Mechanism and Machine Theory 82 (2014) 97–114

Contents lists available at ScienceDirect

Mechanism and Machine Theory

journal homepage: www.elsevier.com/locate/mechmt

Mechanical linkage design and NP-hardness

André Lieutier [a], Jean-François Rameau [a,b,*]

[a] Dassault Systèmes, 10 rue Marcel Dassault, 78140 Vélizy-Villacoublay, France
[b] LISMMA, 3 rue Fernand Hainaut, 93407 Saint Ouen, France

ARTICLE INFO

Article history:
Received 27 November 2013
Received in revised form 11 July 2014
Accepted 4 August 2014
Available online 27 August 2014

Keywords:
Dimensioning
Assembly
Mobility
Algorithmic complexity
NP-hard

ABSTRACT

We prove the NP-hardness of two fundamental problems in mechanism design. Given a set of rigid parts and prescribed joints, the first problem consists in deciding whether there exists an assembly of these parts matching the link constraints. Given a set of rigid parts and prescribed joints with parameterized dimensions, the second problem consists in deciding if there exists some dimensioning making the assembly mobile. The proof of hardness makes use of a reduction from Partition Problem for the first problem and from 3SAT for the second one.

1. Introduction

Motivated by its obvious practical significance, the design of mechanisms has given rise to many studies for over 150 years [1]. With the advent of Computer Aided Design software as universal tools for the design of parts and assemblies in the industry, algorithms and methodologies have been proposed to support and partially automate the design of mechanisms: [2–5] to cite a few of them.

In this work, the theoretical computer science point of view is adopted to investigate the complexity analysis of two problems naturally arising in the automation of mechanism design. Akin to this point of view are a few related works in computational geometry such as [6,7]. However, such approach is not common in the mechanical engineering community, which is more interested in practical, and therefore constructive, results.

Our results, Theorems 1 and 2, are complexity lower bounds and therefore negative results. But we believe that they improve our knowledge of what can be reasonably expected from algorithms in computer aided mechanism design, which may prepare the landscape for constructive results.

We call the first problem *Realization of Prescribed Assembly Problem*, or RPAP: given a set of rigid parts and joints with fixed geometry, does a configuration (i.e. a rigid transformation for each part) that matches the prescribed link constraints exist? A practical instance of this problem would be the following game. You are given a set of labeled bars with some labeled cylindrical holes and cylindrical protrusions matching the holes' diameters. Moreover, you are given a kind of minimal assembly plan: the list of pairs of part protrusion/part hole which must be plugged together. We prove that, even with this assembly plan, the game may be hard. Indeed, any instance of the combinatorial Partition Problem, well known to be NP-complete, can be reduced to an instance or RPAP.

Our second problem, the *Parameterized Mechanical Linkage Mobility Problem*, or PMLMP, addresses the mobility of a parameterized mechanism. Here parameterized mechanism means, as it is common in Computer Aided Design software, a set of parts and joints

* Corresponding author at: Dassault Systèmes, 10 rue Marcel Dassault, 78140 Vélizy-Villacoublay, France. Tel.: +33 1 61 62 40 79; fax: +33 1 70 73 43 88.
E-mail address: jfr@3ds.com (J.-F. Rameau).

http://dx.doi.org/10.1016/j.mechmachtheory.2014.08.002
0094-114X/© 2014 Elsevier Ltd. All rights reserved.

whose geometry is parameterized by a set of numerical variables, such as for example distances between joint anchor points. An instance of a parameterized mechanical linkage consists of the particular geometry defined by the assignment of a value to each parameter. The parameterized mechanical linkage mobility problem consists in deciding if there exists an instance of the parameterized mechanism which is *mobile*, a mechanism being said mobile if the link constraints preserve at least one degree of freedom. We prove that the 3SAT combinatorial problem can be reduced to the parameterized mechanical linkage mobility problem which is therefore NP-hard.

We would like to warn the reader about the specificity of these results. The first difficulty is the introduction of algorithmic complexity in the world of mechanical design. Algorithmic complexity is pure theory and deals with how a problem can be difficult to solve rather than how to solve it. The second difficulty is the structure of an algorithmic complexity theorem. When reading for the first time the proof of such a theorem, it sounds like an example or a particular case is detailed, which is always frustrating. Consequently, we would give the following advice to the reader who is not familiar with algorithmic complexity. The basic reading level is to understand the mechanical design problems under study (RPAP and PMLMP), and to admit that they are difficult to solve. The advanced reading level is to figure out why RPAP and PMLMP are difficult to solve by understanding the particular assembly and particular mechanism of the proofs and their articulation with the combinatorial problems. Anyway, the value for mechanical designers is to keep in mind that an actual complexity is hidden behind their usual subjects of investigation.

2. Setting up a complexity theorem

The structure of a complexity theorem may look unusual to those who are not familiar with the field of theoretical computer science, and the goal of this section is to clarify this point. The input is a problem, noted *T*, arising in a technical domain. The technical domain of our purpose is mechanical design, but it can be of any kind: computer science, geometry, topology, graph theory, etc. The open question is the complexity of problem *T*, more precisely, the complexity of the most efficient algorithm that can exist to solve problem *T*: can it be a polynomial, exponential algorithm or whatever. The art of complexity theory is to provide an answer to this question. First of all, the specialists of the technical domain conjecture a target complexity of problem *T*, NP-hard for our purpose. Then, proving the conjecture "problem *T* is NP-hard" includes two steps. The first step is to select a combinatorial problem, noted *C*, that is known to be NP-hard. The second step, named the "reduction from *C*", is to design one particular case of problem *T*, noted *T′*, that encodes problem *C*, meaning that problem *T′* has a solution if and only if problem *C* has a solution. In addition, the proof includes an investigation about the number of computing steps for the reduction, which must be a polynomial function of the data. This completes the complexity theorem that states that problem *T* is NP-hard, as well as problem *C*.

The key point is that it is enough to select *one particular combinatorial problem C* and to design *one specialization T′ of technical problem T* that encodes the combinatorial problem. Thus, if someone comes with an algorithm for solving problem *T*, then, this algorithm must solve the problem *T′* as well, which is equivalent to solving problem *C*.

Consequently, the next two theorems are organized by describing, each time, the technical problem *T*, the combinatorial problem *C*, and the specialization *T′* of technical problem *T*.

3. Parameterized mechanical linkage: formal definition

This section provides a formal definition of a mechanical linkage, the center object of our purpose. Despite that this concept is familiar, it deserves a rigorous understanding. By definition, a mechanical linkage *M* is a six-tuple

$$M = (J, B, f, L, D, C)$$

the features of which are defined as follows. The set

$$J = \{j_1, \cdots, j_m\}$$

is a finite set. Elements of *J* are called *joints*. The set

$$B = \{b_1, \cdots, b_n\}$$

is another finite set. Elements of *B* are called *bodies*. By definition *B* and *J* are disjoint sets. The mapping

$$f : J \to B$$

associates a unique body to each joint. In other words, $f(j) = b$ means that joint *j* belongs to body *b*. For each body *b*, one particular element of $f^{-1}(b)$ is called the *reference joint* and is noted $j_0(b)$.

Now consider the set of couples of joints defined by

$$I = \bigcup_{b \in B} \{j_0(b)\} \times \left(f^{-1}(b) - \{j_0(b)\}\right).$$