Beifang Bao, Yu Yang*, Aijun Liu, Jiali Zhao and Leiting Li

Task Allocation Optimization in Collaborative Customized Product Development Based on Adaptive Genetic Algorithm

Abstract: Due to the currently insufficient consideration of task fitness and task coordination for task allocation in collaborative customized product development, this research was conducted based on the analysis of collaborative customized product development process and task allocation strategy. The definitions and calculation formulas of task fitness and task coordination efficiency were derived, and a multiobjective optimization model of product customization task allocation was constructed. A solution based on adaptive genetic algorithm was proposed, and the feasibility and effectiveness of the task allocation algorithm were tested and verified using a 5-MW wind turbine product development project as example.

Keywords: Collaborative customized product development, task allocation, task fitness, task coordination efficiency, adaptive genetic algorithm.

Subject classification codes: C931, TP391.

***Corresponding author: Yu Yang,** State Key Laboratory of Mechanical Transmission, Chongqing University, Chongqing 400030, China, e-mail: yuyang@cqu.edu.cn
Beifang Bao and Jiali Zhao: State Key Laboratory of Mechanical Transmission, Chongqing University, Chongqing 400030, China
Aijun Liu: Economy and Management School, Xidian University, Xi'an 710071, China
Leiting Li: Chongqing Academy of Science and Technology, Chongqing 401123, China

1 Introduction

With the fast development of market economy and the ever-increasing updating speed of products, customer demands are developing constantly into the direction of diversification and individuation, which challenges enterprises with extremely fierce competitions [10]. To pursue their favorable positions and development goals in the market, enterprises are keen to further shorten the product development cycle, reduce R&D cost, and improve production efficiency to obtain competitive advantages in terms of time, quality, and cost. All these require companies to come up with an innovative product design approach. Collaborative customized product development, serving as an innovative and potential design mode, combines the advantages of traditional design patterns and advanced design techniques. Under the premise of maximizing the economic benefits of enterprises and with the characteristics of modular design and parts standard, this mode ensures that enterprises can adapt to the market and to customer-oriented requirements more flexibly, and task allocation is one of the most important parts of the collaborative customized product development process. Reasonable task allocation can optimize the allocation of various development resources and make full use of them, leading to a smooth process of collaborative customized product development.

In terms of task allocation, there are many methods introduced by different scholars. Dianxun et al. [1] introduced a new kind of generalized particle model for the parallel optimization of enterprise resources and task allocation problem. Wunhwa and Chinshien [15] introduced a hybrid method combining tabu search algorithm with the noising method to solve the task allocation problems. Fernandez and Lamari [2] studied and discussed the processor number for continuous communication task allocation problem and introduced two kinds of accurate polynomial time algorithms. Min-Hyuk et al. [8] proposed a distributed task allocation algorithm for a team of robots with constraints on energy resources and operates in an unknown dynamic environment, with the objective of maximizing the task completion ratio while minimizing resource usage. Yin et al. [17] presented a multiobjective task allocation algorithm based on particle swarm optimization (PSO) in the presence of system constraints to achieve throughput maximization, reliability maximization, and cost minimization in a distributed computing system. Yung-Cheng et al. [18] proposed a task allocation algorithm for any parallel programs on a given machine configuration by traversing a state–space tree that enumerates all the possible task assignments and applying a pruning rule on each traversed state. Son et al. [11] proposed an efficient workflow task allocation method based on the locality principle in a distributed workflow system that used the concept of graph partitioning to improve the performance of workflow processing by minimizing remote processing costs. Tripathi et al. [13] presented a multiagent-based approach for the part allocation problems in flexible manufacturing systems to cope with the dynamic environment where four agents (communicator, machine, part, and material handling device) were involved in carrying out the tasks of allocating parts on different machines. Zhang et al. [19] designed the PSO algorithm for task optimization allocation in the suppliers' participation in collaborative product development. Jing et al. [4] constructed a task allocation bi-level programming model for suppliers' involvement in product collaborative development. In a comprehensive consideration of product quality, cost, information, and other factors of suppliers, Hou et al. [3] constructed a design task allocation optimization model for interfirm product collaborative development. Most of the above-mentioned literatures constructed task allocation models and introduced corresponding algorithms based on the analysis of the goals of task allocation, whereas the number of researches and discussions related to the relationship between personnel and tasks to optimize the task allocation process is relatively small. However, in the current mode, where customers and suppliers collaboratively participate in customized product development, the fitness of suppliers to the development tasks and the coordination efficiency between suppliers and other factors greatly influence customized product task allocation. Therefore, fitness and coordination efficiency in the customized product development model are very important in task allocation optimization research.

Based on the analysis of related research results of domestic and international task allocation, this article introduces the definitions and calculation methods of task fitness and task coordination efficiency. Based on these, a multiobjective optimization mathematical model of customized product task allocation is constructed. Finally, an adaptive genetic algorithm (AGA) for solving the model is proposed.

The rest of this article is organized as follows. The process and task allocation strategy of collaborative customized product development is analyzed in Section 2. The task allocation mathematics model in collaborative customized product development is established in Section 3. The multiobjective optimization algorithm for task allocation in collaborative customized product development is proposed in Section 4. In Section 5, a 5-MW wind turbine R&D project example is introduced to verify the feasibility and the efficiency of the task allocation algorithm. Finally, Section 6 concludes the paper.

2 Analysis of Collaborative Customized Product Development Process and Task Allocation Strategy

2.1 Analysis of Collaborative Customized Product Development Process

With the increasing degree of product customization, the scope and degree of collaboration among enterprises, suppliers, and the customers is also increasing. An increasing number of enterprises, suppliers, and customers participate in the customized product development design process of core enterprises, relying on customized product collaborative design platform and making full use of all kinds of customized product collaborative tools such as custom products integrated visualization, customized process information communication collaboration, information integration and conflict resolution technology, sharing server, customized system knowledge base, etc. to analyze and map customer needs and perform scientific decomposition and reasonable allocation for customized product parts design tasks, ultimately completing the customized product design [16], which is shown in Figure 1.

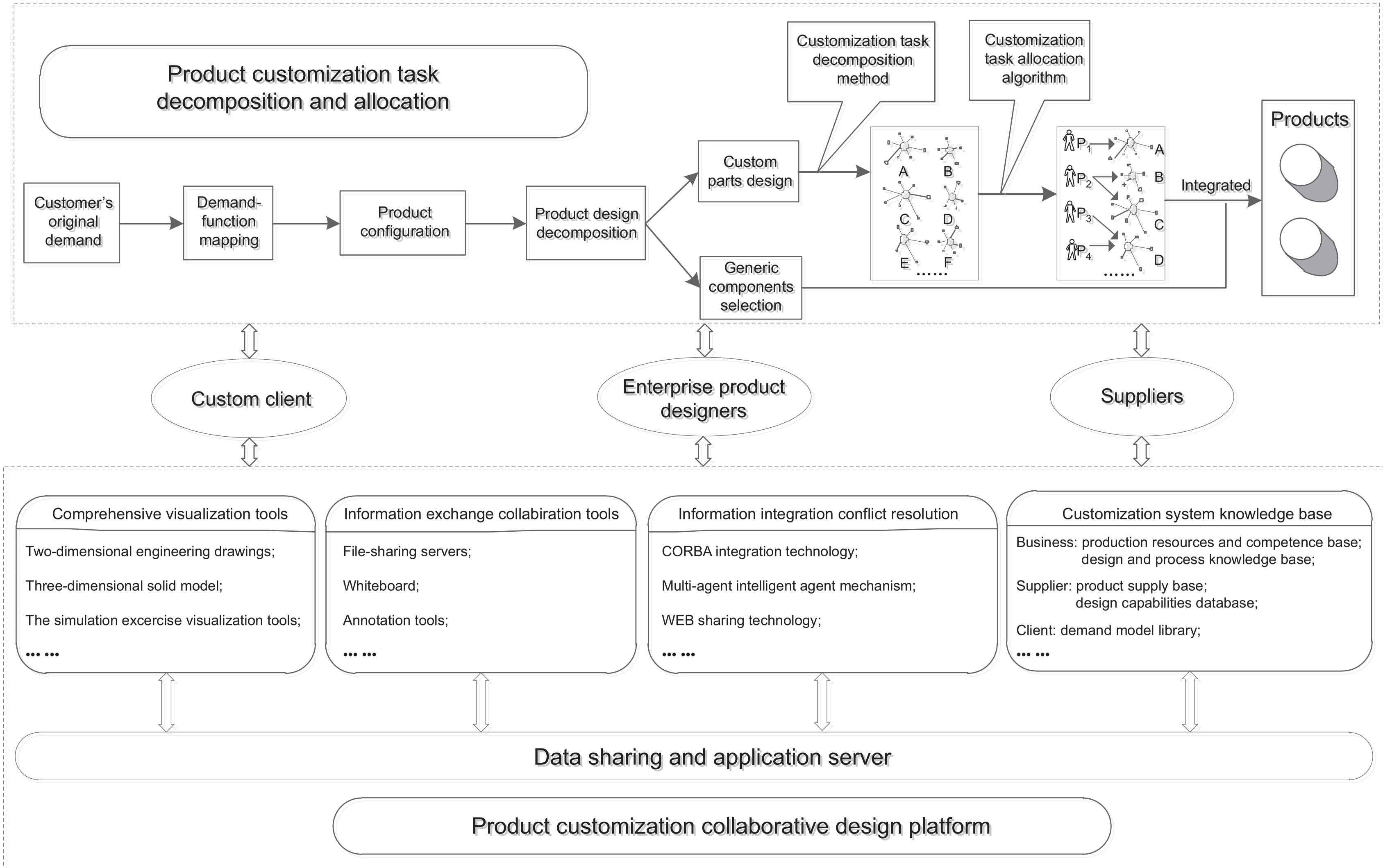

Figure 1. Collaborative Customized Product Development Process.

Figure 1 shows that the relationship among suppliers, customers, and enterprises who participate in the collaborative customized product development chain is no longer the simple cooperative relationship of the material supply type or the supply-and-demand relations type. They are fully involved in the whole product development and designing phase. The collaborative relationship between them changes due to different task decomposition and allocation results.

2.2 Analysis of Task Allocation Strategy

In the process of collaborative customized product development, the purpose of task decomposition is to get a number of executable design subtasks with appropriate granularity, whereas that of task allocation is to allocate these subtasks to the enterprise's development team and the most suitable suppliers to ensure that the task allocation scheme meets technical requirements, time constraint, and other conditions in order to obtain the maximum profit at the same time. In a collaborative customized product development task allocation, apart from taking the traditional influence factors such as task lead time, cost, etc., into consideration, the matching degree between the tasks and the enterprise's development team and that between the tasks and the suppliers, the mutual coordination efficiency between the enterprise's development team and the suppliers, and that between the suppliers will also have a significant impact on task allocation; therefore, influencing factors such as task matching degree and task coordination efficiency should be also considered in task allocation.

The participants of collaborative customized product development task allocation are mainly the enterprise's own development team and suppliers. When the enterprise's grasp for the customer demand is relatively vague or the customer's demand cannot be accurately expressed using general customization collaborative tools, the customers are required to collaboratively participate in the customized product development process. Due to the limitations of the customers' own professional knowledge and technical skills, the customers cannot finish a design task independently, and they collaborate with the enterprise's development team or suppliers to complete a design task. At this time, the main role of the customers' participation is to verify their own customization demands and give feedback to the designers. Compared with the mode where only the suppliers participate in the collaborative development or with the networked customization development mode, which only uses the customers' knowledge, this mode, which directly involves the customers in the customized product development, has the outstanding advantage of obtaining direct and immediate feedback from the customers for a functional unit or a part during the process of customization trial production, not for product samples, when all the design tasks are already finished. This reduces the product-rework design rate to a certain extent, ensures that the manufactured product's standards match the customization demands, and shortens the design trial production time.

During task allocation, the task allocation strategy needs to be considered first, i.e., the basic strategy that the task allocation process should follow. Second, task fitness and coordination efficiency should be considered.

Based on the analysis above, the task allocation strategy is given as follows:

1. Subtasks including function parts with the customers' fuzzy customization demands must be allocated to the suppliers or the enterprise's development team with the collaborative participation of customers.
2. Subtasks including product core technology must be allocated to the enterprise's development team.
3. Subtasks without product core technology can be allocated to suppliers.
4. The task allocation scheme should meet the constraints of time, cost, quality, and so on.
5. The optimal task allocation scheme should maximize the total task fitness and the total task coordination efficiency as far as possible.

Allocating the product customization collaborative development task according to the task allocation strategy above can provide support for the construction of scientific and reasonable collaborative customized product development chain organization structure and the optimal use of resource in product development process.

3 Establishment of Task Allocation Mathematics Model in Collaborative Customized Product Development

3.1 Description of the Problem

Suppose a product customization development project can be decomposed into n design subtasks, namely $T = \{T_1, \ldots, T_n\}$, and it needs to select corresponding suppliers from l sets of suppliers with the enterprise's development team and customers to finish the subtasks collaboratively. The l sets can be represented as $P_1, \ldots, P_i, \ldots, P_l$, and the numbers of the suppliers contained in each set are $k_1, \ldots, k_i, \ldots, k_l$, respectively, and the suppliers in set i can be represented as the set $P_i = \{P_{i1}, \ldots, P_{ik_i}\}$. The n design subtasks are allocated to the most suitable suppliers selected from the $m = \sum_{i=1}^{l} k_i$ suppliers, the enterprise's own development team, and the customers to finish collaboratively. The allocation relationship can be shown by the following mapping diagram in Figure 2.

3.2 Definition and Calculation Formula of Task Fitness

Definition 1: Task fitness. Task fitness is the quantitative level of the fitting degree of design personnel (including suppliers, enterprise's development team, etc.) to design tasks. Factors such as the geographical position of suppliers, the experience and number of times the suppliers' and the enterprise's development team have performed similar development tasks, software and hardware infrastructure conditions, interest in the task, and so on will have an influence on the fitness of the suppliers' and enterprise's development team to design tasks [9], and the detailed description is as follows:

1. Geographical factors: Introduce the geographical position coefficient $d(i, j)$ to describe the geographical position relationship between suppliers and places of task execution. If the supplier is in another place, $d(i, j) = 1$; otherwise, $d(i, j) = 2$.
2. Ability factors: Introduce the ability matrix $B = (b_{ij})_{(m+1)\times n}$ to describe the design ability that the enterprise's development team and m suppliers have for all tasks, where b_{1j} represents the task design ability that the enterprise's development team has for design task j and $b_{ij} (2 \le i \le m+1)$ represents the design ability that supplier $(i-1)$ has for design task j. The value of b_{ij} is represented by a six-dimensional level vector $b_v = [0, 0.2, 0.4, 0.6, 0.8, 1]$, and the division is mainly according to the professional title, performance, experience, and the number of times that similar tasks have been performed, etc.
3. Equipment factors: Introduce the equipped level coefficient $p(i, j)$ of the software and hardware infrastructure to describe the software and hardware level of the suppliers and the enterprise's development team. This is divided into international advanced level, domestic advanced level, and general equipped level, represented as $p(i, j) = 3$, 2, and 1, respectively.
4. Interest factors: Use a number c_{ij} $(0 < c_{ij} < 1)$ to describe the level of interest that each supplier has in task j; the interest matrix $C = (c_{ij})_{(m+1)\times n}$ can then be constructed, where c_{ij} represents the measure of the level of interest that the enterprise's development team has in design task j and $c_{ij} (2 \le i \le m+1)$ represents the measure of the level of interest that supplier $(i-1)$ has in design task j.

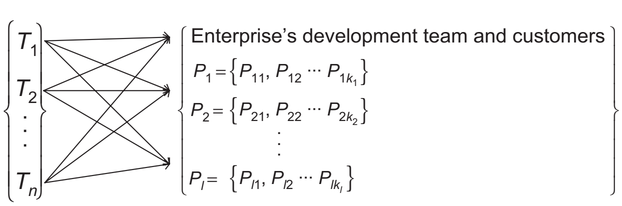

Figure 2. The Mapping Relationship Diagram of Product Customization Development Task Allocation.

Based on the analysis above, the task fitness matrix is as follows:

$$F=\begin{pmatrix} f_{11} & \cdots & f_{1n} \\ \vdots & f_{ij} & \vdots \\ f_{m+1,\,1} & \cdots & f_{m+1,\,n} \end{pmatrix}. \tag{1}$$

Among them, $f_{ij} = \alpha \cdot d(i, j) + \beta \cdot b_{ij} + \delta \cdot q(i, j) + \varepsilon \cdot c_{ij}$; α, β, δ, and ε are the corresponding weighted coefficients; f_{1j} represents the fitness between the enterprise's development team and task j; $f_{ij} (2 \leq i \leq m+1)$ represents the fitness between supplier $(i-1)$ and task j.

3.3 Definition and Calculation Formula of Task Coordination Efficiency

Definition 2: Task coordination efficiency. Task coordination efficiency is the quantitative level of coordination degree between the enterprise's development team and the supplier or between the suppliers [7]. Here, task coordination efficiency is calculated based on the task execution time. For example, the coordination efficiency between suppliers (or between the supplier and the enterprise's development team) who execute subtasks T_i and T_j is $s_{ij}=\frac{t_i+t_j}{t_{i,j}}$. Among them, t_i and t_j represent the separate execution time of tasks T_i and T_j, respectively, and t_{ij} represents the total execution time of the two tasks. According to formula (1), the total execution time is $t_{i,j}=t_i+\frac{t_j+t_i\times d_{i,j}}{1-d_{i,j}\times d_{j,i}}$ and $d_{i,j}$ and $d_{j,i}$ are the interaction coupling coefficients of task T_i and T_j, and the value mainly depends on the convenience of information interaction, experience, and the number of times of cooperation, as is shown in Table 1.

Basing on the above analysis, the task coordination efficiency matrix is

$$S=\begin{array}{c} \\ T_1 \\ \vdots \\ T_i \\ \vdots \\ T_n \end{array}\begin{array}{c} T_1 \cdots T_j \cdots T_n \\ \begin{pmatrix} s_{11} & \cdots & s_{1n} \\ \vdots & s_{ij} & \vdots \\ s_{n,\,1} & \cdots & s_{n,\,n} \end{pmatrix} \end{array}. \tag{2}$$

3.4 Multiobjective Optimization Mathematics Model of Task Allocation

From the task allocation strategy of collaborative customized product development in Section 2.2, it can be seen that collaborative customized product development task allocation aims to allocate the design subtasks to the most suitable suppliers or enterprise's development team, maximize the coordination efficiency between them, and meet the constraints of time, cost, quality, and others. Based on this, the following equations show the mathematical model of collaborative customized product development task allocation:

Table 1. The Value of Interaction Coupling Coefficients and Base of Judgment.

Values of coefficients	Base of judgment
0.1	Interaction between T_i and T_j is far from convenient, no cooperation ever
0.3	Interaction between T_i and T_j is not convenient, less cooperation ever
0.5	Interaction between T_i and T_j is less convenient, few cooperation ever
0.7	Interaction between T_i and T_j is convenient, much cooperation ever
0.9	Interaction between T_i and T_j is very convenient, much more cooperation ever
0.2, 0.4, 0.6, and 0.8	Intermediate value

$$\max Z_1 = \sum_{i=1}^{m+1} \sum_{j=1}^{n} f_{ij} \cdot x_{ij}, \tag{3}$$

$$\max Z_2 = \sum_{i=1}^{m+1} \sum_{j=1}^{n} s_{ij} \cdot x_{ij}, \tag{4}$$

$$\text{s.t.} \quad \sum_{i=1}^{m+1} \sum_{j=1}^{n} t_{ij} \cdot x_{ij} \leq T, \tag{5}$$

$$\sum_{i=1}^{m+1} \sum_{j=1}^{n} t_{ij} \cdot e_{ij} \cdot x_{ij} \leq C, \tag{6}$$

$$\sum_{i=1}^{m+1} \sum_{j=1}^{n} q_{ij} \cdot x_{ij} \geq Q, \tag{7}$$

$$\sum_{i=1}^{m+1} x_{ij} \geq 1 \quad j = 1,\ 2, \ldots, n, \tag{8}$$

$$\sum_{j=1}^{n} x_{ij} \geq 0 \text{ and } \sum_{j=1}^{n} x_{ij} \leq n \quad i = 1,\ 2, \ldots, m+1. \tag{9}$$

Among them, x_{ij} is a 0, 1 selection variable, and its value is as follows:

$$x_{ij} = \begin{cases} 1 & \text{task } j \text{ allocated to supplier } i; \\ 0 & \text{task } j \text{ not allocated to supplier } i;\text{'} \end{cases} \tag{10}$$

where t_{ij} represents the time that supplier i needs to complete task j, e_{ij} represents the cost per unit time that supplier i needs to perform task j, q_{ij} represents the relative quality that supplier i achieves when performing task j. The value of q_{ij} can be obtained through the comprehensive evaluation of product quality offered by suppliers. The evaluation sets V = {unqualified (0), qualified (0–0.2), general (0.2–0.4), medium (0.4–0.6), good (0.6–0.8), excellent (0.8–1.0)}.

The objective function expression (3) represents the total task fitness maximum, and expression (4) represents the total coordination efficiency between task maxima. Constraint function expression (5) represents the execution time of tasks, which should be in the stipulated time T. Expression (6) represents the total execution cost of n tasks, which should be in range of allowing cost C. Expression (7) represents the total relative quality of n tasks, which should not be less than Q. Expression (8) stipulates that each task needs to be allocated to at least one supplier or enterprise itself. Expression (9) stipulates that each supplier and the enterprise itself can get no more than n tasks.

4 Multiobjective Optimization Algorithm of Task Allocation in Collaborative Customized Product Development

Task allocation in collaborative customized product development is a multiobjective combinatorial optimization NP-hard problem. It is difficult to obtain the optimal results quickly by traditional methods. Genetic algorithm is a kind of search optimization algorithm simulating the process of biological evolution. It has a strong global searching capability, but with a poor partial searching capability, and it is easy to fall into local optimum and lead to the premature phenomenon [6]. In view of the limitation of the nonlinear multiobjective combinatorial optimization characteristic of collaborative customized product development mathematical model, AGA is chosen to solve the task allocation model of collaborative customized product development.

The specific descriptions of the AGA operation process follow.

4.1 Coding and Decoding

Here, the binary coding method is adopted. The length of each chromosome is $m + n$. Each chromosome is divided into n gene segments, and each gene segment represents a task. All suppliers that can finish customization task i are ranged in a sequence according to serial number sequence of supplier sets. Assume that there are m_i suppliers, the number of optional allocated objects for customization task i is $m_i + 1$, and the first represents the enterprise itself. The sum of all suppliers that can complete n customized tasks is m, namely $\sum_{i=1}^{n} m_i = m.$ In chromosome coding, a gene value of 0 means that the corresponding task of the gene segment is not allocated to the supplier or the enterprise itself. A gene value of 1 means the task that is assigned to this supplier or enterprise itself. In addition, the chromosome can be decoded according to the chromosome coding method.

A feasible chromosome coding schemes is shown in Figure 3. In this coding scheme, task 1 represents allocation to the first supplier of supplier set 1. Its gene value is 1, and the values of the rest of the genes are 0. For the same reason, task n is allocated to the second supplier of supplier set m.

4.2 The Establishment of Fitness Function

Collaborative customized product development task allocation is a nonlinear multiobjective optimization problem, and there are some implicative relationships between multiple targets; thus, it is difficult to give specific and accurate values for each target at the same time. However, the positive ideal point (the value most expect to achieve) and negative ideal point (the value least expect to achieve) of each target can be given. Thus, it is appropriate to choose the ideal point method to construct fitness function in the AGA.

The ideal point method evaluates the quality of the scheme according to the distance between the objective function value and the ideal point. The smaller the distance, the more optimal the scheme [6, 12]. An ideal point consists of the ideal value of each objective function, which is decided by decision makers or made up of a single objective optimal value. Accordingly, the evaluation function of collaborative customized product development task allocation problem is

$$\min Z = \sqrt{(Z_1 - Z_1^*)^2 + (Z_2 - Z_2^*)^2}, \tag{11}$$

where (Z_1^*, Z_2^*) is the ideal point, which consists of the optimal value of two objective functions, (Z_1, Z_2) is the objective function value of collaborative customized product development task allocation schemes, and Z is the distance between the allocation scheme and the ideal point.

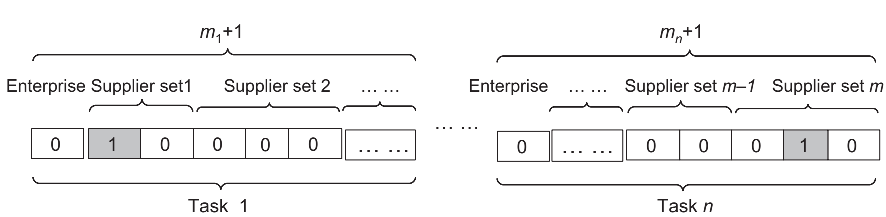

Figure 3. A feasible Chromosome Coding Scheme.

Because the magnitudes of the objective function Z_1 and Z_2 are different, and they are not equally important, it is necessary to deal with dimensionless correction and give the weight coefficient to distinguish. The evaluation function can be modified as

$$\min Z=\sqrt{w_1\left(\frac{Z_1-Z_1^*}{Z_1^*}\right)^2+w_2\left(\frac{Z_2-Z_2^*}{Z_2^*}\right)^2}, \tag{12}$$

where w_1 and w_2 are the weights of the objective functions whose value can be determined by the expert evaluation method and other and $w_1 + w_2 = 1$.

According to the analysis above, the fitness function of AGA is constructed as follows:

$$f(i)=M-\sqrt{w_1\left(\frac{Z_1^i-Z_1^*}{Z_1^*}\right)^2+w_2\left(\frac{Z_2^i-Z_2^*}{Z_2^*}\right)^2}. \tag{13}$$

The function above represents the fitness function value of the i^{th} chromosome, where Z_1^i and Z_2^i are two corresponding objective function value of the chromosome and M is a sufficiently large positive number.

4.3 Selecting Operation

Use the roulette wheel selection method. Each generation determines the probability of copying to the next generation according to the size of individual fitness value. Assume that the population size is *popsize*, the fitness value of individual i is $f(i)$, then the selection probability p_i is

$$p_i=\frac{f(i)}{\sum_{i=1}^{popsize} f(i)}. \tag{14}$$

4.4 Adaptive Crossover and Mutation Operation

Here, a two-point crossover is used to perform the crossover operation, and a double-point mutation method was used to perform the mutation operation. The probability of crossover and mutation is determined by the adaptive selection strategy. In a standard genetic algorithm (SGA), the probability of crossover and mutation is a fixed value, which leads to the premature phenomenon and local convergence. To avoid the defects, AGA uses the adaptive selection strategy for the probability of crossover and mutation: when the bigger fitness value of the two chromosomes performing crossover operation is less than or equal to the average fitness value, the probability of adaptive crossover increases; otherwise, it decreases. When the fitness value of the chromosomes performing mutation operation is less than or equal to the average fitness value, the probability of adaptive mutation increases; otherwise, it decreases [6, 12].

Assume $f_{\max}$, $f_{\min}$, $\bar{f}$ are the largest, minimum, and average fitness values of group, respectively. f represents the larger one of the fitness values of the two crossover individuals; f' represents the fitness value of the mutation individual; $P_{c\max}$ and $P_{c\min}$ represent the maximum and minimum crossover probability of the group, respectively; $P_{m\max}$ and $P_{m\min}$ represent the maximum and minimum mutation probability of the group, respectively. The crossover probability value of AGA is as follows:

$$P_c=\begin{cases} P_{c\min}-\dfrac{f_{\max}-f}{f_{\max}-f_{\min}}(P_{c\max}-P_{c\min}), & f>\bar{f} \\ P_{c\min}+\dfrac{f_{\max}-f}{f_{\max}-f_{\min}}(P_{c\max}-P_{c\min}), & f\le\bar{f}. \end{cases} \tag{15}$$

The mutation probability value of AGA is as follows:

$$P_m=\begin{cases}P_{m\min}-\dfrac{f_{\max}-f'}{f_{\max}-f_{\min}}(P_{m\max}-P_{m\min}), & f'>\overline{f}\\ P_{m\min}+\dfrac{f_{\max}-f'}{f_{\max}-f_{\min}}(P_{m\max}-P_{m\min}), & f'\le\overline{f}.\end{cases} \tag{16}$$

In the same generation, different individuals are endowed with different crossover and mutation probabilities. The protection for individuals with higher fitness value should be undertaken, and the probabilities of crossover and mutation are correspondingly reduced, but the probabilities of crossover and mutation of individuals with lower fitness value should increase. Each individual in each generation group has a different crossover and mutation probability so as to realize adaptive crossover and mutation.

5 Case Study

Take a customized development design project of 5-MW variable-speed, constant-frequency wind turbine generator in a wind power generation company as an example to apply the product customization collaborative development task allocation optimization algorithm proposed above.

The collaborative development project of 5-MW wind turbine can be divided into 10 subtasks, namely, A, spindle design; B, yaw system design; C, gear box design; D, transmission chain system design; E, electrical system design; F, generator set system design; G, control cabinet system design; H, cabin design; I, pitch system design; G, wheel design. According to the task allocation algorithm proposed in this article, the 10 subtasks are allocated.

Tasks A, B, C, D, E, F, G, H, I, and J can be allocated to suppliers set P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10}, respectively. The numbers of suppliers contained in each set are 4, 2, 5, 3, 3, 3, 2, 3, 2, and 2, respectively. Among these, task E can also be designed by the enterprise itself. Let the enterprise itself be the first one in set P_5. The first supplier in P_1 can simultaneously participate in tasks A and C; thus, the supplier is also included in set P_3 as its first supplier. The first supplier in P_4 can simultaneously participate in tasks D and E; thus, the supplier is also included in set P_5 as its second supplier. The first supplier in P_6 can simultaneously participate in tasks F and G; thus, the supplier is also included in set P_7 as its first supplier. Ten tasks (A–J) are allocated to 29 suppliers and the enterprise itself.

Here, all the suppliers and enterprise itself are arranged in a sequence according to the serial number of design tasks. According to the definition of task fitness, the geographical factor vector formed by $d(i, j)$ can be obtained as follows:

d = [2, 1, 2, 1, 1, 2, 2, 1, 1, 1, 1, 2, 1, 2, 2, 1, 2, 2, 2, 2, 1, 1, 2, 1, 2, 1, 2, 1, 1, 1].

The ability factor vector formed by b_{ij} is

b = [0.2, 0.4, 0.2, 0.4, 0.6, 0.8, 0.4, 0.6, 0.8, 0.6, 0.2, 0.6, 0.4, 0.6, 0.8, 0.4, 0.6, 0.4, 0.4, 0.6, 0.8, 0.4, 0.6, 0.2, 0.6, 0.8, 0.8, 0.6, 0.4, 0.6].

The equipment factor vector formed by $p(i, j)$ is

p = [2, 3, 2, 1, 2, 1, 1, 2, 2, 3, 1, 1, 2, 2, 3, 1, 2, 2, 2, 1, 3, 2, 3, 1, 2, 1, 3, 1, 1, 2].

The interest factor vector formed by c_{ij} is

c = [0.6, 0.7, 0.6, 0.8, 0.6, 0.5, 0.8, 0.6, 0.5, 0.3, 0.7, 0.6, 0.6, 0.8, 0.8, 0.6, 0.7, 0.6, 0.5, 0.3, 0.6, 0.4, 0.5, 0.6, 0.8, 0.7, 0.2, 0.3, 0.5, 0.6].

Setting $\alpha = 0.2, \beta = 0.4. \delta = 0.2, \varepsilon = 0.2$, the fitness vector can be obtained as follows:

$F = [1, 1.1, 1, 0.72, 0.96, 1.02, 0.92, 0.96, 1.02, 1.1, 0.62, 0.96, 0.88, 1.2, 1.48, 0.68, 1.18, 1.08, 1.06, 0.9, 1.24, 0.84, 1.34, 0.6, 1.2, 0.86, 1.36, 0.7, 0.66, 0.96]$.

The task execution time of 29 suppliers and the enterprise itself can be expressed with vector as follows:

$t = [215, 210, 200, 220, 160, 140, 360, 345, 350, 355, 360, 200, 180, 210, 262, 270, 268, 156, 220, 216, 210, 180, 196, 160, 146, 180, 160, 172, 130, 120]$.

The cost per unit time can be expressed with vector as follows:

$e = [2.1, 3.3, 2.6, 1.4, 2.2, 1.9, 3, 2.8, 2.9, 3.6, 3.2, 2.4, 3.1, 2.6, 3, 3.2, 3.2, 2.8, 1.8, 1.6, 2.2, 2.2, 2.4, 2.3, 2.3, 1.9, 3.2, 2.6, 2.3, 2.4]$.

The relative quality can be expressed with vector as follows:

$q = [0.5, 0.6, 0.7, 0.8, 0.5, 0.6, 0.3, 0.4, 0.6, 0.8, 0.7, 0.65, 0.72, 0.56, 0.8, 0.7, 0.64, 0.53, 0.6, 0.42, 0.46, 0.6, 0.8, 0.6, 0.4, 0.2, 0.6, 0.4, 0.5, 0.8]$.

According to the definition of task coordination efficiency, the task coordination efficiency matrix can be obtained:

$$S = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \\ I \\ J \end{array} \begin{array}{c} \begin{array}{cccccccccc} A & B & C & D & E & F & G & H & I & J \end{array} \\ \begin{bmatrix} S_{11} & S_{12} & S_{13} & S_{14} & S_{15} & S_{16} & S_{17} & S_{18} & S_{19} & S_{1,10} \\ S_{21} & S_{22} & S_{23} & S_{24} & S_{25} & S_{26} & S_{27} & S_{28} & S_{29} & S_{2,10} \\ S_{31} & S_{32} & S_{33} & S_{34} & S_{35} & S_{36} & S_{37} & S_{38} & S_{39} & S_{3,10} \\ S_{41} & S_{42} & S_{43} & S_{44} & S_{45} & S_{46} & S_{47} & S_{48} & S_{49} & S_{4,10} \\ S_{51} & S_{52} & S_{53} & S_{54} & S_{55} & S_{56} & S_{57} & S_{58} & S_{59} & S_{5,10} \\ S_{61} & S_{62} & S_{63} & S_{64} & S_{65} & S_{66} & S_{67} & S_{68} & S_{69} & S_{6,10} \\ S_{71} & S_{72} & S_{73} & S_{74} & S_{75} & S_{76} & S_{77} & S_{78} & S_{79} & S_{7,10} \\ S_{81} & S_{82} & S_{83} & S_{84} & S_{85} & S_{86} & S_{87} & S_{88} & S_{89} & S_{8,10} \\ S_{91} & S_{92} & S_{93} & S_{94} & S_{95} & S_{96} & S_{97} & S_{98} & S_{99} & S_{9,10} \\ S_{10,1} & S_{10,2} & S_{10,3} & S_{10,4} & S_{10,5} & S_{10,6} & S_{10,7} & S_{10,8} & S_{10,9} & S_{10,10} \end{bmatrix} \end{array}$$

The matrix is a symmetric matrix, and the diagonal elements are zero. For example, s_{12} represents the coordination efficiency between four candidate suppliers of task A and 2 candidate suppliers of task B, and it is a 4×2 matrix as follows:

$$s_{12} = \begin{bmatrix} 0.67 & 0.78 \\ 0.62 & 0.56 \\ 0.64 & 0.62 \\ 0.78 & 0.72 \end{bmatrix}$$

The rest of the elements are similar. Although there are $C_{10}^{2} = 45$ matrix elements, because of space limitations, only s_{12} is listed above. The rest of the data is listed in the task coordination efficiency matrix as follows: The delivery time constraint is 2100, the cost constraint is 5500, and the quality constraint is 6.5, namely,

$$\Sigma t \le 2100; \Sigma c \le 5500; \Sigma q \ge 6.5.$$

$$
S=\begin{bmatrix}
0 & 0 & 0 & 0 & 0.67 & 0.78 & 1 & 0.82 & 0.63 & 0.71 & 0.62 & 0.67 & 0.68 & 0.72 & 0.64 & 0.36 & 0.42 & 0.51 & 0.75 & 0.69 & 0.7 & 0.57 & 0.69 & 0.75 & 0.88 & 0.36 & 0.67 & 0.68 & 0.82 & 0.57 \\
0 & 0 & 0 & 0 & 0.62 & 0.56 & 0.78 & 0.72 & 0.67 & 0.63 & 0.81 & 0.76 & 0.72 & 0.81 & 0.49 & 0.58 & 0.66 & 0.81 & 0.54 & 0.65 & 0.64 & 0.58 & 0.67 & 0.72 & 0.62 & 0.78 & 0.75 & 0.76 & 0.56 & 0.57 \\
0 & 0 & 0 & 0 & 0.64 & 0.62 & 0.75 & 0.46 & 0.82 & 0.61 & 0.82 & 0.68 & 0.72 & 0.66 & 0.51 & 0.65 & 0.67 & 0.51 & 0.42 & 0.63 & 0.58 & 0.56 & 0.81 & 0.26 & 0.37 & 0.42 & 0.82 & 0.83 & 0.82 & 0.72 \\
0 & 0 & 0 & 0 & 0.78 & 0.72 & 0.62 & 0.63 & 0.54 & 0.52 & 0.67 & 0.76 & 0.86 & 0.68 & 0.82 & 0.61 & 0.42 & 0.27 & 0.91 & 0.67 & 0.67 & 0.67 & 0.82 & 0.36 & 0.57 & 0.34 & 0.64 & 0.65 & 0.67 & 0.68 \\
0.67 & 0.62 & 0.64 & 0.78 & 0 & 0 & 0.65 & 0.68 & 0.67 & 0.56 & 0.81 & 0.68 & 0.69 & 0.57 & 0.86 & 0.88 & 0.86 & 0.85 & 0.86 & 0.52 & 0.63 & 0.54 & 0.38 & 0.76 & 0.61 & 0.62 & 0.85 & 0.67 & 0.76 & 0.79 \\
0.78 & 0.56 & 0.62 & 0.72 & 0 & 0 & 0.71 & 0.58 & 0.76 & 0.85 & 0.72 & 0.56 & 0.81 & 0.67 & 0.42 & 0.71 & 0.46 & 0.54 & 0.67 & 0.81 & 0.65 & 0.67 & 0.68 & 0.62 & 0.46 & 0.24 & 0.73 & 0.76 & 0.81 & 0.73 \\
1 & 0.78 & 0.75 & 0.62 & 0.65 & 0.71 & 0 & 0 & 0 & 0 & 0 & 0.81 & 0.71 & 0.72 & 0.56 & 0.81 & 0.32 & 0.53 & 0.51 & 0.46 & 0.54 & 0.67 & 0.68 & 0.81 & 0.55 & 0.71 & 0.71 & 0.78 & 0.44 & 0.72 \\
0.82 & 0.72 & 0.46 & 0.63 & 0.68 & 0.58 & 0 & 0 & 0 & 0 & 0 & 0.68 & 0.42 & 0.56 & 0.85 & 0.74 & 0.55 & 0.76 & 0.58 & 0.54 & 0.76 & 0.78 & 0.82 & 0.81 & 0.56 & 0.65 & 0.76 & 0.66 & 0.55 & 0.61 \\
0.63 & 0.67 & 0.82 & 0.54 & 0.67 & 0.76 & 0 & 0 & 0 & 0 & 0 & 0.88 & 0.71 & 0.81 & 0.76 & 0.75 & 0.68 & 0.42 & 0.75 & 0.78 & 0.79 & 0.66 & 0.63 & 0.52 & 0.54 & 0.61 & 0.73 & 0.65 & 0.58 & 0.63 \\
0.71 & 0.63 & 0.61 & 0.52 & 0.56 & 0.85 & 0 & 0 & 0 & 0 & 0 & 0.86 & 0.46 & 0.32 & 0.77 & 0.63 & 0.82 & 0.65 & 0.72 & 0.76 & 0.78 & 0.71 & 0.73 & 0.61 & 0.72 & 0.42 & 0.81 & 0.72 & 0.67 & 0.67 \\
0.62 & 0.81 & 0.82 & 0.67 & 0.81 & 0.72 & 0 & 0 & 0 & 0 & 0 & 0.85 & 0.54 & 0.53 & 0.76 & 0.42 & 0.65 & 0.56 & 0.82 & 0.83 & 0.66 & 0.66 & 0.68 & 0.71 & 0.62 & 0.62 & 0.66 & 0.58 & 0.63 & 0.68 \\
0.67 & 0.76 & 0.68 & 0.76 & 0.68 & 0.56 & 0.81 & 0.68 & 0.88 & 0.86 & 0.85 & 0 & 0 & 0 & 0.81 & 1 & 0.61 & 0.85 & 0.63 & 0.68 & 0.75 & 0.72 & 0.76 & 0.76 & 0.63 & 0.61 & 0.71 & 0.76 & 0.64 & 0.65 \\
0.68 & 0.72 & 0.72 & 0.86 & 0.69 & 0.81 & 0.71 & 0.42 & 0.71 & 0.46 & 0.54 & 0 & 0 & 0 & 0.56 & 0.54 & 0.72 & 0.61 & 0.76 & 0.82 & 0.61 & 0.81 & 0.65 & 0.72 & 0.71 & 0.86 & 0.86 & 0.82 & 0.73 & 0.64 \\
0.72 & 0.81 & 0.66 & 0.68 & 0.57 & 0.67 & 0.72 & 0.56 & 0.81 & 0.32 & 0.53 & 0 & 0 & 0 & 0.65 & 0.61 & 0.42 & 0.62 & 0.66 & 0.68 & 0.69 & 0.64 & 0.82 & 0.61 & 0.86 & 0.64 & 0.64 & 0.63 & 0.82 & 0.61 \\
0.64 & 0.49 & 0.51 & 0.82 & 0.86 & 0.42 & 0.56 & 0.85 & 0.76 & 0.77 & 0.76 & 0.81 & 0.56 & 0.65 & 0 & 0 & 0 & 0 & 0.85 & 0.64 & 0.63 & 0.67 & 0.63 & 0.67 & 0.76 & 0.45 & 0.66 & 0.68 & 0.67 & 0.69 \\
0.36 & 0.58 & 0.65 & 0.61 & 0.88 & 0.71 & 0.81 & 0.74 & 0.75 & 0.63 & 0.42 & 1 & 0.54 & 0.61 & 0 & 0 & 0 & 0 & 0.54 & 0.56 & 0.75 & 0.71 & 0.82 & 0.46 & 0.37 & 0.69 & 0.85 & 0.32 & 0.86 & 0.85 \\
0.42 & 0.66 & 0.67 & 0.42 & 0.86 & 0.46 & 0.32 & 0.55 & 0.68 & 0.82 & 0.65 & 0.61 & 0.72 & 0.42 & 0 & 0 & 0 & 0 & 0.74 & 0.61 & 0.68 & 0.68 & 0.61 & 0.38 & 0.64 & 0.46 & 0.45 & 0.54 & 0.64 & 0.62 \\
0.51 & 0.81 & 0.51 & 0.27 & 0.85 & 0.54 & 0.53 & 0.76 & 0.42 & 0.65 & 0.56 & 0.85 & 0.61 & 0.62 & 0 & 0 & 0 & 0 & 0.81 & 0.64 & 0.42 & 0.73 & 0.76 & 0.85 & 0.76 & 0.39 & 0.58 & 0.64 & 0.76 & 0.62 \\
0.75 & 0.54 & 0.42 & 0.91 & 0.86 & 0.67 & 0.51 & 0.58 & 0.75 & 0.72 & 0.82 & 0.63 & 0.76 & 0.66 & 0.85 & 0.54 & 0.74 & 0.81 & 0 & 0 & 0 & 1 & 0.61 & 0.39 & 0.37 & 0.48 & 0.64 & 0.69 & 0.64 & 0.63 \\
0.69 & 0.65 & 0.63 & 0.67 & 0.52 & 0.81 & 0.46 & 0.54 & 0.78 & 0.76 & 0.83 & 0.68 & 0.82 & 0.68 & 0.64 & 0.56 & 0.61 & 0.64 & 0 & 0 & 0 & 0.58 & 0.56 & 0.46 & 0.85 & 0.71 & 0.73 & 0.71 & 0.54 & 0.53 \\
0.7 & 0.64 & 0.58 & 0.67 & 0.63 & 0.65 & 0.54 & 0.76 & 0.79 & 0.78 & 0.66 & 0.75 & 0.61 & 0.69 & 0.63 & 0.75 & 0.68 & 0.42 & 0 & 0 & 0 & 0.46 & 0.47 & 0.55 & 0.57 & 0.49 & 0.64 & 0.41 & 0.64 & 0.62 \\
0.57 & 0.58 & 0.56 & 0.67 & 0.54 & 0.67 & 0.67 & 0.78 & 0.66 & 0.71 & 0.66 & 0.72 & 0.81 & 0.64 & 0.67 & 0.71 & 0.68 & 0.73 & 1 & 0.58 & 0.46 & 0 & 0 & 0.56 & 0.68 & 0.54 & 0.56 & 0.58 & 0.67 & 0.68 \\
0.69 & 0.67 & 0.81 & 0.82 & 0.38 & 0.68 & 0.68 & 0.82 & 0.63 & 0.73 & 0.68 & 0.76 & 0.65 & 0.82 & 0.63 & 0.82 & 0.61 & 0.76 & 0.61 & 0.56 & 0.47 & 0 & 0 & 0.74 & 0.76 & 0.61 & 0.55 & 0.56 & 0.38 & 0.46 \\
0.75 & 0.72 & 0.26 & 0.36 & 0.76 & 0.62 & 0.81 & 0.81 & 0.52 & 0.61 & 0.71 & 0.76 & 0.72 & 0.61 & 0.67 & 0.46 & 0.38 & 0.85 & 0.39 & 0.46 & 0.55 & 0.56 & 0.74 & 0 & 0 & 0 & 0.85 & 0.86 & 0.61 & 0.63 \\
0.88 & 0.62 & 0.37 & 0.57 & 0.61 & 0.46 & 0.55 & 0.56 & 0.54 & 0.72 & 0.62 & 0.63 & 0.71 & 0.86 & 0.76 & 0.37 & 0.64 & 0.76 & 0.37 & 0.85 & 0.57 & 0.68 & 0.76 & 0 & 0 & 0 & 0.71 & 0.64 & 0.63 & 0.52 \\
0.36 & 0.78 & 0.42 & 0.34 & 0.62 & 0.24 & 0.71 & 0.65 & 0.61 & 0.42 & 0.62 & 0.61 & 0.86 & 0.64 & 0.45 & 0.69 & 0.46 & 0.39 & 0.48 & 0.71 & 0.49 & 0.54 & 0.61 & 0 & 0 & 0 & 0.61 & 0.38 & 0.81 & 0.67 \\
0.67 & 0.75 & 0.82 & 0.64 & 0.85 & 0.73 & 0.71 & 0.76 & 0.73 & 0.81 & 0.66 & 0.71 & 0.86 & 0.64 & 0.66 & 0.85 & 0.45 & 0.58 & 0.64 & 0.73 & 0.64 & 0.56 & 0.55 & 0.85 & 0.71 & 0.61 & 0 & 0 & 0.61 & 0.39 \\
0.68 & 0.76 & 0.83 & 0.65 & 0.67 & 0.76 & 0.78 & 0.66 & 0.65 & 0.72 & 0.58 & 0.76 & 0.82 & 0.63 & 0.68 & 0.32 & 0.54 & 0.64 & 0.69 & 0.71 & 0.41 & 0.58 & 0.56 & 0.86 & 0.64 & 0.38 & 0 & 0 & 0.57 & 0.65 \\
0.82 & 0.56 & 0.82 & 0.67 & 0.76 & 0.81 & 0.44 & 0.55 & 0.58 & 0.67 & 0.63 & 0.64 & 0.73 & 0.82 & 0.67 & 0.86 & 0.64 & 0.76 & 0.64 & 0.54 & 0.64 & 0.67 & 0.38 & 0.61 & 0.63 & 0.81 & 0.61 & 0.57 & 0 & 0 \\
0.57 & 0.57 & 0.72 & 0.68 & 0.79 & 0.73 & 0.72 & 0.61 & 0.63 & 0.67 & 0.68 & 0.65 & 0.64 & 0.61 & 0.69 & 0.85 & 0.62 & 0.62 & 0.63 & 0.53 & 0.62 & 0.68 & 0.46 & 0.63 & 0.52 & 0.67 & 0.39 & 0.65 & 0 & 0
\end{bmatrix}
$$

According to the literatures [5], the population size directly affects the convergence of the algorithm and computational efficiency. If the population size is too small, it is easy to converge to a local optimal solution. If the population size is too large, the calculation speed will be reduced. The general population size is often set to 10–200, according to the actual situation. The crossover probability controls the use frequency of the crossover operation. A greater crossover probability can enable future generations to fully cross, whereas a small crossover probability may cause the evolutionary speed to relatively slow down. The generally recommended value of the crossover probability is set to 0.4–0.99. The mutation probability controls the mutation operation. If the mutation probability is too small, it is easy to lead to premature convergence. If the mutation probability becomes larger, it will enable the solution that jumped out from the local extreme point to obtain the global optimal solution. The mutation probability is set to 0.0001–0.1. Here, set the initial population size as 50, the maximum number of iteration as 500, $M = 100$, $P_{c\,\max} = 0.45$, $P_{c\,\min} = 0.25$, $P_{m\,\max} = 0.004$, and $P_{m\,\min} = 0.002$; weights of the two objective functions are $w_1 = 0.5$ and $w_2 = 0.5$, respectively, and the ideal point consists of two single-goal optimal function values (12.00, 34.19). Use Matlab R2010a for programming calculation. Obtain the chromosome coding of the optimum scheme 1000 10 01000 001 1000 100 01 010 10 10, with the best fitness value of 99.9145, distance between the corresponding optimal objective function value point and ideal point of 0.0755, and the following chromosome coding: task A is assigned to the first supplier in supplier set P_1, task B is assigned to the first supplier in supplier set P_2, task C is assigned to the second supplier in supplier set P_3, task D is assigned to the third supplier in supplier set P_4, task E is assigned to the enterprise itself, task F is assigned to the first supplier in supplier set P_6, task G is assigned to the second supplier in supplier set P_7, task H is assigned to the second supplier in supplier set P_8, task I is assigned to the first supplier in supplier set P_9, and task J is assigned to the first supplier in supplier set P_{10}.

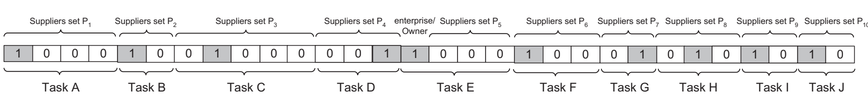

Figure 4. The Final Optimal Allocation Results of Design Tasks of 5-MW Wind Turbine Products.

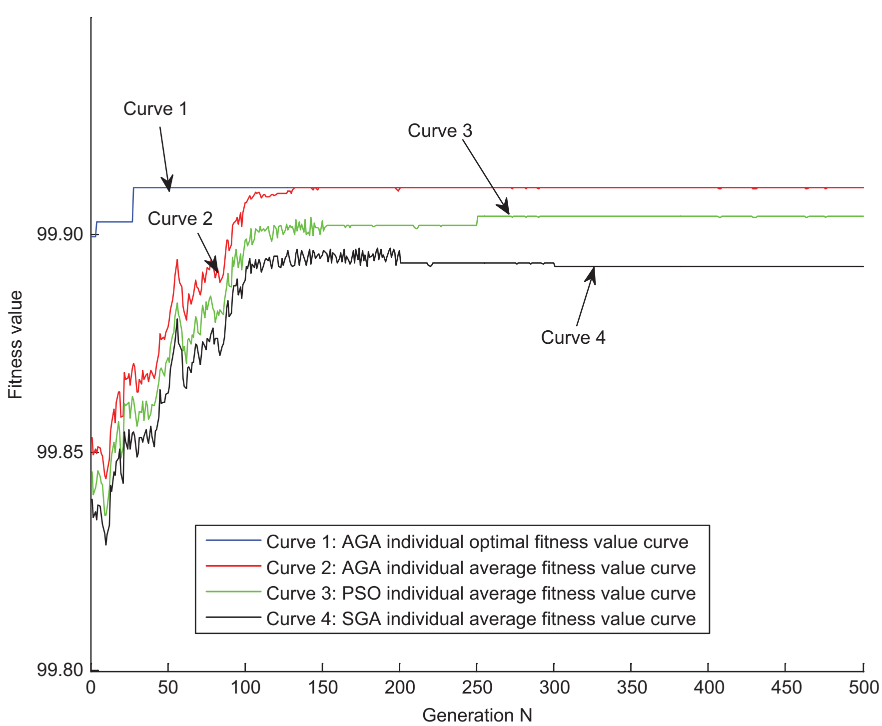

Figure 5. The Operation Results of AGA, PSO, and SGA.

Table 2. Comparison of Different Algorithms' Optimized Results.

Algorithm	Optimal solution	Run time (s)	Iteration number
AGA	99.9145	17.61	131
PSO	99.9037	21.33	153
SGA	99.8607	35.43	302

Table 3. Comparison of Running Results of Different Population Sizes.

Population size	Optimal solution	Run time (s)	Iteration number
50	99.9104	19.63	169
100	99.9141	12.36	88
200	99.9145	10.88	60

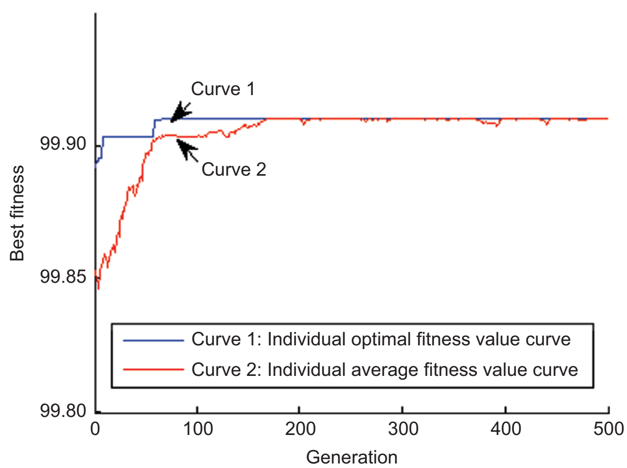

Figure 6. The Convergence Curve when the Population Size is Set to 50.

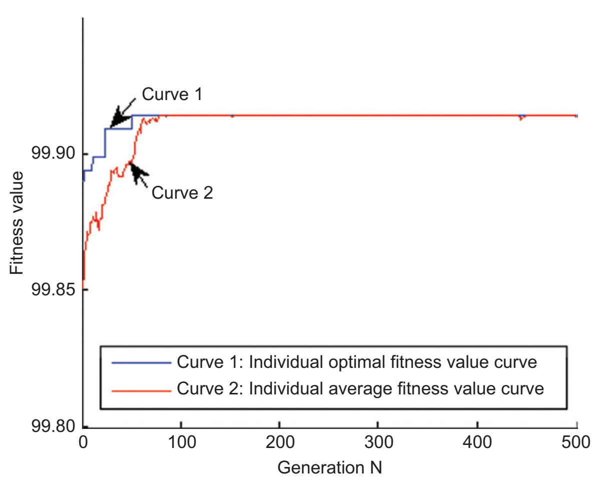

Figure 7. The Convergence Curve when the Population Size is Set to 100.

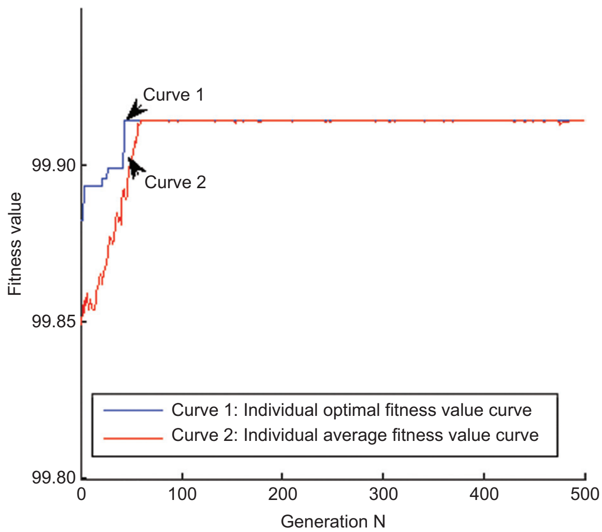

Figure 8. The Convergence Curve when the Population Size is Set to 200.

Table 4. Comparison of Running Results of Different Crossover Probability.

Crossover probability	Optimal solution	Run time (s)	Iteration number
(0.4, 0.2)	99.9104	20.36	183
(0.6, 0.4)	99.9141	16.59	168
(0.8, 0.6)	99.9145	11.67	82

Because of the relative fuzzy grasp on the customization demands of the electrical system of the owners, task E should be completed with the owner's collaboration. To sum up, the final optimal allocation results of all design tasks of 5-MW wind turbine product are shown in Figure 4.

Under the circumstances that the population size and the maximum number of iterations are equal, AGA, PSO, and SGA are used for solving the problem, respectively. The parameters of PSO were set as follows according to the literatures [14]: set the initial inertia weight as 0.9 and the acceleration coefficient as $c_1 = c_2 = 2$. The results are shown in Figure 5. Run independently 50 times, and the best statistical results are shown in Table 2.

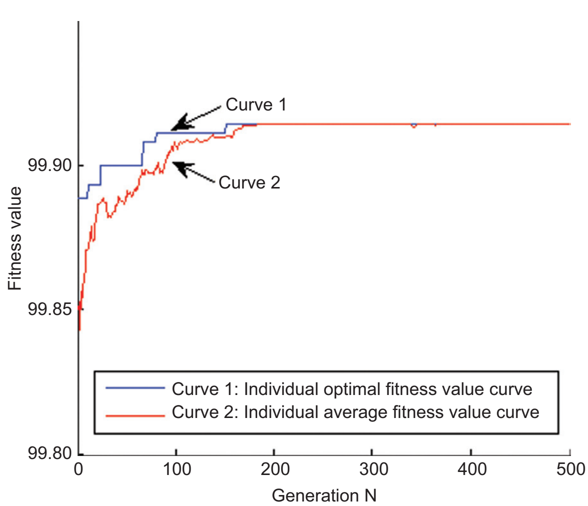

Figure 9. The Convergence Curve when the Crossover Probability is Set to (0.4, 0.2).

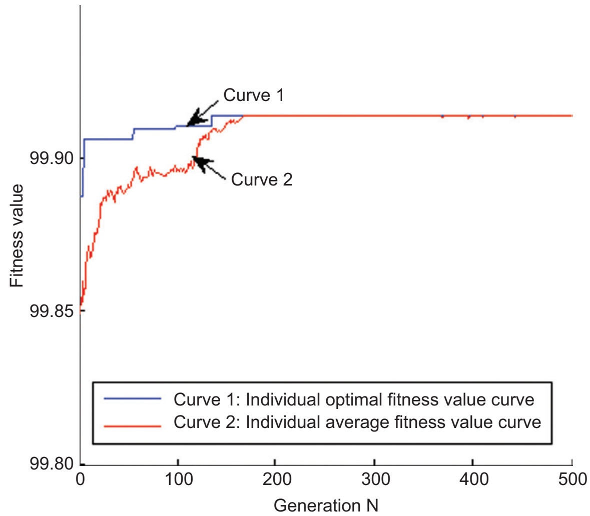

Figure 10. The Convergence Curve when the Crossover Probability is Set to (0.6, 0.4).

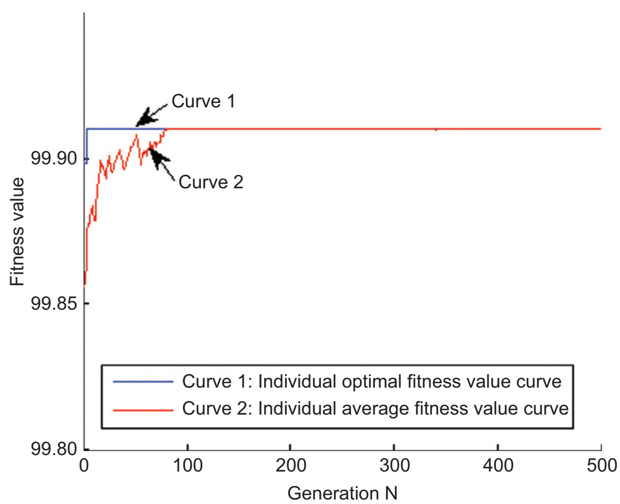

Figure 11. The Convergence Curve when the Crossover Probability is Set to (0.8, 0.6). The Results are Shown in Table 4.

Figure 5 and Table 2 show that AGA converges to the optimal solution after 131 generations, PSO converges to the optimal solution after 153 generations, and SGA converges to the optimal solution after 302 generations. When run on the computer with a dual-core I5-430M CPU, basic frequency of 2.27 GHZ, and 1 GB of memory, the time obtained is 17.61 s for AGA, 21.33 s for PSO, and 35.43 s for SGA.

Thus, it can be seen that AGA converges faster than PSO and SGA, has a greater stability, and has a shorter the running time. Moreover, the optimal value found by AGA is better than those by PSO and SGA. When solving task assignment problems, AGA is superior to PSO and SGA.

Table 5. Comparison of Running Results of Different Mutation Probability.

Mutation probability	Optimal solution	Run time (s)	Iteration number
(0.004, 0.002)	99.9104	18.67	167
(0.006, 0.004)	99.9141	15.66	102
(0.008, 0.006)	99.9145	9.78	49

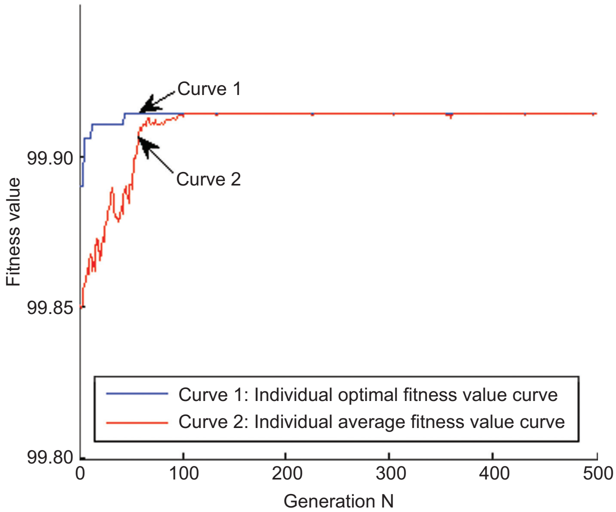

Figure 12. The Convergence Curve when the Mutation Probability is Set to (0.004, 0.002).

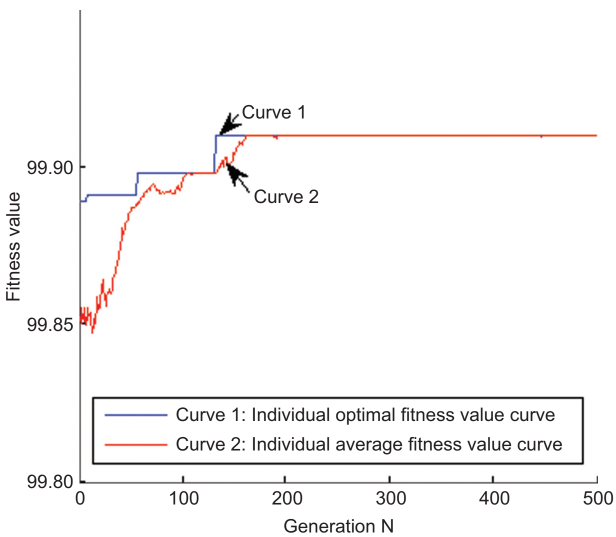

Figure 13. The Convergence Curve when the Mutation Probability is Set to (0.006, 0.004).

When the crossover probability, mutation probability, and other parameters remain the same, the population size was set to 50, 100, and 200. The results are shown in Table 3.

Figures 6–8 and Table 3 show that the larger the population size, the quicker the AGA algorithm converges and the shorter the time needed to find the optimal solution. The number of iterations needed to achieve the optimal solution also decreases.

When the population size, mutation probability, and other parameters remain the same, the maximum and minimum crossover probability was set to (0.4, 0.2), (0.6, 0.4), and (0.8, 0.6). The results are shown in Table 4.

Figures 9–11 and Table 4 show that with the crossover probability increases, the convergence rate increases and the global optimization capability increases. In addition, it can obtain the optimal solution in a shorter time.

When the population size, crossover probability, and other parameters remain the same, the maximum and minimum mutation probability was set to (0.004, 0.002), (0.006, 0.004), and (0.008, 0.006). The results are shown in Table 5.

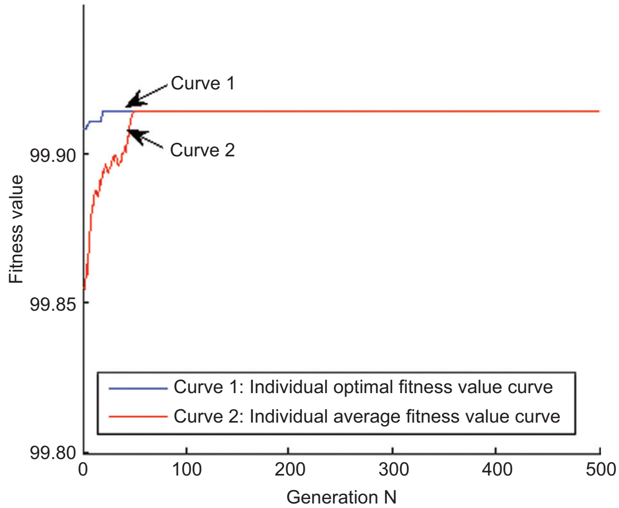

Figure 14. The Convergence Curve when the Mutation Probability is Set to (0.008, 0.006).

Figures 12–14 and Table 5 show that with the mutation probability increases, the convergence enhances and the local searching ability increases. The time required to achieve the optimal solution decreases. It can approximate the optimal value faster.

6 Conclusion

Problem of task allocation in collaborative customized product development was studied based on artificial intelligence technology. Theoretical guidance was provided for optimizing the allocation of development resources, which shortens the product development cycle and improves the efficiency of product collaborative design. The main contributions of this article as the following:

1. The definitions and calculation methods of task fitness and task coordination efficiency were given, and a multiobjective optimization mathematical model of collaborative customized product development task allocation was constructed.
2. The AGA for solving the model was introduced, and a new solution for collaborative customized product development task allocation optimization problems was provided.
3. A 5-MW wind turbine product development project of a wind power generation company was used as an example. The results verified the feasibility and the effectiveness of AGAs in solving task allocation optimization problems compared with methods such as SGA.

Task allocation in a collaborative customized product development process is a complex problem. There are many factors that influence task allocation, and there are relatively more algorithms. In this article, only two dimensions of fitness and coordination efficiency of cooperative tasks are studied. The factors considered are relatively few, and they need to be further studied and expounded. Further enriching and improving the objective function and the constraints and its algorithm of task allocation will be the next research focus.

Acknowledgements: This research was funded by the National Nature Science Foundation of China (71071173) and was jointly supported by the Research Fund for the Doctoral Program of Higher Education of China (20090191110004 and 20120184120040).

Received March 11, 2013; previously published online May 6, 2013.

Bibliography

[1] S. Dianxun, S. Qing and D. Yumin, Particle model to optimize resource allocation and task assignment, *Inf. Syst.* **32** (2007), 987–995.

[2] D. W. Fernandez and M. Lamari, The task allocation problem with constant communication. *Discrete Appl. Math.* **131** (2003), 169–177.

[3] L. Hou, F. Chen and Z. Wen, Design task decomposition and allocation for inter-firm product collaborative development, *J. Zhejiang Univ. (Eng. Sci.)*, **41** (2007), 1976–1981.

[4] Y. Jing, X. Wang and W. Li, Task allocation optimization for supplier involvement in cooperative product development, *China Mech. Eng.* **22** (2011), 2566–2571.

[5] Y. J. Lei, S. W. Zhang, X. W. Li and C. M. Zhou, *Genetic Algorithm Toolbox and Applications*, Xi'an University of Electronic Science and Technology Publishing House, Xi'an, 2005.

[6] J. Li, B. Jing, X. Chen and C. Wang, Model for virtual corporations' partner selection based on adaptive genetic algorithm, *J. Commun.* **30** (2009), 78–83.

[7] X. Liang, G. Peng, Y. Yu and B. Beifang, Function matching and selection for product design based on dependency constraints, *China Mech. Eng.* **22** (2011), 227–231.

[8] K. Min-Hyuk, K. Sang-Phil and L. Seokcheon, Social-welfare based task allocation for multi-robot systems with resource constraints, *Comput. Indust. Eng.* **63** (2012), 994–1002.

[9] H. Pang, Z. Fang, H. Guo and Y. Zhao, Research on task scheduling method of cooperative design, *Syst. Eng. Electron.* **30** (2008), 1899–1903.

[10] E. Qi, H. Li and L. Liu, Research on collaborative resource optimization of virtual enterprises based on genetic algorithm, *Chin. J. Manage. Sci.* **19** (2011), 77–83.

[11] J. H. Son, S. K. Oh, K. H. Choi, Y. J. Lee and M. H. Kim, GM-WTA: an efficient workflow task allocation method in a distributed execution environment, *J. Syst. Software* **67** (2003), 165–179.

[12] L. Tong, Z. Chen, Z. Yuan and L. An, A double population self-adaptive genetic algorithm for partner selection of virtual enterprise, *Comput. Eng.* **32** (2006), 192–194.

[13] A. Tripathi, M. Tiwari and F. Chan, Multi-agent-based approach to solve part selection and task allocation problem in flexible manufacturing systems, *Int. J. Prod. Res.* **43** (2005), 1313–1335.

[14] L. Wang and B. Liu, *Particle swarm optimization and scheduling algorithms*, Tsinghua University Press, Beijing, 2008.

[15] C. Wunhwa and L. Chinshien, A hybrid heuristic to solve a task allocation problem, *Comput. Oper. Res.* **27** (2000), 287–303.

[16] Y. Yang, X. Zhang, F. Liu and Q. Xie, An internet-based product customization system and its integration, *China Mech. Eng.* **14** (2003), 1290–1294.

[17] P.-Y. Yin, S.-S. Yu, P.-P. Wang and Y.-T. Wang, Multi-objective task allocation in distributed computing systems by hybrid particle swarm optimization, *Appl. Math. Comput.* **184** (2007), 407–420.

[18] M. Yung-Cheng, C. Tien-Fu and C. Chung-Ping, Branch-and-bound task allocation with task clustering-based pruning, *J. Parallel Distrib. Comput.* **64** (2004), 1223–1240.

[19] W. Zhang, W. Liu and Z. Zhang, Task assignment for suppliers' participation in collaborative product development, *Comput. Integr. Manuf. Syst.* **15** (2009), 1231–1236.