. The first order optimality conditions for this problem are given in the residual form of *equilibrium equations*:

$$\nabla_u \Pi(a,u,x) = 0, \tag{2}$$

while the second order conditions, in the following called the *stability conditions*, are given by

$$\nabla^2_{uu} \Pi(a,u,x) \succeq 0. \tag{3}$$

Note that $\Pi(a,u,x)$ in general is a non-convex functional which may have no such stationary points. Alternatively, there may exist several displacement vectors u satisfying both (2) and (3) for given $(a,x)^\top$.

In order to consider geometric non-linearity, the following assumptions are made in the mechanical modeling:

1. Large displacements are allowed but the strains are assumed to be small.
2. The strain is given by the Green-Lagrange strain measure.
3. The material is linearly elastic and the material properties remain constant.

The Green-Lagrange strain measure $\epsilon_j(u,x)$ is defined as

$$\epsilon_j(u,x) = \frac{\hat{l}_j^2(u,x) - l_j^2(x)}{2l_j^2(x)}, \tag{4}$$

where $\hat{l}_j(u,x)$ is the deformed length of the j-th bar (see, e.g., [17]). In order to rewrite this strain measure, we first introduce a symmetric positive semidefinite matrix $B_j \in \mathbb{R}^{2N\times 2N}$ which is defined such that the undeformed length of the j-th bar is given by $l_j(x) = (x^\top B_j x)^{1/2}$. The non-zero part of B_j is given by (in local coordinates corresponding to the position of the end nodes of the considered bar):

$$B_j = \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix}.$$

The matrix $\tilde{B}_j$ is the symmetric positive semidefinite $d \times d$ matrix that remains after removing the rows and columns corresponding to the fixed degrees of freedom from B_j. The matrix $\hat{B}_j$ is the $d \times 2N$ matrix which remains after removing the rows corresponding to the fixed degrees of freedom from B_j

The Green-Lagrange strain in the j-th bar may now be reformulated as

$$\epsilon_j(u, x) = \frac{1}{l_j^2(x)} b_j^\top(x) u + \frac{1}{2l_j^2(x)} u^\top \tilde{B}_j u, \tag{5}$$

where $b_j(x) = \hat{B}_j x \in \mathbb{R}^d$. Then the vector $b_j(x)/l_j(x)$ contains the direction cosines of the j-th bar, $b_j^\top(x)u/l_j(x)$ is the linearized elongation of the bar, and $b_j^\top(x)u/l_j^2(x)$ is the linear (small deformation) strain. The potential energy can, using the Green-Lagrange strain (5), be written as

$$\Pi(a, u, x) = \frac{1}{2} u^\top \left(K(a, x) + G(a, u, x) + H(a, u, x)\right) u - p^\top u,$$

where the positive semidefinite matrix $K(a, x) \in \mathbb{R}^{d \times d}$ is given by

$$K(a, x) = K_0 + \sum_{j=1}^{n} \frac{a_j E_j}{l_j^3(x)} b_j(x) b_j^\top(x)$$

and the matrix $G(a, u, x) \in \mathbb{R}^{d \times d}$ is given by

$$G(a, u, x) = \sum_{j=1}^{n} \frac{a_j E_j}{l_j^3(x)} b_j^\top(x) u \tilde{B}_j = \sum_{j=1}^{n} \frac{s_j(x)}{l_j(x)} \tilde{B}_j.$$

We denote the small deformation axial forces by $s_j(x) = a_j E_j b_j^\top(x)u/l_j^2(x)$. The higher order terms are collected in the positive semidefinite matrix $H(a, u, x) \in \mathbb{R}^{d \times d}$ given by

$$H(a, u, x) = \frac{1}{4} \sum_{j=1}^{n} \frac{a_j E_j}{l_j^3(x)} (\tilde{B}_j u)(\tilde{B}_j u)^\top.$$

The matrices $K(a, x)$ and $G(a, u, x)$ are normally called the linear *stiffness matrix* and the *geometry matrix* of the truss, respectively. These matrices play a key role in the stability analysis described in the next section.

2.4 Stability analysis

Of major importance in mechanism design, as well as in traditional truss topology design in linear elasticity, is the possibility of obtaining an optimal design which is unstable. For structures, such as bridges, insufficient elastic stability may cause the structure to fail under loading even if the stresses in the material are not significant. For mechanisms, insufficient elastic stability may hinder the mechanism to function properly. For example, consider the mechanism in Figure 6(e). This mechanism cannot move unless buckling occurs. For this mechanism, the direction of movement is also uncertain and determined by imperfections in the design.

Since we want the optimal design to be able to perform its task repeatedly and with certainty, the elastic stability of the mechanism must be assured. To meet this requirement, we include constraints on the global stability of the mechanism in the problem formulation. The derivation of the stability conditions in this section follows, to a large extent, the derivations in [6, 42].

Intuitively, a structure or mechanism is stable for small external loads, but as the force increases and reaches a certain limit, the mechanism may become unstable. The problem of truss stability is to find a critical load for which the structure becomes unstable. A commonly used model for the stability analysis of trusses is the *linear buckling model* (see, e.g., [4]), which is based on the following assumptions.

1. The displacements depend linearly on the applied load when it is less than the critical buckling load.
2. The vector of these linear displacements is orthogonal to the vector of buckling displacements.
3. The bar axial forces $s_j = a_j E_j b_j^\top(x)u/l_j^2(x)$ remain constant during the deformation caused by buckling.

Assuming that the fourth order terms in $\Pi(a, u, x)$ are negligible for small loads we wish to determine the largest multiplier λ such that the parameterized potential energy

$$\Pi(a, u, x, \lambda) = \frac{1}{2}u^\top(K(a, x) + G(a, u, x))u - \lambda p^\top u$$

has a stationary point with a positive semidefinite Hessian. Using the third part of the linear buckling assumptions, the equilibrium equations associated with $\Pi(a, u, x, \lambda)$ become

$$(K(a, x) + G(a, u, x))u = \lambda p. \tag{6}$$

The value on λ for which the mechanism becomes unstable is called the critical buckling parameter and the corresponding load λp is called the critical buckling load. For small loads, the non-linear term $G(a, u, x)u$ is assumed to be negligible and in this situation u solves

$$K(a, x)u = p.$$

Denote this solution by u_0 and replace $G(a, u, x)$ by $G(a, u_0, x)$ and the equilibrium equations (6) by the linearization

$$(K(a, x) + \lambda G(a, u_0, x))u = \lambda p.$$

This equation is solvable whenever

$$\det(K(a, x) + \lambda G(a, u_0, x)) \neq 0.$$

Furthermore, it is required that the Hessian of the potential energy is positive semidefinite, i.e.

$$K(a, x) + \lambda G(a, u_0, x) \geq 0.$$

The largest such λ is the minimal positive eigenvalue $\bar{\lambda}$ of the generalized eigenvalue problem

$$(K(a, x) + \lambda G(a, u_0, x))z = 0.$$

Since the external load p is given, $0 < \bar{\lambda} < 1$ means that the mechanism is unstable under the given load p, while $\bar{\lambda} \geq 1$ indicate stability of the mechanism. Since the linear stiffness matrix $K(a, x)$ is positive semi-definite, the condition on the truss stability can be stated as the matrix inequality

$$K(a, x) + G(a, u_0, x) \geq 0 \tag{7}$$

where u_0 is a solution to the small-deformation equilibrium equations

$$K(a, x)u_0 = p. \tag{8}$$

Since the stability conditions (7) and (8) are intended for stiff structures rather than mechanisms, the displacement vector u_0 that solves (8) may violate the small displacement assumptions. Therefore, the external load p in the small-deformation equilibrium equations (8) is replaced by γp, where γ is a given scalar satisfying $0 < \gamma < 1$. In the numerical examples, $\gamma = 0.5$ is used.

The global stability conditions (7) guarantee that the mechanism is stable in the initial configuration, while the conditions (3) guarantee that the mechanism is stable in the final configuration. These conditions, however, do not guarantee that the mechanism is stable for the entire load path; it is typically checked that this holds for the entire path after performing optimization.

2.5 Degree of freedom analysis

When designing a mechanism, the first thing that one should pay attention to is the mechanical degrees of freedom of the mechanical linkage. It should be noted that this is a different notion from the degrees of freedom in the sense of finite element analysis. The mechanical degrees of freedom is defined as the number of independent inputs required to determine the position of all links of the mechanism with respect to a global coordinate system [22]. The investigation of the mechanical degrees of freedom is called a degree of freedom analysis.

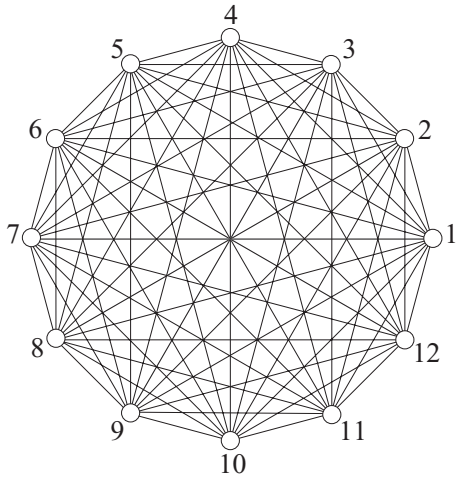


Fig. 7 Connectivity circle for the benchmark example ($\beta = 0$).

There are two standard counting relationships that can be applied to investigate this number, namely Maxwell's rule and the mobility criterion (or Grübler equation). The former attributes degrees of freedom to nodal points and views links as constraints. On the other hand, the latter attributes degrees of freedom to mechanical components and views joints as constraints. They are seemingly different but essentially the same; they consider the same property but from a different view point. The former is popular in structural analysis, while the latter is preferred in the multibody system analysis.

In the thesis Maxwell's rule is chosen since the truss representation is adopted. James Clerk Maxwell originally considered the mechanical performance of structures composed of straight bars connected at their ends by frictionless joints. He concluded that a simply stiff (statically determinate) structure should have (3#nodes − 6) bars in the three-dimensional space [43]. Maxwell's rule has been enhanced to cover so-called *tensegrity structures* [15] (already conjectured by Maxwell) and has been converted into a computer friendly matrix form [51]. Also, it has been extended to treat symmetry to a full extent based on group theory [24].

The rest of this section presents a degree of freedom analysis based on Maxwell's rule. It is independent of the geometry of the mechanism and correctly accounts for the possibility that some bars are overlapping. Assuming for now that a mechanism has *no redundant elements*, the number of mechanical degrees of freedom d_m of the mechanism can be calculated by the following equation based on the two-dimensional version of Maxwell's rule:

$$d_m = 2\#\text{nodes} - \#\text{bars} - \#\text{supports}. \tag{9}$$

The formula (9) is used in this thesis to define necessary conditions on the topology variables so that the mechanism has, for instance, exactly one degree of freedom. In order to count the number of active nodes, additional binary variables, called *node activity variables*, $v_k \in \{0, 1\}$ for $k = 1, \ldots, N$ are introduced. The number of active nodes, i.e. nodes for which $v_k = 1$, can easily be determined by summing the node activity variables.

Since the ground-structure allows connections (bars) between each pair of nodes, it intrinsically contains redundant bars. Therefore, the number of bars cannot be counted in a straightforward way in the mechanical degree of freedom analysis. The counting method applied here works by detecting the number of *independent bars* $m(a)$ in the ground-structure. The number of independent bars is given by the rank of the so-called *linear equilibrium matrix* $C(a, x) \in \mathbb{R}^{2N \times n}$ [15]. The j-th column in the matrix C is given by

$$C_j(a, x) = a_j B_j x / l_j(x), \quad j = 1, \ldots, n.$$

Due to the regularity of the nodal positions in the original ground-structure, the above method fails to detect overlapping elements as independent. This is unfortunate since overlapping bars are often useful for large displacement mechanisms, as in Figure 6(f). In order to distinguish overlapping bars as independent,

the nodal positions are rearranged within the degree of freedom analysis. One approach is to create what is called a *connectivity circle*, shown in Figure 7 for the benchmark example, by relocating all the nodes onto a unit circle with the positions

$$(\tilde{x}_{2k-1}, \tilde{x}_{2k}) = (\cos\theta_k, \sin\theta_k),$$

where the angles θ_k are given by

$$\theta_k = \frac{2\pi}{N}\left((k-1) + \beta 2^{-2^k}\right), \quad k = 1, \ldots, N.$$

The coefficient β is an optional parameter for avoiding parallel edges in the circle (see Appendix in [Paper 1] for further details). In the numerical examples we use $\beta = 1$. The number of independent bars is then defined as

$$m(a) = \mathrm{rank}(C(a, \tilde{x})).$$

Finally, the degree of freedom analysis writes

$$d_m(a, v) = 2\sum_{k=1}^{N} v_k - m(a) - s(v), \tag{10}$$

where $s(v)$ is the number of *active supports*, namely the supports which support active nodes. Especially when the boundary condition itself is not a main concern, the number is often treated as $s(v) = 3$ regardless of the node activities; the number '3' indicates the case where the mechanism is supported in a statically determinate manner.

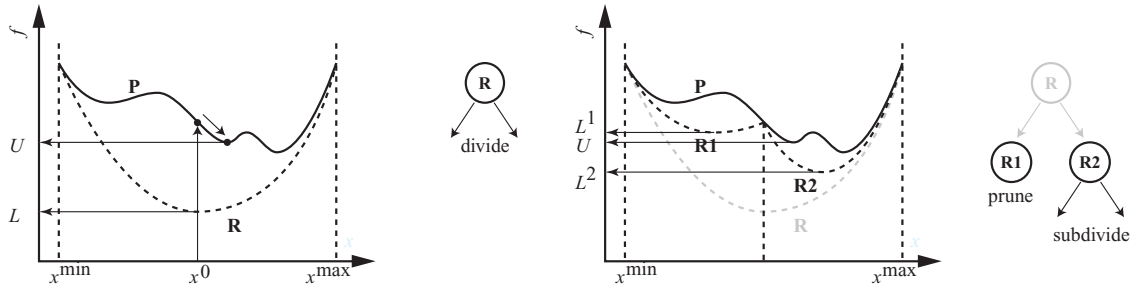
When only cross-sectional areas are used as design variables, the node activity variables are expressed as functions $v_k(a)$ which give $v_k(a) = 1$ if $\sum_{j\in A_k} a_j \geq 1$ otherwise $v_k(a) = 0$, where A_k is an index set of the bars incident to the k-th node. In this case the degree of freedom analysis(10) is rewritten as

$$d_m(a) = 2\sum_{k=1}^{N} v_k(a) - m(a) - s(v(a)). \tag{11}$$

Due to the definition of the $m(a)$ and the sum of the node activity variables that appear in the analysis (10) and (11), the number of mechanical degrees of freedom is only well-defined for $(a, v)^\top \in \mathbb{B}^n \times \mathbb{B}^N$. A differentiable relaxation of the DOF equation (11) is essential for the gradient-based optimization summarized in Section 3.2 (see also [Paper 2] for the detail).

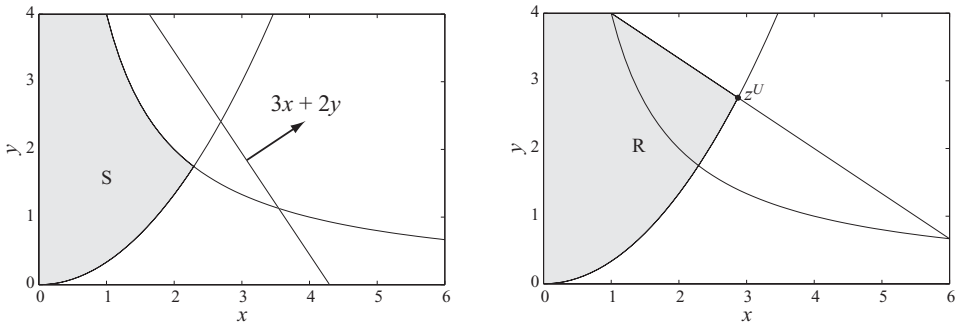
2.6 Branch and bound algorithm

This section is intended to give a short introduction to a branch and bound algorithm for those who are not familiar with the terms such as *branching*, *bounding* and *convex relaxations* since the algorithm has not



(a) Step 1 (b) Step 2

Fig. 8 Minimization of a non-convex univariate function



(a) Original non-convex problem

(b) Convex relaxation

Fig. 9 Illustrative example and its convex relaxation

been so popular yet in the structural optimization society. For further information about the algorithm and deterministic global optimization, see, e.g., [37, 38].

The branch and bound algorithm is, in a word, a divide-and-conquer type implicit enumeration scheme. In order to give a big picture of the algorithm, a minimization problem of a non-convex univariate function is shown in Figure 8. First of all, we need a convex relaxation **R** to the original problem **P**. The convex relaxation **R**, depicted by the dashed line, can be easily solved using gradient-based optimization since it is convex and smooth. The solution to the relaxation gives the lower bound (or underestimation) to the original problem **P**. On the other hand, the upper bound is given by any feasible solutions. Let us assume that a feasible solution U is found by a gradient-based local search from an arbitrarily chosen initial point x^0. Secondly we divide, as shown in Figure 8(b), its domain $[x^{\min}, x^{\max}]$ into two sections by the mean value, i.e. $(x^{\min} + x^{\max})/2$. This operation is called a *bisection*. The divided domains create two sub-problems **R1** and **R2** with typically better convex approximations. Solutions to the left and right relaxations give their lower bounds L^1 and L^2, respectively. But we have already known a feasible solution U that is better than the lower bound of the left sub-problem L^1. We have no reason to keep this domain for further conquering. In the search tree, the node is said to be pruned by bound. We continue to subdivide the active node until convergence.

In reality it is very rare to deal with a univariate function. The next example illustrates a more realistic branch and bound method that contains the very essence of the method applied to the mechanism design problem in [Paper 3]. The illustrative problem shown in Figure 9(a) is stated as

$$z = \max\{3x + 2y : (x, y) \in \mathrm{S}\}, \tag{12}$$

where $\mathrm{S} = \{(x, y) : xy \leq 4, x^2 \leq 3y\} \cap \mathrm{B}$; $\mathrm{B} = [0, 6] \times [0, 4]$.

Branch and bound itself is not a formal algorithmic specification, so the following issues must be clearly spelled out for implementation:

1. Relaxation technique
2. Branching strategy
3. Node selection rule

Firstly, as to the relaxation technique for the problem at hand:

- Leave linear and convex non-linear constraints.
- Reduce the order of non-linearity by introducing additional variables.

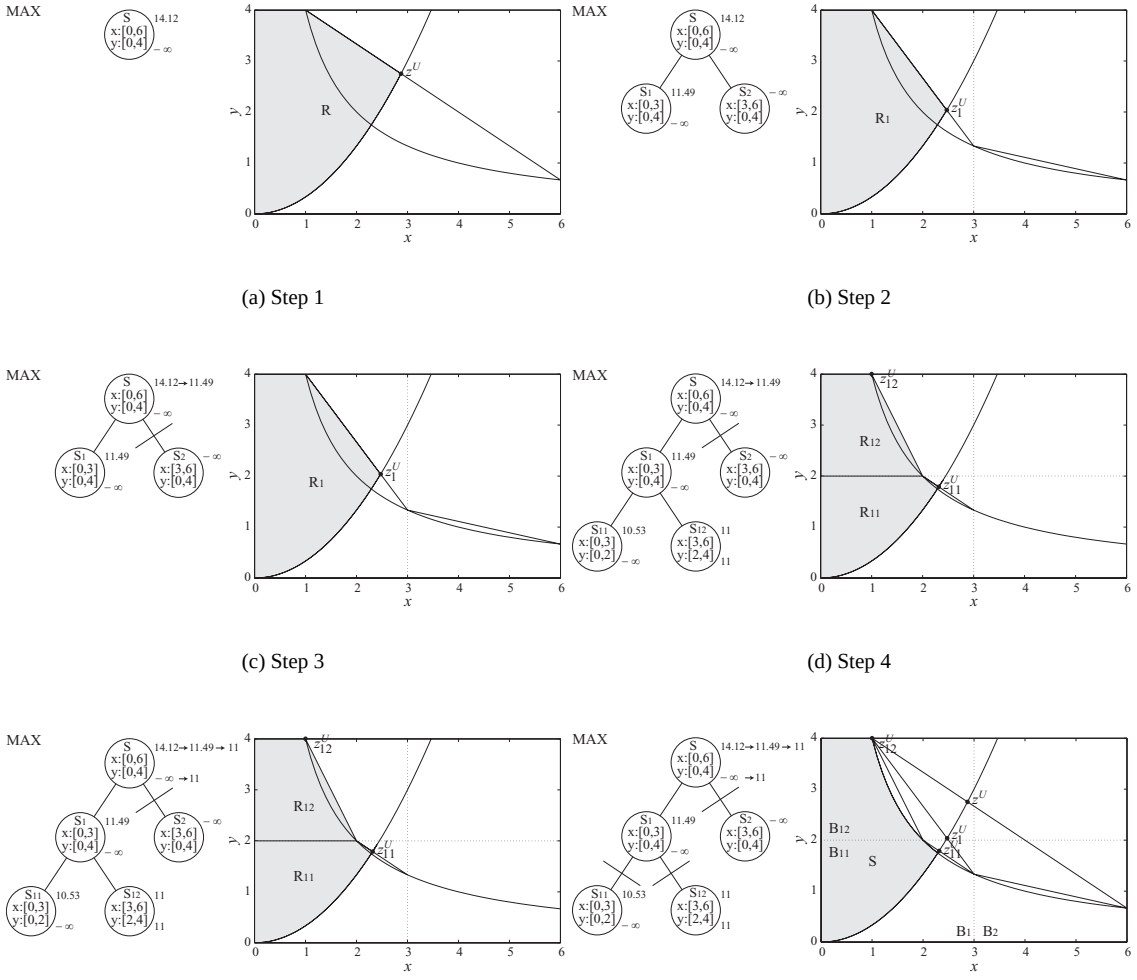


Fig. 10 Step-by-step branch and bound operations

- Replace the bilinear term xy by its convex and concave envelopes:

$$xy \geq \text{maximize} \left\langle \begin{array}{l} x^{\max} y + x y^{\max} - x^{\max} y^{\max} \\ x^{\min} y + x y^{\min} - x^{\min} y^{\min} \end{array} \right\rangle,$$
$$xy \leq \text{minimize} \left\langle \begin{array}{l} x^{\max} y + x y^{\min} - x^{\max} y^{\min} \\ x^{\min} y + x y^{\max} - x^{\min} y^{\max} \end{array} \right\rangle.$$

By applying the above relaxation technique, we obtain the following convex relaxation:

$$z^U = \max\{3x + 2y : (x, y) \in \mathrm{R}\}, \tag{13}$$

where $\mathrm{R} = \{(x, y) : 4x + 6y \leq 28, x^2 \leq 3y\} \cap \mathrm{B}$; $\mathrm{B} = [0, 6] \times [0, 4]$. The convex relaxation is shown in Figure 9(b). This relaxation technique is frequently used in [Paper 3] since most of non-linear terms can be reduced into bilinear terms by introducing auxiliary variables. Secondly, for the branching strategy we adopt a bisection scheme on the largest interval. Finally, we take a best-bound-first node selection rule for the problem (12).

Figure 10 illustrates the branch and bound operations step by step. In the first step, by solving the relaxation, the upper bound $z^U = 14.12$ is obtained, while the lower bound is set as $z^L = -\infty$ since no

feasible solution is found yet. The global upper and lower bounds are always stated at the upper and lower right corners of the root node, respectively. In the second step, we take the domain $x \in [0, 6]$ since it has the largest interval, and branch the domain into two sub-domains $x \in [0, 3]$ and $x \in [3, 6]$. The divided sub-domains produce two active nodes in the branch and bound tree (or search tree). In the left sub-domain $x \in [0, 3]$ the quality of the relaxation is improved by the tighter bounds, while in the right sub-domain $x \in [3, 6]$ there is no feasible region. The upper and lower bounds of the left sub-domain are given by $(z_1^U, z_1^L) = (11.49, -\infty)$, while $z_2^U = -\infty$ for the right sub-domain since $\mathbf{S_2}$ is empty. We have no reason to keep the empty region, so it is immediately pruned by infeasibility in the third step. Also, in this step the global upper bound z^U is updated from 14.12 to 11.49 according to

$$z^U = \underset{i \in \mathcal{I}}{\text{minimize}}\, z_i^U, \tag{14}$$

$$z^L = \underset{i \in \mathcal{I}}{\text{maximize}}\, z_i^L, \tag{15}$$

where $\mathcal{I}$ denotes the index set for the active nodes in the branch and bound tree. The global lower bound stays in the same value $z^L = -\infty$ since no feasible solution is found so far. In the fourth step, we take the domain $y \in [0, 4]$ and divide the domain by the mean value into two sub-domains $y \in [0, 2]$ and $y \in [2, 4]$. Similarly, we evaluate the upper and lower bounds of the two sub-problems corresponding to the regions $\mathbf{S_{11}}$ and $\mathbf{S_{12}}$. In the sub-problem for $\mathbf{S_{11}}$, we obtain $(z_{11}^U, z_{11}^L) = (10.53, -\infty)$, while $(z_{12}^U, z_{12}^L) = (11, 11)$ in the sub-problem for $\mathbf{S_{12}}$. Note that the solution z_{12}^U is also a feasible solution to the original problem. In the fifth step, we update the global upper and lower bounds as $(z^U, z^L) = (11, 11)$. In the final step, the node of $\mathbf{S_{11}}$ is pruned by bound since the local upper bound $z_{12}^U = 10.53$ is governed by the global lower bound $z^L = 11$. Also, the node of $\mathbf{S_{12}}$ is pruned by optimality since the local upper and lower bounds have collapsed as $z_{12}^U = z_{12}^L = 11$.

2.7 Configuration space

For the analysis part of the path-generating mechanism design problem in [Paper 4], we have to deal with a fundamental problem pertaining to the geometry of the configuration space [20, 31]. Let us consider the two-bar link mechanism shown in Figure 11 (see also Appendix 2 in [Paper 4]). Node 3 is connected to an invisible link driven by a motor that is rotating clockwise at a constant angular velocity ω around the center of the grey circle of radius $r = 0.0999$. As the two-bar link cannot be completely stretched out during the motion, the mechanism should stay in the elbow-up configuration while node 3 is moving on the input circle.

Throughout the path-generation problem we only analyze the kinematics of the mechanism and we do not take account of dynamics at all; also, there is no time integration for the calculation of the displacements. The displaced mechanism configuration at each step is obtained by solving static (but geometric non-linear) equilibrium equations for the given input point. This is analogous to the standard kinematic analysis [48] in which constraint equations are solved to determine the mechanism configuration.

Here, in order to simulate the movement by quasi-static analysis, namely kinematic analysis (see, e.g., [48]), we divide the grey input circle into 9 equally sized segments, making 9 input points; the equally distributed input points then correspond to equally discretized time steps.

We first apply an ordinary Newton-Raphson method to the analysis. In the analysis, the Young's modulus is set as $E = 10$ and all the cross-sectional areas are set as $a_i = 1$ for $i = 1, 2$. The result is shown in

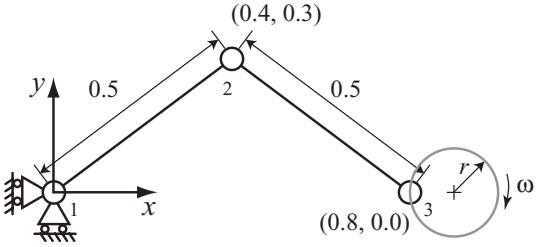
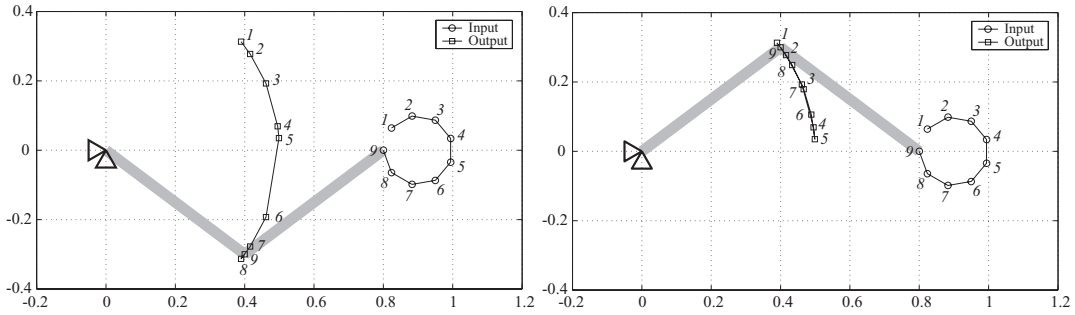


Fig. 11 Two-bar mechanism problem



(a) Newton-Raphson method (b) Levenberg-Marquardt method

Fig. 12 Path-following analysis of a two-bar mechanism

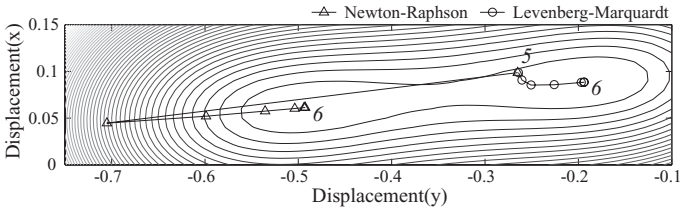


Fig. 13 Two stable points in total potential field

Figure 12(a), in which the corresponding input-output steps are numbered parallel in italics. This solution is wrong because the analysis has failed to keep the elbow-up configuration and a configuration transition occurred in the sixth step.

As an alternative, the Levenberg-Marquardt method [47] can also be applied to the same analysis. This method requires an expression for the total potential energy (1), in which the geometric non-linearity is considered using the Green-Lagrange strain measure (4). The basic computational steps of the Levenberg-Marquardt method are then as follows for fixed design $(a, x)^\top$:

Step 1: solve $(\nabla_{uu}\Pi(a,u,x) + \mu I)\Delta u = -\nabla_u\Pi(a,u,x)$ for Δu $\quad (\mu > 0)$

Step 2: update μ

Step 3: if $u + \Delta u$ is acceptable, then $u := u + \Delta u$

Here **Step 1** expresses that the Levenberg-Marquardt step Δu is a linear combination of the Newton step and the steepest descent step. The update scheme for μ in **Step 2** is a heuristic: increase μ if $\nabla_{uu}(u)$ is not significantly positive definite, otherwise decrease μ. (See Appendix 1 in [Paper 4] for the detail of the algorithm.) Finally, **Step 3** ensures that the Levenberg-Marquardt iterations always decrease the potential function. The combination of these three steps give the Levenberg-Marquardt method the nature of a trust region method as known in mathematical programming [16]. Figure 12(b) shows the result for the two-bar mechanism; the corresponding input-output steps are also here numbered parallel in italics. The analysis is successful since it has kept the elbow-up configuration in all the steps.

In order to investigate why the Newton-Raphson method fails in the analysis, we need to check the iterations in the sixth step. Figure 13 plots the potential field in the sixth step and all the iterations of both the Newton-Raphson method and the Levenberg-Marquardt method. It is notable that the Newton-Raphson method takes the first step across the far basin and converges to the bottom of this basin. In contrast, the Levenberg-Marquardt method takes a very conservative step in the steepest descent direction (due to the term μI in **Step 1**) and converges to the bottom of the near basin; this corresponds to the physically realistic configuration.

3 Summary of main results

The thesis consists of the following four papers presented in the order in which the research work developed. The purpose of this section is to give a perspective of the whole research work by summarizing the main results of these papers. In order to bridge the gaps between the independent papers, some additional results and further discussions are also included. As regards the notations, I must beg your forgiveness and patience in advance about the discrepancies between the papers. In order to keep the overall coherence within the section, some notation used here might be somewhat different from the original papers.

Throughout the papers, geometric non-linearity is considered using the Green-Lagrange strain measure, which is essential for the analysis of articulated mechanisms. In terms of the objective of the articulated mechanism design problems, the first to third papers deal with maximization of output displacement, while the fourth paper solves path-generating problems. From the mathematical programming point of view, the methods proposed in the first and third papers are categorized as deterministic global optimization, while those of the second and fourth papers as gradient-based local optimization. With respect to design variables, only cross-sectional areas are used in the first and second papers, whereas both cross-sectional areas and nodal positions are simultaneously optimized in the third and fourth papers.

3.1 Graph-theoretical enumeration [Paper 1]

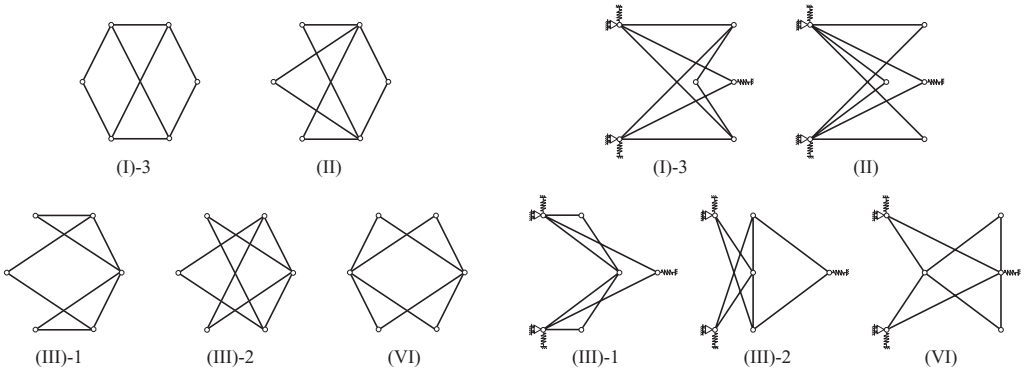
It is convenient to work with a benchmark problem for which a solution is known when one develops a new optimization method. This paper is an attempt to provide such a benchmark problem and its global optimum solution. This will be useful for further developments of more general and systematic methods in the following research. The benchmark problem has already been loosely introduced in Section 2.2 together with the ground-structure shown in Figure 5. The ground-structure contains all possible nodes and connections, as well as all necessary boundary conditions, load cases and other related parameters. The benchmark mechanism design problem is, here stated again, the maximization of the output displacement $c^\top u(a)$ for the given input forces p by picking V_n truss elements out of the n possible connections, so that the resulting mechanism has $d_m(a) = 1$ degree of freedom (DOF) when supported in a statically determinate manner, and that it is symmetric with respect to the horizontal line between nodes 2 and 11. This problem can be formulated as the following binary integer optimization problem for the truss connectivity vector a:

$$\begin{array}{ll} \text{maximize} & c^\top u(a) \\ \text{subject to} & d_m(a) = 1, \\ & e^\top a = V_n, \\ & a \in \{0,1\}^n, \\ & \text{symmetric}, \end{array} \tag{16}$$

where the displacement vector $u(a)$ is a solution to the equation (2) that expresses the residual version of the equilibrium equations. This style is often called a *nested* formulation. Geometric non-linearity should be taken into account, so a force incremental method with Newton-Raphson equilibrium iterations is adopted. The condition of a stable equilibrium point expressed in (3) is not explicitly treated, but, by making the force incremental steps sufficiently small, we typically obtain a stable equilibrium point in practice.

In the case of $V_n = 8$ elements, by applying the proposed graph theoretical enumeration method, an exhaustive analysis of all possible topologies can be performed for the above benchmark problem; one can thus identify the global optimum solution for this case. The basic concept of the graph-theoretical enumeration is first to generate a complete set of topological graphs (mechanism connectivities) and then embed (give actual dimensions and positions) them into the ground-structure in all possible ways. Figure 14(a) shows the five relevant topological graphs and Figure 14(b) shows the best mechanisms created from the five topological graphs. Note that the best mechanisms (III)-1 and (III)-2 in Figure 14(b) are the same as the mechanisms of Figure 6(b) and Figure 6(d), respectively, and that the candidate mechanisms shown in Figure 6(a) - 6(c) and Figure 6(d) - 6(f) are categorized as the topological graphs (III)-1 and (III)-2, respectively.

The conditions of a stable initial motion expressed by the equations (7) and (8) are not directly considered in the analysis model. Nevertheless, we have eventually obtained no such unstable mechanisms in the resulting best mechanisms in Figure 14(b). The mechanism of Figure 6(b) produces an output of less than 0.118 due to the lock between the left-most bars and thus fails to survive. Similarly, the mechanism of



(a) Relevant topological graphs (b) Best mechanisms among embeddings

Fig. 14 Graph-theoretical enumeration

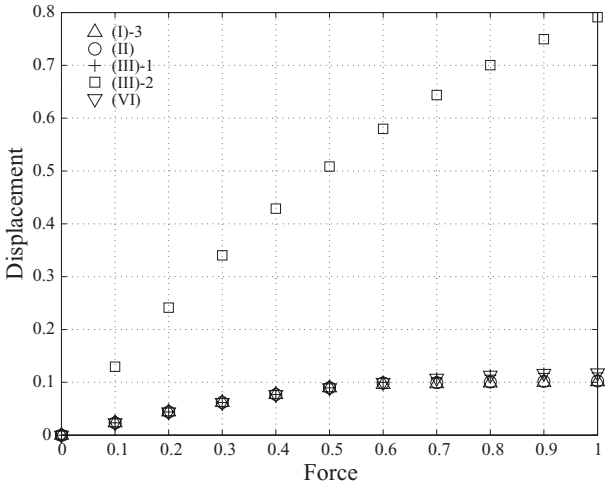


Fig. 15 Comparison of the performance

Figure 6(e) produces only the negative output displacement -0.05 in the analysis and is thus inferior to the mechanism of Figure 6(d).

Figure 15 compares the performance among the best mechanisms from the five different graphs. The mechanism (III)-2 is significantly better than the rest of the mechanisms, which all have a similar performance that is significantly lower than that of the optimal mechanism. This result underlines the great importance of choosing good topologies when designing mechanisms – bad topologies cannot be improved significantly by changing the shape.

Moreover, note that the optimal mechanism produces a scissors-like motion as shown in Figure 16. This is the reason for its outstanding performance. Also, note that this kind of mechanism cannot be obtained by the continuum-based topology optimization technique that has been very successful for compliant mechanism design.

3.2 Gradient-based local optimization [Paper 2]

Following the result of the first paper, a more systematic method is proposed by relaxing the original integer problem (16). The key strategy of the relaxation is to permit the design variable a to take on intermediate values between 0 and 1 as continuous cross-sectional areas and then reintroduce the relating constraints in

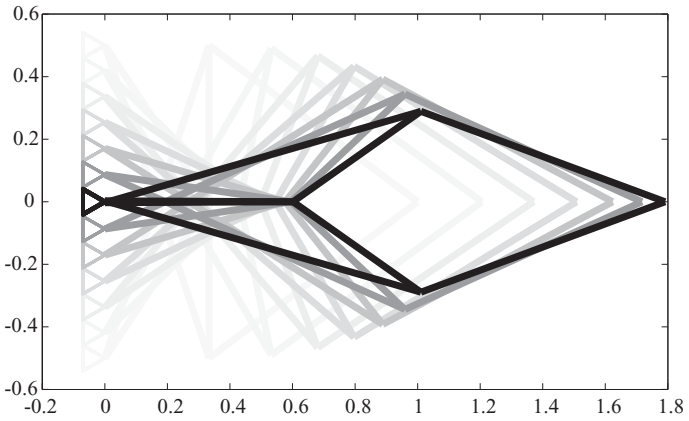
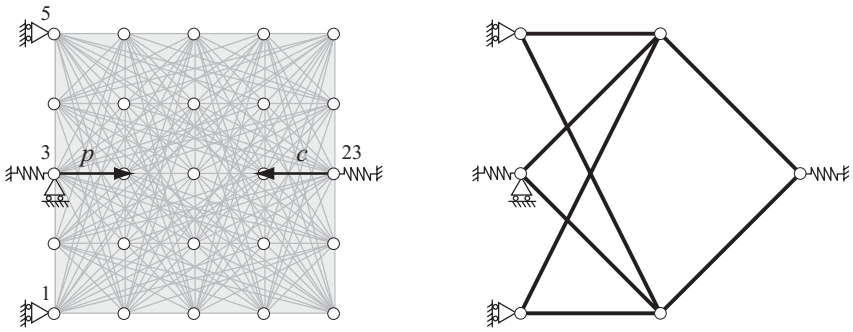


Fig. 16 Motion of the optimal design for the benchmark example

suitable relaxed forms. Furthermore, in order to recover the 0–1 values of a, the intermediate values are penalized by the so-called power law. Since we have a constraint on the total element number of bars, the penalization makes the intermediate values less efficient, thus typically forcing the variables a to go to the extremums for optimized designs. This is a typical technique used in continuum based topology optimization problems (see, e.g., [11] for the detail of power law approach). The relaxed mechanism problem is then formulated as follows:

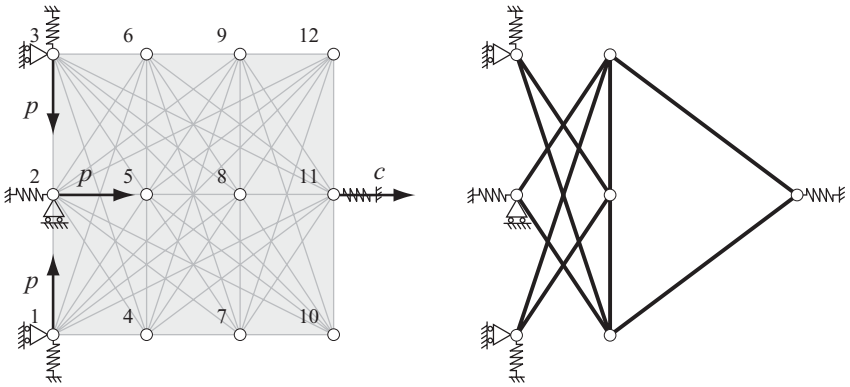
$$\begin{array}{ll} \text{maximize} & c^\top u(a) \\ \text{subject to} & \tilde{d}_m(a) \le 1, \\ & e^\top a \le V_n + \alpha, \\ & a \in [\varepsilon, 1]^n, \\ & \text{symmetric}, \end{array} \tag{17}$$

where ε is a positive lower bound of the design variables. This is introduced in order to avoid numerical difficulties such as a singular stiffness matrix and an insensitivity of the objective function to changes in the design variables around zero values. The extra term α compensates the effect of ε on satisfying the bound V_n (i.e., $\alpha = (n - V_n)\varepsilon$). $\tilde{d}_m(a)$ denotes a relaxed and differentiable version of DOF constraint based on Maxwell's rule (see, e.g., [15, 24, 51]). Specifically for the DOF calculation, a method for detecting independent bars (or equivalently neglecting redundant bars) in the ground-structure has been devised by introducing the connectivity circle as described in Section 2.5. In connection with the relaxation of the DOF equation, the differentiability of the weighted sum of all the involved eigenvalues is proven, for both the cases of simple and multiple eigenvalues. This plays a key role for the stable implementation of the



(a) Ground-structure (b) Optimized design (0.134)

Fig. 17 Inverter mechanism problem



(a) Ground-structure (b) Optimized design (0.791)

Fig. 18 Ten-bar mechanism problem

proposed method.

The test example in the first paper serves the basis for the development so as to compare the proposed method with three other alternative approaches. All the programs are implemented in MATLAB® 6.5 using the method of moving asymptotes (MMA) [57]. MMA requires that one provides subroutines that can calculate the objective function as well as its first derivative. The first derivative can be calculated by the adjoint technique described in [50]. The numerical experiments show the advantage of applying the relaxed DOF constraint. Also, an inverter problem shown in Figure 17(a) is treated where a continuation method is required in order to direct the optimization algorithm towards an integer solution. The proposed continuation method is implemented by introducing an explicit separation function as an additional constraint and has given an integer solution of Figure 17(b). The output displacement is given in the parenthesis.

Even though the proposed method seems promising for these two examples, there exists a critical limitation due to the non-convexity of the problem. The following is a good example to show the difficulty. Figure 18(a) shows a variation of the benchmark example with an extra support and an extra input at node 2. For the setting we seek also a symmetric mechanism but with ten bars. Figure 18(b) shows a ten-bar mechanism design obtained by the above proposed method. This design is a feasible solution to the original integer problem (16) but it is not so impressive that the two bars incident to the node 2 are actually redundant. (It is true that they contribute to the output by supporting the two bars incident to the output node but the effect is marginal.) This design is actually a local optimum. However, it is not surprising that one can end up with this kind of design using gradient-based local optimization techniques since the problem is intrinsically non-convex. This experience naturally leads one to consider the possibility to apply some global optimization techniques.

3.3 Branch and bound global optimization [Paper 3]

Global optimality is the main issue in this paper, so the formulation is totally revised from the global optimization point of view in mind. Some part of the formulation might look uncommon compared with the conventions in mechanics. This is because the number of complicating (non-linear and non-convex) terms and the degree of non-linearity in the expressions are the most decisive factors in the construction of convex relaxations; the fewer the number of the complicating terms and the less non-linear, the better.

The same problems in Figures 5, 17(a) and 18(a) are treated as basic topology optimization problems, and later the problems are extended to include the nodal positions as design variables, namely simultaneous topology and geometry optimization problems.

For the sake of simple and practical implementation of global optimization techniques, a simultaneous design and analysis approach is adopted in the sense that the necessary conditions derived from the analysis problem are explicitly stated in the design problem. The basic topology optimization problem is formulated

as the following non-convex mixed integer problem in the variables $(a, u, u_0, v)^\top \in \mathbb{B}^n \times \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{B}^N$:

$$
\begin{array}{lll}
\text{maximize} & c^\top u & \\
\text{subject to} & \nabla_u \Pi(a,u) = 0, \nabla^2_{uu}\Pi(a,u) \succeq 0, & \\
& K(a)u_0 = p, K(a) + G(a,u_0) \succeq 0, & \\
& \sigma_j^{\min} \le E_j \epsilon_j(u) \le \sigma_j^{\max}, & \forall\ j\ : a_j = 1, \\
& d_m(a,v) = 1, & \\
& e^\top v = V_N, e^\top a = V_n, & \\
& \displaystyle\sum_{j \in A_k} a_j \ge 2v_k, & k = 1, \ldots, N, \\
& a_j \le v_k, & k = 1, \ldots, N, \quad j \in A_k, \\
& u^{\min} \le u, u_0 \le u^{\max}, & \\
& a \in \{0,1\}^n, v \in \{0,1\}^N, & \\
& \text{symmetric}, &
\end{array}
\tag{18}
$$

where A_k is an index set of the bars incident to the k-th node. The primal design variables are truss connectivity variables a and node activity variables v. Also, displacement variables u, u_0 associated with the analysis part are included in the design problem; they are the essential state variables for the problem. The displacement bounds $u^{\min}$ and $u^{\max}$ make the feasible set of the mechanism design problem bounded, which is an essential property for constructing convex relaxations of the problem.

The formulation (18) looks lengthy but each line has an essential role. The first constraint ensures a stable equilibrium, while the second constraint precludes unwanted buckling modes in the initial movement. In other words, the first and second constraints are the stability conditions in the initial and final configurations, respectively. These conditions are explicitly modeled by non-linear matrix inequalities in the above formulation, while implicitly treated by the force incremental method in the nested approach adopted in the previous sections. The third constraint enforces the small strain assumption within the existing bars by posing stress limits $\sigma_j^{\min} < 0$ and $\sigma_j^{\max} > 0$ in compression and tension, respectively. Without this constraint, the optimization would utilize artificial self-stressed situations to produce larger output displacement. The sixth and seventh constrains are derived from valid graph-theoretical conditions based on the fact that, in order to obtain a practical mechanism that can transfer energy or work from the input ports to the output port, at least two bars must be incident to k-th node if the node is active, and that inactive nodes ($v_k = 0$) must have no incident bars.

A convergent branch and bound method is developed based on convex continuous relaxations to solve the mechanism design problem (18) to global optimality. In many aspects the suggested method has much in common with the branch and reduce approach described in [53] and the branch and contract algorithm described in [63]. The technique used in the relaxation is quite standard:

1. Drop matrix inequalities and DOF constraint (treated in feasibility and pruning tests).
2. Relax binary variables (treated by branching).
3. Reduce non-linearity and remove design dependency by introducing auxiliary variables.
4. Approximate bilinear terms by their convex and concave envelopes.
5. Keep quadratic convex constraints.

The best-bound-first node selection rule is adopted since this node selection rule is, by definition, bound improving. The following branching strategy is employed:

1. Binary variables before continuous variables.
2. Maximum integer infeasibility criterion and dichotomy branching on binary variables.
3. Choose one of the most responsible (continuous) variables for approximation errors and branch the variable by the actual value.
4. Bisection is performed when all the approximation errors are not significantly large.

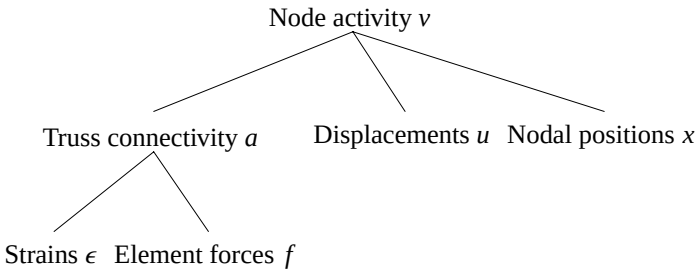


Fig. 19 Hierarchy of the variables

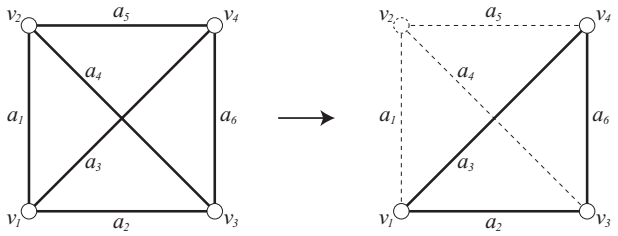


Fig. 20 Deactivation of v_2 and its propagation

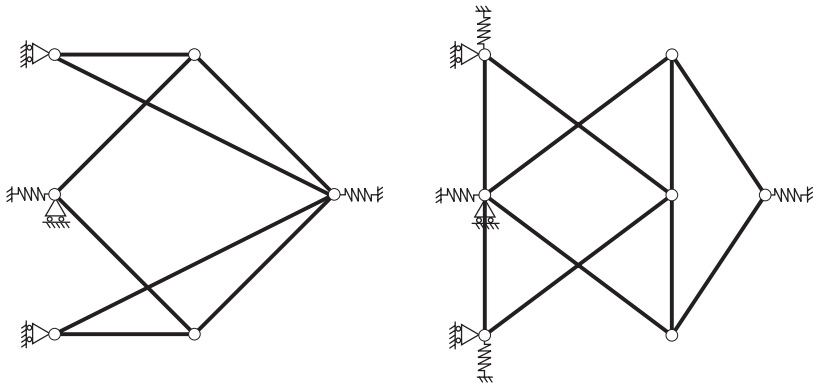
Specifically for the branching order, the hierarchical structure of the variables is highly utilized. Figure 19 shows the variables dependency in which the node activity variables v are placed on the top of the tree, suggesting that the branching should be begun with one of the variables. Figure 20 illustrates a virtual branching step in which the node activity variable v_2 is chosen to be deactivated; this propagates the lower hierarchy as deactivating all the incident bars (a_1, a_4, a_5) and making all the bounds of the dependent continuous variables collapse. In order to improve the quality of the relaxations, valid inequalities are added to the feasible set and bound contraction sub-problems are solved. Besides, pruning tests are enforced to determine if a certain region of the feasible set may be excluded from further investigation due to the concerned constraints. A fine tuning of these techniques is especially beneficial to practical convergence and computational efficiency. A heuristic method is also used to find feasible solutions that give upper bounds (in the minimization setting) to the design problem (18). The proposed heuristic method is based on a rounding technique for the integer variables and subsequent local optimization techniques with the fixed integer values.

As regards the implementation aspect, the whole program code is implemented in MATLAB® 6.5. The MATLAB code controls the overall branch and bound procedure and calls the following packages through interfaces:

1. CPLEX® 9.0 solves all linear programs (simplex solver) and all small-scale convex quadratic programs (barrier solver) prior to relaxations.
2. SNOPT™ 6.2 [26, 27] (sequential quadratic programming package) solves all convex quadratic programs of relaxations.
3. SeDuMi 1.05 [56] solves all linear semidefinite programs in the pruning tests on the stability issues.

The proposed method is a hybrid of several types of mathematical programming. Every single element matters in the implementation and a small improvement of each component may improve the total performance drastically.

The same problems in Figures 5, 17(a) and 18(a) are solved. For the first benchmark problem in Figures 5, the global optimum solution has been already known since the first paper, and the same design shown in Figure 16 is obtained again as expected. But, for the other two problems in Figures 17(a) and 18(a) the global optimum solutions have been newly obtained as the optimal inverter mechanism in Figure 21(a) and the optimal ten-bar mechanism in Figure 21(b), respectively. The output displacements are written in the parentheses.



(a) Inverter mechanism (0.137) (b) Ten-bar mechanism (0.895)

Fig. 21 Global optimum solutions to the basic topology optimization problems

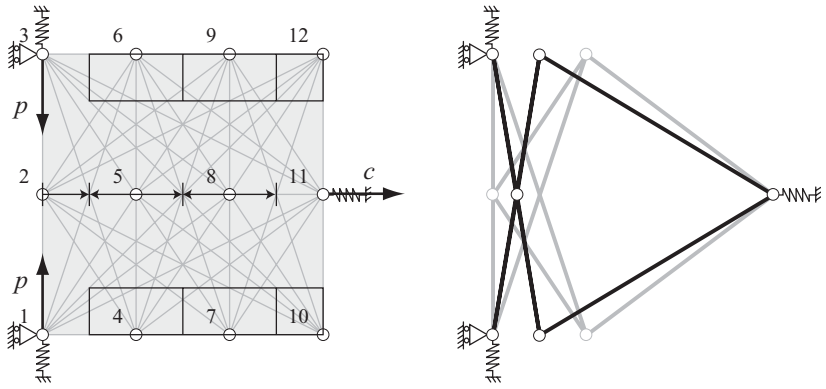
A natural extension of the basic topology design problem (18) is to include also the nodal positions as design variables, i.e. to perform simultaneous topology and geometry optimization. One advantage of this extension is that the ground-structure need not be very dense; thus the problem size can be kept moderate. Nevertheless, the possibility to change the nodal positions greatly increase the degree of non-linearity in the problem. The simultaneous topology and geometry optimization problem is formulated as the following non-convex mixed integer optimization problem:

$$
\begin{array}{lll}
\text{maximize} & c^\top u & \\
\text{subject to} & \nabla_u \Pi(a,u,x) = 0, \nabla^2_{uu}\Pi(a,u,x) \succeq 0, & \\
& K(a,x)u_0 = p, K(a,x) + G(a,u_0,x) \succeq 0, & \\
& \sigma_j^{\min} \le E_j \epsilon_j(u,x) \le \sigma_j^{\max}, & \forall\ j\ : a_j = 1, \\
& d_m(a) = 1, & \\
& e^\top v = V_N, e^\top a = V_n, & \\
& \displaystyle\sum_{j \in A_k} a_j \ge 2 v_k, & k = 1,\ldots,N, \\
& a_j \le v_k, & k = 1,\ldots,N, \quad j \in A_k, \\
& u^{\min} \le u, u_0 \le u^{\max}, & \\
& x^{\min} \le x \le x^{\max}, & \\
& a \in \{0,1\}^n, v \in \{0,1\}^N, & \\
& \text{symmetric}, &
\end{array}
\tag{19}
$$

where the variables are $(a,u,u_0,x,v)^\top \in \mathbb{B}^n \times \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^{2N} \times \mathbb{B}^N$; $x^{\min} \in \mathbb{R}^{2N}$ and $x^{\max} \in \mathbb{R}^{2N}$ are given limits on the nodal positions.

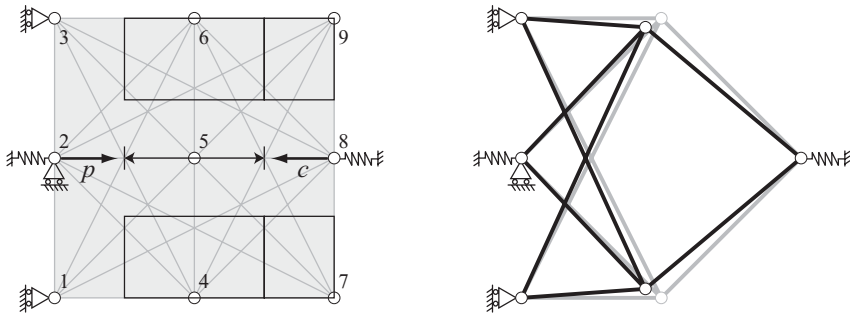
The branch and bound implementation is modified according to the extension and is applied to solve the three problems shown in Figures 22(a), 23(a) and 24(a). The solid boxes in the ground-structures indicate the bounds on the nodal positions involved. The resulting global optimum designs are depicted in black lines and the grey lines depict the connectivities with the original nodal positions; in the first problem, for instance, the bars depicted in grey lines have been chosen to create the mechanism configuration depicted in black lines with the varied nodal positions which have made some bars overlapped by positioning the central node 5 on the intersection of the two long bars. The output displacements are stated in the parentheses. In the second problem, a coarser (3×3)-node ground-structure is used since this can cover both the results in Figures 17(b) and 21(a).

Figure 25 shows the optimization histories of the upper and lower bounds (in the minimization setting) for the third problem. During the optimization several alternative mechanism designs are typically obtained in a single attempt to solve the problem. For the case, only three of them are presented along the upper bound history; one can see how the mechanism design evolves in the optimization process following the



(a) Ground-structure (b) Optimal design (0.971)

Fig. 22 The benchmark example problem with geometry variations

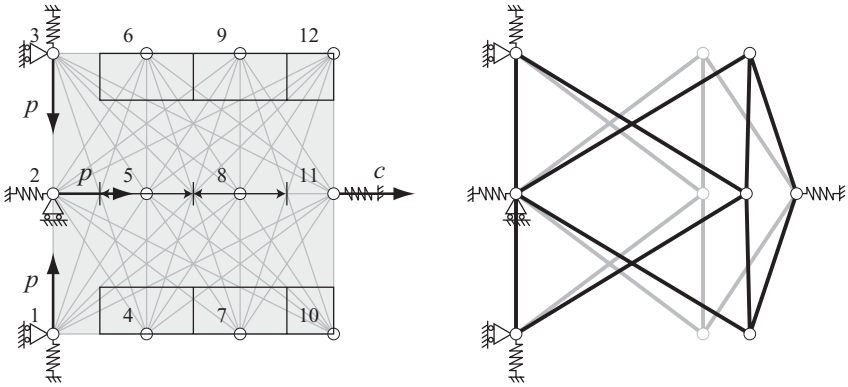


(a) Ground-structure. (b) Optimal design (0.168)

Fig. 23 The inverter mechanism problem with geometry variations

upper bound history. The first design is not good enough because two middle bars incident to the input ports are redundant. The second one is essentially the same as the mechanism in Figure 22(b) from the functional point of view; without two bars on the very left side, the mechanism would work similarly. The extra input force at the node 2 has yet to be utilized. And the final design is the one that can make the best use of available bars and input forces. With reference to the lower bound history, if the implementation were not good enough, for instance, the branching routine were not effective or the quality of relaxation were very poor, the lower bound history might have stalled on the way. It should be noted that the lower bound has been accelerated to converge in the last stage of the optimization because of the synergistic effect between the tight upper bound and the improved quality of relaxations with the tighter bounds on variables.

The numerical examples suggest that the proposed branch and bound method can reliably solve mechanism design problems of realistic size to global optimality. The encouraging computational results would never depreciate other gradient-based optimization methods; to the contrary, the proposed method should benefit by the results from other optimization methods. For the sake of fair evaluation of the implemented methods, the mechanism designs already obtained by the gradient-based method in the second paper have not been used deliberately in the test run of the program; but it is apparent that a good initial design would contribute to reduce the search tree in the branch and bound by providing a tighter upper bound.



(a) Ground-structure (b) Optimal design (0.993)

Fig. 24 The ten-bar mechanism problem with geometry variations

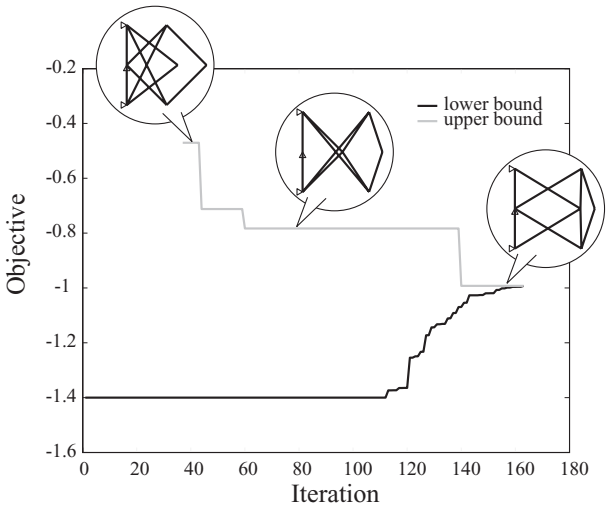


Fig. 25 Optimization history of ten-bar mechanism with geometry variations

3.4 Path-generation problems [Paper 4]

In the 19th century it was expected that kinematics would embody mathematical operations [46]. Alfred Bray Kempe tried to draw an arbitrary polynomial curve on the plane by combining four predefined functional linkages: reversor, multiplicator, additor and translator [40]. Recently his idea has been translated into modern mathematics and used to prove that any polynomial curve can be realized by a functional linkage if sufficient number of bars and constraints are provided [39, 41]. The mathematical proof has nothing to do with optimization but at least assures that it is not an outrageous attempt to design a path-generating mechanism using optimization.

Path-generation problems are still difficult to tackle using global optimization since the problem size is multiplied by the number of precision points and load cases. In this paper, we thus go back to gradient-based optimization again. The paper proposes a simultaneous shape and topology design method for path-generation in the non-linear truss representation using cross-sectional areas and nodal positions as design variables. This leads to a technique for simultaneous type and dimensional synthesis. For the analysis part it is essential to control the mechanism configuration so that the mechanism remains within a given configuration space, thus stabilizing the optimization process and resulting in a realistic solution. This can be achieved by using the Levenberg-Marquardt method, as described in Section 2.7.

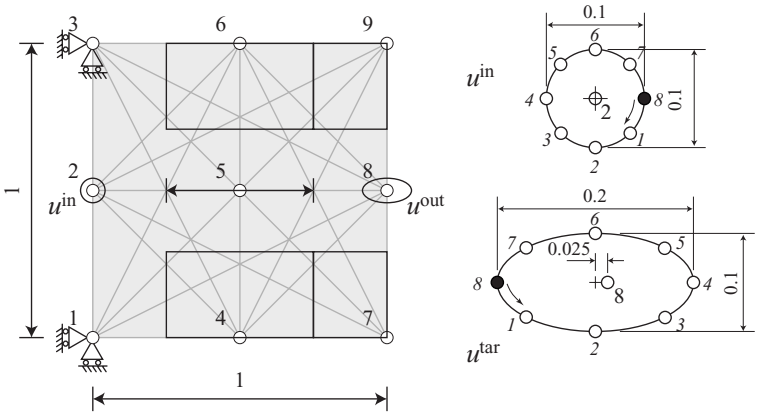


Fig. 26 Ground-structure for path-generation

Figure 26 shows a truss ground-structure for a benchmark example of a path-generation problem in which we are seeking to create a mechanism that can trace a prescribed target path. The target path is depicted by the black ellipse at the output node 8 and this path should be the resulting motion for the given input path depicted by the black circle at the input node 2. The input circular path is discretized into the eight equally distributed input points u^{in}, and then the corresponding target points u^{tar} are also located as indicated in the figure; the corresponding input-output steps are numbered parallel in italics. The input motion can be realized by introducing an extra link that is driven by a rotating motor around the center of the input circle at some constant angular velocity.

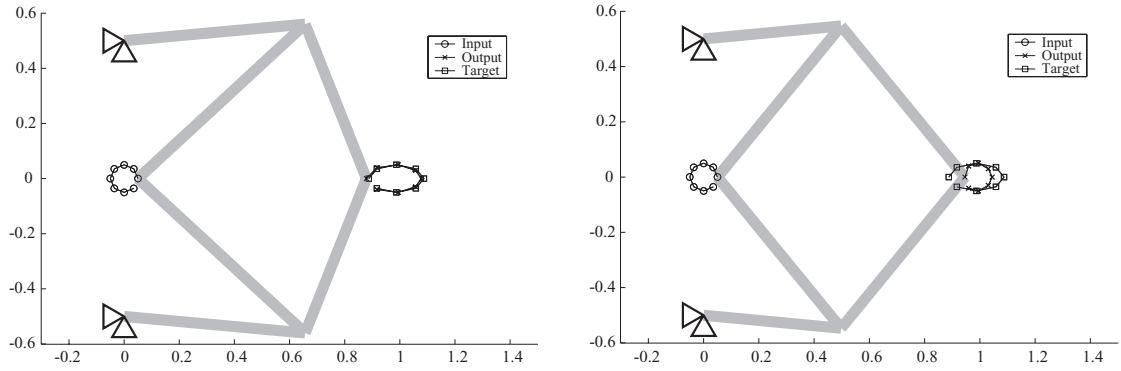
We will seek to generate a mechanism by picking $V_n = 6$ bars and simultaneously relocate the nodes within the bounds depicted by the solid boxes in Figure 26. We require that the resulting mechanism should be symmetric across the horizontal mid-axis. Also, in order to ensure the stability and certainty of the resulting motion, the resulting mechanism should have $d_m(a) = 2$ DOF. In the current case, each step of the kinematic analysis implies that the input port is clamped by removing 2 DOF; this results in a both statically and kinematically determinate structure. Thus, for the final design to be able to move along the input path and to have unique output behavior, we must ensure that $d_m(a) = 2$.

The problem can be formulated as a mixed integer optimization problem given below. As design variables the problem has the truss connectivity vector $a \in \mathbb{B}^n$ as well as the nodal position vector $x \in \mathbb{R}^{2N}$. The problem is formulated so as to minimize the infinity norm of a gap vector $g(a,x) = (g_1, g_2, \ldots, g_S)^\top$ which is defined as $g_i = \|u_i^{\text{out}}(a,x) - u_i^{\text{tar}}\|_2$ between the target points $u_i^{\text{tar}} \in \mathbb{R}^2$ and the actual output points $u_i^{\text{out}}(a,x) \in \mathbb{R}^2$ for $i = 1, 2, \ldots, S$:

$$\begin{array}{ll} \text{minimize} & \|g(a,x)\|_\infty \\ \text{subject to} & d_m(a) = 2, \\ & e^\top a = V_n, \\ & a \in \{0,1\}^n, \\ & x^{\min} \le x \le x^{\max}, \\ & \text{symmetric}, \end{array} \tag{20}$$

where the variables are $(a,x)^\top \in \mathbb{B}^n \times \mathbb{R}^{2N}$; $x^{\min} \in \mathbb{R}^{2N}$ and $x^{\max} \in \mathbb{R}^{2N}$ are given limits on the nodal positions.

In order to solve the problem by using continuous gradient-based optimization techniques such as MMA, the problem is relaxed by permitting the binary variable a to take on intermediate values between 0 and 1. Moreover, by utilizing the fact that the mechanism to be designed forms a both statically and kinematically determinate structure with the input port fixed at the given input points, the DOF constraint $d_m(a) = 2$ is replaced by conditions relating to the behavior with respect to the application of two additional very small load cases at the output node 8, one horizontal p_1 and one vertical p_2. These load cases simulate conditions in which the mechanisms are perturbed by some disturbances and the resulting mechanism should be stable enough against the disturbances when the input port is clamped. The two additional loads are applied separately, so the gap vectors become functions of these forces as $g(a,x,p_j)$ where $j = 0$ for



(a) Cross-sectional areas and nodal positions

(b) Only cross-sectional areas

Fig. 27 Benchmark example of path-generation problem

no load; $j = 1$ for horizontal load; $j = 2$ for vertical load. Finally, the objective is replaced by a weighted gap vector $\sum_{j=0}^{2} w_j g(a, x, p_j)$ for $w_0 = 0.98$ and $w_1 = w_2 = 0.01$. Note that, thus, the two additional load cases do not contradict the original objective, namely path-generation under the unloaded condition. The gap vectors $g(a, x, p_j) = (g_{1j}, g_{2j}, \ldots, g_{Sj})^\top$ are defined as $g_{ij} = \|u_i^{\text{out}}(a, x, p_j) - u_i^{\text{tar}}\|_2$ between the target points $u_i^{\text{tar}} \in \mathbb{R}^2$ and the actual output points $u_i^{\text{out}}(a, x, p_j) \in \mathbb{R}^2$ for $i = 1, 2, \ldots, S$.

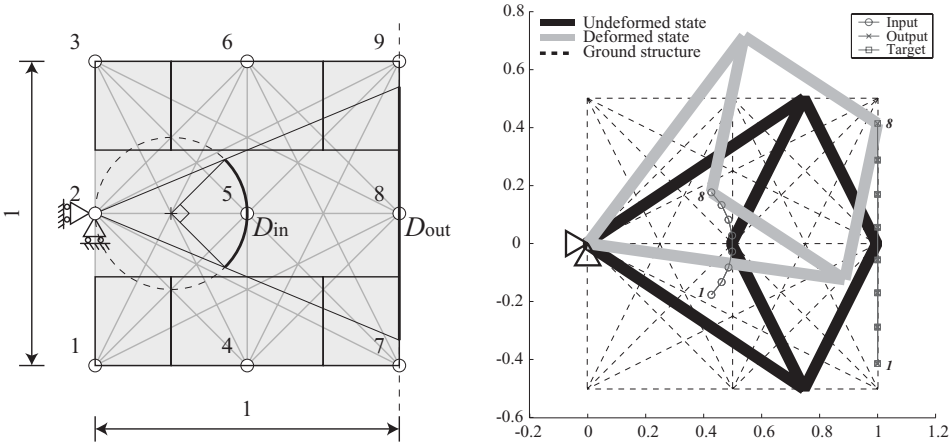
Finally the relaxed problem writes:

$$\begin{aligned} \text{minimize} \quad & \| \sum_{j=0}^{2} w_j g(a, x, p_j) \|_\infty \\ \text{subject to} \quad & e^\top a \le V_n + \alpha, \\ & a \in [\varepsilon, 1]^n, \\ & x^{\min} \le x \le x^{\max}, \\ & \text{symmetric}, \end{aligned} \tag{21}$$

where the design variables are $(a, x)^\top \in \mathbb{R}^n \times \mathbb{R}^{2N}$; $x^{\min} \in \mathbb{R}^{2N}$ and $x^{\max} \in \mathbb{R}^{2N}$ are given limits on the nodal positions. ε is a positive lower bound introduced in order to prevent numerical difficulties such as a singular stiffness matrix and an insensitivity of the objective function to changes in the design variables around zero values. The extra term α compensates the effect of ε on satisfying the bound V_n (i.e., $\alpha = (n - V_n)\varepsilon$).

Furthermore, in order to promote the 0–1 values of a, the intermediate values are penalized by the power law. Since we have a constraint on the total element number of bars, the penalization makes the intermediate values less efficient, thus the variables a are typically forced to go to the extremums, i.e. ε or 1 as the structure else would be too compliant under the load perturbations. This problem can also be solved by using MMA [57, 58] in a min-max formulation. The first derivative required for MMA can be calculated via the sensitivity analysis with an adjoint technique described in, e.g., [50]. Solving the problem by using both the truss connectivity vector a and the nodal position vector x simultaneously can give a result as illustrated in Figure 27(a). Here the resultant mechanism can follow the target path fairly well. In contrast, however, if one solves the problem for fixed nodal positions, a mechanism as shown in Figure 27(b) can be achieved. Due to the limitation of the choice of dimensions the mechanism cannot fully follow the target path. Note that both solutions are feasible in the original problem (20); substituting $\sum_{k=1}^{N} v_k = 6$ (active nodes), $m = 6$ (bars) and $s = 4$ (active supports), the DOF analysis (11) gives $d_m = 2$.

Another example problem treated with the same formulation is shown in Figure 28(a); note that the input and target paths and the boundary condition have been changed. We here seek to create a six-bar mechanism that can transform the arc (depicted by the thick black arc) at the input node 5 into the straight line segment (depicted by the thick black line) at the output node 8. The input circular motion can be realized by adding an extra link that connects the input node 5 and a ground point located at the focal point of the arc. The input arc is discretized into the eight equally distributed input points, and the corresponding target points



(a) Ground structure

(b) Optimized design

Fig. 28 Straight-line mechanism design problem

are located along the straight line as indicated in Figure 28(b); the corresponding input-output steps are numbered parallel in italics. Figure 28(b) also shows the resultant mechanism – the initial configuration is depicted in black and the displaced configuration in grey. Notice that in the final design the nodes 4 and 6 have moved to the right by 0.25. The DOF of the mechanism can also be calculated by the DOF analysis (11); substituting $\sum_{k=1}^{N} v_k = 5$ (active nodes), $m = 6$ (bars) and $s = 2$ (active supports), the DOF is obtained as $d_m = 2$, and this ensures the mechanism is feasible in the original problem (20). The resulting mechanism is the straight-line mechanism [62] invented in 1864 by the French engineer Charles-Nicolas Peaucellier [46]. Here this classical design has been created by a computer algorithm for mechanism synthesis.

4 Concluding remarks

Based on the truss ground-structure representation, the articulated mechanism design problem has been tackled using optimization techniques.

For the problems of maximizing output displacement, both a gradient-based optimization approach and a deterministic global optimization approach have been taken. In the gradient-based optimization approach, a method for detecting independent bars in the ground-structure has been devised; also, a relaxed version of the degree of freedom equation has been implemented. The proposed method has shown both its ability and limitation in terms of the non-convexity of the problem. This experience has lead investigations into the use of deterministic global optimization. In the deterministic global optimization approach, a convergent branch and bound method has been implemented. The method has shown the capability of finding global optimum solutions which have not been previously obtained by the gradient-based optimization. However the computational results show that it requires a more powerful method for finding good feasible solutions that may speed up the branch and bound method remarkably.

It should also be noted that the graph-theoretical enumeration method in the first paper has played an important role for providing the benchmark example and its global optimum solution. It is not clear if the approach can be generalized to more complicated settings – it is the identification of the topological graphs that is a cornerstone of the method and here number synthesis, as described in for example [59, 60], may provide a viable path ahead.

For the problems of generating a prescribed path, only a gradient-based optimization approach has been taken. This problem is still difficult to tackle using global optimization since the problem size is multiplied by the number of precision points and load cases. Specifically for the analysis part, it is essential to control the mechanism configuration so that the mechanism remains within the same configuration space, thus

stabilizing the optimization process and resulting in a realistic solution. This has been achieved by using the Levenberg-Marquardt method. It seems that the proposed method is promising for the presented examples, but one should not forget that the problem is non-convex and that the method has its critical limitation. One possible way to deal with the non-convexity is to apply a branch and bound with non-convex relaxations instead of convex relaxations. To what degree we should leave the non-convexity is unpredictable and this will be future work.

Other directions of future research will be an extension to dynamic problems, in which it is essential to develop, in terms of mathematical programming, a proper method for handling the dynamic analysis together with an associated sensitivity analysis. An extension to spatial mechanism, in which it will be needed to develop conditions to prevent that some members of the mechanism obstruct the displacement path of other members, is another important issue to address.

References

[1] Kinematic Models for Design Digital Library (K-MODDL). `http://kmoddl.library.cornell.edu/`.

[2] W. Achtziger, M.P. Bendsøe, A. Ben-Tal, and J. Zowe. Equivalent displacement based formulations for maximum strength truss topology design. *Impact of Computing in Science and Engineering*, 4(4):315–345, 1992.

[3] G.K. Ananthasuresh, S. Kota, and N. Kikuchi. Strategies for systematic synthesis of compliant MEMS. *Dynamic Systems and Control*, 2:677–686, 1994.

[4] K.-J. Bathe. *Finite Element Procedures in Engineering Analysis*. Prentice-Hall: Englewood Cliffs, NJ, 1982.

[5] A. Ben-Tal and M.P. Bendsøe. A new method for optimal truss topology design. *SIAM Journal on Optimization*, 3(2):322–358, 1993.

[6] A. Ben-Tal, F. Jarre, M. Kočvara, A. Nemirovski, and J. Zowe. Optimal design of trusses under a nonconvex global buckling constraint. *Optimization and Engineering*, 1:189–213, 2000.

[7] A. Ben-Tal, M. Kočvara, and J. Zowe. Two non-smooth methods for simultaneous geometry and topology design of trusses. In M.P. Bendsøe and C.A. Mota Soares, editors, *Topology Design of Structures*, pages 31–42. Kluwer Academic Publishers, 1993.

[8] A. Ben-Tal and A. Nemirovski. Potential reduction polynomial time method for truss topology design. *SIAM Journal on Optimization*, 4(3):596–612, 1994.

[9] M.P. Bendsøe, A. Ben-Tal, and J. Zowe. Optimization methods for truss geometry and topology design. *Structural Optimization*, 7(3):141–158, 1994.

[10] M.P. Bendsøe and N. Kikuchi. Generating optimal topologies in structural design using homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71(2):197–224, 1988.

[11] M.P. Bendsøe and O. Sigmund. *Topology Optimization: Theory, Methods and Applications*. Springer, 2003.

[12] D. Bestle and P. Eberhard. Multi-criteria multi-model design optimization. In D. Bestle and W. Schiehlen, editors, *IUTAM Symposium on Optimization of Mechanical Systems*, pages 33–40. Kluwer, Dordrecht, 1995.

[13] D. Bestle and W. Schiehlen. In D. Bestle and W. Schiehlen, editors, *IUTAM Symposium on Optimization of Mechanical Systems*, pages 26–31. Kluwer, Dordrecht, 1995.

[14] T.E. Bruns and D.A. Tortorelli. Topology optimization of non-linear elastic structures and compliant mechanisms. *Computer Methods in Applied Mechanics and Engineering*, 190(26–27):3443–3459, 2001.

[15] C.R. Calladine. Buckminster Fuller's tensegrity structures and Clerk Maxwell's rules for the construction of stiff frame. *International Journal of Solids and Structures*, 14:161–172, 1978.

[16] A.R. Conn, N.I.M. Gould, and P.L. Toint. *Trust-Region Methods*. SIAM, 2000.

[17] M.A. Crisfield. *Non-linear Finite Element Analysis of Solids and Structures, Vol. 1: Essentials*. John Wiley & Sons, New York, 1997.

[18] W.S. Dorn, R.E. Gomory, and H.J. Greenberg. Automatic design of optimal structures. *Journal de Mécanique*, 3:25–52, 1964.

[19] P. Eberhard, D. Bestle, and U. Piram. Optimization of damping characteristics in nonlinear dynamic systems. In *WCSMO–1st World Congress of Structural and Multidisciplinary Optimization*, pages 863–870, Goslar, Germany, 1995. Pergamon.

[20] D. Eldar. Planar machines' web site: an invitation to topology. Master's thesis, Department of Mathematics, University of Toronto, 1999.
`http://www.math.toronto.edu/~drorbn/People/Eldar/thesis/index.html`.

[21] A.G. Erdman. Computer-aided mechanism design - now and the future. *Journal of Mechanical Design*, 117:93–100, 1995.

[22] A.G. Erdman and G.N. Sandor. *Mechanism Design: Analysis and Synthesis*, volume I. Prentice Hall, 2nd edition, 1990.

[23] P. Fleron. The minimum weight of trusses. *Bygnings statiske Meddeleser*, 35:81–96, 1964.

[24] P.W. Fowler and S.D. Guest. A symmetry extension of Maxwell's rule for rigidity of frames. *International Journal of Solids and Structures*, 37:1793–1804, 2000.

[25] M.I. Frecker, G.K. Ananthasuresh, S. Nishiwaki, N. Kikuchi, and S. Kota. Topological synthesis of compliant mechanisms using multi-criteria optimization. *Journal of Mechanical Design*, 119:238–245, 1997.

[26] P.E. Gill, W. Murray, and M.A. Saunders. User's guide for SNOPT 5.3: a Fortran package for large-scale nonlinear programming. Technical report, NA 97-5 , Department of Mathematics, University of California, 1997.

[27] P.E. Gill, W. Murray, and M.A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Journal on Optimization*, 12(4):979–1006, 2002.

[28] J.M. Hansen. Synthesis of mechanisms using time-varying dimensions. *Structural and Multidisciplinary Optimization*, 7(1):127–144, 2002.

[29] J.M. Hansen and D.A. Tortorelli. An efficient method for synthesis of mechanisms using an optimization method. In D. Bestle and W. Schiehlen, editors, *IUTAM Symposium on Optimization of Mechanical Systems*, pages 129–138. Kluwer, Dordrecht, 1995.

[30] J.M. Hansen and D.A. Tortorelli. Sensitivity analysis of mechanisms based on the joint coordinate method. In *WCSMO–1st World Congress of Structural and Multidisciplinary Optimization*, pages 871–876, Goslar, Germany, 1995. Pergamon.

[31] V.L. Hansen. *Braids and Covering*. Cambridge University Press, 1989.

[32] E.J. Haug. Design sensitivity analysis of dynamic systems. In C.A. Mota Soares, editor, *Computer Aided Optimal Design: Structural and Mechanical Systems*, volume F27. Springer-Verlag, Berlin, 1987.

[33] E.J. Haug. *Computer-Aided Kinematics and Dynamics of Mechanical Systems, Volume I: Basic meathods*. Allyn and Bacon, 1989.

[34] BBC Homepage History. Historic figures: James Watt (1736 - 1819).
`http://www.bbc.co.uk/history/historic_figures/watt_james.shtml`.

[35] BBC Homepage History. Historic figures: Thomas Newcomen (1663 - 1729). http://www.bbc.co.uk/history/historic_figures/newcomen_thomas.shtml.

[36] BBC Homepage History. Industrialisation. http://www.bbc.co.uk/history/society_culture/industrialisation/.

[37] R. Horst, P.M. Pardalos, and N.V. Thoai. *Introduction to Global Optimization*. Nonconvex Optimization and its Applications. Kluwer Academic Publishers, 2nd edition, 2000.

[38] R. Horst and H. Tuy. *Global Optimization: Deterministic Approaches*. Springer-Verlag, 1993.

[39] M. Kapovich and J.J. Millson. Universality theorems for configuration spaces of planar linkages. *Topology*, 41:1051–1107, 2002.

[40] A.B. Kempe. On a general method of describing plane curves of the n-th degree by linkwork. *Proc. London Math. Soc.*, pages 213–216, 1876.

[41] H. King. Planar linkages and algebraic sets. In *Proceedings of 6th Gökova Geometry-Topology Conference*, pages 33–56. Turkish Journal Mathematics, 1999.

[42] M. Kočvara. On the modelling and solving of the truss design problem with global stability constraints. *Structural and Multidisciplinary Optimization*, 23:189–203, 2002.

[43] J.C. Maxwell. On the calculation of the equilibrium and stiffness of frames. *Philosophical Magazine*, pages 294–299, 1864.

[44] A.G.M. Michell. The limit of economy of material in frame structures. *Philosophical Magazine*, 8(6):589–597, 1904.

[45] R.J. Minnaar, D.A. Tortorelli, and J.A. Snyman. On nonassembly in the optimal dimensional synthesis of planar mechanisms. *Structural and Multidisciplinary Optimization*, 21(5):345–354, 2001.

[46] F.C. Moon. Franz Reuleaux: Contributions to 19th c. kinematics and theory of machines. *Applied Mechanics Reviews*, 56(2):261–285, 2003.

[47] H.B. Nielsen. Damping parameter in Marquardt's method. Technical Report IMM-REP-1999-05, Department of Infomatics and Mathematical Modelling, Technical University of Denmark, 1999. http://www.imm.dtu.dk/~hbn/publ/.

[48] P. E. Nikravesh. *Computer-Aided Analysis of Mechanical Systems*. Prentice Hall, 1988.

[49] J.M. Pagalday, I. Aranburu, A. Avello, and J. García de Jalón. Multibody dynamics optimization by direct differentiation methods using object oriented programming. In D. Bestle and W. Schiehlen, editors, *IUTAM Symposium on Optimization of Mechanical Systems*, pages 213–220. Kluwer, Dordrecht, 1995.

[50] C.B.W. Pedersen, T. Buhl, and O. Sigmund. Topology synthesis of large-displacement compliant mechanisms. *International Journal for Numerical Methods in Engineering*, 50:2683–2705, 2001.

[51] S. Pellegrino and C.R. Calladine. Matrix analysis of statically and kinematically indeterminate frameworks. *International Journal of Solids and Structures*, 22:409–428, 1986.

[52] W. Prager and G.I.N. Rozvany. Optimal layout of grillages. *JOURNAL OF STRUCTURAL MECHANICS*, 5(1):1–18, 1977.

[53] H.S. Ryoo and N.V. Sahinidis. A branch-and-reduce approach to global optimization. *Journal of Global Optimization*, 8:107–138, 1996.

[54] O. Sigmund. Some inverse problems in topology design of materials and mechanisms. *Mechanics of Structures and Machines*, pages 277–284, 1996.

[55] O. Sigmund. On the design of compliant mechanisms using topology optimization. *Mechanics of Structures and Machines*, 25:493–524, 1997.

[56] J.F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999.
Version 1.05 is available from `http://fewcal.kub.nl/sturm`.

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

[58] K. Svanberg. A class of globally convergent optimization methods based on conservative convex separable approximation. *SIAM Journal on Optimization*, 12(2):555–573, 2002.

[59] C.R. Tischler, A.E. Samuel, and K.H. Hunt. Kinematic chains for robot hands–I. Orderly number-synthesis. *Mechanism and Machine Theory*, 30:1193–1215, 1995.

[60] C.R. Tischler, A.E. Samuel, and K.H. Hunt. Kinematic chains for robot hands–II. Kinematic constraints, classification, connectivity, and actuation. *Mechanism and Machine Theory*, 30:1217–1239, 1995.

[61] D.A. Tortorelli. Design sensitivity analysis: Overview and review. *Inverse problems in Engineering*, 1:71–105, 1994.

[62] W. Wunderlich. *Ebene Kinematik*. Bibliographisches Institut Mannheim/Wien/Zürich, 1968.

[63] J.M. Zamora and I.E. Grossmann. A branch and contract algorithm for problems with concave univariate, bilinear and linear fractional terms. *Journal of Global Optimization*, 14:217–249, 1999.

[Paper 1]

Brief note

Struct Multidisc Optim 27, 295–299 (2004)
DOI 10.1007/s00158-004-0409-9

Planar articulated mechanism design by graph theoretical enumeration

A. Kawamoto, M.P. Bendsøe and O. Sigmund

Abstract This paper deals with design of articulated mechanisms using a truss-based ground-structure representation. By applying a graph theoretical enumeration approach we can perform an exhaustive analysis of all possible topologies for a test example for which we seek a symmetric mechanism. This guarantees that one can identify the global optimum solution. The result underlines the importance of mechanism topology and gives insight into the issues specific to articulated mechanism designs compared to compliant mechanism designs.

Key words mechanism design, topology optimization, graph theory, enumeration

1 Introduction

Research on the design of compliant mechanisms (e.g., Sigmund (1997), Pedersen (2001), Bruns (2001) and Ananthasuresh (1994)) using topology optimization techniques in continuum structures suggests that it should also be possible to obtain results for articulated mechanisms by applying such techniques (Sigmund 1996; Frecker 1997).

In this paper we examine exhaustively all possible topologies of one test example by a graph theoretical enumeration approach. This allows for identifying the global optimum solution when we assume that the resultant mechanism is symmetric. We view the present work as a first step in devising optimization techniques for design of an articulated mechanism, with the achieved global optimum solution providing a rigorous comparison design with which solutions obtained by methods currently under development can be matched.

Received: 28 November 2003
Revised manuscript received: 10 February 2004
Published online: 4 May 2004

A. Kawamoto[1, ✉], M.P. Bendsøe[1] and O. Sigmund[2]

[1] Department of Mathematics, Technical University of Denmark, Matematiktorvet B303, 2800 Kgs. Lyngby, Denmark
e-mail: a.kawamoto@mat.dtu.dk, m.p.bendsoe@mat.dtu.dk
[2] Department of Mechanical Engineering, Solid Mechanics, Technical University of Denmark, Nils Koppels Allè B404, 2800 Kgs. Lyngby, Denmark
e-mail: sigmund@mek.dtu.dk

2 Definition of test example

The test example we examine in detail is shown in Fig. 1. The figure shows the truss ground structure of possible nodes and connections, as well as all necessary boundary conditions, load cases and other related parameters. One relevant mechanism design problem is the maximization of the output displacement D_{out} for the given input forces F_{in} by picking M^* truss elements out of the M possible connections, so that the resultant mechanism has $f = 1$ degree of freedom (DOF) and is symmetric with respect to the horizontal line between nodes 2 and 11. This problem can be formulated as the following binary integer optimization problem for the truss connectivity vector $\chi = (\chi_1, \ldots, \chi_M)$:

$$\max_{\chi \in \{0,1\}^M} D_{\text{out}} \text{ s.t. } \sum_{e=1}^{M} \chi_e = M^*; \quad f = 1; \quad \text{symmetric.} \tag{1}$$

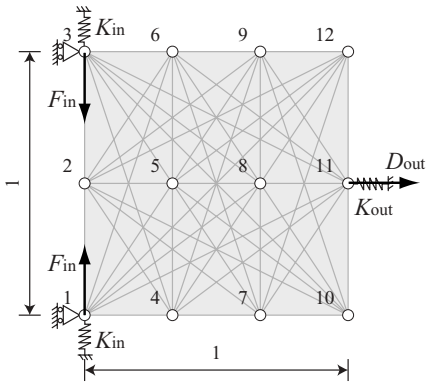


Fig. 1 Truss ground structure

Here D_{out} is given via the displacement vector U which is a solution to an equation $R(U) = 0$ that expresses the residual version of the equilibrium equations. Geometrical non-linearity should be taken into account, so we adopt a force incremental method with Newton–Raphson equilibrium iterations for evaluation of the displacement D_{out}.

3 Graph theoretical enumeration approach

There are 66 potential elements in the ground structure of Fig. 1. When the number of bars in the mechanism is set at $M^* = 8$, the complexity of the problem is ${}_{66}C_8 (\sim 6 \times 10^{10})$, so evaluating the totality of possible mechanisms is not a viable approach.[1]

An alternative way to address the problem is to first generate a complete set of topological graphs and then embed them (as shown in Fig. 2) into the ground structure in all possible ways. Assuming that the mechanism has *no redundant elements* (for brevity, we shall not discuss this here; see, e.g., Calladine (1978), Pellegrino (1986) and Fowler (2000) for details), the degrees of freedom of a mechanism that is supported in a statically determinate manner can be calculated by the following DOF equation based on Maxwell's rule in 2D:

$$f = 2n - m - 3 \tag{2}$$

where n is the number of nodes and m is the number of bars. In this case where $m = 8$ and $f = 1$, a possible mechanism has $n = 6$ nodes (note that this is only a necessary condition, see the above references). If we choose to focus on one topological configuration of eight edges and six vertices (we shall use *edge* and *vertex* for topological graphs), as shown in Fig. 2, we can reduce the design problem to a subproblem that entails picking six ordered nodes from the 12 potential nodes of the ground structure. This results in the assignment of six vertices of the topological graph to six out of 12 nodes in the ground structure. The subproblem can be illustrated by a bipartite matching problem shown in Fig. 3. The complexity of the subproblem is only ${}_{12}P_6 (\sim 6 \times 10^5)$[2]. If we have p different topological configurations, the calculation time is simply multiplied by p. In the present case, $p = 24$ and the complete set of the 24 non-isomorphic graphs, which have six vertices and eight edges, can be found in Harary (1969).

So far, we have not utilized the symmetry constraint and we illustrate now how to use this to reduce the problem further. We begin with vertex configurations without edges. There are two possible symmetric vertex configurations with three rows (see Fig. 4). Here we introduce a symmetric embedding that assigns the vertices to the nodes maintaining the symmetry around the horizontal mid-axis, which means that the vertices on the first row go to the nodes on the first row, the second row in F1 or F2 goes to the second row in Fig. 1, and so on.

[1] If one function evaluation (i.e., one non-linear analysis) takes one second, the total calculation time amounts to about 180 years.

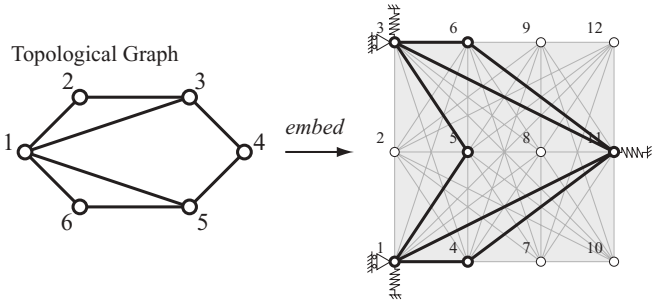


Fig. 2 Embedding of a topological graph (eight edges and six vertices) into a truss ground structure

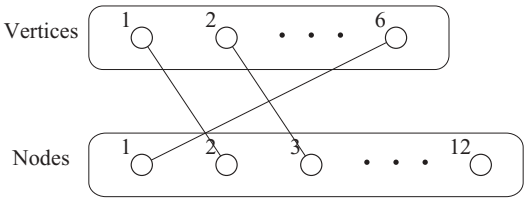


Fig. 3 Bipartite matching problem between vertices and nodes

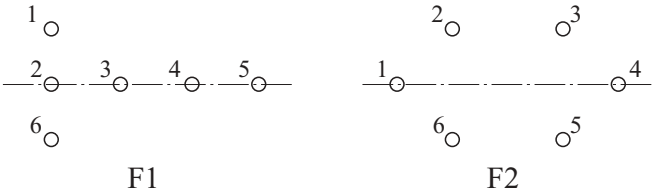


Fig. 4 Two possible vertex configurations

3.1 Case F1

Note that all the possible topological graphs generated from the vertex configuration F1 should satisfy the following condition on the number of incident edges with a vertex, namely vertex degree v_i $(i = 1, \ldots, 6)$:

$$v_1 = v_6 \tag{3}$$

due to the restriction of the symmetry. Besides, if the embeddings involve all possible swaps among vertices 2 through 5, it suffices to consider the cases where vertex degrees meet the constraints:

$$v_2 \geq v_3 \geq v_4 \geq v_5 \,. \tag{4}$$

[2] This amounts to 7.7 days of computational time for evaluating all designs.

This is because vertices 1 and 6 should be assigned to the input ports in the ground structure of Fig. 1 anyway.

We can thus classify the topological graphs generated from the configuration F1 by the vertex degree $v_1 (= v_6)$ running from 1 to 4; these correspond to the classes (i) to (iv) in Fig. 5. Note that the number of possible graphs varies in the different classes. Graph (i) is actually irrelevant as in reality it has two degrees of freedom even though (2) gives $f = 1$ DOF. This is a special case where the structure has one redundant bar in the mid axis. In this case, the structure has extra states of self-stress and mechanism mode. Thus the mechanism DOF increases; see Calladine (1978), Pellegrino (1986) and Fowler (2000) for details. Graphs (ii)-{1, ... ,5}, (iii)-{1,2} and (iv)-1 are also all irrelevant due to the symmetric loading and support conditions; all these graphs have asymmetric mechanism modes with respect to the horizontal mid-axis.

With reference to the mechanisms that have one isolated bar, since the rest of nodes and bars form a structure, these mechanisms cannot transfer energy or work from the input ports to the output port. Also, we have observed that the isolated bars cause the mechanical DOF to increase in graph (i). We are not interested in all of these irrelevant mechanisms, and will consider the topological graphs that satisfy $v_i \geq 2$ as the candidate graphs henceforth.

The final graph (iv)-2 essentially has only one possible embedding into the ground structure because vertices 1 and 6 of the graph are assigned to the input ports in Fig. 1, and swapping any pair of vertices 2 through 5 makes no difference in terms of the embedding. This mechanism is, however, difficult to move because the edge between vertices 1 and 2 and the edge between vertices 2 and 6 are on the straight line. The mechanism relies on buckling, which would only occur due to some imperfection; thus we discard this graph, too.

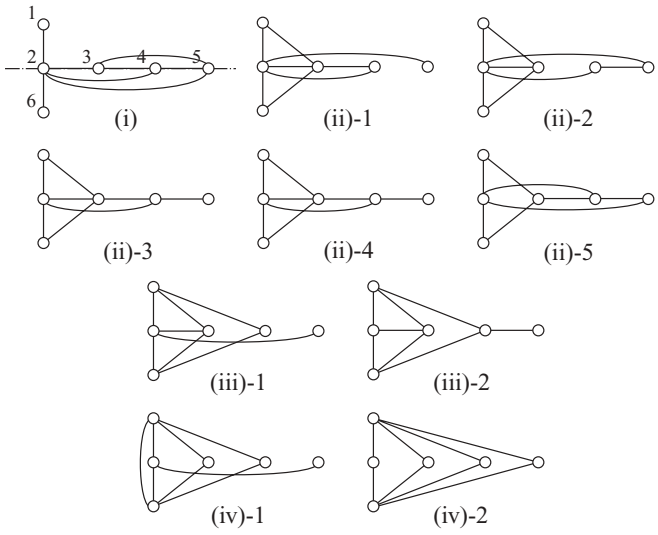


Fig. 5 Topological graphs generated from the vertex configuration F1

3.2 Case F2

There is a formula for the relation between the total vertex degrees and the total number of edges m in a graph (see, e.g., Kaveh (1991)):

$$\sum_{i=1}^{n} v_i = 2m \,. \tag{5}$$

Considering the symmetry of the vertex configuration F2, we obtain

$$v_1 + 2v_2 + 2v_3 + v_4 = 16 \,;\; v_2 = v_6 \,;\; v_3 = v_5 \,. \tag{6}$$

Embedding a graph from the vertex configuration F2 into the ground structure requires that one pair of the vertices 2 and 6 and the vertices 3 and 5 retain their symmetry. If the embedding involves any vertical reflection, it suffices to investigate the following cases.

$$v_1 \leq v_4 \,; \quad v_2 \leq v_3 \,. \tag{7}$$

Besides, because we are not interested in isolated edges and the maximum vertex degree is five (as the number of vertices is six), we can bound each variable as

$$2 \leq v_i \leq 5 \quad (i = 1, \ldots, 6) \,. \tag{8}$$

Based on (6) together with (7) and (8), the vertex configuration F2 produces six classes described in Table 1. Each class has at least one possible graph as shown in Fig. 6. In order to obtain a valid mechanism among the embeddings one has to require that a relative move between vertices 3 and 5 is possible (or, topologically equivalent, between vertices 2 and 6). For that reason, we can discard graphs (I)-1 and (I)-2 immediately because both pairs, vertex 2–6 and vertex 3–5, are linked directly. Likewise, graph (V) can be discarded because vertices 2 and 6 are linked directly and vertices 3 and 5 are on the adjacent two triangles, which form a rigid substructure in the graph. Graphs (IV)-1 and (IV)-2 are irrelevant because the motion of the mechanism is asymmetric. All in all, we can conclude that all the relevant graphs amounts to only five, namely graphs (I)-3, (II), (III)-1, (III)-2 and (VI). The original problem is thus reduced to five embedding subproblems, corresponding to each of the five

Table 1 Classification of topological graphs from F2 by vertex degrees

Class	v_1	v_2	v_3	v_4
I	2	3	3	2
II	2	2	4	2
III	2	2	3	4
IV	3	2	3	3
V	3	2	2	5
VI	4	2	2	4

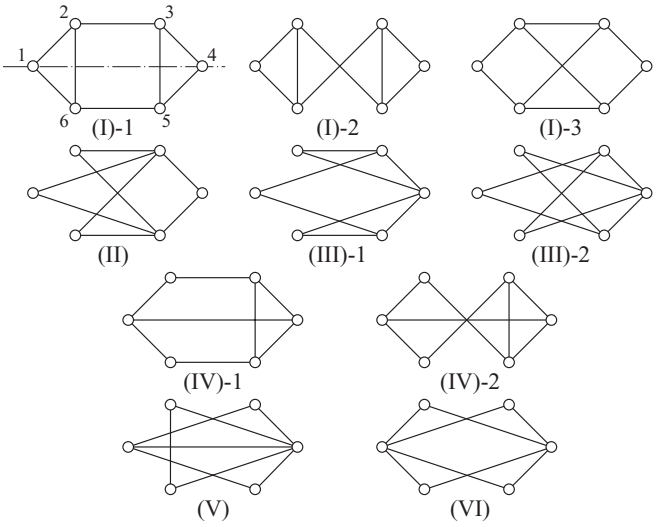


Fig. 6 Topological graphs generated from the vertex configuration F2

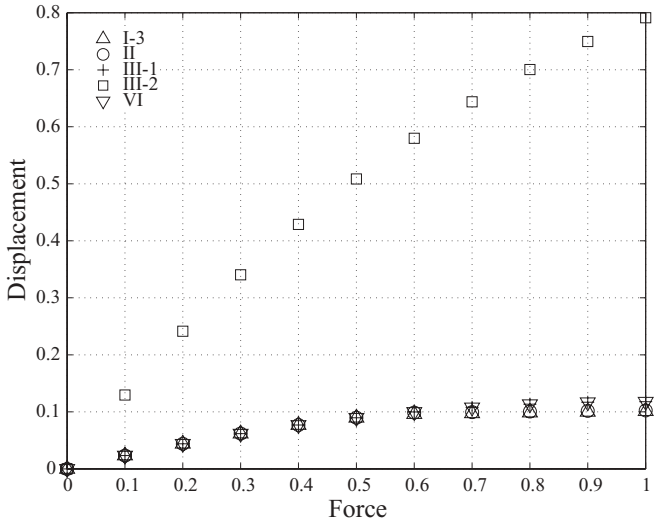


Fig. 8 Comparison of the performance

graphs. (II), (III)-1 and (III)-2 require 36 calculations for each. (I)-3 and (VI) need 18 calculations for each because of their vertical symmetry. The total enumeration corresponds to the evaluation of 144 possible mechanisms, one of which then constitutes the globally optimal (symmetric) mechanism.

Figure 7 shows the best mechanism among the embeddings of graph (III)-2 with the undeformed state depicted in grey and the deformed state in black. Also, this mechanism is the global optimum solution to the problem. Figure 8 compares the performance among the best mechanisms from the five different graphs. Graph (III)-2 is significantly better than the rest of the topologies, which all have a similar performance that is significantly lower than that of the optimal mechanism. This shows the great importance of also choosing good topologies when designing mechanisms – bad topologies cannot be expected to be improved significantly by changing the shape. Moreover, note that the optimal topology (III)-2 forms a scissors-like mechanism when embedded as the globally optimal mechanism and this is the reason for its outstanding performance. This kind of mechanism can not be obtained by the continuum-based topology optimization technique that has been very successful for compliant mechanism, suggesting that an inclusion of hinges in the "ground structure" is crucial for obtaining efficient articulated mechanisms.

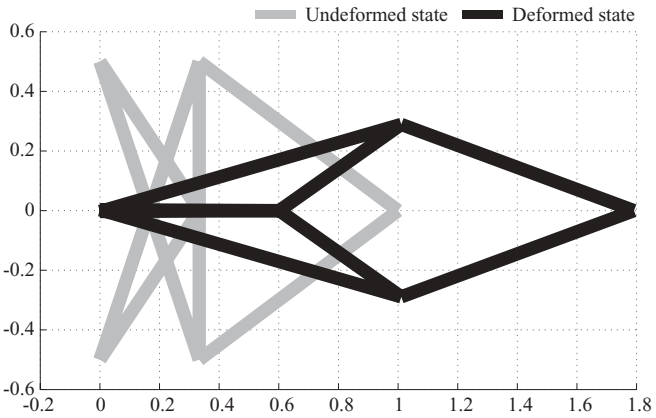


Fig. 7 Best mechanism from graph (III)-2

We close this technical note by remarking that the development above should be seen as a method to obtain a benchmark design that will be useful for further developments in the area; the key is that the *globally optimal solution* has been identified. It is not clear if the approach can be generalized to more complicated settings – it is the identification of the topological graphs that is a cornerstone of the method and here number synthesis, as described by Tischler (1995), may provide a viable path ahead.

Acknowledgements The authors gratefully acknowledge very helpful discussions with Carsten Thomassen on many aspects of graph theory.

References

Sigmund, O. 1997: On the design of compliant mechanisms using topology optimization. *Mech. Struct. Mach.* **25**(4), 493–524

Pedersen, C.B.W.; Buhl, T.; Sigmund, O. 2001: Topology synthesis of large-displacement compliant mechanisms. *Int. J. Numer. Methods Eng.* **50**(12), 2683–2705

Bruns, T.E.; Tortorelli, D.A. 2001: Topology optimization of non-linear elastic structures and compliant mechanisms. *Comput. Methods Appl. Mech. Eng.* **190**(26–27), 3443–3459

Ananthasuresh, G.K.; Kota, S.; Kikuchi, N. 1994: Strategies for systematic synthesis of compliant MEMS. *Dyn. Syst. Control* **2**, 677–686

Sigmund, O. 1996: Some inverse problems in topology design of materials and mechanisms. In: Bestle, D.; Schielen, W.

(eds.) *IUTAM Symposium on Optimization of Mechanism Systems,* pp 277–284. Dordrecht: Kluwer

Frecker, M.I.; Ananthasuresh, G.K.; Nishiwaki, S.; Kikuchi, N.; Kota, S. 1997: Topological synthesis of compliant mechanisms using multi-criteria optimization. *J. Mech. Des.* **119**(2), 238–245

Harary, F. 1969: *Graph theory*, pp 221–222. Boston: Addison-Wesley

Calladine, C.R. 1978: Buckminster Fuller's tensegrity structures and Clerk Maxwell's rules for the construction of stiff frame. *Int. J. Solids Struct.* **14**, 161–172

Pellegrino, S.; Calladine, C.R. 1986: Matrix analysis of statically and kinematically indeterminate frameworks. *Int. J. Solids Struct.* **22**(4), 409–428

Fowler, P.W.; Guest, S.D. 2000: A symmetry extension of Maxwell's rule for rigidity of frames. *Int. J. Solids Struct.* **37**(12), 1793–1804

Tischler, C.R.; Samuel, A.E.; Hunt, K.H. 1995: Kinematic chains for robot hands–I. orderly number-synthesis. *Mech. Mach. Theory* **30**(8), 1193–1215

Kaveh, A. 1991: *Structural mechanics: graph and matrix methods.* Hertfordshire, U.K.: Research Studies Press

[Paper 2]

INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2004; **61**:1520–1545
Published online 21 September 2004 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme.1119

Articulated mechanism design with a degree of freedom constraint

A. Kawamoto[1,‡], M. P. Bendsøe[1,*,†] and O. Sigmund[2]

[1]*Department of Mathematics, Technical University of Denmark, Matematiktorvet B303, DK-2800 Kgs. Lyngby, Denmark*
[2]*Department of Mechanical Engineering, Solid Mechanics, Technical University of Denmark, Nils Koppels Allè B404, DK-2800 Kgs. Lyngby, Denmark*

SUMMARY

This paper deals with design of articulated mechanism using a truss-based ground-structure representation. The proposed method can accommodate extremely large displacement by considering geometric non-linearity. In addition, it can also control the mechanical degrees of freedom (DOF) of the resultant mechanism by using a DOF equation based on Maxwell's rule. The optimization is based on a relaxed formulation of an original integer problem and also involves developments directed at handling the redundancy inherent in the ground-structure representation. One planar test example is selected as the basis for the developments so as to compare the proposed method with other alternative approaches including a graph-theoretical enumeration approach which guarantees the identification of the globally optimal solution. Also, an inverter problem is treated where a continuation method is required in order to direct the optimization algorithm towards an integer solution. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: mechanism design; topology optimization; degrees of freedom

1. INTRODUCTION

Research on design of compliant mechanisms [1–4] using topology optimization techniques in continuum structures suggests that it should also be possible to obtain results for articulated[§] mechanisms by applying such techniques [5, 6]. For design of articulated mechanisms the concept of *mechanical* degrees of freedom (DOF) becomes critical for obtaining a proper design. It is an insignificant feature for compliant mechanism design as such mechanisms do not have real joints. In the present paper, by DOF we mean the number of independent inputs

*Correspondence to: M. P. Bendsøe, Department of Mathematics, Technical University of Denmark, Matematiktorvet B303, DK-2800 Kgs. Lyngby, Denmark.
†E-mail: bendsoe@mat.dtu.dk
‡On leave from Toyota Central R&D Labs., Inc.
§From Oxford Advanced Learner's Dictionary: Articulate (*technical*), to be joined to something else by a joint, so that movement is possible.

Received 8 May 2003
Revised 21 January 2004
Accepted 24 February 2004

required to determine the position of all links of the mechanism with respect to a global co-ordinate system [7].

Kinematic diagrams (schematic drawings of mechanisms) are widely used for conventional mechanism designs and are especially helpful for the understanding of complicated mechanism topologies [7]. It seems natural and advantageous to represent kinematic diagrams by truss elements and pin joints. Firstly, one can directly interpret the truss topology from the kinematic diagram. Also, it is possible to accommodate large displacement without element distortion problems. Finally, with this interpretation one can formulate the optimization problems in terms of the so-called ground-structure approach for truss topology design [8]. The idea of the truss ground-structure approach is: we first give a large set of all potential elements in the design domain; then we eliminate unnecessary elements, or equivalently choose the necessary elements, according to the requirements; finally we obtain the resultant mechanism design.

In this paper, we introduce a constraint on the mechanical degrees of freedom of the mechanism, based on Maxwell's rule [9–11]. However, Maxwell's rule requires that every sub-structure contains no redundant elements. Thus we cannot directly apply the formula to the ground structure which is intrinsically redundant as it includes all potential elements. Therefore, the number of bars cannot be counted in a straightforward way. We have two possibilities: one is to introduce additional constraints to eliminate the redundant elements; the other is to preserve the redundant elements but ignore them in the DOF calculation. The first idea is similar to the sub-tour elimination constraints of the famous *Traveling Salesman Problem* [12]. An alternative, and the one we apply, works by detecting the number of independent bars in the ground structure. The number of independent bars is related to the rank of the (linear) stiffness matrix in the unsupported condition which for our purposes suitably can be determined by an eigenvalue analysis. However, due to the regularity of the nodal positions, this method ignores also overlapping elements which are often useful for large displacement mechanism. In order to distinguish overlapping bars as independent, we need to rearrange the nodal positions and this can conveniently be achieved by relocating all the nodes onto a circle. In this way a standard eigenvalue analysis is capable of identifying the number of independent bars, allowing for the implementation of the constraint on the degrees of freedom. Furthermore, we shall relax the DOF equation so as to solve the problem with gradient-based optimization methods such as MMA—the method of moving asymptotes [13]. Thus we here introduce a differentiable approximation of the step function in order to count non-zero eigenvalues for independent bars and to count non-zero nodal degrees for active nodes.

2. TRUSS REPRESENTATIONS OF ARTICULATED MECHANISMS

Kinematic diagrams (or skeleton diagrams) are the schematic drawings commonly used for conventional mechanism designs and they are especially helpful for the understanding of complicated mechanism topologies [7]. It seems natural to represent kinematic diagrams by truss elements and pin joints because of the following advantages:

- One can directly interpret the truss topology from the kinematic diagram.
- It is possible to accommodate extremely large displacement without any element distortion problems.

- With this interpretation one can formulate the optimization problems in terms of the so-called ground-structure approach for truss topology design [8].

Normally articulated mechanisms are modelled by rigid bars and revolute joints. Nevertheless we use truss representations for articulated mechanisms in the present paper. But note that the articulated mechanisms are not supposed to rely on the elasticity in the final designs. In other words, the functionality of articulated mechanisms never changes after all the truss elements are replaced with rigid bars. This is the critical difference between articulated and compliant mechanisms.

3. DEFINITION OF A TEST EXAMPLE

The test example that we employ is shown in Figure 1. The figure shows the truss ground structure of possible nodes and connections, as well as all necessary boundary conditions, load cases and other related parameters. The springs and forces at the input ports model a strain-based input actuator (e.g. shape memory alloy or piezoelectric device) and the output spring models a resistance to the movement. One relevant mechanism design problem is to maximize the output displacement D_{out} for the given input forces F_{in} by picking M^* truss elements out of the M possible connections, so that the resultant mechanism has $f = 1$ degree of freedom when supported in a statically determinate manner, and so that it is symmetric with respect to the line between nodes 2 and 11. This problem can be formulated as the following binary integer optimization problem for the truss connectivity vector $(\chi_1, \ldots, \chi_M)$:

MECH1:

$$\max_{\chi} : D_{\text{out}}$$
$$\text{s.t.} : \sum_{e=1}^{M} \chi_e = M^*$$

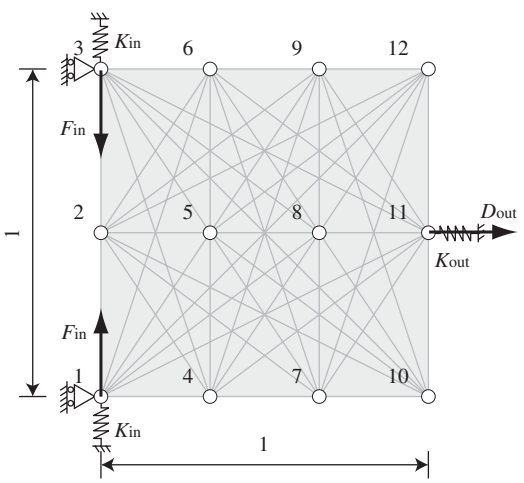


Figure 1. Ground structure.

$$: \chi_e \in \{0, 1\}$$

$$: f = 1$$

$$: \text{symmetric}$$

Here D_{out} is given via the displacement vector U which is a solution to an equation $R(U) = 0$ that expresses the residual version of the equilibrium equations. Geometrical non-linearity is here taken into account, so we use the Green–Lagrange strain measure and adopt a force incremental method with Newton–Raphson iterations for evaluation of the displacement D_{out}.

We note that the global optimum mechanism design is known for **MECH1**, in the case of $M^* = 8$ (this result is based on a graph theoretical enumeration method) [14]. This provides for a solid comparison of the methods that will be developed in the following.

4. RELAXATION

The complexity of the problem, namely the number of all possible combination of bars is calculated as

$$_M C_{M^*} := \frac{M!}{M^*!(M - M^*)!} \tag{1}$$

This number 'explodes' with the increase of M for a certain number M^* of desired bars. Thus it seems reasonable to relax the original problem in order to use gradient-based optimization techniques. One possible relaxation is to permit each variable χ_e to take on intermediate values between 0 and 1. We thus define ρ as the relaxed truss connectivity vector $(\rho_1, \ldots, \rho_M)$ (and leave out the degree of freedom constraint for the moment). The relaxed problem can then be formulated as follows:

RELAX1:

$$\max_{\rho} : D_{\text{out}}^{\text{non-linear}}$$

$$\text{s.t.} : \sum_{e=1}^{M} \rho_e \leqslant M^* + \alpha$$

$$: 0 < \rho_{\min} \leqslant \rho_e \leqslant 1$$

$$: \text{symmetric}$$

where $\rho_{\min}$ is a positive lower bound of the design variables ρ_e ($= 0.001$ typically). This is introduced in order to avoid numerical difficulties such as a singular stiffness matrix and an insensitivity of the objective function to changes in the design variables around zero values. The term α compensates the effect of $\rho_{\min}$ on satisfying the bound M^* (i.e. $\alpha = (M - M^*)\rho_{\min}$).

This kind of problem can be solved by applying gradient-based optimization methods. In the present research, we have used the algorithm named MMA (method of moving asymptotes) [13]. When applying such an algorithm one needs to perform a sensitivity analysis, here for the output displacement $D_{\text{out}}^{\text{non-linear}}$. For this we use the standard adjoint technique (see, for example Reference [2] for an explicit derivation for the present situation).

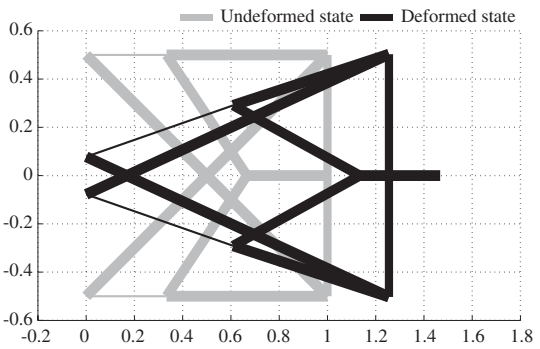


Figure 2. Example I: a solution to **RELAX1**, $D_{\text{out}} = 0.4673$.

Note that we search for an optimal solution following a path through a design space of approximate mechanisms that take relaxed intermediate values of the design variables. Thus the proposed method implicitly assumes that the binary solution can be found as a limit of the relaxed design variables (i.e. density). Therefore, the method cannot be expected to solve the problem if the binary solutions are isolated in the relaxed design space.

4.1. Penalization

In order to recover the 0–1 values of χ, we will penalize the intermediate values of ρ_e by expressing the cross-sectional areas $A_e, e = 1, \ldots, M$ of the bars as:

$$A_e = \rho_e^P A_{\max} \tag{2}$$

where $P > 1$ ($= 3$ typically) and where $A_{\max}$ is an upper bound. Since we have a constraint $M^* < M$ on the total element number of bars the penalization makes the intermediate values less efficient, thus typically forcing ρ to go to the extremums for optimized designs, i.e. $\rho_{\min}$ or 1. This is a typical technique used in continuum-based topology optimization problems [8].

4.2. Example I

For the test example **MECH1**, the global optimum mechanism design is, as already mentioned, known for the case of $M^* = 8$ [14]. For comparison, we will here also show a result obtained with **RELAX1**, for $M^* = 8$.

Figure 2 shows a solution to **RELAX1**. The bars at the lower bound are made invisible simply for aesthetic improvements. If we accept the intermediate solution, this truss may work as a mechanism. But it has more than 1 degree of freedom; even if the input ports are fixed, it still has a moving part at the right end. Note that this mechanism relies on two thin bars in order to function. However, the intermediate solution is not feasible for the original integer formulation **MECH1**. Without these thin bars the mechanism would not work at all. In general, formulation **RELAX1** may give this kind of rather spurious mechanisms (as are also seen in the literature).

5. ALTERNATIVES TO THE DOF CONSTRAINT

The formulation **RELAX1** is obtained by relaxing the 0–1 constraint on the truss connectivity vector. This, however, tends to give unstable mechanisms which have 2 or more degrees of freedom, and it seems imperative to introduce a constraint on the degrees of freedom. But before we proceed with this, we first examine two alternative ideas that may improve the stability of the mechanism against disturbances.

5.1. A mean compliance constraint

A mechanism that has exactly $f = 1$ degree of freedom is called *a constrained mechanism* [7]. This means that the mechanism becomes rigid when one of the revolute joints is fixed. One might expect that this kind of mechanism can be obtained for the test example by introducing an additional perpendicular load case at the output port, as shown in Figure 3. In other words, the resultant mechanism is expected to form a stiff structure against the perpendicular load F_p at the output port when the two input ports are firmly supported. The requirement can be considered in a modified problem, **RELAX2**, as an additional constraint for the mean compliance with respect to the load F_p:

$$C_p = F_p^{\top} U_p \tag{3}$$

where the displacement vector U_p is a solution to a *linear* equilibrium equation for the unactuated mechanism.

This results in a problem form:

RELAX2:

$$\begin{aligned} \max_{\rho} &: D_{\text{out}} \\ \text{s.t.} &: \sum_{e=1}^{M} \rho_e \leqslant M^* + \alpha \\ &: 0 < \rho_{\min} \leqslant \rho_e \leqslant 1 \\ &: \text{symmetric} \\ &: C_p \leqslant C_p^* \end{aligned}$$

where C_p^* is the constant that gives the upper bound for the mean compliance C_p.

5.2. Example II

We will also solve **RELAX2** when $M^* = 8$. We examine the following three examples with the mean compliance constraint values: $C_p^* = 0.001$, 0.0001 and 0.00001. In the examples, the magnitude of the perpendicular load is set as $||F_p|| = 0.01$. Thus, the three upper bounds for the mean compliance correspond to the displacements 0.1, 0.01 and 0.001 in absolute value, respectively. Table I shows the optimization results of the three examples.

Figure 4 shows the resultant mechanisms of the three examples; we only show the effect of the perpendicular load case for Example II-1, see the upper right figure in Figure 4. The mechanism of Example II-1 in Figure 4 has obviously more than 1 degrees of freedom. This can be improved by tightening the mean compliance constraint like Example II-2 and II-3 as

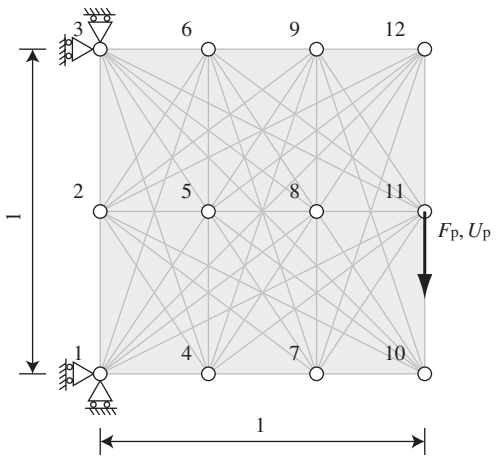


Figure 3. Perpendicular load case.

Table I. Perpendicular load cases.

Example	C_p^*	C_p	D_{out}
II-1	0.001	0.001	0.5392
II-2	0.0001	0.0001	0.2579
II-3	0.00001	0.00005	0.19541

shown in Figure 4. Note that the mean compliance constraint in Example II-3 is too tight to find any feasible solution. However, the resultant mechanisms become more dependent on their flexibility in the way they function. In other words, they cannot move without elongating some bars. Although the perpendicular load case thus may have some effect on the stabilization of the mechanism against disturbances, it seems to conflict with the objective function.

5.3. Node number constraint

Another idea for obtaining a proper mechanism is to apply a constraint on the number of active nodes. Assuming that the mechanism has *no redundant elements* (we shall discuss this in detail later), the degrees of freedom of a mechanism that is supported in a statically determinate manner can be calculated by the following DOF equation based on Maxwell's rule (in 2D) [9–11]:

$$f = 2n - m - 3 \tag{4}$$

Here, n is the number of nodes, m is the number of bars, and '-3' means that the mechanism is supported in a statically determinate manner. In the case of $f = 1$(DOF) and $m = 8$(bars), a possible mechanism has to have $n = 6$(nodes).

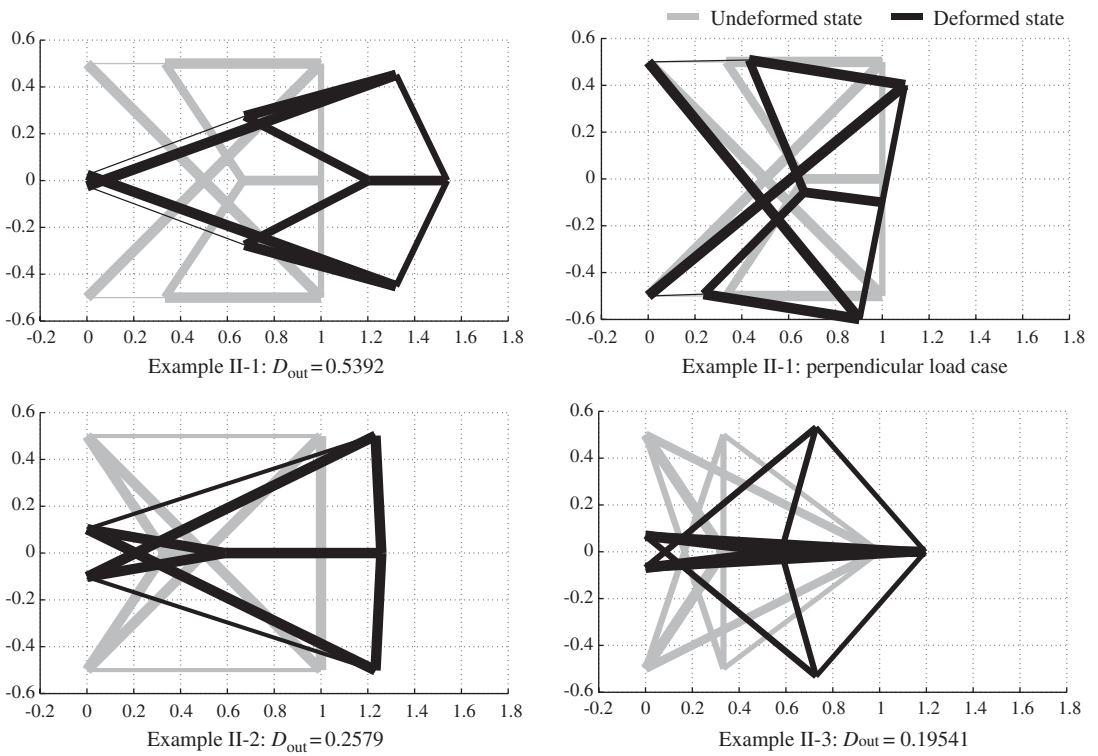


Figure 4. Example II: solutions to **RELAX2**.

In order to introduce the node number constraint into **RELAX1**, we need to relax the active node number n into an approximate number $\tilde{n}$ which, for example, can be calculated as

$$\tilde{n} = \sum_{j=1}^{N} h_1(\tilde{d}_j) \tag{5}$$

Here N is the total node number of the ground structure, $\tilde{d}_i$ is a relaxed counting of the number of incident edges, and $h_1(x)$ is a differentiable approximation of the step function. Here we apply the sigmoid function $h_1(x)$ given as:

$$h_1(x) := \frac{1}{1 + \exp[-T_1(x - H_1)]} \tag{6}$$

Figure 5 shows the case where $(T_1, H_1) = (50, 0.75)$.

As mentioned, $\tilde{d}_j$ is a relaxed value that counts the number of incident edges d_j, that is, the degrees of the jth node in the sense of graph theory [15]. The relaxed $\tilde{d}_j$ is calculated as

$$\tilde{d}_j = \sum_{e \in AL_j} \rho_e^P \tag{7}$$

where AL_j means the adjacency list of edges which are incident with the jth node [16].

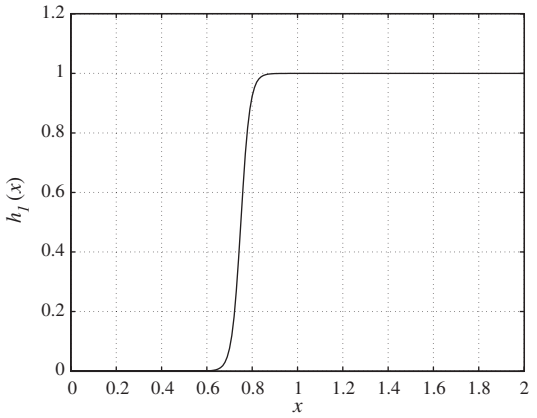


Figure 5. Sigmoid function: $h_1(x)$.

In conclusion, we obtain the following relaxed formulation with node number count:

RELAX3:

$$\begin{aligned} \max_{\rho} &: D_{\text{out}} \\ \text{s.t.} &: \sum_{e=1}^{M} \rho_e \leqslant M^* + \alpha \\ &: 0 < \rho_{\min} \leqslant \rho_e \leqslant 1 \\ &: \text{symmetric} \\ &: \tilde{n} \leqslant N^* \end{aligned}$$

where N^* is the upper bound of the active node number.

5.4. Example III

The problem statement **RELAX3** may give a solution as shown in Figure 6. The right graph in Figure 6 depicts the connectivity of this mechanism with a schematic drawing. Although the mechanism has 8 edges and 6 nodes, it still has 2 degrees of freedom. This is because the resulting mechanism violates the assumption for Maxwell's rule as expressed in Equation (4) namely that the mechanism has no redundant elements. More precisely, the assumption is that every *sub-structure* of the truss has no redundant elements. The curved edge of the schematic drawing in Figure 6 illustrates the redundant element which reinforces the stiffness of the right triangular sub-structure, but which contributes little to the function of the mechanism. This suggests that we should take account of the redundancy when counting the node number and the element number as they appear in Equation (4).

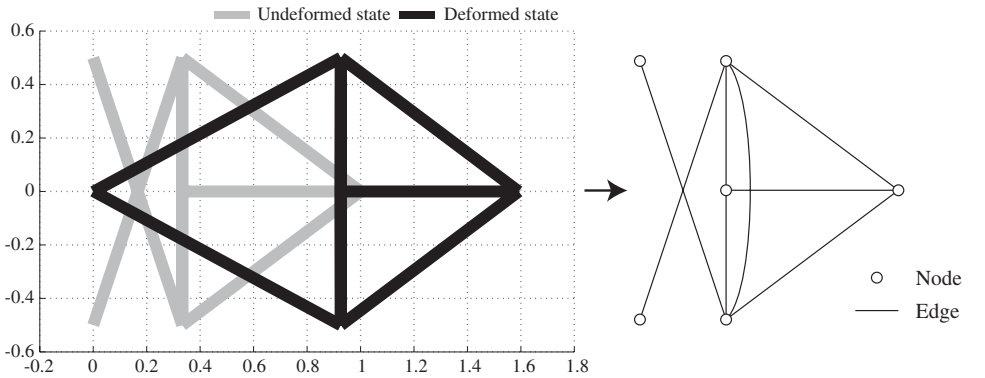


Figure 6. Example III: a solution to **RELAX3**, $D_{\text{out}} = 0.5919$.

6. INTRODUCTION OF A DOF CONSTRAINT

The examples above indicate that it will be useful to work directly with a constraint on the degrees of freedom. Also, one should not directly apply the DOF Equation (4) to the ground structure because it has all potential elements present and intrinsically has some redundancy, i.e. we cannot count the number of bars in a straightforward way.

We envisage two possibilities for catering for this:

1. Introduce additional constraints to eliminate the redundant elements.
2. Preserve the redundant elements but ignore them in the DOF calculation.

The first idea is similar to the sub-tour elimination constraints of the famous *Traveling Salesman Problem* [12] and it involves the investigation of all possible sub-structures. This may be the subject of future work, but here we do not go further in this direction.

Instead, we shall pursue the second possibility which can be accomplished by detecting the number of *independent bars* $\hat{m}$ in the ground structure. The number of independent bars $\hat{m}$ is given as the rank of the direction cosine matrix H in the equilibrium equation between the element force vector and the nodal force vector. From a computational point of view, it is useful to use the relation between the number of the independent bars $\hat{m}$ and the *linear* stiffness matrix K_F in the unsupported condition. As we have that

$$K_F = HDH^{\top} \tag{8}$$

where D is a diagonal matrix whose entries are the stiffness of the bars (D is positive definite), we have that the rank of the direction cosine matrix H is the same as the rank of the global stiffness matrix K_F.

Thus we can use the rank of the global stiffness matrix K_F to count the number of independent bars $\hat{m}$:

$$\hat{m} = \text{rank}(K_F) \tag{9}$$

and Equation (4) for calculating the degrees of freedom is now modified to the form,

$$f = 2n - \hat{m} - 3 \tag{10}$$

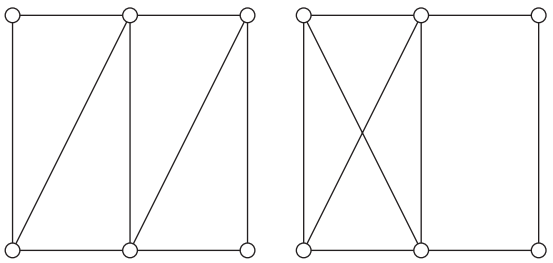


Figure 7. Two plane trusses which have 6 nodes and 9 edges.

We shall demonstrate how this idea works with the following examples which are also discussed by Calladine [9]. In Figure 7, both trusses have $n = 6$ nodes and $m = 9$ edges. If we applied the DOF Equation (4) to them without any consideration, we would have $f = 0$ DOF in both cases. In fact, truss (a) is a simple stiff truss while truss (b) is partly redundant and partly a mechanism. Hence we need to detect the number of the independent bars in order to know the correct number of degrees of freedom of truss (b). Since the stiffness matrix K_F is real, symmetric and positive semidefinite, the rank of K_F is equal to the number of non-zero eigenvalues. Figure 8 shows the results of standard eigenvalue analyses for the two trusses. We see that truss (a) has 9 non-zero eigenvalues (i.e. $\hat{m} = 9$) while truss (b) has 8 non-zero eigenvalues (i.e. $\hat{m} = 8$). Substituting these results into Equation (10), the correct number of degrees of freedom can be given for both cases, namely $f = 0$ DOF for truss (a) and $f = 1$ DOF for truss (b).

Another consideration we should take into account is that the ground structure has some elements that overlap. This is an effect of the regularity of the nodal positions. The problem caused by such overlapping bars can be illustrated by the following simple 3 bar truss, as in Calladine [9]. The truss in Figure 9 has 3 bars with 3 joints that are on line in the undeformed configuration. A transverse displacement Δ generates the restoring force which equilibrates the load F [9]:

$$F = \frac{AE\Delta^3}{2a^3} \tag{11}$$

The restoring force is proportional to the third power of the displacement Δ. Therefore, Equation (9) cannot distinguish overlapping bars as independent since K_F is based on a linear analysis. On the other hand, we can raise the effect of the restoring force from the third power to a linear one in the displacement Δ by giving the central node of the truss in the undeformed configuration a finite offset in the vertical direction.

In the realm of geometric non-linearity, this kind of overlapping elements may give very useful mechanisms such as the scissors shown in Figure 10, and we would like to preserve this possibility in the mechanism design formulation.

From the discussion above it is apparent that we can avoid the difficulty of detecting useful overlapping elements by rearranging the nodal positions. One idea is to create what we call a 'connectivity circle', as shown in Figure 11, by relocating all the nodes on a unit circle to the

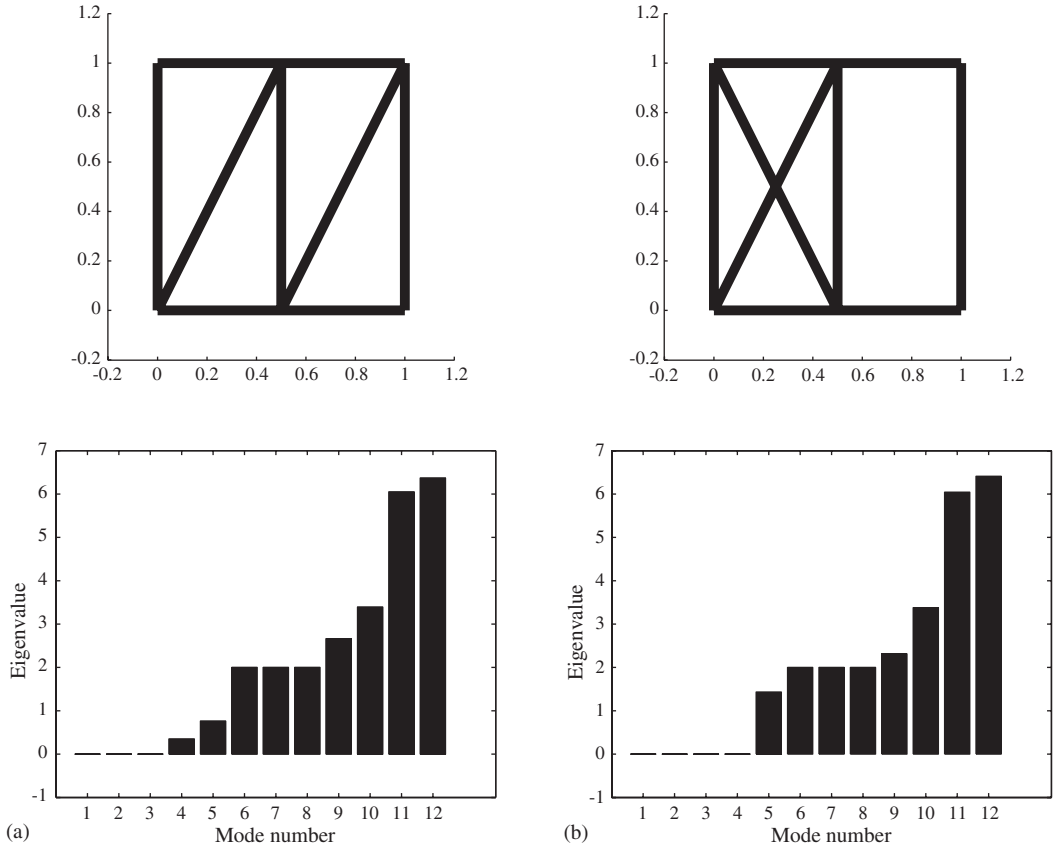


Figure 8. Detection of independent bars by eigenvalue analysis.

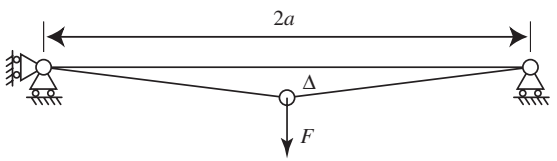


Figure 9. Overlapped 3 bar truss.

following positions (for further details see Appendix A):

$$[x_i, y_i] = [\cos\theta_i, \sin\theta_i] \tag{12}$$

$$\theta_i = \frac{2\pi}{N}((i-1) + \beta 2^{-2^i}) \tag{13}$$

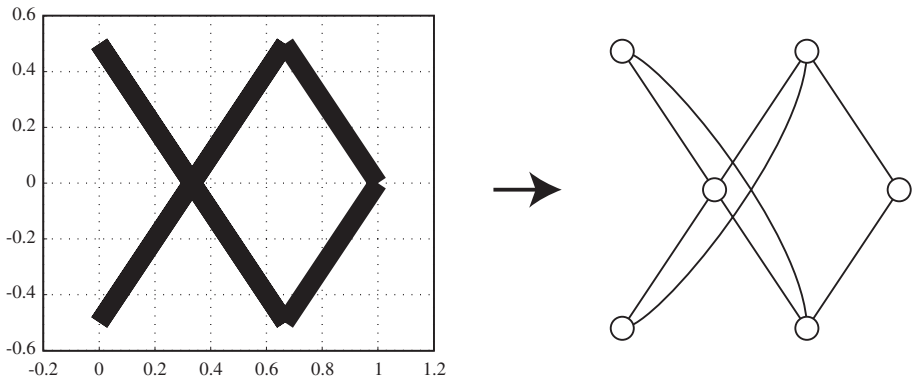


Figure 10. Scissors-like mechanism.

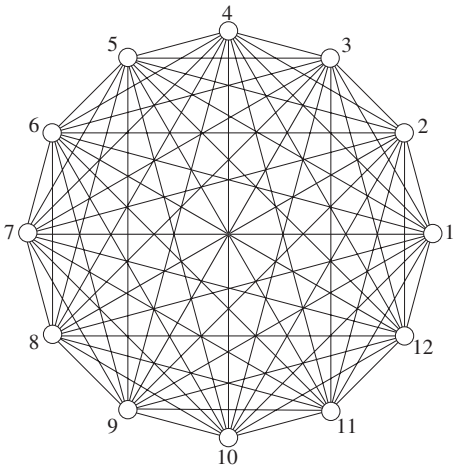


Figure 11. Connectivity circle: $\beta = 0$.

where $i = 1, 2, \ldots, N$ is the nodal numbering, and β is an optional parameter for avoiding parallel edges in the circle. Then we redefine the number of independent bars as the number

$$\hat{m} = \text{rank}(K_R) \tag{14}$$

where K_R is the new (linear) stiffness matrix obtained with the circular nodal positions in the unsupported condition and the same connectivity as the original truss. Since K_R is real, symmetric and positive semidefinite, the rank of K_R is equal to the number of non-zero eigenvalues, and we apply this for evaluating the rank, as we need to generate a differentiable relaxation of it. The eigenvalue problem for K_R is here written as

$$K_R \phi_i = \lambda_i \phi_i, \quad i = 1, \ldots, 2N \tag{15}$$

where λ_i is the eigenvalues, and ϕ_i is the corresponding eigenvector and the following orthogonality is assumed (δ_{ij} denotes Kronecker's delta):

$$\phi_i^\top \phi_j = \delta_{ij}, \quad i, j = 1, \ldots, 2N \tag{16}$$

$$\phi_i^\top K_R \phi_j = \lambda_i \delta_{ij}, \quad i, j = 1, \ldots, 2N \tag{17}$$

6.1. Relaxation of the DOF equation

In order to introduce the degree of freedom equation (10) into **RELAX1**, we need to relax this equation as well. We have already relaxed the number of active nodes as $\tilde{n}$ in Equation (5). The number of independent bars $\hat{m}$ can be relaxed as

$$\tilde{m} = \sum_{i=1}^{2N} h_2(\lambda_i) \tag{18}$$

where N is the total node number of the ground structure. In the two-dimensional case, each node has 2 DOF, and the total degrees of freedom amounts to $2N$. The function $h_2(x)$ is a differentiable approximation of the step function, defined as in Equation (6) (with parameters T_2 and H_2):

$$h_2(x) := \frac{1}{1 + \exp[-T_2(x - H_2)]} \tag{19}$$

Assuming that the eigenvalues are simple, the derivative of $\tilde{m}$ with respect to ρ_e exists and is calculated as

$$\frac{\mathrm{d}\tilde{m}}{\mathrm{d}\rho_e} = \sum_{i=1}^{2N} h_2'(\lambda_i) \frac{\mathrm{d}\lambda_i}{\mathrm{d}\rho_e} \tag{20}$$

where we have used that the derivative of the eigenvalue exists (cf., e.g. Reference [17] and references therein). Moreover, we have the following expression for the simple eigenvalues:

$$\frac{\mathrm{d}\lambda_i}{\mathrm{d}\rho_e} = \phi_i^\top \frac{\partial K_R}{\partial \rho_e} \phi_i \tag{21}$$

For the case of multiple eigenvalues we assume that the solution to Equation (15) yields a N_L-fold eigenvalue:

$$\bar{\lambda} = \lambda_i, \quad i = 1, 2, \ldots, N_L \tag{22}$$

where we can choose a set of orthonormal eigenvectors ϕ_i $(i = 1, 2, \ldots, N_L)$ which span the eigenvector subspace of the eigenvalue in question. It is known that the multiple eigenvalue is non-differentiable [17–20], and that a sub-differential exists and is given as follows [18]:

$$\partial\bar{\lambda} = \left\{ \sum_{i=1}^{N_L} w_i \phi_i^\top \frac{\partial K_R}{\partial \rho_e} \phi_i : w_i \geqslant 0 \text{ for all } i \text{ and } \sum_{i=1}^{N_L} w_i = 1 \right\} \tag{23}$$

Also, for the case at hand, the right-hand side of Equation (21) only gives eigenvalue sub-gradients [21], here labelled as,

$$\lambda_i'(\phi_i) := \phi_i^\top \frac{\partial K_R}{\partial \rho_e}\phi_i, \quad i = 1, 2, \ldots, N_L \tag{24}$$

For considering the differentiability of Equation (18) we first remark that the function value $h_2(\bar{\lambda})$ assigns equal weights to the multiple eigenvalues. Thus it suffices to consider a straight summation of all the sub-gradients of Equation (24) and to see that this is independent of the choice of eigenvectors, meaning that the relevant sub-differential contains just one element. In other words, we note that the summation of all the N_L-fold eigenvalue sub-gradients $\lambda_i'(\phi_i)$ is uniquely determined and is independent on the choice of eigenvectors. To see that this holds, choose another arbitrary set of orthonormal vectors $\bar{\Phi} = [\bar{\phi}_1, \bar{\phi}_2, \ldots, \bar{\phi}_{N_L}]$ for the eigenvalue $\bar{\lambda}$ such that

$$K_R\bar{\Phi} = \bar{\lambda}\bar{\Phi}$$

$$\bar{\Phi}^\top\bar{\Phi} = I$$

There then exists a $N_L \times N_L$ orthogonal matrix R (i.e. $R^\top R = I$) that satisfies

$$\bar{\Phi} = \Phi R$$

where $\Phi = [\phi_1, \phi_2, \ldots, \phi_{N_L}]$ is the original orthonormal eigenvectors in Equations (23) and (24). Therefore, we can calculate the summation of the sub-gradients as

$$\begin{aligned}
\sum_{i=1}^{N_L} \lambda_i'(\bar{\phi}_i) &= \sum_{i=1}^{N_L} \bar{\phi}_i^\top \frac{\partial K_R}{\partial \rho_e}\bar{\phi}_i \\
&= \mathrm{tr}\left[\bar{\Phi}^\top \frac{\partial K_R}{\partial \rho_e}\bar{\Phi}\right] \\
&= \mathrm{tr}\left[R^\top\Phi^\top \frac{\partial K_R}{\partial \rho_e}\Phi R\right] \\
&= \mathrm{tr}\left[\Phi^\top \frac{\partial K_R}{\partial \rho_e}\Phi\right] \quad (\because \text{similar transformation [22]}) \\
&= \sum_{i=1}^{N_L} \phi_i^\top \frac{\partial K_R}{\partial \rho_e}\phi_i \quad \left(= \sum_{i=1}^{N_L} \lambda_i'(\phi_i) \text{ in Equation (24)}\right)
\end{aligned}$$

This means that the summation of the sub-gradients $\lambda_i'(\phi_i)$ is independent of the choice of eigenvectors, and that this summation is differentiable.[¶]

[¶]We note that the summation of all eigenvalues equals the trace of the matrix. In this case differentiability follows directly.

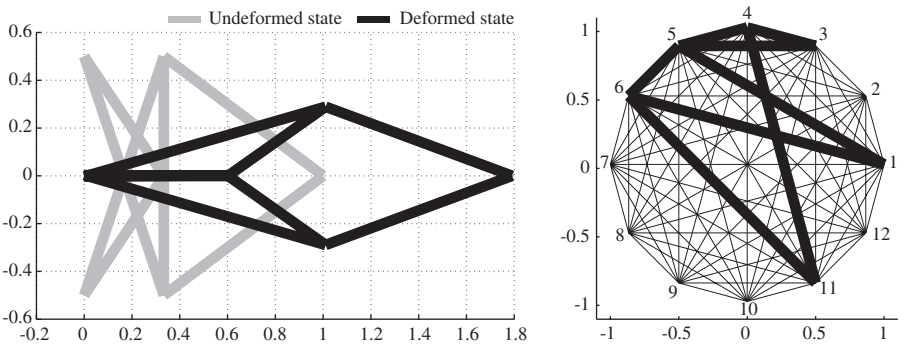


Figure 12. Example IV: a solution to **RELAX4**, $D_{\text{out}} = 0.7913$ and the connectivity circle.

From this we thus infer that the expression for the sensitivity of $\tilde{m}$ for both simple and non-simple eigenvalues can be written as

$$\frac{\mathrm{d}\tilde{m}}{\mathrm{d}\rho_e} = \sum_{i=1}^{2N} h_2'(\lambda_i)\phi_i^\top \frac{\partial K_R}{\partial \rho_e}\phi_i \tag{25}$$

for any choice of orthonormal eigenvectors.

Finally, with these developments, we can introduce the relaxed and differentiable version of the constraint on the degrees of freedom as the expression:

$$\tilde{f} = 2\tilde{n} - \tilde{m} - 3 \tag{26}$$

and apply it to the following formulation of our optimization problem:

RELAX4:

$$\begin{aligned} &\max_{\rho} : D_{\text{out}} \\ &\text{s.t.} : \sum_{e=1}^{M} \rho_e \leqslant M^* + \alpha \\ &\quad : 0 < \rho_{\min} \leqslant \rho_e \leqslant 1 \\ &\quad : \text{symmetric} \\ &\quad : \tilde{f} \leqslant 1 \end{aligned}$$

6.2. Example IV

We have also solved **RELAX4** in the case of $M^* = 8$ bars. Figure 12 shows the resultant mechanism and the connectivity circle. The obtained mechanism in Figure 12 is a feasible solution in the original problem **MECH1** as well, and interestingly enough it equals the global optimum solution found by a graph-theoretical enumeration approach in Reference [14].

Figure 13 shows the optimization histories of Example IV. The optimization process required 29 iterations to converge, which took about 72.5 s with a Matlab code implemented on a Pentium4-1.7 GHz CPU. One notes that the objective function value increases monotonously.

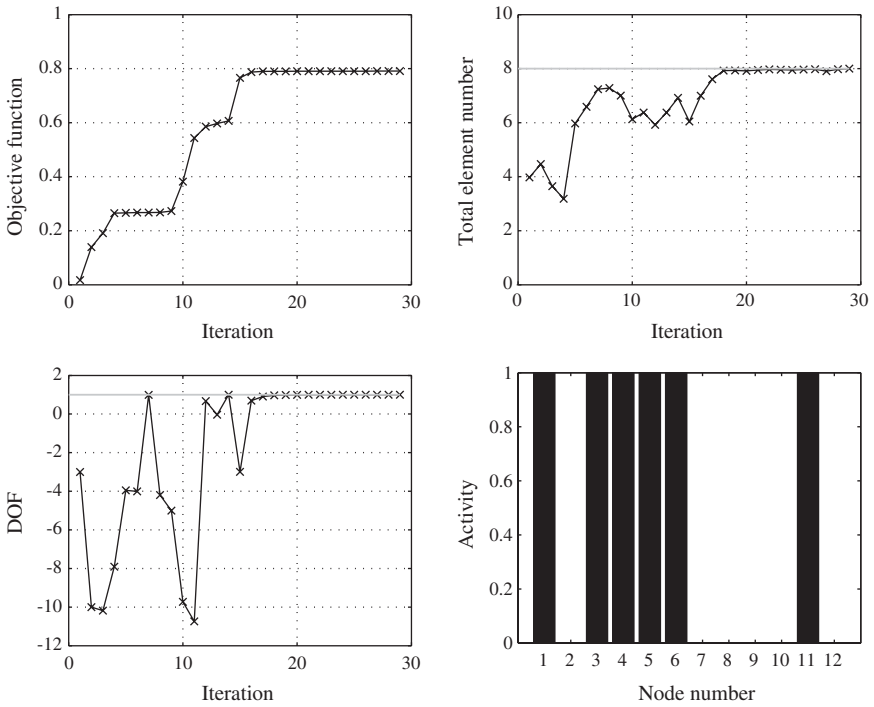


Figure 13. Optimization histories and active nodes at the optimum.

Also, both the constraints on the degrees of freedom and on the total number of elements (depicted as thin lines in Figure 13) are not violated during the optimization process.

We remark here that the constraint $\tilde{f} \leqslant 1$ of **RELAX4** involves the application of two sigmoid functions with parameters (T_1, H_1) and (T_2, H_2), respectively. The choice of these parameter values is somewhat arbitrary, but does influence the results. T_1 and T_2 are connected with the sharpness of the 'step' of the sigmoid functions, and H_1 and H_2 are the thresholds between active and inactive counting of active nodes or non-zero eigenvalues. Good values of (T_2, H_2) can be determined by the longest truss of a connectivity circle. The parameter T_2 is preferred to be as small as possible for numerical stability, but not to be so small that the function fails to recognize an independent bar as active. We have more freedom in the choice of (T_1, H_1). If H_1 is small, the recognition of the active nodes will be delayed. This means that the constraint on the degrees of freedom is made looser so it is easier to find a feasible starting point. The larger H_1 becomes, the more conservative is the degree of freedom constraint and the more difficult it is to find a feasible starting point.

6.3. Effectiveness of the DOF constraint

Table II shows the results of the application of the proposed degree of freedom equation (26) to the results of Examples I–IV. The introduction of the degree of freedom constraint $\tilde{f} \leqslant 1$ eliminates the solutions found in Example I, II-1 and III. It is especially noteworthy that for the solution of Example III the DOF equation has successfully detected the 7 independent bars (instead of 8) and has given the correct number of degrees of freedom as $\tilde{f} = 2$.

Table II. Application of the degree of freedom equation.

Examples	$\tilde{f}$ (DOF)	$\tilde{m}$ (bars)	$\tilde{n}$ (nodes)
I	5.0	8.0	8.0
II-1	4.0	9.0	8.0
II-2	1.0	8.0	6.0
II-3	0.0	9.0	6.0
III	2.0	7.0	6.0
IV	1.0	8.0	6.0

Table III. Mechanisms which have $f \leqslant 1$ DOF.

Example	$\tilde{f}$ (DOF)	C_p	D_{out}
II-2	1.0	0.00010	0.2579
II-3	0.0	0.00005	0.1954
IV	1.0	0.00028	0.7913

For further investigation of the examples which have $\tilde{f} \leqslant 1$ DOF (Examples II-2, II-3, and IV) we need to include an evaluation of the mean compliance given by perpendicular load case of Figure 3. Table III shows this comparison. The mean compliance of Example IV is larger than those of Example II-2 and II-3, so both the compliance constraints were too tight relative to obtaining the result of Example IV. This underlines the difficulty in applying the mean compliance constraint as a method for controlling the degrees of freedom of the mechanism.

7. A SECOND TEST EXAMPLE

The test example used above has the advantage that the global optimum solution is known. Here we now proceed with a problem of generating an *inverter mechanism* utilizing a larger ground structure.

7.1. Inverter mechanism problem

The inverter mechanism problem is shown in Figure 14. The nodes are numbered from left to right and from bottom to top. The ground structure contains all possible nodes and truss connections as well as all necessary boundary conditions, load cases and other related parameters. The binary integer formulation is basically the same as **MECH1**: the maximization of the output displacement $D_{\text{out}}^{\text{non-linear}}$ for the given input forces F_{in} by picking 8 truss elements out of the 300 possible connections, so that the resultant mechanism has $f = 1$ DOF when supported in a statically determinate manner. Also, the mechanism should be symmetric with respect to the line between nodes 3 and 23. We apply **RELAX4** to the inverter problem and solve it, as above, by MMA.

We have obtained a result as shown in Figure 15 (here thin bars close to the lower bound ($\rho_e < 5 \cdot \rho_{\min}$) are made invisible). Figure 16 shows the optimization histories of the objective function value, the total element number, the degrees of freedom, as well as the active nodes

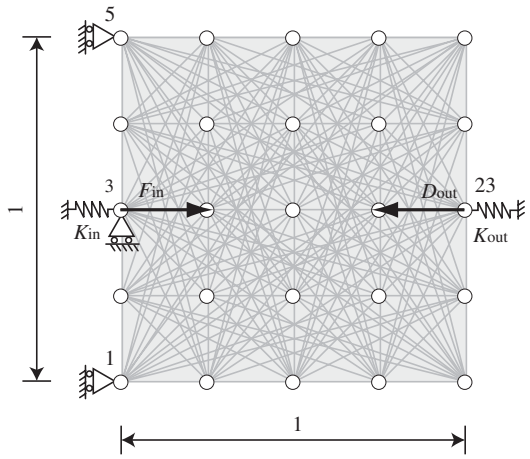


Figure 14. Inverter mechanism problem.

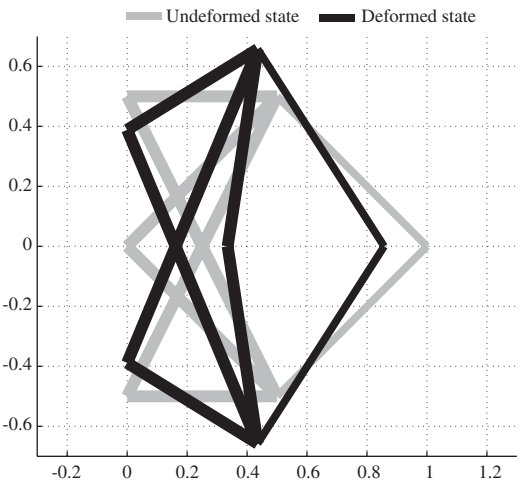


Figure 15. A solution to **RELAX4**, $D_{\text{out}} = 0.1427$.

at the optimum. The optimization process required 34 iterations to converge; this took about 192.6 s (Pentium4-1.7 GHz CPU). As Figure 15 shows, the relaxed formulation did not succeed in providing a (near) integer result; we still in this result have intermediate values in the design variables ρ_e (the two right-end bars of the mechanism have $\rho_e = 0.6457$). Moreover, none of the lower bound constraints ($\rho_{\min} \leqslant \rho_e$) for the design variables is active, and this holds for the DOF constraint as well.

Although we have tried smaller lower bounds such as $\rho_{\min} = 0.0001$ and larger penalization indices such as $P = 4$, the situation apparently cannot be cured without further revisions of the optimization procedure.

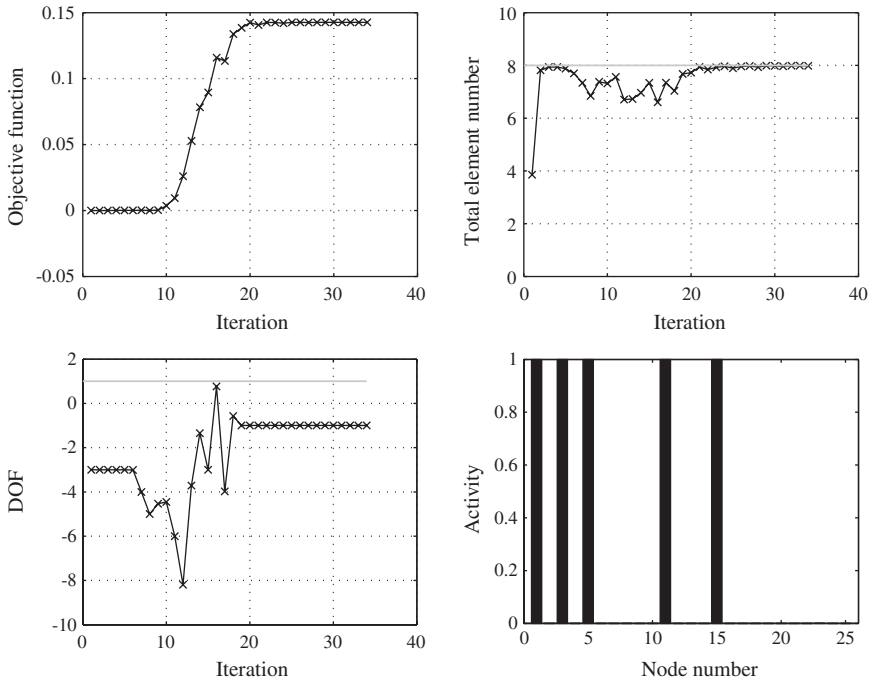


Figure 16. Optimization histories and active nodes at the optimum for the inverter mechanism problem.

7.2. A continuation method

One possibility for revising the optimization scheme is to introduce an additional constraint that will promote the separation of the design variables between the two extrema, i.e. $\rho_{\min}$ or 1. For that purpose we define a separation function as

$$C_s(\rho) := \sum_{e=1}^{M} (\rho_e - \rho_{\min})(1 - \rho_e) \tag{27}$$

This function is positive in the domain where $0 < \rho_{\min} < \rho_e < 1$ $(e = 1, \ldots, M)$, and gives zero when all the design variables attain either $\rho_{\min}$ or 1.

A possible reformulation of our design problem is then:

RELAX5:

$$\begin{aligned} \max_{\rho} &: D_{\text{out}} \\ \text{s.t.} &: \sum_{e=1}^{M} \rho_e \leqslant M^* + \alpha \\ &: \sum_{e=1}^{M} \rho_e \geqslant M^* + \alpha - C_s^* \end{aligned}$$

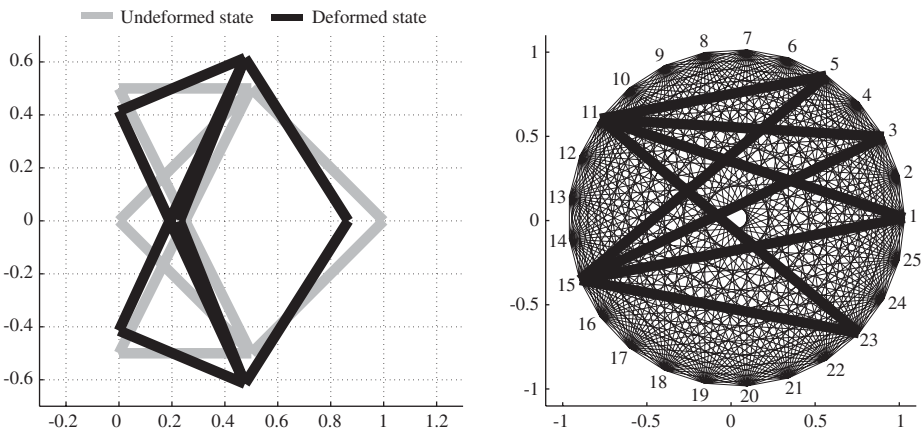


Figure 17. A solution to **RELAX5**, $D_{\text{out}} = 0.1338$ and the connectivity circle.

$$: 0 < \rho_{\min} \leqslant \rho_e \leqslant 1$$

$$: \text{symmetric}$$

$$: \tilde{f} \leqslant 1$$

$$: C_s \leqslant C_s^*$$

where C_s^* is the upper bound of the separation function. The second constraint makes it possible to force the total number of elements towards M^*. In implementations, we start the iterations with a certain positive value for C_s^* and update the value according to the progress of the refinement of the solution. In other words, we apply a *continuation method* that has an outer loop that wraps the inner loop **RELAX5** with a certain value for C_s^*. We have adopted the following update scheme for ${}^iC_s^*$ in the ith outer iteration:

$${}^iC_s^* = 0.9 \times {}^{i-1}C_s^* \quad (i \geqslant 1) \tag{28}$$

where ${}^0C_s^*$ is given from the solution ρ^* to **RELAX4** as ${}^0C_s^* = C_s(\rho^*)$.

We have solved the inverter problem with MMA and obtained an extremum solution as shown in Figure 17. Figure 18 shows the optimization history. The optimization process required 275 iterations for convergence (25 min with a Pentium4-1.7 GHz CPU). As expected, the objective function value of the discrete-valued solution is less than that of the solution to **RELAX4** (which is less constrained). The thin grey lines in the upper right graph in Figure 18 indicate the upper and lower bounds of the total element number; the lower bound increases after each update of ${}^iC_s^*$. The thin grey line in the lower right graph in Figure 18 indicates the upper bound of the separation function ${}^iC_s^*$, which decreases as it is updated in the outer loop.

The crucial difference between the two inverter mechanisms in Figures 15 and 17 is the position of the input port after deformation. The input port of Figure 15 goes deeper into the design area than that of Figure 17 because of the effect of the bars at intermediate areas. This explains why the optimization finds a solution to **RELAX4** that contains intermediate values and why one needs the additional separation constraint of **RELAX5**.

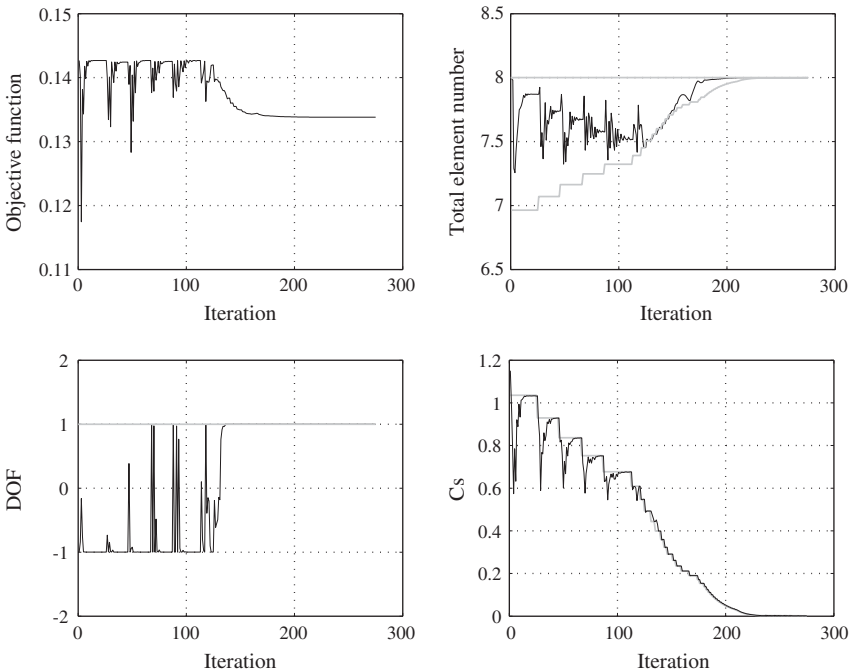


Figure 18. Optimization histories of the continuation method.

In order to be sure that the functionality of the mechanism does not change, we have analysed again the mechanism without the thin bars that are at the lower bound of the density and we have obtained the same deformed configuration with the output displacement: $D_{\text{out}} = 0.1338$.

8. CONCLUSIONS

We have experimented with four different kinds of relaxation approaches to the basic test example in terms of applying the DOF constraint. This encompasses:

RELAX1: No DOF constraint
RELAX2: Mean compliance constraint
RELAX3: Node number constraint
RELAX4: Relaxed DOF constraint

From these examples it is apparent that a DOF constraint is necessary in order to obtain good results.

Specifically for **RELAX4** we note that:

- A method for detecting independent bars in the ground structure has been devised.
- A relaxed version of degree of freedom equation has been implemented.
- The resultant mechanism in Example IV from **RELAX4** is the same as the global optimum solution obtained by an graph-theoretical enumeration approach [14].

We have also found that **RELAX4** may have a difficulty in finding discrete valued solutions in some cases, here illustrated via an inverter design problem. In order to make sure that a (near) integer solution is found, we have proposed the continuation method of **RELAX5** and obtained a solution with the desired properties.

Even though the proposed method seems promising with the presented examples, there exists a critical limitation due to the non-convexity of the problem. It is not clear how much we can use the same strategy for larger problems. One can expect that it will be very difficult to even find feasible binary solutions with more bars in use. So we will need to combine some global optimization techniques for generalization and further applications; thus work is in progress [23].

APPENDIX A: NODAL POSITIONS ON THE CONNECTIVITY CIRCLE

In general, the connectivity circle as illustrated in Figure 11 enables one to detect independent bars in a truss by a standard eigenvalue analysis based on the linear stiffness matrix of the truss. It is, however, important to note that one should be careful about the placement of the nodal points on the connectivity circle for some special cases where symmetry may play a role. For illustration, consider the two trusses shown in Figure A1, in which the numbering of nodes is somewhat artificial.

If we construct their connectivity circles based on a *regular* placing of the nodes on the circle (i.e. $\beta = 0$) like Figure A2 and apply standard eigenvalue analyses to their linear stiffness matrices, we obtain the results shown in Figure A3. Both the connectivity circles have 8 non-zero eigenvalues, but truss (a) has 9 independent bars while truss (b) has 8 independent bars (truss (a) forms a stiff structure, while truss (b) is a mechanism). Here the regular nodal lay-out for truss (a) has created more than two trapezoids (in general this could include twisted trapezoids) and one fails to detect the independent bars correctly because the linear stiffness matrix cannot express any higher order effects of stiffness.

Thus Equations (12) and (13) are designed to create asymmetric nodal positions with irregular intervals between the nodes on the connectivity circle when $\beta > 0$. This ensures that none of the connection between nodal pairs of the truss becomes parallel in the connectivity circle. Figure A4 shows the connectivity circles for the example above but with the irregular intervals ($\beta = 1$), and the prediction of the number of independent bars is now rectified, cf., Table A1.

An alternative to this is to reduce the number of trapezoids in the connectivity circle by swapping two or more nodal positions. This is a little bit more complicated because we first need to find a way to reduce the number of trapezoids in the connectivity circle. The number

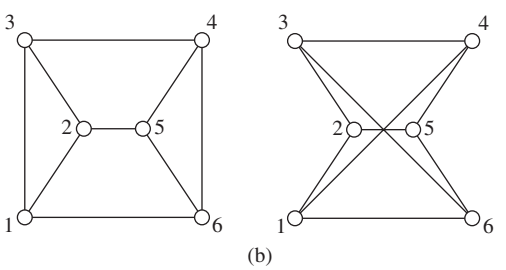


Figure A1. Two plane trusses which have 6 nodes and 9 edges.

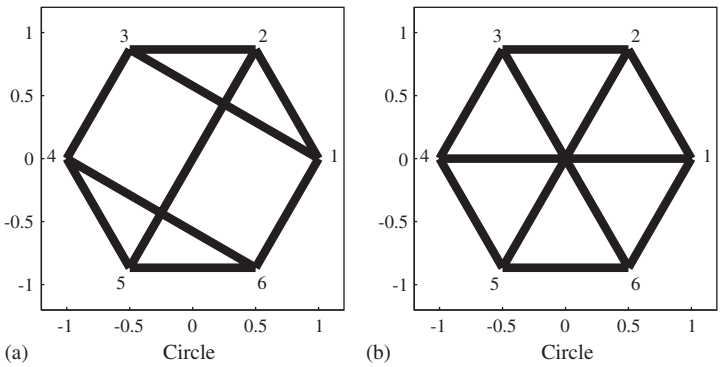


Figure A2. Detection of independent bars by eigenvalue analysis.

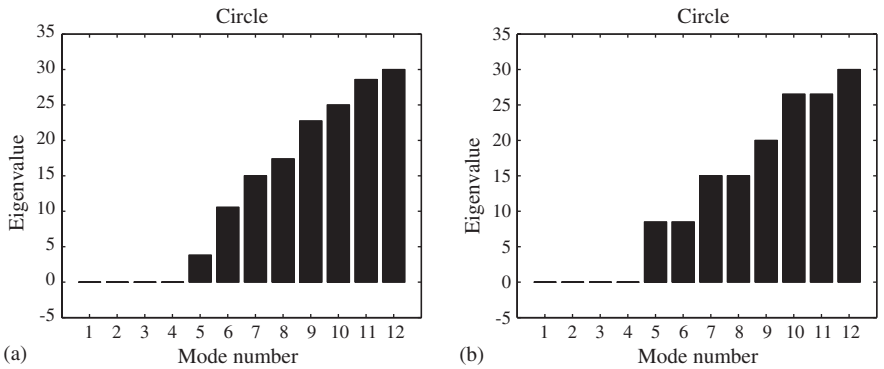


Figure A3. Detection of independent bars by eigenvalue analysis.

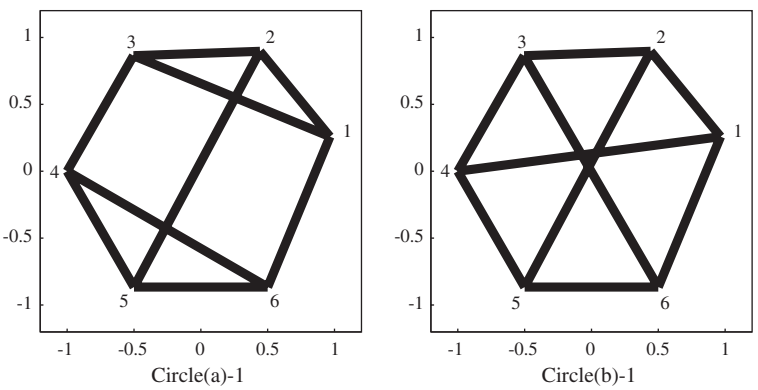


Figure A4. Connectivity circles with irregular intervals.

Table A1. Effect of nodal relocations on eigenvalues.

Mode	Circle(a)	Circle(a)-1	Circle(a)-2	Circle(b)	Circle(b)-1	Circle(b)-2
1	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
2	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
3	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
4	0.00000	0.00159	0.97733	0.00000	0.00000	0.00000
5	3.80866	3.57353	3.03620	8.48612	8.17001	6.61961
6	10.56275	10.51132	8.31595	8.48612	8.69149	8.08050
7	15.00000	14.56579	12.58322	15.00000	14.67180	11.76246
8	17.39342	16.60888	15.92641	15.00000	15.35712	12.71789
9	22.73835	23.45285	22.88706	20.00000	19.98904	17.03465
10	25.00000	25.82366	26.59729	26.51388	24.81688	23.75651
11	28.59083	29.35334	26.64641	26.51388	27.70841	26.40928
12	30.00000	32.10739	29.21816	30.00000	32.54784	29.80711

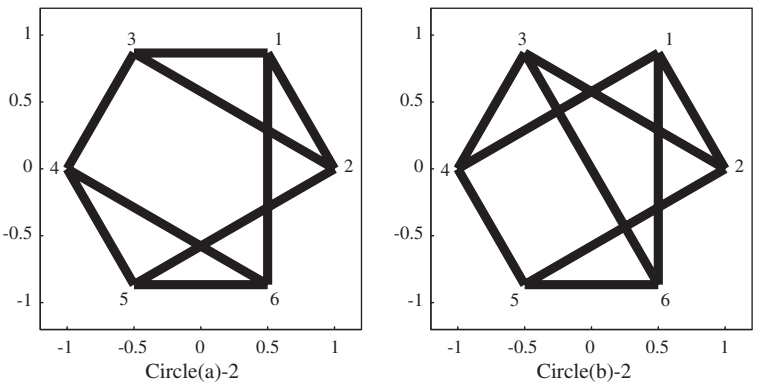


Figure A5. Connectivity circles with swap between node 1 and 2.

of trapezoids in Circle (a) can be reduced by, for instance, swapping the first and second nodes like Circle (a)-2 in Figure A5. However, for Circle (b) we have no possibility to reduce the number trapezoids due to the symmetry. If we swap the first and second nodes of Circle (b), we will have Circle (b)-2 in Figure A5, which still has three trapezoids.

Table A1 shows the effect of the two methods on the eigenvalues of the connectivity circles. We note that the first idea is easy to implement but the effect is quite modest in terms of the perturbation of the eigenvalues, while the second idea is more difficult but we can expect a greater effect on the lowest non-zero eigenvalue.

ACKNOWLEDGEMENTS

The authors would like to thank K. Svanberg for permission to use his software implementation of the MMA algorithm. The authors also gratefully acknowledge many fruitful conversations with C. Thomassen regarding graph theoretical aspects of mechanism design. Finally, the authors appreciate a good discussion with K. Moritzen about eigenvalue problems.

REFERENCES

1. Sigmund O. On the design of compliant mechanisms using topology optimization. *Mechanics of Structures and Machines* 1997; **25**:493–524.
2. Pedersen CBW, Buhl T, Sigmund O. Topology synthesis of large-displacement compliant mechanisms. *International Journal for Numerical Methods in Engineering* 2001; **50**:2683–2705.
3. Bruns TE, Tortorelli DA. Topology optimization of non-linear elastic structures and compliant mechanisms. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**(26–27):3443–3459.
4. Ananthasuresh GK, Kota S, Kikuchi N. Strategies for systematic synthesis of compliant MEMS. *Dynamic Systems and Control* 1994; **2**:677–686.
5. Sigmund O. Some inverse problems in topology design of materials and mechanisms. In *IUTAM Symposium on Optimization of Mechanism Systems*, Bestle D, Schielen W (eds). Kluwer: Dordrecht, 1996; 277–284.
6. Frecker MI, Ananthasuresh GK, Nishiwaki S, Kikuchi N, Kota S. Topological synthesis of compliant mechanisms using multi-criteria optimization. *Journal of Mechanical Design* 1997; **119**(2):238–245.
7. Erdman AG, Sandor GN. *Mechanism Design—Analysis and Synthesis* (2nd edn), vol. 1. Prentice-Hall: New Jersey, 1991; 17–25.
8. Bendsøe MP, Sigmund O. *Topology Optimization—Theory, Methods, and Applications.* Springer: Berlin, Heidelberg, 2003.
9. Calladine CR. Buckminster Fuller's tensegrity structures and Clerk Maxwell's rules for the construction of stiff frame. *International Journal of Solids and Structures* 1978; **14**:161–172.
10. Pellegrino S, Calladine CR. Matrix analysis of statically and kinematically indeterminate frameworks. *International Journal of Solids and Structures* 1986; **22**(4):409–428.
11. Fowler PW, Guest SD. A symmetry extension of Maxwell's rule for rigidity of frames. *International Journal of Solids and Structures* 2000; **37**(12):1793–1804.
12. Wolsey LA. *Integer Programming.* Wiley: New York, 1998; 7–8.
13. Svanberg K. The method of moving asymptotes—a new method for structural optimization. *International Journal for Numerical Methods in Engineering* 1987; **24**:359–373.
14. Kawamoto A, Bendsøe MP, Sigmund O. Planar articulated mechanism design by graph theoretical enumeration. *Structural and Multidisciplinary Optimization* 2004; **27**(4):295–299.
15. Harary F. *Graph Theory.* Addison-Wesley: Reading, MA, 1969; 14–15.
16. Kaveh A. *Structural Mechanics: Graph and Matrix Methods.* Research Studies Press Ltd: Hertfordshire, 1991; 13–21.
17. Seyranian AP, Lund E, Olhoff N. Multiple eigenvalues in structural optimization problems. *Structural Optimization* 1994; **8**:207–227.
18. Clarke FH. *Optimization and Nonsmooth Analysis.* Wiley: New York, 1983; 94–95.
19. Lewis AS, Overton ML. Eigenvalue optimization. *Acta Numerica* 1996; **5**:149–190.
20. Rodrigues HC, Guedes JM, Bendsøe MP. Necessary conditions for optimal design of structures with a non-smooth eigenvalue based criterion. *Structural Optimization* 1995; **9**:52–56.
21. Achtziger W. Methods of non-smooth optimization: algorithms and applications. In *Advanced Topics in Structural Optimization,* Bendsøe MP, Pedersen P (eds). *DCAMM-Report no. S81*, Technical University of Denmark: Lyngby, Denmark, 1998; 18–23.
22. Golub GH, Van Loan CH. *Matrix Computations* (fourth printing). John Hopkins University Press: Baltimore, MD, 1985; 190–199.
23. Kawamoto A, Stolpe M. Design of articulated mechanisms using branch and bound. Submitted.
24. Trefethen LN, Bau III D. *Numerical Linear Algebra.* SIAM: Philadelphia, 1997; 32–37.
25. Ma ZD, Kikuchi N, Cheng HC. Topological design for vibrating structures. *Computer Methods in Applied Mechanics and Engineering* 1995; **121**:259–280.

[Paper 3]

Design of planar articulated mechanisms using branch and bound

Mathias Stolpe[‡] Atsushi Kawamoto[§]

Abstract

This paper considers an optimization model and a solution method for the design of two-dimensional mechanical mechanisms. The mechanism design problem is modeled as a nonconvex mixed integer program which allows the optimal topology and geometry of the mechanism to be determined simultaneously. The underlying mechanical analysis model is based on a truss representation allowing for large displacements. For mechanisms undergoing large displacements elastic stability is of major concern. We derive conditions, modeled by nonlinear matrix inequalities, which guarantee that a stable equilibrium is found and that buckling is prevented. The feasible set of the design problem is described by nonlinear differentiable and non-differentiable constraints as well as nonlinear matrix inequalities.

To solve the mechanism design problem a branch and bound method based on convex relaxations is developed. To guarantee convergence of the method, two different types of convex relaxations are derived. The relaxations are strengthened by adding valid inequalities to the feasible set and by solving bound contraction sub-problems. Encouraging computational results indicate that the branch and bound method can reliably solve mechanism design problems of realistic size to global optimality.

***Key Words*:** mechanism design – branch and bound

1 Introduction

In this paper we consider the two closely associated problems of analysis and synthesis (design) of articulated mechanisms undergoing large deformations, where we by mechanism mean a mechanical device that can transfer motion and force. The approach chosen here is to solve the analysis and design problems simultaneously in the sense that the necessary conditions derived from a variational form of the analysis problem are explicitly stated in the design problem. The combined mechanism design and analysis problem is modeled as a nonconvex mixed integer program. The design variables in this problem are binary variables that describe the connectivity (topology) of the mechanisms and continuous variables that describe dimensions and positions (geometry) of the different parts of the mechanism. The design problem also contains continuous state variables which are associated with the mechanical analysis part and which describe the behavior of the mechanism under loading. The feasible set of the mechanism design problem is described by nonlinear differentiable and non-differentiable constraints as well as nonlinear matrix inequalities. To solve the mechanism design problem a branch and bound method based on convex relaxations is developed. In order to state a convergent branch and bound method, two different types of convex relaxations are derived. The first is a quadratically constrained linear program that is solved by a general purpose sequential quadratic programming package. The second relaxation is a linear semidefinite program that is solved by an interior-point method. To strengthen the relaxations we derive valid inequalities which are added to the feasible set of the relaxation. Furthermore, the quality of the relaxations are improved by solving bound contraction sub-problems at each node of the branch and bound tree. To this end, we make extensive use of modern solvers for linear, quadratic, and linear semidefinite programs.

During the developments of the mechanical model and the early formulations of the optimization problem it became apparent that the branch and bound method would take full advantage of any (and all) shortcomings in the mechanical model as well as in the optimization model such that technically unrealistic or

[‡]Mathias Stolpe: Department of Mathematics, Technical University of Denmark, Matematiktorvet Building 303, DK-2800 Kgs. Lyngby, Denmark. e-mail: M.Stolpe@mat.dtu.dk

[§]Atsushi Kawamoto: Department of Mathematics, Technical University of Denmark, Matematiktorvet Building 303, DK-2800 Kgs. Lyngby, Denmark. e-mail: A.Kawamoto@mat.dtu.dk

Fig. 1 Scissor lift.

faulty solutions would be obtained. Therefore, a large portion of this article is devoted to the development of a mechanical analysis model and motivations for the numerous technological constraints which constitute the major part of the feasible set of the optimization problem.

The choice of using a deterministic global optimization method is motivated by observations that these combinatorial mechanism design problems have many good feasible solutions that are difficult to find using heuristic methods. Furthermore, by using the branch and bound method it is possible to reliably solve planar mechanism design problems of realistic size to global optimality in reasonable time. Before proceeding to the presentation of the suggested approach we introduce some terminology commonly used within mechanism design and show one example of the practical use of articulated mechanisms.

1.1 Introduction to mechanism design

A *mechanism* is a mechanical device that can transfer motion and/or force from a source to an output [22]. A mechanism consists of links (or bars) which are connected by joints. An *articulated mechanism* is a mechanism which gains all of its mobility from its joints. This is in contrast to a *compliant mechanism* which is another type of mechanism that gains some or all of its mobility from the flexibility of its components. A mechanism is called *planar* if it can be considered as two-dimensional from a mechanical analysis point of view and its motion is planar (all bars move in parallel planes).

Articulated mechanisms are used in a great variety of engineering applications. For instance, they are used as part of automotive suspension systems, as current collectors of electrical locomotives, and as part of scissor lifts, see Figure 1. The mechanism shown in Figure 1 can be viewed as essentially planar even though it consists of two planar mechanisms which are joined to form a spatial mechanism. In this example, however, the motion is planar and any out-of-plane motion is highly undesirable.

One important property of articulated mechanisms is the number of *mechanical degrees of freedom* which is defined as the number of independent inputs required to determine the position of all links of the mechanism with respect to a global coordinate system [22]. An input is here defined as an external load applied at the joints of the mechanism. In this paper we focus on design of mechanisms with exactly one mechanical degree of freedom. This means that the mechanism becomes rigid if one of the revolute joints is fixed. A mechanism that has exactly one mechanical degree of freedom is called a *constrained mechanism*, see [22].

Normally, articulated mechanisms are modeled by rigid bars and revolute joints, see e.g. [37] and [29]. The design process is commonly divided into three stages, see e.g. [22] and [21]. Firstly, a design objective is determined for the mechanism to be designed. The objective may, for example, be maximization of output displacement or generation of a prescribed output path for a given input. Secondly, mechanism properties

such as the number of bars and joints, mechanical degrees of freedom, and connectivity are determined. This stage is called *type synthesis* and mainly focuses on topological questions. An elaborate type synthesis technique based on number synthesis is developed in [47, 48]. Finally, the exact dimensions of all the mechanical components are determined using optimization. This stage is called *dimensional synthesis* and is mostly related to geometrical questions. Reviews on dimensional synthesis techniques are found in [37] and [29]. Although type and dimensional syntheses are normally separated in the engineering community, they are actually so closely connected that they should be treated at the same time.

In this paper we will formulate an optimization problem to facilitate *simultaneous topology and geometry design*, i.e. simultaneous type and dimensional synthesis of articulated mechanisms. We then proceed to develop a convergent branch and bound method capable of solving the proposed problem to global optimality. The optimization problem will be a *nonconvex mixed integer program* with a potentially large number of integer and continuous variables. The design problem which will be formulated and solved in this paper can be stated as follows:

"Design an articulated mechanism with exactly one mechanical degree of freedom that given an external input force maximizes the displacement of a certain joint in a desired direction. Furthermore, the mechanism should be able to function repeatedly, with certainty, and without using the flexibility of its members."

Instead of using rigid bars and revolute joints in the analysis of the mechanism we propose to use a truss representation. By *truss* we mean an assemblage of straight elastic bars which are connected at frictionless pin-joints (nodes). It is assumed that the external loadings are applied only at the truss joints and therefore that only axial forces exist within the truss. This choice of model is motivated not only for its simplicity but also for the possibility it gives to avoid the numerical problems associated with the use of rigid bars in the dimensional synthesis (the so called non-assembly problem, see e.g. [37] and [29]).

The use of an elastic truss model is an approximation of the rigid bars and revolute joints. Therefore it is important to choose the modulus of elasticity sufficiently large such that the articulated mechanisms cannot rely on the elasticity of the bars to function. In other words, if the modulus of elasticity is large enough then the functionality of articulated mechanisms will not change if all the truss elements are replaced with rigid bars.

To formulate the design problem we adopt a commonly used technique from topology optimization of stiff structures called the *ground-structure approach* which was introduced in the 1960s in [20]. This technique, applied to so-called *minimum compliance problems*, generated a lot of interest in the mathematical programming community in the early and mid 1990s (see e.g. [11]) since these problems are highly structured. Minimum compliance problems can be modeled as large-scale linear semidefinite programs [49], [12], [13], second-order cone programs [10], convex quadratically constrained quadratic programs [1], [31], or even non-smooth problems [7]. In the ground-structure approach, a large number of potential nodes and an even larger number of potential bars are distributed over a two-dimensional domain. Typically, the nodes are initially evenly distributed over a rectangular design domain and connections are allowed between each pair of nodes. In the following, the number of nodes is denoted by N and the number of potential bars is denoted by $n = N(N-1)/2$. To facilitate simultaneous topology and geometry optimization we associate in the following with every potential bar a binary variable indicating presence or absence of the bar in the final design (topology variables) and with every node we introduce continuous variables representing the initial undeformed position of the node (geometry variables) as well as binary variables indicating absence or presence of the node in the final design.

An introduction to applications and methods for topology optimization of discrete and continuum structures based on the ground structure approach can be found in the monographs [14] and [16] and the review articles [41] and [23].

1.2 A benchmark example

To illustrate the ground-structure approach a benchmark example is shown in Figure 2. The figure shows a truss ground-structure with possible nodes and connections, the boundary conditions, and the external loads. Overlapping bars are not shown in the figure. In all, the ground-structure consists of $N = 12$ nodes and $n = 66$ potential bars. The design domain is defined as the gray square area with height and width equal to one. The single external load, denoted by p is applied at node 1 and 3 simultaneously. The nodes at which the external load is applied are called the input ports. The node at which the desired output should

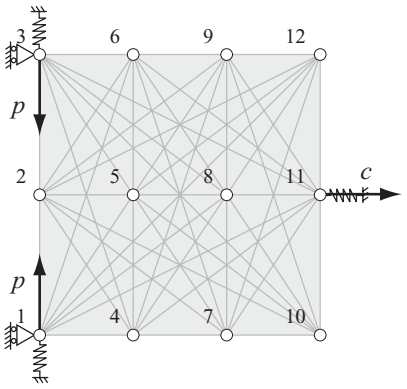


Fig. 2 Ground-structure for the benchmark example.

take place is named the output port. In this example it corresponds to node 11. The direction of the desired output is denoted by c. The springs and forces at the input ports model strain based actuators (e.g. shape memory alloys or piezoelectric devices) and the output spring simulates resistance of movement.

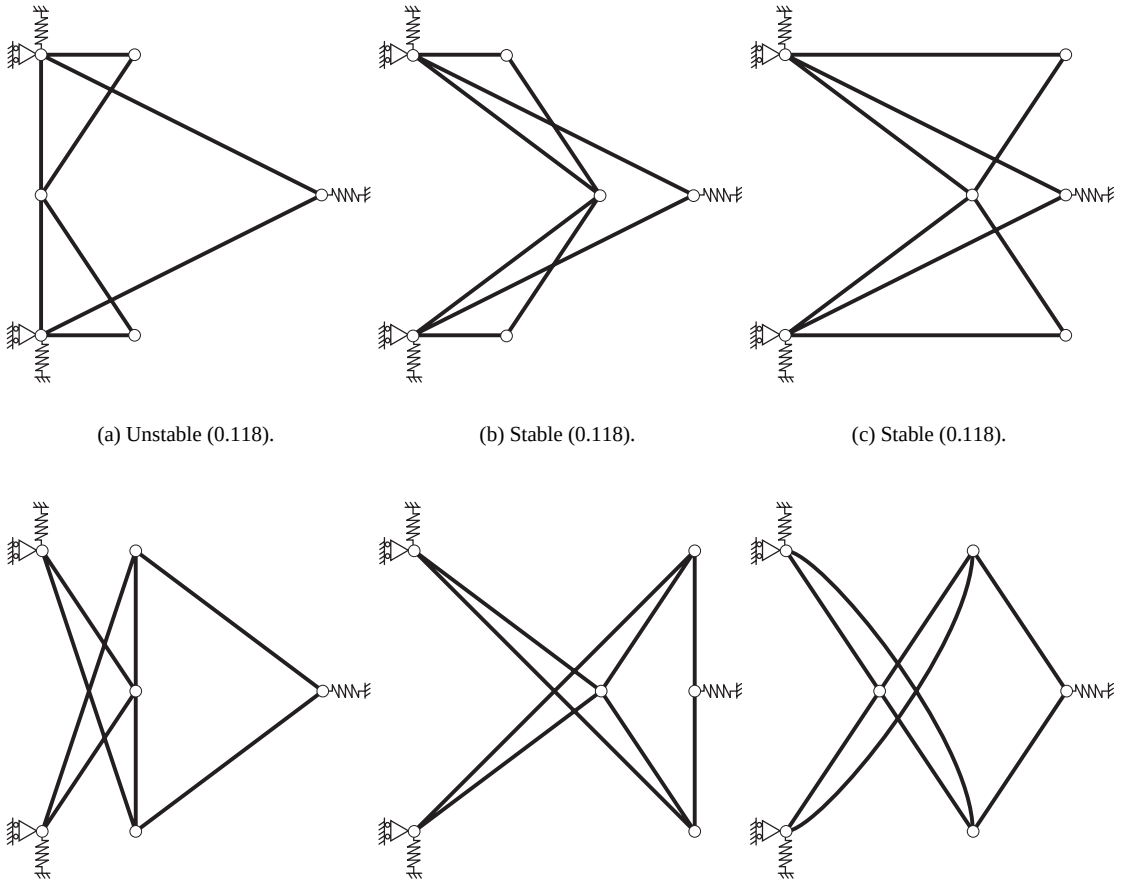
For the benchmark examples shown in Figure 2 it is possible to enumerate all symmetric topologies consisting of eight bars with exactly one degree of freedom, see [33]. For this example there exist 144 different topologies which can produce output displacement in the desired direction. However, as the number of potential bars, active bars, and nodes increase the number of potential topologies grows rapidly and enumeration becomes intractable.

To illustrate the richness of the feasible set of this example, six different topologies with six nodes, eight bars, and one mechanical degree of freedom, which all can generate output in the desired direction, are depicted using kinematic diagrams (schematic drawings of mechanisms) in Figure 3. In these figures it is assumed that the initial nodal positions remain fixed, and hence only the topology is changed. The curved lines in Figure 3(f) represent overlapping bars.

Even though six different topologies with completely different behavior under loading are depicted in Figure 3, the connectivity of the mechanisms shown in Figure 3(a) - 3(c) and Figure 3(d) - 3(f), respectively, are very similar. In fact, if the nodal positions of the three active nodes where no springs are attached are allowed to change, the topologies in Figure 3(a) - 3(c) (and 3(d) - 3(f) respectively) can be made exactly the same! This observation suggests that the positions of the nodes are of the utmost importance and should also be considered as a part of the problem formulation.

A very important concept in mechanism design is that of elastic stability. In our benchmark example instability is particularly noticeable as buckling of sub-structures. This is a global phenomenon in contrast to the local Euler buckling of long thin bars. To illustrate buckling of substructures we consider the mechanism shown in Figure 3(a). Consider the two left-most vertical bars in Figure 3(a). If a small force is applied as indicated in the ground-structure, these two bars will be compressed. In this situation their common node, the node labeled 2 in Figure 2, will initially not move. As the applied force is increased, the node will suddenly move to the right or left depending on imperfections in the design. Unfortunately, this move will be large and instant and the direction is unpredictable. Therefore, this situation is undesirable. Accompanied with each mechanism in Figure 3 is a statement, for now based on engineering experience, if the mechanism is stable or not with respect to buckling of sub-structures. In two of the examples shown in Figure 3 the mechanisms are unstable since buckling must occur before movement of the output port in the desired direction is possible. Also stated below each mechanism is the displacement of the potential movement of the output port in the desired direction.

There exist several reasons for choosing complete deterministic methods to solve the considered mechanism design problems. First of all, the considered topology optimization problem is of combinatorial nature with potentially many good feasible designs (topologies) as indicated in Figure 3. Secondly, the outlined approach gives the possibility to include various types of nonconvex restrictions to the feasible set, such as global stability and local stress constraints. These constraints are important from a mechanical point of view but they have previously been considered too difficult to include due to the nonconvexity of the constraints or due to the drastically increased size of the optimization problem. Thirdly, the choice of method is



(a) Unstable (0.118). (b) Stable (0.118). (c) Stable (0.118).

(d) Stable (0.791). (e) Unstable (0.820). (f) Stable (0.778).

Fig. 3 Some potentially feasible topologies for the benchmark example.

motivated by computational experience with the penalization methods traditionally used in topology optimization with binary variables, see [32]. Finally, computational experience with branch and bound methods for the design of stiff truss structures indicate that it is possible to solve problems of the size which will be required, see [17] and [44].

1.3 Outline

This paper is organized as follows. In Section 2, the mechanical model together with the equilibrium equations, the stability analysis, and degree of freedom analysis are presented. In Section 3, a basic topology optimization problem together with two convex relaxations and some specialized valid inequalities are presented. In Section 4, the topology design problem is extended such that simultaneous topology and geometry design is possible. The branch and bound method for the proposed problems is presented in Section 5. Section 6 contains solutions to several numerical examples. Section 7, finally, contains conclusions and possible directions for future research within this area.

2 The mathematical-mechanical model

In this section we present the mathematical format of the models for the analysis of large-displacement trusses which are used throughout the paper. First we introduce a relevant strain measure and present the mechanical analysis problem for elastic trusses undergoing large displacements. The analysis problem

consists of determining the behavior, quantified by displacements, strains, and stresses of the truss under loading. This problem will be defined by two necessary conditions on the displacements stating that *(i)* the external and internal loads should be in balance for a given design, and *(ii)* that the deformed mechanism should be stable. Then we present stability conditions for the undeformed mechanism, designed to avoid elastic instabilities which may cause the mechanism to malfunction. Finally, we present a procedure to correctly compute the mechanical degrees of freedom of a truss with overlapping bars.

2.1 Notation

Given a real symmetric matrix A, the expression $A \succeq 0$ means that A is positive semidefinite. If $a \in \mathbb{R}$ then the notation a^+ denotes $a^+ = \max\{a, 0\}$. Similarly, a^- is defined as $a^- = \min\{a, 0\}$. If a is a real vector or a real matrix then a^+ and a^- are defined component-wise. The vector e and the matrix I denote the vector of all ones and the identity matrix whose dimensions are defined by the context of their use. $\mathbb{B}^n$ denotes the set of n-dimensional binary vectors.

2.2 Equilibrium equations

The topology variables in the model are the bar areas $a_j \in \{0, 1\}$, $j = 1, \ldots, n$ where n is the number of potential bars in the ground-structure. The geometry variables in the model are the nodal positions $\chi = (x_1, y_1, \ldots, x_N, y_N)^\top \in \mathbb{R}^{2N}$ where N is the number of nodes (frictionless joints) in the ground-structure and (x_k, y_k) is the position of the k-th node. Further, the state variables in the model are collected in the vector $u \in \mathbb{R}^d$ containing the nodal displacements of the design under the external load. The dimension of the displacement vector d is the number of degrees of freedom (in the finite element method sense) of the structure which for a two-dimensional truss $d = 2N - d_f$ where d_f is the number of prescribed fixed displacements (assumed to be zero). The total potential energy $\Pi(a, u, \chi)$ of a truss subjected to the external static nodal force vector $p \in \mathbb{R}^d$ is given by

$$\Pi(a, u, \chi) = \frac{1}{2} u^\top K_0 u + \frac{1}{2} \sum_{j=1}^{n} E_j l_j(\chi) a_j \epsilon_j(u, \chi)^2 - p^\top u,$$

where $K_0 \in \mathbb{R}^{d \times d}$ is the constant spring stiffness matrix due to linear springs attached to some of the nodes in the truss, $E_j > 0$ is the Young's modulus in the j-th bar, $l_j(\chi)$ is the undeformed length, and $\epsilon_j(u, \chi)$ is the strain (defined below) of the j-th bar, respectively.

For a given topology and geometry the displacement vector $u \in \mathbb{R}^d$ should be a minimizer of the potential energy. In order to find a stable equilibrium, we need to find a local minimizer or at least a stationary point of $\Pi(a, u, \chi)$ such that the Hessian of the potential energy with respect to u is positive semidefinite. The problem of finding such a u is called the analysis problem. The first order optimality conditions for this problem, in the form of *equilibrium equations*, are given by

$$\nabla_u \Pi(a, u, \chi) = 0, \tag{1}$$

while the second order conditions, in the following called *stability conditions*, are given by

$$\nabla^2_{uu} \Pi(a, u, \chi) \succeq 0. \tag{2}$$

Note that $\Pi(a, u, \chi)$ in general is a nonconvex functional which may have no such stationary points. Alternatively, there may exist several displacement vectors u satisfying both (1) and (2) for a given design (a, χ).

Throughout the present paper we make the following assumptions on the mechanical model.

1. Large displacements are allowed but the strains are assumed to be small.
2. The strain is given by the Green-Lagrange strain measure.
3. The material is linearly elastic and the material properties remain constant.

The Green-Lagrange strain measure $\epsilon_j(u,\chi)$ is defined as

$$\epsilon_j(u,\chi) = \frac{\hat{l}_j^2(u,\chi) - l_j^2(\chi)}{2l_j^2(\chi)},$$

where $\hat{l}_j(u,\chi)$ is the deformed length of the j-th bar, see e.g. [19]. In order to rewrite this strain measure, we first introduce a symmetric positive semidefinite matrix $B_j \in \mathbb{R}^{2N\times 2N}$ which is defined such that the undeformed length of the j-th bar is given by $l_j(\chi) = (\chi^\top B_j \chi)^{1/2}$. The non-zero part of B_j is given by, in local coordinates corresponding to the position of the end nodes of the considered bar,

$$B_j = \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix}.$$

The matrix $\tilde{B}_j$ is the symmetric positive semidefinite $d \times d$ matrix that remains after removing the rows and columns corresponding to the fixed degrees of freedom from B_j. The matrix $\hat{B}_j$ is the $d \times 2N$ matrix which remains after removing the rows corresponding to the fixed degrees of freedom from B_j.

The Green-Lagrange strain in the j-th bar may now be reformulated as

$$\epsilon_j(u,\chi) = \frac{1}{l_j^2(\chi)} b_j^\top(\chi) u + \frac{1}{2l_j^2(\chi)} u^\top \tilde{B}_j u, \tag{3}$$

where $b_j(\chi) = \hat{B}_j \chi \in \mathbb{R}^d$. Then the vector $b_j(\chi)/l_j(\chi)$ contains the direction cosines of the j-th bar, $b_j^\top(\chi)u/l_j(\chi)$ is the linearized elongation of the bar, and $b_j^\top(\chi)u/l_j^2(\chi)$ is the linear (small deformation) strain. The potential energy can, using the Green-Lagrange strain (3), be written as

$$\Pi(a,u,\chi) = \frac{1}{2} u^\top \left(K(a,\chi) + G(a,u,\chi) + H(a,u,\chi)\right) u - p^\top u$$

where the positive semidefinite matrix $K(a,\chi) \in \mathbb{R}^{d\times d}$ is given by

$$K(a,\chi) = K_0 + \sum_{j=1}^{n} \frac{a_j E_j}{l_j^3(\chi)} b_j(\chi) b_j^\top(\chi)$$

and the matrix $G(a,u,\chi) \in \mathbb{R}^{d\times d}$ is given by

$$G(a,u,\chi) = \sum_{j=1}^{n} \frac{a_j E_j}{l_j^3(\chi)} b_j^\top(\chi) u \tilde{B}_j = \sum_{j=1}^{n} \frac{s_j(\chi)}{l_j(\chi)} \tilde{B}_j.$$

We denote the small deformation axial forces by $s_j(\chi) = a_j E_j b_j^\top(\chi) u / l_j^2(\chi)$. The higher order terms are collected in the positive semidefinite matrix $H(a,u,\chi) \in \mathbb{R}^{d\times d}$ given by

$$H(a,u,\chi) = \frac{1}{4} \sum_{j=1}^{n} \frac{a_j E_j}{l_j^3(\chi)} (\tilde{B}_j u)(\tilde{B}_j u)^\top.$$

The matrices $K(a,\chi)$ and $G(a,u,\chi)$ are normally called the linear *stiffness matrix* and the *geometry matrix* of the truss respectively.

2.3 Stability analysis

Of major importance in mechanism design, as well as in traditional truss topology design in linear elasticity, is the possibility of obtaining an optimal design which is unstable. For structures, such as bridges, insufficient elastic stability may cause the structure to fail under loading even if the stresses in the material are not significant. For mechanisms, insufficient elastic stability may hinder the mechanism to function properly. For example, consider the mechanism in Figure 3(e). This mechanism cannot move unless buckling occurs.

For this mechanism, the direction of movement is also uncertain and determined by imperfections in the design.

Since we want the optimal design to be able to perform its task repeatedly and with certainty, the elastic stability of the mechanism must be assured. To meet this requirement, we include constraints on the global stability of the mechanism in the problem formulation. The derivation of the stability conditions in this section are described in [8] and [34].

Intuitively, a structure or mechanism is stable for small external loads, but as the force increases and reaches a certain limit, the mechanism may become unstable. The problem of truss stability is to find a critical load for which the structure becomes unstable. A commonly used model for the stability analysis of trusses is the *linear buckling model*, see e.g. [6]. Using the linear buckling model, the condition on the truss stability can be stated as the nonlinear matrix inequality

$$K(a,\chi) + G(a,u_0,\chi) \succeq 0, \tag{4}$$

where u_0 is a solution to the small-deformation equilibrium equations

$$K(a,\chi)u_0 = p. \tag{5}$$

Since the stability conditions (4) and (5) are intended for stiff structures rather than mechanisms, the displacement vector u_0 that solves (5) may violate the small displacement assumptions. We therefore replace the external load p in the small-deformation equilibrium equations (5) by αp, where α is a given scalar satisfying $0 < \alpha < 1$. In the numerical examples we use $\alpha = 0.5$.

The global stability conditions (4) guarantee that the mechanism is stable in the initial configuration, while the conditions (2) guarantee that the mechanism is stable in the final configuration. These conditions, however, do not guarantee that the mechanism is stable for the entire load path.

2.4 Degree of freedom analysis

In this section we present a way to compute the mechanical degrees of freedom of a mechanism which is independent of the geometry of the mechanism and correctly accounts for the possibility that some bars are overlapping.

Assuming for now that a mechanism has *no redundant elements*, then the number of mechanical degrees of freedom d_m of the mechanism that is supported in a statically determinate manner can be calculated by the following equation based on a two-dimensional version of Maxwell's rule, see e.g. [18], [40], and [25],

$$d_m = 2\,(\text{number of nodes}) - (\text{number of bars}) - 3. \tag{6}$$

Here, '-3' means that the mechanism is supported in a statically determinate manner, i.e. that the mechanism can neither translate nor rotate. The formula (6) is used to define necessary conditions on the topology variables so that the mechanism has exactly one degree of freedom. In order to count the number of active nodes, additional binary variables, called *node activity variables*, $v_k \in \{0,1\}$ for $k = 1,\dots,N$ are introduced. The number of active nodes, i.e. nodes for which $v_k = 1$, can easily be determined by summing the node activity variables.

Since the ground-structure allows connections (bars) between each pair of nodes it intrinsically contains redundant bars. Therefore, the number of bars cannot be counted in a straightforward way in the mechanical degree of freedom analysis. The counting method applied in this paper works by detecting the number of *independent bars* $m(a)$ in the ground-structure. The number of independent bars is given by the rank of the so-called *linear equilibrium matrix* $C(a,\chi) \in \mathbb{R}^{2N\times n}$ [18]. The j-th column in the matrix C is given by

$$C_j(a,\chi) = a_j B_j \chi / l_j(\chi), \quad j = 1,\dots,n.$$

Due to the regularity of the nodal positions in the original ground-structure, the above method fails to detect overlapping elements as independent. This is unfortunate since overlapping bars are often useful for large displacement mechanisms. In order to distinguish overlapping bars as independent, the nodal positions are rearranged within the degree of freedom analysis. One approach is to create what is called a *connectivity circle*, shown in Figure 4 for the benchmark example, by relocating all the nodes onto a unit circle with the positions

$$(\tilde{x}_k, \tilde{y}_k) = (\cos\theta_k, \sin\theta_k),$$

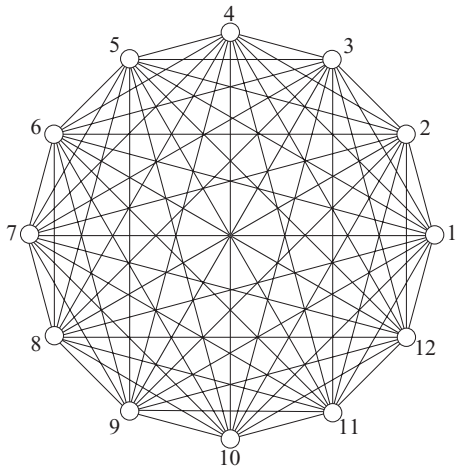


Fig. 4 Connectivity circle for the benchmark example ($\beta = 0$).

where the angles θ_k are given by

$$\theta_k = \frac{2\pi}{N}\Big((k-1) + \beta 2^{-2^k}\Big).$$

Here β is an optional parameter for avoiding parallel edges in the circle, for further details we refer to [32]. In the numerical examples we use $\beta = 1$. The number of independent bars is then redefined as

$$m(a) = \operatorname{rank}(C(a, \tilde{\chi})),$$

where $\tilde{\chi} = (\tilde{x}_1, \tilde{y}_1, \ldots, \tilde{x}_N, \tilde{y}_N)^\top \in \mathbb{R}^{2N}$. Finally, the degree of freedom analysis writes

$$d_m(a, v) = 2\sum_{k=1}^{N} v_k - m(a) - 3. \tag{7}$$

Due to the definition of the $m(a)$ and the sum of the node activity variables that appear in (7) the number of mechanical degrees of freedom is only well-defined for $(a, v)^\top \in \mathbb{B}^n \times \mathbb{B}^N$.

3 The basic topology optimization problem

Since the pure topology optimization problem is of significant importance in its own right and the formulation contains (mathematical) structures which should be exploited, but which are lost when the positions of the nodes in the ground-structures are allowed to change, this problem will be treated separately. The changes and amendments to the method and analysis necessary to enable variation in the nodal positions are outlined in a separate section.

3.1 Motivation for the optimization problem

During the developments of the mechanical model and the early formulations of the optimization problem it became apparent that the branch and bound method would take full advantage of any (and all) shortcomings in the mechanical and optimization models. In the first attempts no global stability or stress constraints were included in the problem formulation.

The first obtained optimal designs therefore violated the small strain assumption by allowing the deformed length of certain bars to approach zero by compressing the bars, thus allowing the strain to approach its lower bound, i.e. $\epsilon_j \to -1/2$. The small strain assumption is therefore enforced by adding stress constraints to the feasible set of the optimization problem. If $\sigma_j^{\min} < 0$ and $\sigma_j^{\max} > 0$ are given stress limits in compression and tension respectively, then the stresses $\sigma_j(u)$ defined as $\sigma_j(u) = E_j\epsilon_j(u)$ must satisfy

$$\sigma_j^{\min} \le E_j\epsilon_j(u) \le \sigma_j^{\max}, \quad \forall\ j\ : a_j = 1. \tag{8}$$

Note that stress constraints must only be enforced for bars which have a nonzero area, otherwise these constraints will make the feasible set overly stringent. Constraints of this type are often called design-dependent.

The lack of global stability constraints further revealed the branch and bound method's capabilities to exploit weaknesses of the mechanical model. If the stability condition $\nabla^2_{uu}\Pi(a,u) \succeq 0$ is omitted, a displacement vector may be found satisfying $\nabla_u\Pi(a,u) = 0$ and $\nabla^2_{uu}\Pi(a,u) \not\succeq 0$ meaning that the mechanism will potentially not be able to return to the original configuration if the external forces are reversed. If, on the other hand, the linear buckling constraints $K(a) + G(a,u_0) \succeq 0$ are omitted, the solution shown in Figure 3(e) becomes the optimal design for the benchmark problem. In this example there exist at least two displacement vectors which satisfy both $\nabla_u\Pi(a,u) = 0$ and $\nabla^2_{uu}\Pi(a,u) \succeq 0$. One with an output port displacement of 0.82 and the other with an output port displacement of -0.05 ! Both the topologies in Figure 3(a) and 3(e) are unstable with respect to the linear buckling conditions.

3.2 Problem formulation

In this section we present the pure topology optimization problem. The problem under consideration is to design a stable mechanism with exactly one mechanical degree of freedom which, given an external input force $p \in \mathbb{R}^d$, maximizes the output port displacement in the desired direction defined by the vector $c \in \mathbb{R}^d$. For manufacturing purposes, we wish to control the number of bars V_n in the final mechanism and therefore add the requirement that

$$\sum_{j=1}^{n} a_j = V_n, \tag{9}$$

where $V_n > 0$ is a given integer. Because of the requirement on the mechanical degrees of freedom the number of active nodes can also be safely controlled by adding the requirement

$$\sum_{k=1}^{N} v_k = V_N, \tag{10}$$

where $V_N > 0$ is an integer chosen such that V_N and V_n satisfy $2V_N - V_n - 3 = 1$. Due to the requirement (10), the number of mechanical degrees of freedom becomes independent of v and for brevity we write $d_m(a) = 1$.

In order to obtain a practical mechanism that can transfer energy or work from the input ports to the output port, it is required that at least two bars incident upon node k are present if node k is active, see [33] for details. This can be modeled by the linear inequalities

$$\sum_{j\in A_k} a_j \geq 2v_k, \quad k = 1,\ldots,N, \tag{11}$$

where A_k is an index set containing the bars incident upon node k. Similarly, nodes which are inactive ($v_k = 0$) are prevented from having any incident bars present. These requirements may be enforced by the following linear constraints

$$a_j \leq v_k, \quad j \in A_k, \quad k = 1,\ldots,N. \tag{12}$$

Formally, the topology design problem is stated as the nonconvex mixed integer optimization problem

$$\begin{array}{lll}
\text{maximize} & c^\top u & \\
\text{subject to} & \nabla_u\Pi(a,u) = 0, \nabla^2_{uu}\Pi(a,u) \succeq 0, & \\
& K(a)u_0 = p, K(a) + G(a,u_0) \succeq 0, & \\
& \sigma_j^{\min} \leq E_j\epsilon_j(u) \leq \sigma_j^{\max}, & \forall\ j\ : a_j = 1, \\
& d_m(a) = 1, & \\
& e^\top v = V_N, e^\top a = V_n, & \\
& \displaystyle\sum_{j\in A_k} a_j \geq 2v_k, & k = 1,\ldots,N, \\
& a_j \leq v_k, & k = 1,\ldots,N, \quad j \in A_k, \\
& u^{\min} \leq u, u_0 \leq u^{\max}, & \\
& a \in \{0,1\}^n, v \in \{0,1\}^N, &
\end{array} \tag{13}$$

where the variables are $(a, u, u_0, v)^\top \in \mathbb{B}^n \times \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{B}^N$; $u^{\min} \in \mathbb{R}^d$ and $u^{\max} \in \mathbb{R}^d$ are given displacement limits. The displacement bounds are included in the formulation for several reasons. They give the designer the possibility to control the size of the deformed mechanism and to prevent movement of the nodes that violate the planar motion assumption such as nodes which cross over each other. The displacement bounds also make the feasible set of the topology design problem (13) bounded, a property which is essential for constructing convex relaxations of the problem.

In order to define a convergent branch and bound method for solving the nonconvex mixed integer program (13) it is necessary to construct a convex relaxation of (13) which can provide good upper bounds on the optimal objective function value of (13). There exist a multitude of possibilities to construct convex relaxations of (13). Our choice is motivated by a compromise between the tightness (quality) of the relaxation and the computational burden to solve it. Furthermore, the choice is colored by the availability of fast and robust solvers for solving the relaxation. Our aim is to construct smooth convex nonlinear programming relaxations which can be solved by standard methods such as sequential quadratic programming.

To construct a convex relaxation of the nonconvex mixed integer problem (13), the nonlinear semidefinite stability constraints $\nabla^2_{uu}\Pi(a, u) \succeq 0$, the stability constraints $K(a)u_0 = p$, $K(a) + G(a, u_0) \succeq 0$ as well as the degree of freedom constraint $d_m(a) = 1$ are relaxed, simply by temporarily removing them as well as the small-displacement variables u_0.

The degree of freedom constraint is removed since it is only well-defined for topologies satisfying $(a, v)^\top \in \mathbb{B}^n \times \mathbb{B}^N$. If a candidate topology is found, either by partitioning or by some heuristic method, this constraint is reintroduced in the form of a *feasibility test*. This feasibility test is also used as a pruning test. If the topology is completely fixed at some node in the search tree, due to branching, the node can safely be fathomed if the feasibility test determines that the degree of freedom constraint is not satisfied.

The stability constraints are removed since relaxing them will introduce linear matrix inequalities and matrix valued variables in the convex relaxations. Although the quality of the relaxations would improve, the solution time would increase drastically.

The stability constraints $K(a)u_0 = p$, $K(a) + G(a, u_0) \succeq 0$ are also reintroduced in a feasibility test which is called every time a candidate topology $\bar{a} \in \mathbb{B}^n$ is found either by branching or by heuristics. In the feasibility test it is determined if there exist a displacement vector u_0 such that $K(\bar{a})u_0 = p$ and $K(\bar{a}) + G(\bar{a}, u_0) \succeq 0$. This can be done by solving a linear semidefinite program.

To guarantee that the stability constraints $\nabla^2_{uu}\Pi(a, u) \succeq 0$ are satisfied, a more elaborate feasibility test, called a *pruning test*, is used. Again, this test is called for a fixed topology and the aim is to deduce if a certain domain defined by displacement bounds (and the fixed topology) may be excluded from further examination of the branch and bound method. This pruning test can also be conducted by solving a linear semidefinite program.

After the removal of these complicating constraints a nonconvex mixed integer relaxation of (13) is obtained. This problem is then reformulated to contain only linear, bilinear, and convex quadratic terms. This reformulation is achieved by introducing additional variables and constraints. The bilinear terms are then relaxed using the convex and concave envelopes of bilinear terms over a box in $\mathbb{R}^2$. Finally, the integer requirements on the topology and node activity variables are relaxed. The relaxation that will be used to determine upper bounds on the objective function value of (13) has a linear objective function, linear equality and inequality constraints, and convex quadratic inequality constraints. The quadratic inequalities are then approximated by linear inequalities to give a linear programming relaxation of (13). This relaxation will then be used to define bound contraction sub-problems that are used to strengthen the nonlinear relaxation.

3.3 Reformulation and relaxation of the problem

To remove the design-dependence in the stress constraints (8) and to reduce the order of nonlinearity on the equilibrium equations additional variables $f = (f_1, \ldots, f_n)^\top \in \mathbb{R}^n$ representing the bar forces and $\epsilon = (\epsilon_1, \ldots, \epsilon_n)^\top \in \mathbb{R}^n$ representing the strains are introduced together with the bilinear equalities

$$f_j = a_j E_j \epsilon_j, \quad j = 1, \ldots, n, \tag{14}$$

and the quadratic equalities

$$\epsilon_j = \frac{1}{l_j^2} b_j^\top u + \frac{1}{2l_j^2} u^\top \tilde{B}_j u, \quad j = 1, \ldots, n. \tag{15}$$

The stress constraints (8) may, using the above introduced force and strain variables, be written as linear inequality constraints. If the stress constraints are multiplied by a_j then the force variables must satisfy

$$a_j\sigma_j^{\min} \leq f_j \leq a_j\sigma_j^{\max}, \quad j = 1, \ldots, n. \tag{16}$$

The following lemma shows how bilinear equalities involving a binary variable and a bounded continuous variable can equivalently be expressed as a set of linear inequalities.

Lemma 3.1 *Let $\epsilon^{\min} \in \mathbb{R}$ and $\epsilon^{\max} \in \mathbb{R}$ be given constants and let $\mathcal{M} = \{(a, f, \epsilon) \in \{0, 1\} \times \mathbb{R} \times \mathbb{R} \mid \epsilon^{\min} \leq \epsilon \leq \epsilon^{\max}\}$. Then $(a, f, \epsilon) \in \mathcal{M}$ satisfies the bilinear equation*

$$f = a\epsilon$$

if and only if $(a, f, \epsilon) \in \mathcal{M}$ satisfies the four linear inequalities

$$a\epsilon^{\min} \leq f \leq a\epsilon^{\max},\ (1-a)\epsilon^{\min} \leq \epsilon - f \leq (1-a)\epsilon^{\max}.$$

A proof of Lemma 3.1 is given in [45].

Because of the boundedness of the displacement variables and the integer constraints on the design variables the bilinear constraints (14) are equivalently written as the linear inequalities

$$a_jE_j\epsilon_j^{\min} \leq f_j \leq a_jE_j\epsilon_j^{\max}, \quad j = 1, \ldots, n, \tag{17}$$

$$(1-a_j)E_j\epsilon_j^{\min} \leq E\epsilon_j - f_j \leq (1-a_j)E_j\epsilon_j^{\max}, \quad j = 1, \ldots, n, \tag{18}$$

where the constant lower and upper bounds on the strain variables, denoted by $\epsilon_j^{\min} \in \mathbb{R}$ and $\epsilon_j^{\max} \in \mathbb{R}$ respectively, can be determined by solving, for each j, the two quadratic programs

$$\epsilon_j^{\min} = \frac{1}{l_j^2} \underset{u^{\min} \leq u \leq u^{\max}}{\text{minimize}} \quad b_j^\top u + \frac{1}{2}u^\top \tilde{B}_j u \tag{19}$$

and

$$\epsilon_j^{\max} = \frac{1}{l_j^2} \underset{u^{\min} \leq u \leq u^{\max}}{\text{maximize}} \quad b_j^\top u + \frac{1}{2}u^\top \tilde{B}_j u. \tag{20}$$

Problem (19) is a convex quadratic program and can thus be efficiently solved using an active set or interior point method. Problem (20) on the other hand is a maximization problem with a convex objective function which must be solved to global optimality. It is well-known that the optimal solution will be located at an extreme point of the feasible set. Since the problem can be transformed to a problem in local coordinates with at most four variables it can be solved by exhaustively enumerating the extreme points of the feasible set.

Since the gradient of the total potential energy is given by

$$\nabla_u\Pi(a, u) = K_0u + \sum_{j=1}^{n} E_jl_ja_j\epsilon_j(u)\nabla\epsilon_j(u) - p$$

where the gradient of the strain in the j-th bar is given by

$$\nabla\epsilon_j(u) = \frac{1}{l_j^2}(b_j + \tilde{B}_ju)$$

it follows that the equilibrium equations $\nabla_u\Pi(a, u) = 0$ expressed in the force and displacement variables are given by

$$K_0u + \sum_{j=1}^{n} \frac{f_j}{l_j}(b_j + \tilde{B}_ju) - p = 0. \tag{21}$$

A nonconvex mixed integer relaxation of the topology design problem (13) containing only linear, bilinear, and quadratic terms is given by the following problem in the variables $(a, u, v, \epsilon, f)^\top \in \mathbb{B}^n \times \mathbb{R}^d \times \mathbb{B}^N \times \mathbb{R}^n \times \mathbb{R}^n$.

$$\begin{array}{ll} \text{maximize} & c^\top u \\ \text{subject to} & (15), (16), (17), (18), (21), \\ & (9), (10), (11), (12), \\ & u^{\min} \le u \le u^{\max}, \\ & a \in \{0,1\}^n, v \in \{0,1\}^N. \end{array} \tag{22}$$

3.4 Convex relaxations

To construct a convex relaxations of the mixed integer nonconvex programs (22) and (13), we follow the practice used in e.g. [35], [3], and [24], and replace every bilinear term with its convex and potentially also concave envelope. The convex envelope of a function g is denoted by co g while the concave envelope is denoted by cav g. Let Ω denote a box in $\mathbb{R}^2$ defined by

$$\Omega = \left\{(\xi, \eta) \in \mathbb{R}^2 \mid \xi^{\min} \le \xi \le \xi^{\max},\ \eta^{\min} \le \eta \le \eta^{\max}\right\}.$$

In the following two lemmas, which will be used frequently throughout the remainder of this paper, the convex and concave envelopes of bilinear terms are stated.

Lemma 3.2 *The convex envelope of $\xi\eta$ over Ω is*

$$co\ \xi\eta = \max\{\xi^{\min}\eta + \eta^{\min}\xi - \xi^{\min}\eta^{\min}, \xi^{\max}\eta + \eta^{\max}\xi - \xi^{\max}\eta^{\max}\}.$$

A proof of Lemma 3.2 is given in [4].

Lemma 3.3 *The concave envelope of $\xi\eta$ over Ω is*

$$cav\ \xi\eta = \min\{\xi^{\min}\eta + \eta^{\max}\xi - \xi^{\min}\eta^{\max}, \xi^{\max}\eta + \eta^{\min}\xi - \xi^{\max}\eta^{\min}\}.$$

A proof of Lemma 3.3 is given in [2].

A critical property of the above convex and concave envelopes to guarantee convergence of a branch and bound method for problems with bilinear terms is stated in the next lemma.

Lemma 3.4 *co $\xi\eta = \xi\eta$ and cav $\xi\eta = \xi\eta$ for all (ξ, η) on the boundary of Ω.*

A proof of Lemma 3.4 is given in [2].

Since the matrices $\tilde{B}_j$ are positive semidefinite, convex relaxations of the quadratic equality constraints (15) are obtained by replacing them by the convex inequalities

$$b_j^\top u + \frac{1}{2}u^\top \tilde{B}_j u \le l_j^2 \epsilon_j, \quad j = 1, \ldots, n. \tag{23}$$

This relaxation is not strong enough to guarantee convergence of a branch and bound method. We therefore also approximate the nonconvex quadratic inequalities

$$b_j^\top u + \frac{1}{2}u^\top \tilde{B}_j u \ge l_j^2 \epsilon_j \tag{24}$$

by replacing them with the concave envelope of the bilinear constraints

$$b_j^\top u + \tfrac{1}{2}u^\top v_j \ge l_j^2 \epsilon_j, \quad j = 1, \ldots, n,$$

where the additional variables $v_j \in \mathbb{R}^d$ are defined by

$$v_j = \tilde{B}_j u, \quad j = 1, \ldots, n. \tag{25}$$

To this end, we need upper and lower bounds on the components in v_j. These bounds are given by

$$v_j^{\max} = \tilde{B}_j^+ u^{\max} + \tilde{B}_j^- u^{\min}$$

and

$$v_j^{\min} = \tilde{B}_j^+ u^{\min} + \tilde{B}_j^- u^{\max},$$

where $\tilde{B}_j^+ \in \mathbb{R}^{d\times d}$ and $\tilde{B}_j^- \in \mathbb{R}^{d\times d}$ denote the matrices consisting of the positive and negative parts of $\tilde{B}_j$, respectively. Introduce additional variables $\omega_{ij} \in \mathbb{R}$ corresponding to the bilinear terms $u_i v_{ij}$. The variables ω_{ij} should satisfy the linear inequalities

$$\begin{aligned} \omega_{ij} &\le u_i^{\min} v_{ij} + u_i v_{ij}^{\max} - u_i^{\min} v_{ij}^{\max}, \\ \omega_{ij} &\le u_i^{\max} v_{ij} + u_i v_{ij}^{\min} - u_i^{\max} v_{ij}^{\min}. \end{aligned} \tag{26}$$

Relaxations of the constraints (24) are now given by the linear inequalities

$$b_j^\top u + \frac{1}{2}\sum_{i=1}^{d} \omega_{ij} \ge l_j^2 \epsilon_j. \tag{27}$$

The equilibrium equations (21) are bilinear in the variables f_j and $v_j (= \tilde{B}_j u)$. Therefore, additional variables $\eta_j \in \mathbb{R}^d$ corresponding to $f_j v_j$ are introduced. The variables η_j must satisfy the linear inequalities

$$\begin{aligned} \eta_j &\ge f_j v_j^{\max} + f_j^{\max} v_j - f_j^{\max} v_j^{\max}, \\ \eta_j &\ge f_j v_j^{\min} + f_j^{\min} v_j - f_j^{\min} v_j^{\min}, \\ \eta_j &\le f_j v_j^{\min} + f_j^{\max} v_j - f_j^{\max} v_j^{\min}, \\ \eta_j &\le f_j v_j^{\max} + f_j^{\min} v_j - f_j^{\min} v_j^{\max}, \end{aligned} \tag{28}$$

where the lower and upper bounds on the force variables are $f_j^{\min} = \sigma_j^{\min}$ and $f_j^{\max} = \sigma_j^{\max}$. A relaxation of the equilibrium constraints (21) is given by the linear equality constraints

$$K_0 u + \sum_{j=1}^{n} \frac{1}{l_j}(f_j b_j + \eta_j) - p = 0. \tag{29}$$

A continuous convex relaxation of the nonconvex mixed integer program (22) is now given by the quadratically constrained linear program

$$\begin{array}{ll} \text{maximize} & c^\top u \\ \text{subject to} & (16), (17), (18), \\ & (23), (25), (26), (27), \\ & (28), (29), \\ & (9), (10), (11), (12), \\ & u^{\min} \le u \le u^{\max}, \\ & a \in [0,1]^n, v \in [0,1]^N, \end{array} \tag{30}$$

where the variables are $(a, u, v, \epsilon, f, \eta, \omega, \nu)$. Since the convex problem (30) is a relaxation of the nonconvex relaxation (22) it is also a relaxation of the original topology design problem (13). The above discussion motivates the following proposition.

Proposition 3.1 *Problem (30) is a convex relaxation of (13).*

If the variables ω_{ij} are forced to satisfy also the linear inequalities

$$\begin{aligned} \omega_{ij} &\ge u_i^{\min} v_{ij} + u_i v_{ij}^{\min} - u_i^{\min} v_{ij}^{\min}, \\ \omega_{ij} &\ge u_i^{\max} v_{ij} + u_i v_{ij}^{\max} - u_i^{\max} v_{ij}^{\max}, \end{aligned} \tag{31}$$

then a relaxation of the quadratic inequalities (24) is given by the linear equality

$$b_j^\top u + \frac{1}{2}\sum_{i=1}^{d} \omega_{ij} = l_j^2 \epsilon_j. \tag{32}$$

If the convex quadratic inequalities (23) and the linear inequalities (27) are replaced by the linear equalities (32) and the additional inequalities (31) are added to the relaxation (30), then the following linear program is obtained.

$$\begin{array}{ll} \text{maximize} & c^\top u \\ \text{subject to} & (16), (17), (18), \\ & (25), (26), (28), (29), (31), (32), \\ & (9), (10), (11), (12), \\ & u^{\min} \le u \le u^{\max}, \\ & a \in [0,1]^n, v \in [0,1]^N. \end{array} \tag{33}$$

Proposition 3.2 *Problem (33) is a linear programming relaxation of (13).*

In practical computations when solving the relaxations (30) and (33), the additional variables v_j need not be explicitly used, but are kept here for readability. Furthermore, since the matrices $\tilde{B}_j$ are structured and very sparse, most of the entries in v_j, η_j, and ω_j can be fixed to zero and therefore omitted in the computations, in fact, at most four entries in each of the above vectors can be nonzero. The number of variables in (30) is, after this reduction, about $15n + d + N$. The number of convex quadratic constraints in (30) is n while the number of linear constraints, excluding bounds, is about $35n + d$.

3.5 Valid inequalities

Since the stability constraints $K(a)u_0 = p$ and $K(a) + G(a, u_0) \succeq 0$ and the degree of freedom constraint $d_m(a) = 1$ have been removed in the relaxations it is possible that the branch and bound method will produce, through branching or heuristics, topologies which cannot satisfy these constraints. In order to prevent these topologies from being generated several times, we propose to exclude these topologies using valid inequalities which are stored in a global cut pool, see e.g. [39]. If the method generates a design, denoted by $\bar{a} \in \mathbb{B}^n$ such that $e^T\bar{a} = V_n$, which does not satisfy the linear stability and/or degree of freedom constraints then every feasible topology $a \in \mathbb{B}^n$ must differ from $\bar{a}$ in at least two components. Since each feasible design $a \in \mathbb{B}^n$ must satisfy $e^T a = V_n$ this condition may be written as $a^\top\bar{a} \le V_n - 1$. This discussion motivates the following two lemmas.

Lemma 3.5 *If $\bar{a} \in \mathbb{B}^n$, $e^\top\bar{a} = V_n$, and $d_m(\bar{a}) \neq 1$ then*

$$a^\top\bar{a} \le V_n - 1 \tag{34}$$

is a valid inequality for (13).

Lemma 3.6 *If $\bar{a} \in \mathbb{B}^n$, $e^\top\bar{a} = V_n$, and $\nexists\, u_0 \in \mathbb{R}^d$ such that $K(\bar{a})u_0 = f$ and $K(\bar{a}) + G(\bar{a}, u_0) \succeq 0$ then*

$$a^\top\bar{a} \le V_n - 1 \tag{35}$$

is a valid inequality for (13).

To determine if $\exists\, u_0 \in \mathbb{R}^d$ such that the given topology $\bar{a}$ satisfies the linear buckling constraints $K(\bar{a})u_0 = p$ and $K(\bar{a}) + G(\bar{a}, u_0) \succeq 0$ we propose to solve the linear semidefinite program

$$\begin{array}{ll} \underset{u_0,\mu}{\text{maximize}} & \mu \\ \text{subject to} & K(\bar{a}) + G(\bar{a}, u_0) \succeq \mu I, \\ & K(\bar{a})u_0 = p, \\ & u^{\min} \le u_0 \le u^{\max}. \end{array} \tag{36}$$

If an optimal solution (u_0^*, μ^*) to (36) is found with $\mu^* \ge 0$ the suggested mechanism satisfies the linear buckling conditions. If on the other hand, an optimal solution with $\mu^* < 0$ is found or if the linear constraints make the problem infeasible, then the valid inequality (35) is added to the cut pool.

3.6 Pruning test for the stability constraints

Since the stability conditions $\nabla^2_{uu}\Pi(a,u) \geq 0$ are not considered in the convex relaxations (30) and (33), additional steps must be taken to ensure convergence of the branch and bound method. To guarantee convergence we propose to deal with the stability conditions using a pruning test in which a liberal estimate of the minimal eigenvalue of $\nabla^2_{uu}\Pi(\bar{a},u) \geq 0$ over the domain $u^{\min} \leq u \leq u^{\max}$ and for a given topology $\bar{a} \in \mathbb{B}^n$ is computed. Consider the nonconvex semidefinite program

$$\begin{array}{ll} \underset{u,\mu}{\text{maximize}} & \mu \\ \text{subject to} & \nabla^2_{uu}\Pi(\bar{a},u) \geq \mu I, \\ & \nabla_u \Pi(\bar{a},u) = 0, \\ & u^{\min} \leq u \leq u^{\max}. \end{array} \tag{37}$$

If there exist a feasible solution (u^*,μ^*) of (37) with $\mu^* \geq 0$ then there exist a displacement vector satisfying the stability conditions (2) over the given domain. If on the other hand an optimal solution to (37) satisfies $\mu^* < 0$ or if problem (37) is infeasible, then it can be concluded that no such displacement vector exist.

A weaker, but tractable, test is obtained if instead a convex relaxation of (37) is solved. We will next present a linear semidefinite program which is a relaxation of (37) and which also satisfy the property that if the upper and lower bound on the displacements approach each other component wise, then the optimal objective value of this relaxation will approach the minimal eigenvalue of $\nabla^2_{uu}\Pi(\bar{a},u)$ over the domain. Again, let $\bar{a} \in \mathbb{B}^n$ denote a given topology generated by the branch and bound method at a node τ in the search tree and let $u^{\min,\tau}$ and $u^{\max,\tau}$ denote the bounds on the displacement variables at the node τ.

The Hessian of the potential energy for the topology $\bar{a}$ can be written as

$$\nabla^2_{uu}\Pi = K_0 + \sum_{j=1}^{n} \bar{a}_j E_j l_j \epsilon_j(u) \nabla^2 \epsilon_j(u) + \sum_{j=1}^{n} \bar{a}_j E_j l_j \nabla \epsilon_j(u) (\nabla \epsilon_j(u))^\top.$$

Using the variables introduced in Section 3.4 the Hessian can be written as

$$\nabla^2_{uu}\Pi = K(\bar{a}) + \sum_{j=1}^{n} \frac{f_j}{l_j} \tilde{B}_j + \sum_{j=1}^{n} \frac{\bar{a}_j E_j}{l_j^3} (b_j \nu_j^\top + \nu_j b_j^\top + \nu_j \nu_j^\top).$$

To relax the bilinear terms $\nu_j \nu_j^\top$ we introduce additional positive semidefinite matrix variables $V_j \in \mathbb{R}^{d\times d}$ together with the relaxation

$$\begin{pmatrix} 1 & \nu_j^\top \\ \nu_j & V_j \end{pmatrix} \geq 0. \tag{38}$$

This relaxation is not strong enough and hence we approximate the bilinear terms $\nu_j \nu_j^\top$ using the linear component-wise inequalities

$$\begin{array}{l} V_j \geq \nu_j^{\max} \nu_j^\top + \nu_j (\nu_j^{\max})^\top - \nu_j^{\max} (\nu_j^{\max})^\top, \\ V_j \geq \nu_j^{\min} \nu_j^\top + \nu_j (\nu_j^{\min})^\top - \nu_j^{\min} (\nu_j^{\min})^\top, \\ V_j \leq \nu_j^{\max} \nu_j^\top + \nu_j (\nu_j^{\min})^\top - \nu_j^{\max} (\nu_j^{\min})^\top, \\ V_j \leq \nu_j^{\min} \nu_j^\top + \nu_j (\nu_j^{\max})^\top - \nu_j^{\min} (\nu_j^{\max})^\top. \end{array} \tag{39}$$

To conclude, the linear semidefinite program

$$\begin{array}{ll} \text{maximize} & \mu \\ \text{subject to} & K(\bar{a}) + \sum_{j=1}^{n} \frac{f_j}{l_j} \tilde{B}_j + \sum_{j=1}^{n} \frac{\bar{a}_j E_j}{l_j^3} (b_j \nu_j^\top + \nu_j b_j^\top + V_j) \geq \mu I, \\ & (14), (25), (26), (28), (29), (31), (32), \\ & (38), (39), \\ & u^{\min,\tau} \leq u \leq u^{\max,\tau}, \end{array} \tag{40}$$

where the variables are $(u, \mu, \epsilon, f, \eta, \omega, \nu, V_1, \ldots, V_n)$, is a relaxation of (37) over the domain $u^{\min,\tau} \leq u \leq u^{\max,\tau}$ that can be used as a pruning test. If an optimal solution to (40) satisfies $\mu^* \geq 0$ then the node in the

branch and bound tree may produce solutions which satisfy the stability conditions (2). If on the other hand an optimal solution to (40) satisfies $\mu^* < 0$ or if problem (40) is infeasible, then the node τ may be pruned from the search tree.

In practice, the size of (40) is rather modest. First of all, we only need to consider those j for which $\bar{a}_j = 1$. Furthermore, since at most four entries in the vectors v_j are non-zero only the essential 4×4 part of the matrices V_j need to be taken into account. Moreover, some of the degrees of freedom are irrelevant and hence the size of the approximate Hessian can be reduced.

4 An extended problem

A natural extension of the pure topology design problem (13) is to include also the positions of the nodes in the ground-structure as variables, i.e. to perform simultaneous topology and geometry optimization (simultaneous type and dimensional synthesis). One advantage of this possibility is that the ground-structure need not be very dense, and thus, the number of binary topology variables can be kept at a modest level. Methods for simultaneous topology and geometry optimization of stiff truss structures is considered in e.g. [38], [15], and [9]. In this section we present the modifications to the topology design problem (13) which are necessary to include the nodal positions as variables. The possibility to change the positions of the nodes in the undeformed configuration greatly influence the degree of nonlinearity in the problem since the undeformed lengths l_j and the vector b_j are now functions of the nodal position variables χ. This section is structured in a similar fashion as the previous section. First the simultaneous topology and geometry problem is presented as a nonconvex mixed integer program. Then this problem is reformulated and relaxed so that the relaxation only contains linear, bilinear, and quadratic terms. Two continuous convex relaxations to the extended problem are presented. Finally, the valid inequalities and pruning test from the previous section are generalized.

4.1 Problem formulation for the extended problem

In this section we present the simultaneous topology and geometry design problem. The problem under consideration is to design a stable mechanism with one mechanical degree of freedom which, given an input force p, maximizes the output port displacement in the desired direction c. Formally, this design problem is stated as the nonconvex mixed integer optimization problem

$$
\begin{array}{lll}
\text{maximize} & c^\top u & \\
\text{subject to} & \nabla_u \Pi(a,u,\chi) = 0, \nabla^2_{uu}\Pi(a,u,\chi) \succeq 0, & \\
& K(a,\chi)u_0 = p, K(a,\chi) + G(a,u_0,\chi) \succeq 0, & \\
& \sigma_j^{\min} \le E_j \epsilon_j(u,\chi) \le \sigma_j^{\max}, & \forall\ j\ : a_j = 1, \\
& d_m(a) = 1, & \\
& e^\top v = V_N, e^\top a = V_n, & \\
& \displaystyle\sum_{j \in A_k} a_j \ge 2 v_k, & k = 1,\dots,N, \\
& a_j \le v_k, & k = 1,\dots,N, \quad j \in A_k, \\
& u^{\min} \le u, u_0 \le u^{\max}, & \\
& \chi^{\min} \le \chi \le \chi^{\max}, & \\
& a \in \{0,1\}^n, v \in \{0,1\}^N, &
\end{array}
\tag{41}
$$

where the variables are $(a,u,u_0,\chi,v)^\top \in \mathbb{B}^n \times \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^{2N} \times \mathbb{B}^N$; $\chi^{\min} \in \mathbb{R}^{2N}$ and $\chi^{\max} \in \mathbb{R}^{2N}$ are given limits on the nodal positions.

4.2 Reformulation and relaxation of the extended problem

To remove the design dependence in the stress constraints (8) we follow the practice from the reformulation of the topology design problem and reintroduce the additional variables $f = (f_1,\dots,f_n)^\top \in \mathbb{R}^n$ representing the bar forces and $\epsilon = (\epsilon_1,\dots,\epsilon_n)^\top \in \mathbb{R}^n$ representing the strains together with the bilinear equalities

$$
f_j = a_j E_j \epsilon_j, \quad j = 1,\dots,n. \tag{42}
$$

Since both the displacement and nodal position variables are assumed to be bounded the strain will also be bounded in the context of topology and geometry optimization. Therefore, the bilinear equalities (42) are equivalently written as the linear inequalities

$$a_j E_j \epsilon_j^{\min} \leq f_j \leq a_j E_j \epsilon_j^{\max}, \quad j = 1, \ldots, n, \tag{43}$$

$$(1 - a_j) E_j \epsilon_j^{\min} \leq E\epsilon_j - f_j \leq (1 - a_j) E_j \epsilon_j^{\max}, \quad j = 1, \ldots, n. \tag{44}$$

Since the Green-Lagrange strain measure is now a function of both displacements u and nodal positions χ, i.e.

$$\epsilon_j(u, \chi) = \frac{\hat{l}_j^2(u, \chi) - l_j^2(\chi)}{2l_j^2(\chi)},$$

the lower and upper bounds on the strain variables $\epsilon_j^{\min}$ and $\epsilon_j^{\max}$ need to be recalculated. To this end it is convenient to introduce additional variables $l = (l_1, \ldots, l_n)^\top \in \mathbb{R}^n_+$ representing the lengths of the undeformed bars and $q = (q_1, \ldots, q_n)^\top \in \mathbb{R}^n_+$ representing the square lengths of the undeformed bars together with the quadratic equalities

$$q_j = l_j^2, \quad j = 1, \ldots, n \tag{45}$$

and

$$q_j = \chi^\top B_j \chi, \quad j = 1, \ldots, n. \tag{46}$$

The lower and upper bounds on the square lengths denoted by $q_j^{\min}$ and $q_j^{\max}$ can be determined by solving the quadratic programs

$$q_j^{\min} = \underset{\chi^{\min} \leq \chi \leq \chi^{\max}}{\text{minimize}} \quad \chi^\top B_j \chi \tag{47}$$

and

$$q_j^{\max} = \underset{\chi^{\min} \leq \chi \leq \chi^{\max}}{\text{maximize}} \quad \chi^\top B_j \chi. \tag{48}$$

Similarly, by defining deformed square lengths $\hat{q} = (\hat{q}_1, \ldots, \hat{q}_n)^\top \in \mathbb{R}^n$ as

$$\hat{q}_j = \hat{\chi}^\top B_j \hat{\chi}, \quad j = 1, \ldots, n,$$

where $\hat{\chi}$ denotes the actual nodal positions after deformation, the lower and upper bounds $\hat{q}_j^{\min}$ and $\hat{q}_j^{\max}$ can be determined by solving the quadratic programs

$$\hat{q}_j^{\min} = \underset{\hat{\chi}^{\min} \leq \hat{\chi} \leq \hat{\chi}^{\max}}{\text{minimize}} \quad \hat{\chi}^\top B_j \hat{\chi} \tag{49}$$

and

$$\hat{q}_j^{\max} = \underset{\hat{\chi}^{\min} \leq \hat{\chi} \leq \hat{\chi}^{\max}}{\text{maximize}} \quad \hat{\chi}^\top B_j \hat{\chi}. \tag{50}$$

Problems (47) and (49) are convex quadratic programs that can be efficiently solved using an active set or interior point method. Problems (48) and (50) are maximization problems with a convex objective function that can be solved to global optimality by exhaustive enumeration of the extreme points of the feasible set. Assuming that $q_j^{\min}$ and $q_j^{\max}$ are strictly positive, the lower and upper bounds on the strains $\epsilon_j^{\min}$ and $\epsilon_j^{\max}$ are given by

$$\epsilon_j^{\min} = \frac{1}{2} \frac{\hat{q}_j^{\min}}{q_j^{\max}} - \frac{1}{2}$$

and

$$\epsilon_j^{\max} = \frac{1}{2} \frac{\hat{q}_j^{\max}}{q_j^{\min}} - \frac{1}{2}.$$

In the context of geometry optimization the equalities defining the strains are no longer quadratic since the lengths are now functions of the nodal positions. We therefore modify the definition of the strains by introducing additional variables $b_j \in \mathbb{R}^d$ satisfying

$$b_j = \hat{B}_j \chi, \quad j = 1, \ldots, n. \tag{51}$$

Using the newly introduced variables q_j and b_j the strains are now redefined with the quadratic equalities

$$\epsilon_j q_j = b_j^\top u + \frac{1}{2} u^\top \tilde{B}_j u, \quad j = 1, \ldots, n. \tag{52}$$

The equilibrium equations (21) contain the terms $1/l_j$. By introducing additional variables $g = (g_1, \ldots, g_n)^\top \in \mathbb{R}^n$ representing scaled forces, i.e. $g_j = f_j/l_j$ the equilibrium equations are rewritten as

$$K_0 u + \sum_{j=1}^{n} g_j (b_j + \tilde{B}_j u) - p = 0, \tag{53}$$

where the variables g_j must satisfy

$$f_j = g_j l_j, \quad j = 1, \ldots, n. \tag{54}$$

A nonconvex mixed integer relaxation of (41) containing only linear, bilinear, and quadratic terms is now given by the problem

$$\begin{array}{ll} \text{maximize} & c^\top u \\ \text{subject to} & (16), (43), (44), (45), (46), \\ & (51), (52), (53), (54), \\ & (9), (10), (11), (12), \\ & u^{\min} \le u \le u^{\max}, \\ & \chi^{\min} \le \chi \le \chi^{\max}, \\ & a \in \{0,1\}^n, v \in \{0,1\}^N, \end{array} \tag{55}$$

where the variables are $(a, u, \chi, v, \epsilon, f, l, q, b)^\top \in \mathbb{B}^n \times \mathbb{R}^d \times \mathbb{R}^{2N} \times \mathbb{B}^N \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n_+ \times \mathbb{R}^n_+ \times \mathbb{R}^{nd}$.

4.3 Convex relaxations of the extended problem

In order to construct a convex relaxation of the nonconvex mixed integer problems (55) and (41), the bilinear terms in (55) are replaced by their convex and concave envelopes.

Since the matrices B_j are positive semidefinite convex relaxations of the quadratic equality constraints (45) and (46) defining the variables l_j and q_j are obtained by replacing the equalities by inequalities according to

$$l_j^2 \le q_j, \quad j = 1, \ldots, n \tag{56}$$

and

$$\chi^\top B_j \chi \le q_j, \quad j = 1, \ldots, n. \tag{57}$$

To approximate the nonconvex quadratic equalities $l_j^2 = q_j$ by linear inequalities we replace them with

$$\begin{array}{l} q_j \ge 2 l_j l_j^{\max} - l_j^{\max} l_j^{\max}, \\ q_j \ge 2 l_j l_j^{\min} - l_j^{\min} l_j^{\min}, \\ q_j \le l_j (l_j^{\min} + l_j^{\max}) - l_j^{\min} l_j^{\max}. \end{array} \tag{58}$$

To approximate the nonconvex quadratic equalities $\chi^\top B_j \chi = q_j$ we replace them with the convex and concave envelope of each of the bilinear terms in $\chi^\top \zeta_j = q_j$ where the additional variables $\zeta_j \in \mathbb{R}^{2N}$ are defined by

$$\zeta_j = B_j \chi, \quad j = 1, \ldots, n. \tag{59}$$

To this end, we need upper and lower bounds on the components in ζ_j which are given by

$$\zeta_j^{\max} = B_j^+ \chi^{\max} + B_j^- \chi^{\min}$$

and

$$\zeta_j^{\min} = B_j^+ \chi^{\min} + B_j^- \chi^{\max}.$$

We also introduce auxiliary variables ξ_{ij} that correspond to the bilinear term $\chi_i \zeta_{ij}$. The auxiliary variables ξ_{ij} should satisfy the linear inequalities

$$\begin{array}{l} \xi_{ij} \ge \chi_i^{\max} \zeta_{ij} + \chi_i \zeta_{ij}^{\max} - \chi_i^{\max} \zeta_{ij}^{\max}, \\ \xi_{ij} \ge \chi_i^{\min} \zeta_{ij} + \chi_i \zeta_{ij}^{\min} - \chi_i^{\min} \zeta_{ij}^{\min}, \\ \xi_{ij} \le \chi_i^{\min} \zeta_{ij} + \chi_i \zeta_{ij}^{\max} - \chi_i^{\min} \zeta_{ij}^{\max}, \\ \xi_{ij} \le \chi_i^{\max} \zeta_{ij} + \chi_i \zeta_{ij}^{\min} - \chi_i^{\max} \zeta_{ij}^{\min}. \end{array} \tag{60}$$

Now, the quadratic equalities $\chi^\top B_j \chi = q_j$ can be approximated by the linear equalities

$$q_j = \sum_{i=1}^{2N} \xi_{ij}, \quad j = 1, \dots, n. \tag{61}$$

The equalities (52) defining the strain in the case of geometry optimization contain bilinear terms in ϵ_j and q_j. Therefore, we introduce auxiliary variables $\varphi = (\varphi_1, \dots, \varphi_n)^\top \in \mathbb{R}^n$ that correspond to the bilinear term $\epsilon_j q_j$. The auxiliary variables φ_j should satisfy the linear inequalities

$$\begin{aligned}
\varphi_j &\geq \epsilon_j q_j^{\max} + \epsilon_j^{\max} q_j - \epsilon_j^{\max} q_j^{\max}, \\
\varphi_j &\geq \epsilon_j q_j^{\min} + \epsilon_j^{\min} q_j - \epsilon_j^{\min} q_j^{\min}, \\
\varphi_j &\leq \epsilon_j q_j^{\min} + \epsilon_j^{\max} q_j - \epsilon_j^{\max} q_j^{\min}, \\
\varphi_j &\leq \epsilon_j q_j^{\max} + \epsilon_j^{\min} q_j - \epsilon_j^{\min} q_j^{\max}.
\end{aligned} \tag{62}$$

With regards to the quadratic terms $u^\top \tilde{B}_j u$ in the equalities (52), we reuse the auxiliary variables $\omega_{ij} \in \mathbb{R}$ introduced in Section 3.4 together with the linear inequalities (26) and (31). Here, we also add the convex quadratic inequalities

$$u^\top \tilde{B}_j u \leq \sum_{i=1}^{d} \omega_{ij}, \quad j = 1, \dots, n, \tag{63}$$

to provide a tight relaxation. Moreover, the equalities (52) contain the bilinear terms $b_j^\top u$ and therefore we need to introduce auxiliary variables $\kappa_{ij} \in \mathbb{R}$ approximating the bilinear terms $b_{ij} u_i$ that satisfy

$$\begin{aligned}
\kappa_{ij} &\geq u_i^{\max} b_{ij} + u_i b_{ij}^{\max} - u_i^{\max} b_{ij}^{\max}, \\
\kappa_{ij} &\geq u_i^{\min} b_{ij} + u_i b_{ij}^{\min} - u_i^{\min} b_{ij}^{\min}, \\
\kappa_{ij} &\leq u_i^{\min} b_{ij} + u_i b_{ij}^{\max} - u_i^{\min} b_{ij}^{\max}, \\
\kappa_{ij} &\leq u_i^{\max} b_{ij} + u_i b_{ij}^{\min} - u_i^{\max} b_{ij}^{\min},
\end{aligned} \tag{64}$$

where the upper and lower bounds on b_j are given by

$$b_j^{\max} = \hat{B}_j^+ \chi^{\max} + \hat{B}_j^- \chi^{\min},$$

and

$$b_j^{\min} = \hat{B}_j^+ \chi^{\min} + \hat{B}_j^- \chi^{\max}.$$

Finally, a linear relaxation of the nonconvex definition of the strain stated in (52) is given by

$$\varphi_j = \sum_{i=1}^{d} \kappa_{ij} + \frac{1}{2} \sum_{i=1}^{d} \omega_{ij}. \tag{65}$$

The equilibrium equations (53) are bilinear in g_j and b_j. We therefore introduce auxiliary variables $h_j \in \mathbb{R}^d$ approximating $g_j b_j$ that satisfy the linear inequalities

$$\begin{aligned}
h_j &\geq g_j b_j^{\max} + g_j^{\max} b_j - g_j^{\max} b_j^{\max}, \\
h_j &\geq g_j b_j^{\min} + g_j^{\min} b_j - g_j^{\min} b_j^{\min}, \\
h_j &\leq g_j b_j^{\min} + g_j^{\max} b_j - g_j^{\max} b_j^{\min}, \\
h_j &\leq g_j b_j^{\max} + g_j^{\min} b_j - g_j^{\min} b_j^{\max},
\end{aligned} \tag{66}$$

where $g_j^{\min}$ and $g_j^{\max}$ denote the lower and upper bounds on g_j. The equilibrium equations (53) are also bilinear in g_j and $\nu_j = \tilde{B}_j u$. We therefore introduce auxiliary variables $\eta_j \in \mathbb{R}^d$ approximating $g_j \nu_j$ that satisfy the linear inequalities

$$\begin{aligned}
\eta_j &\geq g_j \nu_j^{\max} + g_j^{\max} \nu_j - g_j^{\max} \nu_j^{\max}, \\
\eta_j &\geq g_j \nu_j^{\min} + g_j^{\min} \nu_j - g_j^{\min} \nu_j^{\min}, \\
\eta_j &\leq g_j \nu_j^{\min} + g_j^{\max} \nu_j - g_j^{\max} \nu_j^{\min}, \\
\eta_j &\leq g_j \nu_j^{\max} + g_j^{\min} \nu_j - g_j^{\min} \nu_j^{\max}.
\end{aligned} \tag{67}$$

For the bilinear equalities (54), the force variables f_j have to satisfy the linear inequalities

$$\begin{aligned} f_j &\geq g_j l_j^{\max} + g_j^{\max} l_j - g_j^{\max} l_j^{\max}, \\ f_j &\geq g_j l_j^{\min} + g_j^{\min} l_j - g_j^{\min} l_j^{\min}, \\ f_j &\leq g_j l_j^{\min} + g_j^{\max} l_j - g_j^{\max} l_j^{\min}, \\ f_j &\leq g_j l_j^{\max} + g_j^{\min} l_j - g_j^{\min} l_j^{\max}. \end{aligned} \tag{68}$$

Finally, the equilibrium equations (53) are approximated by the linear equalities

$$K_0 u + \sum_{j=1}^{n} (h_j + \eta_j) - p = 0. \tag{69}$$

A continuous convex relaxation of the topology and geometry design problem (41) is now given by the following quadratically constrained linear program in the variables $(a, u, \chi, v, \epsilon, f, l, q, b, \zeta, \xi, h, \eta, g, \varphi, \kappa, \omega, \nu)$.

$$\begin{array}{ll} \text{maximize} & c^\top u \\ \text{subject to} & (16), (26), (31), (43), (44), \\ & (56), (57), (58), (59), (60), \\ & (61), (62), (63), \\ & (64), (65), (66), \\ & (67), (68), (69), \\ & (9), (10), (11), (12), \\ & u^{\min} \leq u \leq u^{\max}, \\ & \chi^{\min} \leq \chi \leq \chi^{\max}, \\ & a \in [0,1]^n, v \in [0,1]^N. \end{array} \tag{70}$$

The above discussion motivates the following proposition.

Proposition 4.1 *Problem (70) is a convex relaxation of (41).*

If the convex quadratic inequalities (56), (57) and (63) are removed from the convex relaxation (70), a linear programming relaxation of (55) and the topology and geometry problem (41) is obtained.

$$\begin{array}{ll} \text{maximize} & c^\top u \\ \text{subject to} & (16), (26), (31), (43), (44), \\ & (58), (59), (60), (61), (62), \\ & (64), (65), (66), \\ & (67), (68), (69), \\ & (9), (10), (11), (12), \\ & u^{\min} \leq u \leq u^{\max}, \\ & \chi^{\min} \leq \chi \leq \chi^{\max}, \\ & a \in [0,1]^n, v \in [0,1]^N. \end{array} \tag{71}$$

Proposition 4.2 *Problem (71) is a linear programming relaxation of (41).*

In practical computations when solving the relaxations (70) and (71), the additional variables b_j, ν_j, and ζ_j need not be explicitly used, but are kept here for readability. Furthermore, since the matrices B_j, $\tilde{B}_j$, and $\hat{B}_j$ are structured and very sparse, most of the entries in ν_j, ζ_j, b_j, η_j, and ω_j can be fixed to zero and therefore omitted in the computations, in fact, at most four entries in each of the above vectors can be nonzero.

4.4 Valid inequalities and pruning tests for the extended problem

In this section we generalize, if possible, the valid inequalities and pruning tests derived for the pure topology design problem in Section 3.5 and 3.6.

Since the number of mechanical degrees of freedom is assumed to be independent of the geometry of the mechanism, it is impossible to change the mechanical degrees of freedom of the mechanism by changing the nodal positions after the topology is fixed. Therefore the following lemma, which is a generalization of Lemma 3.5, holds.

Lemma 4.1 *If $\bar{a} \in \mathbb{B}^n$, $e^\top \bar{a} = V_n$, and $d_m(\bar{a}) \neq 1$ then*

$$a^\top \bar{a} \leq V_n - 1$$

is a valid inequality for (41).

In Section 3.5 we proposed that valid inequalities should be used to conveniently remove topologies that fail to satisfy the linear buckling conditions (5) and (4). It is impossible to generalize this result as stated in Lemma 3.6 to hold for the simultaneous topology and geometry design problem as the following example illustrates. Consider the unstable mechanism in Figure 3(e) and the stable mechanism in Figure 3(f). By moving the three nodes 8, 10, and 12 in the unstable mechanism shown in Figure 3(e) to the left it is possible to obtain the stable mechanism shown in Figure 3(f). Therefore, we propose to use a pruning test similar to the tests developed in Section 3.6 to determine if a certain domain of the feasible set may be excluded from further investigation due to these constraints.

Let $\bar{a} \in \mathbb{B}^n$ denote a topology generated by the branch and bound method at a node τ in the search tree and let $u^{\min,\tau}$ and $u^{\max,\tau}$ and $\chi^{\min,\tau}$ and $\chi^{\max,\tau}$ denote the lower and upper bounds on the displacement and nodal position variables at node τ. To decide if node τ can be removed from the search tree we solve a linear semidefinite relaxation of the nonconvex semidefinite program

$$\begin{array}{ll} \underset{u_0,\chi,\mu}{\text{maximize}} & \mu \\ \text{subject to} & K(\bar{a},\chi) + G(\bar{a},u_0,\chi) \succeq \mu I, \\ & K(\bar{a},\chi)u_0 = p, \\ & \chi^{\min,\tau} \leq \chi \leq \chi^{\max,\tau}, \\ & u^{\min} \leq u_0 \leq u^{\max}. \end{array} \tag{72}$$

If an optimal solution (u_0^*, χ^*, μ^*) to the relaxation of (72) fails to satisfy $\mu^* > 0$ or if the relaxation is infeasible, then node τ in the search tree is fathomed.

To determine if a certain domain of the feasible set may be excluded from further investigation due to the stability constraints $\nabla^2_{uu}\Pi(a,u,\chi) \succeq 0$ we propose to generalize the pruning test from Section 3.6. In this test we solve a linear semidefinite relaxation of the nonconvex semidefinite program

$$\begin{array}{ll} \underset{u,\chi,\mu}{\text{maximize}} & \mu \\ \text{subject to} & \nabla^2_{uu}\Pi(\bar{a},u,\chi) \succeq \mu I, \\ & \nabla_u \Pi(\bar{a},u,\chi) = 0, \\ & \chi^{\min,\tau} \leq \chi \leq \chi^{\max,\tau}, \\ & u^{\min,\tau} \leq u \leq u^{\max,\tau}. \end{array} \tag{73}$$

If an optimal solution (u^*, χ^*, μ^*) to the relaxation of (73) fails to satisfy $\mu^* > 0$ or if the relaxation is infeasible, then node τ is pruned from the search tree.

4.5 Limitations of the models

The major drawback of the mechanism design formulations (13) and (41) and the choice of optimization method is the lack of possibilities to control the load-displacement path during the deformation of the mechanism. Even though the mechanism is stable in its undeformed configuration and also stable in the final deformed configuration it may very well be unstable at some stage between the undeformed and the fully deformed configurations. Furthermore, both the displacement and stress constraints may be violated for intermediate configurations. Thus, the final decision if the mechanism performs well or not falls on the engineering judgment of the designer.

A positive side effect of the branch and bound method and a possible remedy of the above problem is the (likely) possibility to find more than one feasible solution to the considered problem. In fact, all of the six mechanisms shown in Figure 3 were found as by products of a single attempt to solve an instance of (13). This allows the designer to choose an alternative mechanism if the optimal design, for some reason, does not behave satisfactory.

5 The branch and bound method

In this section we present a convergent implementation of the branch and bound method for solving the nonconvex mixed integer mechanism design problems (13) and (41). In many aspects the suggested method resembles the branch and reduce method described in [42] and the branch and contract method described in [50]. For an introduction to branch and bound in the context of continuous global optimization we refer the reader to [30].

Before presenting the method, we would like to reintroduce the notion of *complicating variables* (also called *nonconvex variables*, see e.g. [42] and [50]) which are defined as those variables that appear in nonconvex terms in the optimization problems. Furthermore, we denote the subset of the complicating variables which appear in the original problem statements (13) and (41) as *primary variables*. These variables play a crucial role in the branching rule to guarantee practical convergence of the method since the stability constraints in the original problems are removed in the relaxations.

Upper bounds on the optimal objective function value of (13) and (41) are obtained by solving the nonlinear convex relaxations (30) and (70) respectively. The linear relaxations developed in Section 3.4 and Section 4.3 serve two purposes. They are used to improve the bounds on the variables, by so called feasibility based range reductions described in Section 5.4. This procedure improves the quality of the convex relaxations. The linear relaxations are also used as a fall-back routine if numerical or other problems occur during the solution of the nonlinear relaxations.

5.1 Branching

For brevity, we only describe the branching rule for the topology optimization problem (13). The branching rule for the topology and geometry problem (41) is analogous.

Branching is first performed on the integer variables in a predetermined order of priority. First branching is done on the node activity variables v and then on the design variables a. The branching variable is chosen by the maximum integer infeasibility criterion and branching is done by variable dichotomy.

If all integer variables are fixed or if the optimal solution to the relaxation satisfies the integer requirements then branching is done on one of the complicating continuous variables. Every twenty fifth continuous branching we perform bisection on the primary variable with the largest distance between its upper and lower bound. This rule does not improve the performance of the method in practice but is included for the theoretical convergence.

In all other situations, the approximation errors $|\omega_{ij} - u_i v_{ij}|$ and $|\eta_{ij} - f_j v_{ij}|$ of the bilinear terms are computed at the solution to the convex relaxation. If at least one of these errors is sufficiently large, then the displacement variable or force variable responsible for the largest of the errors is chosen as branching variable. The branching point is chosen as the value of the branching variable in the solution to the convex relaxation. If, on the other hand, all of these errors are small (in the implementation the limit is 10^{-5}) we resort to bisection of the primary variable with the largest distance between its upper and lower bound.

5.2 Node selection

The node chosen for further refinement is the one where the highest upper bound is attained (the best-bound-first rule). This node selection rule is, by definition, bound improving, see [30].

5.3 Heuristics

To find feasible solutions to the topology design problem (13) we propose the following heuristics. Given an optimal solution to the convex relaxation (30) at some node in the search tree the following steps are performed. The design and node variables are rounded to the nearest integer. If the rounded values satisfies the resource constraints (9), (10), (11), (12), the degree of freedom constraint, and there exist a small-displacement vector such that the linear stability constraints (4) and (5) are satisfied, then local minimization is performed on (22) with the area and node variables fixed to the rounded values and the other variables constrained to be within the bounds at the node. If no feasible solution is found using local minimization an attempt is made to solve the mechanical analysis problem using a second order line search method for unconstrained problems, see e.g. [28]. The search direction in this method is a weighted sum of a descent direction and a direction of negative curvature. Since the number of variables, here the number of degrees

of freedom, is modest (≤ 100) the spectral decomposition of the Hessian of the potential energy is used to determine the search directions. If a displacement vector satisfying the equilibrium equations (1) and the stability conditions (2) is found, the strain and force variables are computed using the definitions (15) and (14), respectively.

To find feasible solutions to the topology and geometry design problem (41) we propose the following heuristic. Given an optimal solution to the convex relaxation (70) at some node in the search tree the following steps are performed. The design and node variables are rounded to the nearest integer. If the rounded values satisfies the resource constraints (9), (10), (11), (12), and the degree of freedom constraint, then local minimization is performed on (55) with the area and node variables fixed to the rounded values and the other variables constrained to be within the bounds at the node.

5.4 Pre-processing, range reductions, and pruning tests

Before solving one of the convex nonlinear relaxations (30) or (70) to obtain upper bounds on the objective function value of (13) or (41) the bounds on the complicating variables are, if possible, improved by simple pre-processing and feasibility based range reductions. Pre-processing is in common use in linear, quadratic, and mixed integer programming, see e.g. [5], [36], and [43]. Here we will use implications given by the simple resource constraints (9), (10), (11), (12), and the constitutive law (14) to fix or improve the bounds on some of the variables. Before the relaxations are constructed and solved at node τ in the search tree the following simple pre-processing steps are performed.

- If $v_k^{\max,\tau} = 0$ then $a_j^{\max,\tau} \leftarrow 0$ for $j \in A_k$.
- If $e^\top v^{\min,\tau} = V_N$ then $v_k^{\max,\tau} \leftarrow v_k^{\min,\tau}$ for $k = 1, \ldots, N$.
- If $e^\top v^{\max,\tau} = V_N$ then $v_k^{\min,\tau} \leftarrow v_k^{\max,\tau}$ for $k = 1, \ldots, N$.
- If $e^\top a^{\min,\tau} = V_n$ then $a_j^{\max,\tau} \leftarrow a_j^{\min,\tau}$ for $j = 1, \ldots, n$.
- If $e^\top a^{\max,\tau} = V_n$ then $a_j^{\min,\tau} \leftarrow a_j^{\max,\tau}$ for $j = 1, \ldots, n$.
- If $a_j^{\max,\tau} = 0$ then $f_j^{\max,\tau} \leftarrow 0$, $f_j^{\min,\tau} \leftarrow 0$, $g_j^{\max,\tau} \leftarrow 0$, and $g_j^{\min,\tau} \leftarrow 0$.
- If $a_j^{\min,\tau} = 1$ then $\epsilon_j^{\min,\tau} \leftarrow \max\{\sigma_j^{\min}/E_j, \epsilon_j^{\min,\tau}\}$ and $\epsilon_j^{\max,\tau} \leftarrow \min\{\sigma_j^{\max}/E_j, \epsilon_j^{\max,\tau}\}$

Finally, if the node activity variable is fixed to zero then the corresponding displacement variables are fixed to zero and the corresponding node position variables are fixed to the original position of the node in the ground-structure. These operations are inexpensive and improve the performance of the method directly by improving the quality of the relaxations and indirectly by decreasing the number variables eligible for branching.

To further improve the quality of the convex relaxations at a node in the search tree we follow the practice developed in [50] and solve a sequence of bound contraction subproblems. For each complicating variable, which is not fixed at the current node in the search tree, we in turn maximize and minimize the value of the variable over the feasible set of the linear relaxations (33) or (71). If a bound can be improved it is immediately changed in the linear relaxation. After one pass over all the complicating variables the relaxations are reconstructed. This procedure is repeated four times if all the binary variables are fixed, two times if the node activity variables are fixed, and only once otherwise.

If one of the contraction subproblems is infeasible, then the relaxation is also infeasible and the node is immediately pruned from the search tree. After the convex relaxation is solved optimality based range reductions on the complicating variable bounds is performed as outlined in [42].

The pruning tests for the stability constraints derived in Section 3.6 and 4.4 are called if all binary variables are fixed and the bounds on the primary variables at the current node are sufficiently close. In the implementation this limit is 10^{-2}.

5.5 Convergence

Convergence of the branch and bound algorithm is guaranteed by the following observations, see [30, Theorem IV.3]. *(i)* The node selection rule is, by definition, bound improving. *(ii)* The bounding operation is consistent, see [42]. *(iii)* The valid inequalities added to the problem due to range reductions are valid, see [42] and [50]. *(iv)* The inequalities stated in Lemma 3.5 - 4.1 are valid. *(v)* The valid inequalities implied by the pre-processing steps presented in Section 5.4 are valid. *(vi)* The pruning tests in Section 3.6 and 4.4 guarantee that the stability and degree of freedom constraints, which are neglected in the convex relaxations, are eventually satisfied.

5.6 Implementation

The branch and bound method is implemented in MATLAB. The MATLAB code controls the overall program flow, the branch and bound tree, and the cut pool. All linear programs are solved using the fast and robust simplex solvers in CPLEX 9 while all convex quadratic programs are solved using the barrier solver in the same package. CPLEX is called from MATLAB using an interface which allows problem modifications such that CPLEX can take advantage of the current solution when resolving a problem after slight modifications. All nonlinear programs, such as the convex relaxations (30) and (70), are solved using the sequential quadratic programming package SNOPT 6.2, see [26, 27]. Again, an interface is used to access SNOPT from MATLAB. SNOPT does call back to MATLAB to compute the objective function and the nonlinear constraints as well as the gradient of the objective function and the Jacobian of the nonlinear constraints. Finally, all linear semidefinite programs are solved using the MATLAB toolbox SeDuMi, see [46].

The absolute and relative optimality tolerances in the branch and bound code are both set to 10^{-5}, while the integer feasibility tolerance is set to 10^{-7}. The feasibility tolerance for the simple bounds, linear constraints, and nonlinear constraints is 10^{-8}. The feasibility and optimality tolerances are set to 10^{-6} in SNOPT and 10^{-8} in CPLEX.

6 Numerical examples

In this section we present a number of academic examples and their solutions. The size of these problems, measured in the number of nodes and bars in the ground-structure and in the final design, is realistic and corresponds well to the size of the problems which need to be solved when designing planar mechanisms for real-life applications.

In all the examples presented below the design domain is a two dimensional square with width and height equal to one. The nodes in the ground-structure are evenly distributed over the design domain, N_x nodes in the horizontal direction and N_y nodes in the vertical direction. The total number of nodes is thus $N = N_x N_y$. Connections are allowed between each pair of nodes in the ground-structure, and hence, the number of potential bars is $n = N(N-1)/2$. The modulus of elasticity is in all examples set to $E_j = 10$ for all j. The stiffnesses of the linear springs at the input and output ports are denoted by k_{in} and k_{out}, respectively. The external load vector p and the output direction vector c have $+1$ or -1 entries in the positions corresponding to the degrees of freedom for the input and output port displacements and zero entries otherwise. The node activity variables corresponding to input and output ports are all fixed to one.

6.1 Enforcing symmetry in the examples

In the following examples the external load, the boundary conditions, and the entire ground-structure are symmetric around the horizontal mid-axis in the design domain. The optimal designs are therefore also forced to be symmetric. This restriction provides the possibility to compare the obtained solutions with previous solutions obtained for the same problems, reported in [33, 32].

Symmetry in the topology is enforced by removing redundant area variables. The number of binary topology variables left after the reduction is denoted by $\tilde{n}$. Symmetry conditions on the axial force, strain, displacements, node activity, and node position variables are enforced by explicitly adding linear constraints to the relaxations and heuristic problems. For the given load conditions, the displacement variables are horizontally symmetric and vertically anti-symmetric. Symmetry for the vertical displacements of the nodes

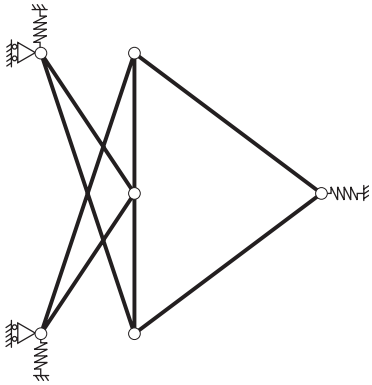


Fig. 5 Optimal design for the benchmark example without geometry variations.

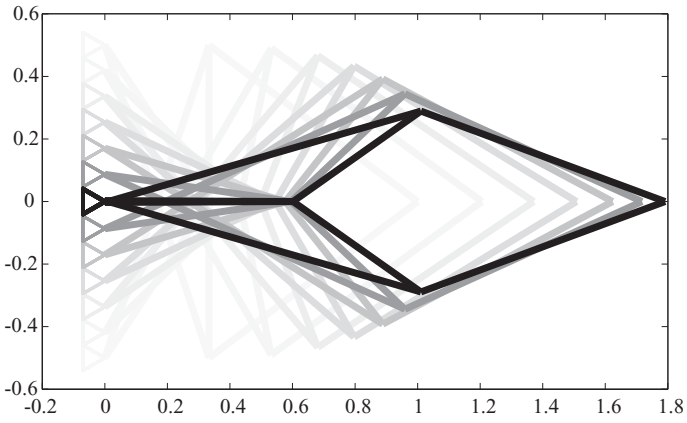


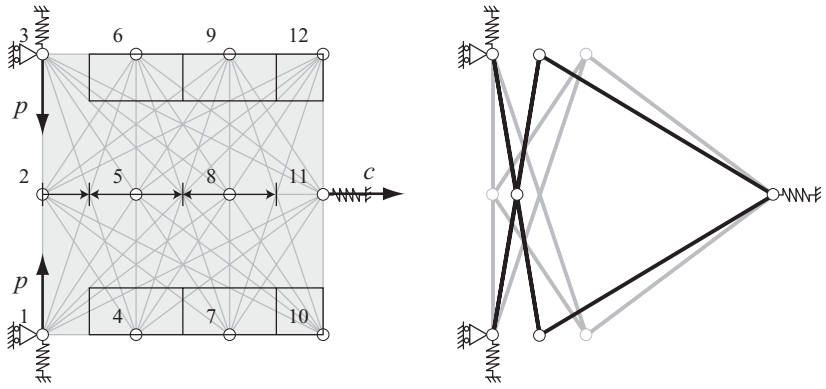
Fig. 6 Motion of the optimal design for the benchmark example.

on the horizontal mid-axis is achieved by the aid of roller supports attached to the nodes. All of these supports are, for clarity, not shown in the figures defining the ground-structures.

6.2 Benchmark example

The ground-structure and load condition for the first example are shown in Figure 2. The number of nodes in the horizontal and vertical direction are $N_x = 4$ and $N_y = 3$, respectively. Hence, the number of nodes $N = 12$ and the number of potential bars $n = 66$. The number of binary topology variables after variable reduction is $\tilde{n} = 38$. The requested number of active bars and nodes are $V_n = 8$ and $V_N = 6$, respectively. The stiffnesses of the linear springs at the input and output ports are set to $k_{in} = 2.0$ and $k_{out} = 0.02$. The lower and upper stress limits are $\sigma_j^{\min} = -0.05$ and $\sigma_j^{\max} = 0.05$ for all j. The displacement limits are ± 0.5 for vertical movement and ± 1.4 for horizontal movement. The optimal design to the topology optimization problem is shown in Figure 5 and its motion in Figure 6. The optimal objective function value is 0.791. The optimal topology shown in Figure 5 is exactly the same as the design found by exhaustive enumeration in [33].

In Figure 7(a) the settings for the simultaneous topology and geometry design problem are shown. The nodes are allowed to move $\pm 1/6$ in all directions if they remain within the design domain and satisfy the symmetry requirements. The bounds on the nodal positions are depicted by black boxes in the figure. In Figure 7(b) the optimal design to the simultaneous topology and geometry design problem is shown in black. To be able to compare the optimal topology with the design from the pure topology problem, the optimal design with initial nodal positions is shown in grey in Figure 7(b). The optimal objective function value for the simultaneous topology and geometry problem is 0.971.



(a) Ground-structure. (b) Optimal design.

Fig. 7 Ground-structure and optimal design for the benchmark example with geometry variations.

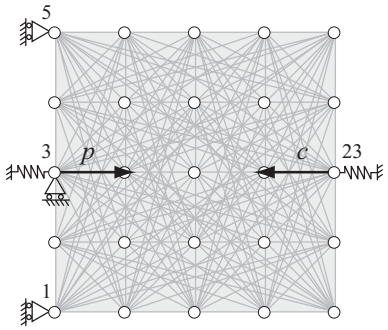


Fig. 8 Ground-structure for the inverter mechanism without geometry variations.

It should be noted that there is a change in the optimal topology between the pure topology design problem and the topology and geometry design problem. The optimal *topology* with node 2 as active in the simultaneous topology and geometry design problem is in fact unstable if the nodal positions are not re-designed (optimized).

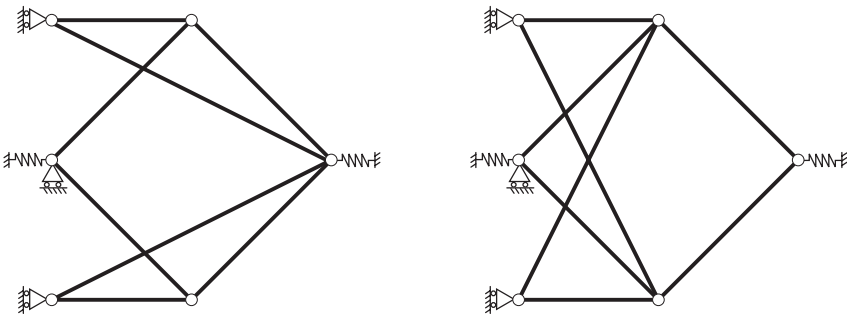
6.3 Inverter mechanism

In the second example, we intend to construct a force inverter, i.e. a mechanism for which the displacement of the output port is in the opposite direction of the direction of the applied external force. The ground-structure for the topology design problem is shown in Figure 8. The number of nodes in the horizontal and vertical direction are $N_x = N_y = 5$. Hence $N = 25$ and the number of potential bars $n = 300$.

In this example $V_n = 8$ and $V_N = 6$. The stiffnesses of the linear springs at the input and output ports are set $k_{in} = 2.5$ and $k_{out} = 2.5$. The lower and upper stress limits are $\sigma_j^{\min} = -0.5$ and $\sigma_j^{\max} = 0.5$ for all j. The displacement limits are ± 0.5 for both vertical and horizontal movement. We solve two instances of the topology design problem for this example. The second instance contains the additional requirements that the nodes labeled 1 and 5 in Figure 8 are restricted to move inwards.

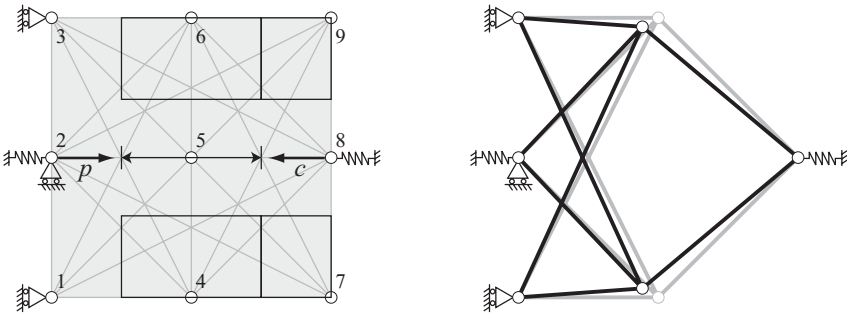
The optimal designs to the pure topology optimization problems for the two inverters are shown in Figure 9(a) and Figure 9(b). The optimal objective function values are 0.137 and 0.134, respectively. The design shown in Figure 9(b) is exactly the same as the design found by a partially heuristic method in [32].

In Figure 10(a) the settings for the simultaneous topology and geometry design problem are shown. The nodes are allowed to move $\pm 1/4$ in all directions if they remain within the design domain and satisfy the symmetry requirements. The bounds on the nodal positions are depicted by black boxes in the figure.



(a) First inverter. (b) Second inverter.

Fig. 9 Optimal designs for the two inverter problems (without geometry variations).



(a) Ground-structure. (b) Optimal design.

Fig. 10 Ground-structure and optimal design for the inverter problems with geometry variations.

In this case a coarser (3×3)-node ground-structure is used. The optimal solution to the topology design problem can, even with this coarser ground-structure, still be achieved. In Figure 10(b) the optimal design to the simultaneous topology and geometry design problem is shown in black. The optimal objective function value for the simultaneous topology and geometry problem is 0.168 regardless of the inward-motion requirement.

6.4 Computational results

Results concerning problem sizes, number of nodes in the search tree, and CPU-time for the numerical examples are collected in Table 1 and Table 2 below. In the tables N^t denotes the total number of nodes and N^m denotes the maximal number of active nodes in the branch and bound tree.

The computations are performed either on an Intel Pentium 4, 2.8 GHz clock frequency and 1 Gb internal memory or a SUN Fire 480 with an UltraSparc III processor, 900 MHz clock frequency and 2 Gb internal memory.

Only a few valid inequalities of the type (35) designed to remove unstable mechanisms are generated when solving the topology design problems and on the above problems they do not significantly improve on the total number of nodes in the search tree compared to if they are not generated. For the benchmark example, both of the unstable topologies shown in Figure 3 gave rise to one valid inequality. Even though only a few inequalities are generated, they are still very important. If they are not generated an unstable solution would be found for the benchmark example. No topologies were found that violated the degree of freedom constraints, and hence no such valid inequalities were generated.

Table 1 Pure topology optimization on the SUN.

Problem	N	n	$\tilde{n}$	d	V_n	V_N	CPU	N^t	N^m	Cuts	$c^\top u^*$
Benchmark	12	66	38	18	8	6	4m30s	81	33	2	0.791
Inverter 1	25	300	160	43	8	6	2h6m	467	115	2	0.137
Inverter 2	25	300	160	43	8	6	2h5m	409	92	2	0.134

Table 2 Simultaneous topology and geometry optimization on the PC.

Problem	N	n	$\tilde{n}$	d	V_n	V_N	CPU	N^t	N^m	$c^\top u^*$
Benchmark	12	66	38	18	8	6	1h12m	134	28	0.971
Inverter 1	9	36	21	13	8	6	2h49m	763	130	0.168
Inverter 2	9	36	21	13	8	6	1h55m	513	108	0.168

It may seem surprising that the total number of nodes in the search tree is small for all examples. This is explained by the problem dependent pre-processing techniques and the extensive range reductions that are performed at every node in the branch and bound tree. An attempt to solve the problems with pre-processing and range reductions turned off indicated that the problems could not be solved with the branch and bound method without these enhancements in reasonable time and with a reasonably small search tree.

For the three examples presented in Table 2 most of the computation time, on average more than 80 %, is spent on solving linear bound contraction problems. About 10 % is spent on solving the convex relaxation (70) and the remainder of the time is spent on solving the linear semidefinite programs for the pruning tests and maintaining the branch and bound tree.

It should be noted that the two inverter problems differ by only two simple bounds on the displacement variables. However, the second problem generates significantly fewer nodes in the search tree, particularly for the simultaneous topology and geometry design problem. In general, the solution time for all problems is sensitive to the bounds on the displacements.

We observed, in particular when solving the simultaneous topology and geometry problems, that for a fixed topology a there often exist many (u, u_0, χ) such that all constraints, except for the stability constraints $K(a,\chi) + G(a,u_0,\chi) \succeq 0$ and/or $\nabla^2_{uu}\Pi(a,u,\chi) \succeq 0$, are satisfied. Typically, these solutions also produce large output displacement and the nodes at which this occurs can therefore not be fathomed from the search tree by value dominance. When this occurs the branch and bound method without the stability pruning tests does not converge. The pruning tests are therefore crucial to guarantee convergence.

7 Conclusions and future research

We have presented a convergent branch and bound method for solving planar mechanism design problems modeled as nonconvex mixed integer programs. By using a truss ground-structure approach it is possible to simultaneously determine the optimal topology as well as the optimal geometry, i.e. to perform simultaneous type and dimensional synthesis. The branch and bound method is enhanced by improving the quality of the convex relaxations using valid inequalities and range reduction techniques. The numerical examples indicate that the suggested method can reliably solve planar mechanism design problems of realistic size. Computational experience shows that it is important to carefully model and include technological constraints such as stability and stress constraints in the optimization problem to prevent the optimal solution from violating implicit or explicit assumptions in the mechanical model.

In order to generalize the models and methods presented in this paper to the design of spatial mechanisms it will be important to develop conditions to prevent that some members of the mechanism obstruct the displacement path of other members. Another possibility for future research is to generalize the models and the branch and bound method to facilitate multiple load conditions. This extension together with a suitably chosen objective function will allow the approach presented in this paper to be used as a design tool for type and dimensional synthesis of path-following mechanisms.

Acknowledgements

We are very grateful to two anonymous referees for their careful reading of the manuscript and several constructive and insightful suggestions and comments. We would like to thank Martin P. Bendsøe at DTU for many fruitful discussions on this subject and valuable comments on early manuscripts. We also gratefully acknowledge the constructive suggestions by Ole Sigmund at DTU and Krister Svanberg at KTH.

References

[1] W. Achtziger, M.P. Bendsøe, A. Ben-Tal, and J. Zowe. Equivalent displacement based formulations for maximum strength truss topology design. *Impact of Computing in Science and Engineering*, 4(4):315–345, 1992.

[2] F.A. Al-Khayyal. Jointly constrained bilinear programs and related problems: An overview. *Computers and Mathematics with Applications*, 19(11):53–62, 1990.

[3] F.A. Al-Khayyal. Generalized bilinear programming: Part I. Models, applications and linear programming relaxation. *European Journal of Operations Research*, 60:306–314, 1992.

[4] F.A. Al-Khayyal and J.E. Falk. Jointly constrained biconvex programming. *Mathematics of Operations Research*, 8(2):273–286, 1983.

[5] E.D. Andersen and K.D. Andersen. Presolving in linear programming. *Mathematical Programming*, 71:221–245, 1995.

[6] K.-J. Bathe. *Finite Element Procedures in Engineering Analysis*. Prentice-Hall: Englewood Cliffs, NJ, 1982.

[7] A. Ben-Tal and M.P. Bendsøe. A new method for optimal truss topology design. *SIAM Journal on Optimization*, 3(2):322–358, 1993.

[8] A. Ben-Tal, F. Jarre, M. Kočvara, A. Nemirovski, and J. Zowe. Optimal design of trusses under a nonconvex global buckling constraint. *Optimization and Engineering*, 1:189–213, 2000.

[9] A. Ben-Tal, M. Kočvara, and J. Zowe. Two non-smooth methods for simultaneous geometry and topology design of trusses. In M.P. Bendsøe and C.A. Mota Soares, editors, *Topology Design of Structures*, pages 31–42. Kluwer Academic Publishers, 1993.

[10] A. Ben-Tal and A. Nemirovski. Potential reduction polynomial time method for truss topology design. *SIAM Journal on Optimization*, 4(3):596–612, 1994.

[11] A. Ben-Tal and A. Nemirovski. Optimal design of engineering structures. *OPTIMA Mathematical Programming Society Newsletter*, 1995. No. 47.

[12] A. Ben-Tal and A. Nemirovski. Robust truss topology design via semidefinite programming. *SIAM Journal on Optimization*, 7(4):991–1016, 1997.

[13] A. Ben-Tal and A. Nemirovski. *Handbook of semidefinite programming*, chapter Structural Design, pages 443–467. Kluwer Academic Publishers, 2000.

[14] M.P. Bendsøe. *Optimization of structural topology, shape, and material*. Springer-Verlag, 1995.

[15] M.P. Bendsøe, A. Ben-Tal, and J. Zowe. Optimization methods for truss geometry and topology design. *Structural Optimization*, 7(3):141–158, 1994.

[16] M.P. Bendsøe and O. Sigmund. *Topology Optimization: Theory, Methods and Applications*. Springer, 2003.

[17] S. Bollapragada, O. Ghattas, and J.N. Hooker. Optimal design of truss structures by logical-based branch and cut. *Operations Research*, 49(1):42–51, 2001.

[18] C.R. Calladine. Buckminster Fuller's tensegrity structures and Clerk Maxwell's rules for the construction of stiff frame. *International Journal of Solids and Structures*, 14:161–172, 1978.

[19] M.A. Crisfield. *Non-linear Finite Element Analysis of Solids and Structures, Vol. 1: Essentials*. John Wiley & Sons, New York, 1997.

[20] W.S. Dorn, R.E. Gomory, and H.J. Greenberg. Automatic design of optimal structures. *Journal de Mécanique*, 3:25–52, 1964.

[21] A.G. Erdman. Computer-aided mechanism design - now and the future. *Journal of Mechanical Design*, 117:93–100, 1995.

[22] A.G. Erdman and G.N. Sandor. *Mechanism Design: Analysis and Synthesis*, volume I. Prentice Hall, 2nd edition, 1990.

[23] H.A. Eschenauer and N. Olhoff. Topology optimization of continuum structures: A review. *Applied Mechanics Reviews*, 54(4):331–390, 2001.

[24] L.R. Foulds, D. Haugland, and K. Jörnsten. A bilinear approach to the pooling problem. *Optimization*, 24:165–180, 1992.

[25] P.W. Fowler and S.D. Guest. A symmetry extension of Maxwell's rule for rigidity of frames. *International Journal of Solids and Structures*, 37:1793–1804, 2000.

[26] P.E. Gill, W. Murray, and M.A. Saunders. User's guide for SNOPT 5.3: a Fortran package for large-scale nonlinear programming. Technical report, NA 97-5 , Department of Mathematics, University of California, 1997.

[27] P.E. Gill, W. Murray, and M.A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Journal on Optimization*, 12(4):979–1006, 2002.

[28] P.E. Gill, W. Murray, and M. Wright. *Practical Optimization*. Academic Press, 1981.

[29] J.M. Hansen. Synthesis of mechanisms using time-varying dimensions. *Structural and Multidisciplinary Optimization*, 7(1):127–144, 2002.

[30] R. Horst and H. Tuy. *Global Optimization: Deterministic Approaches*. Springer-Verlag, 1993.

[31] F. Jarre, M. Kočvara, and J. Zowe. Optimal truss design by interior point methods. *SIAM Journal on Optimization*, 8(4):1084–1107, 1998.

[32] A. Kawamoto, M.P. Bendsøe, and O. Sigmund. Articulated mechanism design with a degree of freedom constraint. *International Journal for Numerical Methods in Engineering*, 61(9):1520–1545, 2004.

[33] A. Kawamoto, M.P. Bendsøe, and O. Sigmund. Planar articulated mechanism design by graph theoretical enumeration. *Structural and Multidisciplinary Optimization*, 27(4):295–299, 2004.

[34] M. Kočvara. On the modelling and solving of the truss design problem with global stability constraints. *Structural and Multidisciplinary Optimization*, 23:189–203, 2002.

[35] G.P. McCormick. Computability of global solutions to factorable nonconvex programs: part I - convex underestimating problems. *Mathematical Programming*, 10:147–175, 1976.

[36] Cs. Mészáros and U.H. Suhl. Advanced preprocessing techniques for linear and quadratic programming. *OR Spectrum*, 25:575–595, 2003.

[37] R.J. Minnaar, D.A. Tortorelli, and J.A. Snyman. On nonassembly in the optimal dimensional synthesis of planar mechanisms. *Structural and Multidisciplinary Optimization*, 21(5):345–354, 2001.

[38] F. Nishino and R. Duggal. Shape optimum design of trusses under multiple loading. *International Journal of Solids and Structures*, 26:17–27, 1990.

[39] M. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale traveling salesman problems. *SIAM Review*, 33, 1991.

[40] S. Pellegrino and C.R. Calladine. Matrix analysis of statically and kinematically indeterminate frameworks. *International Journal of Solids and Structures*, 22:409–428, 1986.

[41] G.I.N. Rozvany, M.P. Bendsøe, and U. Kirsch. Layout optimization of structures. *Applied Mechanics Reviews*, 48:41–119, 1995.

[42] H.S. Ryoo and N.V. Sahinidis. A branch-and-reduce approach to global optimization. *Journal of Global Optimization*, 8:107–138, 1996.

[43] M.W.P. Savelsbergh. Preprocessing and probing for mixed integer programming problems. *ORSA Journal on Computing*, 6:445–454, 1994.

[44] M. Stolpe. Global optimization of minimum weight truss topology problems with stress, displacement, and local buckling constraints using branch-and-bound. *International Journal for Numerical Methods in Engineering*, 61(8):1270–1309, 2004.

[45] M. Stolpe and K. Svanberg. Modeling topology optimization problems as linear mixed 0-1 programs. *International Journal for Numerical Methods in Engineering*, 57(5):723–739, 2003.

[46] J.F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999. Version 1.05 available from `http://fewcal.kub.nl/sturm`.

[47] C.R. Tischler, A.E. Samuel, and K.H. Hunt. Kinematic chains for robot hands–I. Orderly number-synthesis. *Mechanism and Machine Theory*, 30:1193–1215, 1995.

[48] C.R. Tischler, A.E. Samuel, and K.H. Hunt. Kinematic chains for robot hands–II. Kinematic constraints, classification, connectivity, and actuation. *Mechanism and Machine Theory*, 30:1217–1239, 1995.

[49] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38:45–95, 1996.

[50] J.M. Zamora and I.E. Grossmann. A branch and contract algorithm for problems with concave univariate, bilinear and linear fractional terms. *Journal of Global Optimization*, 14:217–249, 1999.

[Paper 4]

Path-generation of articulated mechanisms by shape and topology variations in non-linear truss representation

A. Kawamoto*

Abstract

This paper presents studies on an optimization-based method for path-generation of articulated mechanisms. An extended truss ground-structure approach is taken in which both the shape and topology of the truss are designed using cross-sectional areas and nodal positions as design variables. This leads to a technique for simultaneous type and dimensional synthesis of articulated mechanisms. For the analysis part it is essential to control the mechanism configuration so that the mechanism remains within a given configuration space, thus stabilizing the optimization process and resulting in realistic solutions. This can be achieved by using the Levenberg-Marquardt method. The design method is illustrated by a number of design cases for both closed and open input and output paths.

***Key Words*:** Mechanism design; topology optimization; path-generation

1 Introduction

This paper deals with a design methodology for path-generation of *articulated mechanisms*. An articulated mechanism consists of several links (or bars) connected by joints and gains all the mobility from its joints. This is in contrast to a *compliant mechanism* which is a different type of mechanism that gains some or all of the mobility from the elasticity of its components. The path-generation problems are stated as designing an articulated mechanism that can trace a prescribed path at the output port for an input path given to the input port of the mechanism. Here an approximate synthesis approach (see, e.g., [1, 2]) is adopted to solve the problems in the sense that the prescribed target path is approximated by minimizing the error between the target path and the actual output path; the latter path is calculated through a kinematic analysis. The error is evaluated point-wise and defines a vector collecting distances between the corresponding target and actual points. In the optimization, the error vector is minimized in the infinity norm (i.e., the largest distance between the target and actual points).

A conventional mechanism design process is divided into the following three stages (see, e.g., [3, 4]). The first stage is formulation of the objective. The objective may, for example, be maximization of output displacement and generation of a prescribed output path for a given input. This paper is specifically devoted to the latter path-generation problems. (See references [5, 6] for work on displacement maximization problems of articulated mechanisms.) The second stage is called *type synthesis* which is mainly concerned with topological questions. In this stage the number of bars and joints and their connections are determined. An elaborate type synthesis technique based on number synthesis is developed in [7, 8]. In that approach much effort is invested for generating a complete list of kinematic chains by eliminating isomorphic graphs (a significant unsolved problem in number synthesis). The final stage is called *dimensional synthesis* and is mostly related to geometrical questions. In this stage the exact dimensions of all the mechanical components are determined. Reviews on dimensional synthesis techniques are found in [1, 2]. Although type and dimensional syntheses are normally separated in the engineering context, they are actually so closely connected that they should preferably be treated at the same time. In this paper, the so-called ground-structure approach [9] is taken to the development of a method for simultaneous type and dimensional synthesis.

Research on design of continuum-based compliant mechanisms [10, 11, 12, 13] using topology optimization in the finite element representation suggests that it should also be possible to obtain results for

*Correspondence to: Department of Mathematics, Technical University of Denmark, Matematiktorvet Building 303, DK-2800 Kgs. Lyngby, Denmark. e-mail: A.Kawamoto@mat.dtu.dk

articulated mechanisms in the truss ground-structure representation by applying essentially the same technology [14, 15]. One outstanding feature of the technology is that, unlike other parametric optimization methods, the method can optimize simultaneously both the topology and geometry of a mechanism (or structure) layout. Thus the technology optimization has good potential for being developed into a systematic method that can design articulated mechanisms, combining type and dimensional syntheses.

A common critical factor for designing both compliant and articulated mechanisms is geometric non-linearity. Throughout this paper, geometric non-linearity is considered using the Green-Lagrange strain measure. This is applied to the analysis of the mechanisms since the displacements of the mechanisms are intrinsically large. The consideration of geometric non-linearity, however, causes a numerical difficulty relating to the non-convexity of the equilibrium equations. In the articulated mechanism analysis, the issue is related to the geometry of the configuration spaces of the mechanism [16, 17]. It is essential to implement a numerical method that can consistently detect a stable equilibrium point because the quality of the equilibrium analysis directly affects the result of the associated sensitivity analysis; a sensitivity analysis that contains a lot of noise may impede convergence of the optimization process. Note that this is thus caused by the poor quality of the analysis and not by faults in the optimization algorithm.

The Newton-Raphson method is popular and widely used for solving the non-linear equilibrium equations. However, when the tangent stiffness matrix is not well-conditioned, the behavior of the method is unpredictable. Specifically in the truss-based articulated mechanism design, the problem can be viewed as a *configuration transition problem*. Therefore, it is important to make sure that the mechanism should stay within a given configuration space, thus stabilizing the optimization process and resulting in a realistic solution. This can be achieved by using the Levenberg-Marquardt method.

Another critical factor specific to articulated mechanism design is the *mechanical degrees of freedom* of a mechanism. This is defined as the number of independent inputs required to determine the position of all links of the mechanism with respect to a global coordinate system [4]. As compliant mechanisms do not have real joints, this concept does not play a role for design of such devices. The explicit way of treating the mechanical degrees of freedom in the articulated mechanism design is extensively discussed in [5]. In the present situation, the issue can be treated in an implicit way in which the degrees of freedom constraint is replaced by conditions related to the behavior under some auxiliary load cases.

This paper is organized as follows. Section 2 discusses, from the mechanism synthesis point of view, advantages of a truss representation compared to a representation with rigid bars and revolute joints. Section 3 describes the configuration transition problem related to the non-convexity of the equilibrium equations and a possible solution to the problem. Section 4 demonstrates that dimensional synthesis problems, in which rigid bars and revolute joints are normally used, can be actually solved in the truss representation. Section 5 formally states the simultaneous type and dimensional synthesis problem for path-generation and shows some examples and results. Section 6 finally concludes the paper and suggests possible future work.

2 Advantage of a truss representation for mechanism synthesis

Normally articulated mechanisms are modeled by rigid bars and revolute joints. Nevertheless a truss representation is adopted in the present paper as this gives a number of advantages when using optimization for design.

In the first place, a truss is a reasonable approximation to a rigid-bar articulated mechanism; one can directly interpret a truss topology as a kinematic chain of bars and joints. When designing articulated mechanisms, by setting the Young's modulus large enough one can preclude the possibility that the mechanisms rely on the elasticity to produce displacements. In other words, the functionality of articulated mechanisms in the truss representation will not change after all the truss elements are replaced with rigid bars. This is the critical difference between articulated and compliant mechanisms. Also, a truss can accommodate, by considering its geometric non-linearity, extremely large displacements without any element distortion problem. In this paper, geometric non-linearity is taken into account using the Green-Lagrange strain measure.

In the dimensional synthesis with rigid bars and revolute joints, bar lengths are typically used as design variables [1, 2]. This setting entails the necessity of assembling the mechanism in all possible configurations during the optimization process. Normally, constraint equations are used to ensure the mechanism assembly and the equations are solved independently at each optimization step. In this setting, however, non-assembly problems may occur because the optimization often suggests dimensions that lead to infeasible designs. The situation can be helped by introducing the constraint equations as an extra explicit

constraint into the optimization formulation. An alternative idea is to use truss elements and pin joints in the mechanism representation [23, 24]; there is no need for assembly of the mechanism in the truss representation. Furthermore, by adopting nodal positions as design variables non-assembly problems never occur since new nodal positions always suggest another feasible configuration; this is the approach taken in this paper and in [25].

For the simultaneous type and dimensional synthesis, that is the goal of the developments here, the truss ground-structure approach for truss topology designs (see, e.g., [9] and the references therein) turns out to provide a suitable framework. The idea of this approach is: first, give all potential elements in a predefined design domain; then, eliminate unnecessary elements according to design requirements (or, equivalently, chose necessary elements). In the articulated mechanism design problems, the truss topology is defined by a binary truss connectivity vector (a collection of normalized cross-sectional areas). However, in order to solve the problem by continuous gradient-based optimization techniques, the binary variables are relaxed to take on intermediate values between 0 and 1. During the optimization, the elasticity of the truss plays an important role for approximating intermediate mechanisms since the ground-structure contains all possible connections and is intrinsically redundant. (If we insist on using rigid bars in the ground-structure approach, we have to do something similar for approximating intermediate mechanisms; reference [21] suggests springs as a substitute for revolute joints.)

Typically only cross-sectional areas are used for ordinary truss topology designs. A critical issue is that path-generation problems require fine tuning of dimensions to realize the target path. Therefore, it is natural to include also the nodal positions as design variables. This makes it possible to keep the total complexity low and one can obtain realistic real-life mechanisms that typically have a constraint on the total element number of bars, e.g., $M^* \leq 10 (\ll 100)$. We actually gain an increased flexibility in the design space by only adding $2N$ continuous variables, where N is the number of nodes in the ground structure. The idea is that a relatively coarse ground structure covers the basic mechanism topology with a limited number of elements M^* and that the variation of the nodal positions will take charge of the element dimensions. Note that such an extended ground-structure approach that uses both cross-sectional areas and nodal positions as design variables is well-established for static problems (see, e.g., [22]). It is also used in [6] for maximum displacement problems for articulated mechanisms.

3 Configuration transition problem

For the analysis part of the path-generating mechanism design problem we have to deal with a fundamental problem pertaining to the geometry of the configuration space [16, 17]. Let us consider the two-bar link mechanism shown in Figure 1 (see also Appendix 2). Node 3 is connected to an invisible link driven by a motor that is rotating clockwise at a constant angular velocity around the center of the grey circle of radius $r = 0.0999$. As the two-bar link cannot be completely stretched out during the motion, the mechanism should stay in the elbow-up configuration while node 3 is moving on the input circle.

Throughout the paper we only analyze the kinematics of the mechanism and we do not take account of dynamics at all; also, there is no time integration for the calculation of the displacements. The displaced mechanism configuration at each step is obtained by solving static (but geometric non-linear) equilibrium equations for the given input point. This is analogous to the standard kinematic analysis [26] in which constraint equations are solved to determine the mechanism configuration.

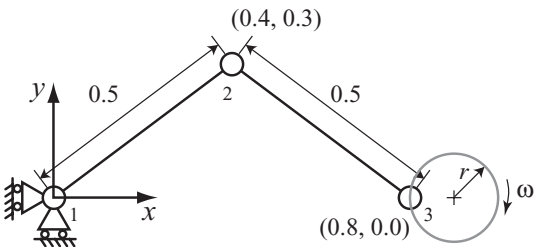
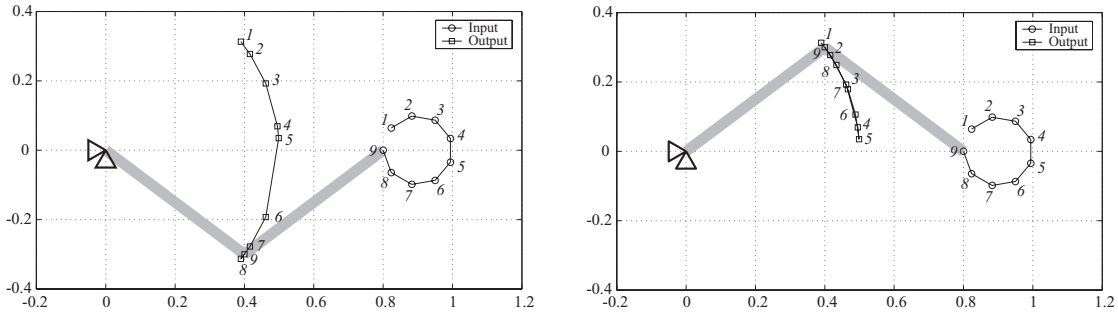


Fig. 1 Two-bar mechanism problem



(a) Newton-Raphson method (b) Levenberg-Marquardt method

Fig. 2 Path-following analysis of a two-bar mechanism

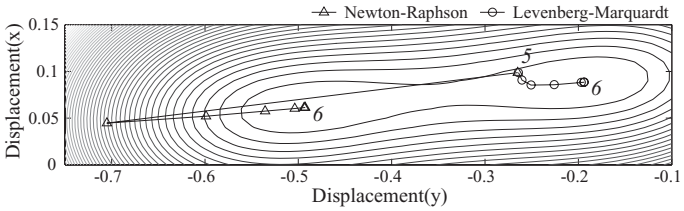


Fig. 3 Two stable points in the total potential field

Here, in order to simulate the movement by quasi-static analysis, namely kinematic analysis (see, e.g., [26]), we divide the grey input circle into 9 equally sized segments, making 9 input points; the equally distributed input points then correspond to equally discretized time steps.

We first apply an ordinary Newton-Raphson method to the analysis. In the analysis, the Young's modulus is set as $E = 10$ and all the cross-sectional areas are set as $A_e = 1$ for $e = 1, 2$. The result is shown in Figure 2(a), in which the corresponding input-output steps are numbered parallel in italics. This solution is wrong because the analysis failed to keep the elbow-up configuration and a configuration transition occurred in the sixth step.

As an alternative, the Levenberg-Marquardt method [27] can also be applied to the same analysis. This method requires an expression for the total potential energy that is here defined as follows [28][29]:

$$\Phi = \frac{1}{2}\sum_{e=1}^{M} A_e E l_e \epsilon_G^2 - F^\top D \tag{1}$$

where F is the nodal force vector and D is the displacement vector, both in the global coordinate system. Moreover, ϵ_G is the Green-Lagrange strain measure that takes geometric non-linearity into account. For a truss element this is expressed as:

$$\epsilon_G = \frac{\hat{l}_e^2 - l_e^2}{2l_e^2} \tag{2}$$

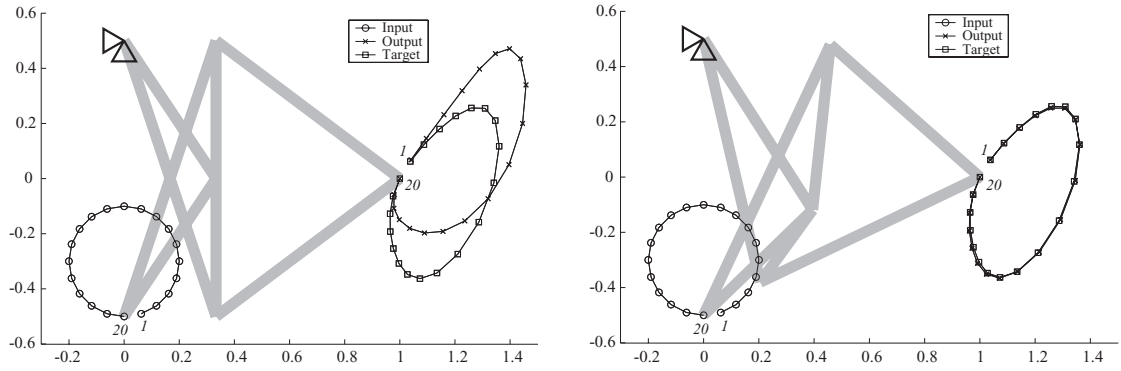
where $\hat{l}_e$ and l_e are the deformed and undeformed lengths respectively.

The basic computational steps of the Levenberg-Marquardt method are then as follows:

Step1: Solve $(K_t(D) + \mu I)\Delta D = -R(D) \qquad (R = \frac{\partial \Phi}{\partial D}; K_t = \frac{\partial R}{\partial D}; \mu > 0)$

Step2: Update μ

Step3: If $D + \Delta D$ is acceptable, then $D := D + \Delta D$



(a) Initial dimensions

(b) Final dimensions

Fig. 4 Dimensional synthesis of eight-bar mechanism

Here **Step1** expresses that the Levenberg-Marquardt step ΔD is a linear combination of the Newton step and the steepest descent step. The update scheme for μ in **Step2** is a heuristic: increase μ if $K_t(D)$ is not significantly positive definite, otherwise decrease μ. (See Appendix 1 for the detail of the algorithm.) Finally, **Step3** ensures that the Levenberg-Marquardt iterations always decrease the potential function. The combination of these three steps give the Levenberg-Marquardt method the nature of a trust region method as known in mathematical programming [30]. Figure 2(b) shows the result for the two-bar mechanism; the corresponding input-output steps are also here numbered parallel in italics. The analysis is here successful since it keeps the elbow-up configuration in all the steps.

In order to investigate why the Newton-Raphson method failed in the analysis, we need to check the iterations in the sixth step. Figure 1 plots the potential field in the sixth step and all the iterations of both the Newton-Raphson method and the Levenberg-Marquardt method. It is notable that the Newton-Raphson method takes the first step across the far basin and converges to the bottom of the far basin. In contrast, the Levenberg-Marquardt method takes a very conservative step in the steepest descent direction (due to the term μI in **Step1**) and converges to the bottom of the near basin; this corresponds to the physically realistic configuration.

4 Example of dimensional synthesis in a truss representation

This section illustrates the fairly recent idea of a truss-based dimensional synthesis for path-generation (see, e.g., [23, 24, 25]); we shall thus see that the dimensional synthesis problems, in which rigid bars and revolute joints are normally used, can actually be solved via a truss representation.

Figure 4(a) shows an example problem setting for the design of an eight-bar mechanism that can trace a prescribed target path for a given input path. The design will be achieved by (optimally) relocating the nodal positions in the initial configuration.

In order to solve this problem using the kinematic analysis described in the previous section, the input path is discretized into 20 equally distributed input points. In the analysis the Young's modulus is set as $E = 10$ and all the cross-sectional areas are set as $A_e = 1$ for $e = 1, \ldots, 8$. The path-generation problem is then stated as the minimization of the gap between the target points and the actual output points for the given input points. Note that both the input and target points are assumed to be given in the beginning and that the positions of the input and target points never change during the optimization process. In other words, the original path-generation problem is reduced to a problem of a one-to-one mapping relationship between the input and output points by point-by-point comparison between the target and output points. In this setting, therefore, time steps do not explicitly enter the formulation.

One can alternatively consider time steps in the kinematic analysis by introducing a physical model of a driver that is operated in the time domain. By doing this, a change of the input points can be viewed as the varying operating speed of the input driver (as treated in [1]). In that sense, the present setting corresponds

to the case where the operating speed (or angular velocity) of a possible input driver is *constant* and the exact target point is specified for each time step since the input and target points are fixed.

The problem can be formulated as an optimization problem that, by varying the nodal position vector $X \in \mathbb{R}^{2N}$, seeks to minimize the infinity norm of a gap vector $G = ({}^1G, {}^2G, \ldots, {}^SG)$ between the target points ${}^iD_{\text{tar}}$ $(i = 1, 2, \ldots, S)$ and the current output points ${}^iD_{\text{out}}$ $(i = 1, 2, \ldots, S)$, where the gaps are defined as ${}^iG = \|{}^iD_{\text{out}} - {}^iD_{\text{tar}}\|_2$:

GEO:

$$\begin{aligned} &\min_X : \|G\|_\infty \\ &s.t. : \; X_{\min} \le X \le X_{\max} \end{aligned}$$

This problem can, for example, be solved by using the optimization algorithm MMA [31, 32] that allows for a min-max formulation. This has proven to be a stable and efficient choice for these problems and it is a well-known tool in topology optimization. Testing other algorithms should be useful but are not part of this proof-of-concept paper. MMA requires that one provides subroutines that can calculate the objective function as well as its first derivative. The first derivative can be calculated by the adjoint technique described in [11] with a minor modification related to the use of nodal positions as design variables. Figure 4(b) shows the final dimensions of the eight-bar mechanism obtained; the application of the Levenberg-Marquardt method was here critical for the success of the calculations.

5 Simultaneous mechanism synthesis for path-generation

In the previous section we have seen how nodal positions of a truss can be used as the design variables for dimensional synthesis. An advantage of the approach is that it does not require the mechanism to be assembled, unlike the representation with rigid bars and revolute joints. Computational experience shows that the algorithm works well as long as a solution exists within the mechanism topology chosen in advance. However, if an arbitrary path is given, we are typically forced to design the topology as well as the geometry, and this is the final topic of the paper.

Figure 5 shows a truss ground-structure for a benchmark example of a path-generation problem in which we are seeking to create a mechanism that can trace a prescribed target path. The target path is depicted by the black ellipse at the output node 8 and this path should be the resulting motion for the given input path depicted by the black circle at the input node 2. The input circular path is discretized into the eight equally distributed input points, and then the corresponding target points are also located as indicated in the figure; the corresponding input-output steps are numbered parallel in italics. In the same fashion as described in the previous dimensional synthesis problem, the input motion can be realized by introducing an extra link that is driven by a rotating motor around the center of the input circle at some constant angular velocity.

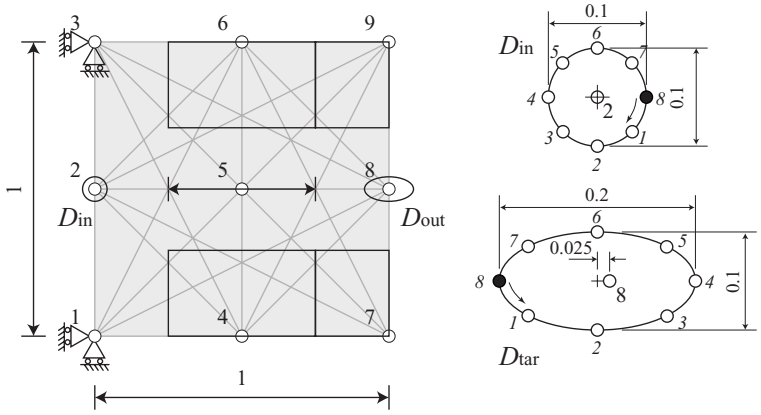


Fig. 5 Ground structure for path-generation

We will seek to generate a mechanism by picking $M^* = 6$ bars and simultaneously relocate the nodes within the bounds depicted by the solid boxes in Figure 5. We require that the resulting mechanism should be symmetric across the horizontal mid-axis. Also, in order to ensure the stability and certainty of the resulting motion, the resulting mechanism should have $f = 2$ mechanical degrees of freedom (DOF); this is defined as the number of independent inputs required to determine the positions of all links of the mechanism with respect to a global coordinate system [4]. In the current case, each step of the kinematic analysis implies that the input port is clamped by removing 2 DOF; this results in a both statically and kinematically determinate structure. Thus, for the final design to be able to move along the input path and to have unique output behavior, we must ensure that $f = 2$.

The problem can be formulated as a mixed integer optimization problem given below. As design variables the problem has the truss connectivity vector $\chi \in \mathbb{B}^M$ as well as the nodal position vector $X \in \mathbb{R}^{2N}$. The problem is formulated so as to minimize the infinity norm of a gap vector $G = ({}^1G, {}^2G, \ldots, {}^SG)$ between the target points ${}^iD_{\text{tar}}$ $(i = 1, 2, \ldots, S)$ and the actual output points ${}^iD_{\text{out}}$ $(i = 1, 2, \ldots, S)$ defined as ${}^iG = \|{}^iD_{\text{out}} - {}^iD_{\text{tar}}\|_2$:

GEOTOP 1:

$$
\begin{aligned}
&\min_{\chi, X} : \|G\|_\infty \\
&\text{s.t.} : \sum_{e=1}^{M} \chi_e = M^* \\
&\quad : \chi_e \in \{0, 1\} \\
&\quad : \ X_{\min} \le X \le X_{\max} \\
&\quad : f(\chi) = 2 \\
&\quad : \text{symmetric}
\end{aligned}
$$

Here, assuming that the mechanism has *no redundant elements* (see [5] for the detail), the DOF of a mechanism can be calculated by the following DOF equation based on the extended Maxwell's rule (in 2D) [18, 19, 20]:

$$f(\chi) = 2n(\chi) - M^* - s \tag{3}$$

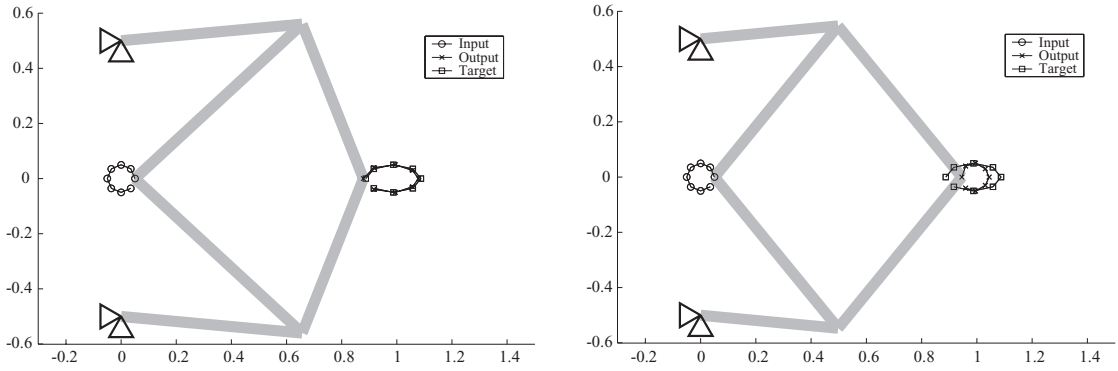
where n is the number of active nodes which have non-zero incident bars; s is the number of supports for the mechanism. In the case of $f = 2$(DOF), $M^* = 6$(bars) and $s = 4$(supports), a possible mechanism has to have $n = 6$(active nodes).

In order to solve the problem by using continuous gradient-based optimization techniques we need to relax the problem by permitting the binary variable χ_e to take on intermediate values ρ_e between 0 and 1.

As regards the DOF constraint which is well-defined for the binary variables χ, there is an explicit way of treating the constraints by relaxing the DOF equation (3) [5]. In this paper, however, by utilizing the fact that the mechanism we are going to design forms a both statically and kinematically determinate structure with the input port fixed at the given input points, we will replace the DOF constraint $f = 2$ by conditions relating to the behavior with respect to the application of two additional very small load cases at the output node 8, one horizontal and one vertical. These load cases simulate conditions in which the mechanisms are perturbed by some disturbances and the resulting mechanism should be stable enough against the disturbances when the input port is clamped. The two additional loads are applied separately, so we name three gap vectors as G_j ($j = 1$ for no load; $j = 2$ for horizontal load; $j = 3$ for vertical load) and use for the optimization a weighted gap vector $\sum_{j=1}^{3} w_j G_j$ ($w_1 = 0.98; w_2 = w_3 = 0.01$). Note that, thus, the two additional load cases do not contradict the original objective, namely path-generation under the unloaded condition. The gap vectors $G_j = ({}^1G_j, {}^2G_j, \ldots, {}^SG_j)$ between the target points ${}^iD_{\text{tar}}$ $(i = 1, 2, \ldots, S)$ and the actual output points ${}^iD_{\text{out}}$ $(i = 1, 2, \ldots, S)$ are defined as ${}^iG_j = \|{}^iD_{\text{out}} - {}^iD_{\text{tar}}\|_2$ $(j = 1, 2, 3)$.

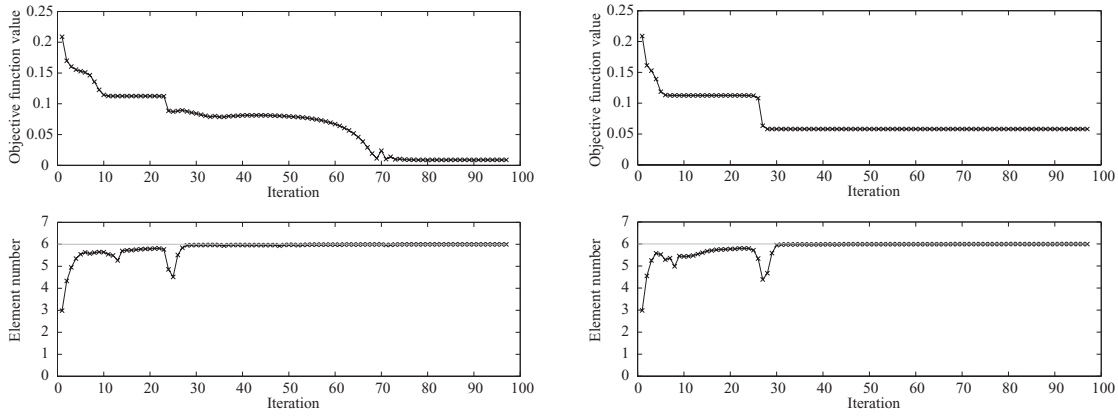
The weighted objective function was originally introduced in [11] in the continuum setting of compliant mechanism designs. The current situation with the additional load cases means that the mechanism is stable enough against disturbances when the input port is fixed. This in turn means that there should not be more than $n = 6$ active nodes, again implying $f \le 2$; computations show that when optimizing for path-generation the strive for flexibility then results in $f = 2$ in the final binary design.

The relaxed problem that we have implemented is thus:



(a) Cross-sectional areas and nodal positions (b) Only cross-sectional areas

Fig. 6 Benchmark example of path-generation problem



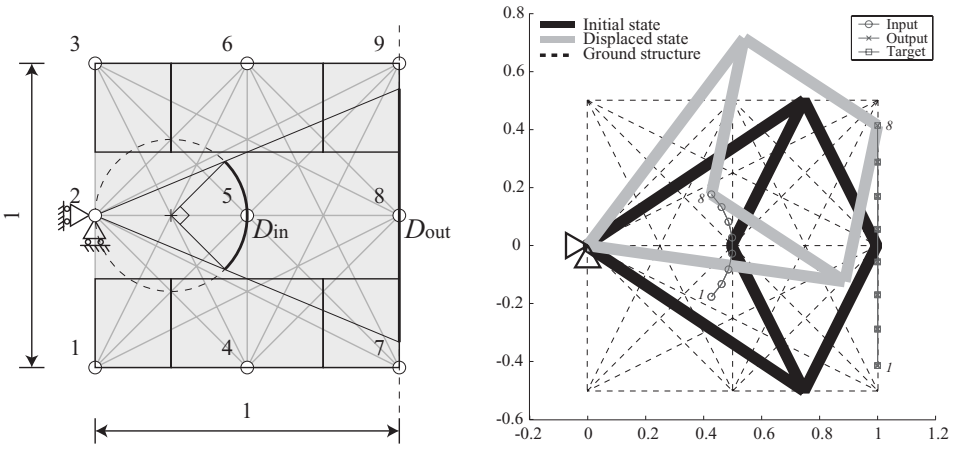
(a) Cross-sectional areas and nodal positions (b) Only cross-sectional areas

Fig. 7 Optimization histories of the benchmark example problem

GEOTOP 2:

$$\begin{aligned} \min_{\rho,X} &: \| \sum_{j=1}^{3} w_j G_j \|_\infty \\ \text{s.t.} &: \sum_{e=1}^{M} \rho_e \le M^* + \alpha \\ &: 0 < \rho_{\min} \le \rho_e \le 1 \\ &: X_{\min} \le X \le X_{\max} \\ &: \text{symmetric} \end{aligned}$$

where $\rho_{\min}$ is a positive lower bound of the design variables ρ_e (= 0.001 typically). This is introduced in order to prevent numerical difficulties such as a singular stiffness matrix and an insensitivity of the objective function to changes in the design variables around zero values. The term α compensates the effect of $\rho_{\min}$ on satisfying the bound M^* (i.e., $\alpha = (M - M^*)\rho_{\min}$). Furthermore, in order to recover the 0–1 values of χ, we will also penalize the intermediate values of ρ_e by expressing the cross-sectional area A_e of the e-th bar



(a) Ground structure

(b) Optimized design

Fig. 8 Straight-line mechanism design problem

as:

$$A_e = \rho_e^P A_{\max} \tag{4}$$

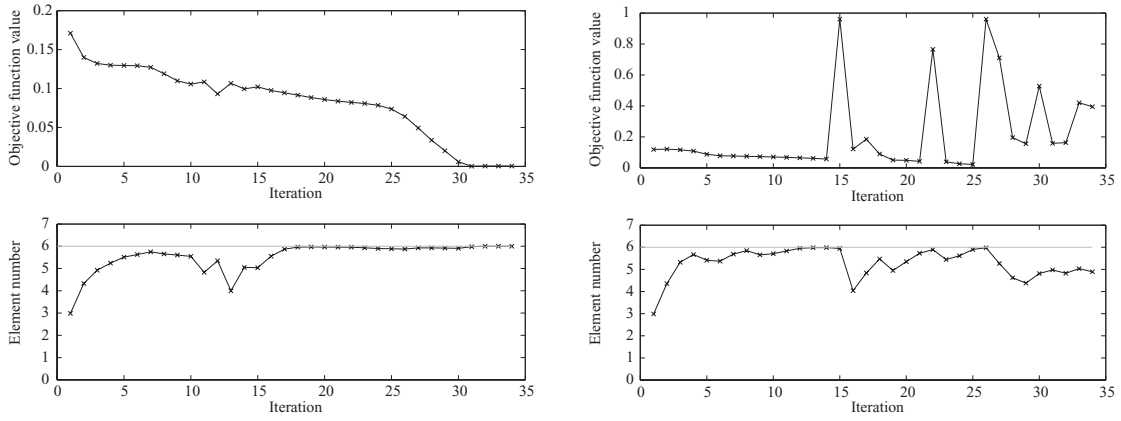
where $P > 1$ (= 5 for the present case) and where $A_{\max}$ is an upper bound (= 1 for the present case). Since we have a constraint $M^* < M$ on the total element number of bars the penalization makes the intermediate values less efficient, thus typically forcing ρ to go to the extremums for optimized designs, i.e. $\rho_{\min}$ or 1 as the structure else would be too compliant under the load perturbations. This is a typical technique used in continuum based topology optimization problems [9].

This problem can also be solved by using MMA [31, 32] in a min-max formulation. The first derivative required for MMA can be calculated via the sensitivity analysis with an adjoint technique described in, e.g., [11]. Solving the problem by using both the truss connectivity vector ρ and the nodal position vector X simultaneously can give a result as illustrated in Figure 6(a). Here the resultant mechanism can follow the target path fairly well. In contrast, however, if one solves the problem for fixed nodal positions, a mechanism as shown in Figure 6(b) can be achieved. Due to the limitation of the choice of dimensions the mechanism cannot fully follow the target path. Note that both solutions are feasible in the original problem **GEOTOP 1** since the number of active nodes is $n = 6$ for both the cases (cf. comment after Eq. (3)). Figure 7(a) and 7(b) show the optimization histories of the objective function value and the total element number constraint in the cases with and without the nodal positions variations.

Another example problem treated with the same formulation is shown in Figure 8(a); note that the input and target paths and the boundary condition have been changed. We here seek to create a six-bar mechanism that can transform the arc (depicted by the thick black arc) at the input node 5 into the straight line segment (depicted by the thick black line) at the output node 8. The input circular motion can be realized by adding an extra link that connects the input node 5 and a ground point located at the focal point of the arc.

The input arc is discretized into the eight equally distributed input points, and the corresponding target points are located along the straight line as indicated in Figure 8(b); the corresponding input-output steps are numbered parallel in italics. Unlike the previous problem setting, the input motion is not closed or periodic. In this setting, if not impossible, it might be unrealistic to associate the meaning of the equally distributed input points with the time discretization. It may be more natural to interpret that this problem is simply to design a mechanism that can map the input points to the corresponding output points in the one-to-one relationship, in accordance with classical kinematic analysis of mechanisms.

Figure 8(b) shows the resultant mechanism – the initial configuration is depicted in black and the displaced configuration in grey. It required 34 iterations for the optimization process to converge, as shown in Figure 9(a). Notice that in the final design the nodes 4 and 6 have moved to the right by 0.25. The DOF of the mechanism can be calculated with the DOF equation (3). Substituting $n = 5$(active nodes),



(a) Levenberg-Marquardt method

(b) Newton-Raphson method

Fig. 9 Optimization histories of the straight-line mechanism design design

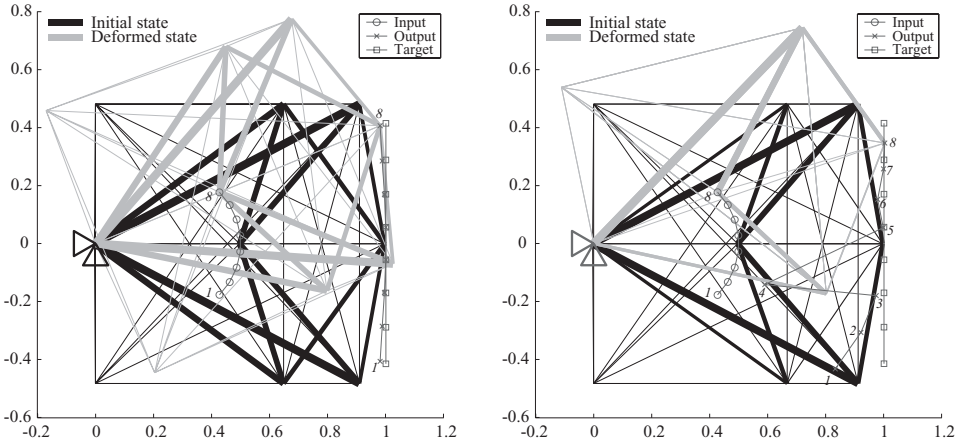
$M^* = 6$(bars) and $s = 2$(supports), the DOF is obtained as $f = 2$, and this ensures the mechanism is feasible in the original problem **GEOTOP 1**. The resulting mechanism is the straight-line mechanism [33] invented in 1864 by the French engineer Charles-Nicolas Peaucellier [34]. Here this classical design has been created by a computer algorithm for mechanism synthesis.

In contrast, if one solves the same problem by applying ordinary Newton-Raphson method instead of the Levenberg-Marquardt method, one encounters the unstable iteration history as shown in Figure 9(b). Note that the objective function value fluctuates severely and never converges, even after 200 iterations. In contrast, with the Levenberg-Marquardt method, convergence is obtained after 34 iterations.

Figure 10(a) and 10(b) show the snapshots at the 25th and 33rd iterations of the optimization using the Newton-Raphson method. In the 25th iteration, as shown in Figure 10(a), the analysis produced the smooth output path; this means no configuration transition occurred during the analysis. On the other hand, in the 33rd iteration of Figure 10(b), the output path looks non-smooth. Besides, if one compares the two configurations before and after the deformation, one notices that the two configurations do not belong to the same configuration space; the initial configuration is somehow flipped along the horizontal mid-axis after deformation.

As regards the computational aspects, the major computational data such as the problem size, the number of iterations and the CPU-time are collected in Table 1. In the table N_P denotes the number of nodal position variables; the number in the parentheses means the number after reduction of the variables considering the symmetry. d denotes the degreed of freedom in the finite element analysis sense; note that there is no symmetry in the analysis. N_C denotes the number of constraints. N_I denotes the number of total iterations and CPU denotes the CPU-time spent until convergence. The value of $\|G\|_\infty$ is the one after convergence. Benchmark 1 and 2 indicate the benchmark example for the simultaneous type and dimensional synthesis with and without nodal position variations respectively. Peaucellier 1 and 2 indicate the straight-line mechanism design problem using the Levenberg-Marquardt method and the Newton-Raphson method respectively.

With reference to the stopping criteria, the optimization is terminated either when the derivative of the objective function becomes less than 10^{-10} or when the maximum relative difference between the current and previous variables becomes less than 10^{-4}. For all the computations, the initial values of the cross-sectional areas are set to $\rho_e = M^*/2M$; also, move limits are imposed on the nodal positions of up to 20% of the bounds at each optimization step. Each problem is solved by just one run of the algorithm and all the computations are performed on an Intel Pentium 4, 2.8 GHz clock frequency and 1 Gb internal memory.



(a) Step 25 (b) Step 33

Fig. 10 The unstable behavior when using the Newton-Raphson method

Table 1 Computational data for the mechanism synthesis problems

Problem	N_P	M	M^*	d	N_C	N_I	CPU	$\|\|G\|\|_\infty$
Benchmark 1	9(5)	36(21)	6	12	1	97	7m43s	0.0088
Benchmark 2	9(5)	36(21)	6	12	1	97	7m46s	0.0581
Peaucellier 1	13(7)	36(21)	6	14	1	34	1m45s	0.0004
Peaucellier 2	13(7)	36(21)	6	14	1	-	-	-

6 Conclusion

A method has been developed for designing both the shape and topology of articulated mechanisms based on the extended truss ground-structure approach using cross-sectional areas and nodal positions as design variables. This leads to a technique for simultaneous type and dimensional synthesis. For the analysis part it is essential to control the mechanism configuration so that the mechanism remains within the same configuration space, thus stabilizing the optimization process and resulting in realistic solutions. This can be achieved by application of the Levenberg-Marquardt method.

The proposed method presupposes that the input and target points on the path and the total number of elements are given in advance. However, these parameters are also crucial for obtaining good results. It will be possible future work to include these parameters as design variables in the optimization.

Even though the gradient-based optimization seems promising for the presented examples, it has its limitations since the problem is intrinsically non-convex. Therefore, it will be necessary to combine some global optimization techniques such as branch and bound in order to generalize the method. From the global optimization point of view, the replacement of the DOF constraint with the auxiliary load cases will be less practical since the problem size is multiplied by the number of load cases; thus direct treatment of DOF constraint might be more advantageous [6]. A relaxed format of the DOF constraint may also be helpful for the continuous format of the problem [5].

Another issue for future study is if the Levenberg-Marquardt method can alleviate the problem of so-called *unstable elements* that are frequently observed in low-density regions in continuum-based compliant mechanism designs [11, 35]. This problem is caused by the ill-conditioned tangent stiffness matrix arising through the use of a ground-structure like design representation.

Acknowledgements

The author is grateful to K. Svanberg at KTH for permission to use his software implementation of the MMA algorithm. Also, the author would like to thank Martin P. Bendsøe and O. Sigmund at DTU for many fruitful discussions on this subject and valuable comments on the early manuscripts.

References

[1] Minnaar RJ, Tortorelli DA, Snyman JA. On nonassembly in the optimal dimensional synthesis of planar mechanisms. *Structural and Multidisciplinary Optimization* 2001; **21(5)**:345–354.

[2] Hansen JM. Synthesis of mechanisms using time-varying dimensions. *Multibody System Dynamics* 2002; **7(1)**:127–144.

[3] Erdman AG. Computer-Aided Mechanism Design - Now and the Future, *Journal of Mechanical Design* 1995, **117**:93–100.

[4] Erdman AG, Sandor GN. *Mechanism Design – Analysis and Synthesis.* Second Edition, Vol. 1, Prentice Hall: New Jersey, 1991.

[5] Kawamoto A, Bendsøe MP, Sigmund O. Articulated mechanism design with a degree of freedom constraint. *International Journal for Numerical Methods in Engineering*; *To appear*.

[6] Stolpe M, Kawamoto A. Design of planar articulated mechanisms using branch and bound. *Mathematical Programming, Series B*; *To appear*.

[7] Tischler CR, Samuel AE, Hunt KH. Kinematic chains for robot hands–I. orderly number-synthesis. *Mechanism and Machine Theory* 1995; **30(8)**:1193–1215.

[8] Tischler CR, Samuel AE, Hunt KH. Kinematic chains for robot hands–II. kinematic constraints, classification, connectivity, and actuation. *Mechanism and Machine Theory* 1995; **30(8)**:1217–1239.

[9] Bendsøe MP, Sigmund O. *Topology Optimization – Theory, Methods, and Applications.* Springer Verlag: Berlin Heidelberg, 2003.

[10] Sigmund O. On the design of compliant mechanisms using topology optimization. *Mechanics of Structures and Machines* 1997; **25(4)**:493–524.

[11] Pedersen CBW, Buhl T, Sigmund O. Topology synthesis of large-displacement compliant mechanisms. *International Journal for Numerical Methods in Engineering* 2001; **50(12)**:2683–2705.

[12] Bruns TE, Tortorelli DA. Topology optimization of non-linear elastic structures and compliant mechanisms. *Computer Methods in Applied Mechanics and Engineering* 2001; **190(26-27)**:3443–3459.

[13] Ananthasuresh GK, Kota S, Kikuchi N. Strategies for systematic synthesis of compliant MEMS. *Dynamic Systems and Control* 1994; **2**:677–686.

[14] Sigmund O. Some inverse problems in topology design of materials and mechanisms. In *IUTAM Symposium on Optimization of Mechanism Systems*, Bestle D, Schielen W (eds). Kluwer: Dordrecht, 1996; 277–284.

[15] Frecker MI, Ananthasuresh GK, Nishiwaki S, Kikuchi N, Kota S. Topological synthesis of compliant mechanisms using multi-criteria optimization. *Journal of Mechanical Design* 1997; **119(2)**:238–245.

[16] Hansen VL. *Braids and Covering* Cambridge University Press, 1989.

[17] Eldar D. Planar machines' web site: an invitation to topology. Master's thesis, Department of Mathematics, University of Toronto, 1999.
`http://www.math.toronto.edu/~drorbn/People/Eldar/thesis/index.html`.

[18] Calladine CR. Buckminster Fuller's tensegrity structures and Clerk Maxwell's rules for the construction of stiff frame. *International Journal of Solids and Structures* 1978; **14**:161–172.

[19] Pellegrino S., Calladine CR. Matrix analysis of statically and kinematically indeterminate frameworks. *International Journal of Solids and Structures* 1986; **22(4)**:409–428.

[20] Fowler PW, Guest SD. A symmetry extension of Maxwell's rule for rigidity of frames. *International Journal of Solids and Structures* 2000; **37(12)**:1793-1804.

[21] Felter C, Sigmund O. Topology Optimization of Rigid Body Mechanism. Master's thesis, Department of Mechanical Engineering, Technical University of Denmark, 2003.

[22] Kočvara M, Zowe J. How to optimize mechanical structures simultaneously with respect to topology and geometry. *Proceedings of the first World Congress of Structural and Multidisciplinary Optimization(edited by N. Olhoff and G.I.N. Rozvany)* 1995; **23**:135–140.

[23] Aviles R., Ajuria MBG., Vallejo J., Hernandez A. A procedure for the optimal synthesis of planar mechanisms based on non-linear position problems *International Journal for Numerical Methods in Engineering* 1997; **40(8)**:1505–1524.

[24] Aviles R., Vallejo J., Ajuria G., Agirrebeitia J. Second-order methods for the optimum synthesis of multibody systems *Structural and Multidisciplinary Optimization* 2000; **19(3)**:192–203.

[25] Buhl T. Design of non-linear mechanisms – topology and shape optimization Dissertation, Department of Mechanical Engineering, Technical University of Denmark, 2002.

[26] Nikravesh PE. *Computer-Aided Analysis of Mechanical Systems*. Prentice Hall, 1988.

[27] Nielsen HB. Damping parameter in Marquardt's method. *Technical Report IMM-REP-1999-05*, 1999. `http://www.imm.dtu.dk/~hbn/publ/`.

[28] Kočvara M. On the modelling and solving of the truss design problem with global stability constraints *Structural and Multidisciplinary Optimization* 2002; **23**:189–203.

[29] Crisfield MA. *Non-linear Finite Element Analysis of Solid and Structure*. Vol. 1, John Wiley & Sons, 1997.

[30] Conn AR, Gould NIM, Toint PL, *Trust-Region Methods* SIAM, 2000.

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

[32] Svanberg K. A class of globally convergent optimization methods based on conservative convex separable approximation *SIAM Journal on Optimization* 2002; **12(2)**:555–573.

[33] Wunderlich W. *Ebene Kinematik.* Bibliographisches Institut Mannheim/Wien/Zürich, 1968; 131-135.

[34] Moon FC. Franz Reuleaux: Contributions to 19th C. Kinematics and Theory of Machines *Applied Mechanics Reviews* 2003; **56(2)**:261–285.

[35] Bruns TE, Tortorelli DA, An element removal and reintroduction strategy for the topology optimization of structures and compliant mechanisms. *International Journal for Numerical Methods in Engineering* 2003; **57(10)**:1413–1430.

Appendix 1: Application of Levenberg-Marquardt method to non-linear truss analysis

In order to find the equilibrium, we need to minimize the total potential energy Φ defined in Eq. (1). The truncated Taylor expansion of the potential function at the current iterate displacement D is:

$$\Phi(D + \Delta D) \simeq Q(\Delta D) \tag{5}$$

$$= \Phi(D) + \Delta D\Phi' + \frac{1}{2}\Delta D^\top \Phi'' \Delta D \tag{6}$$

$$= \Phi(D) + \Delta DR + \frac{1}{2}\Delta D^\top K_t \Delta D \tag{7}$$

where $\Phi' = \partial\Phi/\partial D (= R)$ and $\Phi'' = \partial^2\Phi/\partial D^2 (= K_t)$. We then define the *gain factor* g for the energy Φ as

$$g := \frac{\Phi(D) - \Phi(D + \Delta D)}{Q(0) - Q(\Delta D)} \tag{8}$$

When g is close to one, we assume that the quadratic approximation is good in the neighborhood of the current iterate D. All the steps of the algorithm are then as follows:

Algorithm: Levenberg-Marquardt method [27]
begin
  $D := D_0$; $\mu := \mu_0$; found:=false; $k := 0$;
  repeat
    while $K_t + \mu I$ not positive definite
      $\mu := 2\mu$
    Solve $(K_t + \mu I)\Delta D = -R$
    Compute gain factor g by Eq.(8)
    If $g > \delta$ (typically $\delta \simeq 10^{-3}$)
      $D := D + \Delta D$
      $\mu := \mu \max\{\frac{1}{3}, 1 - (2g - 1)^3\}$
    else
      $\mu := 2\mu$
    $k := k + 1$; Some stopping criteria
  until found=true or $k > k_{\max}$
end

The physical meaning of the term μI is to add extra horizontal and vertical springs that connects all the nodes to the ground. If the node is isolated, namely no bars are incident with the node, the nodes is pinned by the springs; thus the structure becomes more stable.

Appendix 2: Configuration space of two-bar link mechanism

If we have no restrictions for the two-bar mechanism of Figure 1, node 3 can reach any point on the unit disc depicted by the solid line in Figure 11. In this case, a valid representation for the configuration space becomes a *torus*. Here, by *valid* we mean that the representation is a continuous and one-to-one correspondence mapping. However, if we remove the boundary and the origin of the disk, the mechanism has no way to transit from one (elbow-up or elbow-down) configuration to the other because the mechanism needs to be either completely stretched out (the situation corresponds to the boundary of the disc) or completely contracted (the origin of the disc) in the transition. In that case, the configuration space is completely divided into two disjoint spaces as shown in Figure 12. The grey circle of node 3 contains neither the origin nor the boundary points of the disc; therefore the mechanism should stay in the elbow-up configuration when it starts with the elbow-up configuration.

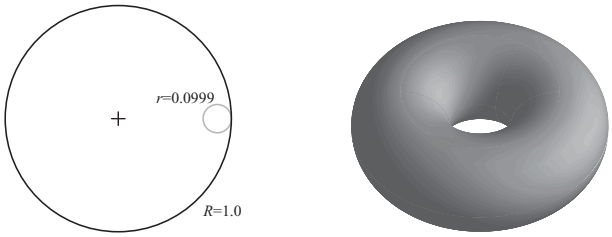


Fig. 11 Configuration space of two-bar mechanism: no restrictions

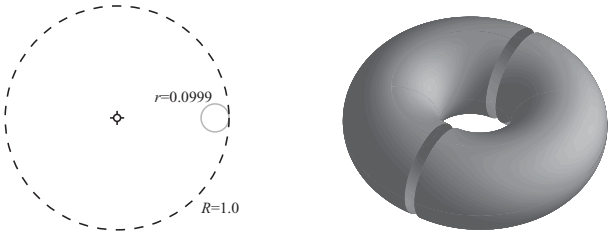


Fig. 12 Configuration space of two-bar mechanism: w/o origin and boundary