Computer-Aided Design 47 (2014) 59–71

Contents lists available at ScienceDirect

Computer-Aided Design

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

Mechanical assembly planning using ant colony optimization

Hui Wang [a,*], Yiming Rong [a,b], Dong Xiang [a]

[a] Department of Mechanical Engineering, Tsinghua University, Beijing, 100084, China

[b] Department of Mechanical Engineering, Worcester Polytechnic Institute, Worcester, MA, 01609, USA

HIGHLIGHTS

- We model an ant colony optimization based method for mechanical assembly planning.
- The computation framework couples both a solution generation and an optimization search.
- The proposed search strategy improves the performance of the assembly planning method.

ARTICLE INFO

Article history:
Received 29 October 2012
Accepted 9 September 2013

Keywords:
Assembly information modeling
Mechanical assembly planning
Ant colony optimization

ABSTRACT

In mechanical assembly planning research, many intelligent methods have already been reported over the past two decades. However, those methods mainly focus on the optimal assembly solution search while another important problem, the generation of solution space, has received little attention. This paper proposes a new methodology for the assembly planning problem. On the basis of a disassembly information model which has been developed to represent all theoretical assembly/disassembly sequences, two decoupled problems, generating the solution space and searching for the best result, are integrated into one computation framework. In this framework, using an ant colony optimization algorithm, the solution space of disassembly plans can be generated synchronously during the search process for best solutions. Finally, the new method's validity is verified by a case study.

1. Introduction

As an important research topic in the manufacturing automation field, the assembly sequence planning problem encompasses some very active sub-problems, e.g., the representation of assembly constraints [1], the generation of feasible assembly plans [2], and the selection of final assembly plans [3]. Mathematically, this is an NP-hard problem. For a complex mechanical product with many parts, the number of possible assembly sequences may be too large to handle efficiently with traditional methods. This challenge is one important driving force to promote research on computerized assembly/disassembly planning. The pioneering work on assembly planning research had been done by Bourjault (1984, [1]), Homem de Mello and Sanderson (1990, [2]), who developed the basic academic idea of assembly/disassembly planning, "*Product assembly model (graph)* + *Optimization algorithm for disassembly solution*"—that is, modeling the solution space and then, finding out the best result. However, the combinational complexity remains a fundamental challenge even for computerized solving tools. Searching the graph model (solution space) for the best plan is not an easy job, particularly, if considering the difficulty of building an evaluation standard on assembly plan performance. Meanwhile, determining the assembly precedence relationships needs to check geometric interferences by manual or automatic computation which is also time-consuming. This is why many intelligent optimization algorithms become the main powerful tool in disassembly planning study.

During the past several years, significant work has been done to develop Artificial Intelligence (AI) and soft computing techniques applicable to assembly planning in order to attain an optimal solution efficiently. Many applications of AI technologies in assembly/disassembly sequence planning have been introduced [4–18]: particularly, genetic algorithms (GA), expert systems, simulated annealing, Petri nets, and neural networks. The genetic algorithm is a widespread approach for the assembly/disassembly sequence planning problem, owing to its capability to evolve towards optimal solutions without processing all the alternatives. In their GA methods for assembly planning, Lazzerini, Marcellon and Dini, et al. [19], designed a three-part chromosome to represent necessary assembly information including products' component, direction of operation, and used gripper. Konggar and Gupta [20] also proposed a genetic algorithm for disassembly process planning. Similar to Lazzerini's method, each chromosome (solution) consists of three parts of equal length, including the disassembly sequence of components, operational directions, and the sign of destructive or nondestructive methods. A similar encoding method was also used by Galantucci L. M. who

* Corresponding author. Tel.: +86 18911028644; fax: +86 10 62773517.
E-mail address: wanghuisx@gmail.com (H. Wang).

0010-4485/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cad.2013.09.001

Available tools	
Elements	Gripper
A	g1
B	g2
C	g2,g3
D	g2,g3
E	g2,g3

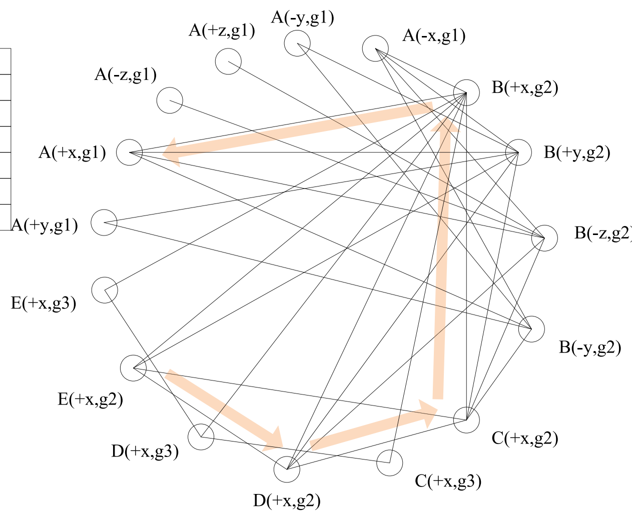

Fig. 1. Using a Disassembly Node Network to represent all possible assembly/disassembly operations.

had presented a fuzzy logic and genetic algorithm system for assembly/disassembly planning [21]. However, the GA search usually heavily depends on the performance of encoding methods and genetic operators. Improper encoding may have a negative impact on describing/searching the space of solutions. For example, binary-coded representations are often affected by the so-called Hamming cliff problem, which may lead to bad offspring, infeasible or far different from parents. All of them may deteriorate the performance of GA algorithms.

The ant colony optimization (ACO) algorithm, one of the most promising biologically inspired optimization algorithms, is emerging as an innovative tool for solving computational problems of finding good paths through graphs. This algorithm paves an interesting way for sequencing problems even though its ability and performance in this field is still to be investigated. For instance, an ACO algorithm for the disassembly line balancing problem (DLBP) has been developed by McGovern and Gupta [22,23]. And the studies done by F. Failli [24] and Wang J. F. [25] were more typical in assembly planning. F. Failli had used a graph called the Disassembly Nodes Network (DNN) to represent all possible solutions (Fig. 1). Each node in the graph represented a disassembly operation, defined as a combination of component, disassembling gripper, and disassembly direction. As F. Failli said, "The software implementation of this method is based on the schematization of the disassembly of a product in a network of nodes connected by links. Each node $\boldsymbol{i}(\boldsymbol{g}, \boldsymbol{d})$ represents a component $\boldsymbol{i}$ grasped by a gripper $\boldsymbol{g}$ and assembled along a direction $\boldsymbol{d}$. The set of all the nodes and all the links constitutes the possible paths for the ants. A single disassembly sequence is defined by a path leading from a start node to an end node and passing through all the components, but not through all the nodes". J. F. Wang used the Disassembly Completed Graph (DCG) to represent all possible assembly sequences (Fig. 2) which looks like a simplification of DNN without considering the tools. In DCG, every disassembly operation was defined by the component and its disassembling direction. Based on the proposed models, Ant Colony Systems (ACS) were used to search the best disassembly/assembly solution(s). Shan also used a disassembly graph in the same way as J. F. Wang before he proposed an ACO method to solve the disassembly sequence planning problem [26].

However, some disadvantages still exist, particularly when processing complex products with many components. Since Homem de Mello proposed his methodology [2] that he developed a cut-set algorithm for assembly sequence planning after giving an AND/OR graph to represent assembly relationships in 1990, almost all following methods did similar work, focusing on the method of assembly plan optimization but no explanation on how to create the

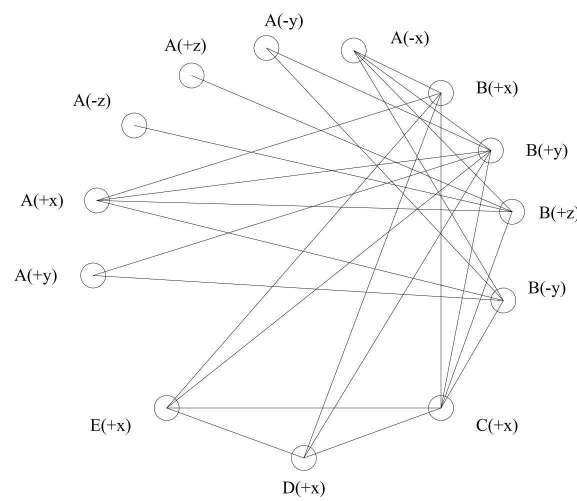

Fig. 2. An instance of Disassembly Completed Graph (it can be regarded as a simplification of DNN).

models (e.g., DCG and DNN) which include all solutions for a search. For example, Shan [9,26] and Liu [27] put their focus on assembly sequence optimization methods with the presumption of the establishment of the assembly/disassembly matrix (a typical adjacency matrix of the proposed assembly sequence graph). And F. Failli pointed out, "*The network is automatically generated from the geometrical relationships existing among components, extrapolated from a representative model of the product. The network is built excluding, a priori, the nodes and the links geometrically unfeasible* (*e.g.: the link* $1(g1, +y) - 2(g3, -y)$ *in Fig* ...)". In fact, it is too difficult to do within an acceptable runtime, particularly for a complex product. Given an assembly with N components, the possible assembly sequences would be $N \times (N-1) \ldots 3 \times 2 \times 1$. Following those studies mentioned above, we should test all possible sequences to generate the assembly sequences graph, confirm the assembly constraints, and then provide the results to optimization algorithms for searching the best. Considering the fact that for testing a disassembly sequence a lot of 2D or 3D geometric computation is required, it is a time-consuming work—even on a powerful workstation, 3D geometric computation remains a costly job. And in real practice, when handling a mechanical assembly plan, engineers hardly have the time and patience to determine the final solution after comparing the performance of all possible assembly sequences in detail.