Journal of Manufacturing Systems 36 (2015) 128–146

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

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

Technical Paper

# Assembly sequence planning of rigid and flexible parts

Somayé Ghandi, Ellips Masehian*

*Faculty of Engineering, Tarbiat Modares University, Tehran 14115-143, Iran*

ARTICLE INFO

*Article history:*
Received 20 November 2014
Received in revised form 29 April 2015
Accepted 19 May 2015
Available online 26 June 2015

*Keywords:*
Assembly sequence planning (ASP)
Flexible assembly parts
Assembly stress matrix (ASM)
Scatter search
Parameter tuning
TOPSIS method

ABSTRACT

Assembly sequence planning (ASP) is the process of computing a sequence of assembly motions for constituent parts of an assembled final product. ASP is proven to be NP-hard and thus its effective and efficient solution has been a challenge for the researchers in the field. Despite the fact that most assembled products like ships, aircrafts and automobiles are composed of rigid and flexible parts, no work exists for assembly/disassembly sequence planning of flexible parts. This paper lays out a theoretical ground for modeling the deformability of flexible assembly parts by introducing the concept of Assembly stress matrix (ASM) to describe interference relations between parts of an assembly and the amount of compressive stress needed for assembling flexible parts. Also, the Scatter Search (SS) optimization algorithm is customized for this problem to produce high-quality solutions by simultaneously minimizing both the maximum applied stress exerted for performing assembly operations and the number of assembly direction changes. The parameters of this algorithm are tuned by a TOPSIS-Taguchi based tuning method. A number of ASP problems with rigid and flexible parts were solved by the presented SS and other algorithms like Genetic and Memetic algorithms, Simulated Annealing, Breakout Local Search, Iterated Local Search, and Multistart Local Search, and the results and their in-depth statistical analyses showed that the SS outperformed other algorithms by producing the best-known or optimal solutions with highest success rates.

## 1. Introduction

The *Assembly Sequence Planning* (ASP) problem concerns with finding a sequence of collision-free operations $o_1, \ldots, o_n$ that bring the assembly parts $p_1, \ldots, p_n$ together, having given the geometry of the final product $A$ and the positions of parts in the final product. ASP plays a key role in the whole lifecycle of a product and has a great impact on variation propagation, production quality, and efficiency of the assembly process [29]. Assembly of manufactured goods constitutes over half of the total production time in manufacturing industry, and about one-third to half of labor costs [32]. The complexity of finding an optimal sequence increases linearly with the size of the space of all potential assembly sequences in the case of exhaustive search. In fact, the ASP problem is classified as an NP-hard problem and thus cannot be solved in polynomial time for large instances as the total number of potential sequences is given by the permutations of parts, $n!$ [22].

In recent years, intensive research efforts have been put in developing intelligent methods to solve the ASP problem, as they have been able to improve the efficiency of finding optimal assembly sequences while avoiding the combinatorial explosion problem as the number of assembly components increases. After Bonneville et al. [4] implemented Genetic Algorithm (GA) in ASP for the first time in 1995, many soft computing/metaheuristic algorithms were developed to solve the problem, such as Simulated Annealing (SA) [25,17], Artificial Immune System (AIS) [5], GA, Ant Colony Optimization (ACO) [37,15], Artificial Neural Networks (ANN) [6,16,31,7], Memetic Algorithm (MA) [11], enhanced Harmony Search (HS) [38], Imperialist Competitive Algorithm (ICA) [43], Particle Swarm Optimization (PSO) [42,36,22,39], Firefly Algorithm (FA) [41], Psychoclonal algorithm [34], and Bacterial Chemotaxis algorithm [44]. An inclusive survey on the ASP and its methods is presented in [18], which overviews the elements of sequence planning such as finding a feasible sequence, determining an optimal sequence according to one or more operational criteria, representing the space of feasible assembly sequences in different ways, applying search and optimization algorithms, and satisfying precedence constraints existing between subassemblies. Also, a survey of works that have applied particularly soft computing approaches in ASP has been presented in [27], covering the years 2001 to 2011.

In all of the abovementioned works, a main assumption is that all parts of the assembly are rigid (not deformable) and their shapes do not change during the assembly process. This assumption is a

---

* Corresponding author. Tel.: +98 2182884939.
*E-mail address:* masehian@modares.ac.ir (E. Masehian).

http://dx.doi.org/10.1016/j.jmsy.2015.05.002

simplifying one since flexible parts introduce additional degrees of complexity to the ASP problem. In fact, most real-world assembled products like ships, aircrafts, and automobiles are composed of rigid and flexible parts, and so automatic generation of assembly sequence plans for such products requires the deformability of flexible parts to be taken into account.

The solution to the ASP problem of a product with rigid and flexible parts depends heavily on the model that is used to simulate the deformation behavior of flexible parts. In general, there are two approaches in modeling the deformation of flexible parts:

*Geometrical*: In this approach, single or multiple control points or shape parameters are used for manual adjustment of a flexible part in order to apply changes in its shape and model its desired configuration. Generally, Geometrical methods rely on the designer's skill rather than on physical principles, are computationally efficient, and aim to find a possible deformation in a reasonable time although they often produce results not matching with desired configurations. A common method for deformation of geometrical properties is the *Functional* method, in which a shape is expressed as a Bezier, B-spline, rational B-spline, or NURBS curve, and the designer reshapes the curve by adding, deleting, moving, or changing weights of the control points [3]. A much more powerful method is the *Free-Form Deformation* (FFD), which deforms the shape of a flexible object by changing the space that encompasses it. This is done by applying translation, rotation, bending, and other transformation matrices to change the model of the object. More complex deformations can be derived via a combination of these matrices [30].

*Physical*: In this approach that has been mostly used by animators and graphic designers, some sort of integration mechanisms and physical principles (e.g., material characteristics, environmental constraints, and external applied forces on a part) are used to compute the shapes or motions of flexible objects. A flexible part is shown as an entity with many degrees of freedom (DOFs), one for each deformable point on the surface of the object. The main disadvantage of Physical methods is the increase of DOFs to an uncontrollable level; thus, a balance must be made between the required time and the degree of realism. The most commonly used strategies for building deformable objects with physical properties are: (1) Mass-spring; (2) Finite Element Methods; and (3) Point-Based systems [12].

Assembly sequence planning of flexible parts has been rarely addressed in the literature. Wolter and Kroll [40] discussed different types of operations on particularly string-like parts such as threads, ropes, wires, cables, and hoses. They described the state of an assembly by a set of relations among the features of its component parts together with the basic algorithms to determine the possibility of performing a particular operation on string-like assemblies. However, that work is not applicable to 2D and 3D flexible parts, and no method for ASP is proposed. A few other works deal with creating *contact states* of the flexible parts of an assembly. In that approach, information for the contact states is extracted from the geometrical models of the parts and the assembly and is used to construct a graph representation proper for assembly sequence and path planning. In the *Contact State Graph*, each node represents a valid contact state between two parts, and each arc between two nodes implies the adjacency of their corresponding contact states. Based on this graph, relative assembly paths can be planned as the sequences of contact state transitions connecting the starting state to the goal state. Remde et al. [28] identified different possible contact states between a linear deformable object and a rigid polyhedral body, and listed the feasible transitions between these states. A further elaboration on this formalism characterizes contact states by their stability and defining contact state transition classes [1]. Also in [21] the contact forces of an elastic tube during assembly operations are modeled based on different types of contact states with the effects of deformation and friction taken into account. None of the above works, however, directly focus on ASP with flexible parts, and as a result this field of research has remained largely untouched.

This paper aims to fill in this gap, and proposes a new formal framework for defining assemblability of flexible parts, in which concepts like *Assembly Interference Matrix* (AIM) and *Assembly stress matrix* (ASM) for flexible parts are mathematically defined from not only geometrical, but also physical and mechanical aspects, as presented and exemplified in Section 2. Another contribution of this work is solving the ASP problem for products with rigid and flexible parts by customizing the components of the Scatter Search (SS) metaheuristic algorithm, tuning its parameters to their best values through a systematic combination of TOPSIS and Taguchi methods as described in detail in Section 3. In Section 4 the presented method is extensively compared with GA, MA, SA, Breakout Local Search (BLS), Multistart Local Search (MLS), and Iterated Local Search (ILS) methods by solving three sample assemblies with different geometries, dimensions, and complexity via the detailed statistical analyses. In this section, a computational and statistical analysis of flexible vs. rigid modeling is also presented. Finally, concluding remarks are presented in Section 5.

## 2. Assembling flexible parts

Usually, assembling a flexible part requires exerting forces to push it through a narrow passage or establish a latching contact with a mating part. Such a force, however, must strain the part within its elastic limit to avoid permanent deformation (and probably damage) of the part. In order to consider the deformability of flexible parts in assembly sequence planning, their elastic behavior under variable forces must be studied first. In this section, new formal definitions and mathematical symbols for showing interference relations between parts of an assembly and the amount of compressive stress needed for assembling flexible parts are presented, and then the process of creating the Assembly stress matrix (ASM) – which is a generalization of the Assembly Interference Matrix (AIM) – using the Abaqus software is described step-by-step to sufficient details. Lastly, the ASM is constructed for a sample assembly with two rigid and four flexible parts.

In this paper, deformability of the flexible parts of assemblies during the assembly process are simulated by the Abaqus$^{\text{TM}}$ software, which is the reference software for Finite Element Method (FEM) analysis, modeling of deformation behavior of mechanical parts, and Computer Aided Engineering (CAE). For the FEM analysis, the user must input the part's geometrical dimensions, material, density, Young's modulus, coefficient of friction, types of elements used for modeling its deformation behavior, as well as the direction of exerted forces, into the Abaqus software. The software then simulates the part's motion along the given force direction and calculates the exerted stresses and deformations for all finite elements of all contacting parts, which are used for constructing the Assembly stress matrix. It is noted that part variations which affect the assemblies are not considered in this paper.

### 2.1. Assembly stress matrix (ASM)

In order to construct the Assembly stress matrix, we precalculate all possible collisions between any two parts at all directions and organize this information as an $n \times n \times 2m$ matrix called *Assembly Interference Matrix* (*AIM*). Each entry of the AIM is denoted by a binary variable $I_{i,j}^{k}$, which takes value 0 if part $p_j$ does not block the movement of part $p_i$ along direction $d_k$, and takes 1 otherwise. This variable is formally defined in (1) based on