

# Complex-shaped beam element and graph-based optimization of compliant mechanisms

M. Sauter · G. Kress · M. Giger · P. Ermanni

Received: 4 February 2007 / Revised: 24 July 2007 / Accepted: 22 August 2007 / Published online: 16 October 2007  
© Springer-Verlag 2007

**Abstract** Compliant mechanisms are designed to be intentionally flexible, providing hingeless mechanisms. This work contributes a complex-shaped beam element formulation in conjunction with the ground structure approach. We identify compliant mechanism design solutions by using evolutionary topology optimization and increase flexibility by using a parametrization concept based on graph theory. The new operators for evolutionary optimization are also explained and sample problems are used to address the question of how our contribution increases design solutions space.

**Keywords** Compliant mechanisms · Curved variable thickness beam · Evolutionary algorithms · Graph

## 1 Introduction

Compliant mechanisms are receiving much attention because of their potential for mass reduction and intrinsic multifunctionality. They are designed to be intentionally flexible, providing hingeless mechanisms (Ananthasuresh and Kota 1995). Topology, shape, or sizing optimization techniques help in finding best design solutions for compliant mechanism problems. The optimization methods can be categorized into two general areas, pseudo-rigid body methods and

continuum-based methods. The pseudo-rigid body methods are kinematic type methods that start with a known rigid-link mechanism and convert it to a compliant mechanism (Howell 2001).

Continuum-based methods, such as the homogenization method (Bendsoe and Kikuchi 1988), were originally developed for finding, for a given amount of material, structure topologies for maximum stiffness. They divide the geometric design space into a number of cells that are often identical with the elements of a finite-element mesh (Suzuki and Kikuchi 1991; Nishiwaki et al. 1998; Yin and Ananthasuresh 2001). The topology is given by the state of such cells, which can be either filled with material or void. Thus, finding the best solution implies an optimization problem with discrete parameters. Because of the astronomically high number of possible combinations, it has proven to be very efficient to introduce a continuous density distribution for each cell, transforming the originally discrete problem into a smooth one and making it accessible to efficient mathematical programming solution techniques. Moreover, the minimum compliance problem has been proven to be convex. A classical example for the compliant mechanism synthesis using the homogenization method is the modified 99 line topology optimization code by Sigmund (Bendsoe and Sigmund 2003; Sigmund 2001) for a force inverter. In contrast to the minimum compliance optimization, the goal is to maximize the displacement, the force, or the work at the output. By specifying different values for the spring constant at the inputs and outputs, the displacement and the stiffness of the mechanism can be controlled. Rahmatalla and Swan (2005) introduced two different sets of springs. The first is an artificial spring set of relative large stiff-

M. Sauter (✉) · G. Kress · M. Giger · P. Ermanni  
ETH Zurich, Centre of Structure Technologies,  
Leonhardstrasse 27, 8092 Zurich, Switzerland  
e-mail: sauterm@ethz.ch

ness that prevents quasi-hinge solutions. The second is only attached to the output with smaller stiffness that represents the resistance of the workpiece as it is manipulated by the mechanism. Yin and Ananthasuresh (2003) applied two different methods for a design of distributed compliant mechanisms. The first method penalizes high-stress regions. The second uses local, relative rotations and restrains it by using an objective function that makes the local deformation uniform throughout the structures. However, only the second method is successful in giving distributed compliant designs.

A primary issue in the synthesis of a distributed compliance mechanism is the efficient transfer of energy from input to output, while providing the desired mechanical traits such as *mechanical* or *geometric advantage* (Kota et al. 2001).

Another way for the representation of the design is the use of ground structures such as beam or truss elements. Frecker et al. (1998), Saxena and Ananthasuresh (2000), and others have adapted the frame element-based ground structure approach to the compliant mechanisms optimization. Joo et al. (2000) performed studies on synthesis of compliant mechanisms using geometrically linear and nonlinear beam elements (Joo and Kota 2004) for ground structures.

The results of an optimization are heavily influenced by the optimization algorithm. The fast gradient-based solution methods generally only identify local minima of non-convex problems, a problem overcome by evolutionary algorithms. Although these typically require a much higher number of function evaluations, multi-objective problems are set without significant additional effort. Genetic algorithms, which belong to the evolutionary algorithms, were first introduced by Holland and his colleagues at the University of Michigan (Goldberg 1989) and, since then, have found application in many areas of engineering optimization. These algorithms are searching techniques based on the natural forces of evolution, forces that join the process of natural selection with a random information exchange among the designs and mutations of genes that represent the designs. They are robust, computationally simple yet capable in their search abilities, and do not require derivatives.

In the past few years, the compliant mechanism design by genetic algorithms has been given increasing attention in the scientific community. Parsons and Canfield (2002) explored genetic algorithms for the multi-criteria topology optimization of compliant structures. Saxena (2005) presented a procedure to synthesize compliant mechanisms for a prescribed nonlinear output path. Lu and Kota (2003) utilized a load

path representation method to efficiently exclude the invalid topologies (disconnected structures) from the genetic algorithm solution space to avoid useless function evaluations. Zhou and Ting (2005) introduced the spanning-tree theory for the topological synthesis of compliant structures. The spanning-tree theory is based on the graph theory where a graph consists of vertices and edges. A valid topology is regarded as a network connecting input, output, support, and intermediate nodes (vertices), which contains at least one spanning-tree among the introduced nodes. Akhtar et al. (2002) combined the graph theory, genetic algorithms, and Bezier curves representation. The Bezier curves represent the underlying topology of the structures. In a second step, the curves are mapped into a continuum finite element model (FEM).

This paper introduces an approach to solving compliant-mechanism problems that combines evolutionary optimization techniques where the use of graph theory and a complex-shaped two-dimensional beam element (Kress et al. 2006) can increase both numerical efficiency and solution space.

The beam element is curved and of variable thickness so that a localized compliant region can be placed within one beam element. This reduces the number of finite beam elements and, with it, the computational effort for one function evaluation.

The ground-structure topology is represented by a mathematical graph whose edges and vertices fall together with the beams and their nodes, respectively (West 2001). All genetic operators, i.e., mutation and crossover, directly apply on the graph representation (vertices and edges). This parametrization concept allows the optimization of topology, geometry, and sizing of the beam structures at the same time. Besides this, the representation can handle individuals of different size.

The graph-theory method used was originally developed by Giger and Ermanni (2006) for link elements. In this work, the complex-shaped beam element was integrated into the program code, and the method was extended to handle the design parameters of the complex-shaped beam element. This implies a set of modified and new genetic operators, new control routines, and fitness functions.

Discrete and multi-criteria optimization sample problems serve to demonstrate the advantages of our new approach.

Section 2 of this paper summarizes the parametrization of the complex-shaped beam element, Section 3 discusses the graph genotype of compliant mechanism optimization, and Section 4 deals with the genetic operators for the evolutionary optimization. Section 6 is

concerned with the implementation of the optimization. Section 7 explains the inverter and the parallel gripper optimization sample problems and presents all results as well as an efficiency discussion.

### 2 Curved, variable thickness beam

The structural element beam is very useful for the representation of compliant mechanism design. Combining beams with high compliance and high stiffness can model compliant mechanism structures, but the necessary number of elements can be quite high.

The development of a beam element with high shape complexity, such variable thickness and curved centerline shape, was motivated by simultaneously increasing the design flexibility and reducing the number of elements for a complex topology (Kress et al. 2006). Additionally, this beam is able to place compliant regions within it. The two-dimensional beam element can be extended to three-dimensional problems without significant effort.

Figure 1 illustrates the shape complexity and how it depends on the parameters used in this work.

Many compliant mechanism design problems require finding a best topology as well as an optimized shape. Often, the two aspects are separated. A first method is used for finding the topology, and a second-step method refines the result by obtaining the best shape solution for the fixed topology. The shape flexibility of this element opens up the possibility to combine the topology and thickness distribution optimization within one computational process.

We use Castigliano’s (1875) theorem for connecting the set of beam parameters with the beam’s structural properties because it provides a good compromise between accuracy and numerical costs for linear elasticity problems (Kress et al. 2006; Lobontiu and Garcia 2003).

#### 2.1 Parametrization

For the curved centerline, the Hermite curve representation

$$\begin{aligned} x(\sigma) &= a_0 + a_1\sigma + a_2\sigma^2 + a_3\sigma^3 \\ y(\sigma) &= b_0 + b_1\sigma + b_2\sigma^2 + b_3\sigma^3 \end{aligned} \tag{1}$$

is chosen, where  $\sigma$  is in the range from  $-1$  to  $1$ .

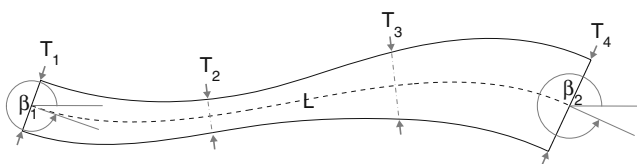


Fig. 1 Curved, variable thickness beam

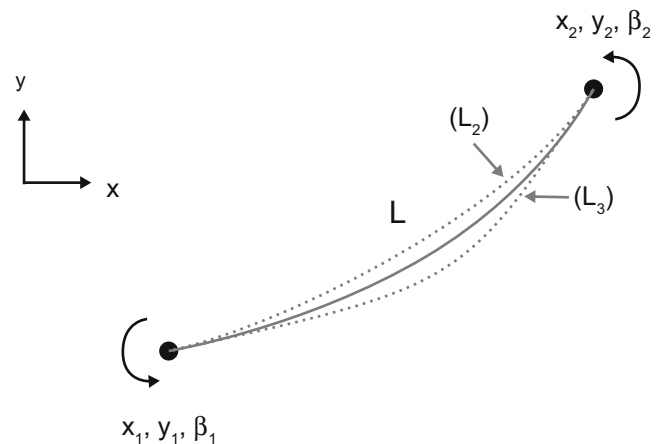


Fig. 2 Centerline defined by start- and endpoint parameters and factor  $L$

The coefficients of these two equations depend on the coordinates of the nodes with which the element is connected, the chosen angle at the nodes, and the factor  $L$ .

$L$  influences the line length of the curved centerlines (see Fig. 2).

$$\begin{aligned} \frac{dx}{d\sigma}(\sigma = -1) &= a_1 - 2a_2 + 3a_3 = L \cos(\beta_1), \\ \frac{dy}{d\sigma}(\sigma = -1) &= b_1 - 2b_2 + 3b_3 = L \sin(\beta_1), \\ \frac{dy}{dx}(\sigma = -1) &= \frac{dy(\sigma = -1)/d\sigma}{dx(\sigma = -1)/d\sigma} = \tan(\beta_1). \end{aligned} \tag{2}$$

The coefficients of the cubic thickness distribution

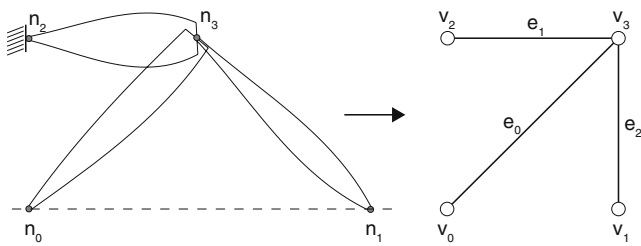
$$t(\sigma) = t_0 + t_1\sigma + t_2\sigma^2 + t_3\sigma^3 \tag{3}$$

are defined by thickness values at locations along the beam centerline as Fig. 1 illustrates.

The parameter set for describing one complex shaped beam element includes the values of

- Four end-point coordinates,
- Two centerline run-out angles at the end points,
- One factor  $L$ , and
- Four thicknesses at four points along the centerline.

The choice of material, if desired for multi-material optimization, adds a 12th parameter to the set. The shape complexity requires six more parameters than would be necessary if simple beam elements were used.



**Fig. 3** Graph representation of the beam structures

### 3 The graph genotype

#### 3.1 Graph topology representation

In this work, the complete topology is represented by a graph, wherein each vertex corresponds to a node and each beam is described by an edge (Fig. 3). In other words, instead of holding the information of the beam structure in a conventional one-dimensional genotype, the graph itself is considered as the genotype, and special operators are directly applied on it.

A graph is an abstract mathematical model; it is an ordered pair  $(V, E)$ , where  $V$  is a finite set called vertex set and  $E$  is a binary relation on  $V$  called edge set. Elements of  $V$  are called *vertices*, elements of  $E$  *edges*.

Some notations that are relevant for the study presented here are briefly explained in the following.

#### Graph:

- An edge of an *undirected* graph is an unordered pair  $(u, v)$  with  $u, v \in V$ . This means that  $e(u, v)$  and  $e(v, u)$  represent the same edge.
- A graph is said to be *labeled* if its vertices are distinguished from one another by labels like  $v_1, v_2, \dots, v_n$ .
- The *order* of the graph  $G(V, E)$  denotes the number of vertices.
- The *size* of the graph  $G(V, E)$  is determined by the number of edges.

#### Vertex:

- An *isolated* vertex is not connected to any other vertex.
- Two vertices are *adjacent* to each other if there is an edge with both vertices as endpoints.

#### Edge:

- Two edges are *parallel* if they have the same endpoints.
- An edge is called a *loop* if the two endpoints are the same.

#### Connected component:

- A *path* is a sequence of vertices where each vertex is connected by an edge to the subsequent vertex in the *path*.
- A vertex is *reachable* from another vertex if a path exists from one to the other vertex.
- A *connected component* is a group of vertices in an undirected graph that are all reachable from one another.

In this work, an *undirected, labeled* graph topology representation with a predefined *order* and a variable *size* is applied. *Loops* and *parallel* edges are not allowed.

Publications in the field of compliant mechanism optimization often rely on beam ground structure approaches with fixed number of nodes and beams (Frecker et al. 1998; Joo et al. 2000; Lu and Kota 2002). To the beam, a weight factor is assigned describing a cross-sectional dimension. A zero weight factor of an element means that there is no structural connection between the respective element nodes. This work is assuming a fixed number of vertices, which can either be connected or not, and a variable number of edges. The graph representation overcomes the ordinary restriction of having constant length genotype. Thus, the design flexibility of the optimization is increased by the independence from the predefined number of edges. Thereby, the graph representation allows an easy modification of the topology by adding or removing edges. Furthermore, the geometry, which is defined by the coordinates of the moveable nodes, can be easily modified. The nodes themselves are represented by labeled vertices and their related coordinates. The optimization process is simplified by the fact that the edges refer to the labels of the vertices (see Tables 1 and 2). Only the properties of the vertices have to be changed, and thereby, the edge parameters remain unchanged.

In addition, the graph theory provides a few very useful and fast routines such as the connected-

**Table 1** Edge/Beam properties

Edge	Label <sub>1</sub>	Label <sub>2</sub>	$\beta_1$	$\beta_2$	$T_1$	$T_2$	$T_3$	$T_4$	$L$	Material	Status
$e_0$	0	3	0.0	4.3	1.0	7.7	10.7	8.6	2.1	1	2
$e_1$	2	3	-1.2	3.2	1.0	9.9	15.8	9.2	2.0	1	2
$e_2$	3	1	0.4	0.4	1.0	5.4	8.5	1.0	2.1	1	2

**Table 2** Vertex/node properties

Vertex	Label	x	y	Moveable
$v_0$	0	0.0	0.0	False
$v_1$	1	100.0	0.0	False
$v_2$	2	0.0	50.0	False
$v_3$	3	49.2	49.9	True

component algorithm that is based on the depth-first search algorithm. For further details, refer to Siek et al. (2002). This algorithm is very useful for determining whether all clamped or loaded vertices are connected to each other.

According to Section 2.1, eight parameters and two nodes are needed to define one curved, variable thickness beam. The parameters are attached to edges as properties and the coordinates of the nodes to the vertices, respectively.

Table 1 contains the edges and their properties, namely two labels of the vertices, two angles, four thicknesses, the parameter  $L$ , and the material. Table 2 displays the vertices and their properties like label of vertices and  $x$ - and  $y$ -coordinates.

In addition, two parameters are introduced.

- A boolean *moveable* defines whether or not node coordinates may change.
- An unsigned number attached to the edge elements describes the *status* of the edge/beam.

Three settings of the status are possible:

- 0 Endpoints and shape (the four thicknesses parameter) are fixed.
- 1 Only endpoints are fixed.
- 2 Endpoints and shape can be modified.

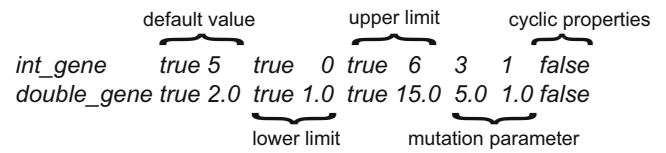
Consequently, edges with status 0 or 1 must not be removed or added from/to the graph and from/to the structure during the optimization.

### 3.2 The universal gene concept

In Section 3.1, the graph representation with element properties was introduced. In this section, the graph representation is extended to the graph genotype that represents the combination of the graph representation and the eoUniGene concept introduced by Koenig (2004).

The basic idea of the universal gene concept is to hold not only the information, like thickness or coordinate, but also its mutation parameters in each gene.

This implementation scheme allows to individually adapt the gene and its mutation properties to the optimization problem. In addition, the procedure to check



**Fig. 4** Example for *int\_gene* and *double\_gene*

whether the parameters are out of range is very easy to handle. The universal gene concept provides a variety of gene types that can be combined to a heterogeneous list of properties. Although various gene types such as *boolean-gene* and *string-list-gene* are available, only *int-gene* and *double-gene* are integrated into the program code, see Fig. 4. For example, the vertex labels are represented by an integer number, the so-called *int-genes*. The first boolean denotes whether the gene is active with 5 as default value. The boolean of the following two pairs specifies whether the lower or upper limits with value 0 or 6, respectively, are activated. Furthermore, the mutation parameters  $\epsilon = 3$  and  $\sigma = 1$  are specified.  $\epsilon$  denotes the range for uniform mutation, if the lower and upper limits are not declared (*unbounded gene*), and  $\sigma$  defines the standard deviation to be used for Gaussian mutation. Equation (4) displays the Gaussian probability distribution.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right), \tag{4}$$

whereas  $f(x)$  measures the probability, that mutations to  $x$  occurs, if  $\mu$  is equal to the current value.

The last boolean indicates that the respective gene does not have cyclic properties. Examples for *double-genes* are the vertex/node coordinates and the thickness properties of edges/beams.

In Fig. 5, a sample genotype containing two lists of genes (Fig. 4) is given. The first list represents the vertices and their properties and the second list the edges and their properties, respectively.

## 4 Genetic graph operators

All information necessary to the genetic operators is contained in the eoUniGene representation. Operators can be applied to the *individuals* within a *population* with a predefined probability, whereas an *individual* denotes one solution of a population and the *population* itself represents a set of solutions of one generation. The  $i$ -th *generation* is equivalent to  $i$ -th optimization step.

Operators can be classified in mutation and crossover operators. Mutation operators change the properties of the parameters with a defined probability

```

2 vertices 0 double_gene true 0 true 0 true 1 0.1 0.1 0 , double_gene true 0
true 0 true 1 0.1 0.1 0 , double_gene false 0 false 0 false 0 0.1 0.1 0 , 0 1
double_gene true 0.492355 true 0 true 1 0.1 0.1 0 , double_gene true 0.999651
true 0 true 1 0.1 0.1 0 , double_gene false 0 false 0 false 0 0.1 0.1 0 , 2

1 edges int_gene true 0 true 0 true 17 1 3 1 , int_gene true 1 true 0 true 17 1 3
1 , double_gene true 0.019664 true -30 true 30 1 1 0 , double_gene true
4.92213 true -30 true 30 1 1 0 , double_gene true 1.00038 true 1 true 20 1 1 0 ,
double_gene true 7.66708 true 1 true 20 1 1 0 , double_gene true 10.6811 true
1 true 20 1 1 0 , double_gene true 8.58696 true 1 true 20 1 1 0 , double_gene
false 0 false 0 false 0 0 0 0 , int_gene false 0 false 0 false 0 0 0 0 , 2

```

**Fig. 5** Sample variable length graph genotype

independently of the other individuals of the generation. The aim of crossover operators is to recombine the properties of two selected parent individuals in the hope of generating even better offsprings. Crossover operators increase the convergence if the individuals differ from one another.

Mutation and crossover operators can be classified in geometric, topology, or sizing operators. For further details, refer to Giger and Ermanni (2006). Only operators introduced for the curved, variable thickness beam are described. The geometry usually depends on the position of the connected nodes. Using the curved, variable thickness beam, the curvature of the centerline also influences the geometry. Anyway new, additional geometric operators are not needed.

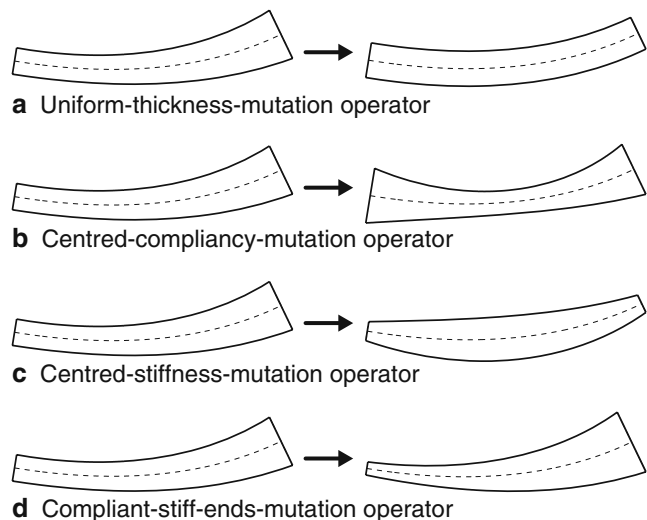
To increase the efficiency of the topology optimization, operators who split one curved, variable thickness beam in two beams and who merge two beams were implemented. For splitting operators, a minimal length of the beam was prescribed to prevent too many and unfeasible beams.

Sizing operators for conventional beams only affect one property, i.e., the cross-sectional area. But the curved, variable thickness beam includes four thickness parameter, which can individually be changed or can be modified in the same step by operators. The latter ones seek to increase the functionality of the compliant mechanism.

Most of the optimization problems include either minimization of the mass or mass restrictions within the fitness function. As illustrated in Fig. 6, these sizing operators are only influencing the shape of the beam, thus the functionality of the structure, and not its mass. In other words, these operators improve the fitness by increasing functionality under the constraint that the mass of the beam, respectively, mass influenced addend of the fitness function, remains constant.

## 5 Control routines

The control routines guarantee that only legal design solutions are evaluated. Under illegal design solutions,



**Fig. 6** Curved, variable thickness sizing operators

we understand that the parts of the domain boundary, where nonzero forces and prescribed displacements are described, are not connected so that the mechanism is not realized and/or the stiffness matrix remains singular. In addition, design solutions not complying with mechanical modeling assumptions must be rejected. We distinguish two types of control routines. The first one regards the FEM analysis and checks whether the topology connects the above-mentioned parts of the boundary and whether all beam elements are sufficiently long with respect to their thickness to comply with beam theory assumptions. The second one regards other aspects of the topology. The here considered two-dimensional design solutions do not permit crossing of beam centerlines or that beams reach across certain regions such as those between grippers. Furthermore, minimum and maximum numbers of members can be defined.

## 6 Implementation

The code is written in C++ and includes four powerful libraries: the Evolving Objects (EO),<sup>1</sup> Boost Graph Library (BGL),<sup>2</sup> eoUniGene concept, and the FELyX<sup>3</sup> FE Code.

Figure 7 displays the optimization loop from the initialization to the solution.

The EO library is providing the basic functionality for the evolutionary optimization and controlling of

<sup>1</sup><http://eodev.sourceforge.net>.

<sup>2</sup><http://boost.org>.

<sup>3</sup><http://felyx.sourceforge.net>.

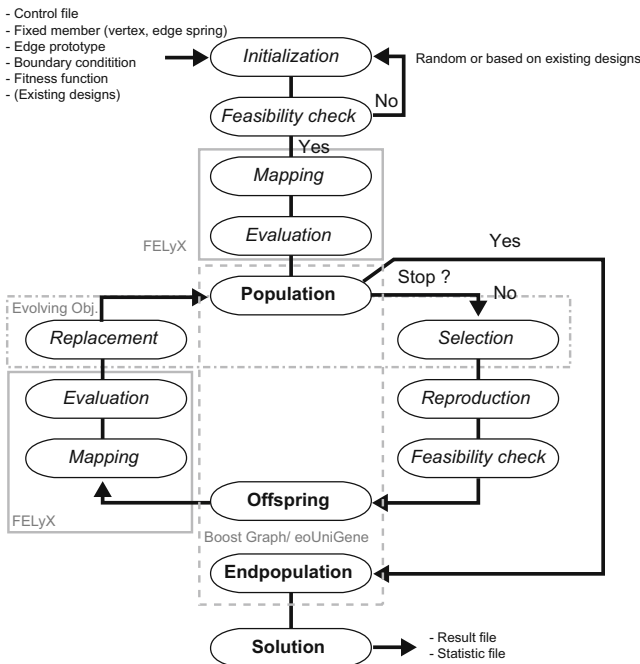


Fig. 7 Evolutionary optimization loop

the actual program run. This includes the initialization of the different objects, ensuring the whole data flow between different populations, calling evaluation modules, and managing storage of any kind of data. It also provides the implementations for selection and replacement strategy. The *selection* is responsible for the identification of parent individuals mostly based on fitness criteria. The *replacement* is the choice of the individuals, which will be part of the next generation. This mechanism is based on the fitness or on other

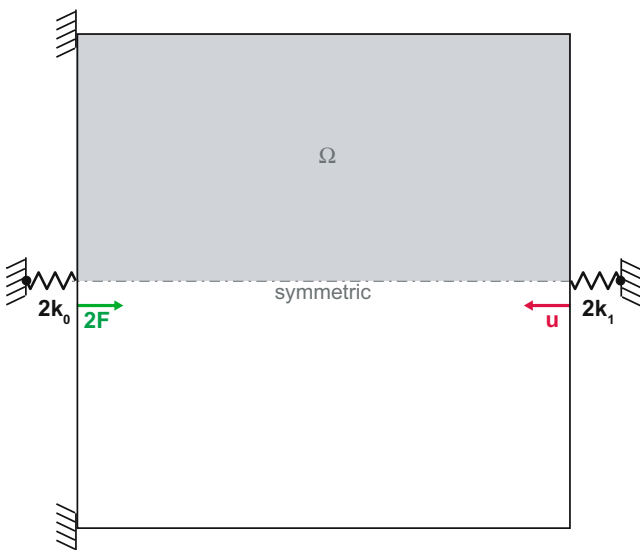


Fig. 8 Inverter optimization setup

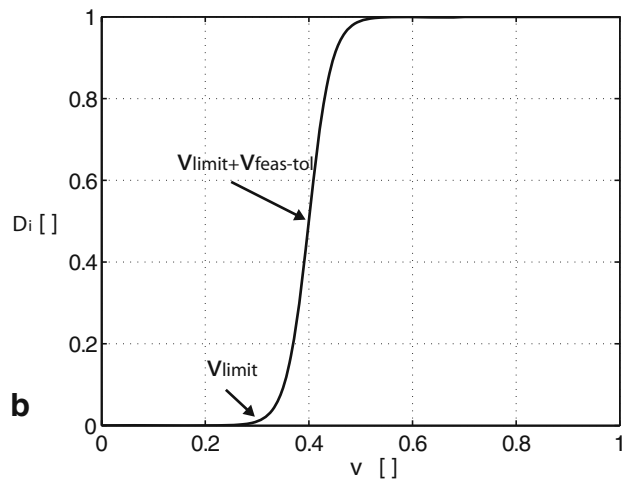
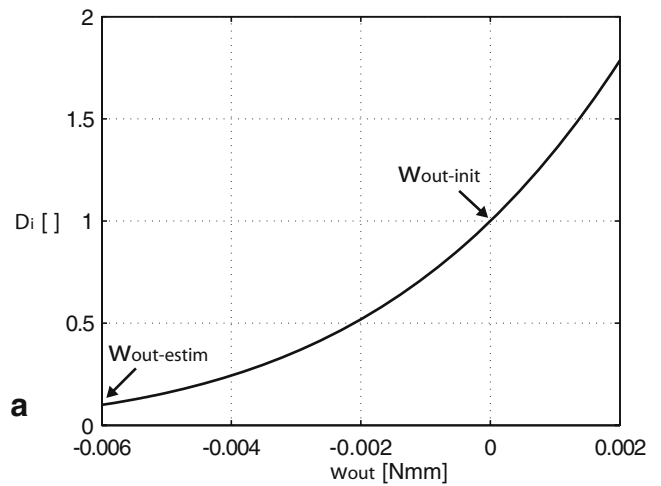


Fig. 9 Mapping of the objective/constraint value; **a** negative of the work  $w_{out}$ , **b** volume fraction  $v$

constrains. All EO functionalities are independent of the optimization problem and the used data structures.

The *initialization* is performed either randomly or based on existing designs. The parameters needed for the optimization are provided by a control file including probabilities for the reproduction. The *reproduction* generates new individuals from the parent individuals. Additional information, such as the edge prototype, the prefixed elements (vertices, edges, or springs), boundary conditions, and the fitness function, are summarized in text files. The edge prototype is needed for the definition of the range of the allowed variation for each parameter.

All data about the structure are stored in the graph genotype using the BGL and the eoUniGene concept. After generating individuals, the control routines perform a *feasibility check* (Section 5). Having ensured the feasibility, the graph-based representations of the individuals are mapped to FEMs. The finite element

**Table 3** Inverter optimization; best individuals consisting of three beams

Beam type	Fixed nodes			Moving nodes		
	Best work (mJ)	(%)	Vol. fraction	Best work (mJ)	(%)	Vol. fraction
Straight uniform beam	0.0041	100.0	0.131	0.0047	100.0	0.123
Straight, variable thickness beam	0.0050	122.0	0.299	0.0058	124.6	0.299
Curved, variable thickness beam	0.0056	135.8	0.299	0.0058	124.6	0.299

computation, which is done by our in-house developed tool, FELyX, allowing for fast *mapping* and *evaluation*, gives the structural information like the deformation or the mass back. Depending on these data, the fitness of the individuals is evaluated.

Any complex fitness function composed of weighted objective, constraints, and/or target value optimization is utilized for determination of the fitness of each individual.

## 7 Numerical examples

We validate the here presented approach, for finding compliant mechanism designs, by applying it to the classical inverter and the parallel gripper problems. First, the classical inverter problem setup is described. Three levels of beam complexity are considered, namely

- Straight beam with constant thickness,
- Straight beam with variably distributed thickness, and
- Curved beam with variably distributed thickness

and each of them is combined with two levels of optimization problem complexity, namely

- Topology problem with fixed nodes and
- Topology problem with moving nodes.

We receive a total of six different optimization setups. Each of them, we performed 30 times, and from the 30 optimization, the 15 bests are taken, and the averages of their results are computed, to eliminate outliers. Initial populations were set by random. As stop criteria, the number of generations ( $n_{\text{noimprovement}} = 5,000$ ) with no improvement and the maximal number of generations ( $n_{\text{max}} = 200,000$ ) are defined. The influence of springs placed at input and output points is also investigated.

### 7.1 Setup of the inverter optimization

The setup of the evolutionary optimization is shown in Fig. 8.

*Domain:* The optimization domain is represented by a square area. The problem is geometrically symmetric, which reduces the design domain  $\Omega$  by a half.

*Boundary:* The displacements and the rotation of the node in the upper left corner are blocked. Nodes on the the symmetric line can only move in the  $x$ -direction, and their rotations are suppressed, too.

*Springs:* At the input and output port, a predefined set of springs is added.

*Actuation:* The actuation is provided by a constantly applied force  $F$ .

*Objective:* The inverter problem has been solved by Bendsoe and Sigmund (2003) using a different method before (see Section 1). The homogenization method uses an indirect way for maximizing the output work. It depends on the mutual energy approach, which is a combination of the results of a dummy load case and the real load case, which allows generating derivatives for each cell. As our algorithm is independent of any gradient, we directly optimize the output work of the inverter.

Our objective is to maximize the work in the negative  $x$ -direction, i.e., to minimize the value of  $w_{\text{out}}$  given by:

$$w_{\text{out}} = \text{sign}(u) \frac{uk_1u}{2}, \quad (5)$$

where  $u$  is the displacement and  $k_1$  the spring constant at the output.

Moreover, the structure must fulfill the constraint of maximum volume fraction of 30%. Greater volume fractions are penalized. The volume fraction  $v$  is

**Table 4** Inverter optimization; best and average of the 15 best individuals using fixed nodes

Fixed nodes Beam type	Best			Average			
	Work (mJ)	(%)	Vol. fraction	Work (mJ)	(%)	Std dev (%)	Vol. fraction
Straight uniform beam	0.0044	100.0	0.232	0.0043	100.0	1.9	0.248
Straight, variable thickness beam	0.0051	116.2	0.299	0.0050	116.0	0.6	0.299
Curved, variable thickness beam	0.0056	128.4	0.299	0.0055	127.1	1.8	0.299



**Table 5** Inverter optimization; best and average of the 15 best individuals using moving nodes

Moving nodes Beam type	Best			Average			
	Work (mJ)	(%)	Vol. fraction	Work (mJ)	(%)	Std dev (%)	Vol. fraction
Straight uniform beam	0.0051	100.0	0.298	0.0050	100.0	1.8	0.297
Straight, variable thickness beam	0.0059	116.0	0.299	0.0058	116.6	0.5	0.299
Curved, variable thickness beam	0.0059	116.4	0.299	0.0059	117.0	0.8	0.299

defined as the ratio of the area covered by the beams  $A_i$  over the area of the domain  $A_\Omega$ .

$$v = \frac{\sum_{i=1}^n A_i}{A_\Omega} \tag{6}$$

*Fitness function:* All objective and constraint values computed during optimization are mapped to an added  $D_i$  of the fitness function. The fitness function is defined as a weighted sum:

$$S = \sum_i w_i D_i \tag{7}$$

where  $D_i$  represents the rating of an objective value or a specified constraint and  $w_i$  is the corresponding weight.

*Mapping of the objectives values:* The mapping, developed by Koenig (2004) as part of his Ph.D. work, is shortly summarized here because it cannot be found in the open literature. The basic idea was to introduce mapping functions, which are scaled to the interval [0,1], to avoid that one of these terms becomes much larger than the other ones and therefore dominant.

We distinguish between objective and constraint mapping functions. The first one is a measure of the objective of the design optimization problem. The second ones penalize violated constraints.

In case of the present problem, the design objective mapping function is defined as:

$$D_i(w_{out}) = (aw_{out} + b)^\alpha \tag{8}$$

where  $w_{out}$  is the objective, the choice of the exponential factor  $\alpha = 5$  is based on experience, and  $a$  and  $b$  are scaling factors defined by conditions:

$$\begin{aligned} D_i(w_{out} = w_{out-init}) &= 1 \\ D_i(w_{out} = w_{out-estim}) &= 0.1 \end{aligned} \tag{9}$$

$w_{out-init}$  represents an estimated initial value of the design objective and  $w_{out-estim}$  is the estimated goal value that can be achieved in the optimization. Figure 9

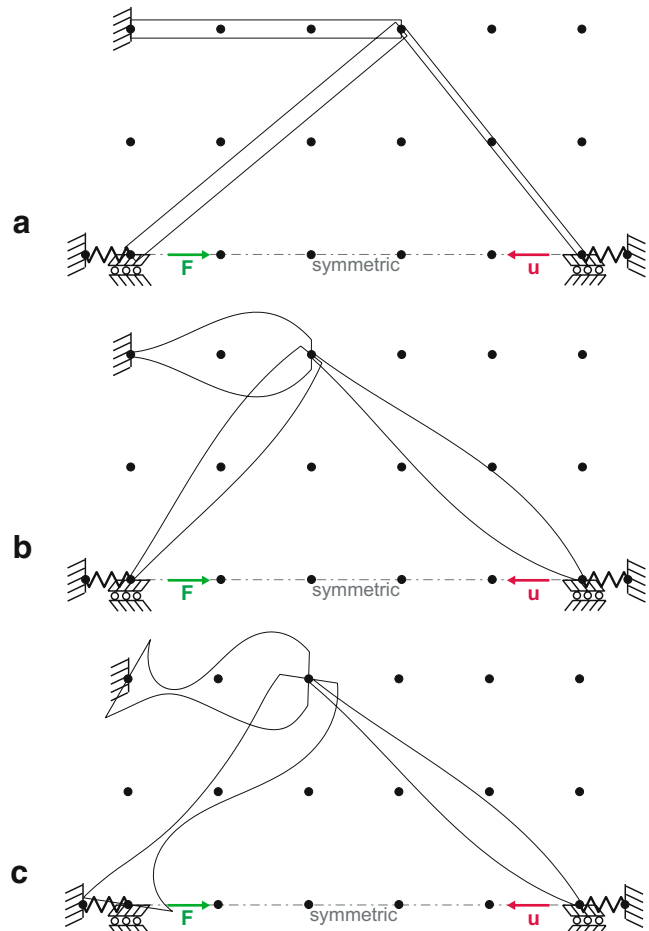
shows the mapping of the objective/constraint value to fitness values  $D_i$ .

The constraint mapping function is defined as:

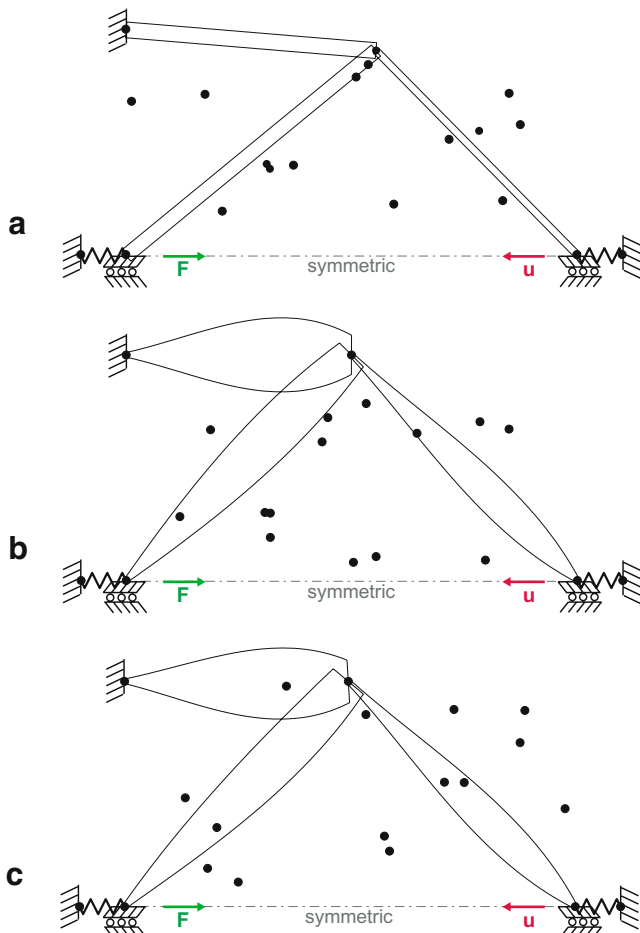
$$D_i(v) = \frac{1}{1 + \exp(-\lambda(v - v_{limit} - \Delta))} \tag{10}$$

where  $v$  is the volume fraction, which has to fulfill our constraint  $v_{limit}$ ,  $\lambda$  and  $\Delta$  are scaling factors defined by conditions:

$$\begin{aligned} D_i(v = v_{limit}) &= 0.01 \\ D_i(v = v_{limit} + v_{feas-tol}) &= 0.5 \end{aligned} \tag{11}$$



**Fig. 10** Best solution employing three beams for fixed nodes; **a** conventional, **b** variable thickness, **c** curved, variable thickness



**Fig. 11** Best solution employing three beams for moving nodes; **a** conventional, **b** variable thickness, **c** curved, variable thickness

where  $v_{\text{limit}}$  is the limited volume fraction and  $v_{\text{feas-tol}}$  is a tolerance value that allows to define the steepness of the constraining function near  $v_{\text{limit}}$  (Fig. 9).

*Evaluation:* Only small displacements are allowed, and linear analysis is carried out.

## 7.2 Results of the genetic optimization

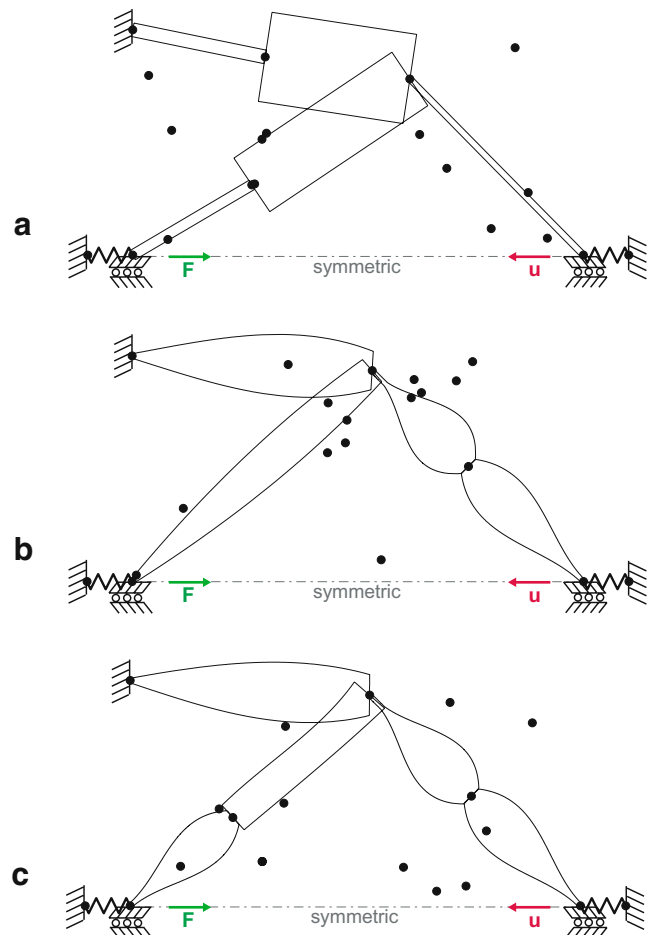
Although the optimization setup is simple, the optimization itself is not as simple as maybe expected, as the solution of the evolutionary optimization tends in the first few steps to minimization of the compliance [see (5)]. Generally, in a later step, the compliant mechanism is established. Tables 3, 4, and 5 summarize the results of this study.

As mentioned at the beginning, we intended to simultaneously increase the functionality and minimize the number of elements. First, the best solutions employing three elements are considered. Three is the

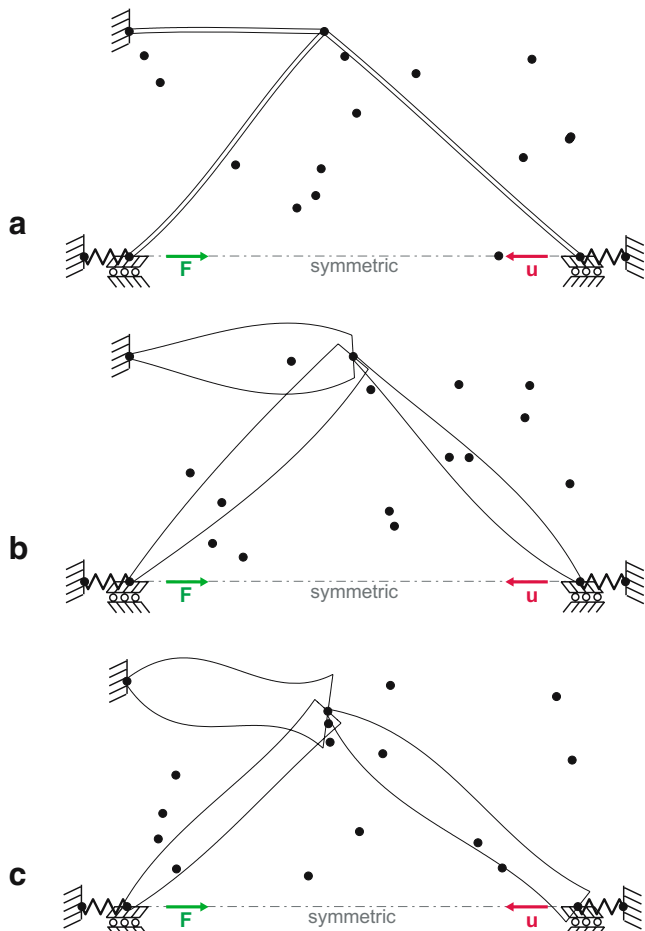
minimal number of beams that can describe a semi-symmetric inverter.

Figure 10 displays the best results, whereas nodes have been uniformly distributed within the domain. As expected, the performance improves by 22.0% using straight, variable thickness beams, by 35.8% using curved, variable thickness beams over that of achieved by straight conventional beams and fixed nodes configuration (see Table 3). The variable thickness distribution exerts an even greater influence than the moving nodes in this configuration. Using linear computation methods, the curved centerlines improve the results only if fixed nodes are specified. It is assumed that they partially compensate the geometrical limitation caused by the fixation of the nodes (see Fig. 10c).

Generally, the volume fraction of the conventional beam solution is even much lower than the allowed maximum amount of material. The two effects, namely, that less material increases the flexibility and more material decreases the loss of elastic energy, work



**Fig. 12** Best solution using moving nodes; **a** conventional, **b** variable thickness, **c** curved, variable thickness



**Fig. 13** Solution dependent on spring factor; **a**  $f = 0.01$ , **b**  $f = 1$ , **c**  $f = 10$

here stronger against each other. By contrast, all solutions using a variable thickness distribution tend to exploit the maximum non-penalized amount of material (see Fig. 11).

The results employing more than three beams confirm the outcomes of the previous configuration (Tables 4 and 5), although the improvements are smaller. The average work increases by 16.0 and 27.1% using a complex-shaped beam and fixed nodes, respectively. Employing moving nodes, the increases are 16.6 and 17.0% over that of the straight conventional beam, respectively.

The fact that the cubic thickness function approach is not optimal is shown by Fig. 12c. The evolutionary algorithms have generated quasi “bubble” beams, which seem to be the best compromise between an elastic hinge at one end and a very stiff section at the other end using as little material as possible. The observed funny shapes are due to the limited solution space of the cubic polynomial thickness parametrization.

### 7.3 Influence of the springs on the results

Springs are used to influence the resistance of the structure. The geometry as well as the sizing depends on the springs. Stiffer springs lead to stiffer mechanisms. Figure 13 illustrates the dependence.

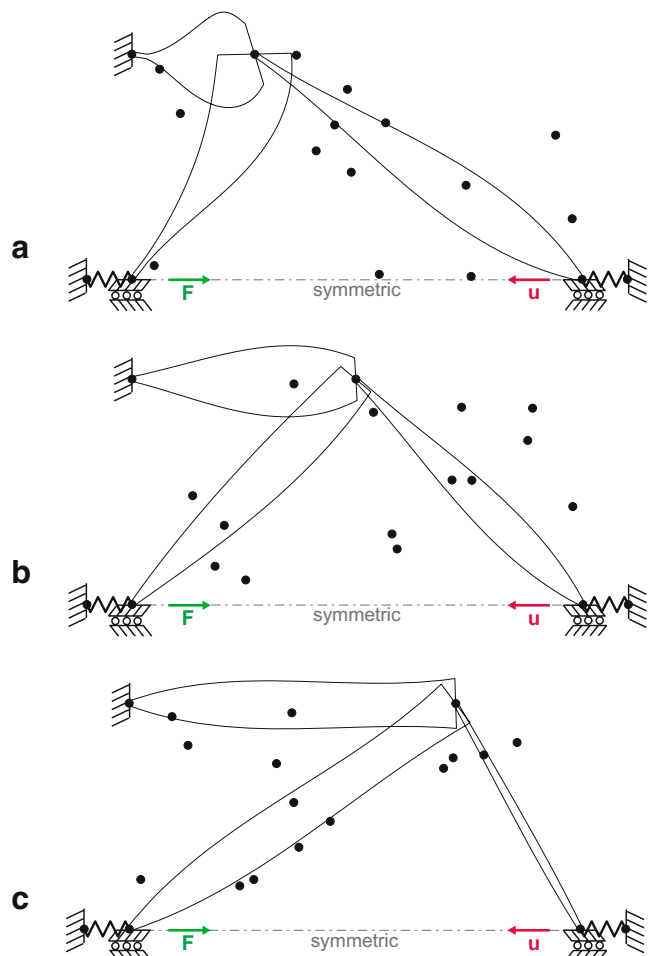
The initial set of spring constants is multiplied by the spring factor  $f$ ,  $k_{new\_i} = fk_i$ , to achieve a new set of spring constants.

According to Rahmatalla and Swan (2005), the device stiffness near the input/output tends to mimic the stiffness of the spring  $k_0/k_1$ . This can be partially explained by considering the energy/work equation:

$$W_{in} = W_{in}^{spring} + W_{in}^{transmitted} \tag{12}$$

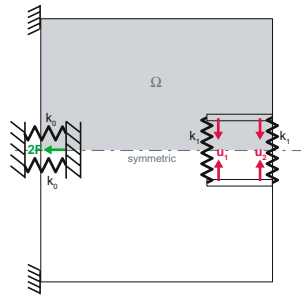
where  $W_{in}^{spring}$  is the energy stored in the input spring and  $W_{in}^{transmitted}$  the energy transferred to the mechanism.  $W_{in}^{transmitted}$  can be divided into:

$$W_{in}^{transmitted} = W_{out} + W_{mechanism}^{stored}, \tag{13}$$



**Fig. 14** Solution dependent on spring ratio; **a**  $k_{in}/k_{out} 1 : 8$ , **b**  $1 : 1$ , **c**  $8 : 1$

**Fig. 15** Gripper optimization setup



where  $W_{out}$  is equal the energy stored  $W_{out}^{spring}$  in the output spring.  $W_{mechanism}^{stored}$  is the energy stored in the compliant mechanism.

If we want to maximize the output work, we have to maximize first the energy  $W_{in}^{transmitted}$  transmitted to the mechanism and then minimize the energy stored  $W_{mechanism}^{stored}$  in the mechanism.

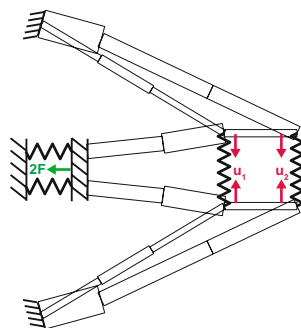
Rahmatalla and Swan's (2005) explanation is that the device stiffness near the input tends to  $k_0$  by using an analogy of two springs in parallel, whereas one spring represents the added spring and the other the stiffness of the structure near the input. Assuming that the added spring has a spring constant  $k_s$  and the structure stiffness is represented by the spring constant  $\alpha k_s$ , the work  $W_{in}^{transmitted}$  of the spring in parallel is maximized if  $\alpha = 1$ :

$$W_{in}^{transmitted} = \frac{\alpha}{2k_s} \left( \frac{F}{1 + \alpha} \right)^2, \tag{14}$$

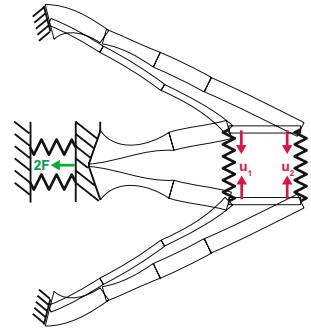
where  $F$  is the constant applied force.

Not only the stiffness itself or the sizing can influence the solution of the structure, but also the geometry of the inverter depends on the spring constants. The ratio of the magnitude of the spring constants defines the optimal  $x$ -coordinate of the node connected by three elements. If the ratio is one, the optimal  $x$ -coordinate is in the middle of the input and output. If one spring is stiffer than the other, the node moves toward the weaker spring (see Fig. 14).

**Fig. 16** Gripper optimization using conventional beams



**Fig. 17** Gripper optimization using curved beams with a quadratic thickness function



### 7.4 Result of parallel gripper optimization

A second compliant mechanism optimization problem, the parallel gripper, is added. As the optimization setup is very similar to the inverter problem, the detailed description of the optimization setup becomes redundant.

The major differences lie in the prescription of the straight beam at the output (see Fig. 15) and in the fitness function. Of the energies of the output springs 1 and 2, we maximize the double of the lower value.

For the fitness function, we use the negative of it and minimize the value  $w_{out}$  given by:

$$w_{out} = \text{sign}(v)vk_1v, \tag{15}$$

where  $k_1$  the spring constant at the output and  $v = \max(v_1, v_2)$ .

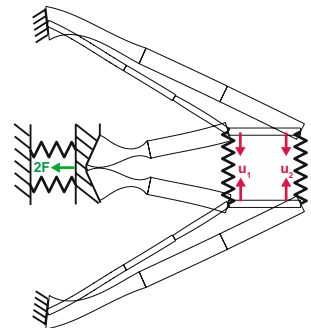
The energy in the system is limited, and first, the optimization maximizes the transmitted energy.

$$w_{transmitted} = w_{out\_1} + w_{out\_2}, \tag{16}$$

where  $w_{transmitted}$  is the energy transmitted to the output,  $w_{out\_1}$  the output energy at the first spring, and  $w_{out\_2}$  the output energy at the second spring.

As the lower output energy at either the first or second spring is maximized and the algorithm seeks to exploit the whole amount of transmitted energy, optimization leads to a result where the output energy at both springs are the same and where parallel deflections are achieved. The computed deflections of the grippers

**Fig. 18** Gripper optimization using curved beams with a cubic thickness function



**Table 6** Gripper optimization; best and average of the 15 best individuals

Moving nodes Beam type	Best			Average			
	Work (mJ)	(%)	Vol. fraction	Work (mJ)	(%)	Std dev (%)	Vol. fraction
Straight uniform beam	0.0159	100.0	0.249	0.0157	100.0	0.4	0.249
Curved beam with quadratic thickness function	0.0171	107.8	0.249	0.0164	104.2	4.1	0.245
Curved beam with cubic thickness function	0.0173	108.9	0.249	0.0170	108.1	0.9	0.249

shown in Figs. 16, 17, and 18 differ by less than 0.002% at the output.

Three different configurations using moving nodes were optimized:

- Using conventional, straight beams
- Using curved beam with a quadratic thickness function
- Using curved beam with a cubic thickness function

The results are displayed in Table 6.

The average improvement of using a complex-shaped beam for the parallel gripper is less than for the inverter problem; 4.1% for the curved, quadratic thickness function beam, and 8.1% for the curved, cubic thickness function beam approach in comparison to the straight beam optimization.

The lower efficiency increase in case of the parallel gripper compared to the inverter problems can be explained by the fact that the demand for compliant and stiff zones within one beam element is less strong.

Nevertheless, the geometries found by solving the curved, variable thickness beam gripper problem are very close to manufacturable designs, which can be regarded as an additional efficiency increase.

## 8 Conclusion and outlook

In this paper, a method for the automated design of compliant mechanisms has been presented. This method is based on evolutionary algorithms and is combining a complex-shaped beam element formulation and a graph theory-based parameter representation. We find several points of advantage. The complex-shaped beam element with thickness variation and curved centerline gives more design freedom than the straight beams, which are conventionally used. The graph theory-based topology representation greatly increases the process efficiency because of its inherent simplicity and its ability to quickly identify feasible solutions. The methodology has been validated on classical inverter and gripper problems.

The variation of the curvature does not seem to significantly improve design solutions. We speculate that curved centerlines may play a more significant role when mapping large displacements with nonlinear modeling. Shape solutions suggest that the finding of even more suitable shape parametrizations remains to be of interest. Furthermore, our future work will consider mechanical constraints stemming from strength, stiffness, and stability requirements.

## References

- Akhtar S, Tai K, Prasad J (2002) Topology optimization of compliant mechanisms using evolutionary algorithm with design geometry encoded as a graph. In: *Computers and Information in Engineering*, DECT, ASME
- Ananthasuresh G, Kota S (1995) Designing compliant mechanisms. *ASME Mech Eng* 117:93–96
- Bendsoe M, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. *Comput Methods Appl Mech Eng* 71:197–224
- Bendsoe M, Sigmund O (2003) *Topology optimization*. Springer
- Castigliano (1875) *Nuova theoria intorno dell' equilibrio dei sistemi elastici*. *Atti Acc. Sci*
- Frecker M, Kota S, Kikuchi N (1998) Optimal design of compliant mechanisms for smart structures applications. In: *Mathematics and Control in Smart Structures*, vol 3323, SPIE, pp 234–242
- Giger M, Ermanni P (2006) Evolutionary truss topology optimization using a graph-based parameterization concept. *Struct Multidisc Optim* 32:313–326
- Goldberg D (1989) *Genetic Algorithms in search, optimization and machine learning*. Addison-Wesley
- Howell L (2001) *Compliant mechanisms*. Wiley, ISBN 0-471-38478-X
- Joo J, Kota S (2004) Topological synthesis of compliant mechanisms using nonlinear beam elements. *Mech Based Des Struct Mach* 32(1):17–38
- Joo J, Kota S, Kikuchi N (2000) Topological synthesis of compliant mechanisms using linear beam elements. *Mech Struct and Mach* 28(4):245–280
- Koenig O (2004) *Evolutionary design optimization*. PhD thesis, ETH Zurich
- Kota S, Joo J, Li Z, Rodgers S, Sniogowski J (2001) Design of compliant mechanisms: applications to mems. *Analog Integr Circuits Signal Process* 29:7–15
- Kress G, Sauter M, Ermanni P (2006) Complex-shaped beam finite element. *Finite Elem Anal Des* 43:112–126
- Lobontiu N, Garcia E (2003) Analytical model of displacement amplification and stiffness optimization for a class

- of flexure-based compliant mechanisms. *Comput Struct* 81:2797–2810
- Lu KJ, Kota S (2002) Compliant mechanism synthesis for shape-change applications: preliminary results. In: Rao V (ed) *Signal processing and control*, vol 4693, SPIE, pp 161–172
- Lu KJ, Kota S (2003) Synthesis of shape morphing compliant mechanisms using a load path representation method. In: Smith R (ed) *Signal processing, and control*, vol 5049, SPIE, pp 337–348
- Nishiwaki S, Frecker M, Min S, Kikuchi N (1998) Topology optimization of compliant mechanisms using the homogenization method. *Int J Numer Meth Eng* 42:535–559
- Parsons R, Canfield S (2002) Developing genetic programming techniques for the design of compliant mechanisms. *Struct Multidisc Optim* 24:78–86
- Rahmatalla S, Swan C (2005) Sparse monolithic compliant mechanisms using continuum structural topology optimization. *Int J Numer Meth Eng* 62:1575–1605
- Saxena A (2005) Synthesis of compliant mechanisms for path generation using genetic algorithm. *ASME J Mech Des* 127:745–752
- Saxena A, Ananthasuresh G (2000) On a optimal property of compliant mechanism. *Struct Multidisc Optim* 19: 36–49
- Siek J, Lee L, Lumsdaine A (2002) *The boost graph library*, C++ in-depth series. Addison-Wesley
- Sigmund O (2001) A 99 line topology optimization code written in matlab. *Struct Multidisc Optim* 21:120–127
- Suzuki K, Kikuchi N (1991) A homogenization method for shape and topology optimization. *Comput Methods Appl Mech Eng* 93:291–318
- West D (2001) *Introduction to graph theory*. Prentice Hall
- Yin L, Ananthasuresh G (2001) Topology optimization of compliant mechanisms with multiple materials using a peak function material interpolation scheme. *Struct Multidisc Optim* 23:49–62
- Yin L, Ananthasuresh G (2003) Design of distributed compliant mechanisms. *Mech Based Des Struct Mach* 31(2): 151–179
- Zhou H, Ting KL (2005) Topological synthesis of compliant mechanisms using spanning tree theory. *ASME J Mech Des* 127:753–759