



Size and configuration syntheses of rigid-link mechanisms with multiple rotary actuators using the constraint force design method

Jae Chung Heo ^a, Gil Ho Yoon ^{b,*}

^a Graduate School of Mechanical Engineering, Hanyang University, Republic of Korea

^b School of Mechanical Engineering, Hanyang University, Republic of Korea

ARTICLE INFO

Article history:

Received 4 June 2012

Received in revised form 3 January 2013

Accepted 11 January 2013

Available online 13 February 2013

Keywords:

Constraint force design method

Rigid body mechanism

Hybrid genetic algorithm

ABSTRACT

This study presents a new synthetic approach for path-generating or function-generating rigid-body mechanisms in the framework of a new hybrid genetic algorithm. In spite of some advantages of synthesis methods in designing rigid-body mechanisms, some inherent issues remain in determining optimal sizes and configurations of rigid links with multiple rotary actuators. To alleviate these difficulties and limitations, we improve our previous contribution, called the constraint force design method, by parameterizing and optimizing the existence of links, the x–y locations of joints modeled by unit masses (particles), and the kind of selection between rigid and string links, using binary, integer, and binary design variables, respectively. This new genotype coding system for GA for the integer and binary phenotypes makes it possible to use a smaller number of unit masses to synthesize manifold configurations of rigid-body mechanisms. In addition, the locations and types of rotary actuators are parameterized with additional integer design variables for the synthesis of realistic rigid-body mechanisms. Furthermore, for efficient optimization of GA, a new h-GA integrated with the Sequential Quadratic Programming (SQP) optimization algorithm is also developed to optimize the locations of joints. To demonstrate the validity of the present constraint force design method, several mechanism synthesis problems are solved.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

The optimal synthesis of rigid-body mechanisms has long been of interest to many scientists and engineers [1–14]. One of the main synthesis problems of rigid-body mechanisms is determining the optimal number of rigid links and the optimal locations and types of joints for user-specified path generation [1–7], function generation [8–10], or guidance [11–14] of a follower, as shown in Fig. 1. Despite the existence of many synthesis approaches, however, it is still difficult to optimize a rigid-body mechanism without given initial designs [1–14]. Thus, the objective of this study is to develop a black-box approach that can determine an optimal number of rigid/string links and optimal locations of rotary actuators and joints not restricting an initial given rigid-body mechanism; our previous contribution to this important mechanism-synthesis problem [15] was confined to size and configuration optimization of a rigid mechanism with a standard GA. In short, the following topics are examined in this study.

Research object 1 Robust size and configuration (topology) optimization of a rigid mechanism

Research object 2 Optimization of the locations of rotary actuators and joints

Research object 3 Link type optimization between rigid and string

Research object 4 Development of a new efficient hybrid genetic algorithm optimized for rigid mechanism synthesis

* Corresponding author at: School of Mechanical Engineering, Hanyang University, Seoul, Republic of Korea.

E-mail addresses: ghy@hanyang.ac.kr, gilho.yoon@gmail.com (G.H. Yoon).

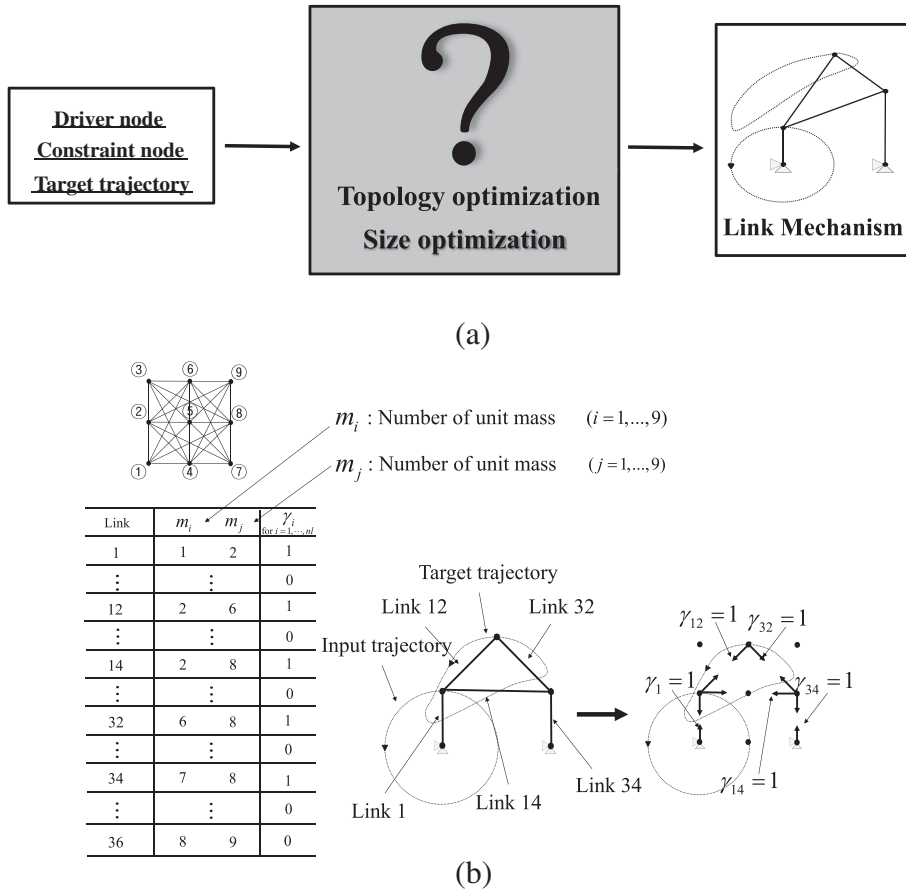


Fig. 1. (a) Black box optimization for a rigid-body mechanism (driver node: joints moved by actuators, constraint node: joints for boundary condition, input trajectory: the curve of driver node, and target trajectory: the curve of follower node), (b) topology optimization of the constraint force design method. The design variables are denoted by γ , and the subscripts i and j represent the connecting i -th and j -th unit masses. See [15] for more details.

If we were confined to a specific, well-known, and well-studied mechanism, such as a four-bar mechanism whose initial configurations, i.e., the number of joints and the connectivity of links, are known a priori, it would be an optimization problem involving changing the lengths of rigid links or the locations of joints to fulfill the design parameters of a given engineering object; such a problem can be solved relatively easily. However, by developing a new black-box method, we hope to optimize the number and type of rigid-links and the locations and types of rotary actuators, as shown in Fig. 1. Note that conventional design methods require manual alternations of the numbers and types of links, joints, and actuators.

To achieve these objectives, this paper examines our previous contribution, the constraint force design method [15], for the purpose of determining the optimal topological configuration of a rigid-body mechanism, to allow simultaneous size and configuration optimization by parameterizing the x-y locations of joints, the geometric constraints, and the existence of rigid links of the SHAKE algorithm [16,17]. (For the SHAKE algorithm, a well-established method, see [16,17]). Additionally, a new hybrid genetic algorithm employing the finite difference method for sensitivity analysis is implemented in order to effectively determine the optimal joint locations of the best individual from the GA with a smaller population and less computation time. The validity of the present extension is shown by synthesizing a set of rigid links and rotary actuators for some path- or function-generating mechanisms.

As stated above, many analytical and numerical methods have been proposed for the structural optimization of rigid-body mechanism synthesis [18–22]. As an exhaustive review of such studies regarding the mechanism synthesis is beyond the scope of this research, only a few studies are mentioned here from an optimization point of view. By extending the existing incident degree code approach, Zhang and Breteler applied the mechanism identification method for the topology optimization of a rigid-body mechanism [18]. Kawamoto et al. [19] used the branch-and-bound method to optimize articulated mechanisms simulated by truss-based ground structure representation. Minnaert et al. presented an optimal synthesis method for planar mechanisms using gradient-based optimization algorithms [20]. Kim et al. also proposed an automatic mechanism synthesis method by designing optimal connections among rigid blocks using a gradient-based optimizer [21]. Wang and Yan presented a rule-based computerized symbolic carrier regeneration method for mechanical conceptual design [22]. In addition to the references mentioned above, there are numerous engineering methods and principles for designing a rigid-body mechanism based on kinematic analysis. Recently, we made a contribution to this important engineering problem based on kinetic analysis [15]; this research develops new parameterization methods for the locations of rotary actuators and joints and the link type based on a hybrid GA. In short, many innovative synthesis

methods have been proposed to address the synthesis problems associated with rigid-body mechanisms, and this study also contributes to this important research topic in the framework of the constraint force design method.

To the best of our knowledge, most existing synthesis methods are based on the kinematic analysis of an initial mechanism with a fixed number of links and joints. In other words, graphical or analytical techniques are commonly used to express the trajectory of a given mechanism with respect to several variables, such as the lengths, angles, and positions of rigid links [23–25]. Optimization algorithms are then applied to determine the optimal sizes of rigid links; it is possible to use a gradient-based optimizer or evolutionary algorithm such as a genetic algorithm (GA) or a simulating annealing (SA) algorithm [26–28]. From a structural optimization point of view, these approaches can be classified as *size optimization*, which does not allow topological or configuration changes. To overcome the restriction of a specified topology during optimization, we proposed a new method, called the constraint force design method, based on a GA framework [15]. Unlike other synthesis methods or approaches, this new synthesis method is based on the kinetic analysis of unit masses connected by artificial forces maintaining relative lengths (the SHAKE algorithm), as shown in Fig. 1. Parameterizing the existence of these artificial forces among unit masses makes it easy to represent topological changes of rigid-body mechanisms. However, because the initial positions of these unit masses are fixed, the previous constraint force design method has two limitations.

Limitation 1 Denser unit masses must be used to determine complex mechanisms, inevitably increasing the number of design variables, which is an unfavorable situation from a GA point of view. Increasing the number of unit masses or resolution results in a longer time required to obtain an optimal design (Fig. 2(a)).

Limitation 2 Given an initial distribution of unit masses representing joints, there is a high probability that no rigid link mechanism satisfies the target trajectory. Because the original constraint force design method uses evenly distributed masses, it is impossible to implement the mechanism shown in Fig. 2(b).

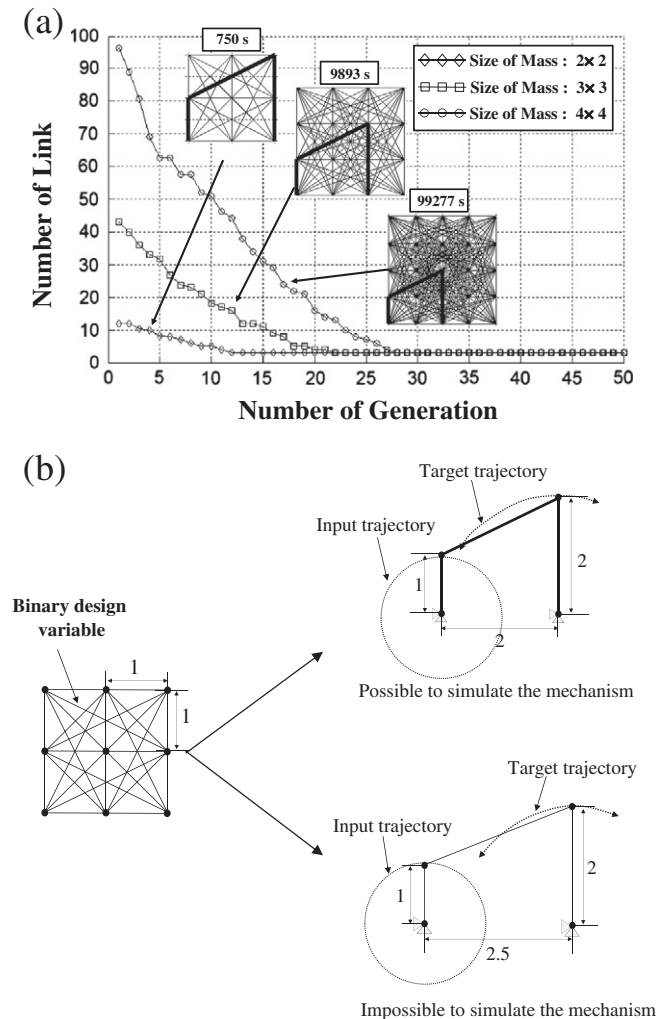


Fig. 2. The issues associated with the constraint force design method [15].

The limitations discussed above and observations made in our previous study [15] highlight the need for further engineering developments and extensions. This study extends this constraint force design method by integrating the concepts of topology optimization [29–32] and size optimization [33–35] with that of the constraint force design method. Furthermore, techniques for determining the optimal locations and types of rotary actuators are enhanced in this study.

The remainder of this paper is organized as follows. After describing the basic analysis of rigid-body mechanisms using the SHAKE algorithm, Section 2 presents a new optimization formulation using integer design variables parameterizing the locations of the unit masses, and binary design variables parameterizing the existence of the artificial forces of the SHAKE algorithm. Section 3 presents the hybrid GA method, which uses a sequential combination of GA and the sensitivity-based optimization algorithm (SQP). Furthermore, a heuristic modification of the SHAKE algorithm is proposed to simulate string connections among unit masses. Section 4 presents several synthesis examples of rigid link mechanisms. Some examples optimizing the type and locations of rotary actuators are also presented. Finally, our findings and topics for future research are summarized and discussed in Section 5.

2. Dynamic simulation of unit masses and configuration (topology) synthesis of rigid link mechanisms

2.1. Dynamic simulation of unit masses: Lagrangian formulation and the SHAKE algorithm

Unlike most rigid-body mechanism synthesis methods based on kinematics, the present constraint force design method is based on kinetics [15]. In this new design method, unit masses and artificial constraint forces maintaining the relative distances among the unit masses simulate joints and rigid links, respectively. Among several numerical approaches used to calculate the constraint forces, the SHAKE algorithm, which is the simplest algorithm, is applied (see [16,17] and the Appendix A for a basic description of the SHAKE algorithm). This sub-section briefly describes this algorithm and the Lagrangian formulation for the sake of completeness.

The set of dynamic equations of masses with forces is described by Newton's second law and the Lagrange equation as follows:

$$L = \frac{1}{2} \sum_i m_i \dot{\mathbf{r}}_i^2 - U(\mathbf{r}_i) \quad (\mathbf{r}_i^2 = \mathbf{r}_i \cdot \mathbf{r}_i) \quad (1)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\mathbf{r}}_i} \right) - \frac{\partial L}{\partial \mathbf{r}_i} = \sum_{k=1}^{N_i^{RL}} \lambda_k (\mathbf{r}_{ij}^2 - d_{ij}^2) \quad (i = 1, 2, 3, \dots, n) \quad (\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j, \mathbf{r}_{ij}^2 = \mathbf{r}_{ij} \cdot \mathbf{r}_{ij}) \quad (2)$$

where L is the Lagrangian (the summation of kinetic energy and potential energy U). The position and velocity of the i -th mass with a constant mass m_i are denoted by \mathbf{r}_i and $\dot{\mathbf{r}}_i$, respectively. The current distance vectors and the imposed relative distances between the i -th and j -th masses are denoted by \mathbf{r}_{ij} and d_{ij} , respectively. The magnitude of \mathbf{r}_{ij} is the length between the i -th and j -th masses. The total number of unit masses is n , and the number of length constraints of the i -th mass is N_i^{RL} . The Lagrangian multiplier for the k -th length constraint is denoted by λ_k in Eq. (2). From the Lagrange equation above, the general equations (Newton's law) regarding the motion of each mass can be obtained as follows:

$$m_i \ddot{\mathbf{r}}_i = \mathbf{f}_i + \mathbf{g}_i \quad (f_i \equiv 0) \quad (i = 1, 2, 3, \dots, n). \quad (3)$$

Constraint force to maintain relative lengths:

$$\mathbf{g}_i = - \sum_{k=C_i}^{N_i^{RL}} \lambda_k \nabla_i \sigma_k \quad \text{and} \quad \nabla_i \sigma_k \equiv 2\mathbf{r}_{ij} \quad (4)$$

where the forces applied to the i -th mass independent of the other masses and the summed force acting on the i -th mass from the length constraints for rigid links are denoted by \mathbf{f}_i and \mathbf{g}_i , respectively [15–17]. For the rigid-body mechanism design, the force \mathbf{f}_i is set to zero. The acceleration vector of the i -th mass is $\ddot{\mathbf{r}}_i$, and the Lagrangian multiplier is denoted by λ_k . To solve the above dynamic equations efficiently, the well-established Leapfrog method and the Verlet method are employed [36,37]. (See the Appendix A for a brief description of the SHAKE algorithm for the auxiliary force \mathbf{g}_i .)

2.2. Constraint force design method: The topological optimization of a rigid-body mechanism

Most synthesis methods for rigid-body mechanisms rely mainly on analytical or graphical approaches to parameterizing and optimizing the locations of joints or the lengths of rigid links for a given initial and fixed topological configuration. In other words, most synthesis methods are incapable of exploring optimal configurations of rigid links, such as 1) the optimal number of rigid links, 2) the optimal connections or connectivities among rigid links, and 3) the optimal number and type of rotary actuators. To overcome these challenges from the fixed configurations of the other synthesis methods, we use our previously developed approach called the constraint force design method [15]. The constraint force design method parameterizes the existence of rigid links with binary variables, as well as the locations of joints with integer variables, and optimizes them using a heuristic optimization algorithm (GA) as

shown in Figs. 1 and 2. This method has the following three main features pertaining to the search for optimal configurations of rigid links and joints.

- Feature 1 The artificial forces of the SHAKE algorithm simulate rigid links among unit masses when the unit masses simulate the joints. It is based on the kinetic analysis of a given mechanism instead of kinematic analysis, and the unit masses represent revolute or prismatic joints depending on the boundary condition.
- Feature 2 An evolutionary optimization algorithm GA is employed to solve the synthesis problem by parameterizing the existence of artificial forces with binary variables.
- Feature 3 Due to feature 2, the initial distributions of unit masses can also confine the design space, and there is a possibility that the global optimal solution will not be obtained, as shown in Fig. 2. To overcome this design space limitation, integer variables parameterizing the locations of joints are introduced. Thus, the design variables of the GA are a combination of the binary and integer design variables.

Applying the constraint force design method makes it possible to search for an optimal configuration with the fixed locations of unit masses. For this purpose, the following binary design variables, which indicate the existence of rigid links, are optimized by GA: The genotype of Fig. 4(a):

$$X = \{x_i | x_i \in [0, 1] \text{ and } i = 1, \dots, nl\} \quad nl = \frac{n(n-1)}{2} \quad (5)$$

where the total number of rigid links is nl . It was possible to construct optimal configurations of a rigid-body mechanism by searching for an optimal genotype that minimizes objectives and the number of links. Nonetheless, there are some limitations (Fig. 2), and it seems that a classical GA is not suitable for this kind of structural optimization problem. Therefore, the extended constraint force design method is developed in this study through the addition of some new concepts in order to determine an optimal rigid-body mechanism with more realistic simulation conditions.

2.3. Optimization formulation

A fitness function must be devised to incorporate a GA into the optimization problem. In parallel with the process of discovering a rigid-body mechanism, we also seek to generate planar rigid-body mechanisms with fewer rigid links. Thus, the two objective functions, the norm of the difference between the target trajectory and the trajectory of a current design and the number of links, are combined as follows:

$$\Phi = \sum_{k=1}^{N_p} \|\mathbf{r}_W^k - \mathbf{r}_{W,Target}^k\| + \alpha \sum_{k=1}^{nl} \gamma_k \quad (6)$$

where the k -th calculated trajectory of a work point of a given design and the k -th target trajectory of the work point are denoted by \mathbf{r}_W^k and $\mathbf{r}_{W,Target}^k$, respectively. The number of spatial points and the number of design variables are denoted by N_p and nl , respectively. The first term in Eq. (6) corresponds to the difference between the path of the work point and the target trajectory as shown in Fig. 3(a). The target trajectory is defined as the position vector of one end point of a follower of a specific mechanism, and the input trajectory is defined as the motions of one end point of a crank. Because the present constraint force design method uses discrete motions of a given mechanism, equally sampled points are used for the input trajectory, and their corresponding output points are set to the target trajectory, in the present research. The second term with a positive scaling factor of α is added to minimize the usage of links. Without the second term, some rigid-body mechanisms with unnecessary branches can be obtained, as shown in Fig. 3(b). In addition, in order to find optimal rigid-body mechanisms, some appropriate GA parameters must be chosen, such as the population size, the number of target points, and the scaling factor [15].

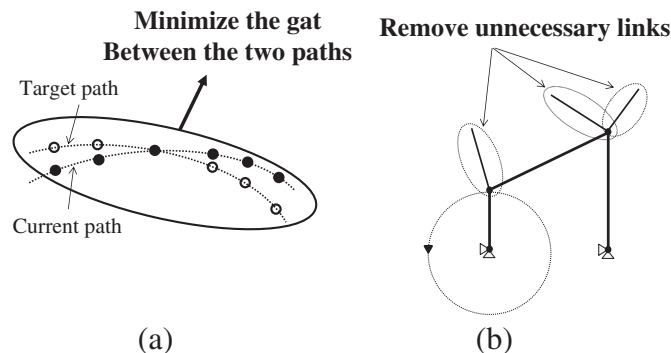


Fig. 3. An optimization formulation.

3. Size and topological optimization of rigid/string mechanisms with actuators

This section is devoted to extending the constraint force design method for size and configuration (topology) optimization of rigid/string links and rotary actuators. For these purposes, parameterization schemes for the x–y locations of joints and the locations and types of rotary actuators are presented. To determine the rigid-string link mechanism, a modified SHAKE algorithm is also developed here. A new hybrid GA with elitism that preserves the best individual in the GA (see any book about GAs) and the sensitivity-based optimization algorithm (SQP) are developed to enhance the performance of the GA.

3.1. Parameterization of the x–y locations of joints and the locations and types of rotary actuators

3.1.1. X–Y locations of joints

As shown in Fig. 2, the original constraint force design method is limited in that the joint locations are fixed; the initial joint distribution confines the spatial design space in which the method can search for solutions. To relax this confinement, a parameterization of the joint locations with integer variables is proposed [15]. In this extension, binary design variables are introduced to represent rigid links, and integer design variables are introduced to represent the joint locations, as follows (see Fig. 4(b)):

$$\gamma = \left\{ \begin{array}{l} \gamma_i | \gamma_i \in [0, 1] \text{ for } i = 1, \dots, nl \text{ (The existence of links) } nl = n(n-1)/2 \\ \text{and } \gamma_i \in [0, \dots, \text{resol} - 1, \text{resol}] \text{ for } i = nl + 1, \dots, nl + 2 \times n \text{ (The re-positioning of joints)} \end{array} \right\} \quad (7)$$

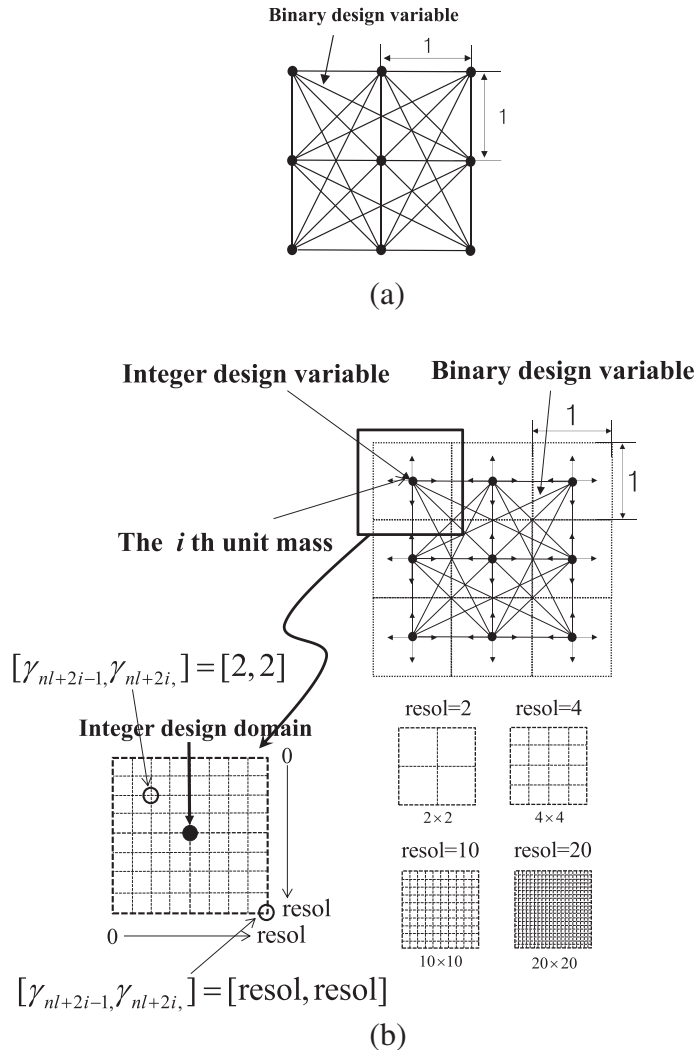


Fig. 4. Parameterization of the existence of joints and the locations of the joints. (a) Parameterization of the existence of links of the original constraint force design method. (b) Simultaneous parameterization of the locations of joints with integer variables.

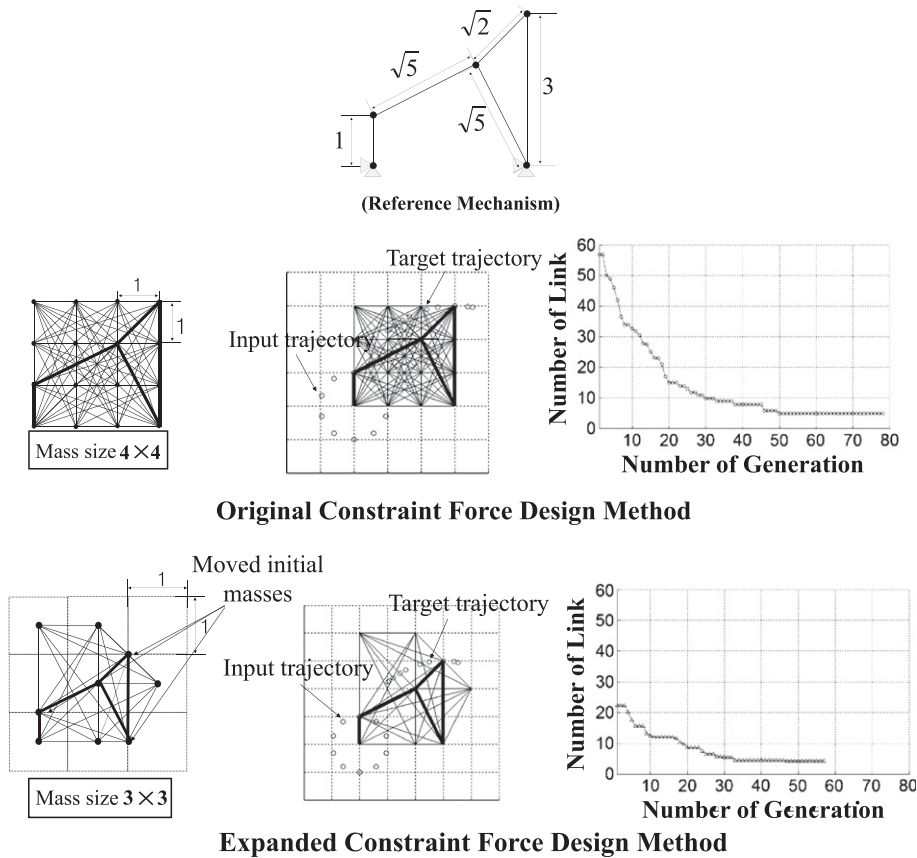


Fig. 5. Examples of the classical constraint force design method and the extended constraint force design method with integer variables.

where nl is the number of rigid links, and $resol \times resol$ is the number of relocation points where the unit mass can move, as shown in Fig. 4; hereafter, the boxes in which the joints can move are outlined by dotted lines, as shown in Fig. 4(b), and the number of added design variables in (7) is $2n$ for the parameterization of the x and y locations of n unit masses. With this extension of Eq. (7), it is possible to explore an expanded design space with a limited number of rigid links. From a structural optimization point of view, this extension can be regarded as the introduction of size optimization [38], and the parameterization of Eq. (7) enables size and topological optimization of rigid-body mechanisms.

3.1.2. An illustrative example

The extended parameterization introduced above can be understood by considering the design example in Fig. 5. To represent the exact positions of the reference rigid-body mechanism, a 4×4 unit mass distribution (120 design variables) must be used in the original constraint force design method. With only the binary design variables, a 3×3 unit mass distribution (54 (36 + 18) design variables) is used in the modified constraint force design method with the parameterization of Eq. (7). From a computational point of view, the analysis time for Newton's equation is significantly reduced. Consequently, it is possible to solve this difficult

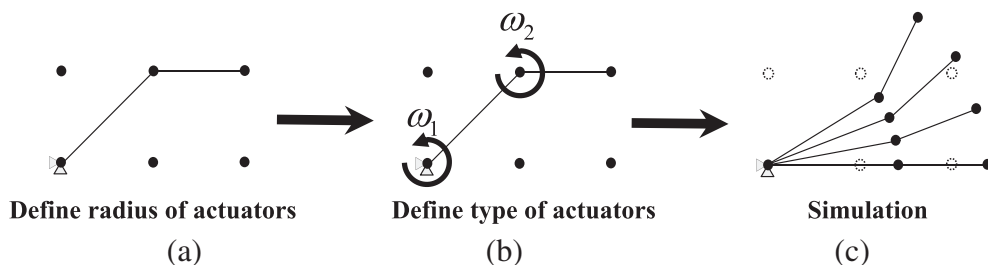


Fig. 6. Simulation of two rotary actuators.

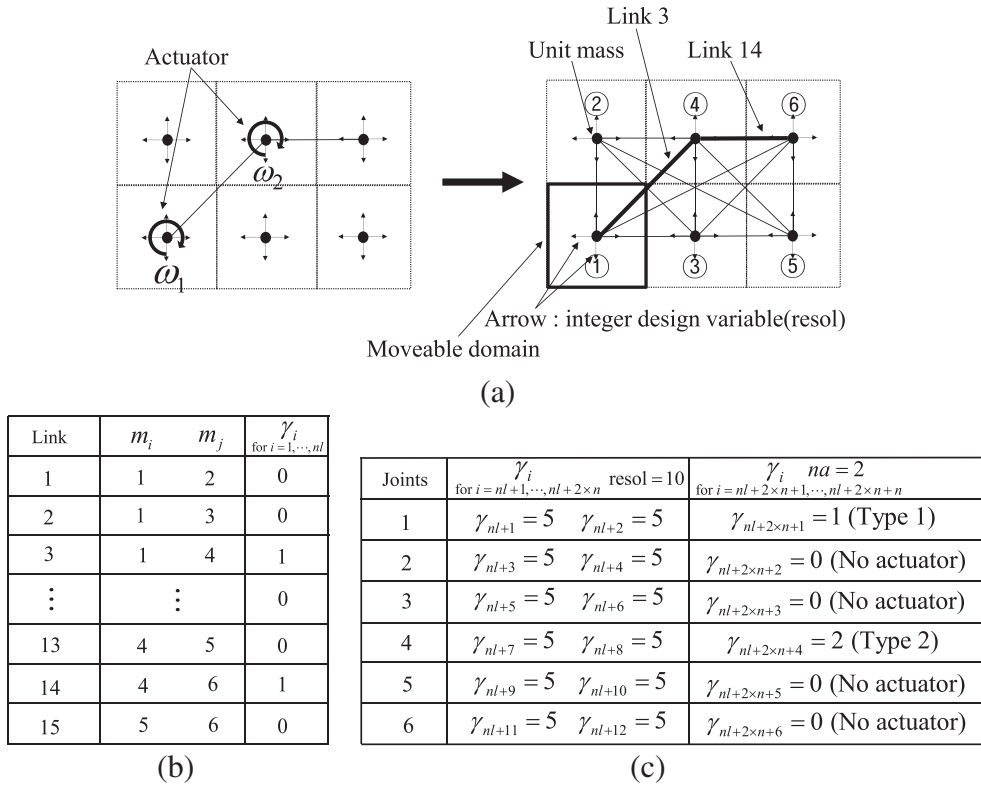


Fig. 7. Design variables for the mechanism illustrated in Fig. 6.

engineering problem faster than ever by reducing the number of design variables. However, even with this extension of the original constraint force design method, it is still limited in that the design space depends on the range of the integer variables, resol. A compromise between the accuracy and efficiency of the schemes should be made for practical applications.

3.1.3. Locations of rotary angular actuators

In addition to the extension of the constraint force design method to determining the optimal joint locations, it is also possible to parameterize the locations of multiple rotary actuators after resolving the motion confliction issue illustrated in Figs. 6 and 8. To the best of our knowledge, the choice of optimal locations of multiple rotary actuators is an important design issue in practical

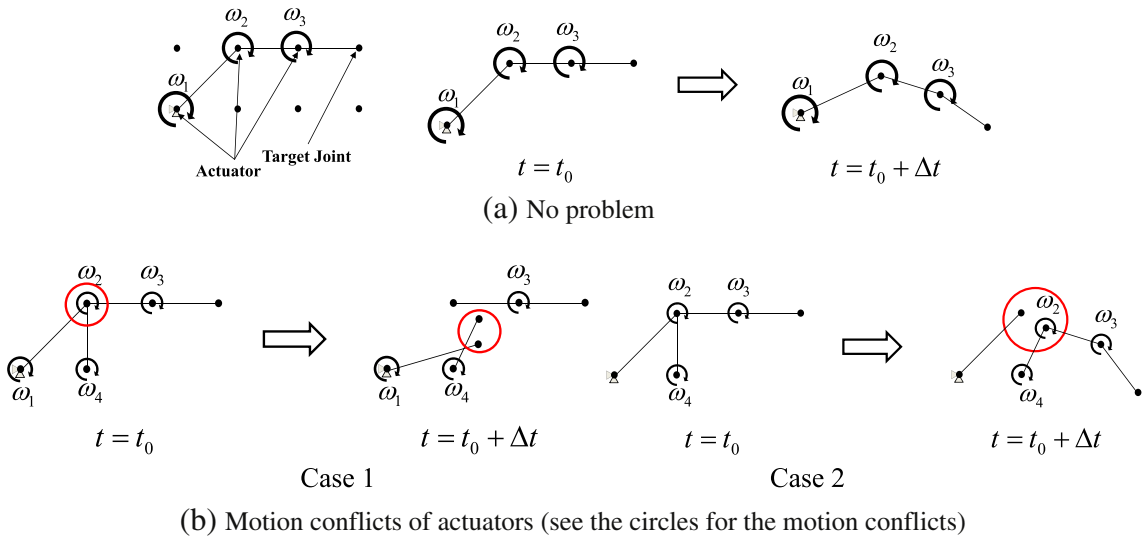


Fig. 8. Cases of motion conflict.

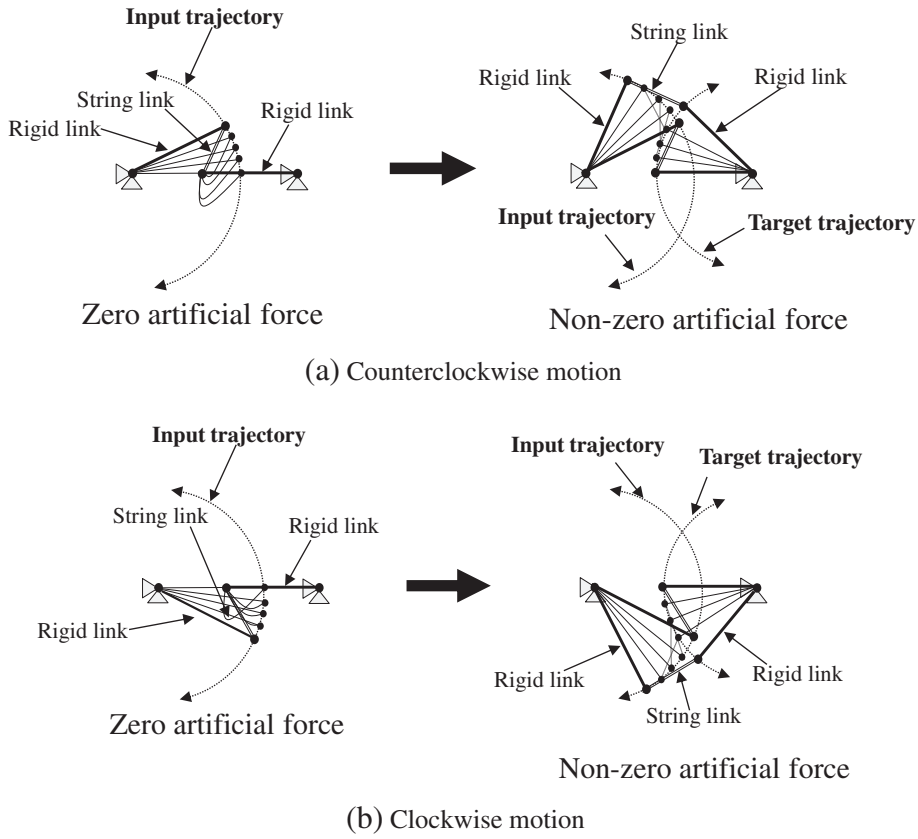


Fig. 9. Counterclockwise and clockwise motions of two rigid bars connected by a string.

fields such as automotive engineering and robotics. Nevertheless, there are few engineering tools that can be used to determine the optimal locations of rotary actuators and the configurations of rigid links simultaneously. In the present constraint force design method, the locations and types of rotary actuators can be taken into consideration by adding n integer variables to the phenotype of Eq. (7).

$$\gamma = \left\{ \begin{array}{l} \gamma_i | \gamma_i \in [0, 1] \text{ for } i = 1, \dots, nl \text{ (The existence of links)} \\ \text{and } \gamma_i \in [0, \dots, \text{resol} - 1, \text{resol}] \text{ for } i = nl + 1, \dots, nl + 2 \times n \text{ (The re-positioning of joints)} \\ \text{and } \gamma_i \in [0, \dots, na - 1, na] \text{ for } i = nl + 2 \times n + 1, \dots, nl + 2 \times n + n \text{ (The locations and types of actuators)} \end{array} \right\} \quad (8)$$

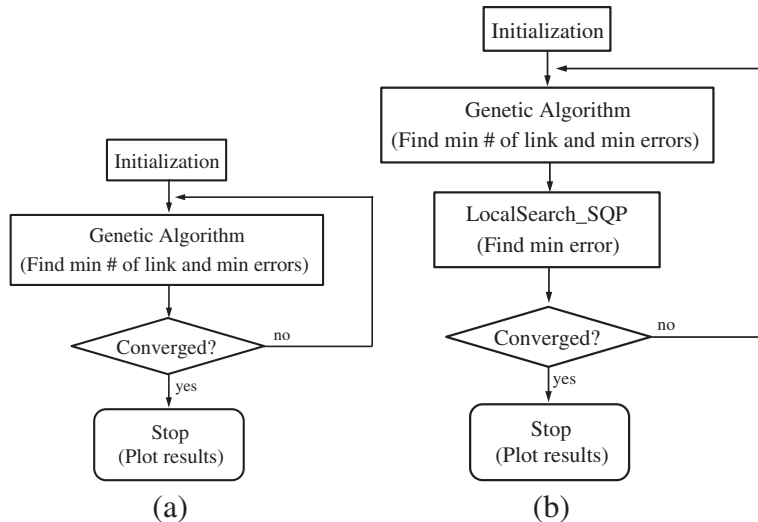


Fig. 10. (a) Flowchart of a simple GA, (b) Flowchart of the proposed hybrid GA.

Table 1

Parameters used for the numerical examples.

	Ex. 1	Ex. 2	Ex. 3	Ex. 4-1	Ex. 4-2	Ex. 4-3	Ex. 5-1	Ex. 5-2
Step (# of target points)	10	10	10	10	10	10	10	10
Population size	100	100	300	400	300	400	150	200
Crossover	Two points	Two points	Two points	Two points	Two points	Two points	Two points	Two points
Mutation rate	0.05	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Design variable type	Binary	Binary and Integer	Binary and Integer	Binary	Binary	Binary and Integer	Binary and Integer	Binary and Integer

Note that as na actuators can be attached to n joints, the associated design variables vary from 0 to na ; 0 indicates that there is no actuator, and the other values indicate the existence of actuators. Before optimization, the meanings of the values from 0 to na are predefined. Examples of parameterization and the design variables are given in Figs. 6 and 7.

With the new parameterization of Eq. (8), we observe that some awkward and unphysical situations occur when more than two rotary actuators describe the motion of one node, as shown in Case 1 of Fig. 8(b). This is due to the conflicting motion descriptions of the rotary actuators. Similar situations are observed when more than two rotary actuators are attached to the link with the clamp condition, as in Case 2 of Fig. 8(b). In our simulation, when these cases are observed, we ignore the motion descriptions of these rotary actuators. From a GA point of view, neglecting these descriptions, which are mathematically discontinuous, does not result in complications.

3.2. An extension to simulate a string

As emphasized before, an important characteristic of the present constraint force design method is that it is based on kinetic analysis rather than kinematic analysis. Thus, the non-zero artificial force \mathbf{g}_i is applied when the distance between the two unit masses is larger or smaller than the corresponding link length in order to maintain the relative distance between the two unit masses [16,17]. This basic concept of the SHAKE algorithm indicates that it is possible to simulate a string or bar having only the

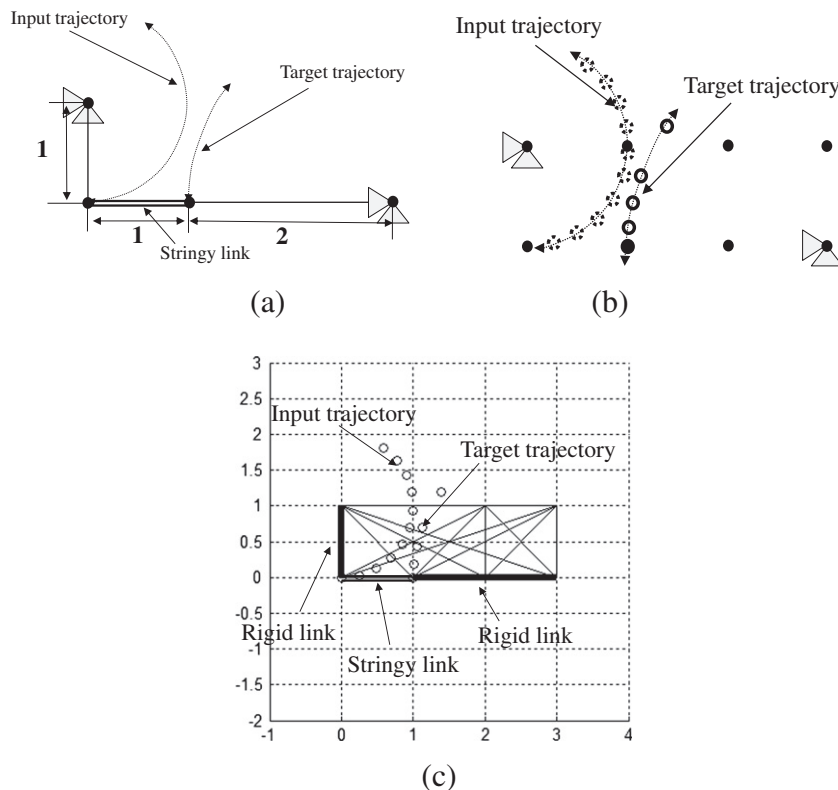


Fig. 11. Synthesis of a string-rigid link mechanism: (a) Reference string-rigid link mechanism, (b) distribution of eight masses, relative displacement inputs, and prescribed target displacements of the string-rigid link mechanism, and (c) the obtained optimal design.

resistance-to-tension force, as shown in Fig. 9. To represent this string element, new binary design variables are added, as in Eq. (9), and the equation for \mathbf{g}_i is modified as follows:

$$\gamma = \left\{ \begin{array}{l} \gamma_i | \gamma_i \in [0, 1] \text{ for } i = 1, \dots, nl \text{ (The existence of links)} \\ \text{and } \gamma_i \in [0, 1] \text{ for } i = nl + 1, \dots, nl + nl(1 : \text{String}, 0 : \text{Rigid link}) \end{array} \right\} \quad (9)$$

$$\mathbf{g}_i = \left\{ \begin{array}{ll} -\sum_{k=C_i}^{N^{RL}} \lambda_k \nabla_i \sigma_k & \text{if } \gamma_i = 1, \gamma_{i+nl} = 1, r_{ij}^2 \geq d_{ij}^2 \text{ (String link)} \\ 0 & \text{if } \gamma_i = 1, \gamma_{i+nl} = 1, r_{ij}^2 < d_{ij}^2 \text{ (String link)} \\ -\sum_{k=C_i}^{N^{RL}} \lambda_k \nabla_i \sigma_k & \text{if } \gamma_i = 1, \gamma_{i+nl} = 0, \text{ (Rigid link)} \\ 0 & \text{if } \gamma_i = 0, \text{ (No link).} \end{array} \right. \quad (10)$$

When the positions and types of rotary actuators or the locations of joints are of interest, Eq. (9) must be modified to include the additional design variables. Fig. 9 shows the simulations of two rigid links connected by a string. Conclusively, with this extension it is possible to design a mechanism with rigid links and string links simultaneously, a process that is quite difficult using existing synthesis methods.

3.3. Hybrid GA

From an optimization algorithm point of view, the binary design variables parameterizing the existence of rigid links and the integer design variables parameterizing the locations of joints or the locations and types of rotary actuators must be used simultaneously in the GA framework, eventually increasing the complexity of the crossover operation and the mutation operation. Some evolutionary computational methods have been developed for this kind of mixed-integer problem. For example, Hideki and Heinz developed an evolutionary algorithm without selection and mutation [39]. Samir and David presented a parallel recombinative simulated annealing method by introducing the concept of the cooling schedule and Boltzmann tournament selection [40]. On the other hand, a hybrid GA method combined with the local search and neighborhood search methods has been developed [41]. This research presents a sequential combination of a GA and a sensitivity-based optimization algorithm (SQP, sequential quadratic programming). Fig. 10(b) is a flow chart of the proposed hybrid GA, whereas Fig. 10(a) is a flow chart of the classical GA. Note that after the GA is run for a few generations, LocalSearch_SQP optimizes the joint locations.

In the proposed hybrid GA shown in Fig. 10(b), the maximum iteration of the GA is fixed, so the populations of the GA might be premature (not converged). Nevertheless, a gradient-based optimizer is employed to design the optimal locations of the joints of the best individual offspring (elitism) identified in the GA optimization process (see [42,43] for a description of GA terminology). In other words, the x–y locations of the joints of the selected best individual offspring are optimized in order to minimize the difference norm of the trajectories of Eq. (6). After the gradient-based optimization process, because the locations of the optimized

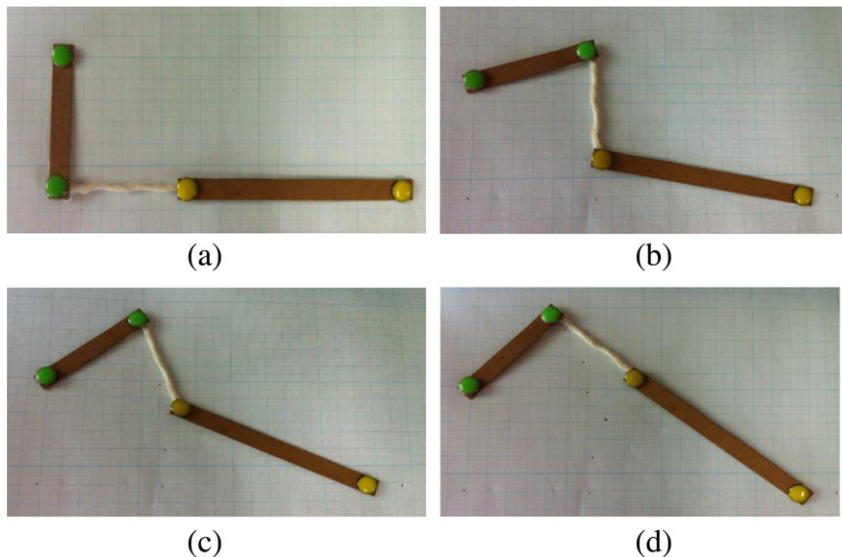


Fig. 12. Implementation of the string-rigid link mechanism. (a), (b), (c) and (d): the string-rigid link mechanisms at the first, eighth, ninth, and final steps, respectively.

joints are real values, they are re-mapped to the closest district points defined by resol before continuing the GA. The best individual optimized by the LocalSearch_SQP algorithm is then reinserted into the GA populations. The above procedures are iterated until the best solutions are obtained or the maximum iteration is reached. The structure of our hybrid GA is shown below.

Algorithm. Hybrid Genetic Algorithm

```

1: Initialize population size, crossover rate, mutation rate
2: Choose design parameters, domain and boundary conditions.
3: Generate Initial Generation
4: for Outer_iteration= 1 to MAX_Outer_Iteration (MAX_Outer_Iteration: Maximum outer iteration)
5:   while not converge do
6:     for pop= 1 to population size
7:       Apply crossover
8:       Apply mutation
9:       Apply elitism (Keep best individual)
10:    end for
11:    gen ← next gen
12:  end while
13: next generation ← LocalSearch_SQP (Best individual)
14: end for

```

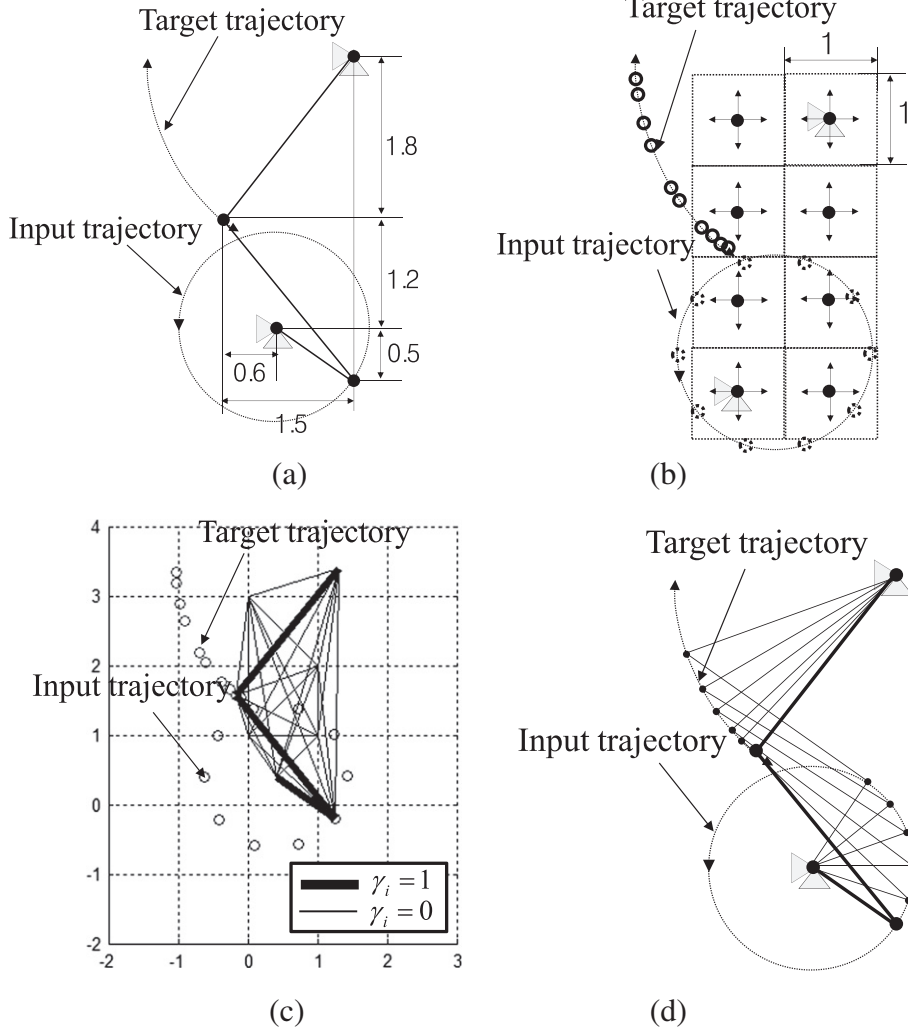


Fig. 13. Synthesis of the first four-bar mechanism: (a) A reference four-bar mechanism, (b) initial distribution of 8 masses, the relative displacement inputs, and the prescribed target displacements of the four-bar mechanism, (c) optimized design, and (d) the actual movements of the design.

3.4. LocalSearch_SQP

In the added LocalSearch_SQP module the x–y positions of joints are optimized using the SQP optimization algorithm by solving the following optimization problem. The sensitivity values (the gradient values) of the position vector of the best individual are calculated by the finite difference method.

$$\text{Min}_x \Phi = \sum_{k=1}^{N_p} \|\mathbf{r}_W^k - \mathbf{r}_{W,\text{Target}}^k\| \quad (11)$$

\mathbf{X} = Position Vector of best individual

After the solution of the above optimization problem is determined, the design variables \mathbf{X} become real variables. Therefore, to insert the optimum design of Eq. (11), the design variables are re-mapped to the closest location defined by resol before continuing the GA.

4. Numerical examples

To validate the usefulness and performance of the developed constraint force design method, several syntheses of two-dimensional rigid-body mechanisms are considered in this section. The first numerical example is devised to illustrate the capability of designing string links as well as rigid links. The second numerical example is devised to show the utility of the proposed method in designing the size and configuration of rigid four-bar mechanisms. The third numerical example showcases the proposed method's ability to determine the best size and configuration of a Peaucellier–Lipkin linkage mechanism. The next numerical example considers the optimal designs of multiple rotary actuators and rigid links. The developed hybrid GA is applied

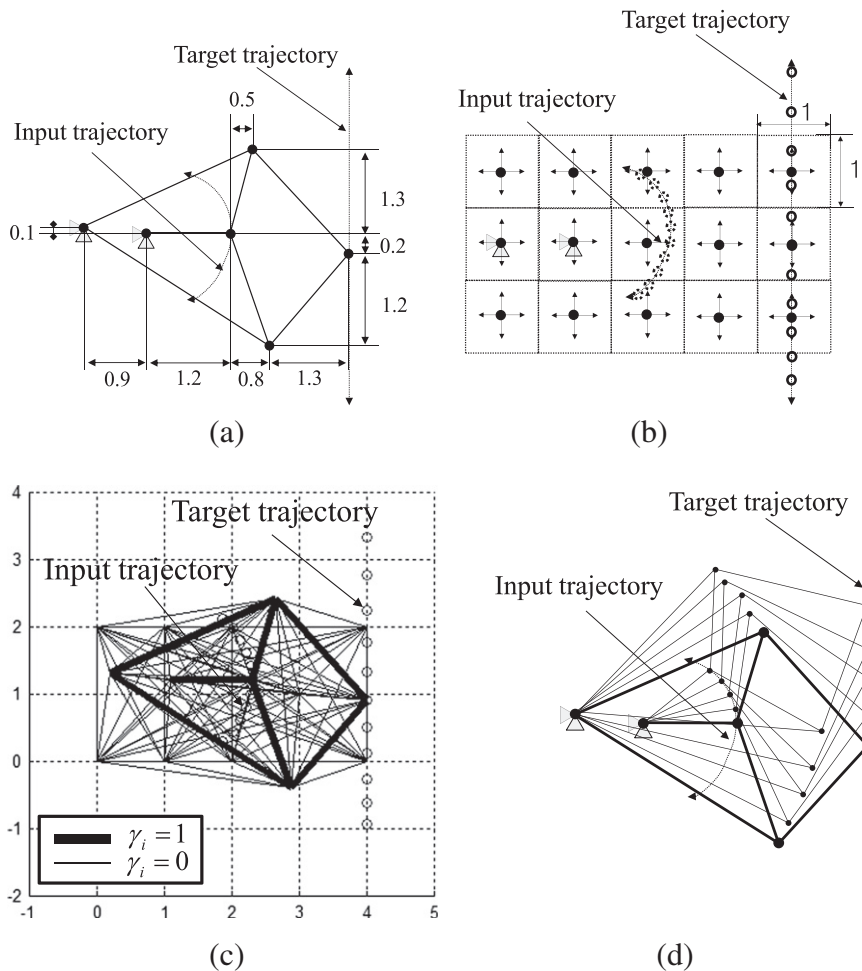


Fig. 14. Synthesis of the Peaucellier–Lipkin mechanism: (a) Reference Peaucellier–Lipkin mechanism, (b) distribution of 15 unit masses, the relative displacement inputs, and the prescribed target displacements of the Peaucellier–Lipkin mechanism, (c) optimized design, and (d) actual movements of the design.

in the last numerical example. To show the efficiency of the hybrid GA, the target and input trajectories of existing mechanisms are considered. The employed detail parameters are shown in Table 1.

Example 1. Two bars connected with a string link

For the first numerical example, the two bars whose ends are connected with a string link in Fig. 11(a) are considered. Until the left link is rotated 90° counterclockwise, the right rigid link does not move. Beyond 90° , the right bar also rotates clockwise because of the connecting string, as shown in Fig. 11(b). To design these rigid bars connected with the string, the target and input trajectories in Fig. 11(b) consisting of ten points are set; the first four positions of the target trajectory of the right bar are identical. To determine this reference mechanism, the eight unit masses are distributed initially, with the total number of design variables in Eq. (9) being 56. The first 28 design variables are for the existence of rigid links, and the remaining 28 are for the determination of string or rigid links. The mechanism in Fig. 11(c) is determined by applying the developed constraint force design method. Double lines represent the string here. As illustrated, the present method can be used to design string links and rigid links simultaneously.

The simple real mechanism of the above design is shown in Fig. 12 to check its performance. As shown, the right rigid bar does not rotate before the string is tightened, as intended.

Example 2. Synthesis of a four-bar link system

For the second numerical example, a simple four-bar link mechanism is considered in Fig. 13. Despite its simple geometry, this mechanism synthesis demonstrates the potential of the developed optimization procedure with integer design variables moving

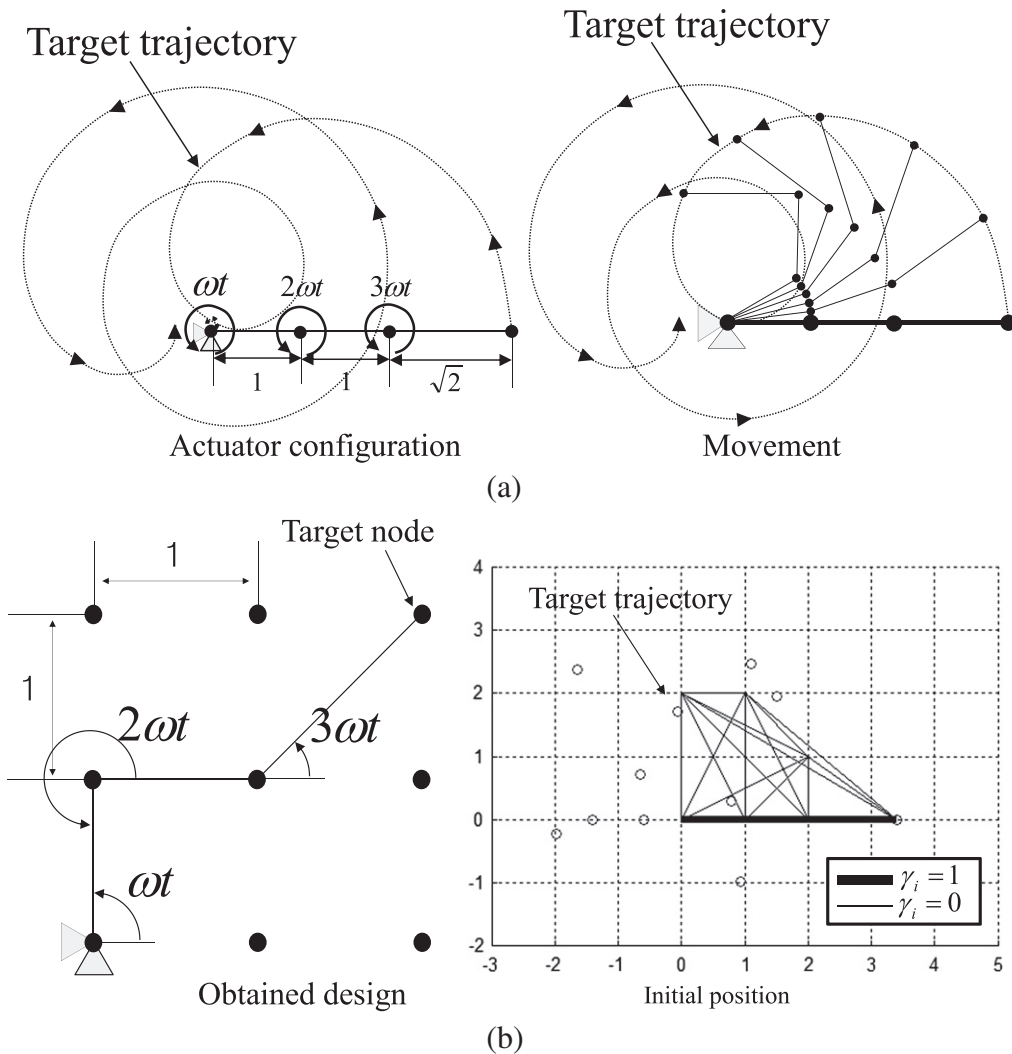


Fig. 15. Synthesis of a link system with rotary actuators: (a) Reference mechanism with rotary actuators and its movement and (b) local optimum and initial position.

the joint locations of Eq. (7). Because the joint locations in Fig. 13(a) are chosen arbitrarily, many initial unit masses must be used in the original constraint force design method with binary design variables. By introducing the integer variables of Eq. (7), however, it is possible to move and optimize the locations of these eight unit masses and use fewer unit masses. The parameter “resol” determines the number of points at which the corresponding joint can move and is set to 10 in this numerical test. The dotted boxes represent the domains where the unit masses can move. As shown in Fig. 13(c) and (d), we can determine the mechanism.

Example 3. Synthesis of the Peaucellier–Lipkin linkage mechanism

For the next numerical example, the synthesis of the popular Peaucellier–Lipkin linkage mechanism transforming rotational motion to perfect straight-line motion is considered in Fig. 14. The parameter resol is set to 10, and the locations of the 15 joints are subject to optimization. There are 135 (105 + 30) design variables; the first 105 design variables correspond to the existence of rigid links, and the next 30 are for the joint locations. As shown, it is possible to design this complex mechanism efficiently.

Example 4. Synthesis of rotary actuators and rigid links

The present constraint force design method can determine the optimal positions of rotary actuators and rigid links simultaneously by introducing the additional integer design variables in Eq. (8). To demonstrate this new feature, the synthesis of the three rotary actuators and the three rigid links with different lengths shown in Fig. 15(a) is considered here. It consists of the three straight rigid links (unit lengths of the first two rigid links and $\sqrt{2}$ for the last link) and the three rotary actuators (with ω , 2ω , and 3ω as their angular velocities). Note that the angles between rigid links with these rotary actuators become ωt , $2\omega t$, and $3\omega t$, respectively, where t is the time. With this configuration, the trajectory of the end point of this mechanism becomes a complex behavior that may be hard for engineers to conceive. To determine a mechanism or mechanisms providing this complex trajectory of the end point, nine unit masses are distributed as in Fig. 15(c). The first 36 ($9 \times 8/2$) binary design variables are set for the existence of the rigid links, and the subsequent 9 integer design variables assigned to each unit mass are used for determining the

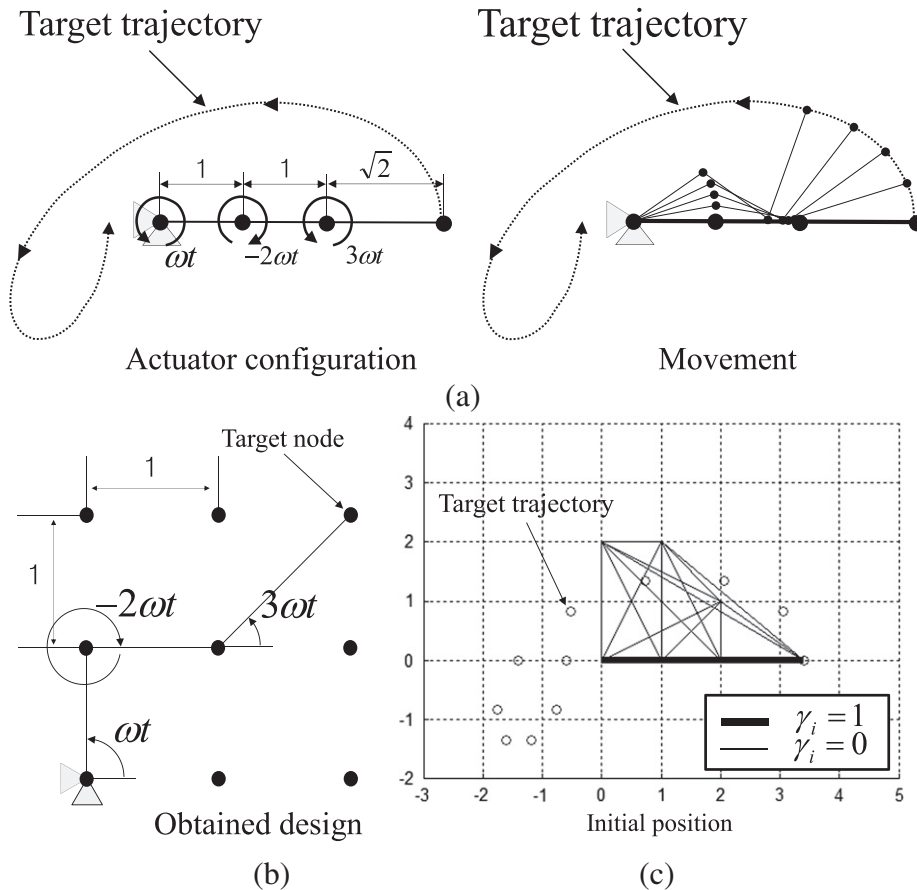


Fig. 16. Synthesis of a link system with different rotary actuators: (a) Reference mechanism with the rotary actuators and its movement, (b) local optimum, and (c) initial position.

velocity of rotary actuators; the parameter na of Eq. (8) is 3. The 37th to 45th design variables, which vary from 0 to 3 for the four angular velocities, i.e., 0 (no-actuator), ω , 2ω , and 3ω , respectively, determine the angular velocities of the rotary actuators assigned to each unit mass. Applying the proposed constraint force design method, the design on the left side of Fig. 15(b) can be obtained. Note that because the three rotary actuators impose their angular velocities and the corresponding angles between the rigid links, the initial shape is a straight line, as shown on the right side of Fig. 15(b).

4.0.1. The synthesis of rotary actuators with local optima issue

For another numerical example, the same rigid links with different rotary actuators (ω , -2ω , and 3ω as their angular velocities, respectively) are used in Fig. 16. With this configuration, the obtained trajectory differs greatly from that of the previous example. Using the same conditions as in the previous example, the best design of Fig. 16 can be obtained.

4.0.2. Extended design variables for rigid link, rotary actuator, and the position of the initial mass

We further expand the design variable spaces in order to optimize the existence of rigid links, the locations of rotary actuators, and the initial positions of unit masses in Fig. 17. The trajectory of the design in the left side of Fig. 17(a) is considered. Note that because the lengths are real variables, the presented optimization framework should determine optimal real values for the lengths of the rigid links, the optimized locations and types of the rotary actuators, and the existence of the rigid links. By applying the developed approach, the design on the left side of Fig. 17(b), whose trajectory is identical to that of the reference design, is obtained. This example shows that the present method is capable of determining optimal rigid links, the locations and types of actuators, and the positions of initial masses.

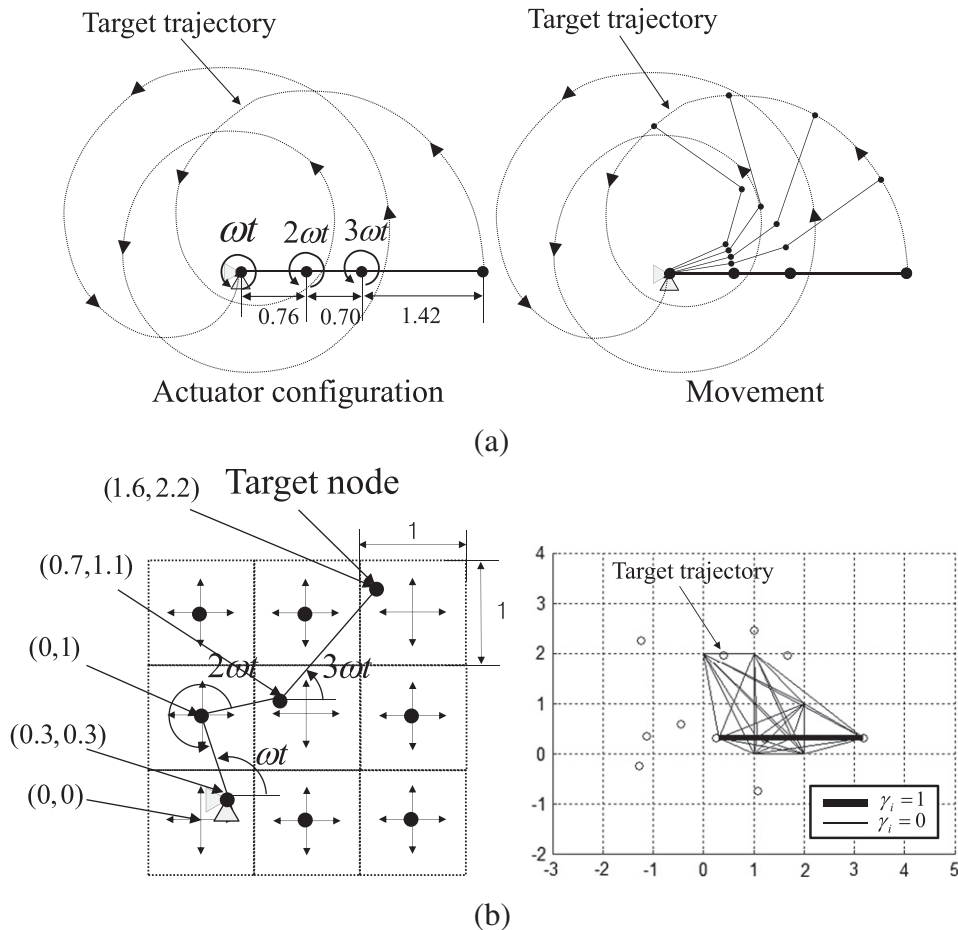


Fig. 17. Synthesis of a link system: (a) Reference mechanism and its movement, and (b) local optimum and the initial position.

Example 5. Synthesis of a rigid link system with the proposed hybrid genetic algorithm

For the last numerical example, the hybrid GA developed in the previous section is applied to the synthesis problem shown in Fig. 18(a). Although the synthesis problem can be solved using the existing GA method, the convergence improvement with the hybrid method is assessed. Fig. 18 shows the optimal layouts determined by the constraint force design method with and without the LocalSearch_SQP. Fig. 19 shows the iteration histories of the problem with and without the LocalSearch_SQP algorithm. GA being premature at the 50th iteration, we stop the optimization process and apply the LocalSearch_SQP in Fig. 19(b). As shown, it is possible to reduce the computation time with the present hybrid GA. With the standard GA, it is possible to obtain a mechanism very similar to the reference design, but not exactly the same. Because the locations of the joints of the best individual are optimized with the gradient-based SQP algorithm, an exact design whose configuration is the same as that of the reference design can be obtained faster. Though not presented, several iterations of the GA and the LocalSearch_SQP are possible.

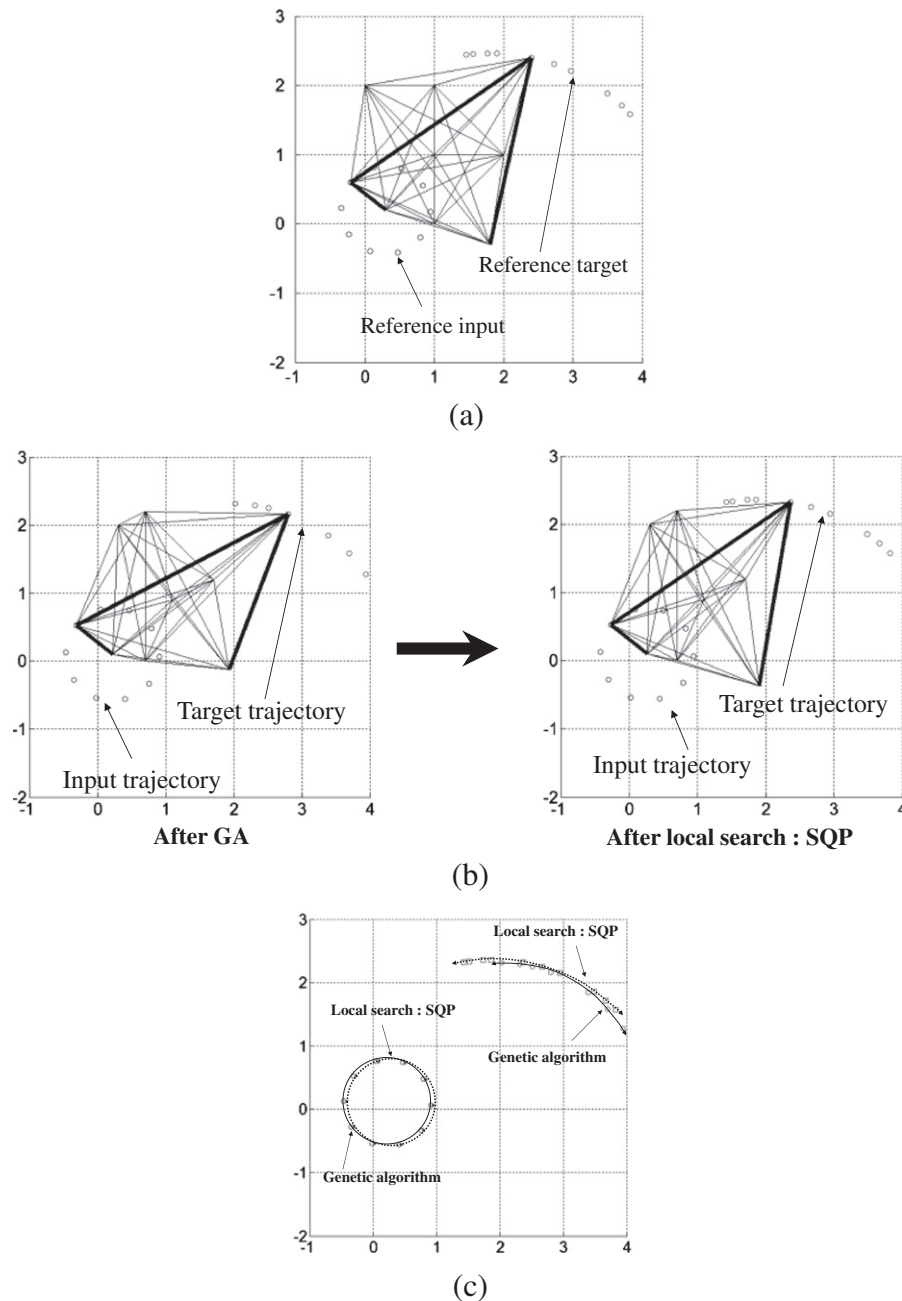


Fig. 18. Synthesis of a four-bar mechanism 1 using the hybrid GA: (a) Reference four-bar mechanism, (b) design obtained using the hybrid GA, and (c) trajectories of the designs.

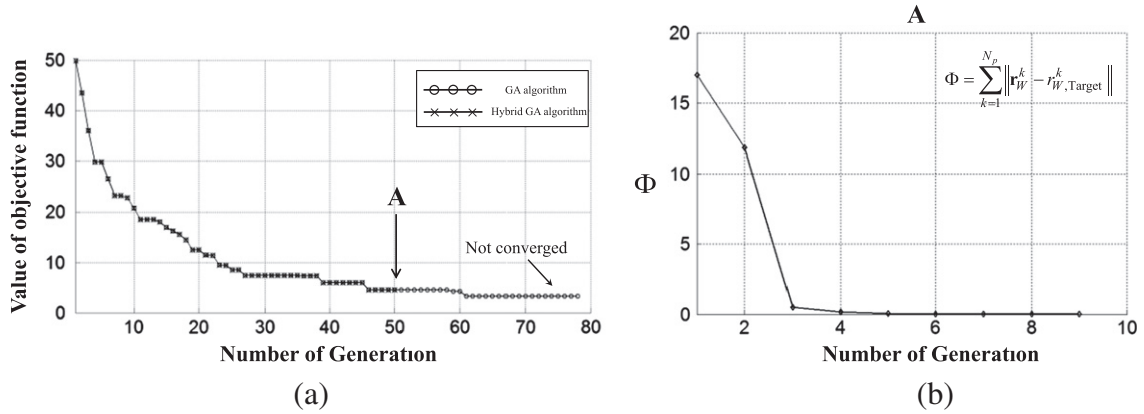


Fig. 19. Comparisons of the generic GA and the present hybrid GA for the four-bar mechanism. (The generic GA: 21,182 s and the present hybrid GA: 16,018 s).

Furthermore, a four-bar mechanism similar to the above example but with an additional triangular structure is considered in Fig. 20. As in the first example, the present hybrid GA can determine the optimal design very efficiently as shown in Fig. 21.

5. Conclusions

An extended constraint force design method for a rigid-body mechanism synthesis problem was designed in this study; compared with our previous contribution [15], the types and locations of actuators, as well as the types of links, are additionally parameterized and optimized. Furthermore, a new hybrid GA algorithm is presented. Topological alternations or configuration changes of rigid-body mechanisms are not allowed in conventional synthesis methods, so it is difficult (sometimes impossible) to automatically determine the optimal numbers and types of links and rotary actuators. Therefore, this paper proposed a novel rigid link mechanism design approach called the extended constraint force design method. In this new synthesis method, in conjunction with the dynamic equation (or Newton's equation) of particles, the particles maintain constant relative distances, which correspond to the lengths of rigid links, by introducing the artificial forces (the Lagrangian forces) calculated by the SHAKE algorithm. Optimization of the existence and types of these artificial forces among unit particles makes it possible to explore the optimal topologies or the optimal configurations for path-generation problems and function-generation problems. Since the initial positions of the unit particles are related to the local optimal configuration issue, this research introduced integer design variables to determine the optimal initial positions of unit masses, in addition to binary design variables for the existence of artificial forces. Additional integer design variables were also introduced to consider the synthesis of rotary actuators. From an evolutionary optimization point of view, a new hybrid GA combining a GA and the SQP optimization algorithm was developed for the sake of computational efficiency. In the proposed hybrid GA framework, the joint positions of the best individual population are optimized by the SQP optimization algorithm with the finite difference method for the sensitivity values of the design variables (the location of joints). Advantages of this hybrid GA include the use of few unit masses for a rigid-body mechanism with spatially complex geometry and the quick convergence of solutions. However, one disadvantage of this hybrid GA is that there is a chance of premature convergence of the GA. Therefore, a compromise between the accuracies and efficiencies of the hybrid GA and the GA should be considered in real applications. In conclusion, this research extends the existing constraint force design method for actuator design and size/shape optimization of joints, and develops a new hybrid GA for mechanism design synthesis. The present research can be extended to design problems for guidance mechanisms or function-generation mechanisms involving velocities and accelerations.

Acknowledgment

This work was supported by the research fund of Hanyang University (HY-2012-G).

Appendix A. Shake Algorithm

To impose the constant length condition among masses, the auxiliary force \mathbf{g}_i must be calculated and imposed upon each mass. The SHAKE algorithm is used to calculate these constraint forces efficiently. To understand the basic concept of this SHAKE method, consider the two masses in Fig. 22.

We assume that the distance between the i -th and j -th masses at time t is constrained to d_{ij} . The uncorrected positions of the i -th and j -th masses are denoted by $\tilde{\mathbf{r}}_i$ and $\tilde{\mathbf{r}}_j$, respectively. Uncorrected positions not satisfying the length constraint must be corrected by considering the length constraint or applying the forces \mathbf{g}_i at time $t+h$. The corrected positions \mathbf{r}_i and \mathbf{r}_j satisfy the length constraints (d_{ij}) as shown in Fig. 22.

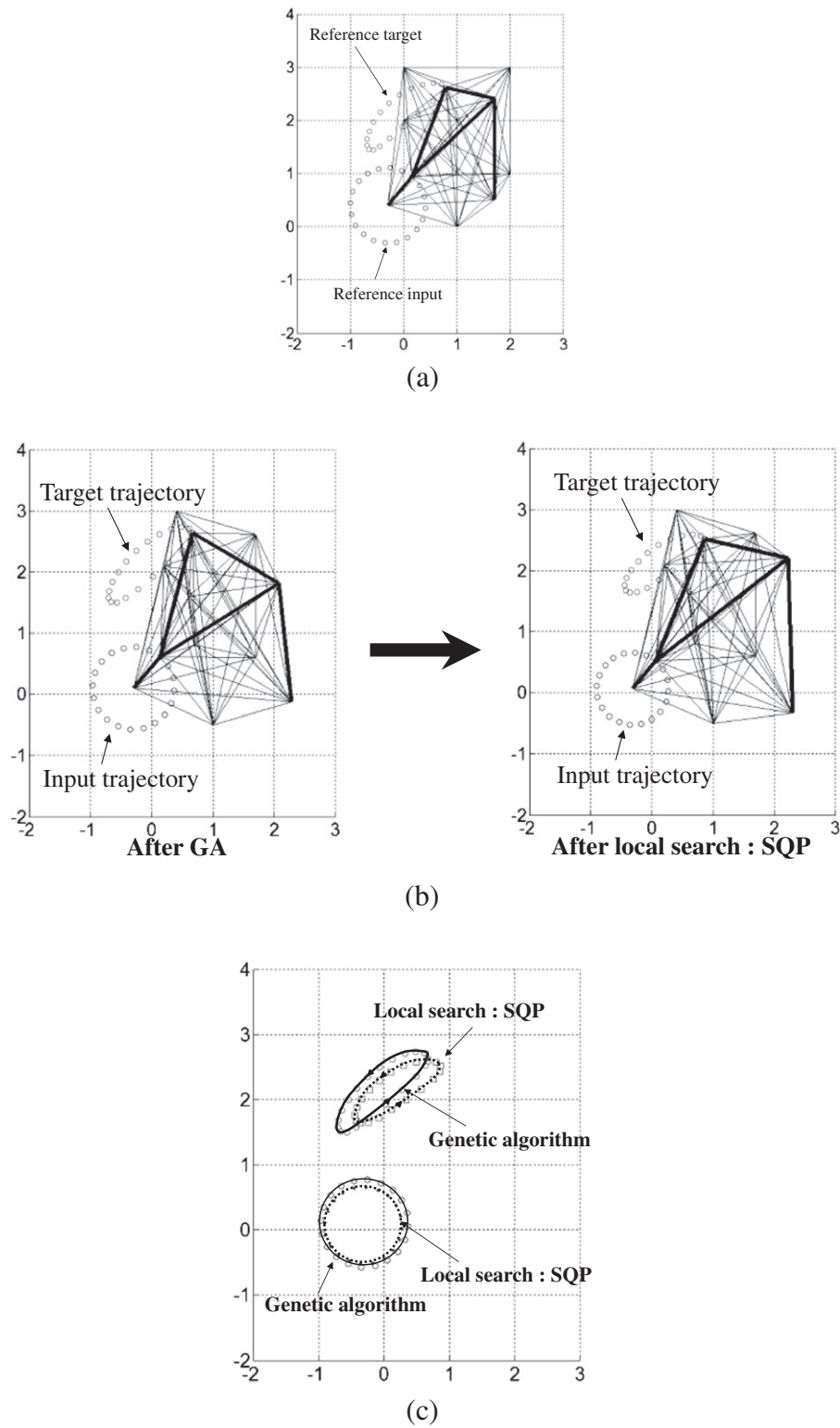


Fig. 20. Synthesis of four-bar mechanism 2 using the hybrid GA: (a) Reference four-bar mechanism, (b) design obtained using the hybrid GA, and (c) trajectories of the designs.

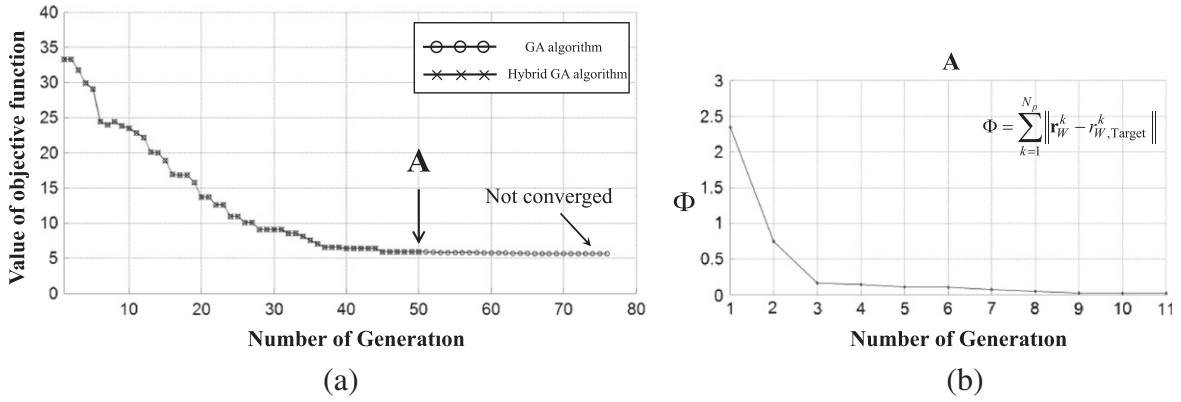


Fig. 21. Comparisons of the generic GA and the present hybrid GA for the four-bar mechanism (the generic GA: 56,008 s and the present hybrid GA: 46,834 s).

The current distance vector between the i -th and j -th masses is set to $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, and the distance for the k -th rigid link is set to d_{ij} as follows:

$$\mathbf{r}_{ij}^2 = |\mathbf{r}_i - \mathbf{r}_j|^2 = d_{ij}^2 \text{ for the } k\text{-th link.} \quad (12)$$

While solving Newton's second law to update the position of each mass, the length constraint described above should be properly imposed using the Lagrangian multiplier in Eq. (3). To impose this length constraint condition, the scalar σ_k is defined as follows:

$$\sigma_k = \mathbf{r}_{i,jk}^2 - d_{i,jk}^2. \quad (13)$$

For the sake of clarity and convenience, the distance vector and the distance for the k -th rigid link are denoted by $\mathbf{r}_{i,jk}$ and $d_{i,jk}$, respectively. The constraint force \mathbf{g}_i in (3) can then be written as follows:

$$\mathbf{g}_i = - \sum_{k=C_i}^{N_{RL}} \lambda_k \nabla_i \sigma_k \text{ and } \nabla_i \sigma_k \equiv 2\mathbf{r}_{ij} \quad (14)$$

where C_i is the set of constraints directly involving \mathbf{r}_i . For this numerical correction step, the Lagrangian multipliers λ_k are unknown and must be determined. Of the many numerical methods used to calculate these Lagrangian multipliers, this research uses the heuristic SHAKE method [16,17].

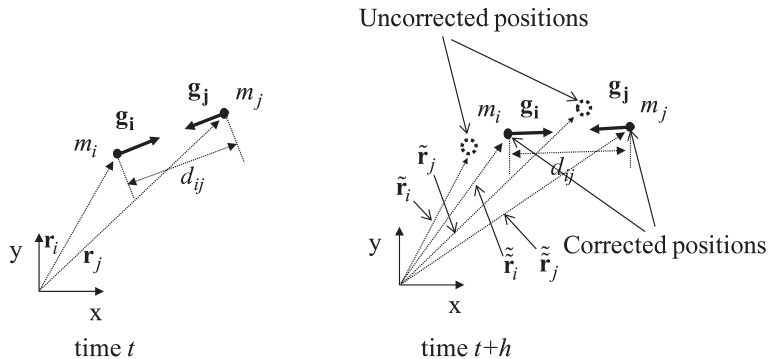


Fig. 22. An illustrative example of two unit masses of the length constraint method. (The index of the rigid link connecting the i -th and j -th masses is k .)

References

- [1] H. Zhou, K.L. Ting, Path generation with singularity avoidance for five-bar slider-crank parallel manipulators, *Mechanism and Machine Theory* 40 (2005) 371–384.
- [2] H. Shimojima, K. Ogawa, A. Fujiwara, O. Sato, Kinematic synthesis of adjustable mechanisms. 1. Path generators, *Bulletin of JSME* (1983) 627–632.
- [3] J.F. McGovern, G.N. Sandor, Kinematic synthesis of adjustable mechanisms. 2. Path generation, *Journal of Engineering for Industry-Transactions of the ASME* 95 (1973) 423–429.
- [4] H. Zhou, K.L. Ting, Adjustable slider-crank linkages for multiple path generation, *Mechanism and Machine Theory* 37 (2002) 499–509.
- [5] A. Saxena, Synthesis of compliant mechanisms for path generation using genetic algorithm, *Journal of Mechanical Design* 127 (2005) 745–752.
- [6] D. Mundo, J.Y. Liu, H.S. Yan, Optimal synthesis of cam-linkage mechanisms for precise path generation, *Journal of Mechanical Design* 128 (2006) 1253–1260.
- [7] Y.P. Singh, D. Kohli, Synthesis of cam-link mechanisms for exact path generation, *Mechanism and Machine Theory* 16 (1981) 447–457.
- [8] J.F. McGovern, G.N. Sandor, Kinematic synthesis of adjustable mechanisms. 1. Function generation, *Journal of Engineering for Industry-Transactions of the ASME* 95 (1973) 417–422.
- [9] H. Shimojima, K. Ogawa, Kinematic synthesis of adjustable mechanisms. 2. Function generators, *Bulletin of JSME* 27 (1984) 1025–1030.
- [10] D.B. Dooner, Function generation utilizing an eight-link mechanism and optimized non-circular gear elements with application to automotive steering, *Proceedings of the Institution of Mechanical Engineers Part C: Journal of Mechanical* 215 (2001) 847–857.
- [11] Z.X. Wang, H.Y. Yu, D.W. Tang, J.S. Li, Study on rigid-body guidance synthesis of planar linkage, *Mechanism and Machine Theory* 37 (2002) 673–684.
- [12] J. Yao, J. Angeles, Computation of all optimum dyads in the approximate synthesis of planar linkages for rigid-body guidance, *Mechanism and Machine Theory* 35 (2000) 1065–1078.
- [13] S.C. Venkataraman, G.L. Kinzel, K.J. Waldron, Optimal synthesis of four-bar linkages for four-position rigid-body guidance with selective tolerance specifications, *22nd Biennial Mechanisms Conference; Scottsdale, AZ; USA; 13–16 Sept, 1992*, pp. 651–659.
- [14] G. Gatti, D. Mundo, Optimal synthesis of six-bar cammed-linkages for exact rigid-body guidance, *Mechanism and Machine Theory* 42 (2007) 1069–1081.
- [15] G.H. Yoon, J.C. Heo, Constraint force design method for topology optimization of planar rigid-body mechanisms, *Computer-Aided Design* 44 (2012) 1277–1296.
- [16] V. Krautler, W.F. Van Gunsteren, P.H. Hunenberger, A fast SHAKE: algorithm to solve distance constraint equations for small molecules in molecular dynamics simulations, *Journal of Computational Chemistry* 22 (2001) 501–508.
- [17] S. Miyamoto, P.A. Kollman, Settle – an analytical version of the shake and rattle algorithm for rigid water models, *Journal of Computational Chemistry* 13 (1992) 952–962.
- [18] W.J. Zhang, A.J.K. Breteler, An approach to mechanism topology identification with consideration of design processes progression, *Proceedings Of The Institution Of Mechanical Engineers Part C—Journal Of Mechanical Engineering Science* 211 (1997) 175–183.
- [19] A. Kawamoto, M.P. Bendsoe, O. Sigmund, Articulated mechanism design with a degree of freedom constraint, *International Journal for Numerical Methods in Engineering* 61 (2004) 1520–1545.
- [20] R.J. Minnaar, D.A. Tortorelli, J.A. Snyman, On nonassembly in the optimal dimensional synthesis of planar mechanisms, *Structural and Multidisciplinary Optimization* 21 (2001) 345–354.
- [21] Y.Y. Kim, G.W. Jang, J.H. Park, J.S. Hyun, S.J. Nam, Automatic synthesis of a planar linkage mechanism with revolute joints by using spring-connected rigid block models, *Journal of Mechanical Design* 129 (2007) 930–940.
- [22] Y.X. Wang, H.S. Yan, Computerized rules-based regeneration method for conceptual design of mechanisms, *Mechanism and Machine Theory* 37 (2002) 833–849.
- [23] J.T. Kimbrell, Graphical synthesis of a 4-bar mechanism, *Mechanism and Machine Theory* 19 (1984) 45–49.
- [24] T.W. Norton, A. Midha, L.L. Howell, Graphical synthesis for limit positions of a 4-bar mechanism using the triangle inequality concept, *Journal of Mechanical Design* 116 (1994) 1132–1140.
- [25] G.N. Sandor, D. Kohli, C.F. Reinholtz, Closed-form analytic synthesis of a five-link spatial motion generator, *Mechanism and Machine Theory* 19 (1984) 97–105.
- [26] N. Diab, A. Smaili, Optimum exact/approximate point synthesis of planar mechanisms, *Mechanism and Machine Theory* 43 (2008) 1610–1624.
- [27] Y. Liu, J. McPhee, Automated kinematic synthesis of planar mechanisms with revolute joints, *Mechanics Based Design of Structures and Machines* 35 (2007) 405–445.
- [28] M.K. Sonpimple, P.M. Bapat, J.P. Modak, S.R. Pimpalpure, A hybrid simulated annealing algorithm for mechanism synthesis with N-accuracy points, *International Journal of Engineering and Technology* 2 (2010) 367–373.
- [29] S. Nishiwaki, M.I. Frecker, S.J. Min, N. Kikuchi, Topology optimization of compliant mechanisms using the homogenization method, *International Journal for Numerical Methods in Engineering* 42 (1998) 535–559.
- [30] O. Sigmund, On the design of compliant mechanisms using topology optimization, *Mechanics of Structures and Machines* 25 (1997) 493–524.
- [31] T.E. Bruns, D.A. Tortorelli, Topology optimization of non-linear elastic structures and compliant mechanisms, *Computer Methods in Applied Mechanics and Engineering* 190 (2001) 3443–3459.
- [32] Z. Luo, L. Chen, J. Yang, Y. Zhang, K. Abdel-Malek, Compliant mechanism design using multi-objective topology optimization scheme of continuum structures, *Structural and Multidisciplinary Optimization* 30 (2005) 142–154.
- [33] J.A. Hetrick, S. Kota, An energy formulation for parametric size and shape optimization of compliant mechanisms, *Journal of Mechanical Design* 121 (1999) 229–234.
- [34] C.C. Lan, Y.J. Cheng, Distributed shape optimization of compliant mechanisms using intrinsic functions, *Journal of Mechanical Design* 130 (2008).
- [35] H. Zhou, K.L. Ting, Shape and size synthesis of compliant mechanisms using wide curve theory, *Journal of Mechanical Design* 128 (2006) 551–558.
- [36] L.F. Shampine, Stability of the leapfrog/midpoint method, *Applied Mathematics and Computation* 208 (2009) 293–298.
- [37] W.Z. Huang, B. Leimkuhler, The adaptive Verlet method, *SIAM Journal on Scientific Computing* 18 (1997) 239–256.
- [38] J.S. Rao, Optimization, *History of Rotating Machinery Dynamics* 20 (2011) 341–351.
- [39] H. Asoh, H. Muhlenbein, On the mean convergence time of evolutionary algorithms without selection and mutation, *Parallel Problem Solving from Nature – PPSN III*, 866, 1994, pp. 88–97.
- [40] S.W. Mahfoud, D.E. Goldberg, Parallel recombinative simulated annealing – a genetic algorithm, *Parallel Computing* 21 (1995) 1–28.
- [41] C.R. Reeves, Genetic algorithms and neighbourhood search, *Evolutionary Computing* 865 (1994) 115–130.
- [42] S.A. Kazarlis, A.G. Bakirtzis, V. Petridis, A genetic algorithm solution to the unit commitment problem, *IEEE Transactions on Power Systems* 11 (1996) 83–90.
- [43] J.H. Holland, Genetic algorithms, *Scientific American* 267 (1992) 66–72.