Globally Optimal Joint Search of Topology and Trajectory for Planar Linkages

Zherong Pan[1], Min Liu[2,4], Xifeng Gao[3], and Dinesh Manocha[4]

[1] Department of Computer Science, University of North Carolina, North Carolina NC 27514, USA,
zherong@cs.unc.edu

[2] School of Computer, National University of Defense Technology, Hunan HN 410073, China,
gfsliumin@gmail.com

[3] Department of Computer Science, Florida State University, Florida FL 32306, USA,
gao@cs.fsu.edu

[4] Department of Computer Science and Electrical & Computer Engineering, University of Maryland at College Park, Maryland MD 20742, USA,
dm@cs.umd.edu

Abstract. We present a new algorithm for computing globally optimal topology and trajectory jointly for 2D planar linkages. Planar linkage structures can generate complex end-effector trajectories using only a single rotational actuator, which is very useful in building low-cost robots. We address the problem of searching for the optimal topology and geometry of these structures and present new optimization methods that consider topology changes that are non-smooth and non-differentiable. We formulate this problem as a mixed-integer convex programming (MICP) problem for which a global optimum can be found using the branch-and-bound (BB) algorithm. As a result, within a finite amount of time, our method can find planar linkage structures with end-effector trajectories that closely match the user-specified target trajectories. We tested our method to search for planar linkages with $5-7$ rigid bodies. Compared with sampling-based methods or simulated annealing, our method improves the quality of the solution by at most $7\times$ and the optimized planar linkage structure has been tested on a 4-legged walking robot.

Keywords: mixed integer, topology, geometry, optimization

1 Introduction

We present a new algorithm for computing 2D planar linkages that trace out a target end-effector curve. A planar linkage is a mechanical structure built with a set of rigid bodies connected by hinge joints. This structure typically has one effective degree-of-freedom actuated by a rotational motor. Since they impose a minimal burden on controller design, these structures are widely used as building blocks for low-cost toys and robots [29,20]. By combining a series of hinge joints, the end-effector of the planar linkage will trace out a complex curve that can fulfill requirements of different types of locomotion, including walking and swimming [11,22].

A challenging but useful problem in mechanics design is to find the linkage structure with an end-effector that will trace out a curve provided by a robot

designer. A a target curve is one of the most intuitive design goals and is easy to specify. This problem is challenging in that it searches over three coupled variables: topology, geometry, and trajectory. The linkage topology determines which rigid bodies are connected and the order of their connections. The topology can be represented by a set of decision variables that are non-smooth and non-differentiable. The linkage geometry determines the shape of each rigid link. Finally, the trajectory determines the pose of the linkage structure at each time instance. The last two variables are smooth and differentiable, but directly optimizing them induces non-convex functions. Previous works [9,29] have proposed various solutions to address problems of this kind. These methods rely on random searches such as A^* [9] and covariance matrix adaptation [29] to explore different topologies. Then, for each topology, they perform non-linear programming (NLP) under the given topology to compute the geometry and trajectory. However, these methods are computationally expensive because a huge number of samples is needed for the random search to converge. Moreover, even after determining the topology, these methods can find only sub-optimal solutions due to the non-convex nature of NLP.

Main Results: Based on the previous works [13,1,22] that represent a large subset of 2D planar linkages, we present a method to search for 2D planar linkages from intuitive user inputs in this subset. Specifically, given the input of a target trajectory, we can efficiently compute a planar linkage structure with globally optimal topology and geometry configurations in the subset and an accurate trajectory reproduction. Technically, we are inspired by recent advances in mixed-integer modeling [24,6,23] and relax this non-convex joint search problem as an MICP problem. The main benefits of our approach include:

- Compared with sampling-based methods [29], we find globally optimal solutions in a finite amount of computational time. This is achieved by approximating the original solution space, i.e. the subset of 2D planar linkages defined by [13,1,22], to make it compatible with MICP.
- Compared with sampling-based methods, we find more accurate solutions in terms of matching user-specified target trajectories.
- By increasing the number of integer decision variables, the approximate solution space can be arbitrarily close to the original solution space, i.e. the subset of 2D planar linkages defined by [13,1,22].

We have evaluated our method in the optimization of complex planar linkage structures with $5-7$ rigid bodies and have evaluated the performance of an optimized structure in a 4-legged robot.

In the rest of the paper, we first review related work in Section 2 and then formulate our joint search problem in Section 3. We present the MICP model and various constraints required for the integrity of the planar linkage in Section 4. We present the results and evaluate our approach on complex benchmarks in Section 5.

2 Related Work

In this section, we review related work in robot design optimization, mixed-integer modeling, and topology optimization.

Robot Design Optimization: Robot design optimization is a superset of conventional topology and truss optimization [16]. The decision variables in topology optimization are only topology or geometry, but the specification of a robot design is given as a movement pattern [10], leading to a joint search in the space-time domain. The joint search problem greatly expands the search space. As a result, many prior methods do not work since they only optimize a subset of decision variables [10,22,1,19,21]. A main contribution of [22,1] is that they use a prior method [13] to represent a subspace of planar linkages; we use a similar method in this work. Recent works [29,9,20] search for all variables simultaneously. However, these methods are based on random search techniques, which usually require a large amount of trial and error to find sub-optimal solutions.

Mixed-Integer Modeling: The main benefit of mixed-integer modeling is the use of the well-studied BB algorithm [14]. BB allows us to find the global optimum of non-convex programming problems while only visiting a small fraction of the search space. Mixed-integer models have been applied to a large variety of problems including inverse kinematics [6], network flows [5], mesh generations [3], and motion planning with collision handling [7]. We adopt a similar technique for collision handling as that in [7], which uses integer decision variables to select a separating direction for two colliding objects. In addition, by applying the big-M method [23], McCormick envelopes, and piecewise approximations [15], general non-convex problems can be easily relaxed as MICP problems. Prior works [12,17] have also formulated topology optimization problems as MICP. However, our work is the first to formulate the planar linkage problem as MICP and we employ MICP to concurrently find the optimal topology, geometry, and trajectory of a planar linkage concurrently.

Topology Optimization: Topology optimization of a continuum is a well-studied problem [16]. An efficient algorithm can smooth the problem and use a gradient-based method to search for locally optimal structures over a search space of millions of dimensions. This technique has been widely used in the design of soft robots [27,26,28]. However, the optimization of articulated robots is more challenging because the optimized structure must satisfy the joint constraints, making the decision variable non-smooth. Existing techniques use mixed-integer [12,17] or random search techniques [29,20] to optimize over these decision variables.

3 Joint Search for Planar Linkages

As illustrated in Figure 1a, we have a set of rod-like rigid bodies connected to each other using hinge joints. The end points of these rigid bodies can take at most N distinct positions, denoted as node set $\mathbf{n}_{1,\cdots,N}$, of which $\mathbf{n}_1$ is the rotational motor and $\mathbf{n}_N$ is the end-effector. Within one limit cycle, $\mathbf{n}_1$ follows

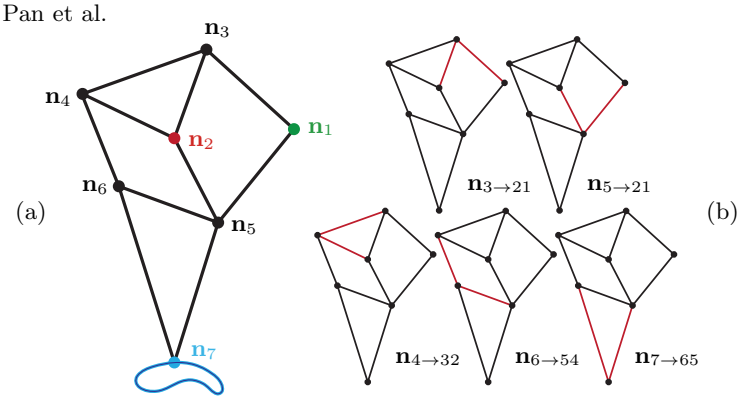

Fig. 1: (a): Jansen's mechanics is a planar linkage involving 7 nodes. The motor node $\mathbf{n}_1$ is green, the fixed node $\mathbf{n}_2$ is red, the movable nodes $\mathbf{n}_{3,4,5,6}$ are black, and the end-effector node $\mathbf{n}_7$ is blue. (b): Our method is based on a prior symbolic representation [13,1,22]. This representation assumes that each node is connected to exactly two other nodes with lower indices: $\mathbf{n}_{3\to21}$, $\mathbf{n}_{5\to21}$, $\mathbf{n}_{4\to32}$, $\mathbf{n}_{6\to54}$, $\mathbf{n}_{7\to65}$.

a circular curve centered at $\left(X_C, Y_C\right)$ with a radius R:

$$\mathbf{n}_1(t) = \left(\mathbf{sin}(t)R + X_C, \mathbf{cos}(t)R + Y_C\right), \tag{1}$$

which induces trajectories of other nodes $\mathbf{n}_i(t)$ via forward kinematics. The other $N-2$ nodes can be one of two kinds: fixed or movable. In addition, a rigid body may exist between each pair of nodes $\mathbf{n}_{i,j}$, in which case $\|\mathbf{n}_i(t) - \mathbf{n}_j(t)\|$ must be a constant for all t.

Given these definitions, the inputs to our problem includes:

- A target end-effector trajectory $\mathbf{n}_N^*(t)$.
- K: The maximal number of nodes in the planar linkage.
- T: The number of samples needed to discretize the end-effector trajectory $\mathbf{n}_N(t)$.
- S: The parameter controlling the accuracy of the MICP formulation. A larger S leads to greater accuracy and higher computational cost.

The output of our method is the following set of variables defining both the topology and geometry of a planar linkage:

- An integer vector of size N (the number of nodes) containing the type of each node: fixed or movable.
- An $N \times N$ symmetric binary matrix $C^{N\times N}$ where $C_{ij} = 1$ means a rigid body connects $\mathbf{n}_{i,j}$.
- The position of $\mathbf{n}_{1,\cdots,N}(t)$ at a certain, arbitrary time instance t.
- X_C, Y_C, R are determined automatically by our MICP formulation.

The goal of our method is to find the globally optimal set of variables that minimizes the cost $\int \|\mathbf{n}_N(t) - \mathbf{n}_N^*(t)\|^2 dt$. For two planar linkage structures, we claim that one is more accurate than the other if its end-effector trajectory incurs a smaller cost.

4 MICP Formulation of Joint Search

In this section, we first introduce our general formulation in Section 4.1, then present the set of topology constraints in Section 4.2, and finally discuss the geometric constraints in Section 4.3.

4.1 The Complete MICP Formulation

We minimize two objective function terms. First, we want the end-effector trajectory to match the target trajectory specified by users. Second, to minimize the manufacturing cost, we want to use as few rigid rods as possible. Putting everything together, we arrive at the following MICP problem:

$$\begin{aligned} \textbf{argmin} \quad & \sum_{d=1}^{T} \|\mathbf{n}_K^d - \mathbf{n}_K^{d*}\|^2 + w \sum_{i=1}^{K} U_i \\ \textbf{s.t.} \quad & \text{Equation 4, 5, 6, 7, 9, 10, 12, 13, 15, 16,} \end{aligned} \tag{2}$$

where $\mathbf{n}_K^{d*}$ are the sampled points on the target trajectory and w is the regularization weight of the cost-efficiency term. The key to Equation 2 is the set of topology constraints introduced in Section 4.2 and the geometric constraints introduced in Section 4.3. Since non-convexity is not accepted by MICP, the solution returned by MICP is only in an approximate solution space. To return a solution with exact constraint satisfaction, we refine the solution by solving an additional NLP locally using the following formulation:

$$\begin{aligned} \textbf{argmin} \quad & \sum_{d=1}^{T} \|\mathbf{n}_K^d - \mathbf{n}_K^{d*}\|^2 + w \sum_{i=1}^{K} U_i \\ \textbf{s.t.} \quad & \text{Equation 4, 5, 6, 7, 9, 10, 8, 14, 16,} \end{aligned} \tag{3}$$

where we fix all the binary variables.

4.2 Topology Constraints

As illustrated in Figure 1b, our method is based on the symbolic representation presented in [13,1,22], which assumes that each movable node is attached to two other nodes. These nodes can be of any type but must have lower node indices. As a result, forward kinematics can be processed sequentially even on linkage structures with closed loops.

Since the number of nodes is unknown, we assume that the maximum number of nodes is $K > N$. We will have all the joints defined, but only N of them should be present, which will be formalized by having the variable U_i. For each node other than the first motor node $\mathbf{n}_1$, $U_i = 1$ indicates that $\mathbf{n}_i$ will be present as a part of the planar linkage structure. In addition, we need another binary variable F_i such that $F_i = 1$ indicates that $\mathbf{n}_i$ is fixed and $F_i = 0$ indicates that $\mathbf{n}_i$ is movable. These two sets of variables are under the constraint that only a

used node can be movable. In addition, we assume that the last node $\mathbf{n}_K$ is the end-effector that must be used. In summary, we introduce the following sets of variables and node-state constraints, denoted as **NodeUsageConstraint**:

$$\begin{aligned} &U_i, F_i \in \{0,1\} \quad \forall i = 1, \cdots, K \\ &1 - F_i \leq U_i \quad U_1 = U_K = 1 \quad F_1 = 0. \end{aligned} \tag{4}$$

Our next set of constraints ensures that each movable node is connected to exactly two other nodes with lower indices. As a result, the movable node and the two other nodes will form a triangle and the position of the movable node can then be determined via the Law of Cosine [10]. We introduce auxiliary variables C^1_{ji} to indicate whether $\mathbf{n}_j$ is the first node to which $\mathbf{n}_i$ is connected. C^2_{ji} indicates whether $\mathbf{n}_j$ is the second node to which $\mathbf{n}_i$ is connected. In addition, we introduce two verbose variables $C^{1,2}_{0i} = 1$ to indicate that $\mathbf{n}_i$ is connected to nothing. The resulting constraint set is:

$$\begin{aligned} &C_{ji}, C^1_{ji}, C^2_{ji} \in [0,1] \quad \forall j, i = 1, \cdots, K \wedge j < i \\ &C_{ji} = C^1_{ji} + C^2_{ji} \quad C^1_{ji} \leq U_j \wedge C^2_{ji} \leq U_j \\ &\sum_{j=1}^{i-1} C_{ji} = 2 - 2F_i \quad \forall i = 2, \cdots, K \\ &C^d_{0i} \in [0,1] \quad \forall d = 1, 2 \\ &\{C^d_{ji} | j = 0, \cdots, i-1\} \in \mathcal{SOS}_1 \quad \sum_{j=0}^{i-1} C^d_{ji} = 1, \end{aligned} \tag{5}$$

which is denoted as **NodeConnectivityConstraint**. When $\mathbf{n}_i$ is fixed in the above formulation, then $F_i = 1$ in Equation 5 and all C_{ji} are zero except for $C^{1,2}_{0i} = 1$ due to the sum-to-one constraints. If $\mathbf{n}_i$ is movable, then $F_i = 0$ and C_{ji} sums to two. As a result, there must be $j_1, j_2 < i$ such that $C^1_{j_1 i} = 1$ and $C^2_{j_2 i} = 1$. Note that j_1 and j_2 must be different because otherwise the constraint that $C_{ji} \in [0,1]$ will be violated. In addition, since the first node $\mathbf{n}_1$ is the motor node, it is excluded from these connectivity constraints. By adopting the idea of a special ordered set of type 1 ($\mathcal{SOS}_1$) [25], we model these constraints using $\mathcal{O}(K\lceil \mathbf{log} K \rceil)$ binary variables. Intuitively, $\mathcal{SOS}_1$ requires that only one variable in a set can take a non-zero value and it can be achieved by using a logarithm number of binary variables.

Finally, we introduce a third set of constraints to ensure global topology correctness. This set of constraints ensures that the linkage structure contains no wasted parts. In other words, each node must have some influence on the trajectory of the end-effector node and the first motor node must be connected to others. We model these constraints using the MICP formulation of network flows [5]. Specifically, each node $\mathbf{n}_i$ will generate an outward flux that equals to U_i, and we assume that there is a flow edge defined between each pair of nodes with capacity Q_{ji}. We require inward-outward flux balance for each node except

the end-effector node:

$$\begin{aligned} & Q_{ji} \in [0, \infty] \quad \forall j, i = 1, \cdots, K \wedge j < i \\ & Q_{ji} \leq C_{ji} K \\ & U_i + \sum_{j=1}^{i-1} Q_{ji} = \sum_{k=i+1}^{K} Q_{ik} \quad \forall i = 1, \cdots, K-1, \end{aligned} \tag{6}$$

which is denoted as **NoWasteConstraint**. Here we adopt the big-M method [23] in the second constraint to ensure that only edges between connected nodes can have a capacity up to K. Big-M is a well-known method in mixed-integer modeling for choosing one element from a discrete set, or for choosing one case from several possible cases. Using a similar idea, we also formulate a constraint that a movable node must be connected to at least one other movable node (otherwise the movable node never moves). We assume that each node $\mathbf{n}_i$ generates a reversed outward flux that equals to $1 - F_i$ and we assume that there is a flow edge defined between each pair of nodes with capacity R_{ji}. We require inward-outward flux balance for each node except for the motor node:

$$\begin{aligned} & R_{ji} \in [0, \infty] \quad \forall j, i = 1, \cdots, K \wedge j < i \\ & R_{ji} \leq C_{ji} K \wedge R_{ji} \leq (1 - F_j) K \\ & \sum_{j=1}^{i-1} R_{ji} = 1 - F_i + \sum_{k=i+1}^{K} R_{ik} \quad \forall i = 2, \cdots, K, \end{aligned} \tag{7}$$

which is denoted as **MovableNodeConstraint**. These four constraints ensure that the planar linkage structure is symbolically correct, independent of the concrete geometric shape.

4.3 Geometric Correctness

The main goal of geometric correctness constraints is to compute the positions $\mathbf{n}_i = \left(x_i, y_i\right)$ of each node in the 2D workspace. These positions are functions of time t and we sample a set of T discrete time instances $t^{1,\cdots,T}$. In this section, we will always use superscripts for timestep indices. For example, at time instance t^d, the position of $\mathbf{n}_i$ is $\mathbf{n}_i^d$. The most important geometric variable is the length of each rigid rod. We define these parameters implicitly using a set of constraints such that, if $\mathbf{n}_i$ and $\mathbf{n}_j$ are connected, then the distance between these two nodes is a constant for all time instances. In other words, we need the following non-convex constraints if $C_{ji} = 1$:

$$\|\mathbf{n}_j^d - \mathbf{n}_i^d\|^2 = \|\mathbf{n}_j^{(d \bmod T)+1} - \mathbf{n}_i^{(d \bmod T)+1}\|^2 \quad \forall 1 \leq d \leq T, \tag{8}$$

after which any distance $\|\mathbf{n}_j^d - \mathbf{n}_i^d\|^2$ can be used as the rigid rod length.

Since any pair of nodes $\mathbf{n}_j$ and $\mathbf{n}_i$ might be connected, a naive formulation will require a number of binary variables proportional to K^2. Instead, we introduce

$$\begin{pmatrix}\alpha\\ \tilde{\alpha}\end{pmatrix} = \sum_{s=1}^{S} \lambda_s \begin{pmatrix}\alpha_s\\ \alpha_s^2\end{pmatrix}$$
$$\{\lambda_{1,\cdots,S}\} \in \mathcal{SOS}_2 \qquad (11)$$
$$\sum_{s=1}^{S} \lambda_s = 1,$$

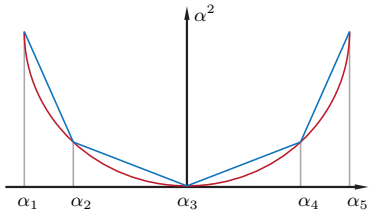

Fig. 2: An illustration of the piecewise linear upper bound (blue) of the quadratic curve α^2 (red) with $S = 5$.

auxiliary term $\mathbf{d}_{1i}^d = \left(dx_{1i}^d, dy_{1i}^d\right)$, which is the relative position between $\mathbf{n}_i$ and the first node connected to it at time instance t^d. Similarly, $\mathbf{d}_{2i}^d = \left(dx_{2i}^d, dy_{2i}^d\right)$ indicates the relative position between $\mathbf{n}_i$ and the second node. These definitions induce the following big-M constraint, which is denoted as **LengthDefinitionConstraint**:

$$\begin{aligned}&\exists\{dx, dy\}_{ki}^d \quad \forall k = 1, 2 \wedge i = 2, \cdots, K \wedge d = 1, \cdots, T\\ &|\{dx, dy\}_{ki}^d - \{x, y\}_j^d + \{x, y\}_i^d| \leq 2B(1 - C_{ji}^k) \quad \forall j = 1, \cdots, i-1,\end{aligned} \qquad (9)$$

where B is the big-M parameter, implying that all the node positions lie in a bounded region $[-B, B]^2$. Note that the first motor node $\mathbf{n}_1$ follows a circular curve (Equation 1), which requires special definitions of $\mathbf{d}_{11}^d, \mathbf{d}_{21}^d$, denoted as **MotorLengthDefinitionConstraint**, as follows:

$$\{dx, dy\}_{11}^d = \{dx, dy\}_{21}^d = \{x_1^d - X_C, y_1^d - Y_C\}, \qquad (10)$$

where the center of rotation $\left(X_C, Y_C\right)$ is used as an additional auxiliary variable. To relax the non-convex constraints, we borrow techniques from [15] and a special ordered set of type 2 ($\mathcal{SOS}_2$) [25]. $\mathcal{SOS}_2$ requires that at most two of the variables in an ordered set with consecutive indices can take non-zero values. To use these formulations, we decompose the range $[-B, B]$ evenly into $S-1$ pieces with S nodes: $\{\alpha_i | -B = \alpha_1 < \alpha_2 < \cdots < \alpha_S = B\}$. As a result, for any $\alpha \in [-B, B]$, a piecewise linear upper bound of α^2 is $\tilde{\alpha}$, which is defined in Equation 11. As illustrated in Figure 2, $\alpha^2 \leq \tilde{\alpha}$ and this upper bound can be arbitrarily tight as $S \to \infty$. This formulation has been used in [6] to discretize the space of unit vectors. In the rest of the paper, we use a tilde to denote such an upper bound. Using these upper bounds, the equidistant constraints can be approximated using the following conic constraints denoted as **SameLengthConstraint**:

$$\begin{aligned}&\forall i = 1, \cdots, K \wedge d = 1, \cdots, T\\ &\|\mathbf{n}_i^d - \mathbf{n}_i^{(d \bmod T)+1}\|^2 \leq (2\sqrt{2}B)^2(1 - F_i)\\ &\forall k = 1, 2 \wedge i = 1, \cdots, K \wedge d = 1, \cdots, T\\ &\|\mathbf{d}_{ki}^{(d \bmod T)+1}\|^2 \leq \tilde{dx}_{ki}^d + \tilde{dy}_{ki}^d + (2\sqrt{2}B)^2 F_i\\ &\|\mathbf{d}_{ki}^d\|^2 \leq \tilde{dx}_{ki}^{(d \bmod T)+1} + \tilde{dy}_{ki}^{(d \bmod T)+1} + (2\sqrt{2}B)^2 F_i,\end{aligned} \qquad (12)$$

where the last term on the right-hand side is the big-M term that excludes fixed nodes and the big-M coefficient $2\sqrt{2}B$ ensures that the constraint will be ignored when it is deactivated by F_i. The idea is to require the length of two vectors to be smaller than the upper bound of one another. Note that Equation 12 converges to Equation 8 as $S \to \infty$. This formulation will introduce $\mathcal{O}(4TK\lceil \mathbf{log} S \rceil)$ binary decision variables. We introduce an additional constraint, denoted as **NonDegenerateConstraint**, to ensure that rigid rods are not degenerate and that they have minimal rod lengths l_{min}:

$$\begin{aligned} &\forall k = 1, 2 \wedge i = 1, \cdots, K \wedge d = 1, \cdots, T \\ &\tilde{dx}^{\,d}_{ki} + \tilde{dy}^{\,d}_{ki} \geq l^2_{min} - ((2\sqrt{2}B)^2 + l^2_{min})F_i. \end{aligned} \tag{13}$$

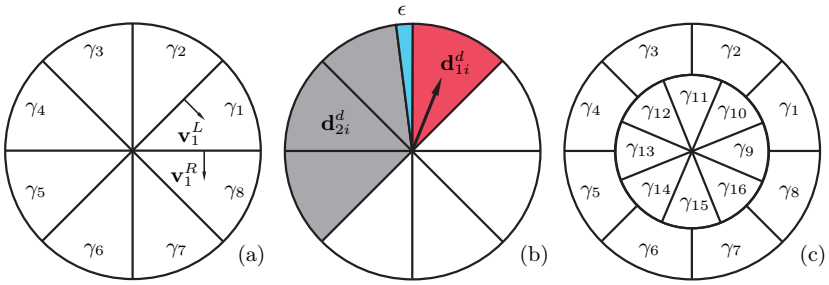

Fig. 3: Linear relaxation of angle constraints. (a): We cut $\mathcal{SO}(2)$ into 8 sectors, each of which is selected by a γ-flag. The sector selected by γ_1 is bounded by its left/right normal vectors $\mathbf{v}_1^L/\mathbf{v}_1^R$. (b): If $\mathbf{d}_{1i}^d$ falls in the red area, then we restrict $\mathbf{d}_{2i}^d$ to the gray area, which is at least ϵ-apart (blue). However, when $\mathbf{d}_{1i}^d$ moves across sector boundaries, the gray area jumps discontinuously. (c): To avoid discontinuous changes for $\mathbf{d}_{2i}^d$ when $\mathbf{d}_{1i}^d$ undergoes continuous changes, we propose to double cover $\mathcal{SO}(2)$.

Unfortunately, degeneracy can still happen if the mechanics system runs into a singular configuration. The most intuitive classification of a singular configuration is the rank-deficiency of the Jacobian matrix [2]. However, a Jacobian matrix is difficult to compute under our representation of rigid rods. Instead, we adopt a heuristic proposed by [22], which avoids singularities by ensuring that, for any movable node $\mathbf{n}_i$, the two vectors $\mathbf{d}_{1i}^d$ and $\mathbf{d}_{2i}^d$ are not colinear. This constraint takes the following bilinear form: $\mathbf{d}_{1i}^d \times \mathbf{d}_{2i}^d \geq \epsilon$, where ϵ is a small constant. Although this constraint is bilinear, we can use McCormick envelopes [15] to relax it. McCormick envelops are techniques to relax non-convex bilinear constraints into piecewise convex constraints. However, a critical limitation of McCormick envelopes is that they are outer-approximations. As a result, the exact linkage structure can still be singular, although its conic relaxation is non-singular. Instead, we propose an inner-approximation for the following constraint:

$$\measuredangle \mathbf{d}_{1i}^d, \mathbf{d}_{2i}^d \geq \epsilon \quad \forall i = 2, \cdots, K \quad d = 1, \cdots, T. \tag{14}$$

Concretely, we cut the space of $\mathcal{SO}(2)$ into S sectors, as illustrated in Figure 3a, so that $\mathbf{d}_{1i}^d$ will only fall into one of the S sectors. If $\mathbf{d}_{1i}^d$ falls in a particular

sector, then we restrict $\mathbf{d}_{2i}^d$ to its left half-space that is at least ϵ-apart, as shown in Figure 3b. If we use an $\mathcal{SOS}_1$ constraint to select the sector in which $\mathbf{d}_{1i}^d$ falls, then only $\mathcal{O}(TK\lceil \mathbf{log}S \rceil)$ binary decision variables are needed. A minor issue with this formulation is that the allowed region of $\mathbf{d}_{2i}^d$ jumps discontinuously as $\mathbf{d}_{1i}^d$ changes continuously. We can fix this problem by double-covering the region of $\mathcal{SO}(2)$ using $2S$ sectors, as shown in Figure 3c.

To formulate these constraints, we assume that each sector of $\mathcal{SO}(2)$ is flagged by a selector variable γ_l, which is bounded by its left/right unit-length plane-normal vectors $\mathbf{v}_l^L/\mathbf{v}_l^R$. Combined with the fact that constraints should only be satisfied for one particular sector and for only movable nodes, we have the following formulation, which is denoted as **NoColinearConstraint**:

$$\begin{aligned}
&\forall i = 2, \cdots, K \quad d = 1, \cdots, T \\
&< \mathbf{v}_l^L, \mathbf{d}_{1i}^d > \geq 2\sqrt{2}B(\gamma_{l,i}^d - 1) - 2\sqrt{2}BF_i \\
&< \mathbf{v}_l^R, \mathbf{d}_{1i}^d > \leq 2\sqrt{2}B(1 - \gamma_{l,i}^d) + 2\sqrt{2}BF_i \\
&< \mathbf{R}(\epsilon)\mathbf{v}_l^L, \mathbf{d}_{2i}^d > \leq 2\sqrt{2}B(1 - \gamma_{l,i}^d) + 2\sqrt{2}BF_i \\
&< \mathbf{R}(\pi)\mathbf{v}_l^R, \mathbf{d}_{2i}^d > \geq 2\sqrt{2}B(\gamma_{l,i}^d - 1) - 2\sqrt{2}BF_i \\
&\{\gamma_{1,1i}^d, \cdots, \gamma_{2S,i}^d\} \in \mathcal{SOS}_1 \quad \sum_{l=1}^{2S} \gamma_{l,i}^d = 1.
\end{aligned} \tag{15}$$

A minor issue with the constraints in Section 4.3 is that the order of trajectory samples is discarded. In practice, we find that better solutions can be found by preserving the order between these samples. This requirement is formulated by making sure that $\mathbf{n}_K^d$ will be visited by the end-effector sequentially when the motor node rotates by 2π either clockwise or counter-clockwise. This requirement is formulated using the following MICP constraints denoted as **OrderConstraint**:

$$\begin{aligned}
&\|\mathbf{R}(\frac{2\pi}{T})\mathbf{d}_{11}^d - \mathbf{d}_{11}^{d+1}\|^2 \leq (2\sqrt{2}B)^2 D \quad \forall d = 1, \cdots, T-1 \\
&\|\mathbf{R}(-\frac{2\pi}{T})\mathbf{d}_{11}^d - \mathbf{d}_{11}^{d+1}\|^2 \leq (2\sqrt{2}B)^2 (1 - D),
\end{aligned} \tag{16}$$

where D is a binary variable to choose in which direction the motor rotates.

5 Results and Evaluations

We have implemented our method using Gurobi [8] as our MICP solver for Equation 2 and Knitro [4] as our NLP solver for Equation 3. All the experiments are performed on a cluster with 24 CPU cores per process (2.5GHz E5-2680 CPU). Compared with prior work [22], the main benefit of our formulation is that we can search for planar linkage structures from a target trajectory of the end-effector, which requires a minimal amount of effort from users. In Figure 4, we show a list of different target trajectories and the optimized planar linkage structures.

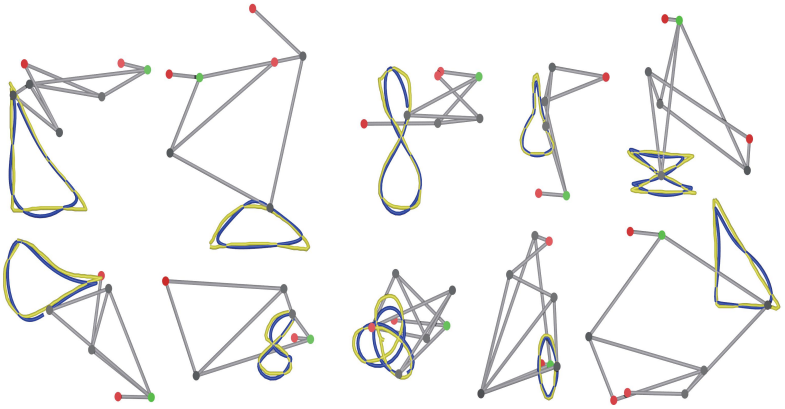

Fig. 4: We show 10 different optimized planar linkage structures with the end-effector trajectory in blue and the user-specified target trajectory in yellow. Our method finds globally optimal solutions for both node positions (green, red, black) and connectivity (gray rod). The end-effector trajectory closely matches the target trajectory. For all these examples, we choose $K = 5 \sim 7$, $S = 9$, and $T = 10 \sim 20$.

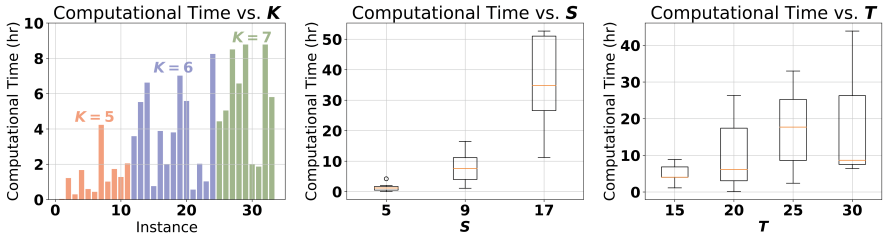

Fig. 5: We plot the average computational time for solving MICP in 35 example problems using different parameters in Figure 4. The computational time for solving MICP grows exponentially with K, $\mathbf{log}S$, and T. The computational time of our method is on the level of hours, which is costlier than the previous method [1] that locally optimizes geometric parameters.

5.1 Computational Cost of our Method

The performance and accuracy of our algorithm heavily depend on the three parameters: the max number of rigid rods K, the number of pieces for approximation S, and the number of samples on the target trajectory T. Since the cost of solving MICP grows exponentially with the number of binary decision variables, which is proportional to K, $\mathbf{log}S$, and T, our method cannot scale to large problems, as illustrated in Figure 5. In practice, we find that, given a maximal computation time of $10 - 15$ hours, we can compute globally optimal solutions for most benchmarks with $K \leq 7$, $S \leq 9$, and $T \leq 20$. This is enough if we design robots part-by-part, as is the case for Theo Jansen's strandbeest [18] illustrated in Figure 9. For other benchmarks, the computational time is longer than 15 hours, but a feasible solution has been found, although it is sub-optimal. In Figure 6, we plot the average convergence history of a typical optimization. Since

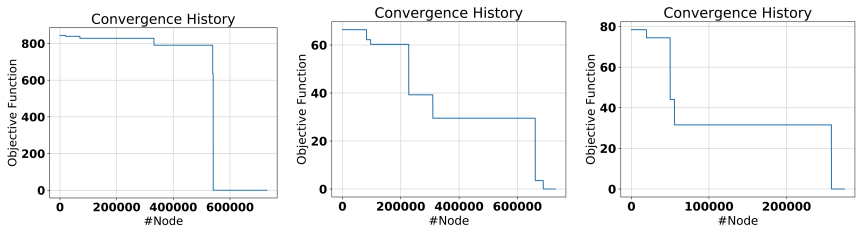

Fig. 6: We plot the convergence history curve for 3 typical optimizations by showing the objective function values plotted against the number of nodes explored in the BB search tree. The BB algorithm spends most of its time exploring infeasible nodes and the first identified feasible solution is usually very close to the optimal solution, so the optimizer will return the globally optimal solution after refining the solution for around 10 times.

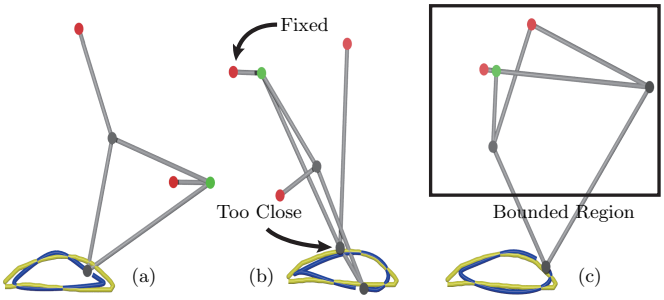

Fig. 7: We illustrate how additional user constraints can be considered. (a): Results with no constraints. The optimizer is guided by the regularization term to use as few nodes as possible. (b): User can fix the center of rotation and the optimizer finds a more complex structure with 6 nodes. However, one of the nodes appears to be too close to the target trajectory. (c): User can add a bounded region and create a constraint that any nodes (other than the end-effector node) should be inside the bounded region.

we express all the topology and geometric requirements as hard mixed-integer constraints, feasible solutions are quite rare in the search space and the optimizer takes most of the computational time pruning infeasible solutions. Once the first feasible solution is found, it is usually very close to the optimal solution and the optimizer refines it for less than 10 times to reach the optimal solution.

5.2 Additional User-Provided Constraints

Usually, the design of a planar linkage structure is not only subject to a target end-effector trajectory, but also to various other user constraints. For example, the user might require certain nodes to be fixed, which can be easily achieved using our MICP formulation. The user may also reserve certain parts of the robot for some functional units that cannot be occupied by the planar linkages. This type of constraint can be expressed as collision avoidance between a planar

linkage structure and a specified convex region, which can be formulated as MICP constraints using a prior method [7]. In Figure 7, we show results taking these constraints into consideration.

5.3 Comparison with Sampling-Based Methods

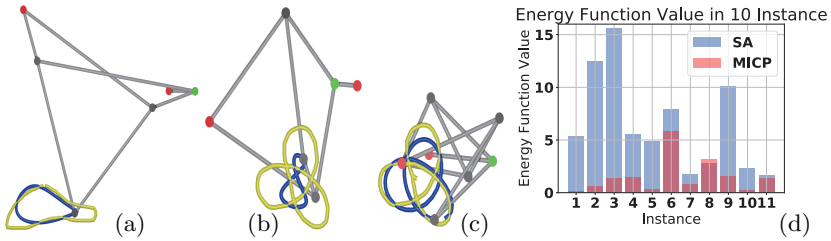

Fig. 8: We compare our method with simulated annealing (SA). SA can find adequate solutions for simple target curves (a). For more complex curve shapes, SA fails (b) while MIP succeeds (c). We also plot the objective function values returned by SA and MICP in 10 computational examples in (d), where MICP outperforms SA in 9 instances. The largest performance improvement is 7× (computational instance #3).

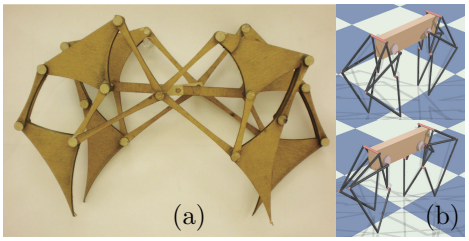

Fig. 9: (a): An example of Jansen's strandbeest. This robot can be manufactured by putting together 4 planar linkages to the left and right of the robot's body; see [18] for more details. (b): Two frames of a 4-legged robot walking on the ground using our optimized planar linkages as legs.

We have also compared our method with conventional global search algorithms such as simulated annealing (SA). We implemented an algorithm similar to the one proposed in [29]. In this algorithm, we randomly generate 1000000 samples by random moves and accept these samples according to the simulated annealing rule. Each random move can be one of three kinds: geometric change, node addition, and node removal. In geometric change, the length of a rigid rod is randomly perturbed. In node addition, a new node is added and the lengths of the new rigid rods are randomly picked. In node removal, the end-effector node is removed and the last movable node is used as the new end-effector node. We enhance the standard SA algorithm by making sure that each random move is valid. In other words, we introduce an inner loop and repeatedly try random moves until the modified planar linkage structure satisfies all the topological constraints and has no singular configurations. As illustrated in Figure 8a, the SA algorithm can find satisfactory results for simple target curves, but SA usually fails for more complex curve shapes (Figure 8bc). In Figure 8d, we also show the objective function values after convergence. The solution of MICP is almost always better than the solution of SA. If we measure the improvement in

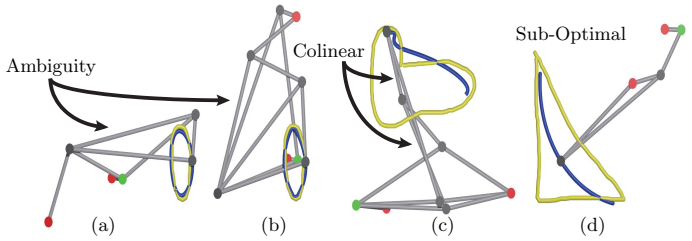

Fig. 10: Failure cases and issues with our formulation. (a,b): MICP only returns the single global optimum. However, similar target trajectories can lead to two different linkage structures. (c): We only satisfy geometric constraints approximately, so the linkage structure might not satisfy these constraints exactly. In this example, we have two rigid rods being colinear. (d): Usually, the early feasible solutions found by MICP are of low quality and we have to wait for the MICP to find the global optimum.

accuracy by the objective function value return by SA divided by the objective function value return by our method, then our method is up to 7× more accurate than SA (computational instance #3). However, SA outperforms MICP in one example, which is probably due to the inexact constraint satisfaction of MICP.

6 Conclusion & Limitations

We present a globally optimal formulation to search jointly for both topology and geometry of a planar linkage structure. Our formulation relaxes the problem into an MICP, for which optimal solutions can be found efficiently using BB algorithms. Our results show that our formulation can search for complex structures from trivial and intuitive user inputs, i.e. target end-effector trajectories. Additionally, various design constraints can be easily incorporated. For moderately complex structures, the optimal solution using these formulations can be found in hours.

As a major limitation, the solve time increases quickly with the number of possible rigid bodies in the planar structure (K) and the number of samples on the target trajectory (T) because the number of decision variables depends on a multiplication of these two parameters. A related issue is that, by using an approximate solution space, MICP only satisfies the geometric constraints approximately. As illustrated in Figure 10c, a predicted target trajectory with approximate constraint satisfaction can be different from a predicted target trajectory with exact constraint satisfaction after solving Equation 3. To reduce the approximation error, we have to increase the approximation granularity by using a larger S, which in turn increases the number of binary decision variables. Finally, note that our formulation does not generate all possible planar linkage structures but only those that can be represented by [13,1,22]. Allowing more general planar linkages is also possible under the MICP formulation by using a new formulation of topology constraints.

6.1 Future Work

Our future research will focus on a balance between global optimality and formulation efficiency. Such a balance could possibly be achieved by using MINLP formulations. In addition, we observe that different planar linkages, as shown in Figure 10ab, can generate very similar target trajectories. This indicates that there exist many local optima with objective functions close to the global optimum. However, a BB algorithm will only return the single global optimum. In addition, we found that we need to wait until the BB algorithm finds its global optimum; the intermediary solutions might not be usable, as illustrated in Figure 10d. A potential future direction is to use algorithms such as Bayesian optimization, which can explore multiple local optima and return many solutions for users to make a choice.

Acknowledgement

This research is supported in part by ARO grant W911NF-18-1-0313, and Intel.

References

1. Bächer, M., Coros, S., Thomaszewski, B.: Linkedit: interactive linkage editing using symbolic kinematics. ACM Transactions on Graphics (TOG) 34(4), 99 (2015)
2. Bohigas, O., Manubens, M., Ros, L.: Singularities of non-redundant manipulators: A short account and a method for their computation in the planar case. Mechanism and Machine Theory 68, 1 – 17 (2013)
3. Bommes, D., Zimmer, H., Kobbelt, L.: Mixed-integer quadrangulation. ACM Transactions On Graphics (TOG) 28(3), 77 (2009)
4. Byrd, R.H., Nocedal, J., Waltz, R.A.: Knitro: An integrated package for nonlinear optimization. In: Large-scale nonlinear optimization, pp. 35–59. Springer (2006)
5. Conforti, M., Di Summa, M., Eisenbrand, F., Wolsey, L.A.: Network formulations of mixed-integer programs. Mathematics of Operations Research 34(1), 194–209 (2009)
6. Dai, H., Izatt, G., Tedrake, R.: Global inverse kinematics via mixed-integer convex optimization. In: International Symposium on Robotics Research, Puerto Varas, Chile. pp. 1–16 (2017)
7. Ding, H., Reißig, G., Groß, D., Stursberg, O.: Mixed-integer programming for optimal path planning of robotic manipulators. In: 2011 IEEE International Conference on Automation Science and Engineering. pp. 133–138. IEEE (2011)
8. Gurobi Optimization, L.: Gurobi optimizer reference manual (2018)
9. Ha, S., Coros, S., Alspach, A., Bern, J.M., Kim, J., Yamane, K.: Computational design of robotic devices from high-level motion specifications. IEEE Transactions on Robotics (99), 1–12 (2018)
10. Ha, S., Coros, S., Alspach, A., Kim, J., Yamane, K.: Joint optimization of robot design and motion parameters using the implicit function theorem. In: Robotics: Science and Systems (2017)
11. Hernández, A., Gómez, C., Crespo, J., Barber, R.: A home made robotic platform based on theo jansen mechanism for teaching robotics. In: INTED2016 Proceedings. pp. 6689–6698. 10th International Technology, Education and Development Conference, IATED (7-9 March, 2016 2016)

12. Kanno, Y.: Topology optimization of tensegrity structures under compliance constraint: a mixed integer linear programming approach. Optimization and Engineering 14(1), 61–96 (2013)
13. Kecskemethy, A., Krupp, T., Hiller, M.: Symbolic processing of multiloop mechanism dynamics using closed-form kinematics solutions. Multibody System Dynamics 1(1), 23–45 (1997)
14. Lawler, E.L., Wood, D.E.: Branch-and-bound methods: A survey. Operations research 14(4), 699–719 (1966)
15. Liberti, L.: Reformulation and convex relaxation techniques for global optimization. Ph.D. thesis, Springer (2004)
16. Liu, J., Ma, Y.: A survey of manufacturing oriented topology optimization methods. Advances in Engineering Software 100, 161 – 175 (2016)
17. Lobato, E., Echavarren, F., Rouco, L., Navarrete, M., Casanova, R., Lopez, G.: A mixed-integer lp based network topology optimization algorithm for overload alleviation. In: 2003 IEEE Bologna Power Tech Conference Proceedings,. vol. 2, pp. 5–pp. IEEE (2003)
18. Nansai, S., Elara, M.R., Iwase, M.: Dynamic analysis and modeling of jansen mechanism. Procedia Engineering 64, 1562–1571 (2013)
19. Saar, K.A., Giardina, F., Iida, F.: Model-free design optimization of a hopping robot and its comparison with a human designer. IEEE Robotics and Automation Letters 3(2), 1245–1251 (2018)
20. Song, P., Wang, X., Tang, X., Fu, C.W., Xu, H., Liu, L., Mitra, N.J.: Computational design of wind-up toys. ACM Transactions on Graphics (TOG) 36(6), 238 (2017)
21. Spielberg, A., Araki, B., Sung, C., Tedrake, R., Rus, D.: Functional co-optimization of articulated robots. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). pp. 5035–5042. IEEE (2017)
22. Thomaszewski, B., Coros, S., Gauge, D., Megaro, V., Grinspun, E., Gross, M.: Computational design of linkage-based characters. ACM Trans. Graph. 33(4), 64:1–64:9 (Jul 2014)
23. Trespalacios, F., Grossmann, I.E.: Improved big-m reformulation for generalized disjunctive programs. Computers & Chemical Engineering 76, 98–103 (2015)
24. Vielma, J.P.: Mixed integer linear programming formulation techniques. Siam Review 57(1), 3–57 (2015)
25. Vielma, J.P., Nemhauser, G.L.: Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. Mathematical Programming 128(1-2), 49–72 (2011)
26. Zhang, H., Kumar, A.S., Fuh, J.Y.H., Wang, M.Y.: Design and development of a topology-optimized three-dimensional printed soft gripper. Soft robotics 5(5), 650–661 (2018)
27. Zhang, H., Wang, M.Y., Chen, F., Wang, Y., Kumar, A.S., Fuh, J.Y.: Design and development of a soft gripper with topology optimization. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 6239–6244. IEEE (2017)
28. Zhu, B., Skouras, M., Chen, D., Matusik, W.: Two-scale topology optimization with microstructures. ACM Transactions on Graphics (TOG) 36(5), 164 (2017)
29. Zhu, L., Xu, W., Snyder, J., Liu, Y., Wang, G., Guo, B.: Motion-guided mechanical toy modeling. ACM Trans. Graph. 31(6), 127:1–127:10 (Nov 2012)