Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A collaborative assembly for low-voltage electrical apparatuses*#

Huanpei LYU[1,2], Libin ZHANG[1], Dapeng TAN[‡1], Fang XU[1]

[1]*College of Mechanical Engineering, Zhejiang University of Technology, Hangzhou 310014, China*

[2]*College of Digital Technology and Engineering, Ningbo University of Finance and Economics, Ningbo 315175, China*

E-mail: lvhuanpei@nbufe.edu.cn; lbz@zjut.edu.cn; tandapeng@zjut.edu.cn; fangx@zjut.edu.cn

Received Sept. 4, 2021; Revision accepted May 10, 2022; Crosschecked May 24, 2023

Abstract: Low-voltage electrical apparatuses (LVEAs) have many workpieces and intricate geometric structures, and the assembly process is rigid and labor-intensive, and has little balance. The assembly process cannot readily adapt to changes in assembly situations. To address these issues, a collaborative assembly is proposed. Based on the requirements of collaborative assembly, a colored Petri net (CPN) model is proposed to analyze the performance of the interaction and self-government of robots in collaborative assembly. Also, an artificial potential field based planning algorithm (AFPA) is presented to realize the assembly planning and dynamic interaction of robots in the collaborative assembly of LVEAs. Then an adaptive quantum genetic algorithm (AQGA) is developed to optimize the assembly process. Lastly, taking a two-pole circuit-breaker controller with leakage protection (TPCLP) as an assembly instance, comparative results show that the collaborative assembly is cost-effective and flexible in LVEA assembly. The distribution of resources can also be optimized in the assembly. The assembly robots can interact dynamically with each other to accommodate changes that may occur in the LVEA assembly.

Key words: Low-voltage electrical apparatus; Collaborative assembly; Artificial potential field based planning; Adaptive quantum genetic algorithm; Dynamic interaction

https://doi.org/10.1631/FITEE.2100423 **CLC number:** TP24

1 Introduction

Low-voltage electrical apparatuses (LVEAs) are used to achieve the switching, control, protection, detection, transformation, and adjustment of circuits or non-electric objects (Li DL, 2004). LVEAs have wide applications in daily life. Due to their various and complex structures, they are often assembled on a traditional assembly line. Nevertheless, due to varied and rapidly changing requirements, traditional assembly lines cannot satisfy market requirements. These traditional assembly lines of LVEA are labor-intensive and inflexible, and cannot enable a manufacturer to obtain a dominant position within a fiercely competitive global market (Ge et al., 2021). Many manufacturers of LVEAs have emphasized the application of a new assembly method to replace traditional assembly lines. Based on the structural characteristics and usage requirements of LVEAs, there are some requirements proposed for LVEA assembly: the assembling resources should be allocated quickly, and the assembling processes should be orderly, stable, and automatic. Additionally, the types of assembled products should be able to be varied. Therefore, it is challenging to construct a new assembling method for assembling LVEAs.

Nowadays, robot assembly (RA) is extensively applied in product assembly. Compared with traditional

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. 52175124), the Zhejiang Provincial Natural Science Foundation of China (No. LZ21E050003), and the Fundamental Research Funds for Zhejiang Universities, China (No. RF-C2020004)

Electronic supplementary materials: The online version of this article (https://doi.org/10.1631/FITEE.2100423) contains supplementary materials, which are available to authorized users

ORCID: Huanpei LYU, https://orcid.org/0000-0002-6980-4723; Dapeng TAN, https://orcid.org/0000-0002-6018-9648

assembly lines, RA can achieve 24-h uninterrupted operation with no need for skilled assembly workers. In the assembly of LVEAs, the application of RA could greatly reduce labor costs (Li L et al., 2020; Wang YY et al., 2021). The study of LVEA assembling methods has become much more significant.

Many different applications of RA have been extensively researched by scholars, such as in the methods, planning, and balancing of assembly (Gao et al., 2009). The results have been useful in the study of LVEA assembly. Yu and Yang (2018) proposed a design method for a flexible assembly to achieve real-time adjustment of assembly tasks based on the assembly characteristics of multi-batch small-scale products. Çil et al. (2017) constructed a robotic parallel assembly system and optimized its assembly process using a beam search approach. For the multi-criterion optimization of a multi-product assembly line, Tavakoli (2020) proposed an integer programming model optimized by the hybrid tabu search simulated annealing algorithm (Zheng et al., 2021). For a mixed-model sequencing problem with stochastic processing time in a multi-station assembly line, Li XL et al. (2021) proposed a hybrid particle swarm optimization (PSO) algorithm to deal with the scheduling problem of flexible assembly systems without intermediate buffers.

In addition to constructing the assembly method and optimizing the assembly planning for RA, it is necessary to optimize the balancing of the complete assembly lines. This is known as the robot assembly line balancing problem (RALBP). In assembly production, two main types of assembly balancing problem occur. The first pertains to minimizing the number of assembly robots within a specific assembly cycle (RALBP-1). The second is related to minimizing the assembly time for a process involving a specific number of robots (RALBP-2) (Grzechca, 2014; Samouei et al., 2016). Regardless of the problem types, the ultimate goal of balanced optimization is to reduce the jobless rate of assembly equipment and improve the efficiency of an entire assembly line. For the balance optimization problem involving robot assembly lines, Rubinovitz et al. (1993) modeled the RALBP-1 problem and applied the best-first search and branch-definition algorithm for optimization. After that, other scholars applied genetic algorithms (GAs), hybrid genetic algorithms (Levitin et al., 2006), and PSO algorithms to optimize the RALBP-2 problem of linear robot assembly lines (Nilakantan et al., 2015; Chen and Tan, 2018). The balance optimization problem of a robot assembly line is an extension of the general assembly balance optimization problem in RA applications (Baybars, 1986). Özcan and Toklu (2009) applied a hybrid improved meta-heuristic algorithm to optimize the RALBP-1 problem for simple linear and U-shaped assembly lines. Other researchers proposed planning methods for human–robot collaborative assembly lines (Rizwan et al., 2020; Wang H et al., 2022), and a performance evaluation model was established to evaluate the feasibility of multi-resource collaborative assembly systems (Johannsmeier and Haddadin, 2017). The publications mentioned above are summarized in Table S1 (see supplementary materials for Tables S1–S15).

First, based on the requirements of LVEA assembly, robots require certain communicative functions. The real-time assembly information for each robot needs to be obtained quickly to make a decision for subsequent assembly tasks. The whole assembly line needs to be able to accommodate changes that are needed for making various new products. Second, robots require certain capacities for self-government. When a fault happens in the assembly process, a robot must be able to continue to carry out assembly tasks itself, not disturbed by the fault of other assembly cells. The whole assembly line needs to maintain stability due to the relative independence of all robots. Third, with a variety of assembly products and an uncertain assembly situation, the robots need to have dynamic interactions with each other. However, according to Table S1, existing assembly lines or methods do not satisfy the assembly requirements of LVEAs. Hence, in this paper, we propose a collaborative assembly for LVEA, which will fully satisfy the assembly requirements of LVEAs. The main contributions of this paper are as follows:

1. A collaborative assembly method is proposed for the assembly of LVEAs.

2. An artificial potential field based planning algorithm (AFPA) is proposed to achieve assembly planning and dynamic interaction in the collaborative assembly of LVEAs.

3. An adaptive quantum genetic algorithm (AQGA) is proposed to optimize the assembly processes.

4. A colored Petri net (CPN) model is constructed for the analysis of the collaborative assembly.

The main content of this paper is shown in Fig. S1 (see supplementary materials for Figs. S1–S16).

2 Problem definition

In the assembly of an LVEA, many specific tasks have predetermined precedence relationships and assembly times for each robot. With assembly balance planning, each robot can perform many more tasks in a given time. However, when changes occur in the assembly processes, such as production shifts or robot failures, the balance will be broken. The whole assembly line needs to be adjusted to accommodate the new situation, and the adjustment has a high cost. In the collaborative assembly of an LVEA, the robots can exchange information with each other, and when uncertainty happens, the robots can quickly adjust themselves to maintain the stability of the assembly processes. For the whole assembly process of an LVEA, the robot assembly is collaborative, and it is necessary to maintain the dynamic interaction of the robots to improve assembly efficiency. In brief, the collaborative assembly of an LVEA has two main challenges: (1) robot interaction and self-government; (2) assembly planning and balance optimization. To solve the problems mentioned above, certain assumptions are made, as follows:

1. Each robot can perform only two kinds of simple assembling actions.

2. The assembly time of a task is constant and known. The time may vary according to the types of tasks.

3. Each task can be completed by one kind of simple assembling action.

4. The time consumed in material flow is not considered in this study.

5. The precedence diagrams are known, and the division of tasks is not allowed.

6. The assembly time of the LVEA is constant.

3 Overview of methodology

This section provides a practical overview with respect to the identification of the collaborative assembly in the assembly of an LVEA. The collaborative assembly processes consist of many necessary basic actions (inserting, putting, rotating, pressing, sticking, and welding) that can be arranged and combined in a particular order to complete the assembly. Each robot can perform two different basic actions. Therefore, during the assembly processes, based on the information and characteristics of the LVEA released by the product information database, robots can be scheduled to achieve specific assembly functions according to a specific assembly action sequence. When the product type is changed, the assembly sequence of robots can be quickly reorganized to complete the assembly of the LVEA. The framework of the proposed assembly line is shown in Fig. 1. Additionally, for the efficient running of a collaborative assembly line, the collaborative assembling methodology is proposed in this section. The architecture of the collaborative methodology for LVEA assembly is shown in Fig. S2. It consists of four parts, parts I–IV. Each part has a specific function. Collaborative assembly can be achieved in LVEA assembly with the cooperation of these four parts.

In part I, assembly planning with AFPA is proposed to assign assembly tasks to robots. The robot number and task priority coefficient are obtained. In part II, an AQGA is proposed to optimize the robot number and task priority coefficient (assembly balance optimization). The distribution of tasks is optimized in this part. Accordingly, the assembly planning of the collaborative assembly is realized by the combination of parts I and II. In part III, the dynamic interaction based on AFPA is proposed. In this part, the robots can autonomously select tasks based on the priority coefficients of all tasks. The robots can communicate assembly information with each other to maintain dynamic interaction by applying AFPA. In part IV, CPN is applied to construct the collaborative assembly model. The feasibility of the collaborative assembly for LVEA is analyzed with the CPN model.

For the evaluation of the proposed collaborative assembly, LE and SI (Özcan and Toklu, 2009; Battaïa and Dolgui, 2013) are introduced. As shown in Eqs. (1) and (2), the maximum value of LE is used to reduce the number of robots, and the distribution of the assembly tasks among robots is balanced by reducing SI. Therefore, in part III, LE and SI are

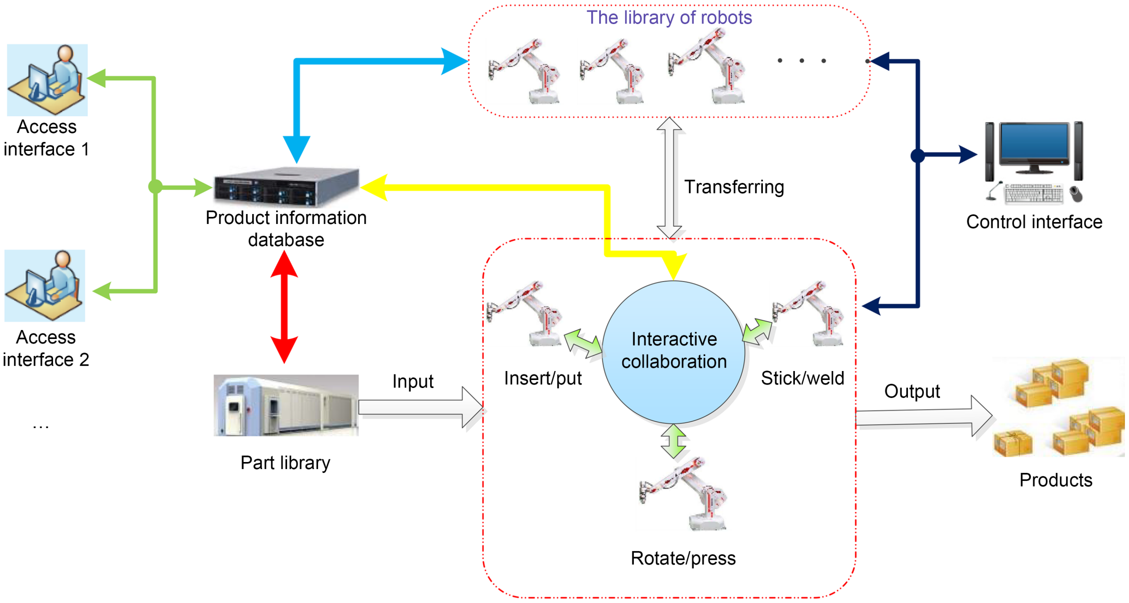

Fig. 1 A collaborative assembly

applied to evaluate the performance of AFPA. In parts I and II, multi-objective optimization fitness (see Eq. (3)) is introduced to evaluate the performance of AQGA. SI_0 and LE_0 are the initial values of SI and LE, respectively.

With dynamic interaction performance, the collaborative assembly has good compatibility in the assembly of various LVEAs. Eq. (4) is introduced to evaluate the compatibility of the assembly line. In Eq. (4), it is assumed that there are two types of LVEAs. The first is model r, and the second is model p. S_r and S_p are the numbers of robots used in assembling model r and model p, respectively. EL_r and EL_p are the maximum values of EL when assembling model r and model p, respectively. SI_r and SI_p are the minimum values of SI when assembling model r and model p, respectively. When S_r=S_p, EL_r=EL_p, SI_r=SI_p, the value of Cmp is equal to 1, and the assembly line has much better compatibility.

$$\mathrm{LE} = \frac{1}{mC}\sum_{s=1}^{m}\mathrm{ST}_s, \tag{1}$$

$$\mathrm{SI} = \sqrt{\frac{1}{m}\sum_{s=1}^{m}\left(\mathrm{ST}_{\max} - \mathrm{ST}_s\right)}, \tag{2}$$

$$\mathrm{Fitness} = \frac{100 - \mathrm{LE}}{100 - \mathrm{LE}_0} + 2\frac{\mathrm{SI}}{\mathrm{SI}_0}\frac{m}{C}, \tag{3}$$

$$\mathrm{Cmp} = \frac{S_r + 1}{S_p + 1}\frac{\mathrm{EL}_r + 1}{\mathrm{EL}_p + 1}\frac{\mathrm{SI}_r + 1}{\mathrm{SI}_p + 1}, \tag{4}$$

where ST_s is the assembly time of assembly station s, $\mathrm{ST}_{\max}$ is the maximum assembly time in all assembly stations, m is the number of assembly stations, and C is the assembly cycle of the product.

4 Artificial potential field based planning algorithm

In the collaborative assembly of an LVEA, each robot has its own tasks. For robots of the same type, confliction may occur when they select the same task. For the tasks with no restriction in the assembly sequence, confliction may occur when the tasks are selected by one robot. Therefore, in the collaborative assembly of LVEA, interactive feedback is necessary between the robots and tasks, and for the flexibility of the collaborative assembly, robustness is necessary in the assembly of LVEA. To satisfy these requirements proposed above, an AFPA is proposed in this paper.

The classical artificial potential field (APF) method is a path planning approach to make a robot move from its starting point to a goal (Khatib, 1986) while avoiding obstacles on its way. In an APF, an attracting APF which attracts robots is assigned to the destination point, and a repelling APF which repels robots is assigned to the obstacles (Montiel et al., 2015; Tan et al., 2018). Based on the influence of these two combined potentials, the robots can move to their destinations. The definition of APF (Zhang JY and Liu, 2007) is listed in Fig. S3.

Based on the theory of APF, AFPA is proposed for the collaborative assembly of LVEA. In the control of AFPA, each task is attracted by the robots, and the different kinds of tasks have different values of the attracting force. According to the value of the attracting force, the assembly tasks can be selected in sequence.

In the collaborative assembly of LVEA, the assembly tasks can be distributed to the robots quickly before assembly, and the robots can interact with each other dynamically during the assembly process. Hence, AFPA has two functions to control the collaborative assembly: (1) distribution of the tasks before assembly; (2) dynamic control of the interaction between robots and tasks.

4.1 Distribution of tasks

In the distribution of the tasks, the main goal is to achieve the scheduling of the robots statically. The sequence of assembly tasks is determined, and the priority coefficient γ_i of each task is obtained. It is assumed that in AFPA, $\rho_{\mathrm{st}i}$ exists between each assembly robot and assembly task (one or more assembly actions):

$$\rho_{\mathrm{st}i}=\omega_{\mathrm{st}i}\eta_{\mathrm{st}i}\,, \tag{5}$$

where $\rho_{\mathrm{st}i}$ is the static potential distance for task i, $\omega_{\mathrm{st}i}$ is the static potential of task i in AFPA, and $\eta_{\mathrm{st}i}$ is the static field charge of task i in AFPA.

$$\omega_{\mathrm{st}i}=\exp\left(1-\frac{1}{m_{\mathrm{st}i}t_i}\right), \tag{6}$$

where $m_{\mathrm{st}i}$ is the static quality factor of task i and t_i is the time cost in the completion of task i.

Therefore, the potential field force is calculated as follows:

$$F_{\mathrm{st\text{-}att}ij}=\mathrm{e}^{\rho_{\mathrm{st}i}}\mu_{\mathrm{st}j}, \tag{7}$$

$$F_{\mathrm{st\text{-}rep}ij}=-\frac{1}{\mathrm{e}^{3\rho_{\mathrm{st}i}}}\mu_{\mathrm{st}j}, \tag{8}$$

where $F_{\mathrm{st\text{-}att}ij}$ is the static potential attractive force of task i, $F_{\mathrm{st\text{-}rep}ij}$ is the static repulsive force generated by task i, and $\mu_{\mathrm{st}j}$ is the class factor.

$$\mu_{\mathrm{st}j}=\begin{cases}1, & j=\alpha,\\ 0, & j\neq\alpha,\end{cases} \tag{9}$$

where α=1, 2, ⋯ is the type number of the assembly robot.

$\eta_{\mathrm{st}i}$ is given as follows:

$$\eta_{\mathrm{st}i}=\left(\mathrm{e}^{F_{\mathrm{st\text{-}co}(i-1)}}-1\right)\Big/\mathrm{e}^{F_{\mathrm{st\text{-}co}(i-1)}}. \tag{10}$$

Therefore, the force of potential field is

$$F_{\mathrm{st\text{-}co}ij}=F_{\mathrm{st\text{-}att}ij}+F_{\mathrm{st\text{-}rep}ij}. \tag{11}$$

According to the math model proposed above, the distribution process of tasks with AFPA is as follows:

1. Initialize each assembly task with $F_{\mathrm{st\text{-}att}ij}$=0 and $F_{\mathrm{st\text{-}rep}ij}$=0.

2. Analyze and calculate the static potential $\omega_{\mathrm{st}i}$ and static field charge $\eta_{\mathrm{st}i}$ of each assembly task in the potential field.

3. Calculate the static potential forces generated by the robots for assembly tasks $F_{\mathrm{st\text{-}co}ij}=F_{\mathrm{st\text{-}att}ij}+F_{\mathrm{st\text{-}rep}ij}$. The robot with the largest potential force $F_{\mathrm{st\text{-}co\_max}}$ performs the corresponding assembly tasks. Reset the potential field load of the selected task to 0.

4. Analyze and calculate the static potential $\omega_{\mathrm{st}i}$ and static field charge $\eta_{\mathrm{st}i}$ of each assembly task in the potential field again. Recalculate the static situational forces $F_{\mathrm{st\text{-}co}ij}$ for unfinished tasks.

5. If $F_{\mathrm{st\text{-}co}ij}=F_{\mathrm{st\text{-}co\_max}}$, perform the i^{th} assembly task and reset the static potential and static field load of the i^{th} task to 0; otherwise, continue searching for $F_{\mathrm{st\text{-}co\_max}}$.

6. Repeat steps 2–5 until the static potential of the final assembly task and the static field load are reset to 0.

In this distribution process, the values of t_i, $\mu_{\mathrm{st}i}$, and $m_{\mathrm{st}i}$ are given first, the priority coefficient γ_i of the tasks is obtained, the scheduling of the assembly robots is achieved, and finally the sequence of tasks is determined.

4.2 Construction of a dynamic interaction pattern based on AFPA in the assembly

After the distribution of assembly tasks with AFPA, each assembly robot acquires assembly tasks correspondingly. If any robot fails, the corresponding tasks will not be completed, which will affect the assembly of the product. Therefore, it is necessary to set up a dynamic interaction pattern between the assembly robots and tasks. Each assembly robot can be fully applied during the assembly process, and the assembly process will not be stalled due to robot failures. In this study, a dynamic interaction pattern based on AFPA is introduced to achieve dynamic, collaborative interaction between a robot and a task. The related mathematical model is defined below.

In the dynamic interaction, the assembly task model is constructed first. Each task has a corresponding essential quality attribute.

$$\mathrm{Mt}_i=\begin{cases}\mathrm{e}^{\lambda_i}, & \text{if task } i \text{ is the initial task},\\ K\mathrm{e}^{\lambda_i}, & \text{otherwise},\end{cases}\tag{12}$$

where $K=\Pi_{j=1}^{u}\mathrm{Mt}_j$ is the product of the primary quality attributes of the predecessor tasks directly adjacent to task i, u is the number of tasks that are of the same assembly type as task i, and Mt_i is the primary quality attribute of task i. The value of the boundary attribute λ_i of task i is shown as

$$\lambda_i=\varepsilon_i\Big/\sum_{i=1}^{n}\varepsilon_i,\tag{13}$$

where n is the number of assembly tasks. Concurrently, the task has a corresponding task excitation factor ε_i and task guidance factor σ_i.

$$\varepsilon_i=\gamma_i\sigma_i+1,\tag{14}$$

$$\sigma_i=1-\frac{1}{n-u}.\tag{15}$$

The role of the assembly robot in the assembly task has a particular scope; namely, there is a boundary between the assembly robot and the assembly task. Only within this boundary can the robot act on the corresponding assembly task. This boundary is defined as the potential field boundary in this study. For each assembly task, the corresponding boundary entry value is given by the potential field boundary St_i:

$$\mathrm{St}_i=\mathrm{Otl}_{i-1}\lambda_i,\tag{16}$$

where Otl_{i-1} is the boundary coefficient of the adjacent predecessor task $i-1$ of task i. When the corresponding assembly task $i-1$ is completed, $\mathrm{Otl}_{i-1}=1$; when task $i-1$ is not completed, $\mathrm{Otl}_{i-1}\in(0, 1)$.

$$\mathrm{Valv}=\min(\lambda_1,\lambda_2,\cdots,\lambda_n),\tag{17}$$

where Valv is the entrance threshold of the potential field boundary. Only when $\mathrm{St}_i\geq\mathrm{Valv}$, can assembly tasks pass through potential field boundaries.

When assembly tasks cross the potential field boundary, they are subject to the dynamic forces $F_{\mathrm{rt}ij}$ of robot j on task i:

$$F_{\mathrm{rt}ij}=\frac{\varDelta_j\psi_j\mathrm{Mr}_j\mathrm{Mt}_i}{\mathrm{dr}_{ij}^2},\tag{18}$$

where Mr_j is the mass property of robot j, dr_{ij} is the dynamic potential distance between robot j and task i, and ψ_j is the state parameter of assembly robot j. ψ_j satisfies

$$\psi_j=\begin{cases}1, & \text{when robot } j \text{ is idle},\\ -1, & \text{otherwise}.\end{cases}\tag{19}$$

Additionally, $\varDelta_j$ is the type factor of assembly robot j. If the assembly tasks belong to assembly robot j, then $\varDelta_j=1$; otherwise, $\varDelta_j=0$.

When $\psi_j=1$, the task is attracted by $F_{\mathrm{rt}ij}$; when $\psi_j=-1$, the task is repelled by $F_{\mathrm{rt}ij}$ for the case of AFPA.

According to the math model proposed above, the dynamic interaction pattern with AFPA in assembly is as follows:

1. Initialize each assembly task. The corresponding boundary coefficient Otl_i ranges in (0, 1). According to the priority coefficient γ_i, calculate the stimulus factor ε_i and its boundary attributes for each task in the initial state.

2. Each assembly robot issues an assembly instruction $\psi_j=1$. The potential field boundary is opened, and the threshold of the boundary entry is taken as $\mathrm{Valv}=\min(\lambda_1,\lambda_2,\cdots,\lambda_n)$. Calculate the boundary entry value St_i for each assembly task.

3. Determine whether St_i of each task is greater than the threshold of the boundary entrance. If St_i is higher than Valv, the corresponding task i enters the potential field boundary.

4. Each assembly robot attracts the task in the potential field. The potential field force $F_{\mathrm{rt}ij}$ is calculated. The robot and the assembly task are matched in order

according to the magnitude of the potential force. When the robot obtains the corresponding assembly task, the state parameter of the assembly robot $\psi_j=-1$.

5. When task i is completed, its corresponding boundary coefficient Otl_i is set to 1. The initial adjacent predecessor task boundary coefficient Otl_{i-1} is reset to its original value, in (0, 1). The state parameter of the assembly robot is $\psi_j=1$.

6. Return to step 3 and perform the calculation again until all assembly tasks are completed.

In this process, the priority coefficient γ_i of the tasks is given first, and the dynamic interaction pattern is established last.

5 AQGA algorithm

In the collaborative assembly, with the different assembly sequences of each assembly task, the required assembly time and the number of assembly robots vary. According to the structure of LVEA, it is necessary to balance and optimize the number of robots. In this study, the assembly time for LVEA is selected. Then, the number of assembly robots (that is, the RALBP-1 assembly balance problem) needs to be optimized. This is a non-deterministic polynomial (NP-hard) problem. Many previous studies have proven that evolutionary algorithms are effective in solving NP-hard problems (Li L et al., 2021). To this end, an AQGA is introduced.

The quantum genetic algorithm (QGA) is a probabilistic search optimization algorithm based on the combination of quantum computing theory and evolutionary algorithms. In QGA, the chromosomes are represented by qubit encoding, and the evolutionary search is completed by quantum gate action and a quantum gate update (Yang et al., 2003). Since QGA has a small population size that does not affect the performance of the algorithm, QGA has a high convergence speed and strong global searchability. Hence, it has attracted attention from researchers worldwide. In this study, a QGA and an AFPA are combined to achieve collaborative assembly control. The specific mathematical model of AQGA can be given as follows.

In QGA, the smallest information unit is represented by a qubit. The state of a qubit can be expressed as follows:

$$\left|\Phi_j\right\rangle = \alpha_j|0\rangle + \beta_j|1\rangle, \tag{20}$$

where α_j and β_j meet the following conditions:

$$\left|\alpha_j\right|^2 + \left|\beta_j\right|^2 = 1,\ j=1, 2, \cdots, n. \tag{21}$$

Concurrently, a pair of complex numbers (α_j, β_j) is known as a probability bit of a qubit, expressed as $\left[\alpha_j, \beta_j\right]^{\mathrm{T}}$.

In the real number encoded QGA, the probability amplitude is directly applied for encoding. The specific encoding scheme is shown in Eq. (22):

$$\boldsymbol{q}_i = \left[\begin{vmatrix}\cos\theta_{i1}\\ \sin\theta_{i1}\end{vmatrix}\begin{vmatrix}\cos\theta_{i2}\\ \sin\theta_{i2}\end{vmatrix}\cdots\begin{vmatrix}\cos\theta_{in}\\ \sin\theta_{in}\end{vmatrix}\right]. \tag{22}$$

In Eq. (22), $\theta_{ij} = 2\pi\cdot\mathrm{rand}$, rand is a random number in (0, 1), i=1, 2, ⋯, m, j=1, 2, ⋯, n, m is the size of the population, and n is the number of quanta.

Assuming that the optimization variable solution range is $[X_{\min}, X_{\max}]$, the observed state of the quantum superposition is solved with

$$\boldsymbol{q}_{\mathrm{ch}i} = \begin{bmatrix} 0.5\left[X_{\max}(1+\cos\theta_{ij}) + X_{\min}(1-\sin\theta_{ij})\right] \\ 0.5\left[X_{\max}(1+\sin\theta_{ij}) + X_{\min}(1-\cos\theta_{ij})\right] \end{bmatrix}, \tag{23}$$

where i=1, 2, ⋯, m, j=1, 2, ⋯, n.

In the quantum phase rotation update, a method based on the gradient of the objective function is proposed (Li SY and Li, 2006) for controlling the rotation phase step size, as shown in Eq. (24). Accordingly, we introduce an adaptive function based on maximum fluctuation. The adaptive function can further control the step size search accuracy to avoid local optima.

$$\Delta_{\theta 1} = \mathrm{sign}(\mathrm{Comp})\left[1 - \frac{\left\|\nabla g(\chi) - \nabla g_{\min}\right\|}{\left\|\nabla g_{\max} - \nabla g_{\min}\right\|}\right], \tag{24}$$

where sign(Comp) is the phase turning function, Comp= $\det(\boldsymbol{\omega}_0, \boldsymbol{\omega}_1)$, $\boldsymbol{\omega}_0=[\alpha_0, \beta_0]^{\mathrm{T}}$ is the probability magnitude of the optimal solution for the current search, and $\boldsymbol{\omega}_1=[\alpha_1, \beta_1]^{\mathrm{T}}$ is the probability amplitude of the current solution.

$$\varDelta_{\theta 2} = k\exp\left(1 - \frac{1}{\text{avg}}\right). \tag{25}$$

In Eq. (25), k is the adaptive coefficient, and avg is the mean fluctuation function:

$$\text{avg} = \max(\delta_1, \delta_2), \tag{26}$$

$$\delta_1 = \left|\text{mean}(\text{fit}(\boldsymbol{q}_{\text{ch}i})) - \min(\text{fit}(\boldsymbol{q}_{\text{ch}i}))\right|, \tag{27}$$

$$\delta_2 = \left|\text{mean}(\text{fit}(\boldsymbol{q}_{\text{ch}i})) - \text{mean}(\text{fit}(\boldsymbol{q}'_{\text{ch}i}))\right|, \tag{28}$$

where δ_1 is the absolute value of the difference between the average Fitness value and the minimum Fitness value, and δ_2 is the absolute difference in the mean Fitness between the two generations. Accordingly, the improved quantum phase rotation update formula is

$$\theta = \theta_0 + \varDelta_{\theta 1}\varDelta_{\theta 2}, \tag{29}$$

where θ is the updated quantum phase and θ_0 is the original quantum phase. The specific implementation process of AQGA is initiated as follows:

1. Initialize the population. The population size n, number of qubits m, and probability of quantum mutation P_m are given. Population Q contains many individuals, i. e., $Q=\{q_1, q_2, \cdots, q_n\}$, where q_j (j=1, 2, $\cdots$, n) is the j^{th} individual, and n is the number of individuals in the population. A specific description is as shown in Eq. (22).

2. Construct the observed state of quantum superposition state $Q_{\text{ch}i}$ according to the probability amplitude of each body in Q_i, where $Q_{\text{ch}i}=\{q_{\text{ch}i1}, q_{\text{ch}i2}, \cdots, q_{\text{ch}in}\}$, and $q_{\text{ch}ij}$ (j=1, 2, $\cdots$, n) is the observation state of the j^{th} individual. A specific description is as shown in Eq. (22). In QGA, the process of constructing the observation state $Q_{\text{ch}i}$ from the probability amplitude Q includes a decoding process, and the actual values of the optimization parameters are obtained after decoding.

3. Calculate the Fitness of the observed state and the optimal value.

4. Retain the best individual and determine whether the termination condition is met. If the condition is satisfied, the algorithm is terminated; if not, go to the next step.

5. Calculate the phase of the quantum-revolving gate according to Eq. (23), and apply the updated quantum phase in Eq. (23) to act on the probability amplitudes of all individuals in the population.

6. Perform quantum phase mutation operations to generate a new population, and update $Q_{\text{ch}i}$.

7. Perform fitness evaluation calculations and calculate the optimal solution after phase update mutation.

8. Compare the previous optimal solutions with the optimal solutions after phase update mutation. If the optimal solution is degraded, the previous optimal solution is retrieved. If the optimal solution evolves, the optimal solution is replaced.

9. Increase the evolution generation and return to step 2. The calculation will not be completed until the minimum robot number is obtained.

6 Construction of the collaborative assembly model based on CPN

To analyze the performance of the proposed collaborative assembly of an LVEA, the assembly model must be constructed. Since the collaborative assembly process of an LVEA is characterized by parallel, asynchronous, event-driven, deadlock, and conflict elements (Xie, 2006; Zeng et al., 2014), the entire assembly is a discrete event dynamic system (Charbonnier et al., 1999; Zelenka, 2010; Wang JX et al., 2023). The primary modeling methods of this type of system are classified as follows (New, 1994; André et al., 2016; Pan et al., 2020; Zhang K et al., 2022): (1) graphic–analytic hybrid modeling theory, e. g., queuing theory, Markov chains, and Petri net modeling theory, (2) modeling theory based on artificial intelligence, e.g., agent systems and Holon whole subsystems, (3) granularity calculation analysis theory, and (4) modeling theory based on matrix operation, among others.

Based on the assembly characteristics of LVEA, CPN modeling theory is applied to establish the assembly model in this study. This model reflects the sequence, parallel, synchronous, and asynchronous characteristics of discrete events, and colored tokens are used to colorize different assembly tasks and instructions. The CPN theory is based on the place/transition Petri nets (Desel and Reisig, 1998), which are the most prominent and best studied class of Petri nets. The formal definition of CPN (Jensen, 1990) is given in Fig. S4.

Accordingly, based on the overall design idea for the collaborative assembly, and the CPN modeling theory, the CPN collaborative assembly model comprises mainly the place set P, transition set T, and arc set A. Based on the CPN modeling theory, the corresponding place, transition, and arc set for each unit in the collaborative assembly are designed with the software package CPN Tools (v.4.0.1). The places and transitions in the assembly model are defined in Figs. S5 and S6, respectively.

7 Simulation analysis

To verify the effectiveness of the collaborative assembly of LVEA, in this work we adopted a two-pole circuit-breaker controller with leakage protection (TPCLP), which is one kind of LVEA, as an example for simulation analysis. The processor of the computer used in this simulation was an Intel® Core™ i5-6200U CPU with 4 GB memory.

During the assembly of TPCLP, six assembly functions were required: inserting, putting, rotating, pressing, sticking, and welding. After disassembling and analyzing the product, in the assembly of this product, 45 assembly tasks had to be performed, each of which was represented by a corresponding letter. In collaborative assembly, each robot can perform only two assembly functions, so three robots with different functions were needed to complete the assembly. The corresponding relationship between the robot and the assembly task is shown in Table S2, where R_p represents an assembly robot with insertion and putting functions, R_s represents a robot with rotation and pressure functions, and R_w represents an assembly robot with welding and adhesion functions. The constraint relationship between tasks is shown in Fig. S7, in which the number next to each task name mentioned above is the time consumed in completing the assembly task.

During the collaborative assembly of TPCLP, some requirements had to be met: (1) In the assembly process, it was necessary to satisfy the assembly constraint relationship between the various tasks shown in Fig. S7. (2) During the assembly process, assembly conflicts between robots and assembly tasks had to be avoided. (3) Scheduling too many robots had to be avoided since this can result in a waste of resources. Therefore, in a given assembly cycle, the total number of robots employed had to be as low as possible.

To verify the feasibility of the collaborative assembly proposed above, some functions were needed in the simulation. First, the CPN model of collaborative assembly was established to analyze the interaction and self-government of the robots. Second, some evaluation parameters, such as LE, SI, and Cmp, were calculated to analyze the assembly planning and dynamic interaction of the collaborative assembly. The main work of this part was as follows:

7.1 CPN model construction of the TPCLP assembly

According to the proposed collaborative assembly, taking TPCLP as an example, the specific assembly of the cooperative assembly process was analyzed using CPN Tools. The entire assembly method model primarily included a task data unit, an assembly robot unit, an information-processing unit, and a product task detection unit.

Additionally, some parameters needed to be set before the assembly analysis. Based on the assembly balance analysis of AQGA, the number of robots varied with the value of the assembly cycle, which can be found in Table S3.

In this assembly analysis, the assembly cycle was set to 90 time units. Accordingly, the number of robots was five, including two R_p-type robots, two R_s-type robots, and one R_w-type robot. The number of assembly products to be completed was 10. With the combination of the assembly units, the CPN collaborative assembly model is as shown in Fig. 2.

Based on the CPN collaborative assembly model, the simulation monitoring unit Monitor in CPN Tools was used to analyze the interaction and self-government of the robots. By applying the "Count Transition" function in the simulation control unit Monitor, some changes were observed in the system (Asm_Rp011, Asm_Rp012, Asm_Rs011, Asm_Rs012, Asm_Rw011, T_Rp, T_Rs, and T_Rw), and the number of completed tasks of the five robots was analyzed. The results are listed in Table S4. By the monitoring of transitions T_ Rp, T_Rs, and T_Rw (Fig. 2), the number of completed assembly tasks in the assembly process was detected. Overall, 330 inserting/putting tasks, 180 spinning/pressing tasks, and 210 welding/sticking tasks were detected. According to the number of tasks obtained by each assembly robot, for the same assembly tasks, the number of tasks for the assembly robots was similar,

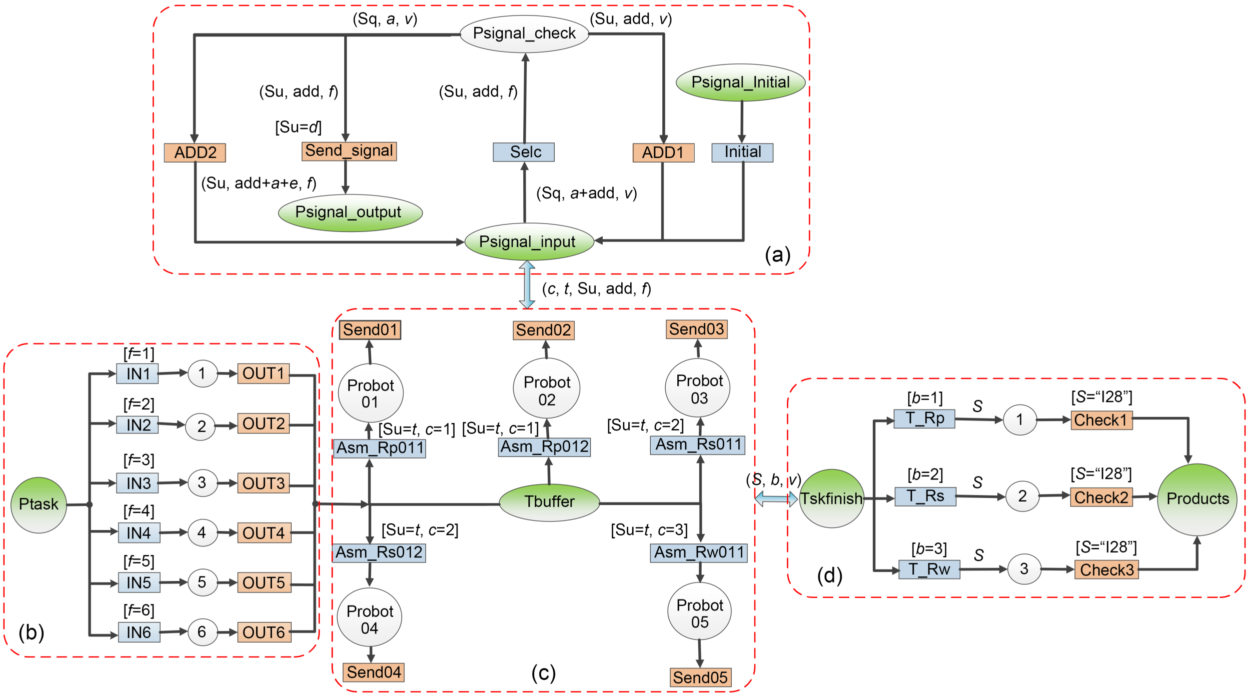

Fig. 2 CPN collaborative assembly model: (a) product and task detection unit; (b) task data unit; (c) assembly robot unit; (d) information processing unit

and each robot maintained good communication to balance the task distribution. Accordingly, as long as the same type of assembly robot operated normally, the failure of some robots would not affect the entire product assembly process. Hence, the robots had a good capability of self-government in the collaborative assembly of LVEA.

The randomness of the distribution process of the tasks should be considered during the assembly. In the multi-batch simulation analysis, three confidence intervals of 90%, 95%, and 99% were set to estimate the probability of the assembly tasks being completed by the robots. As shown in Fig. S8, the average number of assembly tasks performed by each robot was obtained from 10 consecutive simulation analyses. In Fig. S8, the abscissa is a given confidence interval value, and the ordinate is the number of assembly tasks to be performed by each assembly robot. When the confidence interval was 99%, the number of tasks included increased, but they all fluctuated substantially within a stable interval. In Fig. S9, the selected confidence interval was 95%, and the numbers of simulation batches were 5, 10, 15, 20, and 25. According to the confidence curve, during the operation of the assembly system, the number of tasks was stabilized. The number of assembly tasks among robots of the same type also tended to be the same. Therefore, it was found that the robots had good interactivity to maintain the stability of the collaborative assembly.

7.2 Analysis of AFPA and AQGA in collaborative assembly

In the collaborative assembly of TPCLP, AFPA and AQGA were applied to control the assembly process. Before TPCLP assembly, the number of robots and the assembly planning were optimized. During the TPCLP assembly, the assembly processes were in a dynamic balance to maintain the effectiveness and stability of the assembly. In the simulation, we analyzed the effectiveness of AQGA applied in the optimization of the assembly balance by comparison with other algorithms. AFPA was also analyzed to verify its performance for assembly planning and dynamic interaction by comparison with some traditional assembly methods. The details of the simulation analysis processes were as follows:

First, the performance of AQGA was analyzed by introducing typical genetic algorithms, including GA, simulated annealing genetic algorithm (GASA) (Baykasoglu, 2006; Ren et al., 2015), QGA (Yang et al., 2003), and objective function gradient based quantum genetic algorithm (RQGA) (Li SY and Li, 2006), and optimization comparisons were carried out.

In the application of typical genetic algorithms, some critical parameters needed to be set, such as the population size, individual variable dimension, genetic generation, crossover probability, and mutation rate. Based on previous research on GA (Deb, 1998; Ji et al., 2012), the critical parameter value range was as shown in Table S5. If the parameter values were not in the given range, it was difficult to obtain the optimization results. According to the number of tasks, the individual variable dimension was 45. According to the analysis, the quantum rotation angle ranged from 0.01 to 0.5. When the value was smaller than 0.01, the calculation burden was increased. However, if the value was larger than 0.5, the optimization results were missed. The relevant parameters of the compared algorithms were set as shown in Fig. S10.

In the comparative analysis of the algorithms, AQGA was compared with commonly used algorithms GA, GASA, RQGA, and QGA based on a combination with AFPA. In this comparison, according to Table S6, the theoretical assembly cycle C_t of TPCLP was set to 70, 90, 110, 130, and 150 time units. The target values of the analysis and comparison were Fitness, LE, SI, and *S*. The specific analysis results were shown in Fig. 3 when C_t was set to 90 time units. The results showed that the AQGA algorithm proposed in this paper had better optimization performance. This algorithm exhibited better convergence performance than the compared algorithms GA, GASA, RQGA, and QGA, and it can rapidly converge within a small number of iterations without being limited to a local optimum. The detailed results of the comparison of the algorithms at different C_t's are shown in Table S6. We found that through AQGA optimization, the Fitness, LE, SI, *S* values obtained were better than those of the compared algorithms. These results were all better than the optimal values provided by several other algorithms. As shown in Fig. S11, AQGA was better than the compared algorithms in the optimization of the robot number. For the different numbers of assembly cycles, AQGA could calculate the minimum robot number quickly. Hence, AQGA can improve the effectiveness of assembly balancing in the collaborative assembly.

To analyze the performance of AQGA more deeply, a non-parametric statistical test was applied. Twenty samples were selected. The statistical data of the different algorithms are listed in Table S7. The descriptive statistics and ranks of the robot number (RBN), SI, LE, and Fitness are listed in Tables S8 and S9, respectively. By analysis of Mean, Std., Min., Max., and mean ranks,

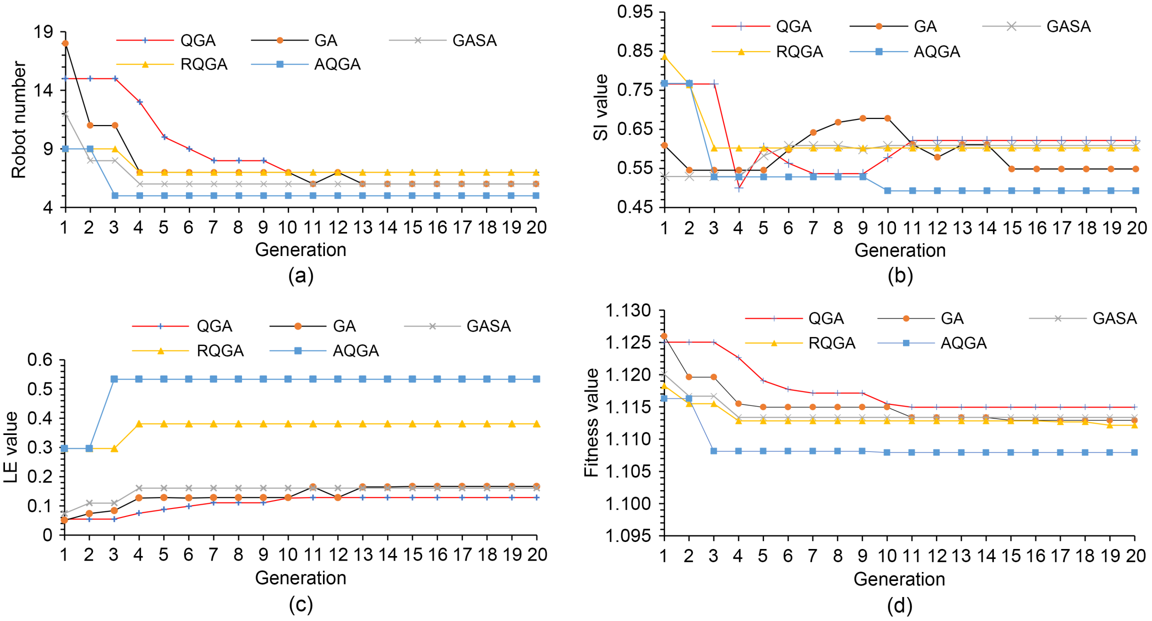

Fig. 3 Comparative analysis of a combination of AFPAs with different optimization algorithms: (a) robot number; (b) SI; (c) LE; (d) Fitness

we can find that the values (RBN, SI, LE, and Fitness) obtained by AQGA were better than those of other algorithms. From Table S10, we can also find that the values of p were all smaller than 0.05, so the sampling data obtained from the algorithms AQGA, GA, GASA, QGA, and RQGA were significantly different.

Then, the performance of assembly planning optimization and dynamic interaction of AFPA was analyzed. The simple straight assembly line (ST_line), two-sided assembly line (TS_line), and U-shaped assembly line (U_line) were compared with the collaborative assembly to verify the good performance of AFPA. Fig. 4 shows the assembly resource allocation and optimization performance of the collaborative assembly for AFPA and AQGA. Assembly cycles of 90 and 150 time units were selected for comparison and analysis, the Fitness values of various assembly methods were compared, and the comparison of LE, SI, and RBN were as listed in Figs. S12–S14. It was assumed that during the assembly processes an assembly robot could perform only two assembly actions (such as spinning/pressing, inserting/putting, and sticking/welding). Therefore, in the basic assembly line (ST_line, TS_line, and U_line), at least three different kinds of robots were required in each assembly station. However, in the collaborative assembly, each assembly station had only one kind of assembly robot. Hence, from Fig. 4 and Figs. S12–S14, it can be found that for different assembly cycles, the collaborative assembly had excellent assembly planning optimization and dynamic interaction performance compared with several other basic assembly lines. AFPA significantly improved the utilization efficiency of each robot and optimized the assembly resource distribution. Furthermore, for the same assembly cycle, the collaborative assembly was able to reduce the number of assembly robots to reduce assembly cost.

To analyze the performance of AT_line more deeply, a non-parametric statistical test was applied to analyze the performance of AT_line, ST_line, TS_line, and U_line. In this analysis, 16 samples were selected. The statistical data, descriptive statistics, and ranks of different assembly lines in terms of RBN, SI, LE, and Fitness are listed in Tables S11–S13. By analysis of Mean, Std., Min., Max., and mean ranks listed in the supplemental materials, we found that the values of RBN, SI, LE, and Fitness obtained by AT_line were better than those of other algorithms. From Table S14, we can also find that the values of p were all smaller than 0.05, so the sampling data obtained from assembly lines AT_line, ST_line, TS_line, and U_line were significantly different.

To analyze the dynamic interaction performance of AFPA, the values of LE, SI, S, C_t, and C_r (actual assembly cycle of the product) were analyzed for comparison with the values obtained before dynamic interaction, and the compatibility assembly of various products was introduced to analyze the performance of dynamic interaction.

In this simulation different assembly cycles of 70, 90, 120, and 150 time units were set to analyze changes in the values of LE, SI, and C_r, compared with the values obtained before the dynamic interaction of AFPA. The results are shown in Table 1 and Fig. S15. Comparative analysis showed that, through the optimization of the dynamic assembly process with AFPA, the assembly collaboration between the assembly robots was greatly improved and the robots were fully used in the assembly process. Additionally, the algorithm can reduce the assembly time and improve the assembly efficiency.

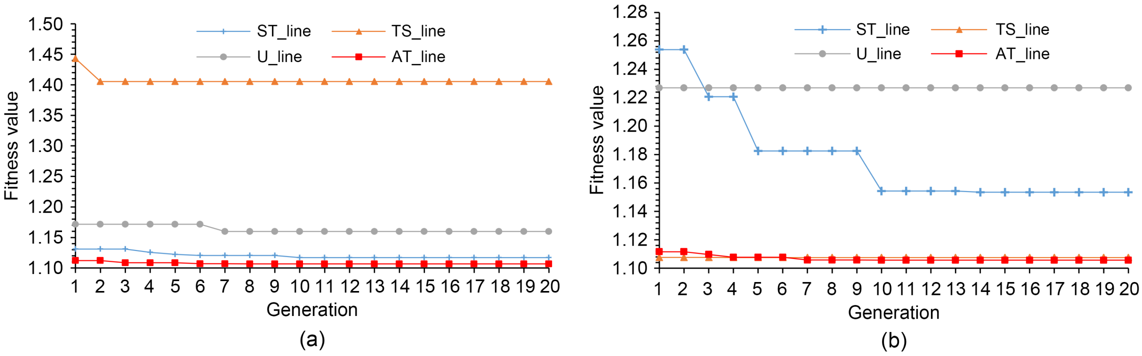

Fig. 4 Fitness values of various assembly methods for different assembly cycles: (a) 90 time units; (b) 150 time units

Table 1 Effectiveness of dynamic balance with AFPA

C_t	Optimization of static assembly resources				Optimization of dynamic interaction			
	S	LE	SI	C_r	S	LE	SI	C_r
70	11	0.31	0.72	70	11	0.74	0.49	63
90	5	0.53	0.53	90	5	0.82–0.84	0.23–0.31	87–90
120	4	0.5	0.29	120	4	0.76–0.79	0.35–0.43	103
150	3	0.53	0.24	150	3	0.89	0.24	107

Under the control of the optimization of the dynamic assembly process with AFPA, the product assembly process was rendered relatively stable. The values of LE, SI, and C_r remained essentially unchanged or fluctuated within a fixed range of values for different cycles.

During the assembly of TPCLP, cooperative assembly was achieved by comprehensively applying AFPA and AQGA. Table S15 shows that compared with several existing assembly lines, for the different assembly cycles, the collaborative assembly was able to achieve a dynamic interaction between robots with fewer robots, so it could improve the assembly efficiency and reduce the assembly cost.

To analyze the compatibility of the collaborative assembly, another kind of TPCLP, called type 2, was introduced in the simulation (the first kind of TPCLP discussed above is called type 1). Compared to the assembly of type 1, type 2 had the same assembly number but different kinds of assembly tasks, as shown in Fig. S16.

In the simulation, the collaborative assembly line AT_line was compared with the existing traditional assembly lines in terms of compatibility of the assembly process. The results are shown in Fig. 5.

According to Fig. 5, in the different assembly cycles, the Cmp of the basic assembly methods (ST_line, TS_line, and U_line) was much larger or smaller than 1. The most significant value was 2.767, and the smallest value was 0.119. However, the Cmp of AT_line was close to 1. The most substantial value was 1.038, and the smallest value was 0.958. Hence, with the dynamic interaction of AFPA, the collaborative assembly line had better compatibility.

8 Conclusions

In this paper, we propose the collaborative assembly of LVEAs. Three objectives were considered: (1) self-government of robots; (2) assembly balance of the collaborative assembly; (3) assembly planning and dynamic interaction of the collaborative assembly. The results are summarized as follows:

1. Through the construction of the CPN collaborative assembly model, the self-government of robots was analyzed in the collaborative assembly. As shown in Figs. S8 and S9, with the different confidence intervals (90%, 95%, and 99%), all of the assembly robots completed the tasks with a relatively stable number. At five different numbers of simulation batches, each robot completed the tasks with a relatively stable number. Hence, the robots had good performance to avoid assembly confliction in the collaborative assembly. We also found that when one robot was in fault, the assembly process was not interrupted. The robot had good self-government ability to improve the passive fault tolerance of the assembly process.

2. AQGA applied to balance the collaborative assembly had better convergence performance than the

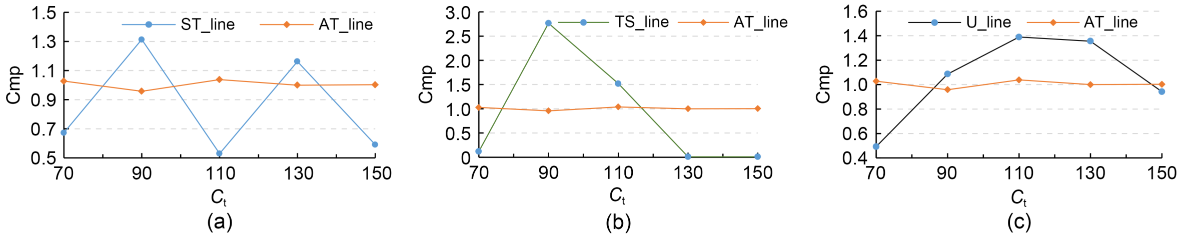

Fig. 5 Cmp comparison among different assembly methods: (a) ST_line vs. AT_line; (b) TS_line vs. AT_line; (c) U_line vs. AT_line

compared algorithms GA, GASA, RQGA, and QGA. Furthermore, with the optimization of AQGA, the Fitness, LE, SI, and *S* values obtained were better than those optimized with the compared algorithms. Hence, AQGA had better performance in collaborative assembly balance.

3. AFPA was proposed to perform assembly planning and dynamic interaction of the collaborative assembly. Figs. S11–S14 showed that in the application of AFPA, the collaborative assembly method could achieve assembly planning and dynamic interaction in the LVEA assembly. Furthermore, compared with the traditional assembly line, the collaborative assembly method achieved better LE, SI, *S*, and Cmp.

Thus, we can conclude that collaborative assembly is effective in the assembly of LVEAs.

Considering an actual assembly situation, however, there are two research directions for LVEA collaborative assembly. First, the collaborative assembly has passive fault tolerance, but when the assembly process is interrupted by multiple faults, the efficiency of LVEA assembly will be affected. Therefore, the active fault tolerance of the collaborative assembly needs to be studied. Second, in LVEA collaborative assembly, the control of assembly robots is distributed. Therefore, the distributed interactive communication method needs to be researched.

Contributors

Huanpei LYU, Libin ZHANG, and Dapeng TAN proposed the idea. All the authors designed the research. Huanpei LYU drafted the paper. Libin ZHANG, Dapeng TAN, and Fang XU revised and finalized the paper.

Compliance with ethics guidelines

Huanpei LYU, Libin ZHANG, Dapeng TAN, and Fang XU declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available within the article and its supplementary materials.

References

André É, Benmoussa MM, Choppy C, 2016. Formalising concurrent UML state machines using coloured Petri nets. *Form Aspects Comput*, 28(5):805-845. https://doi.org/10.1007/s00165-016-0388-9

Battaïa O, Dolgui A, 2013. A taxonomy of line balancing problems and their solution approaches. *Int J Prod Econ*, 142(2): 259-277. https://doi.org/10.1016/j.ijpe.2012.10.020

Baybars İ, 1986. A survey of exact algorithms for the simple assembly line balancing problem. *Manag Sci*, 32(8):909-932. https://doi.org/10.1287/mnsc.32.8.909

Baykasoglu A, 2006. Multi-rule multi-objective simulated annealing algorithm for straight and U type assembly line balancing problems. *J Intell Manuf*, 17(2):217-232. https://doi.org/10.1007/s10845-005-6638

Charbonnier F, Alla H, David R, 1999. Discrete-event dynamic systems. *IEEE Trans Contr Syst Technol*, 7(2):175-187. https://doi.org/10.1109/87.748144

Chen ST, Tan DP, 2018. A SA-ANN-based modeling method for human cognition mechanism and the PSACO cognition algorithm. *Complexity*, 2018:6264124. https://doi.org/10.1155/2018/6264124

Çil ZA, Mete S, Özceylan E, et al., 2017. A beam search approach for solving type II robotic parallel assembly line balancing problem. *Appl Soft Comput*, 61:129-138. https://doi.org/10.1016/j.asoc.2017.07.062

Deb K, 1998. Genetic algorithm in search and optimization: the technique and applications. Proc Int Workshop on Soft Computing and Intelligent Systems, p.58-87.

Desel J, Reisig W, 1998. Place/transition Petri nets. Proc Advanced Course on Petri Nets, p.122-173. https://doi.org/10.1007/3-540-65306-6_15

Gao J, Sun LY, Wang LH, et al., 2009. An efficient approach for type II robotic assembly line balancing problems. *Comput Ind Eng*, 56(3):1065-1080. https://doi.org/10.1016/j.cie.2008.09.027

Ge M, Ji SM, Tan DP, et al., 2021. Erosion analysis and experimental research of gas-liquid-solid soft abrasive flow polishing based on cavitation effects. *Int J Adv Manuf Technol*, 114(11-12):3419-3436. https://doi.org/10.1007/s00170-021-06752-w

Grzechca W, 2014. Assembly line balancing problem with reduced number of workstations. *IFAC Proc Vol*, 47(3):6180-6185. https://doi.org/10.3182/20140824-6-ZA-1003.02530

Jensen K, 1990. Coloured petri nets: a high level language for system design and analysis. Proc Int Conf on Application and Theory of Petri Nets, p.342-416. https://doi.org/10.1007/3-540-53863-1_31

Ji SM, Weng XX, Tan DP, 2012. Analytical method of softness abrasive two-phase flow field based on 2D model of LSM. *Acta Phys Sin*, 61(1):010205 (in Chinese). https://doi.org/10.7498/aps.61.010205

Johannsmeier L, Haddadin S, 2017. A hierarchical human-robot interaction-planning framework for task allocation in collaborative industrial assembly processes. *IEEE Robot Autom Lett*, 2(1):41-48. https://doi.org/10.1109/LRA.2016.2535907

Khatib O, 1986. Real-time obstacle avoidance for manipulators and mobile robots. *Int J Robot Resh*, 5(1):90-98. https://doi.org/10.1177/027836498600500106

Levitin G, Rubinovitz J, Shnits B, 2006. A genetic algorithm for robotic assembly line balancing. *Eur J Oper Res*, 168(3): 811-825. https://doi.org/10.1016/j.ejor.2004.07.030

Li DL, 2004. Electrical Control & the Principle and Application of PLC. Publishing House of Electronics Industry, Beijing, China (in Chinese).

Li L, Lu JF, Fang H, et al., 2020. Lattice Boltzmann method for fluid-thermal systems: status, hotspots, trends and outlook. *IEEE Access*, 8:27649-27675. https://doi.org/10.1109/ACCESS.2020.2971546

Li L, Tan DP, Yin ZC, et al., 2021. Investigation on the multiphase vortex and its fluid-solid vibration characters for sustainability production. *Renew Energy*, 175:887-909. https://doi.org/10.1016/j.renene.2021.05.027

Li SY, Li PC, 2006. Quantum genetic algorithm based on real encoding and gradient information of object function. *J Harbin Inst Technol*, 38(8):1216-1218, 1223 (in Chinese). https://doi.org/10.3321/j.issn:0367-6234.2006.08.002

Li XL, Xing KY, Lu QC, 2021. Hybrid particle swarm optimization algorithm for scheduling flexible assembly systems with blocking and deadlock constraints. *Eng Appl Artif Intell*, 105:104411. https://doi.org/10.1016/j.engappai.2021.104411

Montiel O, Sepúlveda R, Orozco-Rosas U, 2015. Optimal path planning generation for mobile robots using parallel evolutionary artificial potential field. *J Intell Robot Syst*, 79(2): 237-257. https://doi.org/10.1007/s10846-014-0124-8

New S, 1994. Modeling and analysis of manufacturing systems. *J Oper Res Soc*, 45(6):725-726. https://doi.org/10.1057/jors.1994.112

Nilakantan JM, Ponnambalam SG, Jawahar N, et al., 2015. Bio-inspired search algorithms to solve robotic assembly line balancing problems. *Neur Comput Appl*, 26(6):1379-1393. https://doi.org/10.1007/s00521-014-1811-x

Özcan U, Toklu B, 2009. A new hybrid improvement heuristic approach to simple straight and U-type assembly line balancing problems. *J Intell Manuf*, 20(1):123-136. https://doi.org/10.1007/s10845-008-0108-2

Pan Y, Ji SM, Tan DP, et al., 2020. Cavitation-based soft abrasive flow processing method. *Int J Adv Manuf Technol*, 109(9):2587-2602. https://doi.org/10.1007/s00170-020-05836-3

Ren CX, Zhang H, Fan YZ, 2015. Optimizing dispatching of public transit vehicles using genetic simulated annealing algorithm. *J Syst Simul*, 17(9):2075-2077, 2081 (in Chinese). https://doi.org/10.3969/j.issn.1004-731X.2005.09.008

Rizwan M, Patoglu V, Erdem E, 2020. Human robot collaborative assembly planning: an answer set programming approach. *Theory Pract Log Program*, 20(6):1006-1020. https://doi.org/10.1017/S1471068420000319

Rubinovitz J, Bukchin J, Lenz E, 1993. RALB: a heuristic algorithm for design and balancing of robotic assembly lines. *CIRP Ann*, 42(1):497-500. https://doi.org/10.1016/S0007-8506(07)62494-9

Samouei P, Fattahi P, Ashayeri J, et al., 2016. Bottleneck easing-based assignment of work and product mixture determination: fuzzy assembly line balancing approach. *Appl Math Modell*, 40(7-8):4323-4340. https://doi.org/10.1016/j.apm.2015.11.011

Tan DP, Chen ST, Bao GJ, et al., 2018. An embedded lightweight GUI component library and ergonomics optimization method for industry process monitoring. *Front Inform Technol Electron Eng*, 19(5):604-625. https://doi.org/10.1631/FITEE.1601660

Tavakoli A, 2020. Multi-criteria optimization of multi product assembly line using hybrid tabu-SA algorithm. *SN Appl Sci*, 2(2):151. https://doi.org/10.1007/s42452-019-1863-8

Wang H, Chen Z, Huang JH, et al., 2022. Development of high-speed on–off valves and their applications. *Chin J Mech Eng*, 35(1):67. https://doi.org/10.1186/s10033-022-00720-5

Wang JX, Gao SB, Tang ZJ, et al., 2023. A context-aware recommendation system for improving manufacturing process modeling. *J Intell Manuf*, 34:1347-1368. https://doi.org/10.1007/s10845-021-01854-4

Wang YY, Zhang YL, Tan DP, et al., 2021. Key technologies and development trends in advanced intelligent sawing equipments. *Chin J Mech Eng*, 34(1):30. https://doi.org/10.1186/s10033-021-00547-6

Xie N, 2006. Research on Modeling, Scheduling and Controller of Reconfigurable Manufacturing System Using Petri Nets. PhD Thesis, Tongji University, Shanghai, China (in Chinese).

Yang JN, Li B, Zhang ZQ, 2003. Research of quantum genetic algorith and its application in blind source separation. *J Electron*, 20(1):62-68. https://doi.org/10.1007/s11767-003-0089-4

Yu MY, Yang JJ, 2018. Research on flexible assembly system for multi-variety and small-batch products. *Mech Eng Autom*, (6):39-41 (in Chinese).

Zelenka J, 2010. Discrete event dynamic systems framework for analysis and modeling of real manufacturing system. Proc 14th Int Conf on Intelligent Engineering System, p.287-291. https://doi.org/10.1109/INES.2010.5483829

Zeng X, Ji SM, Jin MS, et al., 2014. Investigation on machining characteristic of pneumatic wheel based on softness consolidation abrasives. *Int J Prec Eng Manuf*, 15(10):2031-2039. https://doi.org/10.1007/s12541-014-0560-1

Zhang JY, Liu T, 2007. Optimized path planning of mobile robot based on artificial potential field. *Acta Aeronaut Astronaut Sin*, 28(S1):S183-S188 (in Chinese). https://doi.org/10.3321/j.issn:1000-6893.2007.z1.033

Zhang K, Zhang JH, Gan MY, et al., 2022. Modeling and parameter sensitivity analysis of valve-controlled helical hydraulic rotary actuator system. *Chin J Mech Eng*, 35(1):66. https://doi.org/10.1186/s10033-022-00737-w

Zheng SH, Yu YK, Qiu MZ, et al., 2021. A modal analysis of vibration response of a cracked fluid-filled cylindrical shell. *Appl Math Modell*, 91:934-958. https://doi.org/10.1016/j.apm.2020.09.040

List of supplementary materials

Fig. S1 Main content of this paper

Fig. S2 Architecture of the collaborative assembly methodology

Fig. S3 Definition of the artificial potential field (APF)

Fig. S4 Formal definition of the colored Petri net (CPN)

Fig. S5 Setting of places in the CPN collaborative assembly model

Fig. S6 Setting of model transitions in the CPN collaborative assembly model

Fig. S7 Relationship between the tasks required for a TPCLP

Fig. S8 Average number of assembly tasks of the assembly robots for different confidence intervals

Fig. S9 Number of assembly tasks for each assembly robot