A multi-objective disassembly planning approach with ant colony optimization algorithm

C Lu[1]*, H Z Huang[1], J Y H Fuh[2], and **Y S Wong[2]**
[1]School of Mechatronics Engineering, University of Electronic Science and Technology of China, Chengdu, People's Republic of China
[2]Department of Mechanical Engineering, National University of Singapore, Singapore

The manuscript was received on 7 June 2008 and was accepted after revision for publication on 24 July 2008.

DOI: 10.1243/09544054JEM1252

Abstract: This paper proposes a multi-objective disassembly planning approach with an ant colony optimization algorithm. The mechanism of ant colony optimization in disassembly planning is discussed, and the objectives to be optimized in disassembly planning are analysed. In order to allow a more effective search for feasible non-dominated solutions, a multi-objective searching algorithm with uniform design is investigated to guide the ants searching the routes along the uniformly scattered directions towards the Pareto frontier; based on the above searching algorithm, an ant colony optimization algorithm for disassembly planning is developed. The results of a case study are given to verify the proposed approach.

Keywords: disassembly planning, ant colony optimization, multi-objective, uniform design

1 INTRODUCTION

As an important step in the product life cycle, the disassembly process recycles parts from outdated or discarded products for reuse, remanufacturing, or disposal for environmental protection purposes. Disassembly planning focuses on disassembly sequence planning, which aims to achieve a feasible disassembly sequence with minimal cost or time. An effective disassembly planning approach can not only provide a solution for disassembling the product successfully and economically, it can also help the designer to consider product life cycle issues by focusing on the product disassembly cost or time in the early design stage. In recent years, with the requirement for life cycle-based product development, investigations using the effective disassembly planning approach have attracted much research attention. This paper presents a multi-objective disassembly planning approach with an ant colony optimization (ACO) algorithm, to further enhance the performance of current disassembly planning approaches.

The paper is organized as follows: section 2 is a review of current disassembly planning works; section 3 discusses the mechanism of ACO; section 4 proposes a multi-objective searching algorithm with uniform design; based on section 4, section 5 discusses the application of ACO for disassembly planning with multiple search directions; section 6 presents a case study; finally, a conclusion is given in section 7.

Corresponding author: School of Mechatronics Engineering, University of Electronic Science and Technology of China, Chengdu 610054, People's Republic of China. email: conglu@uestc.edu.cn; lucong98@hotmail.com

2 LITERATURE REVIEW

In recent years, much research attention has been focused on disassembly planning, and a variety of approaches have been proposed. Guo *et al.* [**1**] proposed a modularization-based disassembly sequence planning approach to resolve the problem resulting from a large number of parts in a product, where the hierarchy network graph of the product is created, and the precedence constraints related to the hierarchy network graph are used to generate the disassembly sequence. Chung and Peng [**2**] proposed an integrated approach to selective-disassembly sequence planning, to get a partial disassembly sequence where the parts or components are selected for recycling or reuse. This approach can generate a feasible and practical sequence for selective disassembly using two matrices – a subassembly division precedence matrix and a part disassembly route matrix – to ensure both batch disassembly of components and tool accessibility to fasteners. Torres *et al.* [**3**] proposed a method to represent the hierarchical relationships among components and/or assemblies of the product. Based on this representation, an algorithm is established to generate a partial non-destructive disassembly sequence of a product. Das and Naik [**4**] proposed a descriptive model with a structured format for creating, documenting, and evaluating a disassembly process plan. In addition, the model can transmit the product knowledge from the original product manufacturer to the consumer and the end-of-life disassembler via the disassembly bill of materials. Dong *et al.* [**5**] proposed an approach to generate the disassembly sequence from a hierarchical attributed liaison graph representation of an assembly automatically, by decomposing the assembly into subassemblies recursively. The graph is built according to knowledge of engineering, design, and demanufacturing; for each layer of the graph, the preferred subassembly is selected in terms of mobility, stability, and parallelism. Using the graph, the proposed approach can find the most feasible and practical sequence. Veerakamolmal and Gupta [**6**] proposed a case-based reasoning (CBR) approach to disassembly process planning, with a method to initialize a case memory and to operate a CBR system. The approach can derive a feasible disassembly process quickly by retrieval, reuse, and revision of the product disassembly process plan.

The above references present the different disassembly planning approaches that can provide feasible and practical disassembly plans with different areas of focus. However, these approaches do not adopt the optimization search algorithm, so they can not easily find the optimal or near-optimal solutions.

In addition to the above works, other disassembly planning approaches that use some optimization algorithms are discussed as follows. Andres *et al.* [**7**] proposed a two-phase approach to determine the optimal disassembly sequence with the goal of minimizing machine acquisition costs. A metaheuristic algorithm named GRASP is used to search for the disassembly sequence for each product that leads to the minimum number of intercellular movements. Rai *et al.* [**8**] presented a Petri net model to search a partial reachability graph. With the heuristic function, the proposed approach can generate a feasible and optimal disassembly sequence based on the firing sequence of transitions of the Petri net model. In the above two approaches, only one objective – such as the machine acquisition costs – was considered, and the other objectives in the disassembly process were ignored. Because disassembly planning is a typical multi-objective optimization problem, the above approaches are therefore not suitable for finding optimal or near-optimal solutions considering different objectives in the disassembly process.

As an important method, the genetic algorithm has been widely used in assembly planning [**9–12**]; in the mean time, it is also used in disassembly planning to find the optimal disassembly sequence. Kongar and Gupta [**13**] proposed a genetic algorithm approach to disassembly sequence planning, with the objective of minimizing the number of direction changes, disassembly method changes, and the groups of identical material components. Because assembly planning or disassembly planning are highly constrained problems, when using a genetic algorithm-based approach the solution sometimes can not be converged to a global optimal or near-global optimal solution, or even a feasible solution can not be found in an evolution trial due to the precedence constraints when a product is complex and composed of many parts.

Recently, a new probabilistic evolutionary optimization algorithm – ant colony optimization (ACO) – which simulates the cooperative work of an ant colony for searching the shortest path from the nest to the food source, has been given attention and has been used in some engineering optimization problems, such as the just-in-time (JIT) sequencing problem [**14**], job-shop scheduling [**15**], etc. In addition, some new research works applying ACO in assembly and disassembly planning have been reported. Wang *et al.* [**16**] proposed an ACO approach in assembly sequence planning; in this work, only one objective, the number of orientations during the disassembly process, is considered as the heuristic information to guide the ants moving to the next node; how the other objectives in assembly planning affect the route selection of the ants was not investigated. For an ACO approach used in assembly or disassembly planning with multiple objectives, Failli and Dini [**17**] proposed using ACO in assembly planning. In this approach, two items of heuristic information – the number of gripper changes and the number of orientation changes, which are two objectives considered in assembly planning – are used to guide the moving of the ants. The above two pieces of heuristic information are given a constant value in ACO according to gripper change and orientation change, thus the directions for guiding the ants' movements are fixed. McGovern and Gupta [**18**] proposed an approach using ACO for a disassembly line balancing problem. In this approach, several objectives are considered, but only one objective – the measure of balance – is used as the heuristic information for ACO calculations and trail selection; the other objectives are only considered at the end of each cycle to update the best overall solution. In the above-mentioned ACO approaches for assembly or disassembly planning with multiple objectives, the ants select the route by evaluating the heuristic value according to the objectives. Although the above ACO approaches have had some success in assembly or disassembly planning, the approaches fixed the search directions used to guide the ants' movements, therefore more trade-off solutions for multiple objectives could not be easily found. As disassembly planning is a typical multi-objective optimization problem, mechanisms by which the ACO approach can be used in disassembly planning to effectively guide the ants to search and find more trade-off solutions, to provide the decision maker with more choice, need to be further investigated.

3 PRINCIPLE OF ACO

ACO is the simulation of the behaviour of a colony of ants that are working cooperatively to search and find the shortest path from the nest to the food source. As a key factor in the searching process, pheromone is a chemical substance that is deposited by the ants when they move along the path, and it will be used by the ants to exchange the information. The ants prefer to choose the shorter path, the shorter path will attract more ants to visit it, and thereby more pheromone is deposited on the path by the ants. Meanwhile, the pheromone on all paths is decreased through evaporation due to the time that has passed. The probability that subsequent ants choose the path is based on the amount of pheromone deposited on the path; therefore, the shorter path with a greater amount of pheromone will have more chance of being selected and thus attract more and more ants to visit it subsequently. As a result, the shortest path from the nest to the food source can be found by the ant colony.

In ACO, the probability that ant z will select the next node j is given as

$$P_z(i,j) = \begin{cases} \dfrac{\tau(i,j)[\eta(i,j)]^{\lambda}}{\sum\limits_{s\in \mathrm{Allowed}_z(i)} \tau(i,s)[\eta(i,s)]^{\lambda}}, & \text{if } j \in \mathrm{Allowed}_z(i) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where, $\tau(i, j)$ is the quantity of pheromone deposited on the edge from node i to node j. $\eta(i, j)$ is the heuristic information corresponding to the edge from node i to node j. λ is the parameter that determines the relative importance of $\tau(i, j)$ versus $\eta(i, j)$. $\mathrm{Allowed}_z(i)$ are the nodes that are allowed to be selected by ant z when choosing the next node j. In ACO, the edges with greater $\tau(i, j)$ and $\eta(i, j)$ values are the favourable edges that the ants prefer to choose.

During the search process of ACO, there are two important rules for updating the pheromone: the local updating rule and the global updating rule.

Local updating rule

The local updating rule is used for updating the pheromone level of the edge only when the ants visit it, and it can be represented by

$$\tau(i,j) = (1-\alpha)\tau(i,j) + \alpha\tau_0(i,j) \tag{2}$$

where α is a parameter given in the range [0, 1], which determines the pheromone volatility on the edge from node i to node j, and $\tau_0(i, j)$ is the initial pheromone level on the edge. Through local updating, the visited edges will loss some amount of their pheromone, and this can effectively avoid premature convergence.

Global updating rule

The global updating rule is used for updating the pheromone level of all the edges after the ants have finished the tour, and only the edges belonging to the current global best solution can have extra pheromone added. Meanwhile, the evaporation of the pheromone is performed on all the edges. The global updating rule can be represented by

$$\tau(i,j) = (1-\beta)\tau(i,j) + \beta\Delta\tau(i,j) \tag{3}$$

where β is the pheromone decay parameter given in the range [0, 1]

$$\Delta\tau(i,j) = \begin{cases} F_{(\mathrm{gb})}, & \text{if edge } (i,j) \in \text{global best solution} \\ 0, & \text{otherwise} \end{cases}$$

$F_{(\mathrm{gb})}$ is the fitness value of the global best solution found up to now, and the detailed value of $F_{(\mathrm{gb})}$ in disassembly planning will be given in section 5.

4 MULTI-OBJECTIVE SEARCH DIRECTIONS WITH UNIFORM DESIGN

In order to apply ACO to deal with the multi-objective optimization problem in disassembly planning, this section proposes an algorithm for building the uniformly scattered searching directions towards the Pareto frontier, aiming at finding more non-dominated solutions along the Pareto frontier.

4.1 Non-dominated solutions in the multi-objective optimization problem

For a multi-objective optimization problem, because different objectives are usually conflicting, there exists a set of solutions in the solution space, in which no solution is superior to the others according to each objective. These solutions are usually called non-dominated solutions, which can be regarded as the best trade-off solutions in the multi-objective optimization problem.

The definition of a non-dominated solution can be given as follows. Given a multi-objective optimization problem with n objectives to be minimized: minimize $f_1(x)$, $f_2(x)$,..., $f_n(x)$, $X \in \Omega$; where $f_i(x)$ represents the different objectives, $i \in \{1, 2, \ldots, n\}$, and Ω represents the feasible solution space. For two solutions X_1, X_2, if

$$\begin{cases} f_t(x_1) < f_t(x_2), & \text{for some } t \in \{1, 2, \ldots, n\} \\ f_t(x_1) \leqslant f_t(x_2), & \text{for all } t \in \{1, 2, \ldots, n\} \end{cases}$$

then solution X_2 is dominated by solution X_1. In the feasible solution space Ω, if there does not exist any solution that can dominate solution X, then solution X is called a non-dominated solution.

In the multi-objective optimization problem, a set of non-dominated solutions form the Pareto frontier. An example is shown in Fig. 1, where the filled circles represent the non-dominated solutions that form the Pareto frontier and the open circles represent the dominated solutions. This is a two-objective optimization problem, with the goal of minimizing those two objectives, i.e. to search for the non-dominated solutions located along the Pareto frontier.

4.2 Uniform design for building multiple search directions

In a multi-objective optimization problem, in order to find more non-dominated solutions for the decision maker to choose from, the search directions

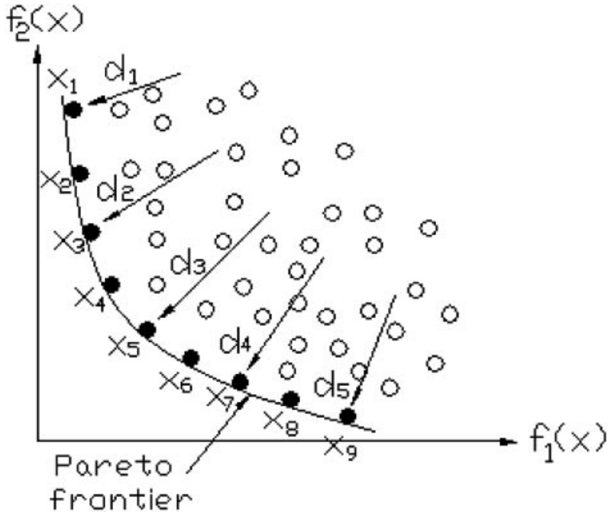

Fig. 1 Non-dominated solutions with multiple search directions

towards the Pareto frontier need to be expanded effectively. In this work, an experimental design method called 'uniform design' is used to expand the search directions.

Uniform design can be used to sample a small set of points from a given large set of points, so as to make the sampled points uniformly scattered over the space of all the given points. The uniform design method can be described as follows. Suppose there are n factors, and each factor has k levels, then there are a total of k^ncombinations. From the above combinations, to select k combinations that are uniformly scattered over the space, a uniform matrix can be given as follows

$$\mathbf{U}(n,k) = [U_{i,j}]_{k\times n} = \begin{bmatrix} U_{11} & U_{12} & \ldots & U_{1n} \\ U_{21} & U_{22} & \ldots & U_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ U_{k1} & U_{k2} & \ldots & U_{kn} \end{bmatrix} \tag{4}$$

In equation (4), $U_{i,j}$ is the level of the factor j in the ith combination. When k is prime and $k > n$, then $U_{i,j}$ can be concluded as follows [**19**]

$$U_{i,j} = (i\sigma^{j-1} \text{mod} k) + 1 \tag{5}$$

where σ is a parameter as shown in Table 1, and it is determined by the number of factors and the number of levels per factor.

For a multi-objective optimization problem, in order to get a set of search directions that are uniformly scattered towards the Pareto frontier in the solution space, the following equation can be used to conclude the weight vectors that determine the above search directions.

$$W_{ij} = \frac{U_{ij}}{\sum_{j=1}^{j=n} U_{ij}} \quad i \in [1,k],\ j \in [1,n] \tag{6}$$

In equation (6), n can be regarded as the number of objective functions, and k as the number of search directions. Then, the weight matrix

$$[W_{i,j}]_{k\times n} = \begin{bmatrix} W_{11} & W_{12} & \ldots & W_{1n} \\ W_{21} & W_{22} & \ldots & W_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ W_{k1} & W_{k2} & \ldots & W_{kn} \end{bmatrix} \tag{7}$$

Each row of the above matrix is a weight vector to be used for building the fitness function. There are a total of k weight vectors, for each weight vector the sum of the weights is equal to one. Using the weight vectors concluded from equation (6), the fitness functions with k uniformly scattered search directions can be built. In this work, uniform design will be used to generate the weight vectors to guide the search directions of the ant colony, as will be discussed in section 5.3.

Table 1 Values of σ for different numbers of levels per factor and different numbers of factors

No. of levels per factor	No. of factors	σ
5	2–4	2
7	2–6	3
11	2–10	7
13	2	5
	3	4
	4–12	6
17	2–16	10
19	2–3	8
	4–18	14

5 APPLICATION OF ACO FOR DISASSEMBLY PLANNING

In this section, the application of ACO with multiple search directions for disassembly planning is discussed.

5.1 Geometric precedence feasibility in disassembly planning

In disassembly planning, the geometric precedence feasibility is a constraint that the ants must satisfy during the moving process. This means only the parts that can be disassembled without any interference can be chosen by the ants in the next step. In order to conclude the geometric precedence feasibility, the interference matrix is used in this work.

The interference matrix was first proposed by Dini and Santochi [**20**] in assembly planning, and it can also be used for precedence feasibility judgment in disassembly planning. For an assembly consisting of n parts, an interference matrix $\mathbf{I}_d$ (d represents the disassembly direction) can be represented as

$$\mathbf{I}_d = \begin{array}{c} \\ P_1 \\ P_2 \\ \cdots \\ P_n \end{array} \begin{array}{c} \begin{array}{cccc} P_1 & P_2 & \ldots & P_n \end{array} \\ \begin{bmatrix} P_{11} & P_{12} & \ldots & P_{1n} \\ P_{21} & P_{22} & \ldots & P_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ P_{n1} & P_{n2} & \ldots & P_{nn} \end{bmatrix} \end{array}$$

$P_1, \ldots, P_n$ are used to represent the n parts in the assembly, let $P_{ij} = 1$ ($i \in [1, n]$, $j \in [1, n]$) if part P_i collides with P_j when P_i is disassembled along the direction d from the current assembly position; otherwise, let $P_{ij} = 0$. Let $P_{ii} = 0$ because the part cannot collide with itself. Because P_{ij} in the $-d$ direction is equal to P_{ji} in the $+d$ direction, three interference matrices $\mathbf{I}_{+X}$, $\mathbf{I}_{+Y}$, and $\mathbf{I}_{+Z}$ can be used to conclude the precedence feasibility in a disassembly sequence: (A Cartesian coordinate system whereby the six axes $\pm X$, $\pm Y$, $\pm Z$ are the principal axes along which the components are disassembled is used in this work.)

In the disassembly process, when part P_i is disassembled before a remaining product assembly S_m consisting of m parts, then the feasible disassembly direction of P_i to S_m can be derived as follows: for disassembly direction d, $d \in \{\pm X, \pm Y, \pm Z\}$, let $P_j \in S_m$, determine $D_d(P_i S_m) = \sum P_{ij}$ (P_{ij} is the element in $\mathbf{I}_d$). If $D_d(P_i S_m) = 0$, then direction d is the feasible disassembly direction of P_i to S_m; otherwise, direction d is infeasible. If none of the six directions is feasible, then P_i can not be disassembled at the current stage; otherwise, P_i can be disassembled from the product without collision interference.

5.2 Three objectives in disassembly planning

The purpose of disassembly planning is to derive a feasible disassembly sequence with the minimal disassembly cost or disassembly time. The disassembly cost or time can usually be determined by three objectives: the number of disassembly orientation changes, tool (gripper) changes, and changes in disassembly operation types. In the disassembly process, a change of disassembly orientation or disassembly tool needs time and usually increases the disassembly cost. Different types of assembly operations are needed to complete the assembly process – such as pressing, screwing, riveting, etc.; accordingly, different disassembly operations are needed for different parts in the disassembly process. Changes of disassembly operation also require tool changes, and thus increase the disassembly time and cost. Hence, in disassembly planning, the above three objectives – disassembly orientation changes, tool changes, and changes in disassembly operation types – should be minimized to reduce the disassembly time and cost.

5.3 Application of ACO with multiple search directions for disassembly planning

In order to apply ACO in disassembly planning, the first part in the disassembly sequence can be regarded as the nest of the ant colony, and the last part in the disassembly sequence can be regarded as the food source. The shortest path can be equivalent to the disassembly sequence with the minimal cost or time; thus in this work, the shortest path can be represented by the optimum disassembly sequence considering three objectives: disassembly orientation change, disassembly tool change, and disassembly operation change.

In the disassembly planning problem, $P_z(i, j)$ in equation (1) can be modified and represented as the probability that ant z selects the disassembly sequence step from part i to part j in a given search direction t, and it can be represented by

$$P_{z(t)}(i,j) = \begin{cases} \dfrac{\tau_t(i,j)[\eta_t(i,j)]^\lambda}{\sum\limits_{s\in \text{Allowed}_z(i)} \tau_t(i,s)[\eta_t(i,s)]^\lambda}, & \text{if } j \in \text{Allowed}_z(i) \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where: $t \in [1, k]$; $\tau_t(i, j)$ is the quantity of pheromone deposited on the disassembly sequence step from part i to part j in search direction t; $\eta_t(i, j)$ is the heuristic value corresponding to the disassembly step from part i to part j in search direction t, and it can be represented as

$$\left.\begin{aligned} \eta_1(i,j) &= 1.5 - (W_{11}f_1 + W_{12}f_2 + W_{13}f_3) \\ \eta_2(i,j) &= 1.5 - (W_{21}f_1 + W_{22}f_2 + W_{23}f_3) \\ &\cdots \\ \eta_k(i,j) &= 1.5 - (W_{k1}f_1 + W_{k2}f_2 + W_{k3}f_3) \end{aligned}\right\} \tag{9}$$

where, f_1, f_2, and f_3 are given as

$$f_1 = \begin{cases} 1, & \text{if need orientation change in disassembly step from part } i \text{ to part } j \\ 0, & \text{if no orientation change in disassembly step from part } i \text{ to part } j \end{cases}$$

$$f_2 = \begin{cases} 1, & \text{if need tool change in disassembly step from part } i \text{ to part } j \\ 0, & \text{if no tool change in disassembly step from part } i \text{ to part } j \end{cases}$$

$$f_3 = \begin{cases} 1, & \text{if need operation change in disassembly step from part } i \text{ to part } j \\ 0, & \text{if no operation change in disassembly step from part } i \text{ to part } j \end{cases}$$

In equation (9), $[W_{i,j}]_{k\times 3}$ is the weight matrix derived from equation (6), which is used for three objectives: disassembly orientation change, disassembly tool change, and disassembly operation change. Thus, $\eta_t(i,j)$ ($t \in [1, k]$) can be used for guiding the ants to search the next disassembly sequence step along k different directions which are uniformly scattered towards the Pareto frontier, as mentioned in section 4.2.

In disassembly planning, for k different search directions, the local updating function $\tau_t(i, j)$ can be represented as

$$\tau_t(i,j) = (1-\alpha)\tau_t(i,j) + \alpha\tau_0(i,j), \qquad t \in [1,k] \tag{10}$$

For k different search directions, the global updating function in ACO can be represented as

$$\tau_t(i,j) = (1-\beta)\tau_t(i,j) + \beta\Delta\tau_t(i,j), \qquad t \in [1,k] \tag{11}$$

where

$$\Delta\tau_t(i,j) = \begin{cases} F_{t(gb)}, & \text{if step } (i,j) \in \text{global best disassembly sequence} \\ 0, & \text{otherwise} \end{cases}$$

$$F_{t(gb)} = Z/(1 + W_{t1}N_1 + W_{t2}N_2 + W_{t3}N_3)$$

$$t \in [1,k]$$

Z = constant parameter used to adjust the added pheromone level in the step (i, j)

N_1, N_2, N_3 = number of orientation changes, number of tool changes, and number of disassembly operation changes in current global best disassembly sequence respectively.

After local updating and global updating of the pheromone, $\tau_t(i, j)$ is the quantity of pheromone deposited on the disassembly sequence step from part i to part j for the search direction t ($t \in [1, k]$).

From the above, it can be seen that for different search directions, the selection probability that ant z selects the disassembly sequence step from part i to part j could be different due to the quantity of pheromone deposited and the heuristic value. The overall ACO algorithm with multiple search directions for disassembly planning is proposed as follows.

Algorithm: overall ACO algorithm for disassembly planning

Step 1. Set the number of factors (objectives) n, and set the number of levels of each factor (search directions) k; derive the parameter σ from Table 1.

Step 2. Conclude the weight matrix using equations (4), (5), and (6).

Step 3. For the assembly consisting of m parts, place k ants on each of the q parts that can be initially disassembled.

Step 4. Set initial quantity of pheromone on each disassembly step as $\tau_t(i, j) = c$.

Step 5. Set the maximal cycle number $N_{c\,(max)}$, and let the cycle number $N_c = 1$.

Step 6. For search direction t ($t \in [1, k]$), let $t = 1$.

Step 7. For the ant z that is searching the route along the direction t, if the ant z has not completed the visit from the first part to the last one, calculate the selection probability $P_{z(t)}(i, j)$ using equation (8), where part j belongs to the remaining parts in the product that have a feasible disassembly direction at this stage.

Step 8. Select the part j as the next part to be disassembled using the roulette-wheel selection method.

Step 9. Move the ant z to the new position – part j.

Step 10. Locally update the pheromone level on the disassembly sequence step from part i to part j using equation (10).

Step 11. If ant z has completed the visit from the first part to the last one, go to step 12; else, go to step 7.

Step 12. Globally update the pheromone level on the best disassembly sequence found so far using equation (11).

Step 13. Let $t = t + 1$, if $t < k$, go to step 7; else, go to step 14.

Step 14. Let $N_c = N_c + 1$, if $N_c < N_{c(max)}$, go to step 6; else, go to step 15.

Step 15. Output the non-dominated solutions found by the ants.

6 CASE STUDY AND DISCUSSION

The proposed disassembly planning approach with the ACO algorithm has been implemented using Visual C++ 6.0. In this section, an assembly product [**12**] (shown in Fig. 2) is used to validate the proposed approaches.

6.1 Case study

In this case, there are three objectives to be optimized, and the number of search directions k is set as five. The parameter σ can then be derived from Table 1 as $\sigma = 2$. From equations (4) and (5), the uniform matrix $\mathbf{U}(3, 5)$ can be derived as follows

$$\mathbf{U}(3,5) = [U_{i,j}]_{5\times 3} = \begin{bmatrix} 2 & 3 & 5 \\ 3 & 5 & 4 \\ 4 & 2 & 3 \\ 5 & 4 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

From equations (6) and (7), the weight matrix can be derived as

$$[W_{i,j}]_{5\times 3} = \begin{bmatrix} 1/5 & 3/10 & 1/2 \\ 1/4 & 5/12 & 1/3 \\ 4/9 & 2/9 & 3/9 \\ 5/11 & 4/11 & 2/11 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

For seven parts that can be initially disassembled in this case, five ants are placed on each part, and each ant will search the route along one of five directions respectively. Based on some reference works [**16**, **17**], some parameters for the ACO algorithm are set as follows: the initial quantity of pheromone on each disassembly step is set as $\tau_0 = 0.5$; the pheromone decay parameters α and β are set as 0.1; the parameter λ is set as 0.8. Through the experiment in the case study, the maximum cycle number $N_{c(max)}$ is set as 500, and the constant parameter used to adjust the added pheromone level Z is set as 3.

In this case, parts 2, 3, and 15 have similar geometric shapes and dimensions, so they can be grasped with the same tool-chuck in the disassembly process, and this tool is assigned with the number 2 in this case. Similarly, the other parts can be grouped

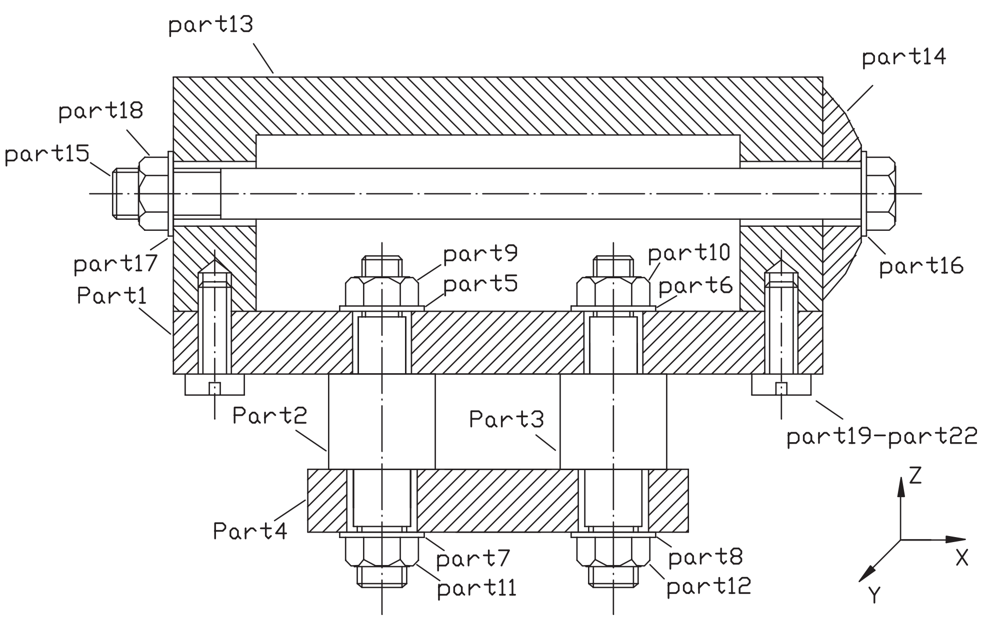

Fig. 2 An assembly consisting of 22 parts

according to their geometric shape, dimension, and weight, and can be grasped with different tools with different tool numbers, as shown in Table 2. For the operation type, parts 19, 20, 21, and 22 can be unscrewed with the screwdriver in the disassembly process, so these four parts are assigned with the same operation type (number 2) in this case; similarly, parts 9, 10, 11, 12, and 18 can be unscrewed with the wrench (operation type number 1) in this case; the other parts do not need any tool to unfasten them in the disassembly process, so they are assigned with operation type number 0 in this case, as shown in Table 2.

6.1.1 *Test 1*

In test 1, the evolution test with 5 uniformly scattered search directions is carried out 20 times, and the result is shown in Table 3. All the 20 trials are converged to the feasible disassembly sequences, during which, 4 trials get 2 non-dominated solutions, 8 trials get 3 non-dominated solutions, and 8 trials get 4 non-dominated solutions. Of the above test results, 4 non-dominated solutions found in the trial are shown in Table 4. In the above non-dominated solutions, the disassembly sequence of non-dominated solution number 4 is given as

18–20-22–21–19–12–11–15–13–9–10–6–5–1–14–16
–17–8–7–4–2–3

the sequence started from part 18, with the search direction ($W_1 = 1/4$, $W_2 = 5/12$, $W_3 = 1/3$), and it has 4 orientation changes, 8 tool changes, and 5 operation changes.

For other non-dominated solutions, non-dominated solution number 1 has 4 orientation changes, 7 tool changes, and 6 operation changes, with the search direction ($W_1 = 4/9$, $W_2 = 2/9$, $W_3 = 3/9$); non-dominated solution number 2 has 3 orientation changes, 8 tool changes, and 7 operation changes, with the search direction ($W_1 = 1/5$, $W_2 = 3/10$, $W_3 = 1/2$); non-dominated solution number 3 has 2 orientation changes, 9 tool changes, and 5 operation changes, with the search direction ($W_1 = 5/11$, $W_2 = 4/11$, $W_3 = 2/11$).

In order to evaluate the evolution performance for 500 generations, in this case the equation $F = 3/(1 + W_1N_1 + W_2N_2 + W_3N_3)$ is used to record the fitness value of the sequence, where N_1, N_1, and N_3 are the number of orientation changes, the number of tool changes, and the number of operation changes respectively, and W_1, W_2, and W_3 are the weight for each of above three objectives respectively. The evolution performance for 500 generations of the sequence in the search direction ($W_1 = 1/4$, $W_2 = 5/12$, $W_3 = 1/3$) is shown in Fig. 3.

6.1.2 *Test 2*

For comparison with test 1, only one fixed search direction ($W_1 = 1/4$, $W_2 = 5/12$, $W_3 = 1/3$) is used in test 2 to guide the ants to search the route. With the

Table 3 Twenty trial results in test 1

Total trials	Trials that get two non-dominated solutions	Trials that get three non-dominated solutions	Trials that get four non-dominated solutions
20	4	8	8

Table 4 Test results of a trial in test 1

Non-dominated solution number	Orientation changes	Tool changes	Operation changes
1	4	7	6
2	3	8	7
3	2	9	5
4	4	8	5

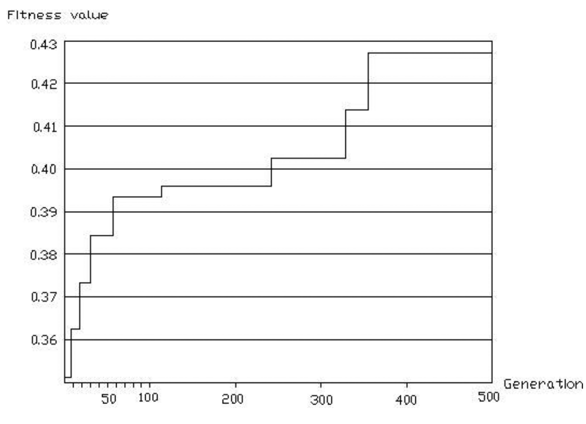

Fig. 3 The evolution performance for 500 generations

Table 2 Tool type and operation type of each part in the assembly

Part number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Tool type	1	2	2	1	3	3	3	3	4	4	4	4	1	5	2	3	3	4	6	6	6	6
Operation type	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1	2	2	2	2

Table 5 Twenty trial results in test 2

Total trials	Trials that get one non-dominated solution	Trials that get two non-dominated solution	Trials that get three non-dominated solutions
20	11	8	1

same settings for the other parameters, the evolution result is shown in Table 5. All the 20 trials are converged to the feasible disassembly sequences, during which 11 trials get 1 non-dominated solution, 8 trials get 2 non-dominated solutions, and 1 trial gets 3 non-dominated solutions.

6.2 Discussion

The above two evolution test results show, compared with the ant colony algorithm with only one search direction, that the ant colony algorithm with multiple uniformly scattered search directions can make it easier to find more non-dominated solutions in one trial. This is probably due to the fact that in the latter algorithm, at each step, the ants are guided along the uniformly scattered search directions towards the Pareto frontier, so the ants have more chance to find more non-dominated solutions located in the Pareto frontier.

Because the assembly sequence can be concluded by reversing the disassembly sequence, the proposed approach can also be used to derive the assembly sequence. Compared with the assembly planning approach using a multi-objective genetic algorithm in the same case study [**12**], the approach using the ant colony algorithm is more stable and faster; all of the 20 trials found the feasible disassembly sequence, and the average run time was 6–8 s to converge to a global optimal or near-global optimal sequence. However, for the 20 trials using the assembly planning approach with a multi-objective genetic algorithm, at least 2 trials could not find the feasible assembly sequence, and the average run time was 20–25 s to converge to a global optimal or near-global optimal sequence. This difference could be analysed as follows: in the disassembly planning approach with the ant colony algorithm, the ants search the route step by step, and only the dismountable part can be selected by the ants, so it can easily avoid the infeasible solution, and this can also help find the feasible solution quickly. However, in the assembly planning approach with the genetic algorithm, the initial solutions are randomly generated as a whole sequence. There could be much assembly interference due to precedence constraints and the solutions are evolved as a whole sequence by genetic operators in the later stages; this could cost much time to repair and evolve the solution to a feasible and optimal solution. Sometimes the solutions can not be evolved to feasible solutions due to the highly constrained combinatory nature of this problem. Therefore, from the above analysis, it can be seen that the disassembly planning approach using the ant colony algorithm could be more efficient than the approach using the genetic algorithm.

7 CONCLUSION

This paper presents a multi-objective disassembly planning approach using the ant colony optimization algorithm. Three objectives in the disassembly process are optimized concurrently to get the optimal or near-optimal disassembly sequence in this work. In order to guide the ants to search comprehensively and find more feasible non-dominated solutions for decision making, uniform design is used for establishing a multi-objective searching algorithm, and an ant colony optimization algorithm for disassembly planning is developed based on the above searching algorithm. Through the case study and the comparison with the approach using a genetic algorithm, it can be verified that the proposed multi-objective disassembly planning approach with the ACO algorithm is more stable, faster, and more efficient for finding more feasible non-dominated solutions.

ACKNOWLEDGEMENT

This work is supported by the Youth Science Foundation of the University of Electronic Science and Technology of China (Grant JX0764), and the authors are grateful for this support.

REFERENCES

1 **Guo, W. X., Liu, Z. F., Liu, G. F., Pan, X. Y.,** and **Huang, H. H.** Disassembly sequence planning based on modularization. *J. Comput. Aided Des. Comput. Graphics*, 2005, **17**(3), 498–504.
2 **Chung, C.** and **Peng, Q. J.** An integrated approach to selective-disassembly sequence planning. *Robotics Comput. Integrated Mfg*, 2005, **21**(4–5), 475–485.
3 **Torres, F., Puente, S. T.,** and **Aracil, R.** Disassembly planning based on precedence relations among assemblies. *Int. J. Advd Mfg Technol.*, 2003, **21**(5), 317–327.
4 **Das, S. K.** and **Naik, S.** Process planning for product disassembly. *Int. J. Prod. Res.*, 2002, **40**(6), 1335–1355.
5 **Dong, T. Y., Zhang, L., Tong, R. F.,** and **Dong, J. X.** A hierarchical approach to disassembly sequence planning for mechanical product. *Int. J. Advd Mfg Technol.*, 2006, **30**(5–6), 507–520.

6 **Veerakamolmal, P.** and **Gupta, S. M.** A case-based reasoning approach for automating disassembly process planning. *J. Intell. Mfg*, 2002, **13**(1), 47–60.

7 **Andres, C., Lozano, S.,** and **Adenso, D. B.** Disassembly sequence planning in a disassembly cell context. *Robotics Comput. Integrated Mfg*, 2007, **23**(6), 690–695.

8 **Rai, R., Rai, V., Tiwari, M. K.,** and **Allada, V.** Disassembly sequence generation: a Petri net based heuristic approach. *Int. J. Prod. Res.*, 2002, **40**(13), 3183–3198.

9 **Smith, G. C.** and **Smith, S. S. F.** An enhanced genetic algorithm for automated assembly planning. *Robotics Comput. Integrated Mfg*, 2002, **18**(5–6), 355–364.

10 **Lazzerini, B.** and **Marcelloni, F.** A genetic algorithm for generating optimal assembly plans. *Artif. Intell. Engng*, 2000, **14**, 319–329.

11 **Chen, S. F.** and **Liu, Y. J.** An adaptive genetic assembly-sequence planner. *Int. J. Comput. Integrated Mfg*, 2001, **14**(5), 489–500.

12 **Lu, C., Fuh, J. Y. H.,** and **Wong, Y. S.** An enhanced assembly planning approach using a multi-objective genetic algorithm. *Proc. Instn Mech. Engrs, Part B: J. Engineering Manufacture*, 2006, **220**(2), 255–272.

13 **Kongar, E.** and **Gupta, S. M.** Disassembly sequencing using genetic algorithm. *Int. J. Advd Mfg Technol.*, 2006, **30**(5–6), 497–506.

14 **McMullen, P. R.** An ant colony optimization approach to addressing JIT sequencing problem with multiple objectives. *Artif. Intell. Engng*, 2001, **15**, 309–317.

15 **Rossi, A.** and **Dini, G.** Flexible job-shop scheduling with routing flexibility and separable setup times using ant colony optimisation method. *Robotics Comput. Integrated Mfg*, 2007, **23**(5), 503–516.

16 **Wang, J. F., Liu, J. H.,** and **Zhong, Y. F.** A novel ant colony algorithm for assembly sequence planning. *Int. J. Advd Mfg Technol.*, 2005, **25**(11–12), 1137–1143.

17 **Failli, F.** and **Dini, G.** Ant colony systems in assembly planning: a new approach to sequence detection and optimization. In Proceedings of the 2nd CIRP International Seminar on *Intelligent computation in manufacturing engineering*, 2000, pp. 227–232.

18 **McGovern, S. M.** and **Gupta, S. M.** Ant colony optimization for disassembly sequencing with multiple objectives. *Int. J. Advd Mfg Technol.*, 2006, **30**(5–6), 481–496.

19 **Fang, K. T.** and **Wang, Y.** *Number-theoretic methods in statistics*, 1994 (Chapman & Hall, London).

20 **Dini, G.** and **Santochi, M.** Automated sequencing and subassembly detection in assembly planning. *Ann. CIRP*, 1992, **41**(1), 1–4.

APPENDIX

Notation

$D_d(P_i S_m)$	represented as $\sum P_{ij}$, $P_j \in S_m$
$\mathbf{I}_d$	interference matrix for assembly direction d
N_1	number of orientation changes
N_2	number of tool changes
N_3	number of disassembly operation changes
N_c	cycle number
$N_{c(max)}$	maximal cycle number
$P_z(i, j)$	probability that ant z selects next node j
$P_{z(t)}(i, j)$	probability that ant z selects the disassembly sequence step from part i to part j in a given search direction t
$U_{i,j}$	level of the factor j in the ith combination
Z	constant parameter used to adjust the added pheromone level in the step (i, j)
α	parameter that determines the pheromone volatility on the edge from node i to node j
β	pheromone decay parameter
$\eta(i, j)$	heuristic information corresponding to the edge from node i to node j
$\eta_t(i, j)$	heuristic value corresponding to the disassembly step from part i to part j in search direction t
λ	parameter that determines the relative importance of $\tau(i, j)$ versus $\eta(i, j)$
$\tau(i, j)$	quantity of pheromone deposited on the edge from node i to node j
$\tau_t(i, j)$	quantity of pheromone deposited on the disassembly sequence step from part i to part j in search direction t
$\tau_0(i, j)$	initial pheromone level on the edge