

Automated Type Synthesis of Planar Mechanisms Using Numeric Optimization With Genetic Algorithms

Yi Liu

Department of Systems Design Engineering,
University of Waterloo,
Waterloo, Ontario N2L 3G1, Canada
e-mail: yiliu@real.uwaterloo.ca

John McPhee¹

Department of Systems Design Engineering,
University of Waterloo,
Waterloo, Ontario N2L 3G1, Canada
e-mail: mcphee@real.uwaterloo.ca

This paper presents a novel method for the automated type synthesis of planar mechanisms and multibody systems. The method explicitly includes topology as a design variable in an optimization framework based on a genetic algorithm (GA). Each binary string genome of the GA represents the concatenation of the upper-right triangular portion of the link adjacency matrix of a mechanism. Different topologies can be explored by the GA by applying genetic operators to the genomes. The evolutionary process is not dependent on the results obtained from enumeration. Two examples of topology-based optimization show the applicability of this method to mechanism type synthesis problems. This method is distinct from others in the literature in that it represents the first fully automated algorithm for solving a general type synthesis problem with the help of a numeric optimizer. [DOI: 10.1115/1.1904049]

1 Introduction

The procedure for systematically conducting mechanism design consists of three steps: task definition, type synthesis, and dimensional synthesis [1]. Dimensional synthesis seeks to determine optimal dimensions and inertial properties of a prescribed type of mechanism; it has been studied in depth. Some commercial software, such as LINCAGES-4 (6) and WATT, is helping designers to efficiently solve kinematic dimensional synthesis problems, although they typically handle only specific types such as planar 4 bars or some 6 bars. Other mechanical system analysis packages, such as ADAMS, may serve as dynamic dimensional synthesis engines when combined with optimization methods. However, type synthesis, i.e., the selection of the optimum topology among feasible mechanism structures for a specified task, is still an open problem in the field of computer-aided mechanism design. Current type synthesis techniques mostly involve the enumeration of distinct mechanisms that satisfy a certain degree of freedom (DOF) requirement. Linear graph theory is used extensively in type synthesis, which can be subdivided into number synthesis, topological synthesis, and topological analysis [2].

Erdman [3] provides an overview and literature review of mechanism design and concludes that "the most critical stage of the entire design process is ...choosing the best topology for a given task" (see Ref. [3], p. 95), otherwise known as topology optimization or type synthesis. Another overview is provided by Norton [4], who also reviews the application of numerical optimization to mechanism synthesis. Soni et al. [5] developed a rules-based expert system for mechanism type selection and automated dimensional synthesis, but the system can handle only specific mechanism design problems and is based on heuristics, not on a numeric optimization framework. Wang et al. [6] proposed another computerized rules-based method for conceptual design of complex mechanisms based on Yan's regeneration-oriented creative design theory [7]. Chiou et al. [8] developed a computer program for the design of function generating mechanisms, adopting the function decomposing and recomposing creative design

method based on a matrix representation scheme and basic kinematic building blocks. However, neither Wang et al. nor Chiou et al. considered the selection of optimum mechanism types for a given task.

Kong et al. have used the screw theory to develop a method for enumerating spatial parallel robots with decoupled degrees of freedom, either translational [9] or rotational [10]. However, their method is a manual approach that is restricted to the special topologies of parallel robots, not an automated approach to the type synthesis of general mechanisms.

Much success has been reported in the area of compliant mechanism design using continuum-based topology optimization techniques; an excellent description of this approach can be found in Bendsoe and Sigmund [11]. In an attempt to extend this work to articulated mechanisms with rigid links, Kawamoto et al. [12] used a truss-based ground-structure representation and graph-theoretical enumeration to perform an exhaustive analysis of all possible topologies. This method was improved by Felter [13], who used an energy or geometry optimization approach to identify and remove redundant joints and links. In the work by Kawamoto et al. and Felter, however, only binary links with a set of fixed dimensions were considered, and Gruebler's criterion for the DOF of a planar mechanism was not utilized.

Eberhard et al. [14] have pointed out the merits of using genetic algorithms (GAs) in multicriteria optimization of multibody system parameters, but topology is not included as a design variable. Leger [15] has accomplished automated synthesis and optimization of open-chain robot configurations with a variant of a GA. However, the type synthesis problem for general mechanisms is not tackled. Rao [16] used a GA to enumerate planar mechanisms, test for isomorphism among kinematic chains, and identify kinematic inversions. However, the selection of the "best" chain and inversion for function generation is not accomplished by the GA, but through the application of some proposed design heuristics.

Norton states that only Fang [17] has attempted to solve the general type synthesis problem using numerical optimization. Actually, Fang used a GA to simultaneously determine the optimal dimensions for several mechanisms of a fixed four-link topology. Topology is not explicitly used as a design variable, and no new topologies are created during the GA evolutionary process. Fang notes that infeasible mechanisms can simply be assigned a null fitness, so that they do not cause any convergence problems.

¹To whom correspondence should be addressed.

Contributed by the Design Automation Committee for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received March 9, 2004; revised November 5, 2004. Associate Editor: K. K. Choi.

In this paper, the topology of a mechanism is explicitly included as one design variable. Given some design objectives, a GA is used to automate the procedure for finding the best mechanism type, using a novel method to encode the topology of a mechanism into a binary string genome. When genetic operations are applied to these genomes, different topologies will be randomly explored by the GA. Since the GA is capable of handling both discrete and continuous parameters, it can also be used to simultaneously integrate type and dimensional synthesis, a step that is necessary for the conceptual design of mechanisms in the early design stage. Therefore, what distinguishes the proposed algorithm is the fact that it fully automates the process of solving a general type synthesis problem within a numeric optimization framework.

This paper is organized as follows. An introduction to the representation of mechanism topology is presented in Sec. 2. An outline of GAs and the framework for the automated type synthesis of planar mechanisms then follows. Next, the applications of the algorithms to two topology-based optimization examples and some observations on computational efficiency are discussed in Sec. 4. Finally, conclusions are drawn in the last section.

2 Representations of Mechanism Topology

The topology of a mechanism contains the essential information about which link is connected to which other link by what type of joint; it can be hierarchically represented in several different ways, including functional schematic representation, structural representation, graph representation, and matrix representation, etc. [18]. Figure 1 shows the functional schematic, kinematic structure, graph, and adjacency matrix representations of a four-link mechanism.

The functional schematic is the familiar cross-sectional drawing of a mechanism in which only those elements that are essential to the function of the mechanism are shown.

In a structural representation, each link of a mechanism is denoted by a polygon whose vertices represent the kinematic joints, and the polygon denoting the ground link is labeled accordingly. The vertices of a structural representation can be colored or labeled for the identification of kinematic joints.

A graph of a mechanism can be formed by representing links as the vertices and joints as the edges, and by labeling the vertex denoting the ground link. The edge connection between vertices corresponds to the kinematic joint connection between links. To make a difference between various joint connections, the edges can be labeled or colored.

For convenience of computer programming, the topology of a mechanism is expressed in matrix forms. The link-to-link adjacency matrix is the most frequently used one. Other representation

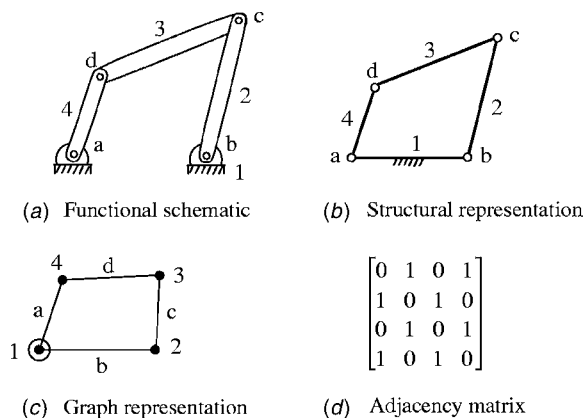


Fig. 1 Four-link mechanism and its kinematic representation

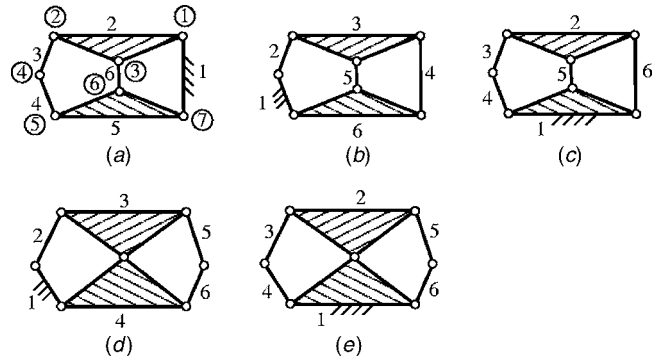


Fig. 2 Five types of six-link, 1-DOF mechanisms

forms, such as the incidence matrix, are also useful for the identification and classification of mechanisms, as well as kinematic and dynamic analysis [19].

The n links of a planar mechanism are numbered sequentially from 1 to n . Its link-to-link adjacency matrix (LAM) is defined as follows:

$$\text{LAM}_{ij} = \begin{cases} 1 & \text{if link } i \text{ is adjacent to link } j \\ 0 & \text{otherwise (including } i=j) \end{cases} \quad (1)$$

By definition, the LAM is an $n \times n$ symmetric matrix with zero diagonal entries. The matrix determines the structural topology of a mechanism up to structural isomorphism (see Section 3.3).

Shown in Fig. 2 are structural representations of five types of six-link mechanisms with one DOF. Subfigures (a)–(e) respectively are Stephenson I, II, and III and Watt I and II mechanisms. Every link attached to a hatched area is the ground link and numbered as link 1. The LAM of the Stephenson I mechanism in Fig. 2(a) is

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (2)$$

Note that the upper-right triangular portion of a LAM contains sufficient information about the topology of a mechanism because of its symmetry. Manipulating the LAMs will reveal some topological characteristics of the mechanisms, such as structural isomorphism, etc.

3 An Approach to Automated Type Synthesis

3.1 Overview of Genetic Algorithms. GAs are general-purpose stochastic optimization methods for finding global optima, especially suitable for poorly characterized solution spaces. GAs can be applied to optimization problems involving either continuous or discrete parameters.

A GA emulates an ecological system in which the mechanics of natural selection and natural genetics are the primary driving force for improving the performance of a population [20]. Candidate solutions are encoded into string structures called genomes, which are analogous to genetic codes. A fitness function will assign a fitness value to every string in the population. A selection scheme is used to choose a prescribed amount of better fit strings from the current generation while maintaining their diversity. The strings are combined among themselves with a structured, yet randomized information exchange in order to form a new generation of

candidate solutions. Controlled by probability, the exchange is based on three operations that mimic the adaptive process of natural systems: *duplication*, *crossover*, and *mutation*. A duplication operation copies the selected string to the new generation. A crossover operation produces a new string by cutting a string at one point, and swapping in the complementary part from another string. A mutation operation changes the value(s) at random location(s) of the string. Theoretically, the strings of the new generation are created by using bits or pieces of the fittest strings in the previous generation. These bits or pieces of the strings contribute to the overall performance of the strings, and create offspring with higher fitness values. The evolutionary process will not terminate until it satisfies a prescribed criterion, e.g., a maximum number of generations.

The GA offers the following advantages over traditional methods: (a) higher reliability to find the global optima due to its multiple-point exploration capability; (b) seeking good solutions by manipulating the genetic material in genomes without any insight about the problem solved (e.g., continuity, differentiability, etc.)—given only a fitness of each genome; and (c) searching using probability rules.

The earlier appealing features come with a price that the number of fitness function evaluations is higher in GAs than in traditional optimization methods. Still, the GAs are far more efficient than methods such as exhaustive searches or random walks. Clearly, if a problem has broader scope than what can be readily handled by traditional optimization methods, and if the computation involved in evaluating the fitness function is not prohibitively high, then GAs are highly competitive. Automated type synthesis of mechanisms may fit into this category.

3.2 Automated Type Synthesis by GAs. In the area of mechanism design, the topology of a mechanism can be hierarchically defined by the number of links, the number of joints and their types, the connectivity of links and joints, which of the links are the ground and output(s), and which of the joints are associated with the inputs. As mentioned in Section 1, most research on type synthesis considered only the enumeration of mechanisms, a process of determining all possible mechanism topologies to perform a given task or combination of tasks without regard to the dimensions of the components. No report has been found in the literature on automating the mechanism type selection process by numeric optimization methods to seek the most appropriate ones for a clearly defined design task.

To be clear, the term “type synthesis” in this paper refers to choosing the optimum mechanism topology. In practice, some essential dimensional information of a mechanism is of utmost importance to judge whether a particular topology can satisfy the given design task requirements. For example, a generic four-link topology will not be capable of generating continuous relative rotation between two links unless it satisfies Grashof’s law.

The mechanism type synthesis problem will be tackled by explicitly using topology as one design variable in a numeric optimization framework with a GA. The GA is well-suited to such a problem involving topology because it allows the design parameters to vary discretely. Based on different design tasks, the design variable set can contain the essential dimensions of various links. In this way, the GA is used to perform automated type synthesis of mechanisms.

The initial focus is on planar linkage systems with simple revolute joints and a specified number (n) of rigid links. The upper-right triangular portion of the symmetric $n \times n$ link-to-link adjacency matrix, which consists of only 0 and 1 s, is concatenated as a binary string genome representing the system topology. The length (l) of each genome is defined by

$$l = n(n - 1)/2 \quad (3)$$

For example, the Stephenson I six-link mechanism in Fig. 2(a) is represented by the binary string in Fig. 3 whose genome length equals 15. Here the link numbered with 1 is always assumed to be

1 0 0 1 0 1 0 0 1 1 0 0 1 0 1

Fig. 3 A binary string representing Stephenson I six-link mechanism

the ground link of the mechanism, which corresponds to the first $n - 1$ bits in its binary string.

By applying mutation or crossover operators to this string, a different system topology can be obtained. For example, the binary string for a six-link Watt chain can be converted to the representation for a six-link Stephenson chain, and vice-versa, by exchanging the positions of a particular 1 and 0. Thus, different n -link topologies can be explored by the GA. Meanwhile, essential dimensional (and inertial) parameters may be bound to each mechanism topology as resolved from the task definition.

In this work on automated type synthesis, the fitness function evaluation is only based on topology; these simple evaluations can be quickly accomplished on a personal computer (PC) workstation in a sequential way. Alternatively, these evaluation tasks can be distributed over a network of computers, because the fitness of candidate designs can be evaluated independently. This lends itself nicely to implementation in a parallel processing environment, such as the Beowulf PC cluster that was used in this work on topology-based mechanism type synthesis. This distributed synthesis process is outlined in Fig. 4. The cluster was relatively inexpensive to develop, and has allowed us to recently extend this type synthesis research to the more computationally expensive kinematic synthesis problem [21].

In our automated type synthesis, we used both a classic generational GA as well as a steady-state GA. The steady-state GA is very similar to the generational GA except that its genetic operations are applied to a smaller portion of the population continuously; it keeps creating new solutions and adds them to the population while removing less-fit solutions to make room.

The initialization process of these GAs in this paper is accomplished by directly creating random binary string genomes for the initial population. This process is crude, but satisfactory for problems only involving mechanism topology.

In each stage of evolution, it is possible for the GA to create infeasible topologies, e.g., with the wrong DOF or with rigid sub-chains. In the authors’ present work, this problem is handled by assigning a penalty value to a genome representing an infeasible topology. When combined with the direct random initialization process that we used, this method is simple and effective in algorithm implementation. In future, special genetic operators may be developed to create only feasible topologies. The genetic operators must obey simple rules, such as the Gruebler’s criterion that defines the DOF of a mechanism; these rules will be established from and dominated by the mechanism type enumeration techniques.

The earlier synthesis method does not rely on heuristics, which may be case dependent. Heuristics can be included for a particular problem by adding penalty terms in the fitness function, and by introducing constant properties into kernel mechanisms that will not be changed by genetic operators, etc. In these ways, the experience and insight of the designer can be brought into the mechanism topology optimization process.

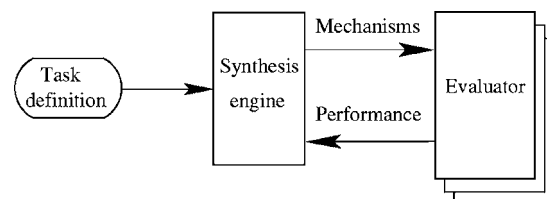


Fig. 4 Distributed architecture for mechanism synthesis

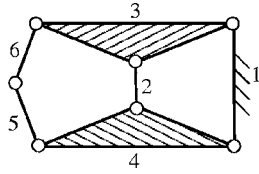


Fig. 5 A relabeling of Stephenson I mechanism

3.3 Challenge of Structural Isomorphism. Two isomorphic mechanisms share the same topological structure; there exists a one-to-one correspondence between their links and kinematic joints such that all incidences are preserved. Structural isomorphism of mechanisms can mathematically be identified by their LAMs (or other methods) [18], because the form of a LAM is dependent on the labeling of links in a mechanism. For example, the structural representation shown in Fig. 5 is obtained from a relabeling of the links of the same mechanism of Fig. 2(a). As a result, the LAM becomes

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \quad (4)$$

Although the structural representations shown in Figs. 2(a) and 5 are for the same mechanism, their LAMs do not assume the same form. However, the two LAMs, Eqs. (2) and (4), are actually related by a congruence transformation; they are identical after a permutation of some specific rows and columns [18].

When a GA initializes and evolves candidate mechanism designs, different binary string genomes in the genotype space are decoded into LAMs in the phenotype space. The LAM and additional decoded dimensional parameters are used to evaluate the fitness of every design. Since different forms of LAMs may actually represent an identical mechanism topology, the performance of the GA can be degraded. How to eliminate those isomorphic genomes without disturbing the evolutionary process is an outstanding research problem. It is viewed as a key point to improving the search efficiency of the GA. Developing special genetic operators is a potential way to solve this problem.

4 Examples of Automated Type Synthesis

4.1 Finding an Optimal Kinematic Inversion for Six-Link Mechanisms. Rao [22] proposes that the accuracy of motion generation by kinematic chains with the same number of links and DOF is dependent on the following factors: (a) the type of link assortment; (b) the type of links connected by each joint, e.g., binary-ternary, etc.; and (c) the size of loops in the chain and their adjacency. In the earlier order, all these factors will hierarchically influence the performance of the chains. The link assortment of a kinematic chain is an orderly sequence of numbers $\{n_2, n_3, \dots, n_i, \dots, n_C\}$, where n_i represents the number of links with i joints in the chain (e.g., n_3 is the number of ternary links), and C is the maximum number of joints in any link of the chain. Design parameters of a link are those required to kinematically specify the link completely, e.g., a binary link has one design parameter (its length).

In our 6 bar example, the criterion that biases different link assortments is not considered since all six-link, 1-DOF mechanisms belong to a unique link assortment, i.e., four binary links and two ternary links. The design parameters of a ground link do not play an effective role in transmitting the motion. As a rule,

fixing a link with fewer joints will generate a better kinematic inversion [22]. Thus, the number of joints on the ground link is selected as one fitness function to be minimized.

However, if there are more distinct links with the same number of joints, one should consider the connectivity of the joints located between the ground link and its adjacent links. Borrowed from information theory, Eq. (5) is used to calculate the energy flow (E) of an inversion in a comparative sense; the greater the value, the better the corresponding kinematic inversion

$$E = - \sum_{i=1}^j \frac{j_i}{J} \log_2 \frac{j_i}{J} \quad (5)$$

where j_i is the joint value of joint i , defined as the number of other joints to which joint i is rigidly connected through one and only one link (except that joints on the ground link will not be counted into the joint value of a grounded joint); j is the total number of joints in the chain; J is the summation of joint values of the chain. For example, for the Stephenson I six-link mechanism in Fig. 2(a), we have $j_1=j_4=j_7=2$, $j_2=j_3=j_5=j_6=3$, $j=7$, and $J=18$, its energy flow is calculated as follows:

$$E = - \left(3 \left(\frac{2}{18} \log_2 \frac{2}{18} \right) + 4 \left(\frac{3}{18} \log_2 \frac{3}{18} \right) \right) = 2.780 \quad (6)$$

In addition to the earlier criteria, all genomes generated by the GA have to satisfy three constraints from the mechanism enumeration techniques [23]:

(a) Gruebler's criterion stipulating the DOF (d) of a mechanism

$$d = 3(n-1) - 2j \quad (7)$$

(b) Number of joints (c) on a single link satisfies

$$2 \leq c \leq C \quad (8)$$

where

$$C = \begin{cases} (n-d+1)/2 & \text{if } d=0,1 \\ \min((n-d-1), (n+d-1)/2) & \text{if } d \geq 2 \end{cases}$$

(c) There are no three-link or five-link rigid subchains, shown in Fig. 6, by, respectively, detecting the following two submatrices in the LAM

$$\begin{bmatrix} \cdot & \cdot & \cdot & 1 & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \quad (9)$$

$$\begin{bmatrix} \cdot & \cdot & \cdot & 1 & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \quad (10)$$

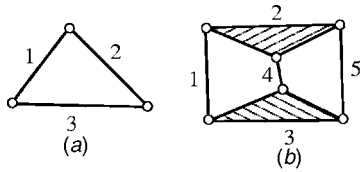


Fig. 6 Structural representations of three-link and five-link basic rigid chains

We used a classic generational GA whose binary string genomes have 15 bits. Other parameters of the GA are listed in Table 1. Note that only 100 generations are needed for the type synthesis problem treated here; in our more recent work on kinematic synthesis [21], more than 1000 generations were required to find globally-optimal solutions.

The GA always found Stephenson I six-link mechanism as its global optima in ten trials. This result conforms to the observation by Rao [22], demonstrating the applicability of the proposed algorithm. Given that there are only five possible solutions, this method for automating the type synthesis of planar mechanisms is encouraging, but not very convincing.

4.2 Optimizing the Rigidity of Eight-Link 3-DOF Planar Parallel Manipulators. Parallel manipulators have drawn much attention recently, as they remedy some of the drawbacks of open-chain robots, such as flexibility, accumulation of mechanical errors, and control problems. The rigidity of a parallel manipulator is due to the existence of multiple closed chains (loops) in the mechanism's structure. In this section, a GA is used to explicitly manipulate mechanism topology as a discrete design variable in order to find the stiffest eight-link, 3-DOF manipulator.

Based on the proposal by Rao [24], a loop-based matrix is adopted as a method that measures the rigidity of kinematic chains in a comparative sense. This method assumes that a closed kinematic chain can be viewed as a system of interconnected springs. Every loop, including the peripheral one, analogies a spring with a stiffness proportional to the number of joints in the loop. To obtain conservative results for the resultant mechanism stiffness, Rao recommends that all the springs or loops are considered to be in series because such a system is always less stiff than the same springs in parallel.

A system of springs in series will have greater resultant stiffness when the springs are of equal stiffness. Consequently, the quantitative criterion for comparing the rigidity of kinematic chains is measured in the form of a symmetric loop-based matrix (K) whose off-diagonal entries are the number of joints common to two loops; all diagonal entries equal zero. Figure 7 shows seven distinct chains for eight-link, 3-DOF mechanisms [23]. For the chain in subfigure (a), if the independent loops are numbered 1 and 2, and loop 3 is its peripheral loop, then its loop matrix is expressed as follows:

Table 1 Parameters of a generational GA for 6-bar inversion problem

Population size	25
Stopping criterion	100 generations
Crossover probability	0.80
Mutation probability	0.01
Scaling scheme	σ truncation
Selection scheme	Tournament plus elitism

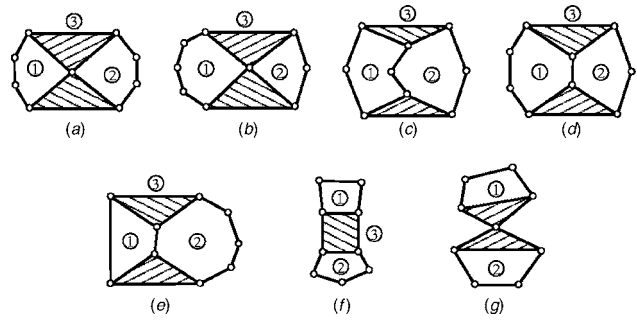


Fig. 7 Seven kinematic chains for eight-link, 3-DOF planar mechanisms

$$K = \begin{bmatrix} 0 & 1 & 4 \\ 1 & 0 & 4 \\ 4 & 4 & 0 \end{bmatrix} \quad (11)$$

Because each joint is counted exactly two times, the sum of each row of matrix K is called the joint value of the corresponding loop. In general, a chain whose loop joint values are more uniform will have greater rigidity. The uniformity is measured by the sum of squares of all loop joint values; the smaller the sum, the greater the uniformity and, therefore, the rigidity. Thus, it is selected as one fitness function.

On the other hand, the number of input joints should be equal to the DOF of the parallel manipulator. To reduce the manipulator inertia, Rao recommends that all drive motors be associated with the joints incident to the ground link [24]. For example, a 3-DOF chain must have a ternary as a ground link so that there are three joints available to be connected to drive motors. This will give a design constraint. The kinematic chains in Figs. 7(f) and 7(g) have fractionated DOF, i.e., they could be split into separate chains, either closed or open, when one of its links is cut into two, such that the sum of the individual DOF equals the DOF of the overall chain [25]. A multi-DOF chain with fractionated DOF cannot satisfy the constraint of placing all motors on grounded joints.

In addition to the above criterion and constraints, Eqs. (7)–(10) in Section 4.1 will be employed as filters through which every genome has to pass. As for the detection of the seven-link basic rigid subchains shown in Fig. 8, the additional submatrices of a LAM needed to be identified are expressed in Eqs. (12)–(14).

$$\begin{bmatrix} \cdot & 1 & 1 & \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & 1 & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \quad (12)$$

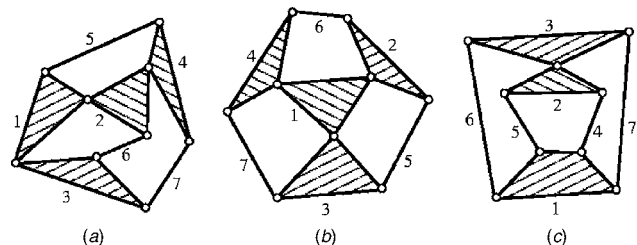


Fig. 8 Three seven-link basic rigid chains

Table 2 Parameters of a steady-state GA for eight-link, 3-DOF parallel manipulators problem

Population size	100
Population overlapping	25
Stopping criterion	200 generations
Crossover probability	0.80
Mutation probability	0.01
Scaling scheme	σ truncation
Selection scheme	Tournament plus elitism

$$\begin{bmatrix} . & 1 & 1 & 1 & . & . & . \\ . & . & . & 1 & 1 & . & . \\ . & . & . & 1 & . & 1 & . \\ . & . & . & 1 & 1 & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & 1 & 1 & 1 & 1 \\ . & 1 & 1 & 1 & . & . & . \\ . & . & . & . & 1 & 1 & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \end{bmatrix} \quad (13)$$

$$\begin{bmatrix} . & . & . & 1 & 1 & 1 & 1 \\ . & 1 & 1 & 1 & . & . & . \\ . & . & . & . & 1 & 1 & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \end{bmatrix} \quad (14)$$

A steady-state GA was used to search for the optimum topology of an eight-link, 3-DOF parallel manipulator, whose genome length is equal to 28. Other parameters are listed in Table 2.

In ten trials, the GA always found the same optima corresponding to the chain in Fig. 7(c), with a ternary as its ground link. This result also verifies Rao's conclusion. It is interesting that the resulting mechanism corresponds to the architecture of a very popular 3-DOF parallel robot manipulator [26]. Given that there are 26 possible kinematic inversions for this problem [23], this example better shows the applicability of the proposed method. It also demonstrates the robustness of the GA, since it always found the global optima from each of ten random initial population.

4.3 Some Comments on Computational Efficiency. Unless mechanism enumeration techniques are incorporated into the GA, one cannot guarantee that infeasible topologies are not generated. However, we have allowed infeasible genomes to exist in each generation because it is believed that even those genomes representing unrealizable physical models will contribute to the GA evolutionary process toward optimum designs. This conjecture has been substantiated by Tay et al. [27] in their work on automated design of dynamic systems, and the same phenomenon was observed in our own work. Furthermore, our foremost objective was to develop a robust and fully automated procedure for type synthesis; maximizing the computational efficiency of this procedure is a subject for further research.

The allowance of many infeasible topologies will have a negative effect on the efficiency of the GA. However, we found the CPU time required for the two topology-based optimization examples to be quite reasonable, especially when compared with a manual approach to these design problems. On average, the first problem requires approximately 20 min of CPU time on a Pentium III PC with 733 MHz CPU and 512 M random access memory, and the second problem averages 50 min over the ten trials. Note that no congruence transformations are applied to the matrices in Eqs. (9), (10), and (12)–(14). Instead, pattern recognition techniques from information theory are used to quickly iden-

tify those mechanisms containing rigid subchains.

Extending this research to kinematic and dynamic synthesis problems will place a greater demand on CPU time because evaluating a mechanism for its time-domain response is generally quite time consuming. It may then be necessary to develop special GA operators that only generate feasible topologies, plus a small number of infeasible topologies, at each new generation.

5 Conclusions

This paper presents a novel method for the automated type synthesis of planar mechanisms and multibody systems. The authors employ a genetic algorithm as an optimization approach to explicitly include mechanism topology as a design variable. The topology of a mechanism is encoded into a binary string genome by concatenating the upper-right triangular portion of its link adjacency matrix. Two examples of topology-based optimization show the applicability of this algorithm to mechanism type synthesis problems.

In relation to other work in the literature, this research fits into the area of computational conceptual design. However, the authors' work is different in a fundamental way: it uses numeric optimization rather than domain-specific rules. The optimization-based approach allows a designer to explore an area for which heuristics are difficult or impossible to establish. It is the first fully automated approach to solving a genuine mechanism type synthesis problem using a numeric optimization framework.

In our latest work [21], we have extended this topology-based optimization to the kinematic synthesis of planar multibody systems. A topology feasibility metric that encapsulates numerous topological requirements was used to guide the GA to generate feasible mechanisms. This significantly increased the efficiency of the process, which exploits the ability of a GA to simultaneously search for feasible topologies in a discrete domain and optimum dimensions in a continuous domain. By combining the GA with a multibody analysis package, this automated synthesis approach can generate mechanism designs that closely match the desired kinematic characteristics.

Acknowledgment

Financial support of this research by the Natural Science and Engineering Research Council of Canada is gratefully acknowledged. The authors would also like to thank Professor Y. Hwang at National Chung Cheng University, Taiwan, R.O.C., for providing a package that enumerates kinematic chains. The authors used the GALIB genetic algorithm package, written by Matthew Wall at the Massachusetts Institute of Technology.

References

- [1] Olson, D. G., Erdman, A. G., and Riley, D. R., 1985, "A Systematic Procedure for Type Synthesis of Mechanisms with Literature Review," *Mech. Mach. Theory*, **20**(4), pp. 285–295.
- [2] Erdman, A. G., and Sandor, G. N., 1990, *Mechanism Design: Analysis and Synthesis*, 2nd ed., Prentice Hall, Englewood Cliffs, NJ.
- [3] Erdman, A. G., 1995, "Computer-Aided Mechanism Design: Now and the Future," *ASME J. Mech. Des.*, **117**, pp. 93–100.
- [4] Norton, R., 1999, *Design of Machinery*, 2nd ed., WCB/McGraw-Hill, Boston, MA.
- [5] Soni, A. H., Dado, M. H. F., and Weng, Y., 1988, "An Automated Procedure for Intelligent Mechanism Selection and Dimensional Synthesis," *ASME J. Mech., Transm., Autom. Des.*, **110**, pp. 130–137.
- [6] Wang, Y., and Yan, H., 2002, "Computerized Rules-Based Regeneration Method for Conceptual Design of Mechanisms," *Mech. Mach. Theory*, **37**, pp. 833–849.
- [7] Yan, H., 1998, *Creative Design of Mechanical Devices*, Springer-Verlag, Singapore.
- [8] Chiou, S. J., and Kota, S., 1999, "Automated Conceptual Design of Mechanisms," *Mech. Mach. Theory*, **34**, pp. 467–495.
- [9] Kong, X., and Gosselin, C. M., 2004, "Type Synthesis of 3-DOF Translational Parallel Manipulators Based on Screw Theory," *ASME J. Mech. Des.*, **126**, pp. 83–92.
- [10] Kong, X., and Gosselin, C. M., 2004, "Type Synthesis of 3-DOF Spherical Parallel Manipulators Based on Screw Theory," *ASME J. Mech. Des.*, **126**, pp. 101–108.

- [11] Bendsoe, M. P., and Sigmund, O., 2003, *Topology Optimization: Theory, Methods and Applications*, Springer, Berlin.
- [12] Kawamoto, A., Bendsoe, M. P., and Sigmund, O., 2004, "Planar Articulated Mechanism Design by Graph Theoretical Enumeration," *Struct. Multidiscip. Optim.*, **27**, pp. 295–299.
- [13] Felner, C. L., 2003, "Topology Optimization of Rigid Body Mechanisms," Master thesis, Technical University of Denmark, Lyngby, Denmark.
- [14] Eberhard, P., Schiehlen, W., and Bestle, D., 1999, "Some Advantages of Stochastic Methods in Multi-Criteria Optimization of Multibody Systems," *Arch. Appl. Mech.*, **69**, pp. 543–554.
- [15] Leger, C., 1999, "Automated Synthesis and Optimization of Robot Configurations: An Evolutionary Approach," Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA.
- [16] Rao, A. C., 2000, "A Genetic Algorithm for Topological Characteristics of Kinematic Chains," *ASME J. Mech. Des.*, **122**, pp. 228–231.
- [17] Fang, W., 1994, "Simultaneous Type and Dimensional Synthesis of Mechanisms by Genetic Algorithms," *Mechanism Synthesis and Analysis ASME DE-70*, pp. 35–41.
- [18] Tsai, L. W., 2001, *Mechanism Design: Enumeration of Kinematic Structures According to Function*, CRC Press, Boca Raton, FL.
- [19] Shi, P., and McPhee, J. J., 2000, "Dynamics of Flexible Multibody Systems using Virtual Work and Linear Graph Theory," *Multibody Syst. Dyn.*, **4**, pp. 355–381.
- [20] Goldberg, D. E., 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA.
- [21] Liu, Y., 2004, "Automated Type and Dimensional Synthesis of Planar Mechanisms Using Numeric Optimization with Genetic Algorithms," Ph.D. thesis, University of Waterloo, Waterloo, Ontario, Canada.
- [22] Rao, A. C., 1998, "Topology Based Rating of Kinematic Chains and Inversions Using Information Theory," *Mech. Mach. Theory*, **33**, pp. 1055–1062.
- [23] Hwang, W., and Hwang, Y., 1992, "Computer-Aided Structure Synthesis of Planar Kinematic Chains with Simple Joints," *Mech. Mach. Theory*, **27**, pp. 189–199.
- [24] Rao, A. C., 1997, "Platform-Type Planar Robots: Topology-Based Selection for Rigidity and Work Space," *J. Rob. Syst.*, **14**, pp. 355–364.
- [25] Davies, T., 1968, "An Extension of Manolescu's Classification of Planar Kinematic Chains and Mechanisms of Mobility $M \geq 1$, using Graph Theory," *J. Mech.*, **3**, pp. 87–100.
- [26] Merlet, J. P., 2000, *Parallel Robots*, Kluwer Academic, Dordrecht.
- [27] Tay, E. H., Flowers, W., and Barrus, J., 1998, "Automated Generation and Analysis of Dynamic System Designs," *Res. Eng. Des.*, **10**, pp. 15–29.