Journal of Electrical and Computer Engineering Innovations
(JECEI)
Journal homepage: http://www.jecei.sru.ac.ir

Research paper

Robot arm Reconfiguration to Minimization Moving Parts

A. Nourollah**, N. Behzadpour

Software Systems Research and Development Laboratory, Faculty of Computer Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran.

Article Info

Article History:
Received 08 August 2017
Revised 10 December 2018
Accepted 17 March 2018

Keywords:
Formal modeling
Robot arm
Linkage reconfiguration
Reachability problem
Computational geometry

*Corresponding author's
Email Address:
nourollah@sru.ac.ir

Extended Abstract

Background and Objectives: This paper presents a new optimization problem in the field of linkage reconfiguration. This is the problem of minimizing moving parts of a given robot arm for positioning the end effector of the given robot arm at the given target point as well as minimizing the movement of the movable parts.

Methods: Initially, formal modeling is accomplished by minimizing the movement problem. At this time, a criterion called AM (Arithmetic Measure) is introduced, and this criterion is used to quantify the motion of the linkage. Afterward, it is indicated that the presented problem is an NP-Hard problem. Consequently, a greedy heuristic algorithm is presented to minimize the movement of the robot's moving components. After identifying the moving components and the movement of these parts, an algorithm is provided to determine the final configuration of the robot arm.

Results: The results indicate that the discussed model successfully reduced the moving parts of the robot arm. Moreover, the results show that the proposed approach fulfills the goal of minimization of the 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 the linkages.

Conclusion: The mentioned algorithm solves the problem by mapping the robot arm with an arbitrary number of links to a robot with a single link or two links. The proposed heuristic approach requires $O(n^2)$ time using $O(n)$ space.

Introduction

Today, robot arms are widely used in the fields of military applications, exploration, transportation, navigation, rescue, medicine, household applications, and entertainment. The use of mobile robots in areas that are not reachable by humans or are dangerous, such as factories, other planets, oceans, deep ground, and military zones is a proper solution. Robots are classified by their movements and tasks. Some examples of robots include stationary, wheeled, legged, flying, and diver robots [1]. Each robot serves a special purpose. Among the existing robot types, industrial robots have more uses [2]. In this regard, there is often debate over the nature of industrial robots.

The study of the mechanics and problems of robot arms is not a new science. It is rather a set of titles derived from the classic backgrounds. Mechanical engineering provides methods for the study of machines in the statistic and dynamic states. Mathematics also provides methods of describing the properties of motion. Computational geometry increases the abstract understanding of robot arm motions and problems. The control theory also presents a means of designing and assessing algorithms for understanding the satisfactory motions and forces. Power engineering deals with the design of sensors for robot arms and computer science sets the scene for programming these robots for certain tasks [3]. The complexity of robot problems prevents the

development of related applied algorithms. For instance, a rescue robot must find a path to allow the end effector access the ruins and the debris must be picked without any unnecessary movement.

Every robot arm is a collection of links and joints, which vertically equal a connected graph expressed as *G= (Joints, Links)*. The vertices of this graph represent the joints and the edges represent the rigid objects called the links. The joints are capable of rotation. In this graph, the length of the links and the angles created between the joints through the connection of the links are of importance [4].

As there are different types of graphs, different methods of connection between the links and joints result in different models of robot arms. A robot arm is a robot in which there is only one path between two selected joints.

One of the problems with robot arms is the problem of reachability. In the decision-making version of this problem, the question is whether a given point on the arm (usually the ending point) can access a point in the Cartesian space housing the arm [5].

A robot arm, which is expressed as $\langle l_1, l_2, \dots, l_{n-1} \rangle$, is a sequence of $n-1$ interconnected links. Moreover, l_i denotes the length of the graph edges and is a positive real number. The reachability space of the arm is the area between two concentric circles centered on the first arm joint, and assuming that the external radius is shown by r_o, (1) is used for the calculation of this parameter [5]. If l_M is the length of the longest link and r_I is the internal radius, (2) is used to calculate this parameter [5].

$$r_O = \sum_{i=1}^{n-1} l_i \tag{1}$$

$$r_I = \begin{cases} 2l_M - r_O, & 2l_M > r_O \\ 0, & 2l_M \le r_O \end{cases} \tag{2}$$

Moreover, if $r_I \le 0$, then the reachability space is a circle with the r_O radius [4],[5] (Fig. 1).

The configuration is a certain state of the robot arm (at a certain moment), in which the coordinates of each point on the arm can be determined. A collection of all of the possible configurations forms the configuration space.

The robot arm reconfiguration problem is another important problem with these arms. The result of this problem becomes significant for a given input if a positive solution is found. In this problem, the arm motion path must be clearly determined to obtain the robot arm reconfiguration [6]. Fig. 2: Possible configurations of a single-link open-chain for the problem of accessing the target point. depicts two of the possible configurations of an open robot arm with two links for the problem of accessing the target point.

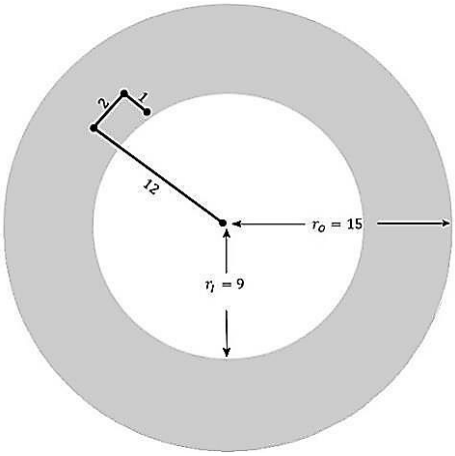


Fig. 1: Reachability space of the robot arm (the external radius is $r_O = 15$ and the internal radius is $r_I = 9$). The entire gray area is the arm reachability space.

Related Works

Since multiple solutions exist for every reconfiguration problem, the goals of such problems can be finding the minimum energy consumption for the robot arm movements, the shortest path, the minimum duration, the minimum area and perimeter of the robot arm in the two-dimensional space, the concave membrane volume in the three-dimensional space, the geometry of the robot arm perimeter, and such.

In [7] a solution is proposed for minimizing the motion cost of robot arms in predefined geometric paths based on the dynamic programming model. In [8], the formalization of a robot arm with six degrees of freedom (DOF) and the ships between the parts in the course of movement was used to solve the forward kinematic problem. In [9], following a formal description of the motions of the surgical robot arm, the automatic theorem proving (ATP) was used to validate the results. Limiting the angular positions of the joints and the velocity and acceleration of the joints is another goal considered in [10] to solve the robot arm motion problem. In [11] an algorithm for the robot arm motion is introduced to omit the intersection between the arm links during motion. Preventing the collision between the arm and the external moving obstacles and imposing limitations on the kinematic variable parameters of the obstacles form the basis of an artificial neural network algorithm which was discussed in [12] in studying the robot arm motions. Minimization of the motion time and elimination of the obstacles are other goals considered in [13] in designing the robot arm path.

The use of approximation and near-optimum algorithms in solving NP-Hard problems has also captured attention [14]. In [15], based on the formation of the robot arm and the reconfiguration problem, a low-cost algorithm with time complexity $O(n^2)$, and memory usage O(n), was proposed for the movement of

a robot arm from the current configuration to the optimum configuration in the one-dimensional space. In [16] a linear time approximation algorithm is introduced for solving the robot arm folding problem, and [17] presents an approximation algorithm with the O(nlogn) time complexity and O(n) memory usage for solving the folding of a snake-shaped robot. The latter algorithm guarantees that the length of the arm does is not twice as long as the longest link.

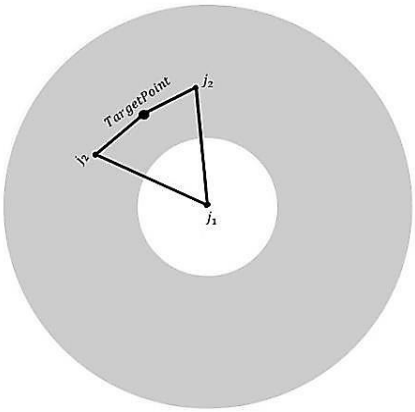


Fig. 2: Possible configurations of a single-link open-chain for the problem of accessing the target point.

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 [18] is an NP-Complete problem, which proof of this subject is presented in [19] 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 [20].

A set of components required for building a mechanism includes rigid bodies accompanied by rotary joints. The goal is to obtain a component layout where connection limitations are not violated. Proving the NP-Hard nature of this problem is offered in [21] via the help of reduction of subset sum. This problem by adding the next parameter is NP-Hard, which in [21] is proven using reduction to 3SAT problem. In [22], an approximate algorithm with time complexity O(1) for navigation polygonal linkage is presented.

The present research goal was to minimize the moving parts of a robot arm. We introduce and formalize a new problem in the field of robot arm movement that has not been addressed by our research so far. Then we show the problem is NP-Hard. This has been used to reduce the SUBSET-SUM problem, which is an NP-Complete problem. Then, an exponential algorithm is proposed for which the algorithm is $O(n^3)$ when the number of links is less than 10. To this end, first, the shortest ending arm chain, which had the target point in its reachability space, was found to minimize the number of the links forming the chain. Next, by maximizing the chain joints that had no relative angular variation, the permutation of the status of the joints required for calculating the final robot arm configuration was obtained. In this permutation, the engaged joints and the type of movement of each joint were determined for the placement of the end effector on the target point. In the end, a method was developed for mapping the chains with more than two links to a dual-link arm reconfiguration problem. It has also introduced a greedy heuristic for when the number of links is more than 14. We show that this greedy approach runs in O(n).

This paper is organized as follows. The next section consists of 4 parts. In this section, the robot arm structure is first formalized. Then the problem of movement minimizing is formally expressed. This section contains 4 sub-sections. In the first sub-section, the moving parts are minimized. In the second part, the minimization of the movement of moving parts is described. In the third sub-section, it is demonstrated that this is an NP-Hard problem. The following sub-section illustrates the exponential states of the answer and presents a greedy algorithm requires $O(n^2)$ time using O(n) space. In the third section, the algorithm for obtaining the final reconfiguration is presented. In the final sub-section, the time complexity and memory consumption of the proposed method are discussed. Section 4 presents the simulation results of the algorithms and comparing the related works with the proposed method, and section 5 concludes the paper.

The Proposed Method

The target point in the Cartesian space and the initial configuration of the robot arm are the problem inputs, and the robot arm reconfiguration calculation is carried out by adopting the determined minimization criteria. Figure 3 shows the overall schema of the proposed method. In the movement minimization section, first, determine the minimum number of links, joints (moving parts) needed to move to the destination point and identify those arms. In this section and the following, the obtained motion is optimized, in the sense that the movement of the moving parts is minimized. In the last section where the problem outputs, the optimal configuration is obtained. Each of the schema sections in Fig. 3, is discussed in this article. This section consists of 4 parts. First, a formal definition of the robot arm is presented along with the description of the constant parameters, the kinematic variable, and the defined hierarchical structure.

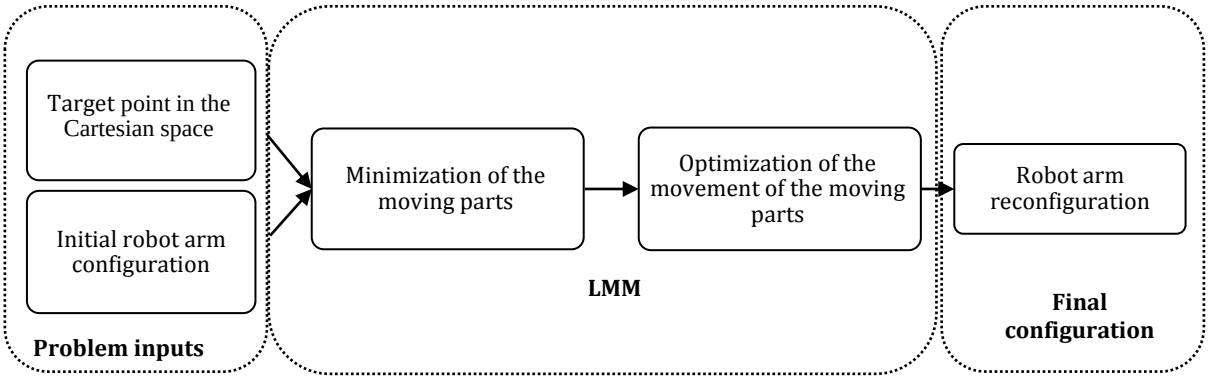


Fig. 3: Overall schema of the proposed method.

In part 2, minimization is carried out, and the linkage final configuration calculation model is described in part 3. In part 4 of this section, the complexity of the proposed method is discussed

A. Formal Definition of the Robot arm

The robot arm uses the linkage to move along a certain path (whose parameters are derived from the smooth functions) without sudden movements so that the end effector is placed on the target point. The formal description of the robot arm and the hierarchical structure as stated, the graph corresponding to the robot arm is a simple path. Hence, a robot arm, which is written as $\langle l_1, l_2, \dots, l_{n-1} \rangle$, is a sequence of $n-1$ interconnected links, where l_i shows the length of the graph edges and is a positive real number. The vertices of this graph represent the arm joints and are used to connect the links. The beginning and end joints of each l_i are shown with the (j_i, j_{i+1}) ordered the j_1 joint is connected to the earth while the j_n joint is free. Each j_i is a tuple as $(\text{Point}_i, \theta_i)$ where $\text{Point}_i: (x_i, y_i)$ ordered triple. Moreover, Point_i shows the Cartesian coordinates of the j_n joint in the two-dimensional space and θ_i is the relative angle between two links(l_i and $, l_{i+1}$)(counterclockwise). θ_1 is the angle between the first joint and the positive side of the x axis, and θ_n is zero and it is the constant parameter of nth joint. The Cartesian coordinates of the first joint also form the kinematic constant of this joint (the top of the robot arm is connected to the ground). In this method, many subsidiary joints are defined for each joint in addition to the motion parameters, and each joint is influenced by the motions of the principal joints. Motion occurs from the principal joint to the subsidiary joints. The subsidiary joints of the j_i joint are expressed as $s_i = \{j_{i+1}, \dots, j_n\}$. Each joint enters one of the active, actuator, passive, or inert states at the time of movement of the robot arm depending on its effect on its subsidiary joints or the effect of the principal joints on it.

If the ith joint rotates in $[t_s, t_f]$ range at the $\ddot{\theta}_i$ angle, its subsidiary joints rotate at this angle and become the source of Point_i. This motion causes a change to the Cartesian parameters of the subsidiary joints of this joint, and the Cartesian parameters of the motion-generating joint remain unchanged while the relative angle changes. The ith joint is called the "actuator" and the subsidiary joints whose Cartesian parameters change due to the rotation of the ith joint are called the "passive" joints. Hence, during motion, the Cartesian coordinates of a joint are a function of the relative angular movement of one or several principal joints. If one of the subsidiary joints shows angular movements, all of the kinematic variable parameters of the joint change. This joint is called the "active" joint. If a particular joint is not the source of movement and is not affected by the movement of another joint, its kinetic parameters are constant and it is called an "inert" joint. Hence, during the movement of a robot arm, only one joint is the actuator, which generates the motions and is essential to the significance of the motions. This joint has the highest propagation of motion in terms of the number of joints, and its precedent joints (if any) are inert joints. In this analysis, j_n joint, as the end effector of the robot arm, always plays a passive role. The joints between the actuator joint and the end effector may be either active or passive. If time unit is shown by t_u, (3) defines the Cartesian velocity and (4) defines the relative angular velocity of the ith joint during the movement.

$$\Delta Point_{t_{s_i}}^{t_f} = \sum_{t=t_s}^{t_f} \left| Point_i(t+t_u) - Point_i(t) \right|$$

$$= \sum_{t=t_s}^{t_f} \left(\left| x(t+t_u) - x(t) \right|, \left| y(t+t_u) - y(t) \right| \right) \quad (3)$$

$$\Delta\theta_{t_{s_i}}^{t_f} = \sum_{t=t_s}^{t_f} \left| \theta_i(t+t_u) - \theta_i(t) \right| \quad (4)$$

Using s (3) and (4) the formal definitions of the dynamic roles of the ith joint in the $[t_s, t_f]$ range were obtained as listed in Table 1.

Each joint accepts one of the roles listed in Table 1 in

each selected range.

Table 1: Dynamic roles of the ith joint

Dynamic role	Definition
Inert	$\Delta Point_{t_s\,i}^{t_f} = (0,0)\ and\ \Delta\theta_{t_s\,i}^{t_f} = 0$
Actuator	$\Delta Point_{t_s\,i}^{t_f} = (0,0)\ and\ \Delta\theta_{t_s\,i}^{t_f} \neq 0$
Passive	$\Delta Point_{t_s\,i}^{t_f} \neq (0,0)\ and\ \Delta\theta_{t_s\,i}^{t_f} = 0$
Active	$\Delta Point_{t_s\,i}^{t_f} \neq (0,0)\ and\ \Delta\theta_{t_s\,i}^{t_f} \neq 0$

B. LMM Problem

The main idea for this research was that for the placement of the end effector at the target point, first the moving parts and then the motions of the moving parts are minimized. Fig. 4, illustrates the movement minimization process. The outline of this section is illustrated in Fig. 3(Movement Minimization Section), and is elaborated in this section.

I) Minimizing the Moving Parts

Defining the number of the inert joints as the first-stage decision variable leads to the definition of a notion known as the "effective sub-chain". If a chain connects two joint points, it is shown by $\mathrm{SC_k}$, where $\mathrm{k} \in \{1,2,\dots,\mathrm{n}-\mathrm{k}\}$ and describes a chain connecting joint k to joint n. The $\mathrm{SC_k} = \langle \mathrm{l_k}, \mathrm{l_{k+1}}, \dots, \mathrm{l_{n-1}} \rangle$ shows the links of a sub-chain. The joints between $\mathrm{j_1}$ and $\mathrm{j_{k-1}}$, are inert joints, while $\mathrm{j_k}$ is the actuator.

The $\mathrm{SC_k}$ reachability space is the area between two concentric circles with $\mathrm{j_k}$ as the center, and assuming that $\mathrm{r_{O_k}}$ is the external radius; (5) is used to calculate parameter.

$$r_{O_k} = \sum_{i=k}^{n-1} l_k \tag{5}$$

where, l_k is the length of link kth link. In addition, if r_{I_k} shows the internal radius of the sub-chain, (6) is used to calculate this parameter.

$$r_{I_k} = \begin{cases} 2l_{M_k} - r_{O_k}, & 2l_{M_k} > r_{O_k} \\ 0, & 2l_{M_k} \leq r_{O_k} \end{cases} \tag{6}$$

where, l_{M_k} is the length of the longest link of SC_k. If the effectiveness of the SC_k sub-chain is expressed as $E(SC_k)$, (7) provides its mathematical definition

$$E(SC_k) = \begin{cases} true, & RA(k,tp) = true\ and \\ & \forall i \in \{k+1,\dots,n-1\}: \\ & RA(i,tp) = false \\ false, & otherwise \end{cases} \tag{7}$$

where, tp shows the target point coordinates. In addition, (8) yields $RA_{k,tp}$

$$RA(k,tp) = \begin{cases} true, & r_{I_k} \leq |j_k - tp| \leq r_{O_k} \\ false, & otherwise \end{cases} \tag{8}$$

Algorithm 1 shows the algorithm for finding an effective sub-chain using the defined s.

Algorithm: Finding $\mathrm{SC_k}$

Input: Initial Configuration,Target Point
Output:$\mathrm{SC_k} = \langle \mathrm{l_k}, \mathrm{l_{k+1}}, \dots, \mathrm{l_{n-1}} \rangle$

```
k ← n − 1
SC_k ← l_{n−1}
E(SC_k) ← RA(k, TP)
while k > 0 or E(SC_k) = false do
    k ← k − 1
    if RA(k, TP) = true then
        SC_k ← SC_k ∪ l_{k−1}
        E(SC_k) ← true
    end if
end while
if E(SC_k) = false then
    return "unreachable target point"
else
    return SC_k
end if
end Algorithm
```

Algorithm 1: Algorithm for finding an effective sub-chain.

II) Minimizing the Movement of Moving Parts

The number of the active joints, which is the decision variable of the second stage, leads to the definition of the virtual link and virtual sub-chain notions. A virtual link is a link obtained by replacing some links with one link such that the relative angle between the substitute links does not change during movement. (9) is the formal expression of this definition.

$$\begin{aligned} & l_{p_1,p_2}: \\ & Length\left(l_{p_1,p_2}\right) = \left|j_p - j_{p'+1}\right|, \\ & \forall i \in \{p+1,..,p'\}, \forall t \in [t_s, t_f]: \theta_i(t) = \theta_i(t_s) \end{aligned} \tag{9}$$

If a chain consists of one or several virtual links it is called a virtual sub-chain and is shown by $\mathrm{VSC_{k,m}}$, where m denotes the number of the actual and virtual links of that chain. The $\mathrm{VSC_{k,m}}$ virtual sub-chain consists of a sequence of links as $\langle \mathrm{l_{p_1,p_2}}, \mathrm{l_{p_2,p_3}}, \mathrm{l_{p_3,p_4}}, \dots, \mathrm{l_{p_m,p_{m+1}}} \rangle$ with $\mathrm{p_1} < \dots < \mathrm{p_{m+1}}$. The reachability of the target point (TP) for a virtual sub-chain is calculated by considering its virtual and actual links. (10) presents the reachability conditions for the $\mathrm{VSC_{k,m}}$ virtual sub-chain.

$$RA(VSC_{k,m}, tp) = \begin{cases} true, & r_{I_{VSC_{k,m}}} \leq |j_k - tp| \leq r_{O_{VSC_{k,m}}} \\ false, & otherwise \end{cases} \tag{10}$$

where, $r_{OVSC_{k,m}}$ is the external radius of the virtual sub-chain and is calculated by (11).

$$r_{O_{VSC_{k,m}}} = \sum_{j'=1}^{m} l_{p_{j'},p_{j'+1}}, l_{p_{j'},p_{j'+1}} = \left|j_{p_{j'}} - j_{p_{j'+1}}\right| \tag{11}$$

In addition, if the longest link of the virtual sub-chain is shown by $l_{MVSC_{k,m}}$ and the internal sub-chain radius is indicated by $r_{I_{VSC_{k,m}}}$, (12) gives the calculation of this parameter.

$$r_{I_{VSC_{k,m}}} = \begin{cases} 2l_{M_{VSC_{k,m}}} - r_{O_{VSC_{k,m}}}, & 2l_{M_{VSC_{k,m}}} > r_{O_{VSC_{k,m}}} \\ 0, & 2l_{M_{VSC_{k,m}}} \le r_{O_{VSC_{k,m}}} \end{cases} \quad (12)$$

The selection of a state from some possible permutations of the joints between the actuator joint and the end joint based on their passive or active nature defines the arithmetic measure. This measure reflects the propagation of the angular movement of the joints, and thus it is the target of the minimization. The rotation of each joint affects the Cartesian coordinates of its subsidiary joints. Hence, as the distance between the initial point and the end effector decrease, fewer joints are influenced by the rotation. The decision variable of the third phase of the analysis is provided in the following. (13) shows the assessment of this measure.

$$AM_{VSCk,m} = \sum_{j'=1}^{m} AM(l_{p_{j'},p_{j'+1}}),$$
$$AM(l_{p_{j'},p_{j'+1}}) = \left(p_{m+1} - p_{j'}\right) \times j' \quad (13)$$

The AM (arithmetic measure) criterion reflects the effect of the movement of the joints with relative angular motions on the subsidiary joints, and thus its minimum value is desired.

Two different states of the AM criterion are depicted in Fig. 5 for a virtual sub-chain with an equal number of active joints.

To find the joints involved in the movement and identify their types, steps are taken based on the introduced decision variables.

In an effective sub-chain with $n-k$ joints (with k showing the number of the static joints), the number of the different linkage states that can be defined is 2^{n-k-2}.

III) Confirming the NP-Hard Nature of the problem of LMM

After confirming the LMM problem, the problem was proved to have an NP-Hard nature by decreasing the sum of the subset problem.

The sum of the subset problem is NP-Complete in the context of computer science. The S set has m available integers.

The objective is to discover all the subsets of S in such a way that the sum of all the members would be between the two levels of low and high.

The LMM decision problem decides if for the given numbers of AM and k, there is a subset where the Arithmetic Measure is greater than the number of AM and whether the number of members is not higher than $k-1$. This problem consists of parameters like LMM apart from the parameters of AM and k.

The sum of the subset decision problem can determine if, for the given numbers of low and high, there is a subset where the sum of the members would be between low and high.

Hypothesis 1: the LMM problem is actually an NP-Hard problem. It is proved by transforming every input sample of the sum of the subset problem to an input sample of the LMM problem in such a way that this transformation would meet the two following conditions [23]:

1- The transformation must take place in polynomial time.

2-The result of the movement minimization problem for an input of the subset sum decision problem has to be positive if the result of its equivalent input in the movement minimization problem is actually positive.

To start the transformation, the S set which consists of m elements from $\{s_1, s_2, \ldots, s_m\}$ as well as two *high* and *low* integers which represent the upper and lower bounds of the selected subset sum of the set are selected. Next, using this input, an input item is created for the movement minimization problem which is displayed here. Let S be an ascending set of links and let be $S' \subseteq S$ which consists of k elements of set S that selected in such a way that the sum of them would be between *high* and *low*. To create set L, the elements of set S' are changed into the set of points as follows

$$Points = \{ j_{i_1} : (0, s_{i_1}), j_{i_1+1} : (0, s_{i_1} - s_{i_2}), j_{i_1+2} : (0, s_{i_1} - s_{i_3}), \ldots, j_{i_1+k} : (0, s_{i_1} - s_{i_{i'}}) \}$$

this set is considered to be the coordinate of the open-chain joints, so $s_{i_1} < s_{i_2} < \cdots < s_{i_{i'}} = s_k$ and the joint of $j_{i_1}\ (1 \le i_1 \le n)$ in this transformation are an «actuator». Next, we try to create the *L* set using these coordinates based on (14).

$$\forall i' \in \{i_1, i_1+1, \ldots, i_1+k-1\}, \forall k' \in \{1, 2, \ldots, k-i'\} : \{l_{i',i'+k'}\} \cup L \quad \text{where} \quad l_{i',i'+k'} = |j_{i'} - j_{i'+k'}| \quad (14)$$

The optimum measure of *AMmax* was set to its highest value and we assume that all the joints are moving. (15) shows how this value is calculated in the above transformation. The (15) can be evaluated as realation (13).

$$\mathrm{AM}_{max} = \sum_{i'=1}^{k} i' \times \left(k - i'\right) \quad (15)$$

The target point is thought to coincide with j_{i1+k}. Fig. 6 displays this transformation's graphical model. Let us imagine that the S' subset has k members and since based on the method of transformation, the largest subset in the L set consists of $k-1$ members with the Arithmetic Measure of AM_{max}, so in the input of the movement minimization problem, we would have subset L' with k' members, in such a way that $k' \le k-1$ and $AM_{SC_k} \le AM_{max}$. (16) is confirmed as is shown in Appendix 2.

On the opposite hand, it is thought that the input of the LMM problem has the SC_k with $k-1$ members and

it also has AM_{max}. (17) describes how the members of S' subset are created.

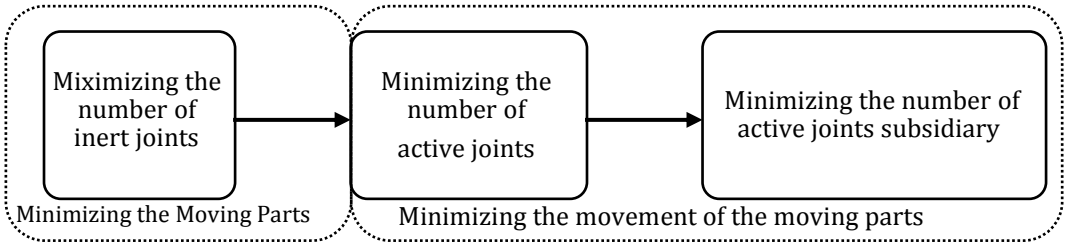


Fig. 4: LMM process.

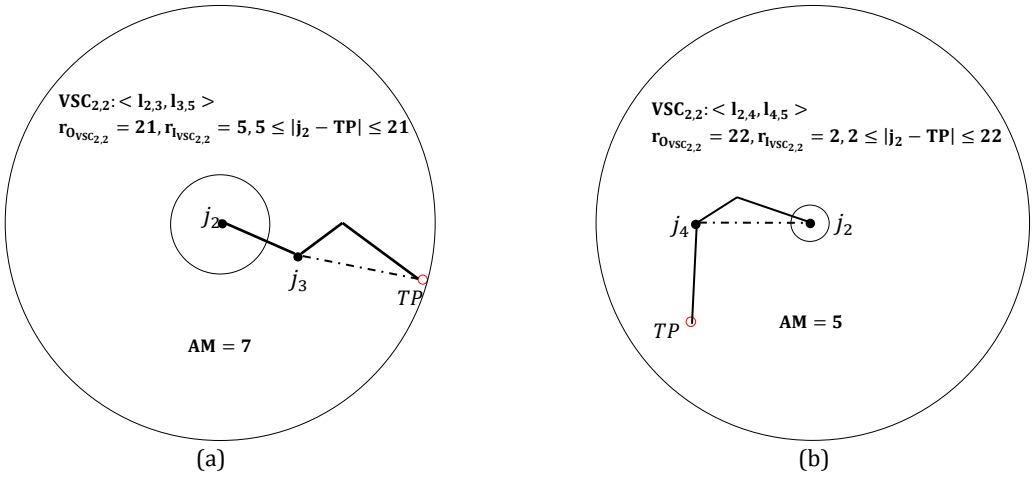


Fig. 5: Two virtual sub-chains with equal passive joints and different *AMs*. The reachability space (a) is a subset of the reachability space (b).

Fig. 7 demonstrates an example of how an input of the sum of the subset problem is transformed into an input of the LMM problem.

To test the validity of the transformation, we examined to make sure that the input of the subset sum problem includes an S' subset combined with k members. Here the sum of the subset sum would be between high and low *iff* the input of the LMM problem has an L' subset with a maximum number of $k-1$ members as well as a maximum Arithmetic Measure of AM_{max}, while (16) is held.

$$r_{I_k} \leq \left| j_{i_1} - j_{i_1+k} \right| \leq r_{O_k} \tag{16}$$

Let us imagine that the S' subset has k members and since based on the method of transformation, the largest subset in the L set consists of $k-1$ members with the Arithmetic Measure of AM_{max}, so in the input of the movement minimization problem, we would have subset L' with k' members, in such a way that $k' \leq k-1$ and $AM_{SC_k} \leq AM_{max}$. (16) is confirmed as is shown in Appendix 2.

On the opposite hand, it is thought that the input of the LMM problem has the SC_k with $k-1$ members and it also has AM_{max}. (17) describes how the members of S' subset are created.

$$\forall i \in \{i_1, i_1+1, \ldots, i_1+k\} : S' \cup \{s_i\} = \begin{cases} s_i = 0, & i = i_1 \\ s_i = \sum_{i'=i_1}^{i-1} l_{i',i'+1}, & i_1+1 \leq i \leq i_1+k \end{cases} \tag{17}$$

Hence, subset S' subset depends on the L' subset, and also an algorithm which can solve the LMM problem is utilized used to calculate the sum of the subset problem.

IV) A greedy heuristic to LLM

The steps described in Movement Minimization of Fig. 2 (the main scheme) are summarized in Fig. 8. It shows in worst case 2^{n-k-2} AM should be calculated.

In algorithm1, the output of the joint is $j_i (1 \leq i \leq n)$ where an «actuator» joint and all the preceding joints are static. Moreover, the SC_k subset is one of the other outputs which is the smallest trailing sub-chain of the arm. In the next step, the objective is to identify the «passive» joints located between the $j_i\,(1 \leq i \leq n)$ joint to the final joint in the arm. In this step, the objective is to remove a number of the members from the $VSC_{k,m}$ and replace them with a lower number of members from the SC_k set. To do this, for every j_k joint located between j_i to j_n,

first, we will assume that it is a «passive» joint. Assuming that $j_k\ (1 \le k \le n)$ is a «passive» joint, $VSC_{k,m}$ is made and the reachability conditions are tested.

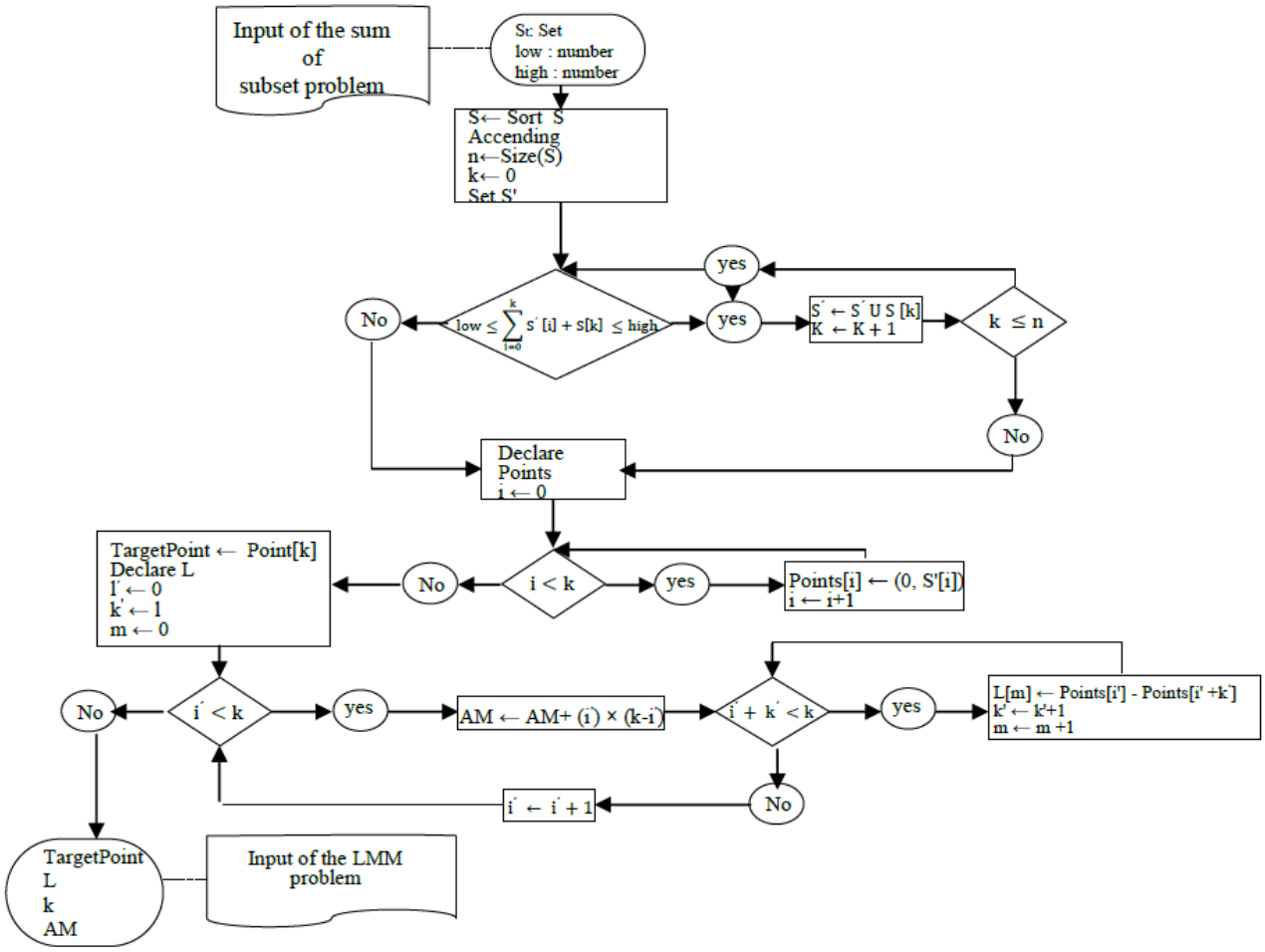


Fig. 6: The graphical model of the above transformation.

C. *Reconfiguration Calculations*

The number of links of the final effective chain (virtual or actual) shows the number of possible solutions. If the chain has only one l_i link, the reconfiguration problem has one solution, which indicates that the target point is located on a circle with the l_i radius and the ith joint is on a circle with the l_i radius and the $Point_i$ center to allow for the placement of the end effector on the target point. Fig. 9 shows the solution to this problem with one link.

The $Update(\theta_{p_1})$ function shows the value of θ_{p_1} at time t_f To interpolate the values of θ_{p_1} in the $[t_s, t_f]$ range it is possible to use various smooth functions. In this research, a smooth function with 3 degrees of freedom was used to generate motion (annex).

If the conditions are not satisfied, is $j_i\ (1 \le i \le n)$ will transform into an «active» joint. Algorithm 2 displays the algorithm used for identifying the «passive» joints and making $VSC_{k,m}$.

In addition, the $Update(Point\ S_{p_1})$ function calculates the Cartesian coordinates of the j_{p_1} joint subset as shown in (18).

$$\forall p' \in \{p_1+1, \ldots, n-1\}:$$

$$x_{p'} = x_p + \sum_{k=p}^{p'-1} l_k \cos \sum_{j=p}^{k} \theta_j \qquad (18)$$

$$y_{p'} = y_p + \sum_{k=p}^{p'-1} l_k \sin \sum_{j=p}^{k} \theta_j$$

If this particular chain has two links, the maximum number of solutions is two. Depending on the position of the target point in each state, there may be one solution to the problem. If the effective chain has more than two links (virtual or actual), there may be infinite solutions to the problem. The number of solutions depends on the status of the circles defined in the following.

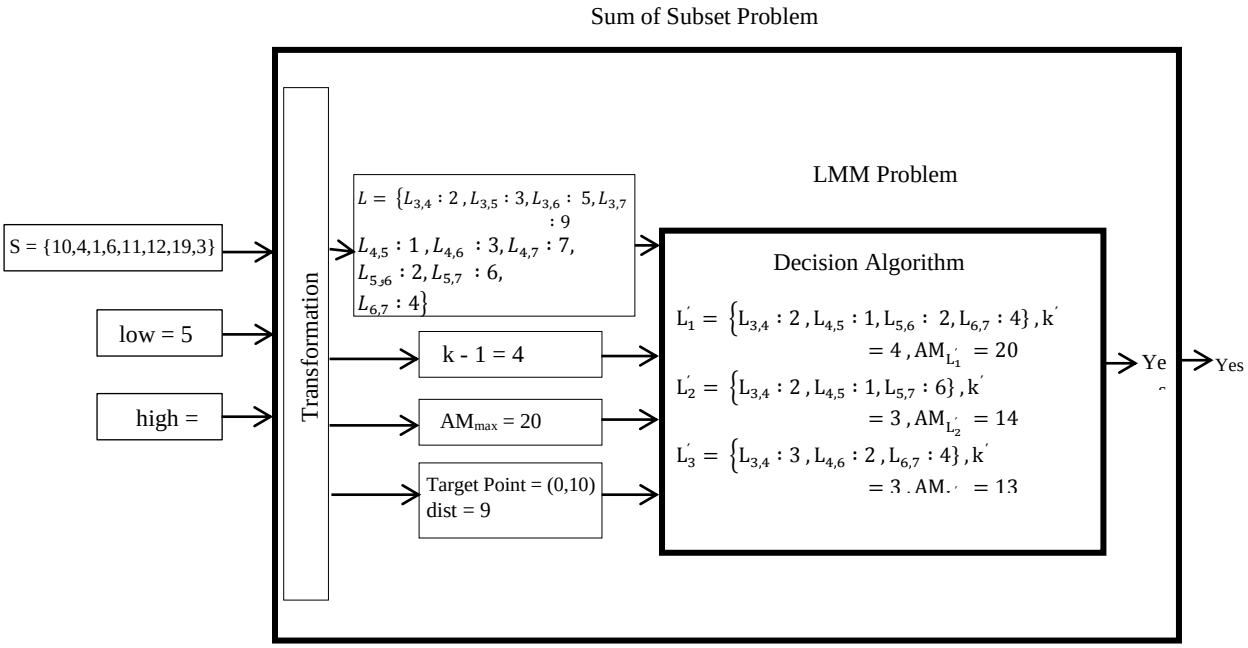


Fig. 7: Transforming a subset sum problem input into the LMM problem.

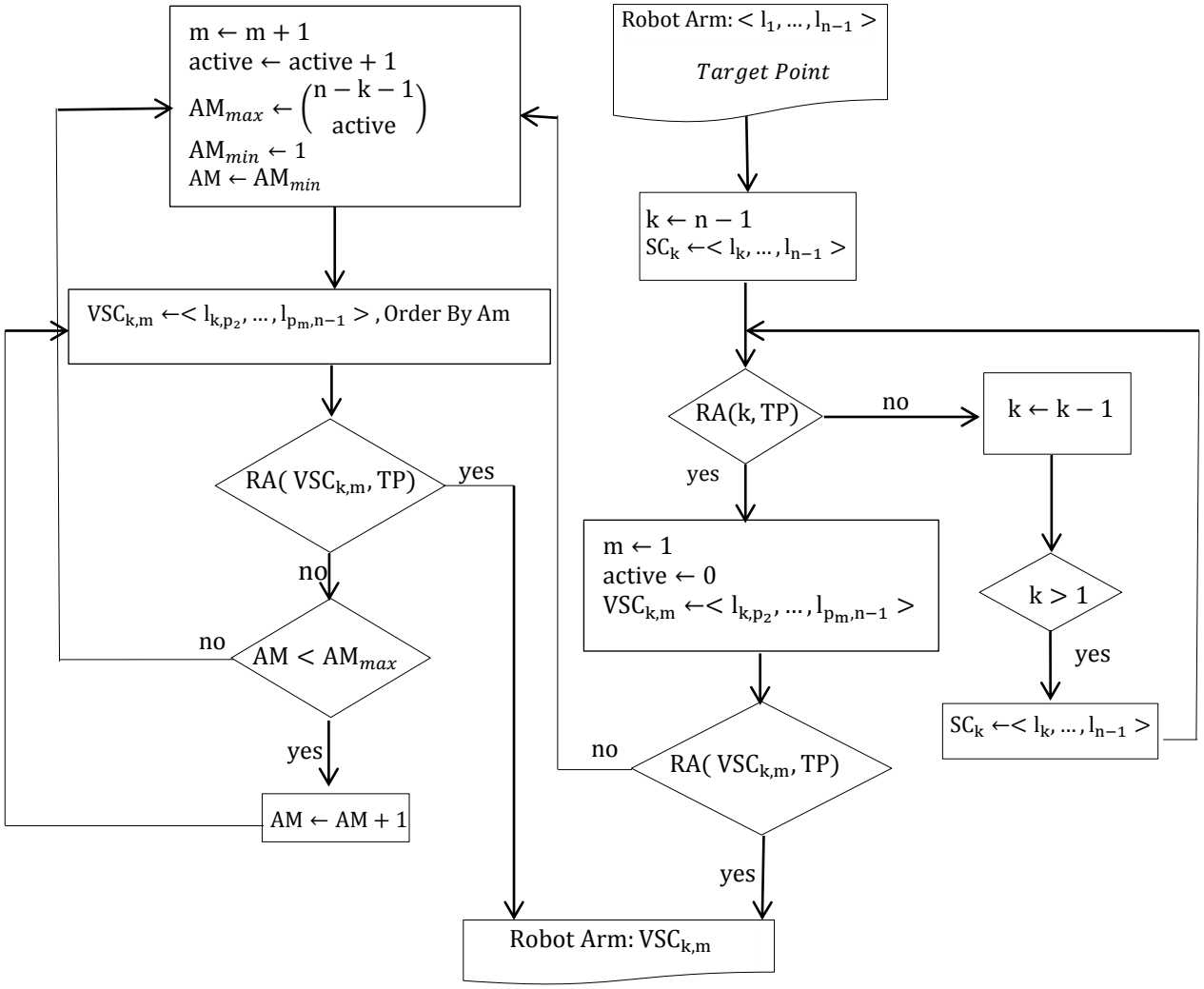


Fig. 8: Flowchart of minimization of the robotic arm moving parts.

In terms of dynamic role, the j_{p_1} joint is the actuator. Therefore, its Cartesian coordinates do not change along the length and its allowable movement occurs on a circle that has the first link as its radius and the moving joint coordinates as its center. The other Cartesian constant of this problem is the target point. Therefore, the second circle has the target point as its center and the second link as its radius. This circle guarantees that the target position of the ending joint is the target point (*TP*). The status of the two circles determines the number of solutions. If the two circles share two points there are two solutions to the problem. The other allowable states for these two circles are the internal tangent and external tangent states which results in one solution to the problem (Fig. 10).

Only these three states are dealt with in this research. In terms of dynamic role, the j_{p_1} joint is the actuator. Therefore, its Cartesian coordinates do not change along the length and its allowable movement occurs on a circle that has the first link as its radius and the moving joint coordinates as its center. The other Cartesian constant of this problem is the target point. Therefore, the second circle has the target point as its center and the second link as its radius. This circle guarantees that the target position of the ending joint is the target point (*TP*). The status of the two circles determines the number of solutions. If the two circles share two points there are two solutions to the problem.

Algorithm: Finding PassiveJoints and making $VSC_{k,m}$

Input: SC_k , i, Target Point //SC_k , i comes from Finding Actuator Joint
Output: $VSC_{k,m}$

$k \leftarrow 0$
$start \leftarrow i$
$passive \leftarrow i + 1$
$end \leftarrow passive + 1$
$VSC_{k,|SC_k|} \leftarrow SC_k$
while $k < n - i - 1$ **do**
  $VSC_{k,m} \leftarrow VSC_{k,m} - \{l_{start,passive}\}$
  $VSC_{k,m} \leftarrow VSC_{k,m} - \{l_{passive,end}\}$
  $VSC_{k,m} \leftarrow VSC_{k,m} \cup \{l_{start,end}\}$
  $m \leftarrow m - 1$
  if $RA(VSC_{k,m}, TP) =$ **false then**
    $VSC_{k,m} \leftarrow VSC_{k,m} - \{l_{start,end}\}$
    $VSC_{k,m} \leftarrow VSC_{k,m} \cup \{l_{start,passive}\}$
    $VSC_{k,m} \leftarrow VSC_{k,m} \cup \{l_{passive,end}\}$
    $m \leftarrow m + 1$
    $start \leftarrow passive$
  end if
  $passive \leftarrow passive + 1$
  $end \leftarrow passive + 1$
  $k \leftarrow k + 1$
end while
return $VSC_{k,m}$
end Algorithm

Algorithm 2: Algorithm for making $VSC_{k,m}$.

Algorithm: Finding 2 Circles

Input: $VSC_{k,m}: \langle l_{p_1,p_2}, \dots, l_{p_m,p_{m+1}} \rangle$, TargetPoint
Output: $Circle_1, Circle_2$

Evaluate $\leftarrow$ **false**
breakjoint $\leftarrow j_{p_2}$
$p_b \leftarrow p_2$
$Circle_1.Center \leftarrow Point_{p_1}$
$Circle_2.Center \leftarrow$ TargetPoint
$Circle_1.Radius \leftarrow \sum_{p'=1}^{b-1} l_{p_{p'},p_{p'+1}}$
$Circle_2.Radius \leftarrow \sum_{p'=b+1}^{m} l_{p_{p'},p_{p'+1}}$
while $b < k$ **and** Evaluate $=$ **false do**
  Evaluate $\leftarrow$ Evaluate 2 Circles
  if Evaluate $=$ **true then**
    $\forall p' \in \{p_1, p_2, \dots, p_m\}, p' \neq p_b$
    $\theta_{p'} \leftarrow \pi$
  Type $\leftarrow$ Flattening
  end if
  $b \leftarrow b + 1$
end while
if Evaluate $=$ **false then**
  $l_b \leftarrow \max \langle l_{p_1,p_2}, \dots, l_{p_m,p_{m+1}} \rangle$
  $\forall p' \in \{p_1, p_2, \dots, p_m\}, p' \neq p_b$
  $\theta_{p'} \leftarrow \pi$
  $\theta_{p_b} \leftarrow 0$
  $Circle_1.Radius \leftarrow \sum_{p'=1}^{b-2} l_{p_{p'},p_{p'+1}}$
  $Circle_2.Radius \leftarrow \left| l_b - \sum_{p'=b+1}^{m} l_{p_{p'},p_{p'+1}} \right|$
  Type $\leftarrow$ Squeeze
end if
return $Circle_1, Circle_2$
end Algorithm

Algorithm 3: Finding algorithm of two circles.

Solution 1 Link : (l_{p_1,p_2})

Step 1: $Update\ (\theta_{p_1})$

$$\theta_{p_1} = \begin{cases} \measuredangle(Point_{p_1-1}, Point_{p_1}, Target\ Point), p_1 : [2, n] \\ \measuredangle(H_{axis}, \overrightarrow{Point_{p_1}, Target\ Point}), p_1 = 1 \end{cases}$$

Step 2: $Update(Point\ S_{p_1})$

Fig. 9: Solution to the problem of reconfiguration a robot arm with one link.

In this research, to calculate the robot arm reconfiguration in a problem with more than two links, the problem is modeled as a dual-link problem. In other words, a joint is selected from the joints at the junction of the chain links, and since the links of the effective chain are replaced with virtual links when needed, the passive joints are excluded. This joint is called the “break joint”.

The relative angle of the existing joints is placed

between the links before and after the break joint at the final position (π *or 0*), and the problem is mapped into a problem with two links. The links before the break joint form one link and the links after the break joint form another link.

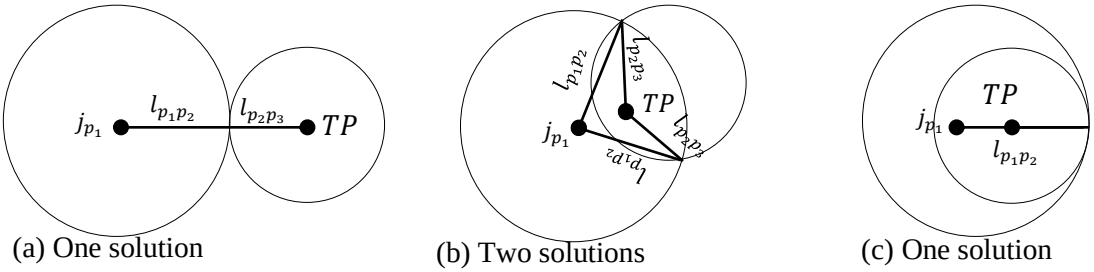


Fig. 10: Reachability with two links.

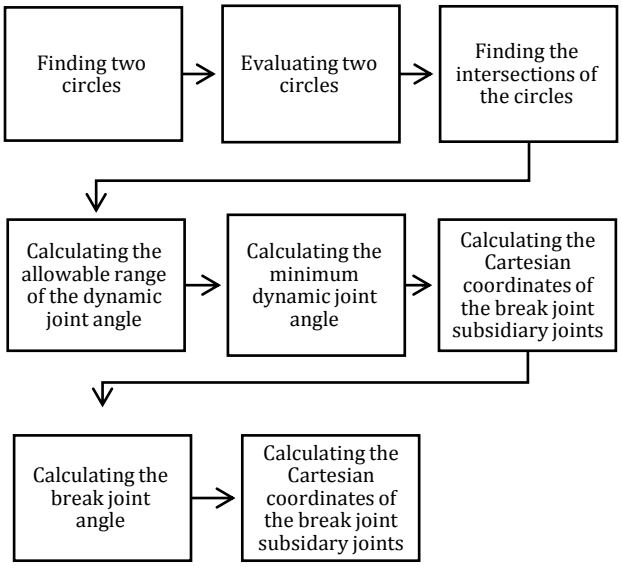


Fig. 11: Schema of the reconfiguration calculation model of a problem with more than two links.

Next, the intersections of the two circles, which are in one of the allowable states concerning each other, are found (Algorithm 4). These two circles can be placed in three states in to one another: having two intersections (which requires the formation of the triangle by two radii and the distance between the target point and the dynamic joint); having an internal tangent (which requires the equal differences between the two radii s based on the distance between the target point and the dynamic joint); and having an external tangent (which requires the equal sums of two radiuses based on the distance between the target point and the dynamic joint).

In problems with more two links, the area between the intersections on the circle is the solution set for the break joint. The number of the solution sets depends on the number of possible permutations of the break joint.

Fig. 11 illustrates the overall schema of the reconfiguration calculation model for a problem with more than two (virtual or actual) links.

To find the circles, if the problem consists of more than two links the radiuses of the circles must be known. Algorithm 3 illustrates the algorithm of finding two circles.

Evidently, the two circles have one intersection if they are externally tangent or internally tangent.

Next, the allowable range of the dynamic joint angle is calculated and the minimum value is assumed for the dynamic joint. If the j_{p_1} joint is not the first robot arm joint, the angle range varies from the joint before the j_{p_1} joint and j_{p_1} and one of the intersections of the two circles.

The end of the angle range is obtained by measuring the angle between the previous joint, the dynamic joint, and the subsequent intersection of the two circles. If the dynamic joint is the first joint, the angle between the horizon axis and the vector between $Point_{p_1}$ and the intersection of the two circles is the angle of the joint. Fig. 12 depicts the performance of the $Update(\theta_{p_1})$ function. In the following, using (14), the Cartesian coordinates of the subsidiary joints of the j_{p_1} joint are updated using the $Update(Point\ S_{p_1})$ function. Afterward, the break joint angle is calculated depending on the status of the two circles. If the two circles have two intersection points, the angle of concern is calculated by measuring the angle between the dynamic joint and the Cartesian coordinates of the broken joint and the target point. However, if the two circles are in the external tangent state, there is no need for calculation because π is the break joint angle. Finally, if the two circles are internally tangent, this angle is 0 (Fig. 13). In the end, the Cartesian coordinates of the subsidiary joints of the break joint are calculated using the $Update(Point\ S_{p_1})$ function and (14).

Update(θ_{p_1})

Step1: Find all answers

$$\theta_{p_{1_{f_1}}} = \begin{cases} \measuredangle(\text{Point}_{p_1-1}, \text{Point}_{p_1}, \text{IntersectionPoint}_1), p_1 \in \{2, \dots, n-1\} \\ \measuredangle(H_{axis}, \overrightarrow{\text{Point}_{p_1}, \text{IntersectionPoint}_1}), p_1 = 1 \end{cases}$$

$$\theta_{p_{1_{f_2}}} = \begin{cases} \measuredangle(\text{Point}_{p_1-1}, \text{Point}_{p_1}, \text{IntersectionPoint}_2), p_1 \in \{2, \dots, n-1\} \\ \measuredangle(H_{axis}, \overrightarrow{\text{Point}_{p_1}, \text{IntersectionPoint}_2}), p_1 = 1 \end{cases}$$

$$\theta_{p_{1_{range}}} = \begin{cases} \left[\theta_{p_{1_{f_1}}}, \theta_{p_{1_{f_2}}}\right], m > 3 \\ \left\{\theta_{p_{1_{f_1}}}, \theta_{p_{1_{f_2}}}\right\}, m \leq 3 \end{cases}$$

Step2: Find minimize θ_{p_1}

$$\theta_{p_1} = \begin{cases} \theta_{p_{1_{f_1}}}, \theta_{p_{1_{f_1}}} - \theta_{p_1}(t_s) < \theta_{p_{1_{f_2}}} - \theta_{p_1}(t_s) \\ \theta_{p_{1_{f_2}}}, \text{Otherwise} \end{cases}$$

Fig. 12: Function update (θp1) performance (Calculation of the allowable value and the minimum dynamic joint angle).

Algorithm: Evaluate 2 Circle

Input: $\text{Circle}_1, \text{Circle}_2$

Output: Evaluate

//Status:{Triangle,ExternalTangent,InternalTangent}

1: $r_1 \leftarrow \text{Circle}_1.\text{Radius}$
2: $r_2 \leftarrow \text{Circle}_2.\text{Radius}$
3: $r_3 \leftarrow |\text{Circle}_1.\text{Center} - \text{Circle}_2.\text{Center}|$
4: **if** $r_1 + r_2 > r_3$ **or** $r_1 + r_3 > r_1$ **or** $r_1 + r_3 > r_2$ **then**
5: Status ← Triangle
6: **else if** $r_1 + r_2 = r_3$ **then**
7: Status ← ExternalTangent
8: **else if** $|r_1 - r_2| = r_3$ **then**
9: Status ← InternalTangent
10: **end if**
11: **if** status = Triangle **or** ExternalTan **or** InternalTan **then** Evaluate ← **true**
12: **return** Evaluate
13: **end Algorithm**

Algorithm 4: Evaluation algorithm of two circles.

Update(θ_{p_b})

$$\theta_{p_b} = \begin{cases} \measuredangle(\text{Point}_{p_1}, \text{Point}_{p_b}, \text{TargetPoint}), \text{Status} = \text{Triangle} \\ \pi, \quad \text{Status} = \text{ExternalTangent} \\ 0, \quad \text{Status} = \text{InternalTangent} \end{cases}$$

Fig. 13: Calculation of the break joint angle.

In Fig. 14 and Fig. 15 two results of using the reconfiguration algorithm with two different target points are shown.

D. Time and space complexity

The input for Algorithm 1 is the output of Algorithm 2, and its output is the subset $VSC_{k,m}$. Algorithm 1 in the proposed approach possesses a time and space complexity of $O(n)$. In Algorithm 2, k joints must be examined in terms of «active» or «passive» state, and this examination 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 carried out at the time $O(n^2)$. Algorithm 3 possesses a time and space complexity of $O(n)$ and Algorithm 4 takes $O(1)$. Thus, the time complexity of the proposed approach is $O(n^2)$ and the space complexity is $O(n)$.

BreakJoint: j_4
$\text{Circle}_1 : (\text{Center}_1 : j_1, \text{Radius}_1 : l_1 + l_2 + l_3)$
$\text{Circle}_2 : (\text{Center}_2 : \text{TP}, \text{Radius}_2 : l_4)$

BreakJoint: j_2
$\text{Circle}_1 : (\text{Center}_1 : j_1, \text{Radius}_1 : l_1)$
$\text{Circle}_2 : (\text{Center}_2 : \text{TP}, \text{Radius}_2 : l_2 + l_3 + l_4)$

BreakJoint: j_3
$\text{Circle}_1 : (\text{Center}_1 : j_1, \text{Radius}_1 : l_1 + l_2)$
$\text{Circle}_2 : (\text{Center}_2 : \text{TP}, \text{Radius}_2 : l_3 + l_4)$

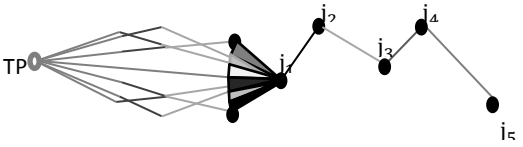


Fig. 14: Solutions to the robotic arm reconfiguration problem with more than two links and one passive joint.

$\text{Circle}_1 : (\text{Center}_1 : j_1, \text{Radius}_1 : l_1)$
$\text{Circle}_2 : (\text{Center}_2 : \text{TP}, \text{Radius}_2 : l_{2,3} + l_4)$
Status (j_1 — TP| Radius$_1$ Radius$_2$) = Triangle

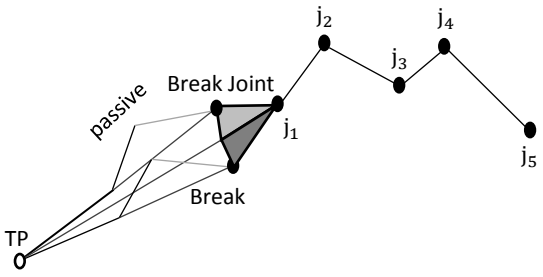


Fig. 15: Possible solutions to a problem with four links.

Simulation Results and Compare with Related Works

The simulation programming language is C#. Table 2 lists the parameters of the proposed method. Table 3 lists the results of simulating the reconfiguration calculation algorithm in terms of the arm status after the application of the algorithm.

Table 2: The parameters of the proposed method

Parameter	Data Type	Input	Output
Number of Links	Integer	Yes	
Target Point	Integer	Yes	
Length of the Links	Real	Yes	
Reachability	Boolean		Yes
VSCk, m	Array		Yes
Properties of Circle 1, Circle 2 (Radius, Center)	Radius: Real Center: Point		Yes

Table 3: Results of simulation of the robot arm reconfiguration algorithm

Status	Triangle	ExternalTan	InternalTan
Frequency percentage	91.24	2.63	6.13

Table 4: Results of Simulating the Robot Arm Moving Parts Minimization Algorithm

Percentage of inert joints		Ratio (%) of the passive joints to the static joints		Ratio (%) of the passive joints to total joints	Ratio (%) of AM similarity of optimal results with greedy results
No static joint	5.03	Less than half	87.04	4.38	94.5
		More than half	12.96	0.65	
Less than half	55.9	Less than half	74.64	41.73	87.2
		More than half	25.36	14.17	
More than half	39.07	Less than half	57.07	22.29	80.8
		More than half	42.93	16.77	

The results of the simulation of the proposed method are presented in this section. These simulations are the result of applying algorithms and their means on 5000 random arms, each with 5000 random target points. The number of the links (between 2 and 13 links), the length, and the initial configuration of each robot arm are randomly determined. Table 4 lists the results of minimizing the robot arm moving parts. The above results suggest that using this method more than half of the joints are often static and are thus omitted from the main chain. Moreover, the simulation results imply that there is no need to change the relative angle of more than half the joints in the resulting chain. For the proposed greedy approach, we will attempt to compare the obtained results by considering them alongside optimal results. In more than 80% of the cases, the results of the greedy approach with optimal results are identical.

Most linkage issues are hardness problems in computer science. However, studies have shown there is no problem with the aim of minimizing the moving parts of the robot arm and minimizing the movement of these moving parts. Table 5 compares some of the related works in this area.

Conclusions

In this article, the problem of optimizing the robot arm components for positioning the end effector of the robot arm at the target point is presented as well as minimizing the movement of these robot components. The problem was first formalized, and then the required criteria and parameters were designed. One of these parameters was AM. The mentioned criterion was introduced to quantify and compare the types of robot arm configurations. It was then shown to be an NP-Hard problem. For this reason, a greedy heuristic was presented that the time complexity of the proposed method was $O(n^2)$ and its memory usage was $O(n)$. In section *4*, the simulated results demonstrated that the presented heuristic difference with the optimal exponential solution was above 80% in most cases.

The results indicate that the discussed model successfully reduced the moving parts of the robot arm. Moreover, the results show that the proposed approach fulfills the goal of minimization of the 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 the linkages. The algorithm presented in the section of obtaining the final configuration solved the problem by mapping the entire robot arm to one arm with one or two links. Future work could include a dynamic programming algorithm to solve this problem. It is also possible to extend the solution for other types of robot arm. Also, it would be developing a method for configuration of simple robot arms with no link intersection.

Author Contributions

A. Nourollah and N. Behzadpour have proposed the issue, shown the complexity of the issue, designed the experiments, carried out the data analysis, interpreted the results and wrote the manuscript.

Conflict of Interest

A. Nourollah and N. Behzadpour declare that there is no conflict of interests regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancy have been completely observed by the authors.

Abbreviations

LMM	Linkage Movement Minimization
AM	Arithmetic Measure
θ_i *(1,2,3 ... k)*	Regression coefficients
CCS	Cuenca del Cañón del Sumidero
d	Durbin-Watson test
Eq.	Equation
F	Fisher test
F_t	Predicted value
H_0	Null hypothesis
k	Number of explanatory variables included in the model

Appendix 1

(19) shows a cubic polynomial that reflects the angular position of the ith joint at time t if velocity is zero at times t_s and t_f.

$$\theta_i(t) = \theta_i(t_s) + \left(\frac{3\left(\theta_i(t_f)-\theta_i(t_s)\right)}{{t_f}^2}\right)t^2 + \left(\frac{-2\left(\theta_i(t_f)-\theta_i(t_s)\right)}{{t_f}^3}\right)t^3 \quad (19)$$

The Cartesian coordinates of the joints in the course of movement can be calculated by adding the time parameter to (19). Fig. 16 illustrates the result of the calculation of the arm movement from the initial position to the final configuration at ten moments.

Table 5: Comparing some of the related works with the proposed method

Reference	Optimal Parameter	Hardness of problem
[7] (Shin et al.)	Motion Cost	Polynomial
[10] (Chettibi et al.)	Limiting the angular positions of the joints and the velocity and acceleration of the joints	Polynomial
[11] (Choi et al.)	omit the intersection between the arm links during motion	NP-Complete
[12] (Zhang et al.)	limitations on the kinematic variable parameters of the obstacles form the basis	NP-Complete
[13] (Ding et al.)	Minimization of the motion time and elimination of the obstacles	
[15] (Nourollah et al.)	low-cost for folding	NP-Complete
Proposed Method	Linkage movement minimization	NP-Hard

Appendix 2

(16) is confirmed by utilizing the triangle inequality hypothesis, lemma 1, and (11).

Lemma 1: for each non-negative numbers of a, b, and c where, $a \le b \le c$, $2a - c \le b$ holds true.

Proof: If $c = a + l$, we have $2a - a - l \le b$. Thus, $a - l \le b$ and Lemma 1 is proven. ■

With the use of the triangle inequality theorem and the definition of $r_{O_L'}$ in (11), the confirmation of $\left|j_{i_1} - j_{i_1+k}\right| \le r_{O_L'}$ is proven. Assuming that the members of the S' subset have an ascending order, then $\left|j_{i_1} - j_{i_1+k}\right|$ would be the largest member of the L set, and therefore, for each L' subset with its biggest member $l_{M_L'}$, the $\left|j_{i_1} - j_{i_1+k}\right| \ge l_{M_L'}$ 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.

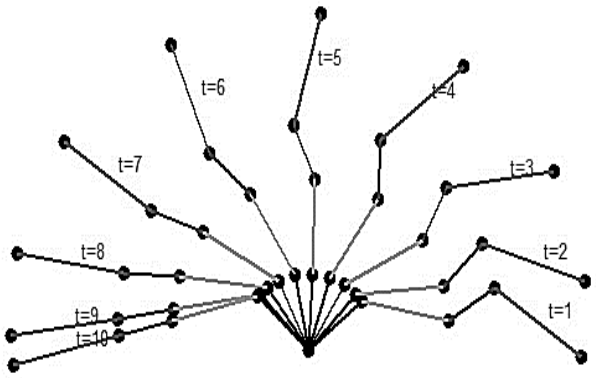


Fig. 16: Movement of the chain from the initial configuration to the final configuration.

References

[1] "All on robots," [Accessed 02 08 2016].

[2] B. Wanlund, "Robots and automation – sage business researcher,".

[3] J. J. Craig, "Path Generation," in Introduction To Robotics, First, Ed., Boston, Addison-Wesley Longman Publishing Co., 257-269, 1989.

[4] L. Guibas, "Motion Modeling," in Discrete and Computational Geometry, 2nd ed., Chapman & Hall/CRC,: 1114-1132, 2004.

[5] R. Connelly, J. O'Rourke, "Linkages," in Geometric Folding Algorithms, Cambridge, Cambridge University Press: 9-11, 2007.

[6] S. L. Devadoss, J. O'Rourke, "Configuration Spaces," in Discrete and Computational Geometry, Princeton, University Press: 215-219, 2011.

[7] K. Shin, N. McKay, "A dynamic programming approach to trajectory planning of robotic," IEEE Transactions on Automatic Control, 31(6): 491-500, 1986.

[8] Wu, Z. Shi, Y. Li, M. Wu, Y. Guan, J. Zhang, H. Wei, "Formal kinematic analysis of a general 6r manipulator using the screw theory," Mathematical Problems in Engineering, 2015(2): 1-7, 2015.

[9] Y. Kouskoulas, A. Platzer, P. Kazanzides, "Formal Methods for Robotic System Control Software," Johns Hopkins APL Technical Digest, 32(2): 490-498, 2013.

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

[11] J. Choi, E. Amir, "Factor-guided motion planning for a robot arm," presented at the IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 2007.

[12] Y. Zhang, X. Lv, Z. Li, Z. Yang, K. Chen, "Repetitive motion planning of pa10 robot am subject to joint physical limits and using lvi-based primal–dual neural network," Mechatronics, 18(9): 475-485, 2008.

[13] H. Ding, M. Zhou, O. Stursberg, "Optimal motion planning for robotic manipulators with dynamic obstacles using mixed-integer linear programming," presented at the Mediterranean Conference on Control & Automation, Makedonia Palace, Thessaloniki, Greece, 2009.

[14] F.L. Traversa, C. Ramella, F. Bonani, M. Ventra, "Memcomputing np-complete problems in polynomial time using polynomial resources and collective states," Science Advances, 1(6): 1-9, 2015.

[15] A. Nourollah, M. R. Razzazi, "Minimum cost open chain reconfiguration," Discrete Applied Mathematics, 159: 1418-1424, 2011.

[16] A. Nourollah, M. R. Razzazi, "A linear time approximation algorithm for ruler folding problem," Journal of Universal Computer Science, 14(4): 566-574, 2008.

[17] A. Nourollah, M. R. Razzazi, "Near-optimum folding for a reconfigurable," Advanced Robotics, 23(1-2): 239-256, 2009.

[18] R. Connelly, E. Demaine, "Geometry and topology of polygonal linkages," Discrete and Computational Geometry: 213-235, 2004.

[19] J. Hopcroft, J. O'Rourke, S. Whitesides, "On the movement of robot arms in 2-dimensional bounded regions," SIAM J. Computer, 14(2): 315-333, 1985.

[20] E.D. Demain, J. O'Rourke, "Geometric and folding algorithms: Linkages, Origami, Polyhedra," Cambridge, University Press: 9-11, 29-31,59-67, 2007.

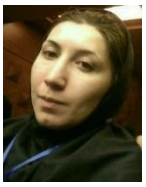
[21] A. Lieutier, J. Rameau, "Mechanical linkage design and NP-hardness," Mechanism and Machine Theory, 82: 97-114, 2014.

[22] G. Panina, D. Siersma, "Motion planning and control of a planar polygonal linkage," Journal of Symbolic Computation, 88: 5 -20, 2018.

[23] C. Haasea, S. Kiefer, "The complexity of the kth largest subset problem and related problems," Information Processing Letters, 116(2): 111-115, 2016.

Biographies

Ali Nourollah received the M.Sc. degree in Computer Engineering from Sharif University of Technology at Tehran (1998) and the Ph.D. degree in Computer Engineering from the Amirkabir University of Technology at Tehran (2008). Currently he is an Assistant Professor in the Faculty of Computer Engineering at Shahid Rajaee Teacher Training University at Tehran. His primary areas of research are Computational Geometry, Graph Algorithms, and Design & Analysis of Algorithms.



Nooshin Behzadpour received the M.Sc. degree in Computer Engineering from Shahid Rajaee Teacher Training University at Tehran (2016). Her primary areas of research are Computational Geometry, Robot arm Movement Algorithms, and Formal Modeling.

How to cite this paper:
A. Nourollah, N. Behzadpour, "Robot arm reconfiguration to minimization moving parts," Journal of Electrical and Computer Engineering Innovations, 6(2): 235-249, 2018.
DOI: 10.22061/JECEI.2019.5231.204
URL: http://jecei.sru.ac.ir/article_1116.html