Modular Analysis of Complex Products Based on Hybrid Genetic Ant Colony Optimization in the Context of Industry 4.0

Yichun Shi[1]*, Qinhe Shi[2]
Department of Information Engineering, Jiangsu Union Technical Institute, Nanjing, 210000, China[1]
School of Accounting, Nanjing University of Finance and Economics, Nanjing, 210000, China[2]

***Abstract*—With the development of science and technology, industrial construction has entered the era of 4.0 intelligent construction, and various algorithms have been widely applied in the modularization of production products. This study focuses on the modular optimization problem of complex products and establishes a hybrid genetic algorithm based on the ant colony algorithm framework. The new algorithm incorporates visibility analysis of the genetic algorithm, using the obtained solution as the pheromone source for the new algorithm to quickly obtain the optimal solution. The results showed that the algorithm could quickly achieve modularization of complex industrial products, adapt to products with a large number of parts and complex compositions, and obtain the optimal solution. The new algorithm reduced the running time of modular complex products by 35.06% compared to the particle swarm optimization algorithm. The new algorithm optimized the product design process for core components, reducing production costs by 23.46% and increasing production efficiency by 39.20%. Consequently, the novel algorithm modularizes complex products, thereby enhancing production efficiency and providing a novel intelligent method for the design process of complex products.**

Keywords*—Industry 4.0; genetic algorithm; ant colony; complex products; modularization; production efficiency

I. Introduction

In recent years, technological advancements have made traditional manufacturing unable to meet the current society's production needs for complex products [1]. Industry 4.0 is not only an upgrade to traditional supply chain automation and monitoring, but also builds a highly interconnected ecosystem and intelligent driven products through deep integration of intelligent technology [2]. The core concept is to utilize advanced information and communication technology to promote the transformation of the manufacturing industry towards a more flexible, efficient, and personalized intelligent manufacturing model [3]. In this context, the hybrid Genetic Algorithm (GA) has become a powerful tool for solving multidimensional and nonlinear optimization problems in the design of Complex Products Modularization (CPM) due to its unique advantages of strong global search capability, flexible genetic mechanism, and compatibility [4]. The objective of modular design is to disaggregate complex products into a series of relatively autonomous and functionally distinct modules. Through the combination and reconstruction of diverse modules, the design process can rapidly adapt to market demands, enhancing design efficiency and product flexibility [5].

In response to the demand from various sectors of society for the construction of industrial responsibility products in the context of Industry 4.0, a large number of scholars have conducted extensive research on CPM. Li Y et al. proposed a CPM method that combines modularity and design change propagation scope to reduce the impact of design change propagation. This method constructed the adjacency matrix of a weighted directed network model and solved the model using non-dominated sorting GA. The results of verifying the modularity of the driver's cab of a specific electronic sanitation vehicle showed that this method was practical and effective [6]. Wang X et al. proposed a moderated mediation model to address the issue of product modularization in R&D outsourcing practices. Based on survey data from 273 Chinese manufacturing enterprises, hypotheses were tested using hierarchical regression and PROCESS macro-models. The role of product modularization in R&D outsourcing practice was more effective when the trust level of R&D outsourcing partners was high [7]. Wang S et al. proposed a sub-item method oriented towards core components to address the challenges of structural modeling difficulties and unreasonable modular solutions in CPM for complex product sub-item problems. The new method simplified the structural model of complex products, further reduced the difficulty of modeling, and improved the efficiency of solving module partitioning schemes [8]. Forti A W et al. proposed a new structural matrix modularization method to solve the problem of difficult integration of multi-component products in the automotive industry, which combines the use of quality function deployment and design attribute matrix indication matrix. This systematic modular process has effectively played a role, making cross-functional teamwork easier [9].

The application value of hybrid algorithms as an efficient optimization strategy in CPM design is becoming increasingly prominent. Zhao J et al. proposed a multi-ACA approach that combines community relationship networks to address the challenge of balancing solution accuracy and convergence speed in large-scale TSP for the Ant Colony Algorithm (ACA). It improved the accuracy of the solution by collecting the route information of all ants and constructing a route relationship network. The performance of the new algorithm in large-scale TSP was significantly better than other improved algorithms

[10]. Arasteh B et al. proposed a hybrid method based on a grey wolf optimization algorithm to solve the problems of poor CPM quality, low success rate, and limited stability in existing methods, to achieve sub-items of complex products. The new algorithm improved the clustering quality and outperformed other heuristic algorithms in terms of modularity and convergence speed [11]. Liu C et al. proposed an improved discrete imperialist competitive hybrid algorithm to address product design compatibility and quality issues, considering a nonlinear programming approach that maximizes the per capita contribution margin of reliability loss. This hybrid algorithm has improved the solution quality by 6%~17% and 5%~14% compared to GA and simulated annealing algorithms [12].

In summary, the combination of modularity and the scope of design change propagation effectively reduces the impact of design changes. The trust-based product modularization method has improved the modularization efficiency of R&D outsourcing. These methods provide multidimensional ideas for solving CPM. In terms of CPM, most algorithms have only implemented modularity, and there has not been much research on achieving high-efficiency production. Based on this, this study proposes a new hybrid genetic ACA. To adapt to the rapid modularization with fewer components, this study innovatively integrates operations such as mixing, crossover, and mutation into ACA to quickly obtain visibility. To better obtain the optimal solution, the genetic part of the solution is used as a pheromone to enable the ant colony to quickly obtain the optimal solution. Pheromones are chemical substances that are synthesized by ants in ACAs and serve to guide other ants in navigating their environment. The issue of information redundancy in multi-part problems has been addressed by the adoption of ant colony solving, which has supplanted genetic solving. This development has enabled the implementation of a novel algorithm capable of adapting to CPM in multi-part scenarios, thereby facilitating the enhancement of production efficiency for complex products. The article is divided into five sections in total. The first section introduces the research status and importance of CPM under the background of Industry 4.0. The second section elaborates on the framework of Hybrid Genetic Ant Colony Optimization (HGACO) algorithm and its application in CPM. The third section verifies the performance of HGACO algorithm through experiments and compares it with other algorithms for analysis. The fourth section discusses the article and provides personal insights and opinions. The fifth section summarizes the research conclusions and proposes future research directions.

II. METHODS

A. Establishment of Hybrid Genetic Ant Colony Algorithm

In the era of Industry 4.0, traditional design methods cannot meet modern needs. This study designs an innovative HGACO method based on the ACA framework. Through a unique encoding and decoding mechanism, HGACO optimizes the decomposition process of product components. Simplifying complex products will be beneficial for industrial production. GA solving product modularization problems requires the use of various genetic components on chromosomes. The dimensions of each component are represented by chromosome length, forming an initial group [13]. The size of the fitness function determines the quality of an individual. The fitness function is transformed from the objective minimum function, as shown in Eq. (1).

$$F(t) = \frac{1}{Q_m} \tag{1}$$

In Eq. (1), $F(t)$ is the fitness function. Q_m is the shortest path. The formula for selecting arithmetic factors using the turntable method is shown in Eq. (2).

$$F(t) = \begin{cases} f(t) - C_{\max}, f(t) > C_{\max} \\ 0, f(t) = C_{\max} \\ C_{\max} - f(t), f(t) < C_{\max} \end{cases} \tag{2}$$

In Eq. (2), $f(t)$ and $C_{\max}$ are the values and maximum values of the fitness function. The initial population uses a random method to generate multiple chromosomes. The size of chromosomes represents the size of a population. Genetic individuals are screened, and the selected excellent individuals are subjected to subsequent crossover mutations. The selection method adopts the turntable method, and the probability of being selected is determined by the ratio size of individuals, as shown in Eq. (3).

$$P_a = \frac{f_a}{\sum_{a=1}^{N_z} f_a} \tag{3}$$

In Eq. (3), P_a is the probability of being selected, N_z is the individual fitness value, and $\sum_{a=1}^{N_z} f_a$ is the population fitness value. The selected excellent individuals are subjected to crossover operations to generate new offspring individuals. The crossover probability is shown in Eq. (4).

$$P_b = \begin{cases} \dfrac{\alpha(f_d - f_b^d)}{f_d - f_p}, f_b^d \geq f_p \\ \beta, f_b^d < f_p \end{cases} \tag{4}$$

In Eq. (4), P_b is the crossover probability. f_d and f_p are the maximum and average fitness values of the population. f_b^d is the party with a higher fitness value participating in the crossover process. α and β are constants. After selection and crossover, to enhance the comprehensive performance of the algorithm's retrieval ability, mutation is also required. The mutation probability is shown in Eq. (5).

$$P_c = \begin{cases} \dfrac{\gamma(f_d - f_b)}{f_d - f_p}, f_b \geq f_p \\ \lambda, f_b < f_p \end{cases} \tag{5}$$

In Eq. (5), P_c is the mutation probability, f_b is the fitness value of the mutated individual, and γ and λ are constants. The calculation process of GA is shown in Fig. 1.

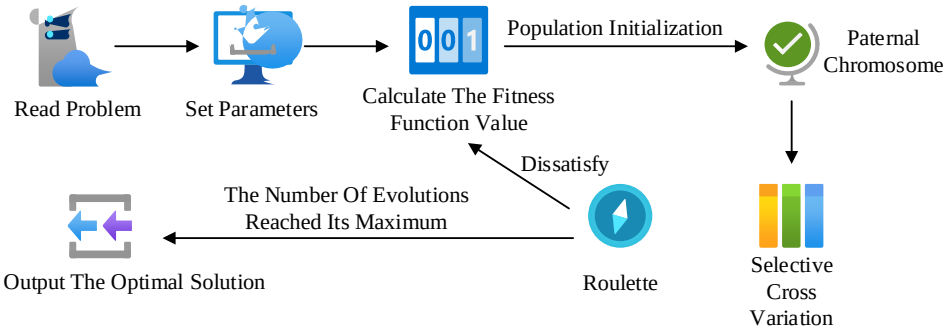


Fig. 1. Ga solution flow chart.

In Fig. 1, in the overall process of GA solving, step 1 is to read the problem being solved and set the corresponding parameters. Step 2 is to set the fitness function to calculate the fitness value, and initialize the population to produce parental chromosome individuals. The next step is to operate on individuals based on the selection, crossover, and mutation probability formulas, using the turntable method to select the transformed individuals. If the individual has reached the maximum number of evolutions, the solution value is directly output, otherwise it returns to the mutation step. GA is integrated into ACA, which is an exploratory algorithm with features of group cooperation, simultaneous computation, and forward selection. Each ant moves according to a fixed rule [14]. The number of ants is shown in Eq. (6).

$$N = \sum_{a=1}^{n} M_a(t) \tag{6}$$

In Eq. (6), N is the number of ants. $M_a(t)$ is the number of ants at time t. The information of each path of ants during the transfer process varies, and the probability calculation method of ant movement is shown in Eq. (7).

$$P_{ab}^{x} = \begin{cases} \dfrac{\tau_{ab}^{p}\eta_{ab}^{q}}{\sum_{S \in xl_x} \tau_{as}^{p}\eta_{as}^{q}}, b \in xl_x \\ 0, b \notin xl_x \end{cases} \tag{7}$$

In Eq. (7), P_{ab}^{x} is the probability that ant x moves from point t to point b at time a. τ_{ab}^{p} is the number of pheromones from point a to point b under the information heuristic factor p. η_{ab}^{q} is the visibility from point a to point b. xl_x is a point that ants have not passed through. After the ant completes a path selection, the pheromones on other paths are shown in Eq. (8) [15].

$$\tau_{ab}t = \varphi \cdot \tau_{ab}t + \Delta\tau_{ab}t \tag{8}$$

In Eq. (8), $\tau_{ab}t$ is the number of pheromones from point a to point b after the t-th cycle. φ is the residual coefficient of pheromones. $\Delta\tau_{ab}t$ is the residual value of pheromones. The pheromone changes of Ant x from point a to point b during a path iteration are shown in Eq. (9).

$$\Delta\tau_{ab}^{x} = \begin{cases} \dfrac{k}{H(C)}, x \text{ participation loop} \\ 0, \text{else} \end{cases} \tag{9}$$

In Eq. (9), $\Delta\tau_{ab}^{x}$ is the pheromone change of ant x from point a to point b in one path iteration. k is the intensity of pheromones. $H(C)$ is the sum of ant journeys after multiple iterations. Fig. 2 shows the calculation process of ACO.

In Fig. 2, the system first initializes the data by randomly placing ants at various points and then begins the loop. According to the formula for calculating the probability of ant movement, the ants are moved. The points that ants pass through are marked. The ant colony cycle model is used to determine whether ants have passed through all points. If they have not passed through all points, the number of cycles is increased by 1. If the ant passes through all points, the pheromone is updated according to the pheromone calculation formula. If the calculated path is the shortest path, the path is updated; otherwise, the ant is asked to select a new point and output the shortest path. ACA lacks pheromones in the early stage, so the process of searching for the optimal solution is relatively slow. GA has high adaptability and fast running speed, but it is prone to generating redundant junk information. By integrating two algorithms, GA can solve the problem of the slow operation of ACA in the early stage. In the later stage, the accumulation of pheromones in ACA reaches a certain level, which increases the running speed and avoids the situation where GA continues to run and generate a large amount of junk information. The key to mixing GA and ACA is to find the time point for mixing. The appropriate timing can enable HGACO to achieve optimal performance. The process of the HGACO algorithm is shown in Fig. 3.

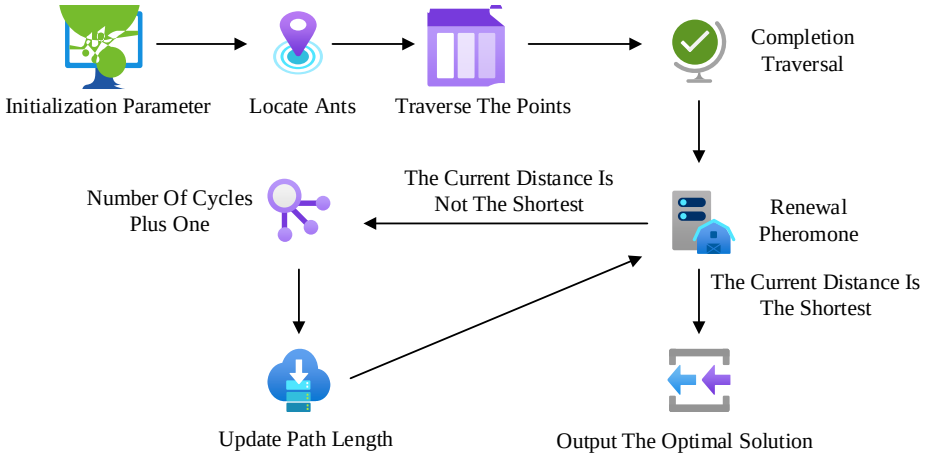


Fig. 2. Flow chart of ACA solution.

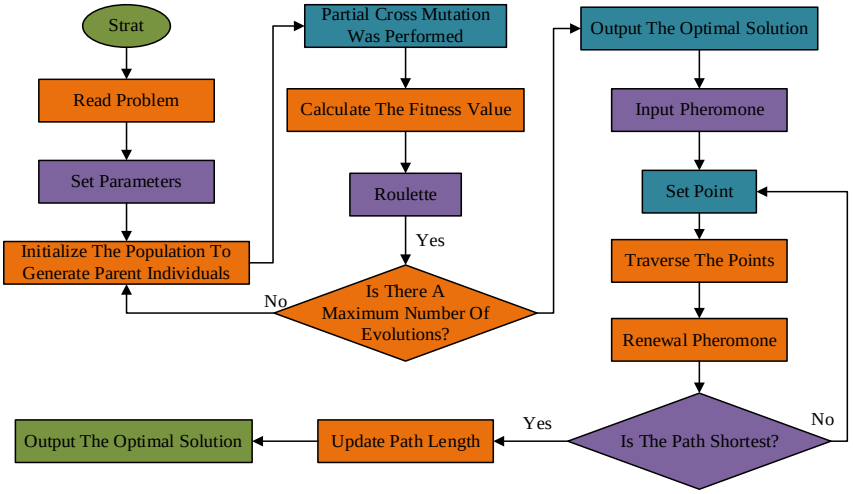


Fig. 3. Flow chart of HGACO.

In Fig. 3, the HGACO algorithm first calculates the optimal solution through the GA part, converting the optimal solution into pheromones while leaving some visibility information. Visibility information refers to the quality information of solutions in GAs, which is used to guide ACAs to quickly find the optimal solution. The visibility data are numbered to represent the spatial numbering in the CPM problem-solving process. The pheromones transformed into the optimal solution are transmitted to the ACA section, and the optimal solution is obtained through the mechanism of simultaneous calculation in the ACA section. Finally, the optimal solution obtained by the HGACO algorithm is outputted. By utilizing the HGACO algorithm, modular processing of complex products can assist in production design. Complex products often have a large number of parts. Modular processing can simplify these parts and reduce the complexity of encoding and decoding.

B. CPM Design

The use of HGACO algorithm for CPM can solve the production efficiency problem in the context of Industry 4.0. The main factor affecting production efficiency is that product quality requirements are often high, and most products are customized with complex structures [16]. These reasons make product production relatively slow in the design process. After disassembling the information on these customized products, the indicators of each part can be obtained. The various components are interrelated. Fig. 4 is a complex product decomposition structure diagram.

In Fig. 4, the internal relationships within the structure of complex products are complex. Various large devices are composed of different components. Components are composed of different components, and components are composed of multiple part modules, which contain multiple small parts. The components at different levels are connected in a complex manner through serial and parallel connections. The logistics list is used to organize components at different levels, determine their mutual demand relationships, and thus form integration. The adaptation relationship between different components will affect the quality performance of the product. There are various indicators of product quality. Reliability is the most important quality indicator. Fig. 5 shows the serial structure between the parts.

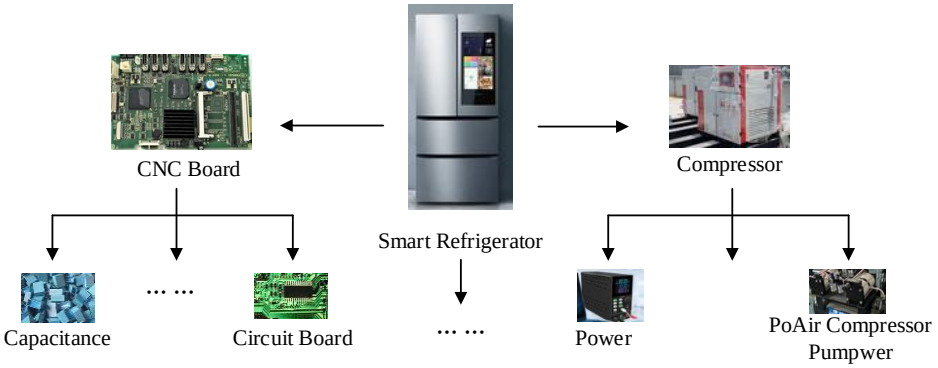


Fig. 4. Complex product breakdown structure diagram.

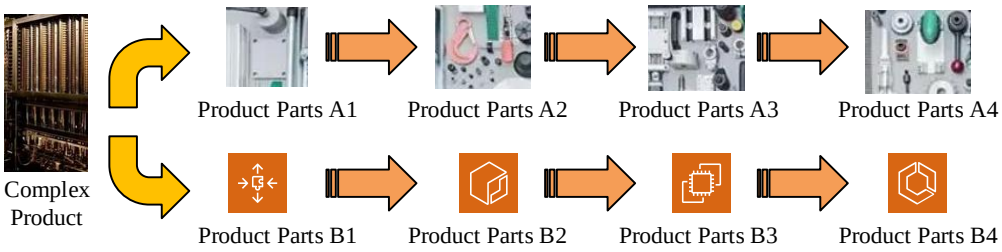


Fig. 5. Series system result model diagram.

In Fig. 5, A1, A2, A3, A4, B1, B2, B3, and B4 respectively represent the component numbers. In a serial system, each component module is sequentially connected to form a linear sequence. The reliability of each component module directly affects the performance of its subsequent modules. Consequently, the overall reliability of the system depends on the reliability of each component module. The characteristic of this structure is that the failure of any component module may lead to the failure of the entire system. Multiple serial components together form a component module. In actual production, these component modules form a propagation mode of parallel connections. Parallel connected components further form components, and most of the components are interleaved in series and parallel. Among them, the reliability between component modules affects each other. The overall reliability of complex products is shown in Eq. (10).

$$D(x) = K(X_1 > x, X_2 > x, ...X_m > x) = \prod_{a=1}^{m} K(X_a > x) = \prod_{a=1}^{m} D_a(x) \quad (10)$$

In Eq. (10), $D(x)$ represents the overall reliability of the product system. K is the reliability coefficient. X_m is the reliability prediction of the m-th component. x is the reliability standard. $\prod$ is a quadrature sum operation. The reliability of the system decreases with the increase of the number of components, and the decrease in system reliability is reflected in the failure rate [17]. The lower the failure rate of components, the higher the reliability of the system. The calculation method for failure rate is shown in Eq. (11).

$$\eta_m = \sum_{a=1}^{m} \eta a \quad (11)$$

In Eq. (11), η_m is the overall failure rate of the system. ηa is the failure rate of each component. Based on the calculation method of failure rate, the average time for system failure is shown in Eq. (12).

$$T_{bf} = \frac{1}{\eta_m} = \frac{1}{\sum_{a=1}^{m} \eta a} \quad (12)$$

In Eq. (12), T_{bf} is the average time between failures, which is inversely proportional to the overall failure rate of the system. If one of the components is adjusted and integrated into parallel mode, the overall reliability calculation method of the system will be adjusted as shown in Eq. (13).

$$D = D_1 \times [1-(1-D_2)^n] \times \prod_{a=3}^{m} D_a, n \in (1, \infty) \quad (13)$$

In Eq. (13), D is the system reliability with parallel features, and $D \in (0,1)$. D_a is the reliability of the a-th component. n is the total number of parts. The calculation method of the first derivative of the reliability of the system as a whole based on the number of parts is given by Eq. (14).

$$\frac{dD}{dn} = -(1-D_2)^n \ln(1-D_2) D_1 \prod_{a=3}^{m} D_a > 0 \quad (14)$$

In Eq. (14), $\frac{dD}{dn}$ is the first derivative of system reliability. With parallel connections in the system, as the number of parts increases, the system becomes significantly more reliable. The expression for the second derivative is given by Eq. (15).

$$\frac{d^2D}{dn^2} = -(1-D_2)^n \ln^2(1-D_2) D_1 \prod_{a=3}^{m} D_a < 0 \quad (15)$$

Complex products often contain a large number of

component modules. Different component modules form various complex system structures. Serial and parallel modes are often mixed together, making the factors affecting system reliability more complex [18]. There is interaction between the components of the product. The impact of interaction is called the influencing factor. There is no interactive influence relationship between the components that make up the parallel state, and there is no subsequent performance impact. The influence relationship between the components that form a mixed state of serial and parallel is relatively complex, and when improving the system, it is greatly affected by the interference variables of the components. The coupling between different levels can improve the overall quality of the system by selecting the quality, quantity, and shape of parts [19]. Based on the analysis of serial and parallel quality performance of parts, this study uses the HGACO algorithm to integrate component modules in a hybrid structure and analyze the impact mechanisms of multiple component systems in the hybrid structure. By reallocating different components and configuring quality performance parameters, more precise and improved system modules are obtained [20]. Fig. 6 shows the technical process of using the HGACO algorithm for CPM.

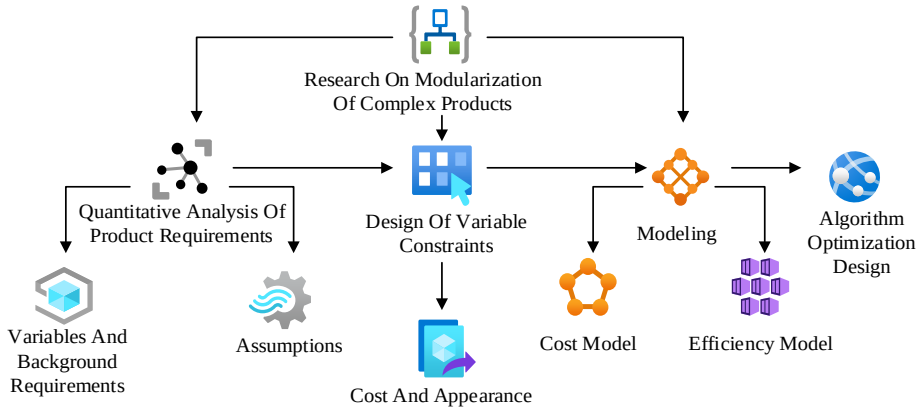


Fig. 6. Technical flow chart of modular design of complex products.

In Fig. 6, the first step is to conduct a quantitative analysis of product requirements, including functional description, size, and cost requirements. Subsequently, design constraints for variables such as cost and appearance. During the design process, it is also necessary to explain the constraints. After obtaining a series of required parameters, a basic cost and profit model for complex products is constructed. The study is predicated on the fundamental model to optimize the design of the HGACO algorithm. Ultimately, the HGACO algorithm is utilized for calculation to obtain the optimal functional module configuration scheme.

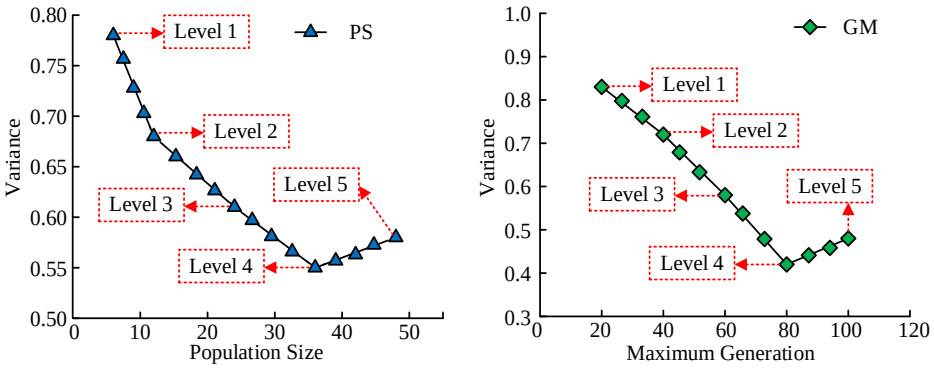
III. Results

A. Performance Analysis of Mixed Genetic ACA

Simulation experiments are designed to analyze the performance of the HGACO algorithm and solve the CPM analysis problem. The experimental language is Java, and the model solving is Matlab. The operating environment is Windows 10, the CPU is Intel Core i7-2600 3.40 Ghz, and the RAM is 4GB. The performance of an algorithm mainly depends on its control parameters. The control parameters are mainly divided into Iterations (GM), Mutation Rate (PM), Population Numbers (PS), and Crossover Rate (PC). The Analysis of Variance (ANOVA) of the algorithm under different parameter controls reflects its performance, with smaller values indicating better performance and greater stability. Fig. 7 shows the variance of PS and GM corresponding to the HGACO algorithm at different levels.

The statistical significance of the variance data in Fig. 7 is shown in Table I. In Fig. 7 (a), PS increases with the increase of the number of levels. When PS is 36, the variance has its minimum value, and as PS continues to expand, there is no significant change in variance. In Fig. 7 (b), when GM is less than 80, the variance shows a stable downward trend, but when GM is greater than 80, the variance suddenly increases. This is because when the iteration parameters are too high, the HGACO algorithm is prone to lagging due to the concentration of pheromones, leading to an increase in ANOVA. Therefore, the optimal PS is 36 and the optimal GM is 80. There are crossover and mutation processes in the HGACO algorithm, and the corresponding PC and PM also have an impact on the variance. Fig. 8 shows the corresponding variances of PC and PM at different levels.

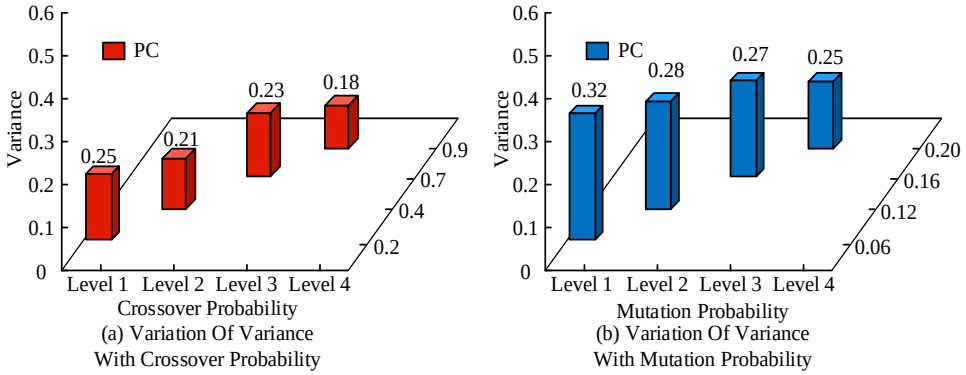
In Fig. 8 (a), the ANOVA transformation is less affected by PC, and the variance is minimized when PC is 0.90. In Fig. 8 (b), the variance is minimized when PM is 0.20. This is because cross-selection makes it easier to generate offspring with significant differences from the parent, while mutation operations have relatively less impact on offspring, so the mutation probability is less affected by hierarchy and has little effect on variance. The PC of the HGACO algorithm is 0.90, and the PM is 0.20. This condition has the best effect on generating the optimal population for the algorithm. Fig. 9 shows the initial values of the independent variables and the distribution of the independent variables after multiple iterations in the HGACO algorithm.



(a) Variation Of Variance With Population Size

(b) Variation Of Variance With Maximum Generation

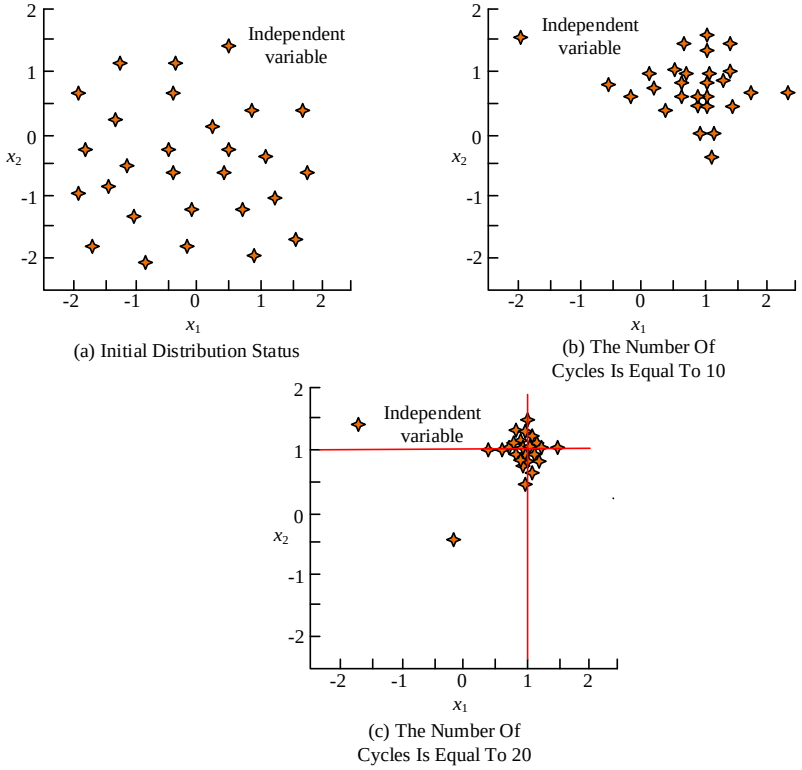
Fig. 7. The Variance of PS and GM of the algorithm at different levels.



(a) Variation Of Variance With Crossover Probability

(b) Variation Of Variance With Mutation Probability

Fig. 8. The Variance of PC and PM of the algorithm at different levels.



(a) Initial Distribution Status

(b) The Number Of Cycles Is Equal To 10

(c) The Number Of Cycles Is Equal To 20

Fig. 9. Distribution of independent variables under different cycles.

TABLE I STATISTICAL ANALYSIS OF VARIANCE DATA

PS		**GM**	
Population Size	variance	Population Size	variance
5	0.78	20	0.83
7	0.75	27	0.8
10	0.71	33	0.75
13	0.67	40	0.71
18	0.65	47	0.68
21	0.63	53	0.62
24	0.60	60	0.58
28	0.58	66	0.55
31	0.56	70	0.48
36	0.54	75	0.45
38	0.55	80	0.41
41	0.56	87	0.44
44	0.57	92	0.46
48	0.58	100	0.48
P	0.0001	P	0.00005
F	12.34	F	15.67
95% Confidence intervals	[0.59,0.67]	95% Confidence intervals	[0.50,0.72]
Standard deviations	0.07	Standard deviations	0.19

In Fig. 9 (a), the initial value distribution of the ant colony is quite scattered and needs to undergo a self changing cycle. In Fig. 9 (b), when the number of iterations reaches 10, the independent variable distribution of the HGACO algorithm begins to converge towards the vicinity of the first independent variable $x_1 = 1$ and the second independent variable $x_2 = 1$. In Fig. 9 (c), when the number of self-variable loops reaches 20, the independent variables of the algorithm converge well. The independent variables of the algorithm gradually converge and reach a convergence state. Continuing to increase the number of loops will cause an unnecessary burden on the computational part of the algorithm. Therefore, the optimal number of iterations for the HGACO algorithm is 20, which results in the best convergence. After determining that the HGACO algorithm can achieve optimal performance under the above parameter conditions, it is necessary to analyze other performance indicators of the algorithm. The most intuitive way to determine whether an algorithm is optimal is to compare its performance with other algorithms under the same conditions.

B. Performance Comparison of Modularization of Complex Products using different Algorithms

In the previous section, various optimal parameters are determined through performance analysis of the HGACO algorithm. To validate the performance of the HGACO algorithm, the Swarm Behavior Heuristic Algorithm (SBH) and the traditional Particle Swarm Optimization (PSO) algorithm are compared with the proposed HGACO algorithm. This experiment compares the algorithm performance under CPM using algorithms. The modularization process is to use different algorithms to modularize complex products with m parts and n associations, where $m \in \{10, 20, 30, 40, 50\}$ corresponds to $n \in \{20, 40, 60, 80, 100\}$ and the problem scale is represented as x_{mn}. The training and validation running times of each algorithm on the DSM dataset are displayed in Fig. 10.

In Fig. 10 (a), during the training process, the initial running time of the HGACO algorithm is relatively long. However, as the number of parts increases, the running time of HGACO is shorter than that of PSO, while the running time of SBH always remains linear. This is because HGACO has insufficient pheromones at the beginning, and the process of cross-selection and mutation takes a long time. However, with the extension of training and the accumulation of pheromones, the speed at which HGCAO seeks the optimal solution increases. In Fig. 10 (b), due to the training of HGACO, it can maintain a relatively short running time throughout the validation process. The average running time of HGACO has been reduced by 35.06% compared to PSO. Table I shows the maximum modular value standard deviation and mean of HGACO, SBH, and PSO during the modular process.

In Table II, when there are few parts, all algorithms can obtain reasonable solutions, with an average value generally within 10. As the data size increases, the average values obtained by PSO and SBH show significant discrepancies, with some solutions even reaching 48900, which is clearly unreasonable. Due to the different solving rules of the algorithms, the model solutions obtained by PSO and SBH do not meet the requirements when there are too many parts. Therefore, only when the optimal solution of HGACO is within a reasonable range in all cases, can it meet the requirements. The production efficiency, duration, and cost of modularizing four complex products C_1, C_2, C_3, and C_4 using three algorithms are shown in Fig. 11.

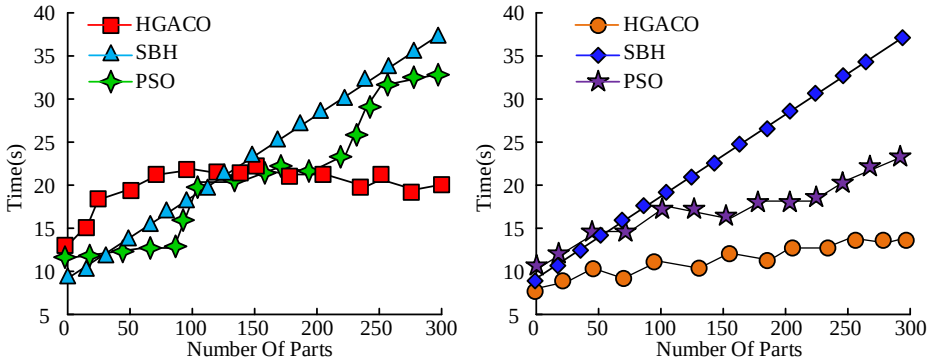


Fig. 10. Algorithm training and validation run time on DSM data set.

TABLE II COMPARISON OF DIFFERENT ALGORITHMS ON MODULARITY VALUES

	HGACO		**PSO**		**SBH**	
Problem Scale	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation
x_{1020}	5.87	4.52	4.52	6.25	5.65	6.58
x_{2040}	7.82	7.16	7.16	8.02	2158	1589
x_{3060}	6.21	5.55	145	165	23.1	18.5
x_{4080}	4.59	6.21	3.54	3.51	256	298
x_{50100}	6.51	8.51	48900	36580	10.1	9.99

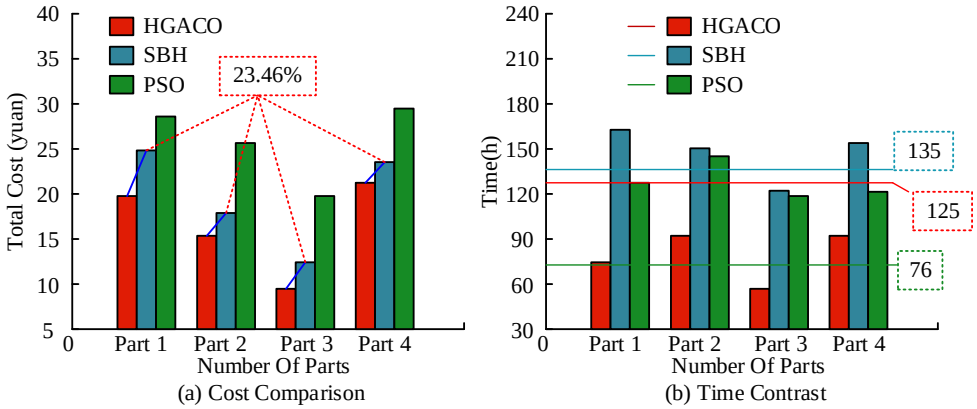


Fig. 11. Production data after modularization of complex products by algorithms.

In Fig. 11 (a), the production costs of modularizing different products using the HGACO algorithm are 19.98 yuan, 15.84 yuan, 9.21 yuan, and 21.02 yuan, respectively. The production costs after modularization using the SBH algorithm are 24.89 yuan, 17.94 yuan, 13.25 yuan, and 23.55 yuan, respectively. Compared with the SBH algorithm, the HGACO algorithm reduces the production costs of different products by 24.69%, 13.25%, 43.87%, and 12.03%, respectively, saving an average of 23.46% of production costs. In Fig. 11 (b), the optimal solution obtained by HGACO can significantly improve efficiency. After modularizing complex production, the average production time is 125 hours, while the average production time under the PSO algorithm is 76 hours. Compared with the ABC algorithm, the HGACO algorithm reduces production time by 39.20%, resulting in a 39.20% increase in production efficiency. This is because HGACO has made product design simpler and more efficient after CPM. Enterprises can carry out intelligent production based on simpler design solutions. Therefore, by modularizing complex production and solving it, HGACO can significantly improve the production efficiency of complex products and reduce production costs.

IV. DISCUSSION

With the advancement of Industry 4.0, CPM design has become a key means to improve production efficiency and reduce costs. The conventional manufacturing paradigm proves challenging in meeting the demands of intricate product development. Conversely, CPM exhibits a capacity to expeditiously adapt to market fluctuations, enhancing design efficiency and product adaptability. In recent years, numerous scholars have devoted themselves to studying CPM and proposed various algorithms, such as GA, ACA, and PSO. While these methods have proven advantageous in addressing modular problems, they are not without limitations. These

limitations include but are not limited to, insufficient applicability and low operational efficiency in multi-component complex products. To improve the operational efficiency of CPM, a CPM optimization method based on HGACO has been proposed. This method combines the global search capability of GA and the pheromone optimization mechanism of ACA. The transformation of the solution of GA into a pheromone, followed by its transmission to ACA, results in the rapid convergence and efficient optimization of the system. In the experimental section, the superiority of the HGACO algorithm is verified by comparing its performance with other algorithms. With respect to the duration of execution, the HGACO algorithm exhibits a longer execution time during the initial training stages. However, as the number of components increases, the execution time of the HGACO algorithm experiences a gradual decrease in comparison to the PSO algorithm. This is primarily due to the necessity for HGACO to amass a sufficient quantity of pheromones during the initial training phases to facilitate ant colony navigation. As pheromones are accumulated, the efficacy of the algorithm for search purposes undergoes a substantial enhancement. A comparison of modular values reveals that the HGACO algorithm can obtain reasonable modular values under different problem scales. In contrast, the PSO and SBH algorithms demonstrate significant deviations in the standard deviation and mean of modular values when there are a large number of components. This indicates that the HGACO algorithm has higher stability and adaptability when dealing with complex products. The underlying rationale pertains to the efficacy of the HGACO algorithm in circumventing local optima through the integration of crossover and mutation operations of GAs. This is complemented by the utilization of the pheromone mechanism of ACA, which facilitates the acceleration of global search, thereby ensuring the maintenance of optimal performance in complex environments.

In summary, the HGACO algorithm has shown significant advantages in the modular design of complex products. It not only outperforms traditional algorithms in terms of running time but also demonstrates excellent stability and adaptability in modular values. These results indicate that the HGACO algorithm can effectively address the challenges in CPM design. Despite the demonstrated efficacy of the HGACO algorithm in experimental and practical test results, its performance may be constrained by the increasing complexity of products and the scale of production systems that accompany the advancement of Industry 4.0. The performance of the HGACO algorithm is crucial in handling large-scale complex products. Future research needs to further enhance the scalability of algorithms, enabling them to efficiently handle large-scale complex products. By using distributed computing and parallel processing techniques, the computational tasks of algorithms can be allocated to multiple processors or computing nodes, significantly improving the execution efficiency of algorithms to meet more complex industrial needs.

V. Conclusion

This study mainly focused on the modular processing and analysis of complex products using algorithms in the context of Industry 4.0. A new HGACO algorithm has been proposed to further improve production efficiency. This study first extracted the problem, then initialized the parameters, and solved it through GA selection, crossover, and mutation. The obtained solution was utilized as a pheromone and visibility information to input into the ACA part to quickly obtain the optimal solution. Finally, experimental verification and comparison were conducted on the optimal solutions obtained by different algorithms. When the PS of the HGACO algorithm was 36, GM was 80, PC was 0.90, and PM was 0.20, the variance was minimized, indicating that the algorithm has the best performance at this time. When the number of cycles reached 20, HGACO just converged and could adapt to complex products with different numbers of parts, and could obtain reasonable optimal solutions. By using HGACO to modularize the responsible products, the running time was reduced by 35.06% compared to the PSO algorithm. HGACO has improved the production efficiency of complex products by 39.20% while reducing production costs by 23.46%. In summary, the comprehensive performance of HGACO is superior to other traditional algorithms, providing a theoretical basis for manufacturing enterprises to solve configuration problems and improve production efficiency for complex products. Although this study has solved the optimal solution problem of CPM, there are still some issues, such as the need to consider weighting for the requirements of different complex parts. Therefore, further research should be conducted on the multi-condition constraint weighting of the algorithm in the future.

References

[1] Khan A S. Multi-objective optimization of a cost-effective modular reconfigurable manufacturing system: An integration of product quality and vehicle routing problem. IEEE Access, 2021, 10(1): 5304-5326.

[2] Zuefle M, Krause D. Multi-Disciplinary Product Design and Modularization–Concept Introduction of the Module Harmonization Chart (MHC). Procedia CIRP, 2023, 119(1): 938-943.

[3] Mertens K G, Rennpferdt C, Greve E, Krause D, Meyer M. Reviewing the intellectual structure of product modularization: Toward a common view and future research agenda. Journal of Product Innovation Management, 2023, 40(1): 86-119.

[4] Lima M B, Kubota F I. A modular product design framework for the home appliance industry. The International Journal of Advanced Manufacturing Technology, 2022, 120(3): 2311-2330.

[5] Ameer M, Dahane M. NSGA-III-based multi-objective approach for reconfigurable manufacturing system design considering single-spindle and multi-spindle modular reconfigurable machines. The International Journal of Advanced Manufacturing Technology, 2023, 128(5-6): 2499-2524.

[6] Li Y, Ni Y, Zhang N, Liu Z. Modularization for the complex product considering the design change requirements. Research in Engineering Design, 2021, 32(4): 507-522.

[7] Wang X, Lee H, Park K, Lee G. The strategic role of R&D outsourcing practices and partners in the relationship between product modularization and new product development efficiency. Journal of Manufacturing Technology Management, 2024, 35(1): 185-202.

[8] Wang S, Li Z, He C, Liu D, Zou G Y. Core components-oriented modularisation methodology for complex products. Journal of Engineering Design, 2022, 33(10): 691-715.

[9] Forti A W, Ramos C C, Muniz Jr J. Integration of design structure matrix and modular function deployment for mass customization and product modularization: a case study on heavy vehicles. The International Journal of Advanced Manufacturing Technology, 2023, 125(3): 1987-2002.

[10] Zhao J, You X, Duan Q, Liu S. Multiple ant colony algorithm combining community relationship network. Arabian Journal for Science and Engineering, 2022, 47(8): 10531-10546.

[11] Arasteh B, Abdi M, Bouyer A. Program source code comprehension by module clustering using combination of discretized gray wolf and genetic algorithms. Advances in Engineering Software, 2022, 173: 103252.

[12] Liu C, Yang X, Wang J. Optimization of product line considering compatibility and reliability via discrete imperialist competitive algorithm. RAIRO-Operations Research, 2021, 55(6): 3773-3795.

[13] Hao J, Gao X, Liu Y, Han Z. Module division method of complex products for responding to user's requirements. Alexandria Engineering Journal, 2023, 82(1): 404-413.

[14] Arasteh B. Clustered design-model generation from a program source code using chaos-based metaheuristic algorithms. Neural Computing and Applications, 2023, 35(4): 3283-3305.

[15] Zhang Z, Lu B, Xu X, Shen X, Feng J, Brunauer G. CN-MgMP: a multi-granularity module partition approach for complex mechanical products based on complex network. Applied Intelligence, 2023, 53(14): 17679-17692.

[16] Lammers T, Guertler M, Skirde H. Can product modularization approaches help address challenges in technical project portfolio management? –Laying the foundations for a methodology transfer. International Journal of Information Systems and Project Management, 2022, 10(2): 26-42.

[17] Silva T, Santos C. Challenges of Product Modularization Methods in SMEs: Lessons Learned from a Manufacturer of Rigid Inflatable Boats. Proceedings of the Design Society, 2023, 3(1): 847-856.

[18] Persson M, Hsuan J, Hansen P K. Improving decision making in product modularization by game-based management training. Proceedings of the Design Society, 2021, 1(1): 1837-1846.

[19] Aryavalli S N G, Kumar G H. Futuristic Vigilance: Empowering Chipko Movement with Cyber-Savvy IoT to Safeguard Forests. Archives of Advanced Engineering Science, 2023, 1(8): 1-16.

[20] Monetti F M, Maffei A. Towards the definition of assembly-oriented modular product architectures: a systematic review. Research in Engineering Design, 2024, 35(2): 137-169.