*Artificial Intelligence for Engineering Design, Analysis and Manufacturing* (2003), *17*, 325–333. Printed in the USA.
Copyright © 2003 Cambridge University Press 0890-0604/03 $16.00
DOI: 10.1017/S0890060403174045

# Intelligent selective disassembly using the ant colony algorithm

J.F. WANG,[1] J.H. LIU,[2] S.Q. LI,[1] AND Y.F. ZHONG[1]

[1] School of Mechanical Science and Engineering, Huazhong University of Science & Technology, Wuhan, People's Republic of China
[2] School of Mechanical Engineering and Automation, Beihang University, Beijing, People's Republic of China

(RECEIVED April 10, 2003; ACCEPTED October 20, 2003)

**Abstract**

Selective disassembly is an important issue in industrial and mechanical engineering for environmentally conscious manufacturing. This paper presents an intelligent selective disassembly approach based on ant colony algorithms, which take inspiration from the behavior of real ant colonies and are used to solve combinatorial optimization problems. For diverse assemblies, the algorithm generates different amounts of ants cooperating to find disassembly sequences for selected components, minimizing the reorientation of assemblies and removal of components. A candidate list that is composed of feasible disassembly operations, which are derived from a disassembly matrix of products, guides sequence construction in the implicit solution space and ensures the geometric feasibility of sequences. Preliminary implementation results show the effectiveness of the proposed method.

**Keywords:** Ant Colony Optimization; Disassembly Matrix; Disassembly Sequence Planning; Selective Disassembly

## 1. INTRODUCTION

With the increasing interest in environmentally conscious manufacturing (Gungor & Gupta, 1999), product remanufacturing, part reuse, and materials recycling are advocated for obtaining both resource conservation and waste reduction. Disassembly is becoming an important issue in industrial and mechanical engineering. Because of technical and economic constraints, disassembly cannot usually be performed to the full extent. Therefore, the concept of incomplete disassembly (Lambert, 2002) is introduced. Given the product model, selective disassembly (SD), which is a kind of incomplete disassembly, involves the determination of sequences and directions for a selected set of components satisfying an objective function (Srinivasan et al., 1997). The purposes of SD include assembling, maintenance, and recycling of components. A clear insight into the process of selective disassembly not only enables the design of effective reuse/recycling processes but also supports new concepts of product design that seek to optimize the product's life cycle.

Reprint requests to: J.F. Wang, School of Mechanical Science and Engineering, Huazhong University of Science & Technology, Wuhan, Hubei 430074, People's Republic of China. E-mail: wangjf2000@tom.com

Over the past decades, many studies have described efforts to complete disassembly (CD), which involves disassembling all the components in the assembly. The objective of CD is to find optimal or near-optimal disassembly process plans that minimize the cost of disassembly. Most researchers used graph-based approaches such as an assembly sequence diagram (De Fazio & Whitney, 1987), AND/OR graph (Homem De Mello & Sanderson, 1991), nondirectional blocking graph (Jones et al., 1997), component-fastener graph (Kuo et al., 2000), disassembly graph (Lambert, 2002), and so on to solve the CD problem. However, the number of alternative disassembly process plans increases dramatically with the number of parts. It is nearly impossible to generate and evaluate all the disassembly sequences to seek the optimal one using a graph-based approach. Thus, other significant studies have been made to develop artificial intelligence (AI) techniques for the CD problem. The AI methods, such as case-based reasoning (Zeid et al., 1997), expert systems (Watkins et al., 1995), petri nets (Zha & Lim, 2000; Zussman & Zhou, 1999), neural networks (Huang et al., 2000), and genetic algorithms (Seo et al., 2001) have been examined. In general, AI methods can improve the efficiency of the process to search for feasible/optimal disassembly sequences.

The CD does not give optimal solutions, although a SD sequence can be obtained from it (Srinivasan & Gadh, 2000).



https://doi.org/10.1017/S0890060403174045 Published online by Cambridge University Press

There have been relatively few investigations of techniques for SD. A geometric algorithm called "wave propagation" (Srinivasan & Gadh, 1998, 2000; Srinivasan et al., 1999) was presented for the SD of single and multiple component(s). The algorithm analyzes the product from the selected parts outward and determines a set of sequences by topologically ordering components in the assembly. By evaluating the resulting sequences based on the objective of minimum component removals, an optimal sequence is identified. Srinivasan and Gadh (2002) also discussed global selective disassembly considering spatial and user-defined constraints. A disassembly tree structure was proposed for single component disassembly (Woo & Dutta, 1991); later it was used to recognize parallel assemblies as well as to derive an efficient disassembly sequence for multiple components in sequential assemblies (Dutta & Woo, 1995). Starting with the precedence graph of the assembly, Garcia et al. (2000) presented an SD algorithm that computes the minimum distance from the exterior components of the assembly to the rest of components. Li et al. (2002) used a hybrid graph to model the product under maintenance. From the target part outward, a reasoning mechanism is used to obtain necessary disassembly operations to move the target part. Then, a genetic algorithm is integrated to optimize the sequences of resulting disassembly operations, and optimal disassembly sequences for a maintenance part are attained.

SD has varying starting components and end components because of different targets in disassembly tasks. Two situations (i.e., single-target component and multiple-target components) should also be considered. Moreover, the structures of disassembly sequences of SD are nonhomogeneous for their distinct lengths of sequences. Such characteristics make SD more complicated than CD. This paper presents an SD approach based on ant colony optimization (ACO) algorithms, which are new metaheuristic methods for combinatorial optimization problems. To our knowledge, this is the first attempt to use AI methods for the SD problem. The experimental results suggest that the performance of the methodology is desirable and the proposed approach is applicable.

## 2. AN OVERVIEW OF ACO ALGORITHMS

ACO algorithms (Dorigo, 1992) are a branch of a larger field referred to as swarm intelligence (Bonabeau et al., 1999), which is the behavior simulation of social insects such as bees, ants, wasps, and termites. More specifically, it is an AI procedure that imitates the behavior of social insects in an attempt to solve combinatorial optimization problems within a reasonable amount of time.

Some research results show that ants can find the shortest path between their nest and the source of foods. Each ant leaves behind a kind of volatile secretion (pheromone) on the path it has used; then, other ants will choose their way according to the amount of pheromone on the path. With the continuous action of the colony, the shorter paths are more frequently visited and become more attractive for subsequent ants. By contrast, the longer paths are less attractive because the pheromone trail will evaporate with passing time. The shortest way between the nest and the source of foods is finally found. The process is illustrated in Figure 1, which shows a distribution of ants over a set of pathways between a nest and food source over time. Initially, the ants are equally distributed, but eventually they favor the shorter route.

The first ACO algorithm was successfully applied to the traveling salesman problem (TSP) that can be expressed as searching for optimal paths on graphs (Dorigo et al., 1996; Dorigo & Gambardella, 1997). Later, the application results in manufacturing, such as just in time sequencing (McMullen, 2001) and flow-shop scheduling (T'kindt et al., 2002), show that it is better than or as effective as many other heuristic approaches. As to the assembly/disassembly planning problem, Failli and Dini (2000) attempted to use the ant system for the detection and optimization of assembly sequences. The influence of parameters on the performances of the algorithm has been carried out.

The steps of generic ACO algorithms are as follows.

1. *Initialization*. The problem is represented by a graph. Set the initial population of the colony and the pheromone trail. Place all ants randomly on the starting nodes.
2. *Solution construction*. Taking into account the problem-dependent heuristic information and the trail intensity of the path, each ant chooses the next node that has not been visited according to some probability. Repeat the step until a completed solution is constructed.
3. *Trail update*. Evaluate the solutions and deposit pheromone on the solutions path according to the quality of solutions. The better the solution, the greater the amount of pheromone deposited. In addition, the pheromone trail of all paths is decreased by some constant factor at the end of an iteration to build solutions.

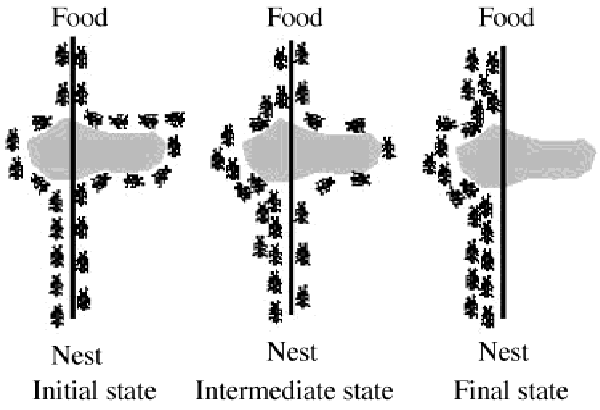



**Fig. 1.** The process of ants finding the shortest path.

https://doi.org/10.1017/S0890060403174045 Published online by Cambridge University Press

There are some disadvantages when ACO algorithms are applied to combinatorial optimization problems. The performance of ACO algorithms strongly depends on the values initially assigned to the parameters. The selection of the proper values is also not supported by an explicit theory that gives univocal directions for solving different problems. The main characteristics of ACO algorithms are positive feedback, distributed computation, and constructive greedy heuristic search. Positive feedback accounts for rapid discovery of good solutions, distributed computation avoids premature convergence, and the greedy heuristic search helps find acceptable solutions in the early stages of the search process.

## 3. ACO-BASED SELECTIVE DISASSEMBLY

The SD problem considered in this paper is to find optimal disassembly sequences for single or multiple target components, which form a target list of SD. The optimal sequences concern geometric constraints of the assembly, reorientations of the assembly during the disassembly process, and removals of components in the assembly. The objective is to reduce the disassembly cost of SD. The proposed ACO-based algorithm finds target components starting from the initial dismountable components inward by multiple ants simultaneously. Three conditions are required when applying ACO algorithms to a combinatorial problem: an appropriate graph representation searched by many simple agents, the constraint satisfaction by tabu list, and the positive feedback process and heuristic search that allow a constructive definition of the solutions.

The assumptions for the current research, which are standard assumptions in automated assembly/disassembly analysis (Srinivasan & Gadh, 2002), are as follows:

1. the relative motions of the components are determined without considering the tools, fixtures, or robots required to achieve these motions;
2. assemblies are assumed to be rigid, frictionless, and defined by nominal geometry;
3. components are "1-disassemblable" (a single linear motion along the directions of coordinate axes from the assembly); and
4. disassembly sequences are monotonic (the components are totally removed while disassembling) and nondestructive.

An illustrative ballpoint pen assembly (De Fazio & Whitney, 1987) shown in Figure 2 is used to introduce the approach, where component 5 (ink) is chosen as the target component to be disassembled.

### 3.1. Graph representation of the problem

A disassembly sequence is an ordered list of disassembly operations (DOs). When considering the disassembly direction of components, a single DO can be depicted as a duple,

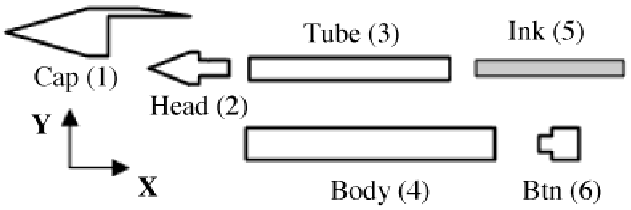



**Fig. 2.** An illustrative ballpoint pen assembly.

$$\mathrm{DO} = (N, D), \tag{1}$$

where $N$ is the number of the component in an assembly and $D$ stands for the disassembly direction of the components. In this paper each component has six possible DOs: $(N, +x)$, $(N, +y)$, $(N, +z)$, $(N, -x)$, $(N, -y)$, and $(N, -z)$. If an assembly is made up of $n$ components, a disassembly completed graph (DCG) that has $6n$ DO nodes can be built. A DCG is an undigraph in which every DO node in the graph has a connection with all other DO nodes. Starting from any node, a geometric feasible or unfeasible disassembly sequence will be generated after traveling all the nodes that have different component numbers from each other. All these sequences form the solution space of the SD problem.

Usually, an assembly has a few initial dismountable components that form the initial feasible DOs. From these DOs, different sequences of DOs containing the target components can be found by searching the DCG of the assembly. A partial DCG of the ballpoint pen assembly is shown in Figure 3. From the node $(1, -x)$, which is an initial feasible DO of the assembly, a possible disassembly sequence for component 5, $\{(1, -x) - (2, -x) - (5, -x)\}$, can be gained according to the above search principle.

It is worth mentioning that the DCG of the SD problem is not constructed explicitly in advance in the algorithm of this paper. It can easily be generated dynamically in an implicit manner during the search process of ants. Therefore, large amounts of storage space are saved and many modeling loads can be reduced.

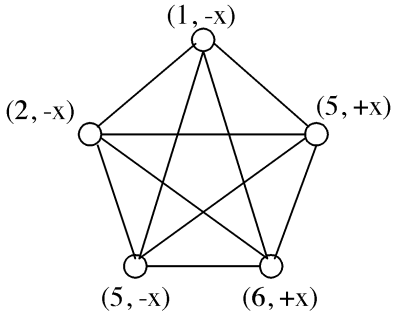



**Fig. 3.** A partial DCG of the illustrative assembly.

https://doi.org/10.1017/S0890060403174045 Published online by Cambridge University Press

In ACO algorithms, a pheromone matrix (PM) is used as a shared memory of all ants. It records the pheromone amount deposited on all edges to guide the search of ants. In this paper the PM is expressed as a $6n \times 6n$ matrix because the DCG has $6n$ nodes. An element of PM at time $t, \tau_{ij}(t)$, indicates the visiting status from node DO $i$ to node DO $j$. Apparently, PM is an asymmetric matrix because of disassembly constraints among components of products.

### 3.2. Geometric constraints of assemblies

Comparatively, the SD problem has more constraints than the TSP when comparing the components of SD and the cities of TSP. For example, a disassembly sequence cannot start from any component of the assembly, but a tour can start from any city. Not all components may be reached from any other component, but any nonvisited city may be reached from any other city. Not any valid connection between two components can be done at any time, but any valid connection between two cities can be done at any time. A disassembly sequence is unfeasible, not to mention optimal, if some hard constraints such as geometric precedence constraints are not satisfied.

When ants build disassembly sequences by searching the DCG of the products, the subsequences of DOs with the same component number (e.g., $\{(5, -x) - (5, +x)\}$ as shown in Fig. 3) are obviously illegal because a component cannot be disassembled twice in an assembly. Other subsequences may be unfeasible because they violate geometric constraints of the assembly (e.g., $\{(2, -x) - (1, -x)\}$ shown in Fig. 3). In order to guarantee the validity and feasibility of sequences, a disassembly matrix (DM; Huang & Huang, 2002) derived from the interference matrix (Dini & Santochi, 1992) is chosen as the mathematical model of the products and is adapted to the algorithm. It is used to generate initial feasible DOs and a candidate list of DOs. The number of initial feasible DOs will be taken as the quantity of ants, and these nodes in the DCG will be set as the starting nodes of each ant. The candidate list of DOs is used to restrict the choice of next DO during the search process of ants.

Given that an assembly comprises $n$ components represented as $(e_1, e_2, \ldots, e_n)$, DM denotes the interference relationship of a component with all other components in the assembly during its disassembly along the $\pm k$ axis, where $k \in (x, y, z)$. If the component $e_i$ of the assembly does not interfere with another component $e_j$ in the direction of the $+k$ axis, the component $e_i$ can be disassembled freely from the component $e_j$ in the direction of the $+k$ axis. Consequently, the component $e_j$ can be disassembled from the component $e_i$ in the direction of the $-k$ axis. If the component $e_i$ that moves in the direction of the $+k$ axis has interference with another component $e_j$, then the component $e_j$ that moves in the direction of the $-k$ axis will in turn have interference with the component $e_i$. DM is described as a $n \times 3n$ matrix:

$$\mathrm{DM} = \begin{array}{c} \\ e_1 \\ e_2 \\ \ldots \\ e_n \end{array} \begin{array}{c} \begin{array}{cccc} e_1 & e_2 & \ldots & e_n \end{array} \\ \begin{bmatrix} I_{11x}I_{11y}I_{11z} & I_{12x}I_{12y}I_{12z} & \ldots & I_{1nx}I_{11y}I_{11z} \\ I_{21x}I_{21y}I_{21z} & I_{22x}I_{22y}I_{22z} & \ldots & I_{2nx}I_{2ny}I_{2nz} \\ \ldots & \ldots & \ldots & \ldots \\ I_{n1x}I_{n1y}I_{n1z} & I_{n2x}I_{n2y}I_{n2z} & \ldots & I_{nnx}I_{nny}I_{nnz} \end{bmatrix} \end{array}, \quad (2)$$

where $I_{ijk}$ is equal to ***1*** if component $e_i$ interferes with component $e_j$ during disassembling along the $+k$-axis direction; otherwise, $I_{ijk}$ is equal to ***0***. As a convention, $I_{iik}$ is always equal to ***0***. The initial DM of the ballpoint pen assembly is given below, where only $+x$, $-x$, $+y$, and $-y$ directions are considered.

| | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $x$ | $y$ | $x$ | $y$ | $x$ | $y$ | $x$ | $y$ | $x$ | $y$ | $x$ | $y$ |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 3 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 4 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 5 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

From the DM, all the feasible DOs of each component at a disassembly step can be derived. For example, the feasible DOs of component $e_i$ along with $\pm k$ axis are computed as follows.

$$\mathrm{DO}_{i,(+k)} = \bigcup_{j=1}^{n} I_{ijk}, \quad (3)$$

$$\mathrm{DO}_{i,(-k)} = \bigcup_{j=1}^{n} I_{jik}, \quad (4)$$

where $\cup$ is the Boolean operator "OR." The result will be equal to ***0*** if all elements involving in the operation are ***0***, which means that component $e_i$ can be disassembled freely in the corresponding direction and one DO is gained. If any element taking part in the operation is equal to ***1***, the result will be equal to ***1*** and the component $e_i$ cannot be disassembled in that direction. In addition, a specific tabu list recording the components that have been disassembled is also used. The DOs of components in the tabu list are not computed when using Eqs. (3) and (4). For the ballpoint pen assembly, the initial feasible DOs computed are $(1, -x)$ and $(6, +x)$.

It is noted that some commonsense knowledge is also incorporated into the algorithm so that reasonable candidate DOs can be achieved. For example, fasteners that constrain the same parts are considered as one part in DM and take precedence over the fastened components that are to be dismounted according to the information of fasteners input to the algorithm. All reasonable DOs consist of the candidate list of an ant. After a component $e_i$ has been chosen as the next one to be disassembled, it is added to the tabu list of the ant and then the DM is updated. What is necessary is to set the value of related elements [i.e., $I_{ijk}$ and

https://doi.org/10.1017/S0890060403174045 Published online by Cambridge University Press

$I_{jik}$, where $j \in (1, 2, \ldots, n)$] in the DM to $\boldsymbol{0}$. The new DM will be used the next time to generate the candidate list of the ant.

Suppose that two ants are searching disassembly solutions for component 5 on the partial DCG of the ballpoint pen assembly shown in Figure 3. For the first ant starting from the initial DO $(1, -x)$, one of the feasible subsequent DOs [i.e., $(2, -x)$ and $(6, +x)$] will be chosen as the next DO in its disassembly sequence. As to the second ant, one of the subsequent DOs [i.e., $(1, -x)$ and $(5, +x)$] will be chosen when it starts from the initial DO $(6, +x)$. Evidently, not all of the edges of the DCG are in effect during the search processes of ants when candidate lists are used. The effective edges and two possible trails of two ants on the partial DCG are shown in Figure 4, where the useless edges are not denoted.

It is the continuous update of the DM together with the generation of candidate lists that realize the dynamic construction of DCG implicitly and the choice of the next DO of disassembly sequences. Each ant owns the same copy of the initial DM of the final product at the beginning. In the course of the search process for target components, ants generate the candidate list and tabu list, build the sequence, and update the DM of themselves independently.

### 3.3. Solution generation

In ACO algorithms of TSP, a so-called random-proportional probabilistic state transition rule is used to choose the next city by tabu list at time *t*. Concerning the SD problem, an ant generates a candidate list of DOs and compares the candidate list with the target list at each disassembly step. If none of the target components are included in the candidate list, the probability with which ant *k* in DO *i* chooses DO *j* at time *t* is defined as in Eq. (5). Otherwise, one or more target components have been found and the probability is computed as in Eq. (6).

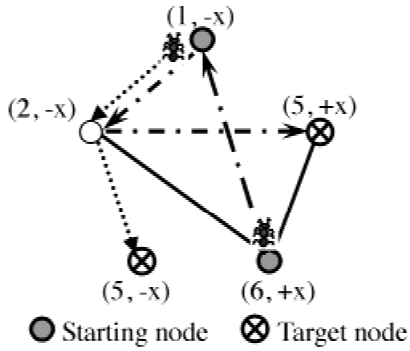



**Fig. 4.** The effective edges and two possible trails of ants on the partial DCG.

$$p_{ij}^k(t) = \begin{cases} \dfrac{\tau_{ij}(t)\cdot[\eta_{ij}]^\beta}{\sum_{u\in CC_i^k(t)} \tau_{iu}(t)\cdot[\eta_{iu}]^\beta}, & \text{if } j \in CC_i^k(t) \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

$$p_{ij}^k(t) = \begin{cases} 1, & \text{if } j \in PC_i^k(t) \text{ and } PC_i^k(t) \text{ has only one DO} \\ \dfrac{\tau_{ij}(t)\cdot[\eta_{ij}]^\beta}{\sum_{u\in PC_i^k(t)} \tau_{iu}(t)\cdot[\eta_{iu}]^\beta}, & \text{otherwise.} \end{cases} \tag{6}$$

In the above two formulas, $\tau_{ij}(t)$ is the quantity of pheromone corresponding to the positive feedback loop, $\eta_{ij}$ is the problem-dependent heuristic information corresponding to the greedy search, $CC_i^k(t)$ is the complete candidate list generated by the DM after the component of DO *i* has been dismounted, $PC_i^k(t)$ is the partial candidate list generated by removing the DOs that do not belong to target components from the complete candidate list, and $\beta$ is a parameter that determines the relative importance of pheromone versus heuristic information. Here, parameter $\eta_{ij}$ is equivalent to the distance between two cities of the TSP. It concerns changes of the disassembly direction (i.e., the disassembly reorientations) from DO *i* to DO *j* and is defined as

$$\eta_{ij} = \begin{cases} \eta_1, & \text{if the direction do not change} \\ \eta_2, & \text{otherwise} \end{cases}, \tag{7}$$

where $\eta_1$ is a large positive constant and $\eta_2$ is a small one. In this work, $\eta_1 = 1$ and $\eta_2 = 0.3$.

While constructing the sequence, if an ant carries out the transition from DO *i* to DO *j*, the pheromone value of the corresponding edge in the PM is modified using the local updating rule:

$$\tau_{ij}(t) = (1-\rho)\cdot\tau_{ij}(t) + \rho\cdot\tau_0, \tag{8}$$

where $0 < \rho < 1$ is the rate of pheromone evaporation and $\tau_0$ is the initial value of pheromone. During one tour of all ants, the local updating of the pheromone makes the edges used by early ants become slightly less desirable to subsequent ants because these edges lose some of their pheromone. This makes the desirability of DOs change dynamically, thus avoiding premature convergence. All ants would search in a narrow neighborhood of the best sequences found previously without local updating.

The task of an ant is to construct a disassembly sequence that contains all the target components. Once an ant has finished its task, it will stop building the sequence. If all ants have finished their tasks, the edges belonging to the iteration-best solutions, which usually are not unique to the SD problem, are reinforced with extra amounts of pheromone. At the same time, evaporation of pheromone on all edges also occurs. The global updating rule is

$$\tau_{ij}(t+1) = (1-\gamma)\cdot\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^k(t), \tag{9}$$

https://doi.org/10.1017/S0890060403174045 Published online by Cambridge University Press

where $0 < \gamma < 1$ is the pheromone decay parameter, and $m$ is the number of ants that find the iteration-best sequences. The quantity of pheromone deposited by ant $k$ is calculated as

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \dfrac{q}{Q_k}, & \text{if } (i,j) \in \text{sequence of ant } k \\ 0, & \text{otherwise} \end{cases}, \tag{10}$$

where $q$ is a constant number to adjust the increment of pheromone and $Q_k$ is used to evaluate the sequence quality (defined as the disassembly cost in this paper):

$$Q_k = c_r \times R + c_l \times L, \tag{11}$$

where $R$ and $L$ stand for reorientations and length (removals of components) of the sequences, respectively, and $c_r$ and $c_l$ are the weights of $R$ and $L$. The solutions with the least $Q_k$ value are the iteration-best solutions. The global updating rule enforces the pheromone according to the sequence quality. The better sequences with smaller $R$ and $L$ values tend to have a greater increment of pheromone and become more attractive to subsequent ants.

During the construction of sequences, local pheromone updating encourages "exploration" of alternative solutions, whereas global pheromone updating encourages "exploitation" of the most promising solutions.

### 3.4. Proposed algorithm for SD

In the algorithm, the parameters such as importance factor $\beta$ and evaporation rate $\rho, \gamma$ are set initially. The DM of the final assembly is provided by means of interference checking on a solid computer-aided design assembly model of the products. During a cycle of the search process, an ant finds a disassembly sequence for target components independently. All sequences found by ants are evaluated and the PM is updated. After a number of iterations, the optimal or near-optimal disassembly sequences for target components will be obtained. The detailed procedure of the proposed ACO-based SD algorithm is given below.

#### *3.4.1. ACO-based selective disassembly algorithm*

The *input* consists of the target components list, the fastener information, the total cycles ($NC_{max}$), the initial pheromone ($\tau_0$), the DM of the final assembly, the importance factor ($\beta$), the evaporation rate ($\rho, \gamma$), the weight coefficient ($c_r, c_1$), and the constant number ($q$). The *output* comprises the optimal or near-optimal disassembly sequences for the target components.

***Begin***

1. Generate the initial reasonable DOs and compute their amount *ma*;
2. Set cycle counter NC := ***1***;
3. **While** NC $<$ $NC_{max}$
4. Place *ma* ants on the initial reasonable DO nodes of the DCG;
5. **While** each ant has not finished its task
6. Put the current DO into the sequence of the ant;
7. Generate a candidate list of the ant by Eqs. (3) and (4);
8. Compare the candidate list with the target list;
9. Calculate selected probability $p_{ij}^{k}(t)$ using Eqs. (5) and (6);
10. Decide the next DO $j$ based on Monte Carlo selection;
11. Move the ant to DO $j$;
12. Add the component number of DO $j$ to the tabu list of the ant;
13. Locally update the PM according to Eq. (8);
14. Evaluate the quality of all solutions;
15. Globally update the PM using iteration-best solutions by Eq. (9);
16. Update the best sequence of each ant if the iteration sequence is the best one of the ant found so far;
17. Empty the disassembly sequence, candidate list, and tabu list of each ant.
18. Set NC :=NC + 1;
19. Output the best sequence of each ant.

***End***

The final solutions are the optimal or near-optimal disassembly sequences for target components found by all ants. It is reasonable that the preferred sequences are chosen from the final solutions by experienced disassembly technicians because not all the disassembly knowledge can be incorporated into the computer programs.

## 4. IMPLEMENTATION AND DISCUSSION

### 4.1. Case study

The algorithm has been implemented in C++ on a 400 MHz Pentium-II PC with 128 MB of memory. To validate the proposed algorithm, a wheel support assembly from Srinivasan et al (1999) shown in Figure 5 is analyzed. The set of target components is ($C_3$, $C_6$). The assembly has nine components and each component owns six possible disassembly directions along three axes. Thus, the implicit DCG has 54 nodes. The initial reasonable DOs computed are (8, $+z$) and (9, $+z$). Two ants are set to search the optimal selective disassembly sequences. Based on certain guidelines (Dorigo et al., 1997; Failli & Dini, 2000), the following parameters have been chosen after trial computations: $\tau_0 = 1$, $\beta = 0.8$,

https://doi.org/10.1017/S0890060403174045 Published online by Cambridge University Press

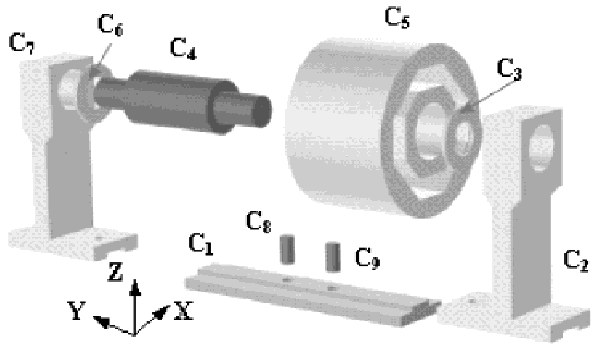



**Fig. 5.** The solid model of a wheel support assembly (adapted from Srinivasan et al., 1999).

$\rho = \gamma = 0.1$, $NC_{max} = 300$, $q = 9$, $c_r = 0.1$, and $c_l = 1$. The values of $c_r$ and $c_l$ are chosen to emphasize the importance of the removal of components so that the results can be compared with those in the literature.

The final solutions of two ants are $\{(C_8, +z) - (C_7, +y) - (C_6, +y) - (C_4, +y) - (C_3, +z)\}$ and $\{(C_9, +z) - (C_2, -y) - (C_3, -y) - (C_4, -y) - (C_6, +x)\}$, which have equal sequence quality with two reorientations of assembly and five removals of components. The disassembly cost of two sequences is $Q_k = 5.2$. The obtained sequences are the same as those of the wave propagation approach (Srinivasan et al., 1999). The CPU time of 300 cycles is 5.268 s, and there were slight differences among trial computations.

The product shown in Figure 6 is an industrial driver assembly (Boothroyd et al., 1994). The component for maintenance is the sensor (component 9). There are 96 nodes in the DCG of the 16-component assembly, where the screws

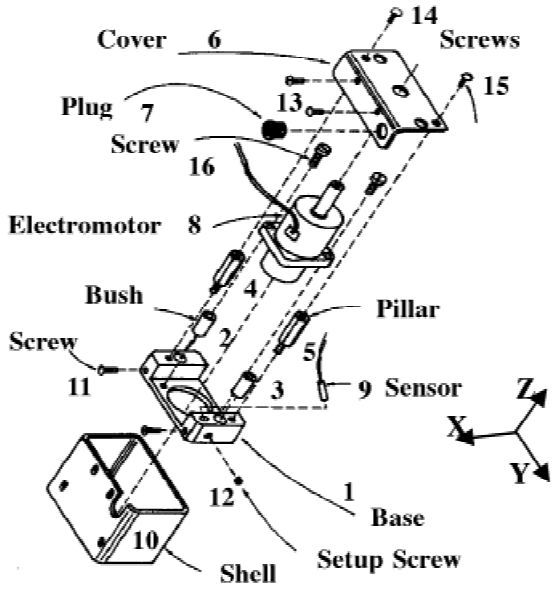



**Fig. 6.** A driver assembly example (adapted from Boothroyd et al., 1994).

that fasten the same two components are grouped as one component. The parameters are chosen as follows: $\tau_0 = 1$, $\beta = 0.8$, $\rho = \gamma = 0.1$, $NC_{max} = 300$, $q = 16$, $c_r = 1$, and $c_l = 1$. In this assembly it can be computed that DO $(7, +x)$, $(11, +x)$, $(13, +x)$, $(14, +z)$, and $(15, +z)$ are the initial reasonable DOs, where component 7 is a plug and components 11, 13, 14, and 15 are all screws. Thus, five ants (ants 1, 2, 3, 4, and 5) are initialized in the algorithm.

Using the average disassembly cost of five sequences belonging to an ant after five cycles as the sampling data, the behaviors of ant 1 and ant 4 in the evolutionary process are illustrated in Figure 7a, which is chosen randomly from trial runs. During the process, the ants perhaps find the best sequences with the least disassembly cost attributing to their stochastic search characteristics. Fewer visits to the best sequences do not make the algorithm converge yet. However, with the increase of cycles, the best sequences are visited frequently because of their higher attraction to the ants, and the algorithm finally converges. Figure 7b shows the evolutionary process of the minimal and average disassembly costs of five ants. It can be observed that the minimal and average costs of the sequences gradually decrease under the efforts of all ants. The total computation time of 300 iterations is 25.196 s. The typical global optimal sequence with three reorientations and nine removals of components belonging to the final sequences of ant 1 is $\{(7, +x) - (11, +x) - (13, +x) - (10, -z) - (12, +y) - (14, +z) - (15, +z) - (6, +z) - (9, +z)\}$, which has a disassembly cost $Q_k = 12.0$. Ant 2 and ant 3 can also find the optimal sequences with the same disassembly cost as ant 1. The major difference among these sequences is the order of the first three DOs in the above sequence. Ant 4 and ant 5 find the near-optimal sequences with four reorientations and nine removals of components. One of them is $\{(14, +z) - (15, +z) - (11, +x) - (13, +x) - (7, +x) - (10, -z) - (12, +y) - (6, +z) - (9, +z)\}$, which has a disassembly cost $Q_k = 13.0$.

### 4.2. Discussion

The results show that the optimal disassembly sequences for SD are not unique because of the diverse starting DOs of each ant. This is different from the solutions of the TSP, which has only one travel path when the algorithm converges. However, in fact, some of the optimal sequences have the same sense to the target components (e.g., the optimal sequences of ants 1, 2, and 3 for the driver assembly). For an ant, the optimal sequences with the same disassembly cost may also have slight differences in the SD applications. The reason is that the positions of some components in the sequences (e.g., components 14 and 15 of the driver assembly in the global optimal sequence of ant 1 shown above) are interchangeable, but the disassembly cost does not change.

The quantity of ants in traditional ACO algorithms is decided on the basis of the scale of the problem (Dorigo

https://doi.org/10.1017/S0890060403174045 Published online by Cambridge University Press

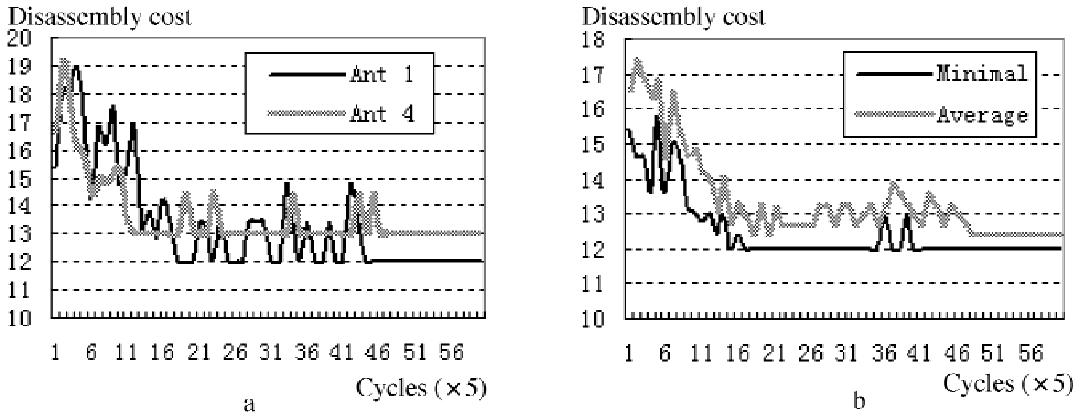



**Fig. 7.** The evolutionary process for SD of the driver assembly.

et al., 1996). As to the SD problem, assemblies usually have different amounts of initial dismountable components. It is an appropriate choice to decide the quantity of ants according to the number of initial dismountable components and start the search process from these components because of the highly constrained characteristics of the problem. Although the quantity of ants is relatively small because of a few initial feasible DOs in the assemblies, the algorithm is still efficient from the test examples by using the candidate list to reduce the search space.

In the wave propagation approach (Srinivasan & Gadh, 1998, 2000; Srinivasan et al., 1999), geometric constraints are used to reason and enumerate all the sequences for SD. The alternative sequences will dramatically increase with the complexity of products. The proposed algorithm of this paper does not need to enumerate all the alternative sequences. In order to find the target components, the algorithm uses geometric reasoning by DM of the assembly to obtain the dismountable components that guarantee the feasibility of sequences and uses the ACO method to optimize the sequences.

The parameters strongly influence the convergence of the algorithm. Among the trial computations, a higher rate of pheromone evaporation weakens the effects of the positive feedback and more cycles in the algorithm are needed. Under certain amounts of cycles, the algorithm will not converge when a small $q$ value is used. However, a larger $q$ value will make the algorithm converge to the local optimal solutions. As a trade-off, it is appropriate to set the value of $q$ as the amount of components in the assembly according to the trial computations.

The limitation of the approach is that only sequential, linear, and monotone sequences can be generated currently. The function of disassembly cost is relatively simple and should include more practical requirements. The relevance and general contribution of this paper is that a novel ant colony algorithm for selective disassembly is presented and implemented here first. The strengths of graph-based geometric reasoning and AI-based optimization are integrated into one scheme for solving the SD problem. Virtual ants are used to address an important issue in manufacturing.

## 5. CONCLUSIONS

An intelligent SD approach based on ACO algorithms is presented in this paper. The objective is to minimize the disassembly cost concerning the reorientation of products and removal of components for a single or multiple target components. The initial reasonable disassembly operations are chosen as the starting places of ants, and its amount is set as the quantity of ants in the algorithm. The solution space is constructed dynamically and implicitly. Geometric reasoning based on the DM is used to generate candidate lists, which ensure the geometric feasibility of the sequences and decrease the search space. Some preliminary implementation results show the encouraging performance of the approach for solving the SD problem of both two- and three-dimensional assemblies.

Future work will consider user-defined constraints (Srinivasan & Gadh, 2002), such as the grouping of components as subassemblies and directional constraints on component geometry. By means of a “contracted” matrix (Dini & Santochi, 1992) of the assembly, a group of components can be treated as a single unit in the DM. Directional constraints, such as one that restricts a component to move or one that constraints along one or more disassembly directions, can also be incorporated with the generation of candidate list. An SD tool integrated with ProEngineer™ 2000 is also being developed, which allows users to select components for the SD, perform intelligent SD analysis, and visualize the selective disassembly sequences.

## ACKNOWLEDGMENTS

This research is supported by grants from the National Nature Science Foundation of China (Grant 59990470-2). The authors thank Prof. F. Failli for his help and the anonymous reviewers for their comments and constructive suggestions to improve this paper.

https://doi.org/10.1017/S0890060403174045 Published online by Cambridge University Press

## REFERENCES


Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford: Oxford University Press.

Boothroyd, G., Dewhurst, P., & Knight, W. (1994). *Product Design for Manufacture and Assembly*. New York: Marcel Dekker.

De Fazio, T.L., & Whitney, D.E. (1987). Simplified generation of all mechanical assembly sequences. *IEEE Journal of Robotics and Automation, RA-3(6),* 640–658.

Dini, G., & Santochi, M. (1992). Automated sequencing and subassembly detection in assembly planning. *Annals of the CIRP, 41(1),* 1–4.

Dorigo, M. (1992). *Optimization, learning and natural algorithms*. PhD Thesis. Politecnico di Milano, Italy.

Dorigo, M., & Gambardella, L.M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation, 1(1),* 53–66.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). The ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics Part B, 26(1),* 29–41.

Dutta, D., & Woo, T.C. (1995). Algorithm for multiple disassembly and parallel assemblies. *ASME Journal of Engineering for Industry, 117(1),* 102–109.

Failli, F., & Dini, G. (2000). Ant colony systems in assembly planning: a new approach to sequence detection and optimization. *Proc. Second CIRP Int. Seminar on Intelligent Computation Manufacturing Engineering,* pp. 227–232.

Garcia, M.A., Larre, A., Lopez, B., & Oller, A. (2000). Reducing the complexity of geometric selective disassembly. *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems,* pp. 1474–1479.

Gungor, A., & Gupta, S.M. (1999). Issues in environmentally conscious manufacturing and product recovery: a survey. *Computers & Industrial Engineering, 36(4),* 811–853.

Homem De Mello, L.S., & Sanderson, A.C. (1991). A correct and complete algorithm for the generation of mechanical assembly sequences. *IEEE Transactions on Robotics and Automation, 7(2),* 228–240.

Huang, H.H.T., Wang, M.H., & Johnson, M.R. (2000). Disassembly sequence generation using a neural network approach. *Journal of Manufacturing Systems, 19(2),* 73–82.

Huang, Y.M., & Huang, C.T. (2002). Disassembly matrix for disassembly processes of products. *International Journal of Production Research, 40(2),* 255–273.

Jones, R.E., Wilson, R.H., & Calton, T.L. (1997). Constraint based interactive assembly planning. *IEEE Proc. Int. Conf. Robotics Automation,* pp. 913–920.

Kuo, T.C., Zhang, H.C., & Huang, H.S. (2000). Disassembly analysis for electromechanical products: a graph-based heuristic approach. *International Journal of Production Research, 38(5),* 993–1007.

Lambert, A.J.D. (2002). Determining optimum disassembly sequences in electronic equipment. *Computers & Industrial Engineering, 43(3),* 553–575.

Li, J.R., Khoo, L.P., & Tor, S.B. (2002). A novel representation scheme for disassembly sequence planning. *International Journal of Advanced Manufacturing Technology, 20(8),* 621–630.

McMullen, P.R. (2001). An ant colony optimization approach to addressing a JIT sequencing problem with multiple objectives. *Artificial Intelligence in Engineering, 15(3),* 309–307.

Seo, K.K., Park, J.H., & Jang, D.S. (2001). Optimal disassembly sequence using genetic algorithms considering economic and environmental aspects. *International Journal of Advanced Manufacturing Technology, 18(5),* 371–380.

Srinivasan, H., Figueroa, R., & Gadh, R. (1999). Selective disassembly for virtual prototyping as applied to de-manufacturing. *Robotics and Computer Integrated Manufacturing, 15(3),* 231–245.

Srinivasan, H., & Gadh, R. (1998). A geometric algorithm for single selective disassembly using the wave propagation abstraction. *Computer Aided Design, 30(8),* 603–613.

Srinivasan, H., & Gadh, R. (2000). Efficient disassembly of multiple components from an assembly using wave propagation. *ASME Journal of Mechanical Design, 122(2),* 179–184.

Srinivasan, H., & Gadh, R. (2002). A non-interfering selective disassembly sequence for components with geometric constraints. *IIE Transactions, 34(4),* 349–361.

Srinivasan, H., Shyamsundar, N., & Gadh, R. (1997). A framework for virtual disassembly analysis. *Journal of Intelligent Manufacturing, 8(4),* 277–295.

T'kindt, V., & Monmarche, N., Tercinet, F., & Laught, D. (2002). An ant colony optimization algorithm to solve a 2-machine bicriteria flowshop scheduling problem. *European Journal of Operational Research, 142(2),* 250–257.

Watkins, R.D., Kleban, S.D., & Luger, G.F. (1995). Expert system support for environmentally conscious manufacturing. *International Journal of Environmentally Conscious Design & Manufacturing, 4(2),* 53–66.

Woo, T.C., & Dutta, D. (1991). Automatic disassembly and total ordering in three dimensions. *ASME Journal of Engineering for Industry, 113(1),* 207–213.

Zeid, I., Gupta, S.M., & Bardasz, T. (1997). A case-based reasoning approach to planning for disassembly. *Journal of Intelligent Manufacturing, 8(2),* 97–106.

Zha, X.F., & Lim, S.Y.E. (2000). Assembly/disassembly task planning and simulation using expert Petri Nets. *International Journal of Production Research, 38(15),* 3639–3676.

Zussman, E., & Zhou, M.C. (1999). A methodology for modeling and adaptive planning of disassembly processes. *IEEE Transactions of Robotics and Automation, 15(1),* 190–194.


---


**J.F. Wang** received a BS in mechanical engineering from Northern Jiaotong University in 1991. He is currently a doctoral candidate in mechanical engineering at Huazhong University of Science & Technology. His research interests include design for assembly and disassembly and soft computing in design.

**J.H. Liu** received a PhD in mechanical engineering from Tokyo Metropolitan University in 1996. He is currently a Professor in the School of Mechanical Engineering and Automation at Beihang University. Dr. Liu's research interests include conceptual design, artificial intelligence in design, and concurrent engineering.

**S.Q. Li** received a PhD in mechanical engineering from Huazhong University of Science & Technology in 1992. He is currently a Professor in the School of Mechanical Science and Engineering at Huazhong University of Science & Technology. Dr. Li's research interests include product virtual design/manufacturing, finite element modeling, analysis, and visualization.

**Y.F. Zhong** received a BS in mechanical engineering from Huazhong University of Science & Technology in 1958. He is currently a Professor in the School of Mechanical Science and Engineering at Huazhong University of Science & Technology. His research interests include optimal design, computer integrated manufacturing, and AI.


https://doi.org/10.1017/S0890060403174045 Published online by Cambridge University Press