Introducing a New NP-Hard Problem Regarding an Open Chain Linkage Using a New Greedy Method

Ali NOUROLLAH and Nooshin BEHZADPOUR

Faculty of Computer Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran

Abstract

Linkages exhibit numerous applications especially in modelling robot arms. Until now, NP-Complete and PSPACE-Hard problems have been introduced within the linkage context. The main subject of this paper is to introduce a new NP-Hard problem regarding movement of open chain linkages. The objective of this problem is to minimize the moving components of the linkage and their related movement impact, in a way that the end effector is ultimately placed at the target point. For this purpose, first, the problem is formalized and its NP-Hard condition is proved using reduction of sum of subset problem. A greedy algorithm with the time complexity of $O(n^2)$ and space complexity of $O(n)$ is proposed for solving the problem, and computation results from implementing the algorithm are compared with the optimized results. This comparison demonstrates the efficiency and capability of the proposed algorithm.

Keywords: Open Chain Linkage, NP-Hard Problems, Reconfiguration Problem, Reachability Problem, Greedy Method.

1. Introduction

Every linkage is generally equivalent to a connected graph, and an open chain linkage is a linkage that is a graph corresponding to the simple path. The edges of the graph represent the rigid bodies known as links, and the vertices show the joints of the arm, which are used to connect the links to each other. The joints are rotational. The links are zero-depth, so in practice, the links are able to afford a suitable level of abstraction for modelling the robot arm [1].

Reachability is amongst the problems related to linkage geometry. The decision prescription for this problem is whether a specific point on a linkage (normally the end point) can reach a point in the Cartesian space where the linkage resides or not [2]. The region where the end effector can stand within is called Reachability Space and the manner of defining this region is known as the Reachability Problem. The open chain reachability space consists of the region between two concentric circles with the center j_1 and an external radius of $r_o = \sum_{i=1}^{n-1} l_{i,i+1}$, whereas if the length of the largest link is shown as l_m, the internal radius will be $r_i = 2l_m - r_o$. Moreover, if $r_i \leq 0$, the reachability space is a circle with the radius of r_o [2] (Fig. 1).

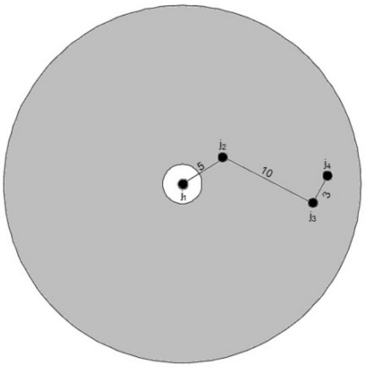

Fig. 1: Reachability region with $\mathbf{r_i} = 2$ and $\mathbf{r_o} = 18$, gray region is reachability space of the arm**.**

Reconfiguration Problem is another important problem related to linkages in computational geometry. In this problem, in which the positive solution of the reachability problem for the given input becomes significant, the linkage movement path must be explicitly specified and reconfiguration of the robot arm be obtained [1]. Considering the various solutions available for solving the open chain reconfiguration problem, goal setting in these types of problems may encompass minimum energy consumption for movement of the robot arm [3], dynamic limitations and constraints such as considering speed and acceleration of joints [4], non-intersection of links [5], movement of the arm within minimum time [6], considering the environmental geometry of the linkage [7], and similar cases.

Computation complexities predominantly stand in NP-Complete and PSPACE-Hard problem classes [7- 9]. The problem of folding an open chain in a way that the goal is to minimize the length of the chain in 1-Dimensional space, area or circumference surrounding 2-Dimensional space, and convex hull within 3-Dimensional space [9], is an NP-Complete problem, which proof of this subject is presented in [10] using reduction to set partitioning problem. The complexity of the reachability problem in an obstructed space and by authorizing intersection of chain links whilst moving is NP-Hard. The same problem for an open chain linkage within a non-obstructed space with intersected links is PSPACE-Hard [7].

A set of components required for building a mechanism includes rigid bodies accompanied by rotary joints. The goal is obtaining a component layout where connection limitations are not violated. Proving the NP-Hard nature of this problem is offered in [11] via the help of reduction of subset sum. This problem by adding the next parameter is NP-Hard, which in [11], it is proven using reduction to 3SAT problem.

Applying approximate and near optimal algorithms for solving NP-Hard problems is of great significance [12]. In [13], linkage conformation and reconfiguration problem of an algorithm is proposed with a time complexity of $O(n^2)$ and space complexity of $O(n)$ with least cost for transforming the open chain linkage of the current configuration to its optimal configuration within 1-Dimensional space. In [14], the approximate linear time algorithm for solving the linkage folding problem is presented. In [15], an approximate algorithm with time complexity of $O(nlogn)$ and space complexity of $O(n)$ is presented for folding the snake linkage which ensures that the length of the chain is no more than two times the length of the largest link.

Obtaining moving linkage joint, which their movement is mandatory for placement of the end effector at the target point, and that the movement imposes minimum effect on other components of the chain is a minimum and NP-Hard optimization problem. In this paper, while proving this issue, a greedy approach to solving the problem is proposed and the results of this algorithm with optimal results is compared. The paper is organized as follows. In section 2, the formal definition of the problem is presented. In section 3, proof of the NP-Hard nature of the problem is provided. Designed algorithms for solving the problem will be discussed in section 4. In section 5, experimental results are presented, and the conclusion is expressed in section 6.

2. Linkage Movement Minimization Problem

In this section, after providing a formal definition for linkages, the linkage movement minimization problem will be proceeding.

An open chain in form of $\langle l_{1,2}, l_{2,3}, \ldots, l_{n-1,n} \rangle$ is a sequence of $n-1$ interconnected links where $l_{i,i+1}$ $(1 \leq i < n)$ shows the length of the edges of the graph and is a positive fixed real number. The arm joints are the vertices of the graph. $l_{i,i+1}$ $(1 \leq i < n)$ is located between the two $j_i, j_{i+1} (1 \leq i \leq n)$ joints. The j_1 is connected to the ground, and j_n is a free joint. Joint j_n is the end effector of the robot arm. Every $j_i (1 \leq i \leq n)$ possesses a three-parameter kinematic variable. $Point_i (1 \leq i \leq n)$ is the Cartesian coordinate of joint $j_i (1 \leq i \leq n)$ in 2_dimensional space, and θ_i $(1 \leq i \leq n)$ is the relative angle between the two links $l_{i,i+1}$ and $l_{i+1,i+2}$ (Counter Clock Wise). θ_1 is the angle of $l_{1,2}$ with the positive direction of the $x-axis$, and θ_n equals zero and is the fixed parameter of the last joint. It is obvious that the Cartesian coordinates of the first joint is the fixed kinematic of the joint.

Rotation of a joint affects its subsequent joints. Is the subset of joint$j_i (1 \leq i \leq n)$ is defined as J_{j_i} in a way that $\mathrm{J}_{j_i} = \{j_{i+1}, \ldots, j_n\}$, it may be said that each joint is under effect of the angular movement of its superset joints, and the propagation direction of each joint is from the superset joint toward subset joints. The relative angular movement of the$i$th joint results in the Cartesian parameter change of J_{j_i}. Hence, the movement analysis of the Cartesian coordinate of a joint is the function of the angular movement of one or more joints of its superset.

In the modelling conducted in this study, the first joint of the arm where its relative angle begins to change is called the «actuator» joint, and preceding joints (if present) are called «inert» joints. If a joint between a moving joint and the trailing joint of the arm does not exhibit relative angular movement, it is called«passive»; otherwise it is called«active». The trailing joint of the arm is always«passive». If the ith joint while moving is «passive», only its $Point_i$ will change and if it is «active», all of its considered kinematic parameters will change. If a joint is an «actuator», only θ_i will change, and if it is «inert», none of its kinematic parameters will change. Let start, end, and unit of movement are denoted byt_s, t_f, and t_u, respectively. Relation 1 and 2 define the Cartesian and relative angular speed of theith joint during execution of the movement.

$$\Delta Point_{i_{t_s}}^{t_f} = \sum_{t=t_s}^{t_f} |Point_i(t+t_u) - Point_i(t)| \tag{1}$$

$$\Delta \theta_{i_{t_s}}^{t_f} = \sum_{t=t_s}^{t_f} |\theta_i(t+t_u) - \theta_i(t)| \tag{2}$$

Using Relations 1 and 2, the formal definitions of the movement roles of theith joint during the period $[t_s, t_f]$ are given in Table 1.

The aim of this study is to find a permutation of linkage joint movement roles with the highest number of «inert» joint and lowest possible number of«active» joints, which enable the end effector to be placed at the target point.

Table 1: The ith joint's movement roles

Dynamic role	Definition
inert	$\vartriangle Point_i = (0,0) \wedge \vartriangle \theta_i = 0$
actuator	$\vartriangle Point_i = (0,0) \wedge \vartriangle \theta_i \neq 0$
passive	$\vartriangle Point_i \neq (0,0) \wedge \vartriangle \theta_i = 0$
active	$\vartriangle Point_i \neq (0,0) \wedge \vartriangle \theta_i \neq 0$

Now the linkage movement minimization problem is defined. A set named$L$ with m numbers is available, in a way that $m = n(n-1)/2$ and n is the number of linkage joints where for every two j_{i_1} and j_{i_2} joints we have $l_{i_1,i_2} = |j_{i_2} - j_{i_1}|(1 \leq i_1, i_2 \leq n, i_1 > i_2)$, and the joints between j_{i_1} and j_{i_2} (if available) are «passive».

The linkage movement minimization problem consists of minimizing the linkage's moving components and their effect on other components in a manner that fulfills the conditions of the reachability problem. In order to minimize the involved components, the condition of maximum «inert» joints must be met. In order to maximize the number of «inert» joints, the «actuator» joint is specified by obtaining the smallest trailing arm sub chain where the target point resides in its reachability space. The closer the «actuator» is to the trailing arm joint; the less other components are involved in the movement. In order to minimize the effect of moving components on linkage joints, we need to minimize «active» joints and the effect of their movement on other components of the open chain. The inputs of this problem are a preliminary configuration of open chain linkages and target point in 2-Dimensional space. The goal in this

$$r_{O_{L'}} = \sum_{j=1}^{k-1} l_{i_{j'},i_{j'+1}} , l_{i_{j'},i_{j'+1}} = |j_{i_{j'}} - j_{i_{j'+1}}| \tag{4}$$

problem is to find a L' subset comprising $\{l_{i_1,i_2}, l_{i_2,i_3}, l_{i_3,i_4}, \ldots, l_{i_{k-1},i_k}\}$ of set L in a way that

$$RA(L') = \begin{cases} true, & r_{I_{L'}} \leq |j_{i_1} - TargetPoint| \leq \\ false, & otherwise \end{cases} \tag{3}$$

$i_1 < i_2 < \cdots < i_k = n$ and conditions for reachability of subset L' are met. Joint $j_{i_1}(1 \leq i_1 \leq n)$ in subset L' acts as an «actuator». The reachability problem of subset L' is given in Relation 3.

where $r_{O_{L'}}$ is derived from Relation 4.

If $l_{M_{L'}}$ is the largest member of set L', then $r_{I_{L'}}$ is derived from Relation 5.

$$r_{I_{L'}} = \begin{cases} 2l_{M_{L'}} - r_{O_{L'}}, & 2l_{M_{L'}} > r_{O_{L'}} \\ 0, & 2l_{M_{L'}} \leq r_{O_{L'}} \end{cases} \tag{5}$$

In order to demonstrate the moving joints' effect on other joints, we need to define a measure entitled Arithmetic Measure[1]. Relation 6 gives the formal definition of this measure.

$$\mathrm{AM}_{L'} = \sum_{j=1}^{k-1} \mathrm{AM}(l_{i_{j'},i_{j'+1}}), \mathrm{AM}(l_{i_{j'},i_{j'+1}}) = (i_k \tag{6}$$

[1] AM

The measure $\mathrm{AM}_{L'}$ states the number of joints affected by the rotation of every moving joint of the arm (Appendix 1).

3. Proof of the NP-Hard Nature of the Linkage Movement Minimization Problem

After conformation of the linkage movement minimization problem, proof of the NP-Hard nature of the problem is proceeding via reduction to the sum of subset problem.

Sum of subset problem is NP-Complete in computer science context. The set *S* comprises *m* available integers. The goal is to find all subsets of *S* in a way that the sum of all members falls between *low* and *high.*

The linkage movement minimization decision problem determines whether for assumed numbers *AM* and *k*, exists a subset where the Arithmetic Measure is more than AM and the number of members are not more than $k-1$. This problem has parameters such as linkage movement minimization in addition to *AM* and *k* parameters.

The sum of subset decision problem determines whether for assumed numbers *low* and *high*, exists a subset where the sum of members falls between *low* and *high.*

Theorem 1: the linkage movement minimization problem is an NP-Hard problem.

Proof includes converting every input sample of the sum of subset problem into an input sample of the linkage movement minimization problem in a way that this transformation meets the following two conditions [16]:

- Transformation must be in polynomial time.
- The solution of the movement minimization problem to an input of the subset sum decision problem must be positive if the solution to its equivalent input in the movement minimization problem is positive.

In order to initiate transformation, set *S* including *m* elements of $\{s_1, s_2, \dots, s_m\}$ and two *high* and *low* integers that are the upper and lower limits of the selected subset sum of the set are considered, then from this input, an input is formed for the movement minimization problem as follows.

Set *S* is ascendingly ordered and subset S' comprising *k* members from set *S* are selected in a way that the sum of its members falls between *high* and *low*. In order to build set *L*, the members of set S' are converted into set $Points = \left\{ j_{i_1} : (0, s_{i_1}), j_{i_1+1} : (0, s_{i_1} - s_{i_2}), j_{i_1+2} : (0, s_{i_1} - s_{i_3}), \dots, j_{i_1+k} : (0, s_{i_1} - s_{i_{i'}}) \right\}$ where this set is considered as the coordinate of open chain joints, in a way that $s_{i_1} < s_{i_2} < \cdots < s_{i_{i'}} = s_k$ and joint j_{i_1} $(1 \le i_1 \le n)$ in this transformation is an «actuator», then we proceed to build set *L* from these coordinates and according to Relation 7.

$$\begin{gathered} \forall i' \in \{i_1, i_1+1, \dots, i_1+k-1\}, \forall k' \in \{1, 2, \dots, k-i'\}: \\ \{l_{i',i'+k'}\} \cup L \ where \ l_{i',i'+k'} = |j_{i'} - j_{i'+k'}| \end{gathered} \tag{7}$$

The optimal measure AM_{max} is set at its maximum value and it is assumed that all joints are

$$\mathrm{AM}_{max} = \sum_{i'=1}^{k} i' \times (k - i') \tag{8}$$

moving. Relation 8 demonstrates the calculation of this measure in the mentioned transformation.

The target point is considered conincinding with j_{i_1+k}. Figure 2 depicts the graphical model of this transformation.

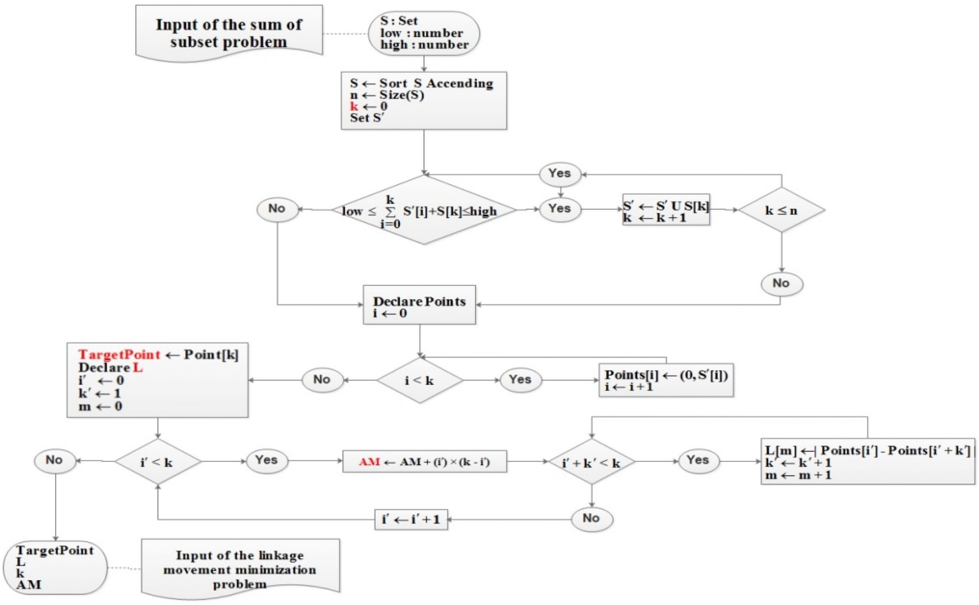

Fig. 2: Transforming an input of the subset sum problem into the movement minimization problem

Figure 3 shows an example of transforming an input of the sum of subset problem into input of the linkage movement minimization problem.

In order to examine the validity of the transformation, it is checked to ensure that the input of the subset sum problem has and $S^{'}$ subset with *k* members where the subset sum falls between *high* and *low*, iff the input for the linkage movement minimization problem possesses an $L^{'}$ subset with maximum $k-1$ members and a maximum Arithmetic Measure of AM_{max}, and Relation 9 is satisfied.

$$r_{I_{L'}} \leq \left| j_{i_1} - j_{i_1+k} \right| \leq r_{O_{L'}} \tag{9}$$

Assume subset S with *k* members exists, and due to the fact that according to the manner of transformation, the largest subset in set *L* has $k-1$ members and Arithmetic Measure of AM_{max}, then at the input of the movement minimization problem, we have subset $L^{'}$ with $k^{'}$ members, in a way that $k^{'} \leq k-1$ and $\mathrm{AM}_{L^{'}} \leq \mathrm{AM}_{max}$. Proof of Relation 9 is presented in Appendix 2.

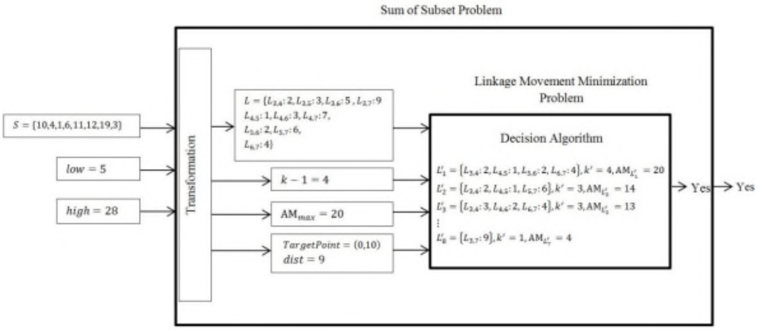
Sum of Subset Problem
$S = \{10,4,1,6,11,12,19,3\}$
$low = 5$
$high = 28$
Transformation
$L = \{L_{3,4}:2, L_{3,5}:3, L_{3,6}:5, L_{3,7}:9$
$L_{4,5}:1, L_{4,6}:3, L_{4,7}:7,$
$L_{5,6}:2, L_{5,7}:6,$
$L_{6,7}:4\}$
$k-1=4$
$\mathrm{AM}_{max}=20$
$TargetPoint = (0,10)$
$dist = 9$
Linkage Movement Minimization Problem
Decision Algorithm
$L'_1 = \{L_{3,4}:2, L_{4,5}:1, L_{5,6}:2, L_{6,7}:4\}, k' = 4, \mathrm{AM}_{L'_1} = 20$
$L'_2 = \{L_{3,4}:2, L_{4,5}:1, L_{5,7}:6\}, k' = 3, \mathrm{AM}_{L'_2} = 14$
$L'_3 = \{L_{3,4}:3, L_{4,6}:2, L_{6,7}:4\}, k' = 3, \mathrm{AM}_{L'_3} = 13$
⋮
$L'_8 = \{L_{3,7}:9\}, k' = 1, \mathrm{AM}_{L'_7} = 4$
Yes
Yes

Fig. 3: Transforming an input of the subset sum problem into the linkage movement minimization problem

On the contrary it is assumed that the input of the linkage movement minimization problem has the subset of $L^{'}$ with $k-1$ members and AM_{max}. Relation 10 explains the creation of the members of subset $S^{'}$.

$$\forall i \in \{i_1, i_1+1, \dots, i_1+k\}: S^{'} \cup \{s_i\} = \begin{cases} s_i = 0, & i = i_1 \\ s_i = \sum_{i^{'}=i_1}^{i-1} l_{i^{'},i^{'}+1}, & i_1+1 \le i \le i_1+k \end{cases} \tag{10}$$

Therefore, subset $S^{'}$ is dependent on subset $L^{'}$, and any alogorithm than can solve the linkage movement minimization problem is used to solve the sum of subset problem.

4. Greedy Approach to Solving the Problem

If n is the number of open chain joints and i is the number of «inert» joints, $n-i-2$ exist, which to fulfill the purpose of the linkage movement minimization problem, must be examined in terms of their «passive» or «active» nature. Total possible configurations for open chain joints are 2^{n-i-2}. Therefore a solution that can lead to an optimal answer will not be possible in Polynomial time. In this section a greedy approach is introduced for solving the linkage movement minimization problem.

The main idea behind this approach is based on the fact that in step one, all involved components must be minimized, and in step two, the effect of movement of these components on other components must be minimized. In order to achieve this purpose, first, by finding the position of the «actuator» joint in the open chain, the smallest trailing sub chain in the arm, which the target point is within its reachability space is obtained, and in the second stage, «passive» joints of the chain are specified.

In the proposed method, the open chain is exmined from finish to start, the first trailing joint of the arm in which the destination point is situated within in its sub chain is returned as the «actuator»joint. Figure 4 demonstrates the method of finding the «actuator» joint.

Algorithm: Finding ActuatorJoint

Input: *Initial Configuration,Target Point*
Output: $i, L^{'}$ // i is the number of «inert» joints **,** $L^{'}$ is subset of set L comprising $\{l_{i,i+1}, \dots, l_{n-1,n}\}$

```
$i \leftarrow n-1$
$L^{'} \leftarrow L^{'} \cup l_{i,i+1}$
while $i > 0$ do
    $i \leftarrow i-1$
    $L^{'} \leftarrow L^{'} \cup \{l_{i,i+1}\}$
    if $RA(L^{'})$ = true then
        break
    end if
end while
if $RA(L^{'})$ = false then
    return "unreachable target point'
else
    return $i, L^{'}$
end if
end Algorithm
```

Fig. 4: Finding «actuator» joint

The output of the joint is $j_i(1 \le i \le n)$ which a «actuator» joint and all preceding joints are static. In addition, subset L' is one of the other outputs that is the smallest trailing sub chain of the arm.

In the next stage, the goal is to find the «passive» joints between joint j_i $(1 \le i \le n)$ until the last joint in the arm. In other words, in this stage the goal is to eliminate a numebr of members from subset L' and substitute them with a lower number of members from set L. In order to achieve this, for every j_k joint between j_i to j_n, first we assume that it is a «passive» joint. With the assumption of j_k $(1 \le k \le n)$being a «passive» joint, L' is created and reachability conditions are examined. If the conditions are not met, j_i $(1 \le i \le n)$ transforms into an «active» joint. Figure 5 shows the algorithm for finding «passive» joints.

Algorithm: Finding PassiveJoints

Input: $L', i, Target\ Point$ *//L', i from Find Actuator Joint*

Output: L' //L' is $\{l_{i_1,i_2}, l_{i_2,i_3}, l_{i_3,i_4}, \ldots, l_{i_{k-1},i_k}\}$ in a way that $i_1 < i_2 < \ldots < i_k = n$

```
$k \leftarrow 0$
$start \leftarrow i$
$passive \leftarrow i + 1$
$end \leftarrow passive + 1$
while $k < n - i - 1$ do
    $L' \leftarrow L' - \{l_{start,passive}\}$
    $L' \leftarrow L' - \{l_{passive,end}\}$
    $L' \leftarrow L' \cup \{l_{start,end}\}$
    if $RA(L') = false$ then
        $L' \leftarrow L' - \{l_{start,end}\}$
        $L' \leftarrow L' \cup \{l_{start,passive}\}$
        $L' \leftarrow L' \cup \{l_{passive,end}\}$
        $start \leftarrow passive$
    end if
    $passive \leftarrow passive + 1$
    $end \leftarrow passive + 1$
    $k \leftarrow k + 1$
end while
return $L'$
end Algorithm
```

Fig. 5: Finding passive joints

The input for this algorithm is the output of it previous algorithm, and its output is the subset L'. The first step in the proposed approach possesses a time and space complexity of $O(n)$. In the second phase, *k* joints must be exmained in terms of «active» or «passive» state, which this exmaination must be carried out for *k* joints in every stage of the algorithm. In the worst case, if the «actuator» joint is the first joint of the arm, the second step will be arried out in the time $O(n^2)$. Thus, the time complexity of the proposed approach is $O(n^2)$ and the space complexity is $O(n)$.

Figure 6 shows an example of the decision tree of the proposed solution for «passive» joints. The word «Yes» on each branch states establishment of $RA(L')$ for L' in each node, and «No» states the non-establishment of such.

5. Experimental Results

In this section, while presenting the experimental results of the proposed greedy approach, we will attempt to compare the obtained results by considering them alongside optimal results. The conducted simulation is the result of implementing the proposed approach explained in the previous section 1000 random linkages and each of 1000 random target points. The number of links are between 4 to 20, and the length and initial configuration of every linkage is random. In more than 80% of the cases, the results of the greedy approach with optimal results are identical.

Moreover, these results show that in most cases, by applying this method, more than half of the joints are static and thus, are eliminated from the main chan. In addition, the results of this simulation state that the obtained sub chain, in more than half of the joints, does not require any modification to relative angles. These results are provided in Table 2. The results are from an average 10 iterations of the simulation.

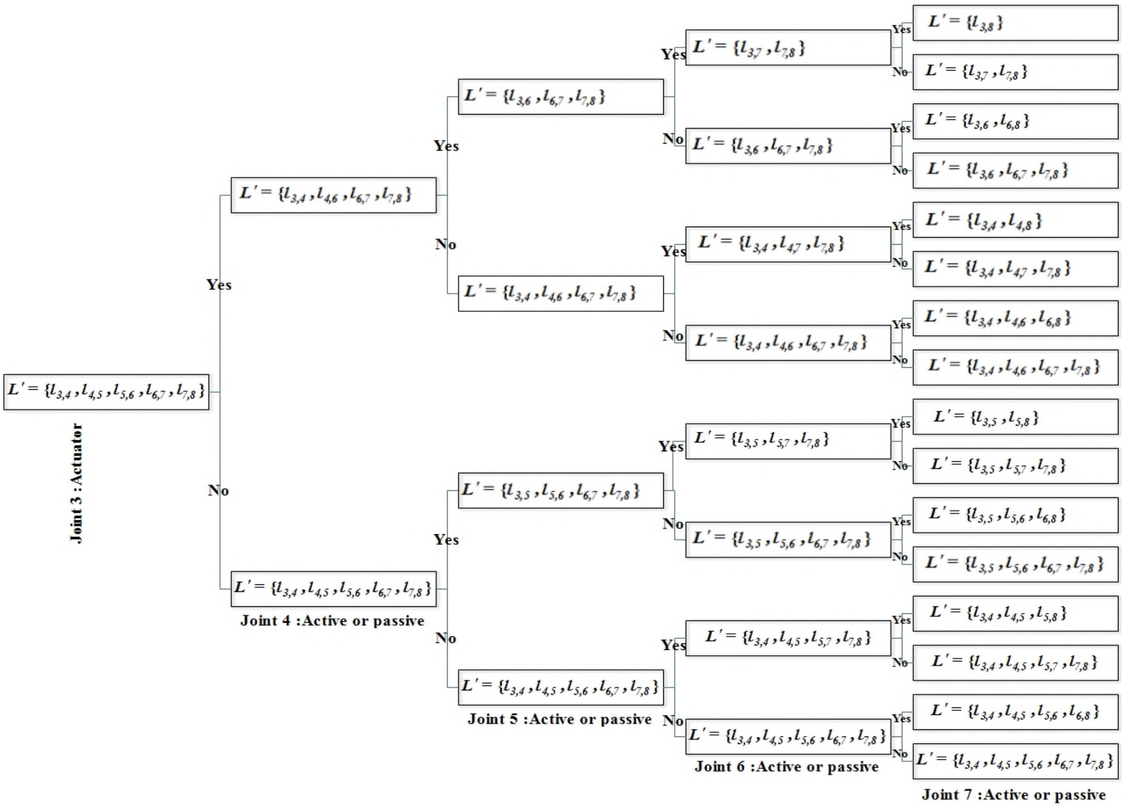
$L' = \{l_{3,4}, l_{4,5}, l_{5,6}, l_{6,7}, l_{7,8}\}$
Joint 3 :Actuator
Yes
$L' = \{l_{3,4}, l_{4,6}, l_{6,7}, l_{7,8}\}$
No
$L' = \{l_{3,4}, l_{4,5}, l_{5,6}, l_{6,7}, l_{7,8}\}$
Joint 4 :Active or passive
Yes
$L' = \{l_{3,6}, l_{6,7}, l_{7,8}\}$
No
$L' = \{l_{3,4}, l_{4,6}, l_{6,7}, l_{7,8}\}$
Yes
$L' = \{l_{3,5}, l_{5,6}, l_{6,7}, l_{7,8}\}$
No
$L' = \{l_{3,4}, l_{4,5}, l_{5,6}, l_{6,7}, l_{7,8}\}$
Joint 5 :Active or passive
Yes
$L' = \{l_{3,7}, l_{7,8}\}$
No
$L' = \{l_{3,6}, l_{6,7}, l_{7,8}\}$
Yes
$L' = \{l_{3,4}, l_{4,7}, l_{7,8}\}$
No
$L' = \{l_{3,4}, l_{4,6}, l_{6,7}, l_{7,8}\}$
Yes
$L' = \{l_{3,5}, l_{5,7}, l_{7,8}\}$
No
$L' = \{l_{3,5}, l_{5,6}, l_{6,7}, l_{7,8}\}$
Yes
$L' = \{l_{3,4}, l_{4,5}, l_{5,7}, l_{7,8}\}$
No
$L' = \{l_{3,4}, l_{4,5}, l_{5,6}, l_{6,7}, l_{7,8}\}$
Joint 6 :Active or passive
Yes
$L' = \{l_{3,8}\}$
No
$L' = \{l_{3,7}, l_{7,8}\}$
Yes
$L' = \{l_{3,6}, l_{6,8}\}$
No
$L' = \{l_{3,6}, l_{6,7}, l_{7,8}\}$
Yes
$L' = \{l_{3,4}, l_{4,8}\}$
No
$L' = \{l_{3,4}, l_{4,7}, l_{7,8}\}$
Yes
$L' = \{l_{3,4}, l_{4,6}, l_{6,8}\}$
No
$L' = \{l_{3,4}, l_{4,6}, l_{6,7}, l_{7,8}\}$
Yes
$L' = \{l_{3,5}, l_{5,8}\}$
No
$L' = \{l_{3,5}, l_{5,7}, l_{7,8}\}$
Yes
$L' = \{l_{3,5}, l_{5,6}, l_{6,8}\}$
No
$L' = \{l_{3,5}, l_{5,6}, l_{6,7}, l_{7,8}\}$
Yes
$L' = \{l_{3,4}, l_{4,5}, l_{5,8}\}$
No
$L' = \{l_{3,4}, l_{4,5}, l_{5,7}, l_{7,8}\}$
Yes
$L' = \{l_{3,4}, l_{4,5}, l_{5,6}, l_{6,8}\}$
No
$L' = \{l_{3,4}, l_{4,5}, l_{5,6}, l_{6,7}, l_{7,8}\}$
Joint 7 :Active or passive

Fig. 6: An example decision tree for selecting «passive» joints

Table 2: Results obtained for 1000 linkage simulations in 1000 target points

Percent of inert joints		Percent of AM similarity of optimal results with greedy results	Joint classification	Greedy results		Optimal results	
				Percent of passive joints relative to total	Percent of passive joints relative to static joints	Percent of passive joints relative to total	Percent of passive joints relative to static joints
Without static joint	13.5	94.5	Less than half	11.5	85	12.2	90.4
			More than half	2	15	1.3	9.6
Less than half	65.2	87.2	Less than half	44.8	68.7	52.7	80.7
			More than half	20.4	31.3	12.6	19.3
More than half	21.3	80.8	Less than half	11.5	53.7	15	70.5
			More than half	9.8	46.3	6.3	29.5

6. Conclusion

In this paper, a new NP-Hard problem was proposed regarding open chain linkage movement. Providing a greedy approach was considered after the precise definition of linkage movement minimization problem and approval of being NP-Hard. Experimental results show that the results of applying this approach with optimal values are similar results in 80% of cases. Moreover, the results show that the proposed approach fulfills the goal of minimization of linkage components. Furthermore, this method leads to erosion of arm, reduces energy consumption and the required parameters and variables for calculating the final configuration of linkages. Future research would be considered on the studying applications of the other types of robot arms.

Appendix 1:

For example, for the subset $L' = \{l_{2,3}, l_{3,8}, l_{8,11}\}$, the third and eighth joints are «active» The second joint is an «actuator», the first joint is «inert», and other joints are «passive». Furthermore, $\mathrm{AM}(l_{3,8}) = (11-3)*2$ and states that eight joints (from joint 3 to joint 11) are affected by the rotation of two joints (joint 2 and 3). In this subset,$\mathrm{AM}_{L'} = 9+16+9$.

Appendix 2:

Proof of Relation 9 is done using the triangle inequality theorem, Relation 4, and Lemma 1.
Lemma 1: for every non-negative numbers a, b, and c where$a \le b \le c$, the relation $2a - c \le b$ holds.
Proof: If $c = a + l$, we have $2a - a - l \le b$. Thus, $a - l \le b$ and Lemma 1 is proven.

Using the triangle inequality theorem and definition for$r_{O_{L'}}$ in Relation 4, establishment $\left|j_{i_1} - j_{i_1+k}\right| \le r_{O_{L'}}$of is proven.Considering that members of the subset S' are ordered ascendingly, $\left|j_{i_1} - j_{i_1+k}\right|$ is the largest member of set *L*, and thus, for every subset L' with its largest member $l_{M_{L'}}$, the relation $\left|j_{i_1} - j_{i_1+k}\right| \geq l_{M_{L'}}$ is established. If in Lemma 1 $r_{O_{L'}}$, $\left|j_{i_1} - j_{i_1+k}\right|$, and $l_{M_{L'}}$ are input for a, b, and c, respectively.

References

[1] Devadoss, S.L. and O'Rourke, J., "Discrete and Computational Geometry", *Princeton University Press*, 2011, pp. 215-219.

[2] O'Rourke, J., "Computational Geometry In C", *Cambridge University Press*, 1997, pp. 322-331.

[3] Zhang, Y., Lv, X., Li, Z., Yang, Z., and Chen, K., "Repetitive motion planning of robot arm subject to joint physical limits and using LVI-based primal–dual neural network", *Mechatronics*, vol. 18, no. 9, pp. 475-485, 2008.

[4] Chettibi, T.; Lehtihet, H. E.; Haddad, M.; Hanchi, S., "Minimum cost trajectory planning for industrial robots", *European Journal of Mechanics A/Solids*, vol. 23, p. 703–715, 2004.

[5] Choi, J. and Amir, E., "Factor-Guided Motion Planning for a Robot Arm" , *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, 2007.

[6] Ding, H., Zhou, M., and Stursberg, O., "Optimal Motion Planning for Robotic Manipulators with Dynamic Obstacles using Mixed-Integer Linear Programming", *Mediterranean Conference on Control & Automation*, Makedonia Palace, Thessaloniki, Greece, 2009.

[7] Demain, E. and O'Rourke, J., "Geometric and Folding Algorithms", *Cambridge, University Press*, 2007, pp. 9-11,29-31,59-67.

[8] Canny, J.F., "The Complexity of Robot Motion Planning", *MIT Press*, 1987, pp. 180-185.

[9] Connelly, R. and Demaine, E., "Geometry and Topology of Polygonal Linkages", *Discrete and Computational Geometry*, pp. 213-235, 2004.

[10] Hopcroft, J.,O'Rourke, J., and Whitesides, S., "On the movement of robot arms in 2-dimensional bounded regions", *SIAM J. Computer*, vol. 14, no. 2, pp. 315-333, 1985.

[11] Lieutier, A. and Rameau, j., "Mechanical linkage design and NP-hardness", *Mechanism and Machine Theory*, vol. 82, pp. 97-114, 2014.

[12] Traversa, F., Ramella, C., Bonani, F., and Ventra, M., "Memcomputing NP-complete problems in polynomial time using polynomial resources and collective states", *Science Advances*, vol. 1, no. 6, pp. 1-9, 2015.

[13] Nourollah, A. and Razzazi, M.R., "Minimum cost open chain reconfiguration", *Discrete Applied Mathematics*, vol. 159, pp. 1418-1424, 2011.

[14] Nourollah, A. and Razzazi, M.R., "A Linear Time Approximation Algorithm for Ruler Folding Problem", *Journal of Universal Computer Science*, vol. 14, no. 4, pp. 566-574, 2008.

[15] Nourollah, A. and Razzazi, M.R., "Near-Optimum Folding for a Reconfigurable Snake-Like Robot ", *Advanced Robotics*, vol. 23, pp. 239-256, 2009.

[16] Haasea, C. and Kiefer, S., "The Complexity of the Kth Largest Subset Problem and Related Problems", *Information Processing Letters*, vol. 116, no. 2, pp. 111-115, 2016.