



# A novel reinforcement learning framework for disassembly sequence planning using Q-learning technique optimized using an enhanced simulated annealing algorithm

Mirothali Chand  and Chandrasekar Ravi 

Department of Computer Science and Engineering, National Institute of Technology Puducherry, Karaikal, India

## Research Article

**Cite this article:** Chand M and Ravi C (2024). A novel reinforcement learning framework for disassembly sequence planning using Q-learning technique optimized using an enhanced simulated annealing algorithm. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, **38**, e5, 1–15  
<https://doi.org/10.1017/S0890060424000039>

Received: 27 May 2022

Revised: 24 December 2023

Accepted: 18 February 2024

### Keywords:

disassembly planning; simulated annealing; machine learning; optimization; reinforcement learning

### Corresponding author:

Chandrasekar Ravi;

Email: [chand191987@gmail.com](mailto:chand191987@gmail.com)

### Abstract

The increase in Electrical and Electronic Equipment (EEE) usage in various sectors has given rise to repair and maintenance units. Disassembly of parts requires proper planning, which is done by the Disassembly Sequence Planning (DSP) process. Since the manual disassembly process has various time and labor restrictions, it requires proper planning. Effective disassembly planning methods can encourage the reuse and recycling sector, resulting in reduction of raw-materials mining. An efficient DSP can lower the time and cost consumption. To address the challenges in DSP, this research introduces an innovative framework based on Q-Learning (QL) within the domain of Reinforcement Learning (RL). Furthermore, an Enhanced Simulated Annealing (ESA) algorithm is introduced to improve the exploration and exploitation balance in the proposed RL framework. The proposed framework is extensively evaluated against state-of-the-art frameworks and benchmark algorithms using a diverse set of eight products as test cases. The findings reveal that the proposed framework outperforms benchmark algorithms and state-of-the-art frameworks in terms of time consumption, memory consumption, and solution optimality. Specifically, for complex large products, the proposed technique achieves a remarkable minimum reduction of 60% in time consumption and 30% in memory usage compared to other state-of-the-art techniques. Additionally, qualitative analysis demonstrates that the proposed approach generates sequences with high fitness values, indicating more stable and less time-consuming disassemblies. The utilization of this framework allows for the realization of various real-world disassembly applications, thereby making a significant contribution to sustainable practices in EEE industries.

## Introduction

Rapid advancements in technologies, rise in population, and urbanization led to a high demand for electrical and electronic products. After usage due to their depreciation and damages, the products accumulate, which is a significant problem. The repair–reuse–recycle (RRR) process comes into action to manage this equipment’s wastage. To carry out this process smoothly, efficient planning has to be followed. Here, the disassembly planning comes into action. Disassembly planning is the process of forming an efficient plan to dismantle the products into separate entities. After that, the RRR process is carried out based on their quality and requirements.

The disassembly sequence planning (DSP) comes under the primary disassembly planning method. In DSP, a sequence is generated based on which the parts are disassembled. The sequence generation is based on the various input data of the product. Both humans and robots can be utilized to disassemble the products. But robot-based disassembly decreases manpower and time consumption. Also, it increases the safety of the laborers. DSP is considered a problem in the manufacturing sector because of its difficulty in planning sequences for complex products, time, and cost consumption.

After completing the assembling process in the manufacturing unit, the next stage is the repair and maintenance process. Based on the product model obtained from the assembling unit, the repair and maintenance of the products are done in factories and service centers. The disassembling process has three primary purposes. They are repairing, remanufacturing, and recycling the products. These three purposes help reduce the mining of new raw materials and encourage the re-usage of the same product/parts. However, in most cases, due to high time and cost consumption, the disassembling process is discouraged and not utilized by the manufacturers. So, to overcome this problem, an efficient method for generating a disassembly sequence must be planned. This particular disassembly sequence must reduce the respective time and cost consumption. DSP is classified based on the types of disassembly and levels of disassembly (Chand

and Ravi, 2023). The type of DSP varies from product to product based on its structure and complexity.

Based on the type of disassembling process, DSP is divided into the following:

1. Sequential DSP – The disassembly of parts is sequentially done one by one.
2. Parallel DSP – The disassembly of parts is done in a parallel manner, where two or three parts are disassembled simultaneously at a time.

Similarly, DSP is divided into three types based on the levels of disassembly.

1. Complete DSP – Every part of the product is dismantled into individual parts.
2. Partial DSP – Dismantling is done up to a particular level. Based on the requirements, it can be the product's initial, middle, or final levels.
3. Selective DSP – A particular part of the product is selectively disassembled based on its requirement.

Various algorithms have been employed for the DSP sequence generation process, including primary, traditional methods to advanced meta-heuristic techniques. In the DSP process, two primary stages are involved. Initially, the product model undergoes analysis, and disassembly attributes are extracted. Subsequently, algorithms analyze these disassembly attributes to generate both feasible and optimal/near-optimal disassembly sequences.

### Disassembly attributes

The disassembly attributes are extracted from the computer-aided design (CAD) models of the product automatically. The various types of matrices used for product representation in this work are explained below using the ink-pen example shown in Figure 1.

1. Stability
2. Liaison
3. Geometric feasibility
4. Precedence

### Stability matrix

The stability matrix (S) is generated based on the stability data between the parts. The stability data gives information about the product, whether the other parts of the product remain stable when one part is disassembled. The matrix shows the relationship between part  $i$  and part  $j$ . If a part  $i$  ( $P_i$ ) faces no disturbance at any direction during the dismantling of part  $j$  ( $P_j$ ), then this relation is considered as “completely stable” and denoted as “2” in the matrix S. In case  $P_i$  is stable in one direction and gets disturbed in another direction during the disassembly of  $P_j$ , then this relation is termed as “partially stable” and denoted as “1.” If the  $P_i$  has no possibility of proper stability during the disassembly of  $P_j$ , it is considered as “unstable” and denoted as “0” in the matrix S. This stability equation is given in Eq. (1), and the stability matrix for the considered ink-pen example is represented in Figure 2.

$$S(P_i, P_j) = \begin{cases} 2, & \text{if } P_i \text{ is completely independent of } P_j \\ 1, & \text{if } P_i \text{ is partially dependent of } P_j \\ 0, & \text{if } P_i \text{ is completely independent of } P_j \end{cases} \quad (1)$$

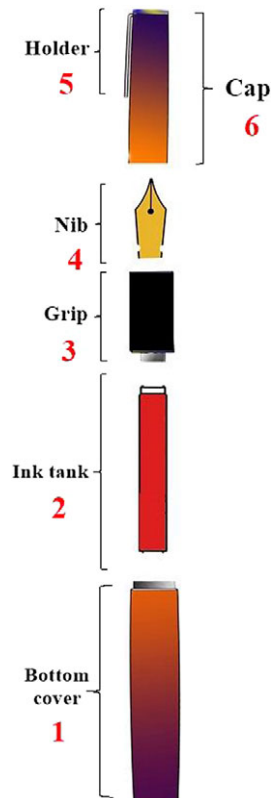


Figure 1. Ink-pen example.

	1	2	3	4	5	6
1	0	2	0	2	2	1
2	2	0	0	2	2	2
3	0	0	0	0	2	2
4	2	2	0	0	2	2
5	2	2	2	2	0	0
6	1	2	2	2	0	0

Figure 2. Stability matrix.

### Liaison matrix

This matrix L denotes the contact relationship between the elements. If part  $i$  has contact or connection with part  $j$ , it is denoted by “1.” If not, it is denoted by “0.” The contact within the parts of a product is represented in this matrix. The liaison matrix equation is given in Eq. (2). The liaison matrix for the ink-pen example is given in Figure 3.

$$L(P_i, P_j) = \begin{cases} 1, & \text{if there exists relations between } P_i \text{ and } P_j \\ 0, & \text{if there exist no relation} \end{cases} \quad (2)$$

### Disassembly feasibility matrix (geometric)

The parts' geometric direction-based relationship is given in the disassembly feasibility matrix (D). A part can be disassembled in any direction based on the product structure. The  $\pm XYZ$  directions (d) in which a part can be dismantled from another part are given as six matrices. If part  $i$  can be disassembled from part  $j$  in that direction,

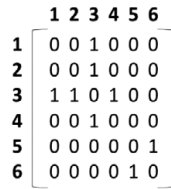


Figure 3. Liaison matrix

it is denoted as “1.” If it is not feasible for dismantling, it is denoted as “0.” The XY-axis represents the horizontal and vertical directions, respectively, whereas the Z-axis represents the gravitational direction. To explain the concept of geometric feasibility with a simple example, six directions are represented. In the case of more oversized products with complex connections, the number of directions is high. A part can be disassembled in different angles, which cannot be grouped into these six directions. In such cases, each possible angle will be considered as a direction (Anil Kumar et al., 2021). This representation is termed as “oblique-directional interference matrix.” However, the objective will be to reduce the total number of directional changes that occurred during the disassembly process.

The disassembly feasibility matrix equation is given in Eq. (3). The disassembly feasibility matrix for ink–pen is given in Figure 4. The orientation change ( $O_c$ ) score is calculated from the matrix ( $D$ ) based on the number of possible directions between part connections  $i$  and  $j$ . It is given in Eq. (4).

$$D(P_i, P_j) = \begin{cases} 1, & \text{if } P_i \text{ can be disassembled without disturbing } P_j \\ 0, & \text{if } P_i \text{ disturbs } P_j \text{ during disassembly.} \end{cases} \quad (3)$$

$$O_{c,ij} = ((+X) + (+Y) + (+Z) + (-X) + (-Y) + (-Z))_{ij} \quad (4)$$

### Precedence matrix

The precedence matrix ( $P_r$ ) gives information regarding the precedence relationship between the parts of a product. The matrix is constructed between any two parts  $i$  and  $j$ . It is given the value “1” if a particular part  $i$  has to be disassembled before part  $j$ . If the part has no dependency on the other part and can be disassembled freely, the matrix value is “0.” These matrix data are used to check the feasibility condition of the disassembly sequence. The precedence matrix for the ink–pen example is given in Figure 5, and its equation is given in Eq. (5).

$$P_r(P_i, P_j) = \begin{cases} 1, & \text{if } P_i \text{ has to be disassembled before } P_j \\ 0, & \text{if } P_i \text{ can be disassembled before } P_j \end{cases} \quad (5)$$

Usually, a product comprises various parts with multiple connections, resulting in many possible disassembly sequences. As the

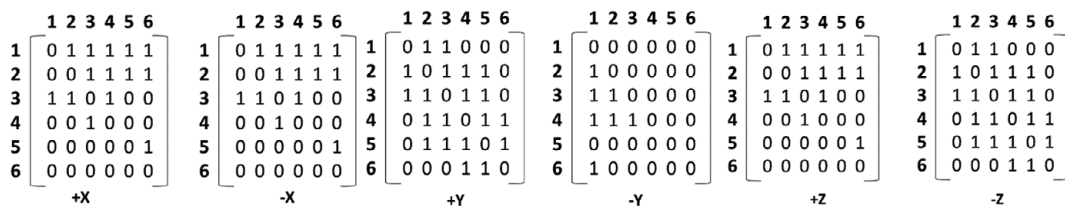


Figure 4. Disassembly feasibility matrix.

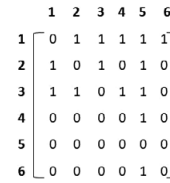


Figure 5. Precedence matrix.

product’s part count increases, the connections and its complexity also increase exponentially. In addition to this, DSP requires the processing of more product and materials’ data based on the objectives considered. This makes the DSP, a challenging and complex problem that requires an efficient and systematic framework to solve it and generate feasible and optimal sequences.

### The objectives of DSP

The common objective of DSP is to minimize the time and cost taken for the disassembling of a complete product into separate parts. The primary disassembly time ( $D_p$ ) is the essential time required to dismantle one part of the given product. A constant value is taken as the primary disassembly time (Luo et al., 2016). The disassembly fitness function derived from the total time and cost incurred to do the disassembly process is used to evaluate the objective. The optimality of a solution is based on the minimization of disassembly time ( $D_t$ ) and disassembly cost ( $D_c$ ). The disassembly time ( $D_t$ ) depends on the number of directional changes. The disassembly cost is based on the cost consumed for various tools and labors used for the disassembly process.

In this work, due to the non-availability of actual cost data, the feasibility property between various parts is considered for the calculation of  $D_c$ . Here, the cost does not refer to the actual cost. Instead, it refers to how the parts’ connection affects the disassembly sequence’s feasibility. So, if a sequence has a high cost, the particular sequence is less feasible or non-feasible. Based on the liaison, stability attributes, and assumed weight factors,  $D_c$  is calculated. The liaison property is the basic need for a feasible sequence; hence, its weight factor must be high and taken as 10. If the connection of parts ( $P_i, P_j$ ) that are to be disassembled has no liaison relationship between them, the initial cost ( $D_c = 1$ ) is increased by 10. The next important requirement is that the parts in the sequence must be either completely stable or partially stable, so their weight factor is given accordingly. If the connection of the parts in the sequence has low stability, the cost is increased by 3, and for no stability, it is 5. This is done in order to make the feasible sequence have a low-cost value. After calculating the cost for every part connection, the disassembly cost ( $D_c$ ) data for all the part connections are generated. The disassembly time ( $D_t$ ) is calculated based on the time penalty ( $T_p$ ) and the primary disassembly time ( $D_p$ ) as mentioned in Eq. (6). In  $T_p$ , a time penalty of 3 is given for

**Table 1.** Sequence and directional data

Part sequence	Directions	No. of directional changes
5-6-3-1-4-2	-X, -X, -X, -X, +Y, -Y	2
5-3-1-4-2-6	-X, -X, -X, +Y, -Y, -X	3

part connections with the number of feasible orientations less than or equal to 2. The  $D_p$  is assumed to be “1.” The DSP fitness function ( $D_f$ ) is based on the maximization of the inverse of  $D_t$  and  $D_c$ . It is given in Eq. (7).

$$D_t = T_p + D_p \quad (6)$$

$$D_f = \max (1/(D_t + D_c)) \quad (7)$$

Table 1 represents the disassembly sequences for the ink–pen with their respective directions and directional changes, which are generated through the optimization algorithms used for comparison purposes in this work.

### Related works

Several research works have been published based on the DSP problem. For better clarity, the related works are divided based on their basic types and explained in this section.

#### Graph-based methods

At the beginning stage of research in DSP, graph-based methods were used by most of the researchers. The nodes denote the parts in the graph representation, and the edges represent the relationship between those parts. The most used graph-based approach is the AND/OR graph (De Florian and Nagy, 1989), and an extended version of the AND/OR graph was introduced by Ma et al. (2011). Other graph methods include the extended process graph (Kim and Lee, 2017; Tian et al., 2019a), the disassembly precedence graph (Han et al., 2013), and the graph cut-set method (Gunji et al., 2021). The graph-based approach is preferred because it generates feasible sequences from the product data. To process the uncertainties in dynamic DSP, Vongbunpong et al. (2012) proposed a cognitive robot-based DSP approach. Later work (Vongbunpong et al., 2013) introduced a cognition-based robot’s basic and advanced behavior control strategies for disassembly planning. The graph-based method gives a proper representation of various disassembly possibilities, but the definition for graphs needs to be given manually; also, it has the problem of combinatorial explosion when generated in an automatic manner.

#### Matrix-based methods

Various matrix-based representations like contact (liaison), translational (Smith et al., 2012), precedence (Azab et al., 2011; Ren et al., 2017), directional, and interference (Kheder et al., 2017) matrices are proposed by researchers. These matrices depict the pairwise relationship between the parts of a product regarding their stability, priority, and space interference. Matrix-based data are the input in the optimization and sequence generation processes. Apart from graph and matrix-based approaches, Petri net-based (Petri and Reisig, 2008; Kuo, 2013) representation is also used. The matrix-based representation is the most suitable type

for different computational processes. Another advantage is that the matrix data are obtained from the respective product’s CAD models automatically.

#### Mathematical and meta-heuristic methods

Various computational techniques and optimization algorithms are used to generate feasible or optimal disassembly sequences. Mathematics-based computational methods like branch-and-bound (Kim and Lee, 2017), linear (Zhu et al., 2013), and nonlinear (Ullerich, 2014) methods are used to solve DSP. The mathematical models have the capability to find the optimal solutions, but the quality of the solution is entirely based on the product’s objective function and representation format. Meta-heuristic methods like genetic algorithm (GA) (Giudice and Fargione, 2007; Hui et al., 2008; Tseng and Lee, 2018), ant colony optimization (ACO) (Wang et al., 2003; Kheder et al., 2017), particle swarm optimization (PSO) (Kheder et al., 2017), and artificial bee colony optimization (Ren et al., 2018; Tian et al., 2019a,b; Liu et al., 2018, 2020) are used by various researchers to solve the DSP problem. The meta-heuristic approaches can be applied to complex products to get near-optimal or optimal solutions, but the quality of solutions varies based on the constraints of that particular approach.

Hence, these approaches are enhanced and combined with other meta-heuristic methods to produce hybrid approaches (Tian et al., 2018). For instance, Tseng et al. (2019) introduced a hybrid technique based on ACO. A combined version of GA and artificial fish swarm algorithm was submitted by Guo et al. (2019). The recent work of Xing et al. (2021) is based on an improved version of the max–min and ant colony system (IMMAS). Hybrid methods can generate better solutions when compared to single-heuristic or meta-heuristic methods. One main problem with these methods is that they are not in a straightforward manner. They use different techniques in different stages of the algorithm, which is not a generalized approach.

#### Other approaches in DSP

In addition to computational and optimization approaches, various advanced techniques and technologies have been explored for DSP. They are simulation-dependent techniques like CAD (Issaoui et al., 2017), decision-making-based de-manufacturing (Anil Kumar et al., 2021), virtual reality (Mitrouchev et al., 2016, 2017), and augmented reality (AR) (Osti et al., 2017; Chang et al., 2020). The robot–human collaborative approach for DSP was implemented following the artificial bee colony algorithm (Liu et al., 2018, 2020; Xu et al., 2020). This work tries to minimize the labor process and utilizes human knowledge for both single-objective and multi-objective problems. All these advanced techniques can be used to generate disassembly sequences for specific products with special orientations so that they cannot be used as a general technique for other products, and benchmarking with standard algorithms is a difficult process.

#### Synthesis of literature study

Though many works are done in DSP, most are based on traditional optimization algorithms and nature-inspired heuristic algorithms. The human–robot collaborative approaches have been employed for various disassembly problems. But specifically, for DSP only a few works have been published. More

research work can be done in those areas to solve the DSP problem. Machine learning approaches are used only to a limited extent in DSP due to its problems of slow convergence and local minima. To address these challenges, the utilization of adaptive parameter techniques has emerged as a viable solution. Numerous adaptive parameter techniques have been introduced across various learning problems, including learning automata-based approaches (Beigy and Meybodi, 2000, 2001; Meybodi and Beigy, 2000, 2002). Simulations of these techniques demonstrate their feasibility and effectiveness in various learning problems. In particular, the adaptation of parameters in the back-propagation (BP) algorithm has been applied to train multilayer neural networks. Inspired by this concept of parameter adaptation to overcome the problems of local minima and slow convergence rate, an enhanced simulated annealing (ESA) algorithm has been implemented in this work. By enhancing the convergence rates, a wide range of machine learning models can be explored to address the challenges in DSP (Syed Shahul Hameed and Rajagopalan, 2022, 2023).

Based on the study of related works, it is evident that there is a lack of exploration of machine learning methods, particularly reinforcement learning (RL), in solving DSP problems. Additionally, there is a significant need for an efficient framework capable of effectively handling various products.

**Research contributions**

To address these gaps, an optimized Q-learning (OQL)-based RL approach, specifically tailored for DSP (RL-DSP), is proposed in this research. The primary objective of this work is to overcome the challenges in DSP by utilizing the potential of RL technique. The contributions in this work are as follows:

1. A novel RL framework for DSP: An original RL framework has been contributed that generates optimal sequences for various products with diverse levels of complexity. While limited DSP attributes have been incorporated for processing in most existing works, all necessary attributes have been considered and appropriately weighted according to their influence in generating optimal disassembly sequences in this study.
2. Implementation of proposed ESA: An ESA is proposed in this work. Its innovative approach narrows the search space over time and optimally balances between exploration and exploitation, thus resulting in avoiding local optima and enhancing the likelihood of finding global optima.
3. The development of the OQL approach: The proposed OQL approach innovatively integrates an epsilon-greedy (EG) approach and the ESA. This approach improves the action selection process by introducing a flexible, adaptive learning model that increases the ability to solve problems more robustly. This leads to the OQL method outperforming the classic QL (CQL) technique in a variety of complex scenarios.

The proposed work is benchmarked against standard and state-of-the-art techniques, selected based on their demonstrated performance in current literature. The results show that the proposed RL-DSP approach provides better solutions in terms of optimality, as well as reduced time and memory consumption.

The organization of the remainder of this paper is as follows: Section “Background study” presents a background study of RL and

QL in relation to the research. Section “Proposed methodology” outlines the proposed methodology, including the novel ESA approach and the OQL-based RL framework for DSP. Section “Experiments and analysis” analyzes the experimental study and discusses the results, highlighting the effectiveness of the proposed approach. Finally, the paper concludes with a summary of key findings, contributions, and suggestions for future research.

**Background study**

**Reinforcement learning (RL)**

The RL technique is explored because of its efficiency in handling nondeterministic polynomial time (NP)-hard problems (Sutton and Barto, 2018). This technique aims to make the agent select an appropriate action in the current state that produces the best results. Based on the reward (feedback) received for every action, the agent (decision-maker) analyzes the current state and environment (scenario) to take a particular action (step) at that time. The state of a given problem changes based on the actions picked by the agent. If the agent takes appropriate action to solve the problem, it is rewarded. The agent aims to get maximum reward points by taking the desired actions to maximize the reward based on its feedback experience. In this method, the state (S) denotes the group of all possible states in a problem, action (A(S)) represents the group of possible actions that can be taken at a particular state (S), and reward (R) denotes the reward points given to the agent for taking a desirable action; the penalty is the low reward or negative points given to the agent for taking an undesirable or wrong action, the cost is the measure that denotes the quality of the solution or state, and time indicates the period taken for the learning process. The general framework of RL is illustrated in Figure 6. The main task of an agent is policy ( $\pi$ ) learning,  $\pi: S \rightarrow A$ . The policy (knowledge) learned must be able to generate a maximum of total rewards (M). Both learning rate ( $\alpha$ ) and discount factor ( $\gamma$ ) must be  $0 < \alpha$  and  $\gamma < 1$ . The total rewards M are defined as  $r_0 + \gamma*r_1 + \gamma^2*r_2 + \dots$

**Classic Q-learning (CQL) approach**

CQL is the standard QL technique, which is based on temporal difference (TD) learning. The updating of Q-values is based on the Q-value [Eq. (8)]. The calculated values are updated in the Q-table. From Eq. (8),  $s$  and  $a$  are the state and action at the current time (t).  $Q_t(s, a)$  is the current Q-value considering the current state and the action taken.  $Q_{t+1}$  is the Q-value of the following state ( $s'$ ) and action ( $a'$ ).  $R(s, a)$  denotes the reward obtained for that pair of (s, a).  $\max Q'(s', a')$  defines the maximum number of rewards gained given the new state ( $s'$ ) and all possible actions at the new state. Algorithm 1 gives the step-by-

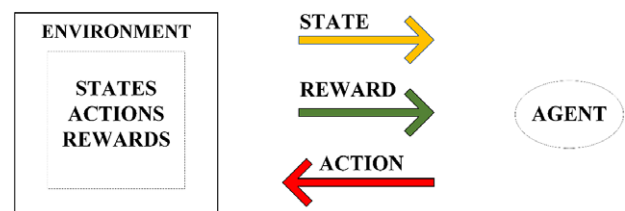


Figure 6. General framework of reinforcement learning process.

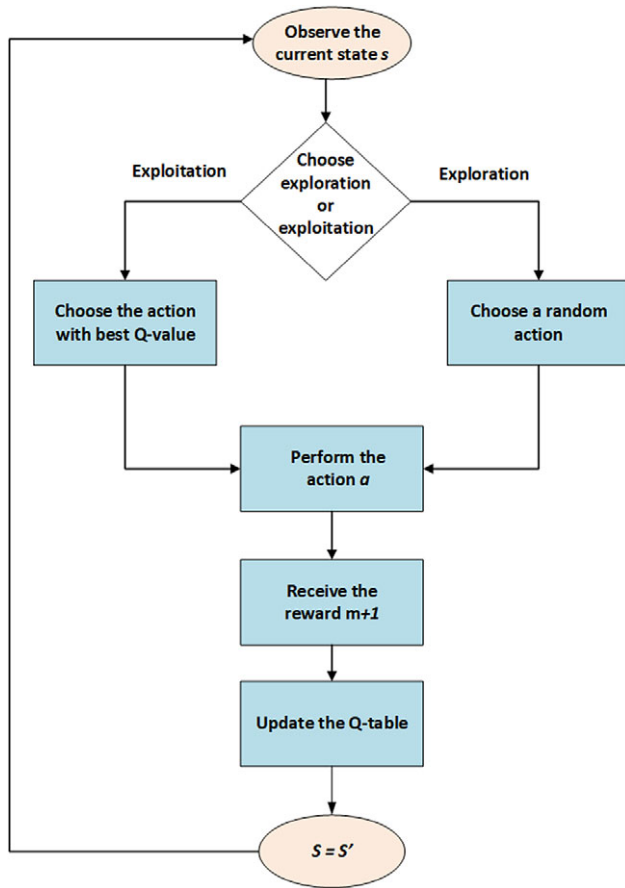


Figure 7. Classic Q-learning (CQL) flowchart.

step approach of the CQL algorithm. Figure 7 depicts the flowchart of the CQL process.

$$Q_{t+1}(s, \alpha) = Q_t(s, \alpha) + \alpha(R(s, \alpha) + \gamma \max_{\alpha'} Q(s', \alpha') - Q_t(s, \alpha)) \quad (8)$$

**Algorithm 1.** Classic Q-Learning (CQL) Algorithm.

```

1 Set the parameters: (learning rate  $\alpha$ , discount factor  $\gamma$ )
2 For each pair of state (s) & action (a), Q-matrix (s,a) = 0
3 Observe the state s
4 repeat
5   Select the action a using Epsilon-greedy method
6   Take the action a
7   Receive immediate reward r(s, a)
8   Observe the new state s'
9   Update Q (s, a) with Eq. (8)
10  s = s'
11 until all the stopping criterion is satisfied
  
```

### Epsilon-greedy (EG) approach

The exploitation and exploration trade-offs are the pivotal aspect of RL. The agent must choose the best from already exploited actions to maximize the rewards. Still, it is also required to explore more actions to find the other potentially best solutions for a given

problem. For action selection, there are various strategies followed. One of them is the epsilon-greedy method. The epsilon ( $\epsilon$ ) values should be in the range of  $0 < \epsilon < 1$ . Initially, when the epsilon rates are higher ( $\sim 1$ ), the agents explore the environment more; eventually, the epsilon rate decreases, and consequently, the agent starts to exploit more.

Based on the increasing exploration process, the agent gets more knowledge about the environment and the required policy is built. The policy  $\pi(s)$  is applied according to the given Eq. (9) (Ottoni et al., 2021). However, this EG approach lacks efficiency in its epsilon-decreasing structure.

$$\pi(s) = \begin{cases} a^*, & \text{with probability } 1 - \epsilon \\ ar, & \text{with probability } \epsilon \end{cases} \quad (9)$$

$\pi(s)$  denotes the decision policy for the current state  $s$ ,  $a^*$  denotes the best-estimated action for the state  $s$  at the current time, and  $ar$  denotes the random action selected with probability  $\epsilon$ .

### Proposed methodology

#### Proposed enhanced simulated annealing (ESA) algorithm

The proposed ESA algorithm is the enhanced version of the standard SA method. In the standard SA method, the new solution is found using the solution function and compared with the old for calculating the difference value (Kirkpatrick et al., 1983). The difference value is compared with the temperature ( $temp$ ) factor, and the next decision is taken. For annealing, the temperature reduction factor ( $\beta$ ) is used. The standard SA algorithm is given in Algorithm 2.

**Algorithm 2.** The Standard Simulated Annealing Algorithm.

```

1 Set the parameters: (temp,  $\beta$ )
2 solution_new = sol_function
3 diff = Solution_new - Solution_prev
4 repeat
5   if diff < 0 or  $e^{-diff/temp} > \text{random}(0, 1)$ ;
6     solution = new_solution
7     temp = temp *  $\beta$ 
8 until all the stopping criterion is satisfied
  
```

In SA, the temperature has to be reduced in a phased manner from its initial rate. This helps the algorithm achieve convergence. In ESA, the additional parameters  $\epsilon$ ,  $\lambda$ , and  $t_r$  are used. The SA needs a more structured decreasing rate for the temperature. In order to solve this issue,  $\lambda$  and  $t_r$  are added.  $\lambda$  and  $t_r$  denote the temperature decay factor and the temperature regularizing criterion, respectively. Their values are given between 0 and 1. The temperature regularizing criterion is used to reduce the temperature rate in a slow-phased and structured manner, whereas the temperature decay factor is used to maintain the temperature value within the positive range of values.

Based on these parameters, the equation for ESA is formulated. The epsilon gets decreased for each run based on Eq. (10). Then, a decision is taken based on the comparison with random values as given in Algorithm 3.

$$\epsilon = \epsilon - (1/(\log(temp + \lambda) + t_r)) \quad (10)$$

**Algorithm 3.** Proposed Enhanced Simulated Annealing (ESA) Algorithm.

```

1  Set the parameters: (temp, λ, tr)
2  solution_new = sol_function
3  diff = Solution_new – Solution_prev
4  repeat
5      if diff < 0 or (ε > random (0, 1));
6          solution = new_solution
7          ε = ε – (1 / (log (temp+?) + tr))
8  until all the stopping criterion is satisfied
    
```

```

7      Receive immediate reward r(s, a)
8      Observe the new state's
9      Update Q (s, a) with Eq. (6)
10     s = s'
11 until all the stopping criterion is satisfied
    
```

**Proposed optimized Q-learning (OQL) approach**

The existing EG approach in CQL uses immense randomness to decrease the epsilon rate, resulting in unstructured decay of epsilon values. A structured decrease in the epsilon rate is required to obtain more accurate results from the algorithm. The proposed ESA approach acts as a good optimization technique to gradually reduce the epsilon value. Its cooling schedule, which reduces the exploration rate over time, aligns well with the RL learning process. This approach is used in conjunction with the standard EG approach to make the eventual decay of epsilon value more controlled and organized. To define the standard values for the various parameters in this proposed OQL technique, the temperature (*temp*) is defined as the total number of iterations. This is because the more iterations the program runs, the less exploration is required to find the best solution.

Thus,  $1/\log(temp)$  decreases the epsilon value based on the number of iterations consumed by the problem. Additional parameters are used to add more regularization to the decaying epsilon values. The value of decay rate ( $\lambda$ ) is set as  $(1-\alpha)$  due to the interdependence of exploration and learning factors. The decreasing rate of epsilon is also based on the learning rate value of the algorithm. As the agent learns more about the environment, less effort is required for exploration. There is a need for a regularizing value that further adds structure to the decreasing rate. For that, the temperature regularizing criterion ( $t_r$ ) is set to the range (0, 1) to maintain the same proportion as the other parameters used in RL. The algorithm for the proposed OQL technique is given in Algorithm 4.

**Algorithm 4.** Proposed Optimized Q-Learning Technique (OQL) Algorithm.

```

1  Set the parameters: (learning rate α)
2  For each pair of state (s) & action (a), Q-matrix (s,a) = 0
3  Observe the state s
4  repeat
5      Select the action a using OQL method
           ε ∈ (0, 1); Ξ = random(0,1); tr = 0.1995
           temp = no. of. Iterations; λ = (1-α);
           ε - = 1 / (log (temp + λ) + tr);
      If Ξ > ε:
           Select the best action
      Else:
           Select a random action
6      Take the action a
    
```

For the proposed ESA [Eq. (10)], the following values are taken: epsilon ( $\epsilon$ ) = 1.0; temperature (*temp*) = no. of iterations;  $\lambda = 1 -$  learning rate ( $\alpha$ ); and temperature regularizing criterion ( $t_r$ ) = 0.1995.

To determine the optimal value for  $t_r$ , a range of values from 0.01 to 0.2 was tested through a series of experiments. It was observed that initially the values of ( $D_p$ ) showed significant variation, but they eventually converged when the value of  $t_r$  exceeded 0.1995. These experiments were conducted across all eight different products, and the results consistently indicated that the OQL method achieved better convergence and higher quality solutions when  $t_r$  was equal to or greater than 0.1995. Based on these findings,  $t_r$  is chosen as 0.1995. Since the OQL technique offers better convergence and higher quality solutions by controlling the decay of epsilon values in a structured manner, this proposed OQL technique helps in obtaining the required results in less time compared to CQL.

**Proposed RL framework for DSP**

RL is preferred to solve DSP because of its efficiency in solving optimization problems. The supervised and unsupervised learning techniques are not used because of the unavailability of huge datasets for the DSP problem. DSP requires the processing of disassembly attributes and generating the solution based on the objectives but without any training process, which is different from the usual problems solved using supervised/unsupervised learning techniques. For applying RL to DSP, the precedence, stability, liaison, and geometric feasibility data are considered and taken as conditions to generate the reward and penalty values. The initial state is the complete product with all parts connected, and the terminal state is the disassembled product with individual parts. The disassembly sequence is the required output. The primary objective of the RL agent is to generate a sequence that consumes less disassembly time and cost. Every part of a product's disconnection is taken as an action to perform. So, the state and action of DSP are formulated based on the parts disassembled and the remaining parts to be disassembled.

The main objective of this RL framework is to prepare the agent to predict an efficient DSP sequence that is feasible and optimal with minimal disassembly directions. The RL structure for DSP is given as follows:

**States:** The states are based on the total number ( $n$ ) of parts of the product to be disassembled. Initially, the complete product with all the products is the starting state. As each part is removed from the product, the state gets changed eventually, and at the final state, only the last part will be left.

**Action:** Action is the disassembling of one part from the remaining parts of the product. Action is taken based on the information of the set of parts that can be possibly disassembled given the current state. In each state, an action is taken (a part is removed), and it is carried out till there is no part left to disassemble.

**Rewards:** These functions are defined to associate the disassembly time/cost with the dismantling process. For each action (disassembly of a part), a reward is given. This reward generation

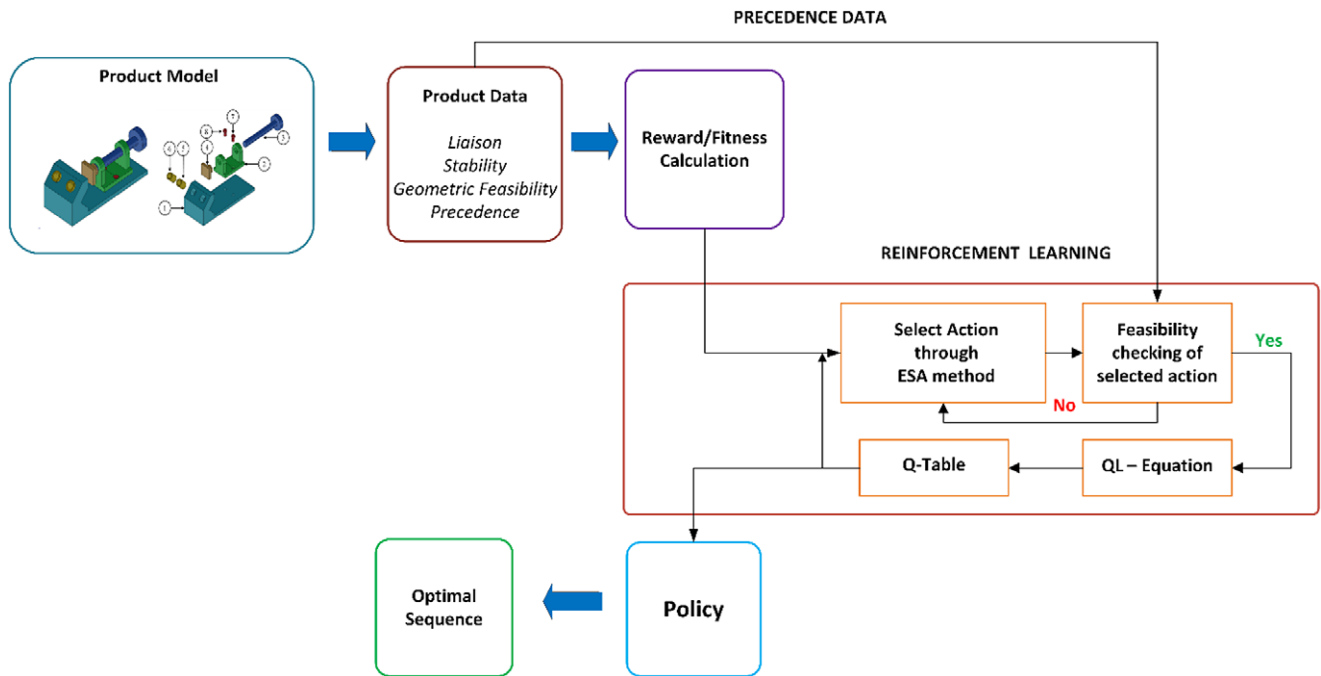


Figure 8. Proposed reinforcement learning framework for DSP – flowchart.

is based on the disassembly fitness ( $D_f$ ) function given in Eq. (11). The better the optimality, the higher the reward is obtained.

$$\text{Reward}(R) = 1000 * (D_f) \quad (11)$$

#### RL-DSP methodology

The RL-DSP methodology proposed in this work is built on four steps.

1. The stability, liaison, and geometric data are processed to generate the cost ( $D_c$ ) and time values ( $D_t$ ) of the disassembly. Then, the disassembly fitness ( $D_f$ ) value is calculated based on the  $D_c$  and  $D_t$  values.
2. The  $D_f$  values are given as the reward matrix to the RL program; then, the actions are taken based on the OQL method. This is followed by the feasibility checking process using precedence data.
3. Once the feasibility conditions are satisfied, the Q-table is generated based on the states, actions, and the QL formula given in Eq. (8).
4. Based on the Q-table values, the RL agent generates the disassembly sequences. The parameters  $t_r$ ,  $\alpha$ ,  $\gamma$ , and  $\lambda$  are tuned to get better optimality.

The flowchart of the proposed RL-DSP framework is shown in Figure 8. The calculation process of fitness and reward values is given in Algorithm 5. The DSP attributes are considered, and the  $D_f$  value is calculated based on the  $D_c$  and  $D_t$  values. The  $D_c$  and  $D_t$  values are allotted to each attribute based on their importance in generating feasible and optimal sequences. Reward values are declared as directly proportional to  $D_f$ . So, the main objective of the agent will be to maximize the rewards, thereby increasing the fitness value.

#### Algorithm 5. Proposed RL-DSP Algorithm.

```

1 Initialize RL-DSP product data; (stability, liaison, geometric
  feasibility);  $D_p = 1$ ;
2 Calculate  $D_c$ ,  $T_p$ ,  $D_b$ ,  $D_f$ 
3 repeat
4 If  $L = 0$ ,
    $D_c = 15$ ;
  Else,
    $D_c = 1$ ;
5 If  $S = 0$ ,
    $D_c = D_c + 10$ 
  Else if  $S = 1$ ,
    $D_c = D_c + 5$ 
  Else  $D_c = D_c$ 
6 If  $O_{c_{ij}} < 2$ , then  $T_p = T_p + 3$ 
  Else,  $T_p = 0$ 
7  $D_t = T_p + D_p$ 
8  $D_f = 1 / (D_c + D_t)$ 
9 Reward =  $1000 * D_f$ 
10 until all the stopping criterion is satisfied
  
```

The rewards' data are taken as the OQL algorithm's input. Randomly, a part is selected, and Q-value calculation is initiated. It is followed by the generation of subsequent part connection's Q-values. The OQL is followed for taking proper action after analyzing the exploration and exploitation factors. Based on that analysis, the algorithm either searches for new solutions in the entire search space or searches for the best solution within the explored space. Then, the action is checked for feasibility conditions based on the precedence matrix ( $P_r$ ). If it satisfies the



criteria, the action is selected, else another action is taken. This process is repeated until all the part connections are processed. Then, the Q-values are calculated for all the part disassemblies, and the final sequence is generated. Finally, a disassembly sequence with the highest reward is chosen as the solution by the agent.

## Experiments and analysis

Several algorithms presented in the current literature were analyzed for their performance, and the selection was based on the following factors:

1. This study preferred methods widely employed for solving DSP problems across various products, including traditional methods such as brute force (BF), dynamic programming (DP), and GM. In the case of meta-heuristic methods, ACO and GA are often preferred, either in their original, enhanced, or hybrid forms. Hence, these methods are considered for comparison purposes.
2. With this consideration, the recent advancement in DSP is the improved max–min ant system (IMMAS), which is included in the comparison (Xing et al., 2021).
3. Some methods were excluded due to their inferior performance or unsatisfactory results (Alshibli et al., 2016; Ren et al., 2017).
4. Additionally, many algorithms proposed in other research were product-specific, limiting their effectiveness across various products (Guo et al., 2019; Tseng et al., 2011; Yeh et al., 2012; Wu et al., 2019).
5. The CQL method, which employs a standard QL approach, provides a necessary comparison to the proposed OQL method, an enhanced version of CQL.

## Experimental setup

The RL-based DSP program is implemented as Python program and tested on different products using the PyCharm IDE running on a Windows 10 system. The results are analyzed and visualized using the Matplotlib library. The system used for testing has the following specifications: 8GB random access memory (RAM), 1 TB hard disk drive (HDD), and an Intel i5 6th Generation processor. For testing the efficiency of the proposed RL-DSP technique, eight different products with disassembly attributes are taken into consideration. The product information is given in Table 2.

Out of eight products, five are taken from the literature work from which the data are extracted manually, and for the remaining three products, the attributes are obtained from their CAD models. All these products are of different complexities with multidirectional disassembly feasibilities.

## Influence of parameters

The proposed RL-DSP framework has been tested on eight different products with multiple parts across various episodes (100, 200, 500, and 1000). Along with the number of episodes, parameters such as temperature regularizing criterion ( $t_r$ ), alpha ( $\alpha$ ), gamma ( $\gamma$ ), and epsilon ( $\epsilon$ ) also play a crucial role in generating solutions. Different parameter settings have been explored to generate optimal sequences for the diverse set of products. Considering the

satisfactory results obtained within 100 episodes, the default number of episodes is set to 100. To determine the appropriate  $t_r$  value, fitness values of various disassembly sequences are compared with their respective  $t_r$  values across 10 runs. The experimental study reveals that  $t_r$  values below 0.1895 exhibit significant variation in solution quality. However, for  $t_r$  values of 0.1995 and 0.2095, the quality of solutions remains consistent. This can be attributed to  $t_r$ , which facilitates the reduction of temperature and structured decay of epsilon rate. Consequently, the algorithm achieves a more organized and controlled epsilon decay, leading to improved convergence. Similarly, the values of  $\alpha = 0.825$ ,  $\gamma = 0.35$ , and  $\lambda = 0.0125$  are chosen based on their ability to produce high-quality results.

Based on the experimentation, it is concluded that higher learning rates ( $\alpha$ ) and lower discount factors ( $\gamma$ ) contribute to the generation of optimal solutions. On the other hand, higher values of  $\lambda$  result in increased randomness of epsilon ( $\epsilon$ ), which negatively affects the convergence rate, leading to higher iterations and time consumption.

## Evaluation metrics

For the evaluation of the proposed RL framework, three metrics are considered. They are time consumption, memory consumption, and the optimality (fitness value) of the solution. The time consumption given in seconds (s) denotes the period required by the various optimization methods to give the results. The memory consumption given in terms of megabytes (MB) refers to the amount of RAM consumed by the application (optimization methods) to run the program and provide the results. The optimality of the solution is determined by fitness values, where higher values indicate better quality results. An optimal sequence with a high fitness value exhibits feasibility, stability, and minimal directional changes. In this work, the optimality criterion is categorized as no-results, infeasible, near-optimal, and best near-optimal results for clarity and explanation purposes. Based on these three metrics, the performance of the RL-DSP framework (CQL and OQL) and other algorithms is compared.

## Results and discussion

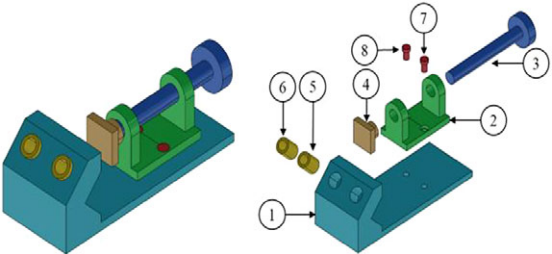
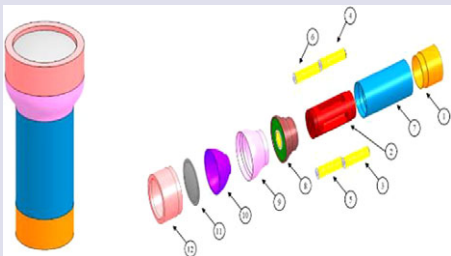
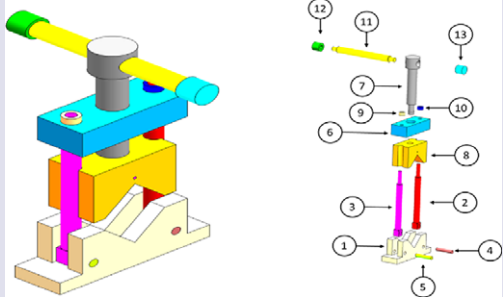
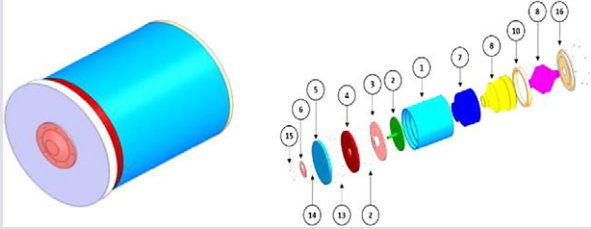
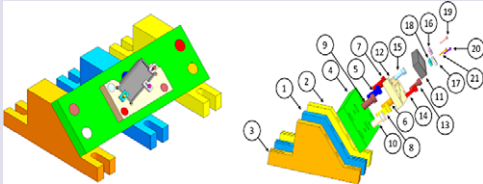
A detailed analysis and a comprehensive discussion interpreting the results based on its performance metrics are given in this section.

### Time and memory observation

The exact methods explore all the possible solutions. Hence, it produces correct results all the time but consumes more time for the huge number of parts. The optimization algorithms such as ACO, GA, and IMMAS process only a selected set of possible solutions based on their strategies. Being strategic in their processing of solutions, these algorithms give near-optimal results for products of high complexity and matched optimal solutions for smaller or less complex products. Thus, they demonstrate a lower consumption of time and memory compared to the exact methods. The gap in metrics such as time and memory between the tested algorithms is mainly due to the nature of their processing and searching mechanisms. Table 3 and Figure 9 show the time consumption comparisons of the various algorithms for the considered products. Similarly, the memory consumption comparison is given in Table 4 and Figure 10.

From Tables 3 and 4, it is evident that traditional algorithms exhibit increased time and memory consumption for products

**Table 2.** Product details

Products	Product images
Hypothetical Product – 4 Parts (Ghandi and Masehian, 2015)	–
Electric Plug – 6 Parts (Bahubalendruni and Varupala, 2021)	–
Transmission Device – 9 Parts (Tian et al., 2013)	–
Hypothetical Product – 8 Parts (Anil Kumar et al., 2021)	
Torch Light – 12 Parts (Anil Kumar et al., 2021)	
Hypothetical Product – 13 Parts	
Hypothetical Product – 16 Parts	
Hypothetical Product – 21 Parts	

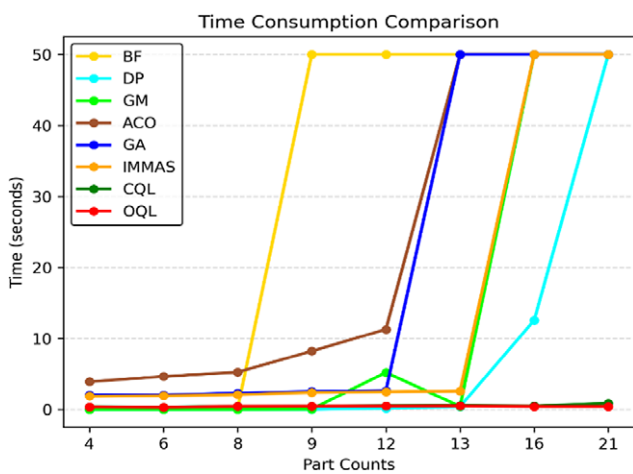
with larger part counts. The BF approach requires more than 500 seconds and 900 MB of memory for products with more than nine parts. The DP approach experiences longer execution time and memory usage for 21-part products and does not consistently yield optimal results. Similarly, the GM consumes

more time and memory for products with 16 and 21 parts, failing to provide optimal solutions for products with fewer parts (8, 12, and 13). In terms of optimization algorithms, IMMAS demonstrates superior time performance compared to ACO and GA.

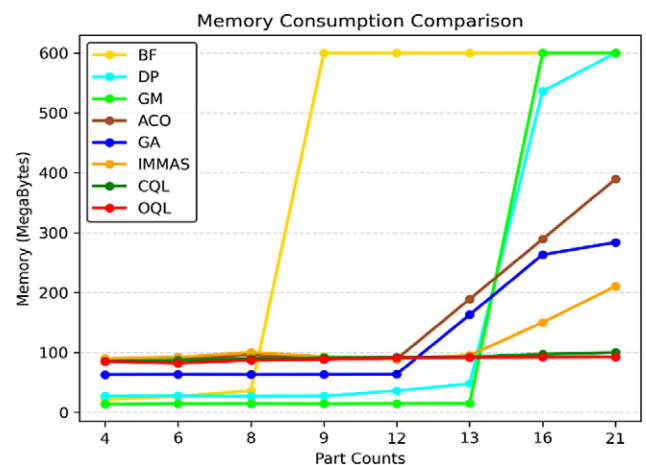
**Table 3.** Time consumption comparison of various algorithms

Parts/Algorithms	BF	DP	GM	ACO	GA	IMMAS	CQL	OQL
4	0.01	0.001	0.011	3.89	2.03	1.85	0.36	0.35
6	0.09	0.002	0.011	4.64	2.036	1.93	0.31	0.29
8	0.11	0.005	0.013	5.25	2.28	2.08	0.43	0.42
9	500>	0.007	0.03	8.21	2.5	2.39	0.453	0.450
12	500>	0.16	5.2	11.27	2.61	2.47	0.51	0.47
13	500>	0.397	0.417	11.45	2.83	2.59	0.54	0.49
16	500>	12.55	500>	21.5	2.89	2.68	0.47	0.40
21	500>	500>	500>	18	3.27	3.09	0.85	0.70

■ No Solution   
 ■ Infeasible Solution   
 ■ Near-Optimal Solution   
 ■ Best Near-Optimal Solution



**Figure 9.** Time consumption analysis chart.



**Figure 10.** Memory consumption analysis chart.

Additionally, IMMAS delivers best near-optimal solutions for products of up to 13 parts. Following IMMAS, GA exhibits satisfactory performance in terms of time and memory consumption for sequence generation compared to ACO. ACO performs better than all the traditional algorithms but falls short when compared to other optimization algorithms such as GA and IMMAS. In the case of RL methods, both the CQL and the proposed OQL outperform other algorithms in terms of solution quality. The advantage of RL methods is their efficient learning and decision-making processes.

They offer best near-optimal solutions for all products by learning from the consequences of past actions, while minimizing time and memory consumption.

When comparing CQL and the proposed OQL, it is observed that OQL consumes less time and memory than CQL. From the observation, the OQL approach seems to give better performance for products with more parts. The graphical representation of these data in Figures 9 and 10 indicates that the overall time and memory consumption of RL methods are lesser than those of other

**Table 4.** Memory consumption comparison of various algorithms

Parts/Algorithms	BF	DP	GM	ACO	GA	IMMAS	CQL	OQL
4	21	27	13.86	88.9	62.86	90	85.1	84.8
6	26.5	27.5	14.23	89	73.1	92	87.2	82
8	35.9	26.4	14.34	95.6	82.92	100	89.5	87
9	900>	26.81	14.2	99	90.98	93	90.89	88.2
12	900>	35.95	14.5	108.85	96.72	89.6	91.8	91.03
13	900>	47.87	14.87	128.9	112.89	95	92.5	91.56
16	900>	536	900>	159.07	123.32	107.3	97	92.06
21	900>	900>	900>	189.36	153.02	150	99.5	92.50

■ No Solution   
 ■ Infeasible Solution   
 ■ Near-Optimal Solution   
 ■ Best Near-Optimal Solution

**Table 5.** Fitness value comparison of various algorithms

Parts/Algorithms	BF	DP	GM	ACO	GA	IMMAS	CQL	OQL
4	0.076	0.076	0.076	0.076	0.0769	0.076	0.0769	0.0769
6	0.005	0.05	0.05	0.05	0.05	0.05	0.05	0.05
8	0.071	0.003	0.0031	0.066	0.071	0.071	0.0714	0.0714
9	0.0714	0.071	0.071	0.071	0.07142	0.0714	0.0714	0.07144
12	0.00171	0.0017	0.0017	0.037	0.037	0.0416	0.04166	0.04166
13	0.0093	0.037	0.030	0.030	0.3030	0.037	0.0370	0.0375
16	0.0028	0.027	0.0030	0.029	0.037	0.0370	0.0625	0.071
21	0.0023	0.0023	0.0023	0.033	0.0434	0.0434	0.0759	0.0769

■ No Solution   
■ Infeasible Solution   
■ Near-Optimal Solution   
■ Best Near-Optimal Solution

algorithms. Moreover, the time and memory consumption for RL methods steadily increases as the number of parts in the product rises. When comparing CQL and OQL methods, the OQL technique demonstrates a slight advantage and performs better for products with more parts.

#### Fitness and reward value observation

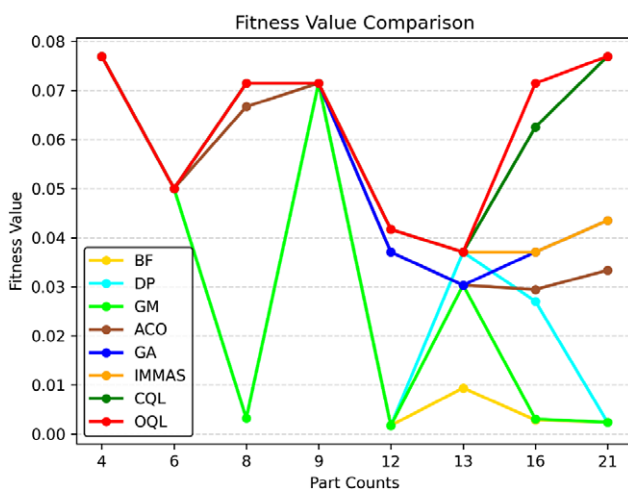
The fitness values for the disassembly sequences generated by all the considered algorithms are presented in Table 5 and visualized in Figure 11. High fitness values indicate stable sequences with minimal directional changes. From Figure 11, it is observed that both RL methods (CQL and OQL) consistently generate sequences with high fitness values for all the products considered. On the other hand, traditional algorithms, due to their high time consumption, are limited to generating solutions within 900 seconds, resulting in poorer fitness values. Both RL methods, CQL and OQL, provide best near-optimal solutions with slight differences in fitness values for large products. Although this difference may seem small, it can have a significant impact on the performance of larger and more complex products.

To compare the performance of CQL and OQL techniques in the RL aspect, the rewards obtained over 100 episodes by both RL methods for four products with 4, 8, 12, and 21 parts are analyzed and visualized in Figure 12. These four products are selected

specifically to showcase the variation in rewards obtained in both RL methods.

Observing Figure 12, it is evident that for the 4-part product, the highest reward is achieved at the 40<sup>th</sup> episode and starts converging in the OQL method, while the CQL approach reaches its highest reward only at the end of 80 episodes. In the case of the 8-part product, convergence occurs within 50 episodes with OQL, while CQL starts converging only from the 80<sup>th</sup> episode. For the 12-part product, convergence begins at 50 episodes with OQL, whereas CQL achieves convergence only after 70 episodes. In the case of a large product with 21 parts, OQL produces the best result after 50 episodes, while CQL, with variations, fails to achieve high reward result within 100 episodes.

Summarizing the results, it is clear that OQL achieves good solutions within 100 episodes, with the rewards stabilizing beyond 60 episodes on average for all the considered products. In contrast, CQL exhibits more variations in rewards and only reaches good solutions toward the end of the 100 episodes in most of the cases. Based on this experimental study, it is evident that the OQL method shows significant improvements over CQL with the EG approach. The introduction of the ESA technique in OQL allows for a more structured and organized decay of epsilon values, resulting in faster convergence and higher-quality solutions compared to the CQL method. The results clearly indicate that the OQL method can provide better results faster than CQL, making it a more efficient approach for addressing the DSP problem.

**Figure 11.** Fitness value analysis chart.

#### Conclusion

The need for an efficient DSP method for effectively managing the repair–reuse–recycle (RRR) process has been addressed in this work. A RL framework has been proposed for the DSP problem. The QL technique has been employed for generating optimal disassembly sequences. To resolve the exploration–exploitation dilemma, an OQL method based on the proposed ESA technique has been introduced. The proposed RL framework with the OQL approach outperforms the standard benchmark algorithms and state-of-the-art frameworks in terms of time, memory consumption, and solution optimality. The optimality of the solution has been evaluated using the DSP objective function. The results have demonstrated that the proposed RL-DSP framework is effective for various products and yields best near-optimal results. In conclusion, this work has demonstrated that the DSP problem can be effectively solved using the RL approach. Moreover, when the proposed ESA method is incorporated into the QL technique, it

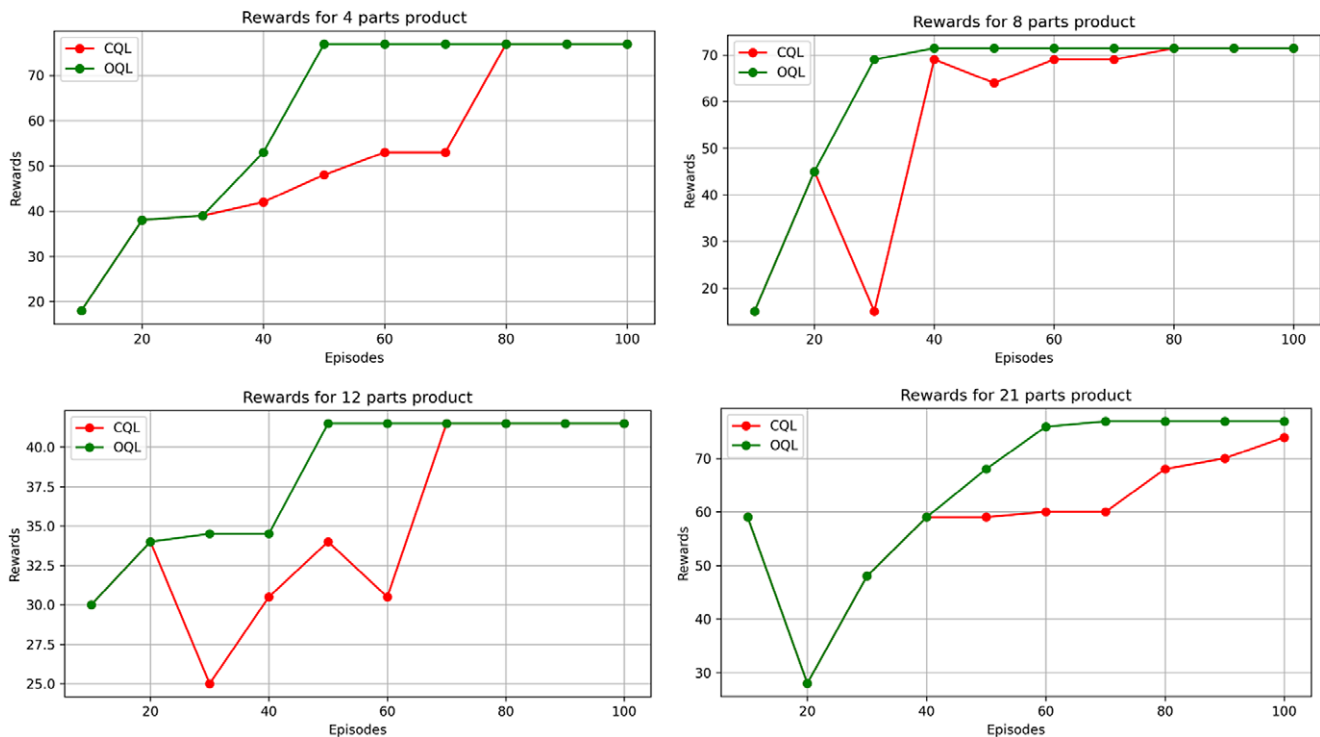


Figure 12. Reward vs episode comparison – CQL and OQL.

has been shown to produce superior results compared to the CQL method. In future work, the employment of deep RL (DRL) techniques to handle large products with multiple parts, connections, and sub-connections is planned.

**Data availability statement.** The data that support the findings will be available upon valid request.

**Funding statement.** This work received no specific grant from any funding agency, commercial or not-for-profit sectors.

**Competing interest.** The authors declare none.

## References

- Alshibli M, El Sayed A, Kongar E, Sobh TM and Gupta SM (2016) Disassembly sequencing using Tabu search. *Journal of Intelligent and Robotic Systems: Theory and Applications* **82**(1), 69–79. <https://doi.org/10.1007/s10846-015-0289-9>.
- Anil Kumar G, Bahubalendruni MVAR, Prasad VSS and Sankaranarayanan K (2021) A multi-layered disassembly sequence planning method to support decision making in de-manufacturing. *Sadhana - Academy Proceedings in Engineering Sciences* **46**(2), 1–16. <https://doi.org/10.1007/S12046-021-01622-3>.
- Azab A, Ziout A and ElMaraghy W (2011) Modeling and optimization for disassembly planning. *Jordan Journal of Mechanical and Industrial Engineering* **5**(1), 1–8.
- Bahubalendruni MVAR and Varupala VP (2021) Disassembly sequence planning for safe disposal of end-of-life waste electric and electronic equipment. *National Academy Letters* **44**(3), 243–247. <https://doi.org/10.1007/S40009-020-00994-0>.
- Beigy H and Meybodi MR (2000) Adaptation of parameters of BP algorithm using learning automata. In *Proceedings - Brazilian Symposium on Neural Networks*, 2000-January, vol 1, pp. 24–31. IEEE, Rio de Janeiro, Brazil. <https://doi.org/10.1109/SBRN.2000.889708>
- Beigy H and Meybodi MR (2001) Backpropagation algorithm adaptation parameters using learning automata. *International Journal of Neural Systems* **11**(3), 219–228. World Scientific, Singapore. <https://doi.org/10.1142/S0129065701000655>.
- Chand M and Ravi C (2023) A state-of-the-art literature survey on artificial intelligence techniques for disassembly sequence planning. *CIRP Journal of Manufacturing Science and Technology* **41**, 292–310. <https://doi.org/10.1016/j.cirpj.2022.11.017>.
- Chang MML, Nee AYC and Ong SK (2020) Interactive AR-assisted product disassembly sequence planning (ARDIS). *International Journal of Production Research*, **58**(16), 4916–4931. <https://doi.org/10.1080/00207543.2020.1730462>
- De Florian L and Nagy G (1989) Graph model for face-to-face assembly. *ICRA*, **1**, 75–78. <https://doi.org/10.1109/robot.1989.99970>.
- Ghandi S and Masehian E (2015) Assembly sequence planning of rigid and flexible parts. *Journal of Manufacturing Systems* **36**, 128–146. <https://doi.org/10.1016/j.jmsy.2015.05.002>.
- Giudice F and Fargione G (2007) Disassembly planning of mechanical systems for service and recovery: A genetic algorithms based approach. *Journal of Intelligent Manufacturing* **18**(3), 313–329. <https://doi.org/10.1007/s10845-007-0025-9>.
- Gunji BM, Pabba SK, Rajaram IRS, Sorakayala PS, Dubey A, Deepak BBVL, Biswal BB and Bahubalendruni MVAR (2021) Optimal disassembly sequence generation and disposal of parts using stability graph cut-set method for end of life product. *Sadhana - Academy Proceedings in Engineering Sciences* **46**(1), 21. <https://doi.org/10.1007/S12046-020-01525-9>.
- Guo J, Zhong J, Li Y, Du B and Guo S (2019) A hybrid artificial fish swam algorithm for disassembly sequence planning considering setup time. *Assembly Automation* **39**(1), 140–153. <https://doi.org/10.1108/AA-12-2017-180>.
- Han HJ, Yu JM and Lee DH (2013) Mathematical model and solution algorithms for selective disassembly sequencing with multiple target components and sequence-dependent setups. *International Journal of Production Research* **51**(16), 4997–5010. <https://doi.org/10.1080/00207543.2013.788794>.

- Hui W, Dong X and Guanghong D (2008) A genetic algorithm for product disassembly sequence planning. *Neurocomputing* 71(13), 2720–2726. <https://doi.org/10.1016/j.neucom.2007.11.042>.
- Issaoui L, Aifaoui N and Benamara A (2017) Modelling and implementation of geometric and technological information for disassembly simulation in CAD environment. *The International Journal of Advanced Manufacturing Technology* 89(5), 1731–1741. <https://doi.org/10.1007/s00170-016-9128-9>.
- Kheder M, Trigui M and Aifaoui N (2017) Optimization of disassembly sequence planning for preventive maintenance. *International Journal of Advanced Manufacturing Technology* 90(5–8), 1337–1349. <https://doi.org/10.1007/s00170-016-9434-2>.
- Kim HW and Lee DH (2017) An optimal algorithm for selective disassembly sequencing with sequence-dependent set-ups in parallel disassembly environment. *International Journal of Production Research* 55(24), 7317–7333. <https://doi.org/10.1080/00207543.2017.1342879>.
- Kirkpatrick S, Gelatt CD and Vecchi MP (1983) Optimization by simulated annealing. *Science* 220(4598), 671–680. <https://doi.org/10.1126/SCIENCE.220.4598.671>.
- Kuo TC (2013) Waste electronics and electrical equipment disassembly and recycling using Petri net analysis: Considering the economic value and environmental impacts. *Computers and Industrial Engineering* 65(1), 54–64. <https://doi.org/10.1016/j.cie.2011.12.029>.
- Liu J, Zhou Z, Pham DT, Xu W, Ji C and Liu Q (2020) Collaborative optimization of robotic disassembly sequence planning and robotic disassembly line balancing problem using improved discrete bees algorithm in remanufacturing ☆. *Robotics and Computer-Integrated Manufacturing*, 61, 101829. <https://doi.org/10.1016/j.rcim.2019.101829>.
- Liu J, Zhou Z, Pham DT, Xu W, Yan J, Liu A, Ji C and Liu Q (2018) An improved multi-objective discrete bees algorithm for robotic disassembly line balancing problem in remanufacturing. *International Journal of Advanced Manufacturing Technology* 97(9–12), 3937–3962. <https://doi.org/10.1007/s00170-018-2183-7>.
- Luo Y, Peng Q and Gu P (2016) Integrated multi-layer representation and ant colony search for product selective disassembly planning. *Computers in Industry* 75, 13–26. <https://doi.org/10.1016/j.compind.2015.10.011>.
- Ma YS, Jun HB, Kim HW and Lee DH (2011) Disassembly process planning algorithms for end-of-life product recovery and environmentally conscious disposal. *International Journal of Production Research* 49(23), 7007–7027. <https://doi.org/10.1080/00207543.2010.495089>.
- Meybodi MR and Beigy H (2000) A note on learning automata based schemes for adaptation of BP parameters. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 1983, pp. 145–151. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/3-540-44491-2\\_22](https://doi.org/10.1007/3-540-44491-2_22).
- Meybodi MR and Beigy H (2002) New learning automata based algorithms for adaptation of backpropagation algorithm parameters. *International Journal of Neural Systems* 12(1), 45–67. World Scientific, Singapore. <https://doi.org/10.1142/S012906570200090X>.
- Mitrouchev P, Wang C and Chen J (2016) Virtual disassembly sequences generation and evaluation. *Procedia CIRP* 44, 347–352. <https://doi.org/10.1016/j.procir.2016.02.001>.
- Mitrouchev P, Wang C and Chen J (2017) Disassembly process simulation in virtual reality environment. In Eynard B, Nigrelli V, Oliveri SM, Fajarnes GP and Rizzuti S (eds.), *Advances on Mechanics, Design Engineering and Manufacturing. Lecture Notes in Mechanical Engineering*. Springer International Publishing, pp. 631–638. [https://doi.org/10.1007/978-3-319-45781-9\\_63](https://doi.org/10.1007/978-3-319-45781-9_63).
- Osti F, Ceruti A, Liverani A and Caligiana G (2017) Semi-automatic design for disassembly strategy planning: An augmented reality approach. *Procedia Manufacturing*, 11, 1481–1488. <https://doi.org/10.1016/j.promfg.2017.07.279>.
- Ottoni ALC, Nepomuceno EG, de Oliveira MS and de Oliveira DCR (2021) Reinforcement learning for the traveling salesman problem with refueling. *Complex & Intelligent Systems*, 8, 2001–2015. <https://doi.org/10.1007/s40747-021-00444-4>.
- Petri CA and Reisig W (2008) Petri net. *Scholarpedia* 3(4), 6477. <https://doi.org/10.4249/scholarpedia.6477>.
- Ren Y, Tian G, Zhao F, Yu D and Zhang C (2017) Selective cooperative disassembly planning based on multi-objective discrete artificial bee colony algorithm. *Engineering Applications of Artificial Intelligence*, 64, 415–431. <https://doi.org/10.1016/j.engappai.2017.06.025>.
- Ren Y, Zhang C, Zhao F, Xiao H and Tian G (2018) An asynchronous parallel disassembly planning based on genetic algorithm. *European Journal of Operational Research* 269(2), 647–660. <https://doi.org/10.1016/j.ejor.2018.01.055>.
- Smith S, Smith G and Chen WH (2012) Disassembly sequence structure graphs: An optimal approach for multiple-target selective disassembly sequence planning. *Advanced Engineering Informatics* 26(2), 306–316. <https://doi.org/10.1016/j.aei.2011.11.003>.
- Sutton RS and Barto AG (2018) *Reinforcement Learning: An Introduction*. MIT Press Ltd, Massachusetts.
- Syed Shahul Hameed AS and Rajagopalan N (2022) SPGD: Search party gradient descent algorithm, a simple gradient-based parallel algorithm for bound-constrained optimization. *Mathematics* 10(5), 800. <https://doi.org/10.3390/math10050800>.
- Syed Shahul Hameed AS and Rajagopalan N (2023) MABSearch: The bandit way of learning the learning rate—A harmony between reinforcement learning and gradient descent. *National Academy Science Letters* 1, 1–6. <https://doi.org/10.1007/S40009-023-01292-1>.
- Tian G, Ren Y, Feng Y, Zhou MC, Zhang H and Tan J (2019a) Modeling and planning for dual-objective selective disassembly using and/or graph and discrete artificial bee Colony. *IEEE Transactions on Industrial Informatics* 15(4), 2456–2468. <https://doi.org/10.1109/TII.2018.2884845>.
- Tian Y, Zhang X, Liu Z, Jiang X and Xue J (2019b) Product cooperative disassembly sequence and task planning based on genetic algorithm. *International Journal of Advanced Manufacturing Technology* 105(5–6), 2103–2120. <https://doi.org/10.1007/s00170-019-04241-9>.
- Tian G, Zhou M and Chu J (2013) A chance constrained programming approach to determine the optimal disassembly sequence. *IEEE Transactions on Automation Science and Engineering* 10(4), 1004–1013. <https://doi.org/10.1109/TASE.2013.2249663>.
- Tian G, Zhou MC and Li P (2018) Disassembly sequence planning considering fuzzy component quality and varying operational cost. *IEEE Transactions on Automation Science and Engineering* 15(2), 748–760. <https://doi.org/10.1109/TASE.2017.2690802>.
- Tseng HE, Chang CC, Lee SC and Huang YM (2019) Hybrid bidirectional ant colony optimization (hybrid BACO): An algorithm for disassembly sequence planning. *Engineering Applications of Artificial Intelligence*, 83, 45–56. <https://doi.org/10.1016/j.engappai.2019.04.015>.
- Tseng HE and Lee SC (2018) Disassembly sequence planning using interactive genetic algorithms. In *ICNC-FSKD 2018 - 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, pp. 77–84. IEEE, Huangshan, China. <https://doi.org/10.1109/FSKD.2018.8686887>.
- Tseng YJ, Yu FY and Huang FY (2011) A green assembly sequence planning model with a closed-loop assembly and disassembly sequence planning using a particle swarm optimization method. *International Journal of Advanced Manufacturing Technology* 57(9–12), 1183–1197. <https://doi.org/10.1007/s00170-011-3339-x>.
- Ullerich C (2014) Advanced disassembly planning: Flexible, price-quantity dependent, and multi-period planning approaches. In *Advanced Disassembly Planning: Flexible, Price-Quantity Dependent, and Multi-Period Planning Approaches*. Springer Gabler, Wiesbaden. <https://doi.org/10.1007/978-3-658-03118-3>.
- Vongbunpong S, Kara S and Pagnucco M (2012) A framework for using cognitive robotics in disassembly automation. In *Leveraging Technology for a Sustainable World - Proceedings of the 19th CIRP Conference on Life Cycle Engineering*, pp. 173–178. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-29069-5\\_30](https://doi.org/10.1007/978-3-642-29069-5_30).
- Vongbunpong S, Kara S and Pagnucco M (2013) Application of cognitive robotics in disassembly of products. *CIRP Annals - Manufacturing Technology* 62(1), 31–34. <https://doi.org/10.1016/j.cirp.2013.03.037>.
- Wang JF, Liu JH, Li SQ and Zhong YF (2003) Intelligent selective disassembly using the ant colony algorithm. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 17(4), 325–333. <https://doi.org/10.1017/S0890060403174045>.
- Wu Y j, Cao Y and Wang Q f (2019) Assembly sequence planning method based on particle swarm algorithm. *Cluster Computing* 22(s1), 835–846. <https://doi.org/10.1007/s10586-017-1331-4>.

- Xing Y, Wu D and Qu L** (2021) Parallel disassembly sequence planning using improved ant colony algorithm. *International Journal of Advanced Manufacturing Technology* **113**(7–8), 2327–2342. <https://doi.org/10.1007/s00170-021-06753-9>.
- Xu W, Tang Q, Liu J, Liu Z, Zhou Z and Pham DT** (2020) Disassembly sequence planning using discrete bees algorithm for human-robot collaboration in remanufacturing. *Robotics and Computer-Integrated Manufacturing* **62**, 101860. <https://doi.org/10.1016/j.rcim.2019.101860>.
- Yeh W, Lin C and Wei S** (2012) Disassembly sequencing problems with stochastic processing time using simplified swarm optimization. *International Journal of Innovation, Management and Technology Management* **3**(3), 226–231.
- Zhu B, Sarigecili MI and Roy U** (2013) Disassembly information model incorporating dynamic capabilities for disassembly sequence generation. *Robotics and Computer-Integrated Manufacturing* **29**(5), 396–409. <https://doi.org/10.1016/j.rcim.2013.03.003>.
- M.C.** is Research Scholar in the Department of Computer Science and Engineering at the National Institute of Technology Puducherry, Karaikal. His research interests include artificial intelligence, machine learning, and optimization.
- C.R.** is Assistant Professor in the Department of Computer Science and Engineering at the National Institute of Technology Puducherry, Karaikal. His area of interests includes artificial intelligence, soft computing, and augmented reality.