Struct Multidisc Optim 24, 78–86 © Springer-Verlag 2002
Digital Object Identifier (DOI) 10.1007/s00158-002-0216-0

Developing genetic programming techniques for the design of compliant mechanisms

R. Parsons and S.L. Canfield

Abstract Compliant mechanisms achieve desired force and displacement characteristics through elastic deformation of their structure. Current research in the synthesis of compliant mechanism topology has pursued multiobjective optimization using gradient-based search methods. This paper will explore the use of a random-guided search method for multiobjective optimization of compliant mechanisms through genetic programming techniques. The combination of genetic algorithms and compliant mechanisms is an effective and interesting combination of two biologically inspired engineering design areas. This paper will describe and demonstrate the successful use of genetic programming to create a general design tool for topological synthesis of compliant mechanisms. Features that exploit the implementation of genetic algorithms to compliant mechanism design, such as multiple criteria specification, multiple-design parameter variation, and final selections from a family of solutions will be presented and discussed. Finally, the use of this design tool will be demonstrated on several familiar examples for validation and discussion.

Key words compliant mechanisms, genetic algorithms, topological optimization

1 Introduction

Synthesis of compliant mechanisms has received significant attention recently due to the interesting and useful characteristics of these devices. Unlike traditional rigid-link mechanisms where elastic deformation is detrimental to performance, a compliant mechanism is designed to take advantage of the flexibility of the material continuum. Compliant mechanisms have many advantageous features including ease of manufacture and suitability for small scale applications (Sigmund *et al.* 1998; Saxena and Ananthasuresh 2000). Two primary methods have been developed for the synthesis of compliant mechanisms, pseudo-rigid-body methods and continuum-based methods. A number of researchers have investigated topology synthesis of distributed-compliance mechanisms using continuum-based methods (see, for example Sigmund and Petersson 1998; Saggere and Kota 1997; Larsen *et al.* 1996). A primary issue in the synthesis of a distributed-compliance mechanism is the efficient transfer of energy from the input to the work-piece, while providing the desired mechanical traits such as mechanical or geometric advantage. Therefore, the synthesis of compliant mechanisms requires consideration of multiple objectives. Optimal techniques for this multiobjective synthesis have been presented by many researchers, generally through formation of a single, inclusive objective function. For example, Frecker *et al.* (1997) present compliant mechanism development using formulations for output deformations combined with strain energy. Saxena and Ananthasuresh (2000) present a formulation based on strain energy and mutual potential energy to include stiffness, flexibility, and mechanical advantage. Hetrick and Kota (1998) form an objective function based on mechanical efficiency and a desired mechanical or geometric advantage specified through equality constraints. Larsen *et al.* (1996) present solutions of compliant mechanisms by combining both geometric advantage and mechanical advantage with target values for each in their objective functions for reaching an optimal design of the structure. Joo *et al.* (2000) demonstrate the use of nonlinear synthesis with an objective function formed from strain energy and geometric advantage and a corresponding sensitivity analysis derived. The approaches all use gradient-based optimization techniques, generally

Received December 4, 2000
Revised manuscript received July 6, 2001
Communicated by J. Sobieski

R. Parsons and S.L. Canfield

Department of Mechanical Engineering, Tennessee Technological University, Cookeville, TN 38505, USA
e-mail: `Scanfield@tntech.edu`

sequential linear programming or sequential quadratic optimization approaches.

In this paper a successful approach is demonstrated for topological optimization and design of compliant mechanisms using genetic programming techniques. Genetic algorithms and compliant mechanisms provide a complementary combination of two biologically inspired engineering design areas (Anantahsurech and Kota 1995; Goldberg 1989). The genetic algorithm is an attempt to emulate the driving force in evolution while compliant mechanism research has been motivated at times by nature's use of structures optimized for both strength and flexibility. There is much merit for basing a general design tool for topological synthesis of compliant mechanisms on genetic programming techniques, as will be demonstrated in this paper. Genetic programming is highly applicable to compliant mechanism design because of its nongradient, robust searching capabilities. As a guided random search, genetic algorithms avoid the need for gradient derivations, allowing much greater freedom in the selection of objective functions and the use of linear or nonlinear finite element analysis tools. In application, the genetic algorithm gives the designer freedom in selecting the final design by returning results as a family of solutions, each with slightly different characteristics, while meeting the design criteria. Genetic algorithms can be readily combined with other search tools, often providing solutions for final refinement by a specific optimization tool. Finally, genetic algorithms have the ability to perform multiobjective optimization as required by a general design tool, here demonstrated based on similarity template matching among solutions ranked according to the various design criteria.

Genetic algorithms were first introduced by John Holland and his colleagues and students at the University of Michigan (Goldberg 1989), and have since found application in many areas of engineering optimization (see for example Chapman *et al.* 1994; Tai *et al.* 2000). Genetic algorithms are searching techniques based on the natural forces of evolution, forces that join the processes of natural selection with random information exchange among designs and slight perturbations or mutations to the system to create a robust and appealing optimization approach. In genetic programming applications, each design carries a blueprint of its characteristics, called a chromosome, in the form of a string structure. Designs combine in pairs to create new designs made up of elements of each of the original designs. Following the evolutionary precedent, only the fittest designs contribute largely to future generations, while poorly suited designs are removed from the population. Occasional mutation of individual traits occurs throughout the process, ensuring diversity in the gene pool, allowing designs to continue to evolve. The cycle of breeding, mutation, and attrition continues for as many generations as specified by the user. In the end, the engineer is provided a group or family of good solutions rather than a single solution. Genetic algorithms are robust, computationally simple yet capable in their searching abilities. They are not generally limited by existence of derivatives or gradient searching, continuity of the search space, or the efficiency losses associated with random search methods. Finally, genetic algorithms differ from traditional search techniques in that they search from a population of points over the space rather than a single point, they use objective function information rather than gradients for guidance, and they rely on probabilistic rather than deterministic rules in moving toward an optimum Goldberg (1989).

The remainder of this paper will investigate issues in the use of genetic programming techniques for topological optimization of distributed-compliance, compliant mechanisms. Specifically, this paper will describe and demonstrate the successful use of genetic programming to create a general design tool for topological synthesis of compliant mechanisms. Features that exploit the implementation of genetic algorithms to compliant mechanism design, such as multiple criteria specification, multiple-design parameter variation, and final selections from a family of solutions will be presented and discussed. Finally, the use of this design tool will be demonstrated on several familiar examples for validation and discussion.

2 Procedure

The procedure for optimal topological synthesis will be patterned after the genetic algorithm discussed by Goldberg (1989). Potential solutions to the problem, which will be called designs, collectively form a population of potential solutions. Each design is distinct, and carries its own unique identifying code, an information structure that will be called a chromosome. The chromosome is composed of genes and is represented by an ℓ-bit string of base j. Each gene may take on some number of values, called alleles. For each individual gene there are j possible variations that provide unique definition of that characteristic. For example, each gene in the chromosome string may represent distinct regions of a general design domain while the value of each gene may define local mass and structural properties. With this design representation, the total number of distinct designs, P_{tot} equals j^{ℓ}.

For the compliant mechanism topology optimization, the population represents a planar region of base material of maximum allowable size for the compliant mechanism. This region is represented by a network of discrete elements that collectively form the design domain as shown in Fig. 1. Each discrete element represents the mass and stiffness properties of a unique section of the overall material and will be modelled here as three-degree-of-freedom beam elements. Properties associated with each gene are characteristic of the homogenous material and represent the degree of structural stiffness and mass at that unique

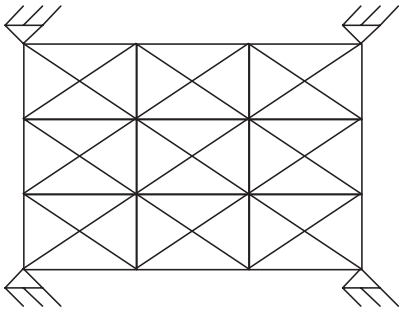

Fig. 1 Design domain

location. In order to facilitate the use of similarity templates, each gene will be represented as a binary number within the chromosome string. Thus, each gene requires m bits if 2^m allele values for each characteristic are to be defined. The chromosome of each design uniquely represents the occurrence and properties of beam elements over the design domain, and hence uniquely defines a possible solution for the compliant mechanism.

With a means of identifying potential solutions for the compliant mechanism defined, the process of determining the best solution or family of solutions proceeds using three operators, reproduction, crossover, and mutation (Goldberg 1989). In reproduction, individual strings are assigned a probability to produce offspring in the next generation according to their fitness values and are then copied into a mating pool consistent with this probability. This gives designs with higher fitness values higher probability to contribute one or more children to the next generation. The operation of reproduction is implemented by first assigning a probability to each member in proportion to its fitness,

$$p_i = \frac{f_i}{\sum_{i=1}^{n} f_i}\,, \tag{1}$$

where p_i is the probability to reproduce and f_i is the fitness value of population member i in a population with n members. The expected number of offspring from each member is given as,

$$\text{offspring}_i = n * p_i\,. \tag{2}$$

The number of offspring can be assigned using a probability-biased roulette wheel or random number generator with appropriate fitness scaling to regulate the number of offspring (Goldberg 1989). The resulting copies are entered into the mating pool to form the new population. Crossover is the process of combining the members in the mating pool to produce the new population. Demonstrated here is a simple crossover that consists of first a random pairing of designs for mating, followed by a simple exchange of string values about some randomly chosen integer position k along the string length ℓ, with each pair of mating designs resulting in a pair of offspring.

Each successive generation is based on the previous generation, with genetic information passed on in various combinations from the original population. Diversity in the genetics is essential, and is provided through a process of mutation and recombination (Avers 1974) that occurs at the breeding process. Mutations will occur on a small percentage of the genes of the offspring, (1%) with mutation performed on a bit-wise basis, changing the value of a given gene at a small rate called the probability of mutation rate.

To begin the optimization, an initial population is formed of n members or designs chosen through a random generation of chromosome strings over the solution space domain. The fitness of each member, f_i is evaluated based on a defined scalar design criteria. Based on the fitness evaluation, a new population is formed through the operations of reproduction, crossover, and mutation. The process then proceeds repeating the steps of reproduction, crossover and mutation, with new generations acquiring successful traits from the previous. The genetic algorithm ends when the overall increase in fitness of the population reaches steady-state values. At that point, the results are provided in a family of solutions, each ranked according to its fitness. This family of solutions provides the designer with several design options, for further refinement or selection based on specific problem needs.

2.1 Discussion of similarity templates

The simple genetic algorithm consists of highly fit chromosomes strings combining to result in improved solutions to the design problem. Goldberg (1989) discusses the use of a schema as a similarity template describing a subset of the chromosome strings with similarities at certain string positions. A schema can be represented as a string, similar to the original chromosome string with the additional symbol * added to represent no similarity in the gene position. Hence, a schema represents positions along the chromosome string in which similarity between designs occur. Alternatively, a schema represents some number of individual designs, all containing certain similarities. For example, the schema [**1**001**] can describe any of the four members

[**100001**], [**101001**], [**110001**], [**111001**].

The value of schemata is that they provide a compact form for considering defined similarities among chromosome strings of a population. Schemata can be used to define important similarities among designs that have high fitness values. Thus, schemata with above-average fitness contribute more to the following generation; those schemata with the highest fitness levels contribute the greatest. In this respect, the similarities between two highly-fit designs are propagated forward to the next gen-

eration. For example, consider the mating between two members, [**0 1 1 0 0 1 1 0**] and [**1 1 0 1 0 1 0 1**]. The schema representing these two designs is

[*** 1 * * 0 1 * ***].

Offspring resulting from a crossover about the fourth chromosome bit are

[**0 1 1 0 0 1 0 1**] and [**1 1 0 1 0 1 1 0**]

which share the same schema as the parent designs.

These highly-fit schema represent features in the member or design that contribute to meeting the objectives of the design. This gives a tool for tracking the progress of the genetic algorithm in the topological optimization approach. It also demonstrates how the results of genetic algorithms result in a family of solutions for the designer to choose from. As generations in the GA optimization converge, one or several schemata are produced which give the optimal results as families of solutions. Each schema represents a family of solutions of size j^d where j is the number of alleles and d is the number of * (don't care) operators in the schema. These variations in the members of a schema involve changes in value at the genes with the * operators, and in general give various solutions with similar effectiveness meeting the design objectives. Schemata also provide a powerful tool for multi objective design.

2.2 Multiobjective design

Optimal topological synthesis of compliant mechanisms based on multiple objectives is a necessary tool for most design applications. Genetic algorithms and the idea of schemata provide a significant tool for multiobjective optimization and will be demonstrated here. This approach is based on the notion that fit schemata represent similarities among population members with above-average fitness, and combine to propagate these advantageous features forward to new generations. For the case of multiobjective optimization, the entire population is evaluated based on its ability to meet multiple objectives, with a unique fitness value assigned to each member according to each design objective. The process of reproduction now forms copies of the members of the original population into several pools, one for each design objective. The members are copied into each pool according to their level of fitness in that design objective (level of fitness gives probability to reproduce). Once the pools are filled with copies for mating, the pools are recombined to allow crossover between all member copies. Crossover followed by mutation produces a new population, and the process of multifitness evaluation for reproduction begins again. This process is shown schematically in Fig. 2. Note that the number of members in the original population and the sum of the members of mating copies in the pools must remain the same to maintain the size of the new generation.

3 Fitness function definition

The fitness or objective function is defined based on the goals of the problem. In the design of compliant mechanisms, both the desired motion requirements and the structural requirements must be met simultaneously (Frecker *et al.* 1997), to avoid situations in which the device is too flexible to meet the required task force objectives, or is too stiff requiring too much input force for a specific task. These issues have been dealt with extensively (for example, Frecker *et al.* 1997; Saxena and

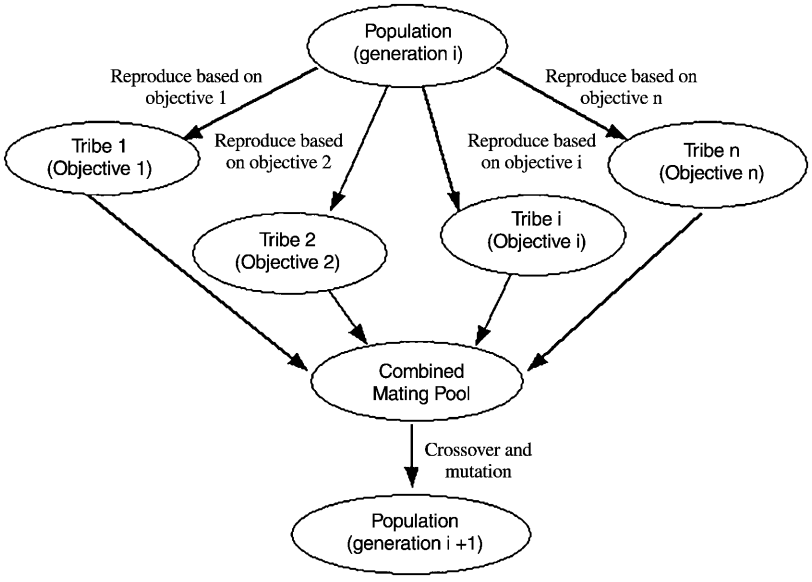

Fig. 2 Multiobjective optimization scheme

Ananthasuresh 2000; Salamon and Midha 1998; Hetrick and Kota 1998) resulting in several scalar parameters or objectives that can be combined to overcome these difficulties. Some examples of these objectives include mutual potential energy for flexibility requirements, strain energy for stiffness requirements, mechanical and geometric advantage to provide ratios of input to output force or motion, and mechanical efficiency to maximize work out per given work in. All compliant mechanism techniques require the addition of a force or spring to the finite element model to represent the level of work expected at the output of the device. It should be noted that most formulations include strain energy or mechanical efficiency with an additional task-specified requirement to ensure successful practical mechanisms. For demonstration of the implementation of genetic algorithms to optimization of compliant mechanisms, several objectives will be defined; maximizing mechanical efficiency, meeting a desired geometric advantage or mechanical advantage, and minimizing maximum compressive loads in the design. Mechanical efficiency, the ratio of work out to work in will be defined as

$$f^1 = \eta_{\text{efficiency}} = \frac{W_o}{W_i}\,, \tag{3}$$

where W_o is the work out,

$$W_o = \text{sign}(u_o) * \frac{1}{2} * k_s * (u_o)^2\,, \tag{4}$$

and W_i is the work into the system,

$$W_i = \delta S + W_o\,. \tag{5}$$

In these equations, u_o is the scalar output displacement in the direction of the spring, k_s is the stiffness of the external spring, and δS is the strain energy stored in the mechanism. If the external spring or springs representing the work piece is included in the mechanism stiffness matrix, then W_i becomes

$$W_i = \frac{1}{2} * \mathbf{u}^T * \mathbf{K}_{\text{tot}} * \mathbf{u}\,, \tag{6}$$

with $\mathbf{u}$ the vector of nodal displacements and the K_{tot} the stiffness matrix representing both the mechanism and the external springs. Therefore, the mechanical efficiency is given as,

$$f^1 = \frac{\text{sign}(u_o) * \frac{1}{2} * k_s * (u_o)^2}{\frac{1}{2} * \mathbf{u}^T * \mathbf{K}_{\text{tot}} * \mathbf{u}}\,. \tag{7}$$

The fitness functions evaluating the geometric advantage and mechanical advantage are given as

$$f^2 = \frac{u_o}{u_i} = GA\,, \tag{8}$$

$$f^3 = \frac{F_o}{F_i} = MA\,, \tag{9}$$

where u_o and u_i are the input and output node displacements in the desired directions. In the geometric or mechanical advantage fitness, designs are given increasing probability for reproduction as they approach the desired level, while no additional probability is given to designs that exceed the desired level of mechanical or geometric advantage.

A third objective, to minimize the maximum compressive load is defined as

$$f^4 = \frac{1}{\max(|\sigma_{\text{comp}}|)}\,, \tag{10}$$

where σ_{comp} is negative when the member is under compressive load.

Evaluation of these fitness functions is implemented using a linear finite element analysis paired with the specific objective function formulations. Once implemented in the genetic algorithm, the probability to reproduce for each fitness function is determined independently for each fitness function, creating multiple pools for mating

$$p_i^k = \frac{f_i^k}{\sum_{i=1}^{n} f_i^k}\,. \tag{11}$$

In cases where some fitness values are negative, the entire fitness vector for that specific objective is scaled such that the smallest value is zero. Then, probabilities to reproduce are assigned and the fitness values are returned to their original state.

4 Examples

4.1 Example 1: force inverter

The genetic programming approach is first demonstrated on the force inverter problem to create a compliant device that produces an output displacement and force in a direction opposite that of the input force (Saxena and Ananthasuresh 2000). The problem begins with a square design domain with pin supports at each corner and input and output loads as shown in Fig. 3 and Table 1 with an external spring added to represent the expected task. This domain is discretized into 20, three degree-of-

Table 1 Design parameters for example 1

$k_s = 10$ N/cm (output spring stiffness)
$E = 2 \times 10^9$ MPa (nylon 6/6)
rectangular cross-section
thickness = 0.1 cm
width = 0.10 cm

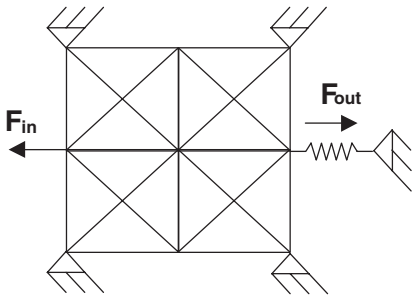

Fig. 3 Design domain of Example 1

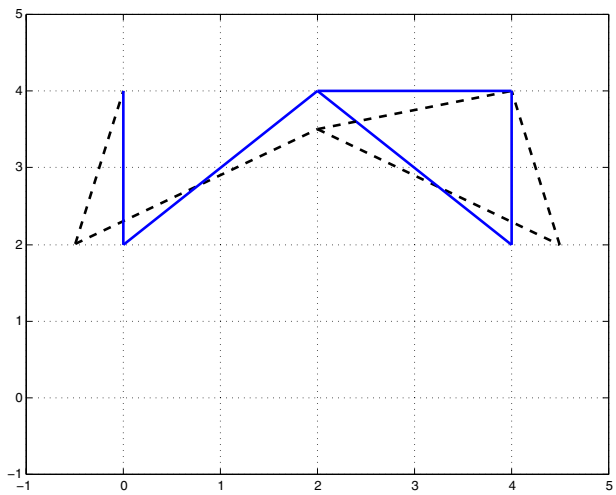

Fig. 4 Fitness values ME = 0.3939 and MA = 0.4013 for Example 1

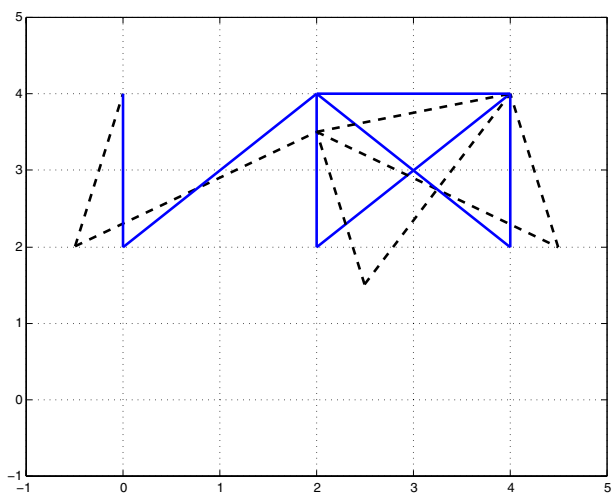

Fig. 5 Fitness values ME = 0.3676 and MA = 0.3747 for Example 1

freedom beam elements with area and modulus properties defined. The problem is then solved based on the multiple criteria of maximizing mechanical efficiency, maximizing geometric advantage, and minimizing the maximum compressive loads. The final results are delivered as a family of solutions, shown in Figs. 4 and 5 and represent members selected from the final generation of the genetic algorithm. In each, the solid line represents the truss topology while the dashed line gives the deformed position of the truss.

4.2 Compliant mechanism optimization using multibit alleles and diploids

The first example established the abilities of the genetic algorithm optimization tool to produce a family of suited solutions to a given problem, with the resulting solutions corresponding favorably to other published work. This first genetic algorithm demonstrated simple implementation using single chromosome strings with single-bit alleles to represent members of the population. A new algorithm is now presented that will demonstrate the use of multibit alleles, and the use of diploids, or double-string chromosomes. Multibit alleles allow the values designated to each gene to vary over a range with resolution:

$$\text{resolution} = \frac{\text{range}}{2^q}$$

where q is the number of bits representing each allele. The use of multistring chromosomes allow multiple design characteristics to be accompany each design, with each characteristic sharing the same processes of reproduction, crossover and mutation. These two additions are applied to the compliant mechanism design tool as follows. First, variation of the dimensional properties of each element is permitted through the use of a multibit allele in the first string of the design chromosome. Next, node wandering is implemented as suggested and demonstrated by Hetrick and Kota (1998). Node wandering allows the nodes forming the elements in the discretized workspace to move location in order to allow a much greater range of possible designs considered during the optimization process. The node positions are therefore part of the design's characteristics and are carried in a second string in the chromosome. With these two strings, the design's represent diploid or double-string chromosome organisms. With the combination of node wandering and element property variation, there is much greater potential for a truly optimized design to result from the initial uniform discretized workspace.

4.3 Example 2: force inverter with node wandering

Application of this modified approach is demonstrated in the following example. This example considers again the force inverting mechanism of Example 1. Now, the multibit allele, double-string chromosome genetic algorithm searches the original workspace shown in Fig. 3 for optimal solutions. Note that the number of possible de-

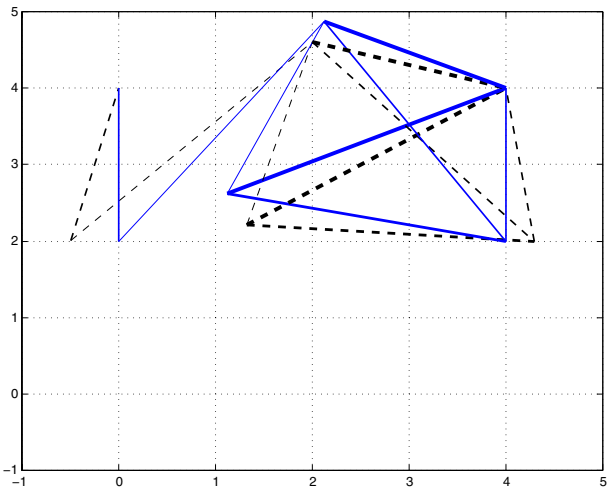

Fig. 6 Fitness values ME = 0.9730 and MA = 1.685 for Example 2

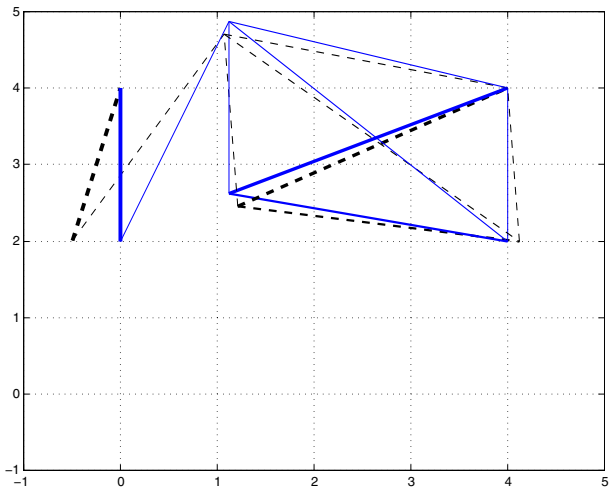

Fig. 7 Fitness values ME = 0.945 and MA = 3.92 for Example 2

signs is significantly greater, this time 2^{52}. Solutions to this problem are again returned as a family of solutions, shown in Figs. 6 and 7. These solutions show significant improvements to those presented in Example 1, but contain many similarities in topology.

The convergence history of the fitness functions in evolving populations is shown in Fig. 8 which demonstrates both the "fittest" individual in each population as well as the average. The resulting topologies are both diverse and agree well with results from similar approaches.

4.4 Example 3: compliant gripper

In the last example, the familiar compliant gripper problem is considered. The problem begins with an initial design domain consisting of 56 elements and boundary conditions located along the symmetrical midline (Fig. 9) and design parameters as shown in Table 2. The input is specified at the gripper handles as a given displacement while the output is measured at the gripper jaws. A spring is also placed at the gripper jaws represented an expected

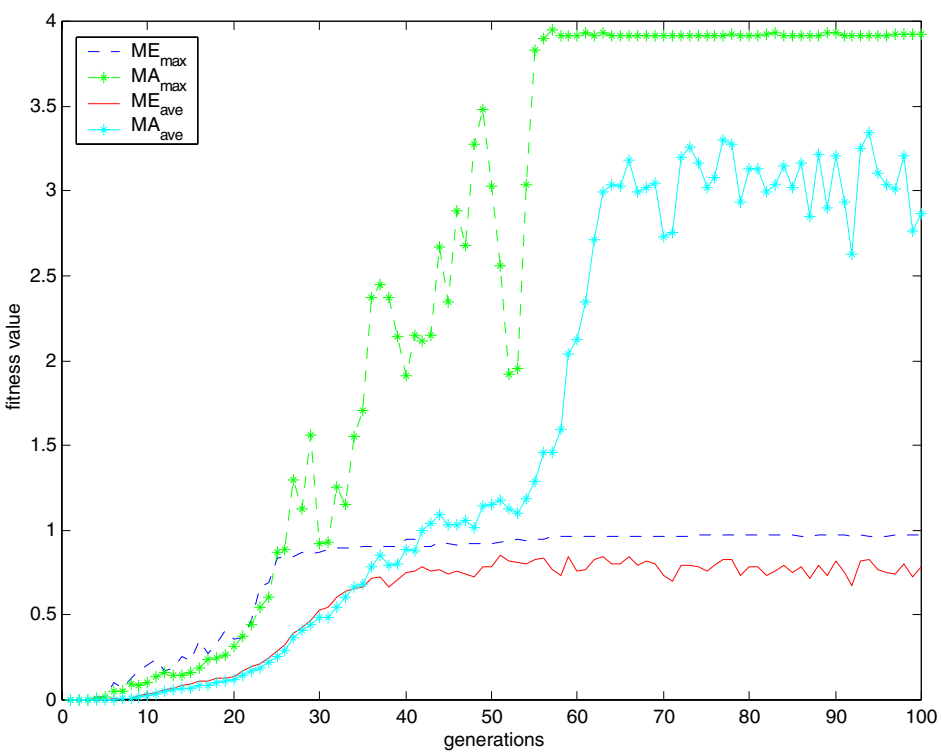

Fig. 8 Convergence history of Example 1

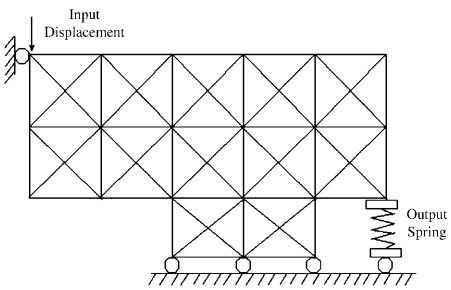

Fig. 9 Design domain of Example 3

Table 2 Design parameters for Example 3

$k_s = 10\,\mathrm{N/cm}$ (output spring stiffness)
$E = 2 \times 10^9$ MPa (nylon 6/6)
rectangular cross-section
thickness = 0.1 cm
maximum width = 0.15 cm
node wander range = 1.0 cm

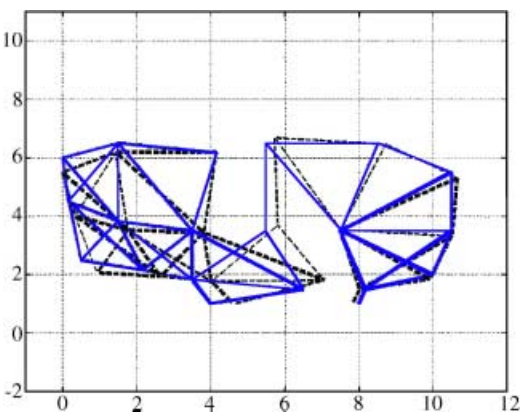

Fig. 10 Gripper with fitness values ME = 0.89 and MA = 3.03

output load. The genetic algorithm performs the optimal synthesis allowing node wandering and cross-sectional area variation as in Example 2 with one candidate solution shown in Fig. 10. Again this result is comparable with topologies of other published examples (for example Hetrick and Kota 1998), sharing common features such as stiffening around the gripper jaw, lengthening of the handle, and a crossed-link formation between the jaw and handle.

5 Results and conclusions

This paper has demonstrated the application of specific genetic algorithms to synthesize families of compliant mechanism topologies. The examples established several of the features of genetic algorithms including robust, nongradient-based searching capabilities, results returned as families of solutions and successful implementation of multicriteria optimization. Through the multicriteria formulation, many additional factors in the design of compliant mechanism could be considered such as mechanical impedance, damping and resonance. The family of results may also benefit the designer when considering desirable properties for a specific application and when using the results of the genetic algorithm as a starting point for a gradient based optimization routine. Once the basic application of the genetic algorithm was outlined, the use of multibit alleles and double-string chromosomes were demonstrated, allowing node wandering and element property variation in the synthesis of compliant mechanisms. The addition of these capabilities are significant in that a much greater domain of potential solutions may be searched while keeping the number of elements at a practical size. An added characteristic of this algorithm is the ability to eliminate unnecessary elements and thus remove unconnected nodes. This may result in topologies that more accurately represent a physical structure. In summary, the application of genetic algorithms to topological synthesis of compliant mechanism provides another significant and effective design tool that may be exploited in developing these devices.

Acknowledgements The authors express their appreciation to Dynamic Systems and Structures of Franklin, Tennessee for their support and the support of BMDO (Ballistic Missile Defense Organization), contract number DASG60-00-M-0056.

References

Ananthasuresh, G.K.; Kota 1995: Designing compliant mechanisms. *Mech. Engrg.* November, 93–96

Avers, C. 1974: *Evolution.* New York: Harper and Row

Chapman, C.D.; Saitou, K.; M.J. Jakiela 1994: Genetic algorithms as an approach to configuration and topology designs. *Trans. ASME J. Mech. Des.* **116**, 1005–1013

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

Goldberg, D. 1989: Genetic algorithms in search. *Optimiz. Mach. Learning.* Reading, MA: Addison-Wesley

Hetrick, J.A.; Kota, S. 1998: Size and shape optimization of compliant mechanisms: an efficiency formulation. *Proc. ASME Design Engineering Technical Conf.* DETC98/MECH-5943

Joo, J.; Kota, S.; Noboru, K. 2000: Nonlinear synthesis of compliant mechanism: topology design. *Proc. ASME Design Engineering Technical Conf.*, DETC2000/MECH-14141

Larsen, U.D.; Sigmund, O.; Bouwatra, S. 1996: Design and fabrication of compliant micromechanisms and structures with negative Poisson's ratio. *Proc. 9-th IEEE Workshop on Micro-Electro-Mechanical Systems*, pp. 365–370

Saggere, L.; Kota, S. 1997: Synthesis of distributed compliant mechanisms for adaptive structures application: an elasto-kinematic approach. *Proc. ASME Design Engineering Technical Conf.*, DETC1997/DAC-3861

Salamon, G.A.; Midha, A. 1998: An introduction to mechanical advantage in compliant mechanisms. *Trans. ASME J. Mech. Des.* **120**, 311–315

Saxena, A.; Ananthasuresh, G.K. 2000: On an optimal property of compliant topologies. *Struct. Multidisc. Optim.* **19**, 36–49

Sigmund, O.; Petersson, J. 1998: Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Struct. Optim.* **16**, 68–75

Sigmund, O.; Torquato, S.; Aksay, I.A. 1998: On the design of 1-3 piezocomposites using topology optimization. *J. Mat. Res.* **13**, 1038–1048

Tai, K.; Cui, G.Y.; Ray, T. 2000: Design synthesis of path generating compliant mechanisms By evolutionary optimization of topology and shape. *Proc. ASME Design Engineering Technical Conf.*, DETC2000/DAC-14