Ant Colony Optimization in Green Manufacturing

Cong Lu
School of Mechatronics Engineering,
University of Electronic Science and Technology of China, Chengdu 611731,
China

1. Introduction

In recent years, with more and more requirement on energy sources saving and environmental protection, green manufacturing has become an important approach in production. In green manufacturing, product disassembly process is a key step, by which the parts can be recycled from the outdated or discarded products for reuse, remanufacturing or disposing, thus can save the energy sources and protect the environment efficiently. To find an effective disassembly process, disassembly planning is usually carried out, which aims at a feasible and optimal disassembly sequence with minimal cost or time. An effective disassembly planning approach can not only provide a solution to disassemble the product successfully and economically, it can also help the designer to consider the 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 green manufacturing technology, investigation on effective disassembly planning approach has attracted much research attention and a variety of approaches have been proposed. Guo et al. [1] proposed a modularization based disassembly sequence planning approach to resolve the problem resulted from a large number of parts in the product, where the Hierarchy Network Graph of product is created, and the precedence constraints related to the hierarchy network graph is 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 by two matrix- subassembly division precedence matrix and 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. And 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 the knowledge in engineering, design and demanufacturing, for each layer of the graph, the preferred subassembly is selected in terms of mobility, stability, and parallelism. With the graph, the proposed approach can find the feasible and practical sequence. Veerakamolmal and Gupta [6] proposed a case-based reasoning 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 retrieve, reuse, and revise the product disassembly process plan.

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

Besides the above works, the other disassembly planning approaches with some optimization algorithm 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 disassembly process were ignored. Because disassembly planning is a typical multi-objective optimization problem, so the above approaches are not suitable to find the optimal or near optimal solutions considering different objectives in disassembly process.

As an important method, the genetic algorithm (GA) 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 GA approach to disassembly sequence planning, with the objective to minimize the number of direction changes, disassembly method changes, and the group of the identical material components. Because assembly planning or disassembly planning are highly constrained problem, using GA-based approach, sometimes the solution can not be converged to a global optimal or near global optimal solution, or even a feasible solution cannot 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 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 JIT sequencing problem [14], job-shop scheduling [15], etc. Also, 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 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 the 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 heuristic information -number of gripper changes and number of orientation changes which are two objectives considered in assembly planning are used to guide the moving of the ants. The above two heuristic information are given the constant value in ACO according to gripper change and orientation change, thus the directions for guiding the ants moving are fixed. McGovern and Gupta [18] proposed an approach using ACO for 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 made some success in assembly or disassembly planning, however, these approaches fixed search directions that are used to guide the ants moving, so more trade off solutions for multiple objectives could not be easily found. As disassembly planning is a typical multi-objective optimization problem, how 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 more choice to achieve green manufacturing needs to be further investigated.

2. Principle of ant colony optimization

ACO is the behavior simulation 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 for the ants to exchange the information. The ants prefer to choose the shorter path, the shorter path will attract more ants to visit, 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 past. The probability that the subsequent ants choose the path is based on the amount of the pheromone deposited on the path, so, the shorter path with greater amount of pheromone will get more chance to be selected and thus attract more and more ants to visit later. As a result, the shortest path from the nest to the food source can be found by the ant colony.

In ant colony optimization, the probability that ant z select next node *j* is given as follows:

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

Where, τ(*i,j*) is the quantity of pheromone deposited on the edge from node *i* to node *j*. η(*i,j*) is the heuristic information corresponding to the edge from node *i* to node *j*. λ is the parameter that determine the relative importance of τ(*i,j*) versus η(*i,j*). Allowed$_z$(*i*) are the nodes that are allowed to be selected by ant *z* when choosing next node *j*. In ACO, the edges with greater τ(*i,j*) and η(*i,j*) are the favorable edges for the ants prefer to choose.

During the search process of ACO, there are two important rules for updating the pheromone -Local Updating Rule and Global Updating Rule.

Local Updating Rule:

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 the Eq. (2)

$$\tau(i,j)=(1-\alpha)\ \tau(i,j)+\ \alpha\tau_0(i,j) \quad (2)$$

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 its pheromone, and this can effectively avoid the premature convergence.

Global updating rule:

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 the Eq. (3)

$$\tau(i,j)=(1-\beta)\ \tau(i,j)+\ \beta\Delta\tau(i,j) \quad (3)$$

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

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

$F_{(\mathrm{gb})}$ 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.

3. 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 Pareto frontier, aiming at finding more non-dominated solutions along Pareto frontier.

3.1 Non-dominated solutions in 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 none of them 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 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, $i\in\{1, 2, \ldots\ldots, n\}$, and Ω represents the feasible solution space. For two solutions X_1, X_2, if

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

then solution X_2 is dominated by solution X_1. In the feasible solution space Ω, if there does not exist any solution which can dominate solution X, then solution X is called as 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 [19], where the solid circles represent the non-dominated solutions which form the Pareto frontier, while the hollow circles represent the dominated solutions. This is a two-objective optimization problem, with the goal to minimize those two objectives, i.e. to search for the non-dominated solutions located along the Pareto frontier.

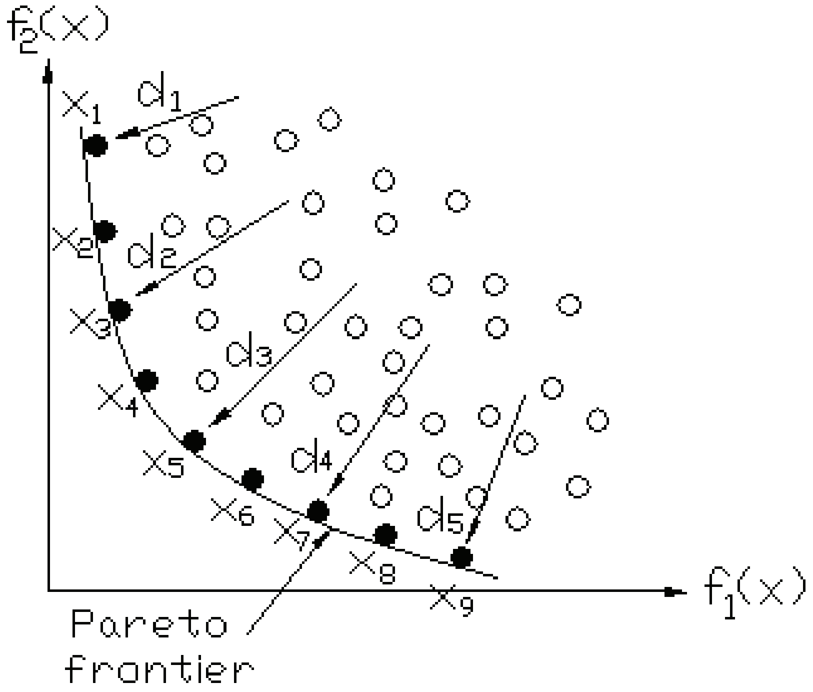


Fig. 1. Non-dominated solutions with multiple search directions

3.2 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 make choice, the search directions towards 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 totally k^n combinations. From the above combinations, to select k combinations that are uniformly scattered over the space, a uniform matrix can be given as follows:

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

In Eq. (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 [20]:

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

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.

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

Table 1. Values of σ for different No. of levels per factors and different No. of factors

For a multi-objective optimization problem, in order to get a set of search directions that are uniformly scattered toward Pareto frontier in the solution space, Eq.(6) can be used to conclude the weight vectors that determine the above search directions. In Eq. (6), n can be regarded as the number of objective functions, and k as the number of the search directions.

$$W_{ij} = \frac{U_{ij}}{\sum_{j=1}^{j=n} U_{ij}} \quad \text{where, } i \ \ [1,k], \text{j} \ \ [1,n] \qquad (6)$$

Then, the weight matrix

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

Each row of the above matrix is a weight vector to be used for building the fitness function. There are totally 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, which will be discussed in section 5.3.

4. Application of ACO for disassembly planning

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

4.1 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 which can be disassembled without any interference can be chosen by the ants in the next step. To conclude the geometric precedence feasibility, the interference matrix is used in this work.

The interference matrix was first proposed by Dini [21] 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 I_d (d represents the disassembly direction) can be represented as follows:

$$I_d = \begin{array}{c|cccc} & P_1 & P_2 & \cdots & P_n \\ P_1 & P_{11} & P_{12} & \cdots & P_{1n} \\ P_2 & P_{21} & P_{22} & \cdots & P_{2n} \\ \cdots & \multicolumn{4}{c}{\cdots\cdots\cdots\cdots\cdots\cdots} \\ P_n & P_{n1} & P_{n2} & \cdots & P_{nn} \end{array}$$

P_1, ….., P_n are used to represent the n parts in the assembly, let $P_{i\,j}$ =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 $P_{i\,j}$ =0. Let $P_{i\,i}$ = 0 because the part cannot collide with itself. Because $P_{i\,j}$ in the -d direction is equal to $P_{j\,i}$ in the +d direction, three interference matrices $I_{+X,}$ I_{+Y} and I_{+Z} can be used to conclude the precedence feasibility in a disassembly sequence (A Cartesian co-ordinate system whereby the six axes ±X, ±Y, ±Z are the principal axes along which the components are disassembled is used in this work).

In 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 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 cannot be disassembled at current stage; otherwise, P_i can be disassembled from the product without collision interference.

4.2 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 usually can 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 the disassembly orientation or disassembly tool needs time and usually can increase 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 the disassembly operations also require tool change, 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.

4.3 Application of ACO with multiple search directions for disassembly planning

To apply ACO in disassembly planning, the first part in disassembly sequence can be regarded as the nest of ant colony, and the last part in 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 disassembly planning problem, $P_z(i, j)$ in Eq.(1) can be modified and represented as the probability that ant z select the disassembly sequence step from Part i to Part j in a given search direction t, and it can be represented in Eq.(8):

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

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 in Eq.(9):

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

Where, f_1, f_2 and f_3 are given as follows:

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

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

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

In Eq. (9), $[W_{i,j}]_{k\times 3}$ is the weight matrix derived from Eq. (6), which are used for three objectives: disassembly orientation change, disassembly tool change, and disassembly operation change, respectively. 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 toward 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 in Eq. (10):

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

And for k different search directions, the global updating function in ACO can be represented as in Eq. (11):

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

Where,

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

$F_{t\ (\mathrm{gb})} = Z/(1+W_{t1}N_1+W_{t2}N_2+W_{t3}N_3)$, $t\in[1, k]$, Z is a constant parameter used to adjust the added pheromone level in the step(i, j), and N_1, N_2, N_3 are number of orientation change, number of tool change, and number of disassembly operation change 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 select 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:

<u>Algorithm:</u> 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 $\sigma$ from Table 1;
Step 2.  Conclude the weight matrix using Eq.(4), Eq.(5) and Eq.(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 $Nc$(max), and let the cycle number $Nc$ =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 Eq.(8), where, Part $j$ belong to the remaining parts in the
         product that have the feasible disassembly direction at this stage;
Step 8.  Select the Part $j$ as the next part to be disassembled using 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 $j$ by Eq. (10);
Step 11. If the 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
         by Eq. (11);
Step 13. Let $t$=$t$+1, if $t<k$, go to Step 7; else, go to step 14;
Step 14. Let $Nc$ = $Nc$ +1, if $Nc< Nc$(max), go to Step 6; else, go to Step 15;
Step 15. Output the non-dominated solutions found by the ants.
```

5. Case study and discussion

The proposed disassembly planning approach with ant colony optimization 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.

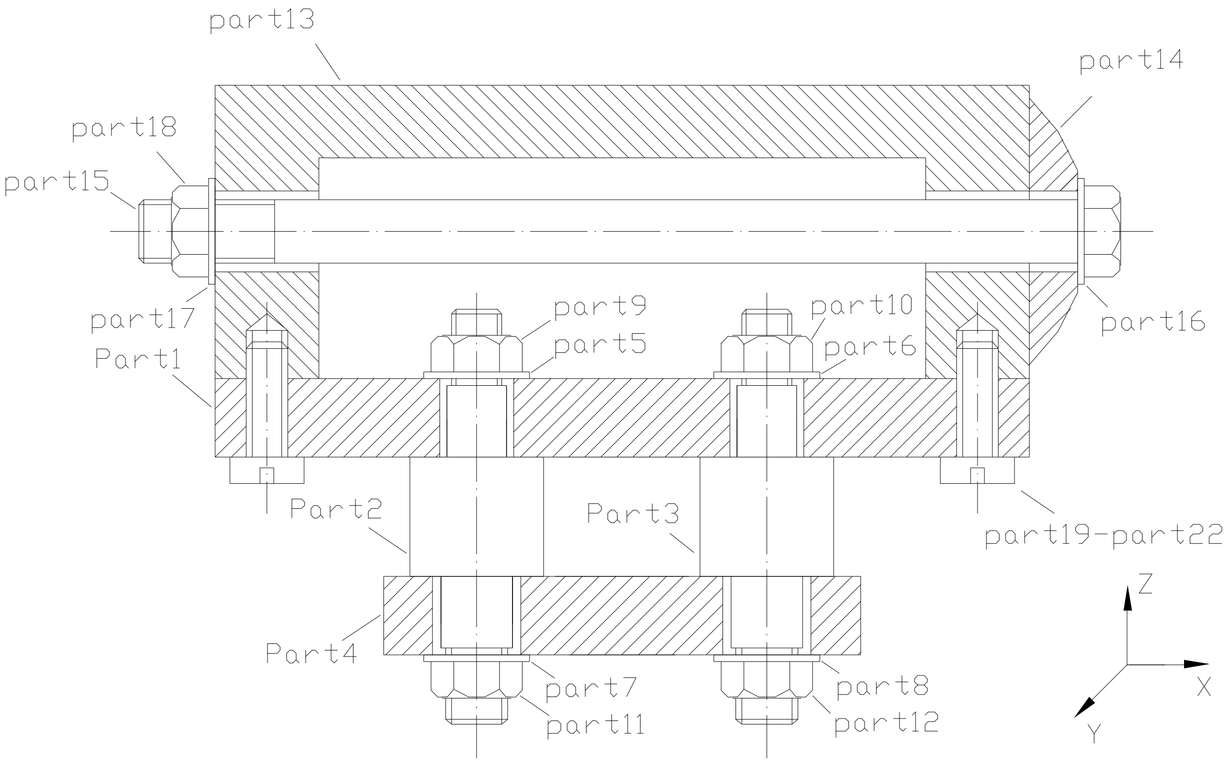


Fig. 2. An assembly consisting of 22 parts

5.1 Case study

In this case, there are 3 objectives to be optimized, and the search directions k is set as 5, then the parameter σ can be derived from Table 1 as: σ = 2. From Eq. (4) and Eq. (5), the uniform matrix U(3, 5) can be derived as follows:

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

From Eq. (6) and Eq. (7), the weight matrix can be derived as follows:

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

For 7 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 ACO algorithm are set as follows: The initial quantity of pheromone on each disassembly step is set as τ_0 = 0.5; the pheromone decay parameter *a* and *β* are set as 0.1; the parameter λ is set as 0.8. Through the experiment in the case study, the maximal cycle number *Nc*(max) is set as 500, and the constant parameter used to adjust the added pheromone level *Z* is set as 3.

In this case, the part 2, part 3 and part 15 have similar geometric shape and dimension, 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 according to their geometric shape, dimension and weight, and can be grasped with different tool with different tool number, as shown in Table 2. For the operation type, part 19, part 20, part 21 and part 22 can be unscrewed with the screw driver in disassembly process, so these four parts are assigned with the same operation type (number 2) in this case, similarly, part 9, part 10, part 11, part 12 and part 18 can be unscrewed with the wrench (operation type number 1) in this case, the other parts do not need any tool to unfasten in disassembly process, so they are assigned with the operation type (number 0) in this case, as shown in Table 2.

Part No.	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 2. Tool type and operation type of each part in the assembly

(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 [19]. All the 20 trials are converged to the feasible disassembly sequences, during which, 4 trials get 2 non-dominated solution, 8 trials get 3 non-dominated solutions, and 8 trials get 4 non-dominated solutions.

Total trials	**Trials that get 2 non-dominated solutions**	**Trials that get 3 non-dominated solutions**	**Trials that get 4 non-dominated solutions**
20	4	8	8

Table 3. 20 trial results in Test 1

In above test results, 4 non-dominated solutions found in a trial are shown in Table 4 [19].

Non-dominated solution No.	**Orientation changes**	**Tool changes**	**Operation changes**
1	4	7	6
2	3	8	7
3	2	9	5
4	4	8	5

Table 4. Test results of a trial in Test 1

In the above non-dominated solutions, the disassembly sequence of non-dominated solution No.4 is given as follows:

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 the part No.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, the non-dominated solution No. 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); the non-dominated solution No. 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); the non-dominated solution No. 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).

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, N_3 are the number of orientation changes, number of tool changes, and number of operation changes, respectively, and W_1, W_2, 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 as Fig.3 [19].

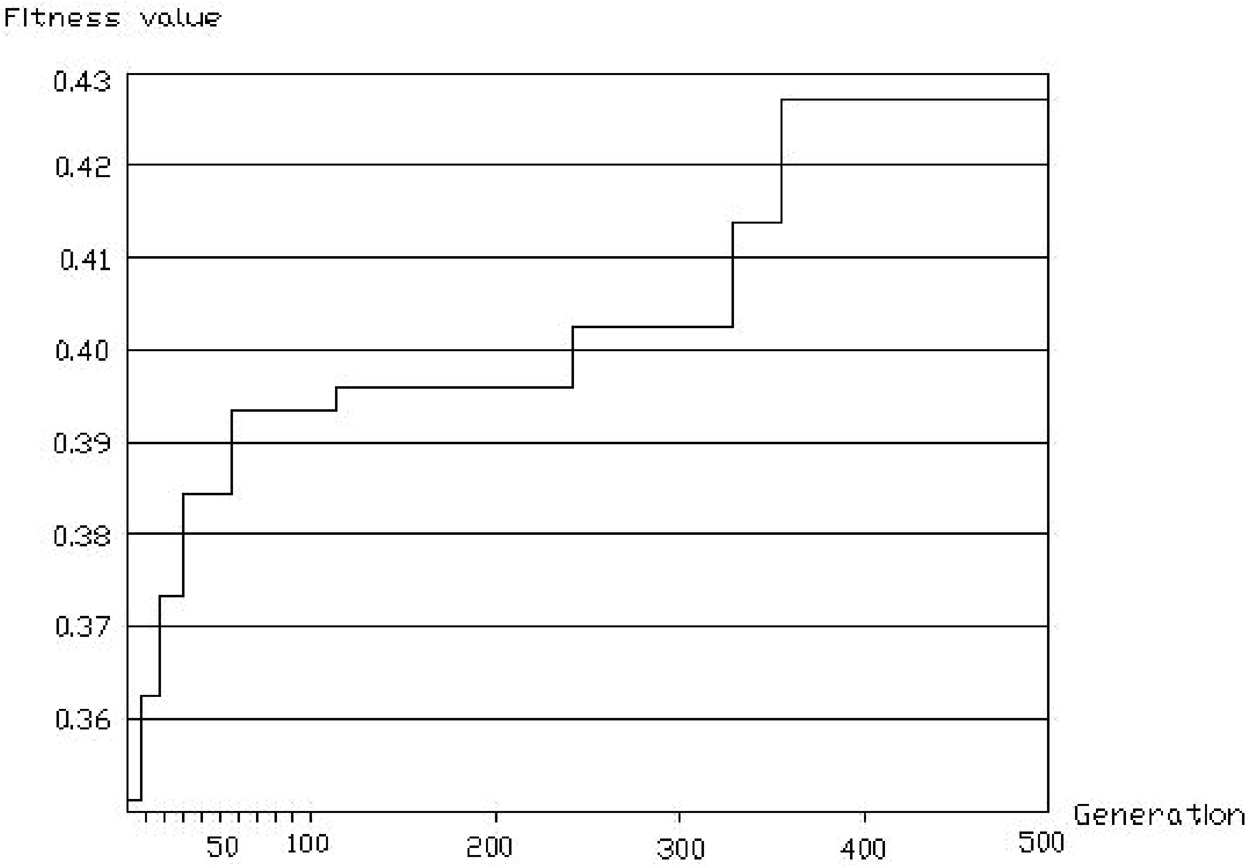


Fig. 3. The evolution performance for 500 generations

(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 same setting of the other parameters, the evolution result is shown in Table 5 [19]. 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 get 3 non-dominated solutions.

Total trials	Trials that get 1 non-dominated solutions	Trials that get 2 non-dominated solutions	Trials that get 3 non-dominated solutions
20	11	8	1

Table 5. 20 trial results in Test 1

5.2 Discussion

The above two evolution test results show, compared with the ant colony algorithm with only one search direction, the ant colony algorithm with multiple uniformly scattered search directions can be easier to find more non-dominated solutions in one trial, this is probably due to that in the later algorithm, at each step, the ants are guided along the uniformly scattered search directions toward 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, so the proposed approach can be used to derive the assembly sequence in the mean time. Compared with the assembly planning approach with multi-objective genetic algorithm with the same case study [12], the approach with ant colony algorithm is more stable and faster, all of the 20 trials can find the feasible disassembly sequence, and the average run time is 6-8 seconds to converge to a global optimal or near global optimal sequence, however, for the 20 trials using assembly planning approach with multi-objective genetic algorithm, at least 2 trials cannot find the feasible assembly sequence, and the average run time is 20-25 seconds to converge to a global optimal or near-global optimal sequence. This difference could be analyzed as follows: in the disassembly planning approach with 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 avoid the infeasible solution easily, and this can also help find the feasible solution quickly. However, in the assembly planning approach with genetic algorithm, the initial solutions are randomly generated as the whole sequence, there could be much assembly interference due to the precedence constraints, and these solutions are evolved as the whole sequence by genetic operators in the later stage, this could cost much time to repair and evolve the solution to a feasible and optimal solution, and sometimes they can not be evolved to the feasible solutions due to this highly constrained combinatory problem. So, from the above analysis, it can be seen that the disassembly planning approach with ant colony algorithm could be more efficient than the approach with genetic algorithm.

6. Conclusion

In order to achieve green manufacturing effectively, this chapter presents a multi-objective disassembly planning approach with ant colony optimization algorithm. Three objectives in 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 genetic algorithm, it can be verified that the proposed multi-objective disassembly planning approach with ant colony optimization algorithm is more stable, faster and efficient for finding more feasible non-dominated solutions.

7. References

[1] Guo W. X., Liu Z. F., Liu G. F., Pan X. Y., and Huang H. H. Disassembly sequence planning based on modularization. Journal of Computer-Aided Design and Computer Graphics, 2005, vol.17, no.3, pp.498-504.

[2] Chung C, and Peng, Q. J. An integrated approach to selective-disassembly sequence planning. Robotics and Computer-Integrated Manufacturing, 2005, vol.21, no.4-5, pp.475-485.

[3] Torres F., Puente S. T., and Aracil, R. Disassembly planning based on precedence relations among assemblies. International Journal of Advanced Manufacturing Technology, 2003, vol.21, no.5, pp.317-327.

[4] Das S. K., and Naik S. Process planning for product disassembly. International Journal of Production Research, 2002, vol.40, no.6, pp.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. International Journal of Advanced Manufacturing Technology, 2006, vol.30, no.5-6, pp.507-520.

[6] Veerakamolmal P., and Gupta S. M. A case-based reasoning approach for automating disassembly process planning. Journal of Intelligent Manufacturing, 2002, vol.13, no.1, pp.47-60.

[7] Andres C., Lozano, S., and Adenso D. B. Disassembly sequence planning in a disassembly cell context. Robotics and Computer-Integrated Manufacturing, 2007, vol.23, no.6, pp.690-695.

[8] Rai R., Rai V., Tiwari M. K., and Allada V. Disassembly sequence generation: a Petri net based heuristic approach. International Journal of Production Research, 2002, vol.40, no.13, pp.3183-3198.

[9] Smith, G. C., and Smith, S. S. F. An enhanced genetic algorithm for automated assembly planning. Robotics and Computer Integrated Manufacturing, 2002, vol.18, no.5-6, pp.355-364.

[10] Lazzerini, B., and Marcelloni, F. A genetic algorithm for generating optimal assembly plans. Artificial Intelligence in Engineering, 2000, vol.14, pp.319-329.

[11] Chen, S. F., and Liu, Y. J. An adaptive genetic assembly-sequence planner. International Journal of Computer Integrated Manufacturing, 2001, vol.14, no.5, pp.489-500.

[12] Lu C., Fuh J. Y. H., and Wong Y. S. An enhanced assembly planning approach using a multi-objective genetic algorithm. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 2006, vol.220, no.2, pp.255-272.

[13] Kongar E., and Gupta S. M. Disassembly sequencing using genetic algorithm. International Journal of Advanced Manufacturing Technology, 2006, vol.30, no.5-6, pp.497–506.

[14] McMullen P. R. An ant colony optimization approach to addressing JIT sequencing problem with multiple objectives. Artificial Intelligence in Engineering, 2001, vol.15, pp.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 and Computer Integrated Manufacturing, 2007, vol. 23, no.5, pp.503-516.

[16] Wang J. F., Liu J. H., and Zhong Y. F. A novel ant colony algorithm for assembly sequence planning. International Journal of Advanced Manufacturing Technology, 2005, vol.25, no.11-12, pp.1137-1143.

[17] Failli F, and Dini G. Ant colony systems in assembly planning: a new approach to sequence detection and optimization. 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. International Journal of Advanced Manufacturing Technology, 2006, vol.30, no.5-6, pp.481-496.

[19] Lu C., Huang H. Z., Fuh J.Y.H. and Wong Y.S. A multi-objective disassembly planning approach with ant colony optimization algorithm. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 2008, vol. 222, no.11, pp.1465-1474.

[20] Fang K. T., and Wang Y. Number-Theoretic Methods in Statistics, London, U.K.: Chapman & Hall, 1994.

[21] Dini, G., and Santochi, M. Automated sequencing and subassembly detection in assembly planning. Annals of the CIRP, 1992, vol.41, no.1, pp.1-4.

APPENDIX

Notation

$D_d(P_i\, S_m)$ represented as $\sum P_{i\,j}$, $P_j \in S_m$

I_d the interference matrix for assembly direction d

N_1 number of orientation change,

N_2 number of tool change,

N_3 number of disassembly operation change

Nc cycle number

Nc(max) maximal cycle number

$P_z(i, j)$ the probability that ant z select next node *j*

$P_{z(t)}(i, j)$ the probability that ant *z* select the disassembly sequence step from Part *i* to Part *j* in a given search direction *t*

$U_{i,j}$ the level of the factor *j* in the *i*th combination

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

α a parameter which determines the pheromone volatility on the edge from node *i* to node *j*

β the pheromone decay parameter

$\tau(i,j)$ the quantity of pheromone deposited on the edge from node *i* to node *j*

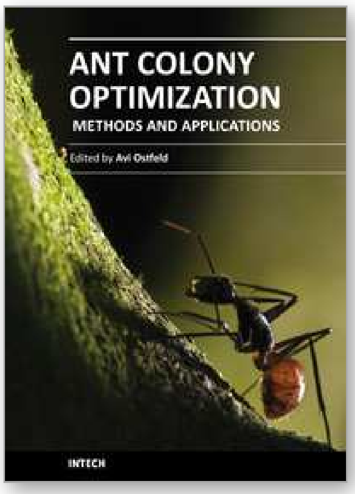
$\tau_0(i,j)$ the initial pheromone level on the edge

$\tau_t(i, j)$ the quantity of pheromone deposited on the disassembly sequence step from Part *i* to Part *j* in search direction *t*

$\eta(i,j)$ the heuristic information corresponding to the edge from node *i* to node *j*

$\eta_t(i, j)$ the heuristic value corresponding to the disassembly step from Part *i* to Part *j* in search direction *t*

λ the parameter that determine the relative importance of $\tau(i,j)$ versus $\eta(i,j)$



Ant Colony Optimization - Methods and Applications
Edited by Avi Ostfeld

ISBN 978-953-307-157-2
Hard cover, 342 pages
Publisher InTech
Published online 04, February, 2011
Published in print edition February, 2011

Ants communicate information by leaving pheromone tracks. A moving ant leaves, in varying quantities, some pheromone on the ground to mark its way. While an isolated ant moves essentially at random, an ant encountering a previously laid trail is able to detect it and decide with high probability to follow it, thus reinforcing the track with its own pheromone. The collective behavior that emerges is thus a positive feedback: where the more the ants following a track, the more attractive that track becomes for being followed; thus the probability with which an ant chooses a path increases with the number of ants that previously chose the same path. This elementary ant's behavior inspired the development of ant colony optimization by Marco Dorigo in 1992, constructing a meta-heuristic stochastic combinatorial computational methodology belonging to a family of related meta-heuristic methods such as simulated annealing, Tabu search and genetic algorithms. This book covers in twenty chapters state of the art methods and applications of utilizing ant colony optimization algorithms. New methods and theory such as multi colony ant algorithm based upon a new pheromone arithmetic crossover and a repulsive operator, new findings on ant colony convergence, and a diversity of engineering and science applications from transportation, water resources, electrical and computer science disciplines are presented.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Cong Lu (2011). Ant Colony Optimization in Green Manufacturing, Ant Colony Optimization - Methods and Applications, Avi Ostfeld (Ed.), ISBN: 978-953-307-157-2, InTech, Available from: http://www.intechopen.com/books/ant-colony-optimization-methods-and-applications/ant-colony-optimization-in-green-manufacturing

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821