

Yu, Guodong; Yu, Yang; Xing, Qingsong; Li, Fei

Article

Research on the time optimization model algorithm of Customer Collaborative Product Innovation

Journal of Industrial Engineering and Management (JIEM)

Provided in Cooperation with:

The School of Industrial, Aerospace and Audiovisual Engineering of Terrassa (ESEIAAT),
Universitat Politècnica de Catalunya (UPC)

Suggested Citation: Yu, Guodong; Yu, Yang; Xing, Qingsong; Li, Fei (2014) : Research on the time optimization model algorithm of Customer Collaborative Product Innovation, Journal of Industrial Engineering and Management (JIEM), ISSN 2013-0953, OmniaScience, Barcelona, Vol. 7, Iss. 1, pp. 137-152,
<https://doi.org/10.3926/jiem.838>

This Version is available at:

<https://hdl.handle.net/10419/188595>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<https://creativecommons.org/licenses/by-nc/3.0/>

Research on the Time Optimization Model Algorithm of Customer Collaborative Product Innovation

Guodong Yu¹, Yang Yu¹, Qingsong Xing², Fei Li¹

¹*State Key Laboratory of Mechanical Transmissions (China)*

²*Chongqing Jiao Tong University (China)*

yuguodong@cqu.edu.cn, yuyang@cqu.edu.cn, xingqingsong_2003@126.com, feili@cqu.edu.cn

Received: April 2013

Accepted: October 2013

Abstract:

Purpose: To improve the efficiency of information sharing among the innovation agents of customer collaborative product innovation and shorten the product design cycle, an improved genetic annealing algorithm of the time optimization was presented.

Design/methodology/approach: On the basis of the analysis of the objective relationship between the design tasks, the paper takes job shop problems for machining model and proposes the improved genetic algorithm to solve the problems, which is based on the niche technology and thus a better product collaborative innovation design time schedule is got to improve the efficiency. Finally, through the collaborative innovation design of a certain type of mobile phone, the proposed model and method were verified to be correct and effective.

Findings: An algorithm with obvious advantages in terms of searching capability and optimization efficiency of customer collaborative product innovation was proposed. According to the defects of the traditional genetic annealing algorithm, the niche genetic annealing algorithm was presented. Firstly, it avoided the effective gene deletions at the early search stage and guaranteed the diversity of solution; Secondly, adaptive double point crossover and swap mutation strategy were introduced to overcome the defects of long solving process and easily converging local minimum value due to the fixed crossover and mutation probability; Thirdly,

elite reserved strategy was imported that optimal solution missing was avoided effectively and evolution speed was accelerated; Finally, the new algorithm is tested by three typical functions and the job shop scheduling problems. Respectively, the simulation results show that the proposed algorithm can avoid the stagnation, which improves the global convergence ability and attains better optimization performance.

Originality/value Firstly, the improved genetic simulated annealing algorithm overcomes some defects such as effective gene easily lost in early search. It is helpful to shorten the calculation process and improve the accuracy of the convergence value. Moreover, it speeds up the evolution and ensures the reliability of the optimal solution. Meanwhile, it has obvious advantages in efficiency of information sharing among the innovation agents of customer collaborative product innovation. So, the product design cycle could be shortened.

Keywords: genetic simulated annealing algorithm, niche, customer collaborative product innovation scheduling, simulation

1. Introduction

The prosperity of the market and the rapid development of technology lead to the increasing complexity of the products. Each agent of product innovation design needs to deal with an increasing amount of information, and the division of labor is becoming more and more refined, which makes it hard for a single enterprise to complete the product development and manufacturing independently (Andresen, Brasel & Morig, 2008). Conversely, the strength of the customers, suppliers and scientific research institutes and other units need to be fully utilized for the collaborative innovation. Thus, product collaborative innovation design research around the cross-organizational and cross-unit has become an important direction in the field of product development (Hou, Han & Lin, 2006).

Song, Yang and Yang (2008) subjectively gave the remaining processing time to task weights and tasks identified, they adopted the least finish time on-line scheduling method to study the collaborative design task sorting in different states, so as to shorten the product development cycle, and improve the efficiency of product design. In order to shorten the product development cycle and improve the efficiency of product innovation design, people like Cai-Mao and Ke (2006) adopted a combination method of fuzzy design structure matrix and the expanded directed graph to study the task coupling and dependency which influences the process design efficiency during the collaborative product development. Joglekar and Ford (2005) saw the subject as one of design resources, and built the resource optimally allocated models that can improve product collaborative design efficiency according to the design task

priority. Hou, Chen and Wen (2007) took the decomposition and distribution of product design task as the breakthrough point, and through considering the interaction and influence between different tasks, studied the decomposition of inter-organizational collaborative product design task and selection and matching of coordinational suppliers which was based on the fuzzy matching method. Ming and Yan (2008) made a systematic and qualitative description for the process framework of the collaborative design and manufacturing which was based on product life cycle theory and the consideration of the technical features of the product design, while lacking of the quantitative analysis of product design time and design efficiency. Sheng, Lin and Ding (2006) used the theory and method of attributes importance in the rough set to determine the similarity of design tasks and knowledge source case, and calculated the weight of the task, so as to define task priorities, eliminate confliction of product collaborative design task and improve the efficiency of collaborative design. Considering to problems exist for collaborative design task scheduling, in the establishment of designing task network chart, Pang, Fang and Guo (2008) took comprehensive consideration of the specific autoimmune factors and constraints of collaborative projects and tasks. Based on Equilibrium - moderation principle, a task assignment from tasks to team members extended mathematical model was established, and Hungary algorithm was used for solving. Zhang, Liu and Zhang (2009) depended on the structure matrix to construct the overall time model of collaborative product development, designed the particle swarm optimization algorithm for task optimization distribution, and achieved optimal allocation of development subtasks among organizations, so as to guarantee manufacturers to finish product development projects in the scheduled time and response to the rapid changes in the market. As, for the multitasking complex cross-coordination difficult problems in distributed collaborative design environment, Liang, Ge and Yang (2011) had a logic decomposition for collaborative design business rules and task - resource dependence relationship, which was based on the coordination theory (CT) depending on the analysis method and project management of work breakdown structure (WBS) method. And single main task coordination model proposed in complex collaborative design environment was put forward. At the same time, considering the coupling relationship between the coordination tasks, the grey relational analysis (GRA) was adopted for the assignment of collaborative design task weight, and the weighted design task was sorted so as to optimize the overall time of product collaborative innovation design.

To sum up, researches about efficiency of collaborative product innovation design are mainly focused on the design task priority and their mutual dependence. Through subjectively weighting, the use of various analytic algorithms and taking optimal time as the goal, to make a reasonable distribution for design task and subject, so as to improve the efficiency of collaborative product innovation design. But during the process of empowering the design task important degree and priority, because of the difference between knowledge background and evaluation system, the weight endowed is led to be with larger subjectivity, which eventually makes the theory for design task scheduling largely deviate to the actual product collaborative

design time consuming. Meanwhile, algorithms the research institutes above selected are mainly non-intelligent. When the quantity of tasks and agents participating in collaborative innovation design increasing, the problem solving time would increase or even be difficult to obtain effective solution. For this reason, the paper based on of the analysis of the objective relationship between the design tasks, takes the kinds of job shop problems for machining model and proposing the improved genetic algorithm to solve the problems, which is based on the niche technology and thus a better time schedule of collaborative product innovation design is got to improve the efficiency. Finally, through the collaborative innovation design of a certain type of mobile phone, the proposed model and method were verified to be correct and effective.

2. Modeling of time optimization for collaborative innovation design of products

For product collaborative innovation designing organizations, there are m design agents, a certain object to be designed and n tasks need to be done. The goal of product collaborative innovation designing is to determine the start time and end time of each task by distributing the n tasks to the m design subjects according to the design sequence optimization, and ultimately to guarantee the minimum of time to finish the design of the scheduled target.

2.1. Related definitions and constraints

Definition 1, design task set: the collection of all pending design tasks, which is represented by $Ta = \{ Ta_i | 1 \leq i \leq n \}$ and n indicates the number of tasks.

Definition 2, design the subject - task relationship

$$X_{ijk} = \begin{cases} 1 & \text{Series } j \text{ of Task } i \text{ operated by subject } k \\ 0 & \text{others} \end{cases}$$

$$R_{i^1 j^1 i^2 j^2 k} = \begin{cases} 1 & \text{Series } j^1 \text{ of Task } i^1 \text{ and Series } j^2 \text{ operated by subject } k, \text{ Series } j^2 \text{ completed before Series } j^1 \\ 0 & \text{others} \end{cases}$$

Besides, TF_i is the task completion time, and P_{ijk} is the operating time for the subject k of designing to complete the design sequence j of the task Ta_i . TS_{ijk} represents the operation starting time for the subject k of designing to complete the j design sequence of the task Ta_i . TF_{ijk} is the operating completion time for the subject k of designing to complete the j design sequence of the task Ta_i . However, in the actual process of products collaborative innovation design, the design subject was often forced to interrupt design task by various reasons. For this, this paper assumes that the operation of every design sequence in every task are not allowed to be interrupted when the subject of the designing is working, and all the design task can be processed in the starting moment.

(1) The constraint of designing task sequence: the operation of the later design sequence of a task must be done after the consummation of the former design sequence.

$$TF_{ijk} - TF_{i(j-1)l} - P_{ijk} \geq 0 \quad (1 < j \leq N_i, X_{ijk} = X_{i(j-1)l} = 1) \quad (1)$$

(2) The constraint of design subject: the design subject k can only start another task after one design task operation is completed by design subject k,

$$TF_{i^2j^2k} - TF_{i^1j^2k} - P_{i^2j^2k} \geq 0, \quad X_{i^1j^2k} = X_{i^2j^2k} = 1, \quad R_{i^1j^1i^2j^2k} = 1 \quad (2)$$

(3) The completion time constraint of the task:

$$TF_{ijk} = \max\{TF_{i(j-1)l}, ST_{ijk}\} + P_{ijk} \quad i=1, \dots, n; \quad j=2, \dots, n_i; \quad k, l=1, 2, \dots, m \quad (3)$$

At the same time, the finish time TF_{ilk} of the first design sequence in the task Ta_i needs to meet the constraint:

$$TF_{ilk} = ST_{ilk} + P_{ilk} \quad (4)$$

2.2. The model of time optimization

On the basis of the above definitions and constraints, m design subjects (M_1, M_2, \dots, M_m) and n design tasks (Ta_1, Ta_2, \dots, Ta_n) form a product collaborative innovation design organization. Under the circumstances of knowing the design task processing time and the constraint of operation sequence for the design subject to complete the design task, the goal to minimize the maximum completion time of the design task is defined. And the mathematical model aiming at shortening the product collaborative innovation design cycle and improving the efficiency of collaborative design should be constructed.

$$G = \min[\max(\sum_{i=1}^n TF_i)] \quad (5)$$

3. Implementation procedure of model algorithm

3.1. Process of improvement genetic annealing algorithm based on niche technology

During the model optimization of product collaborative innovation design, under the relevant constraints, the combinatorial optimization calculation of a large number of data will be involved. Therefore, it is necessary to seek suitable heuristic algorithm for the calculation, while there will be different degree of defects when using single heuristic algorithm to solve

the specific problems, so a variety of methods are needed to be merged to make up for lack of their fusion, and finally realizing the correct and effective solution of the problem. Genetic annealing algorithm based on niche technology should firstly define the individual sharing function according to the hamming distance between the individuals, and the fitness of each individual of the population should be adjusted through the sharing function, and adjusted fitness will be treated as the basis of genetic operators, so as to maintain the diversity of the population and improve the probability of the optimal solution (Chunqiang, Ling & Jun,2007). Through the ecological niche concentration control mechanism realize the excellent individual choice of the initial population, and use the advantage of genetic algorithm on the global search generating a group of excellent. Thus, the gene individuals will be adjusted combined with the characteristics of the local search of the simulated annealing, which will be operated iteratively until the termination conditions is satisfied. At the same time, because the method used a positive feedback mechanism of group optimization formatted by the cognitive memory function of each chromosome itself and the cooperative behavior of group to complete the global optimization process, it is similar to the thought of multi-agent cooperation and coordination in collaborative innovation, which is more suitable for solving the product collaborative innovation design coordination and optimization problems with multiple subjects to participate in. Specific steps of this algorithm and are as follows:

Step1. Set evolution algebra counter $t=1$, randomly generate initial population $Pop_0(t)$ with m individuals; then calculate the fitness $f_i(i = 1,2,\dots,m)$ of each individual, and range them in descending orders according to the size, and store the highest fitness of the individual and target value in the optimal solution concentration;

Step2. Determine whether it meets the convergence conditions, if it meets then terminate the operations and output the calculation results; If it doesn't meet then turn to Step 3;

Step3. Niche concentration control the selection operator, with m individuals obtained in the step 1, work out the Hamming distance between the every two individuals and the fitness according to the following formula, namely,

$$d = \sqrt{\sum_{a=1}^m \sum_{b=1}^m \sum_{k=1}^n (ObjV(a, k) - ObjV(b, k))^2} \quad (6)$$

While $a \neq b$, $ObjV$ was the individual target value, and k presented the decision variables of the problems to be solved. When $d < L$ (L is the distance parameters between the pre-specified niche, comparing the size of the fitness of the individuals, and imposed a penalty function $f_{\min(x_i, x_j)} = penalty$ to the lower one, so as to dramatically reduce its fitness, namely their probability of being eliminated was increased.

Step4. Conduct elite selection operation on populations $Pop_0(t)$ and elite selection; then, adaptive crossover operator was used on the populations $Pop_0(t)$ for crossover operation, so as to get the population $Pop_1(t)$. At the same time, the optimal solution was stored into the knowledge base. During the search process, firstly the knowledge base was inquired to avoid the loss of the optimal solution and improve the ability of evolution. Finally, adaptive uniformity mutation operator was used to conduct the mutation operation for population $Pop_1(t)$, and population $Pop_2(t)$ was got.

Step5. Define G_1 and G_2 as the gold point of division and annealing temperature controlling value Θ , $G_1 = G_2 - \text{round}(0.618 * G_2)$, $G_2 = \text{MaxGen} - \text{round}(0.618 * \text{MaxGen})$, among them MaxGen is the largest cyclic algebra. Determine whether executive annealing operation is needed according to the cyclic algebra j . If it meets the annealing conditions ($L = 1$) then step (6) would be executed, otherwise ($L = 0$), turn to step (7), steps are as follows:

Step6. Define $Pop_2(t)$ as the initial population, then annealing operation was simulated to each individual in it, namely, when the temperature was T , new state j was generated from the current state i , and the energy of each are E_i and E_j . If $E_j < E_i$, then the new state j is accepted as the current state. Otherwise, if the probability $pr = \exp[-(E_j - E_i)/kT]$ is greater than any random number in the interval $[0, 1]$, then the new state j is still accepted as the current state, if not, then state i is restrained as the current state. Among them is the Boltzmann constant, T is the absolute temperature. A new population $Pop_3(t)$ with a scale of m , and the individuals were ranged in descending order according to the size of fitness.

Step7. Merge $Pop_1(t)$ and $Pop_2(t)$ to form a new group, and range the fitness of it. At the same time, updating the individuals of $Pop_3(t)$, then turn to step 1. The specific processes is shown in Figure 1.

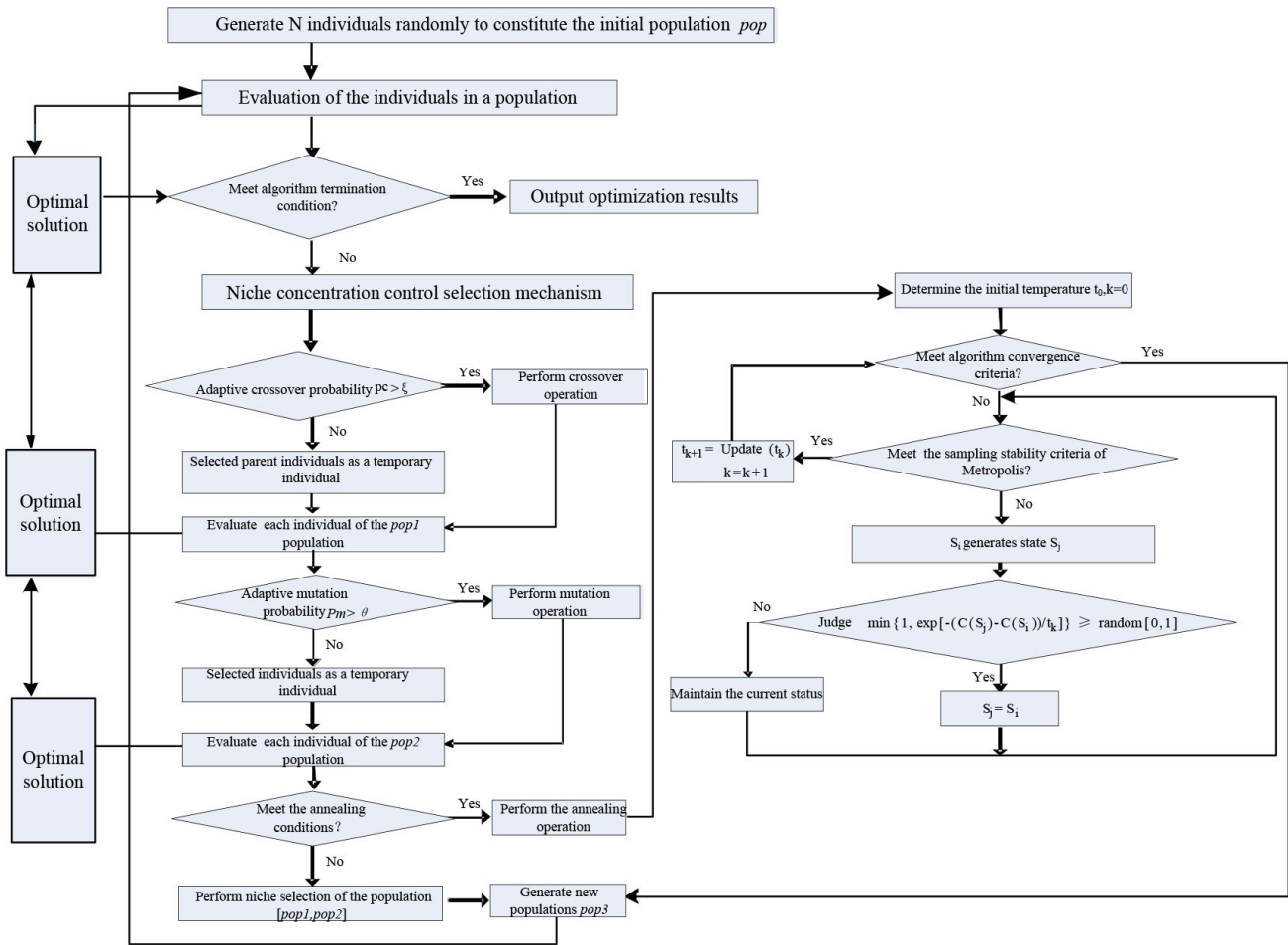


Figure 1. Flow chart of improved genetic annealing algorithm

3.2. The realization of model algorithm

(1) Algorithm coding. In the process of initialization, the method based on the design of the task sequence was used for coding. For example, the problem 5*5 in Table 1, if the given chromosome are [23514 13254 31524 43251 25413]. Among them, 1 represents task Ta_1 , 2 represents task Ta_2 , 3 represents task Ta_3 , 4 represents task Ta_4 , 5 represents task Ta_5 . Because of each design task includes 5 design sequences, each one appears five times in a chromosome. The corresponding relationship of chromosome, design task and design subject is shown in Figure 2.

Design task	Design sequence				
Ta ₁	M ₃	M ₁	M ₅	M ₄	M ₂
Ta ₂	M ₁	M ₅	M ₂	M ₄	M ₃
Ta ₃	M ₄	M ₂	M ₃	M ₁	M ₅
Ta ₄	M ₂	M ₅	M ₃	M ₁	M ₄
Ta ₅	M ₅	M ₂	M ₄	M ₃	M ₁

Table 1. Collaborative innovation problems of five design subjects and design tasks

Chromosome	2	3	5	1	4	...	2	5	4	1	3
Design task	Ta ₂	Ta ₃	Ta ₅	Ta ₁	Ta ₄		Ta ₂	Ta ₅	Ta ₄	Ta ₁	Ta ₃
Design subject	M ₁	M ₄	M ₅	M ₃	M ₂		M ₃	M ₁	M ₄	M ₂	M ₅

Table 2. Corresponding relationship of chromosome, design task and design subject

(2) Adaptive double point crossover operation. To make full use of historical information, this paper adopts double point crossover to increase the population diversity and to prevent the operation of algorithm premature and stagnation. As shown in Figure 2.

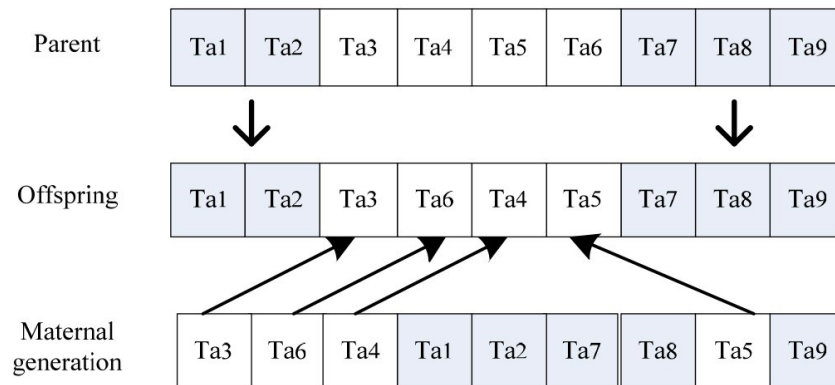


Figure 2. Double point crossover

In the evolutionary process, when the population individual fitness of the current generation is below the average fitness value, the crossover rate should be improved to achieve faster search speed. Conversely, the crossover rate need to be reduced and can be automatically adjusted with the fitness value. For this reason, this paper puts forward dynamic adjustment strategy of crossover probability. The adaptive adjustment formula is:

$$P_C = \begin{cases} P_{Cl} - \frac{(P_{Cl} - P_{C2})(f - f_{avg})}{f_{max} - f_{avg}} & f \geq f_{avg} \\ P_{Cl} & f < f_{avg} \end{cases} \quad (7)$$

(3) The adaptive random mutation operation. In order to increase the diversity of population, maintain the diversity of population, and improve the local search ability, the paper adopts the random interchangeable variability that is to select bit of the two genes in the chromosome randomly, so as to achieve the purpose of preventing premature. As shown in Figure 3.

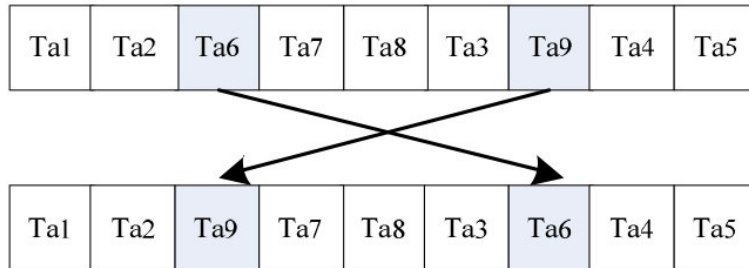


Figure 3. Random interchangeable variation

Meanwhile, adaptive mutation probability is used to improve the convergence speed, and the adaptive adjustment formula is:

$$P_M = \begin{cases} P_{M1} - \frac{(P_{M1} - P_{M2})(f_{max} - f^*)}{f_{max} - f_{avg}} & f^* \geq f_{avg} \\ P_{M1} & f^* < f_{avg} \end{cases} \quad (8)$$

(4) Introduce the niche elitist retention strategy based on the sharing mechanism. The pseudo code is as follows:

```

Identify distance=zeros(popsiz, popsiz); DDistance=zeros(popsiz, popsiz);
share=zeros(popsiz, popsiz); SHare=zeros(1, popsiz);

Do Two individual hamming spacing
distance(i, j)=distance(i, j)+(Chrom(i, k)-Chrom(j, k))^2; DDistance=sqrt(distance);

Do Two individual fitness distance

adapt-abs(i, j)=abs(Chrom(i, length+1)-Chrom(j, length+1));

Do Hamming spacing and fitness mean

average1=sum(sum(DDistance))/(popsiz*popsiz);
average2=sum(sum(distance))/(popsiz*popsiz);
    
```

Do The adjustment of sharing function adjustment

if $D_{Distance}(i,j) < average1 \& adapt-abs(i,j) \geq average2$

share(i,j)=1-D_{Distance}(i,j)/average1;

elseif $D_{Distance}(i,j) \geq average1 \& adapt-abs(i,j) < average2$

share(i,j)=1-adapt-abs(i,j)/average2;

elseif $D(i,j) < e1 \& adapt-abs(i,j) < average2$

share(i,j)=1-D_{Distance}(i,j)*adapt-abs(i,j)/average1*average2;

else *share*(i,j)=0;

Do The adjustment of individual fitness value

SHare(1,i)=*SHare*(1,i)

+*share*(i,j);*fitnew1*(1,i)=*Chrom*(i,n+1)/*SHare*(1,i);*fitnew*=abs(*fitnew1*);

Pop ← initialize

Min-fitness, adapt (popsize) ← *Adapt*(pop)

Best-fitness, best-individual ← *keepbest* (*min-fitness, adapt*(i),pop)

For k=1: maxgen

pop ← *Select*(pop, adapt)

pop1 ← *Cross*(pop, adapt, min-fitness)

min-fitness, adapt(popsize) ← *Adapt*(pop1)

Update(*Best-fitness, best-individual*) ← *keepbest*(*Best-fitness, min-fitness, adapt*(popsize).*best-individual*, pop1)

Pop2 ← *mutation*(pop1,adapt,min-fitness)

min-fitness, adapt(popsize) ← *Adapt*(pop2)

Update(*Best-fitness, best-individual*) ← *keepbest*(*Best-fitness, min-fitness, adapt*(popsize).*best-individual*, pop2)

If need simulated annealing operation

Pop3 ← *annealing* (pop2)

Else

$pop3 \leftarrow niching\ choice(pop1, pop2)$

$min\text{-}fitness, adapt(popsize) \leftarrow Adapt(pop3)$

$Update(Best\text{-}fitness, best\text{-}individual) \leftarrow keepbest(Best\text{-}fitness, min\text{-}fitness, adapt(popsize).best\text{-}individual, pop3)$

4. Case Study

4.1. The Application case analysis

The F Company mainly engages in the research and development of mobile phone. The company is developing a new product Duff, and the innovative design of the product D is directed by a collaborative innovation design alliance composed of 10 design agents (M_1, M_2, \dots, M_{10}) (Figure 4).

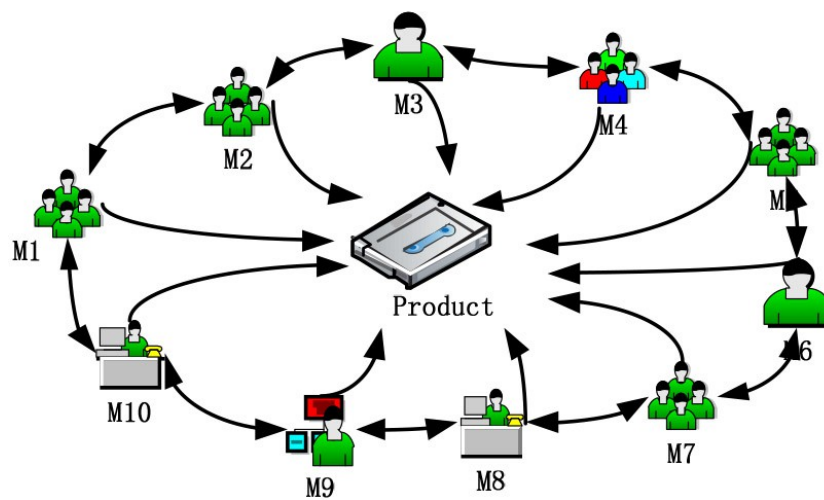


Figure 4. 10 design agents of the Duff development

Through the task partitioning to each part of the object to be designed and in the Pentium 4, CPU clocked at 2.71GHz, memory at 2.0 GB, Windows7 operating system. MATLABR2010a is used to conduct a simulation experiment based on the above model and algorithm. The parameters are as follows: the population size is 500, the maximum iterative number is 100, the crossover rate is 0.6, the mutation rate is 0.06, the attenuation parameter is 0.95, and the iterative initial temperature is 500. The simulation results are shown in Table 3.

Algorithm name	Minimum completion time	Average completion time	Average time-consuming solving
Genetic algorithm	970	990	229
Genetic annealing algorithm	930	976	397
Niche genetic annealing algorithm	930	950	384

Table 3. Product collaborative innovation design time-consuming in the different algorithms

4.2. Algorithm performance analysis

In order to validate the solving efficiency, effectiveness and reliability of the above niche

genetic annealing algorithm, three typical nonlinear functions ($f_1 = \sum_{i=1}^n x_i^2$, $x \in [-1000, 1000]$);

$$f_2 = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i)^2 + (x_i - 1)^2] \quad x_i \in [-30, 30];$$

$f_3 = 4x_1^2 - 21x_1^4 + x_1^6/3 + x_1x_2 - 4x_2^2 + 4x_2^4$, $x_1, x_2 \in [-5, 5]$) are adopted as examples to verify the algorithm performance, and each of them run 20 times independently. The results compared with other algorithms are shown in Table 4. In the experiments:

- (1) Genetic algorithm crossover and mutation probability are set as 0.85 and 0.1 respectively.
- (2) The optimization parameters accuracy of the niche genetic annealing algorithm is set as 10^{e-4} , the adaptive crossover probability varying from 0.6 to 0.99, the adaptive mutation probability varying from 0.01 to 0.1, the upper bound of distance threshold is 3, initial temperature is 100, temperature reduction parameter is 0.98, and annealing temperature controlling value is 0.15.
- (3) The population size of the annealing genetic algorithm is set as 100, evolution algebra as 800, crossover and mutation probability as 0.7 and 0.1 respectively, initial temperature as 100, temperature reduction parameter as 0.98.
- (4) The improved cooperative particle swarm divided into five subgroups, the population scale is 100, evolution algebra is 800, inertia weight is 0.4, learning factor $c_1 = c_2$ is 1.49, disturbance factor is 150.
- (5) The variable precision of the hybrid quantum genetic algorithm is 10^{e-6} , crossover and mutation probability is 1 and 0.05 respectively.

Method	Test function	Optimal solution	Worst solution	Average	Time	Standard deviation
Genetic Algorithm	$f_1(100,1000)$	4.768×10^{-6}	4.768×10^{-6}	4.768×10^{-6}	14.64	0
Niche Genetic Annealing Algorithm		7.153×10^{-11}	7.153×10^{-11}	7.153×10^{-11}	4.01	0
Genetic Annealing Algorithm	$f_2(100,800)$	0.0016	0.464	0.1899	6.25	0.1346
Niche Genetic Algorithm		6.11×10^{-5}	0.008	0.0016	150.5	0.0024
Improved Cooperative Particle Swarm Optimization Algorithm		0.0304	-	14.3211	28.96	-
Niche Genetic Annealing Algorithm		0.0004	0.0844	0.0189	4.48	0.0257
Quantum Genetic Algorithm	$f_3(100,30)$	-1.031	-	-0.9817	6.578	0.0615
Niche Genetic Annealing Algorithm		-1.031	-0.9822	-1.0165	0.986	0.0189

Table 4. Performance comparison of each method

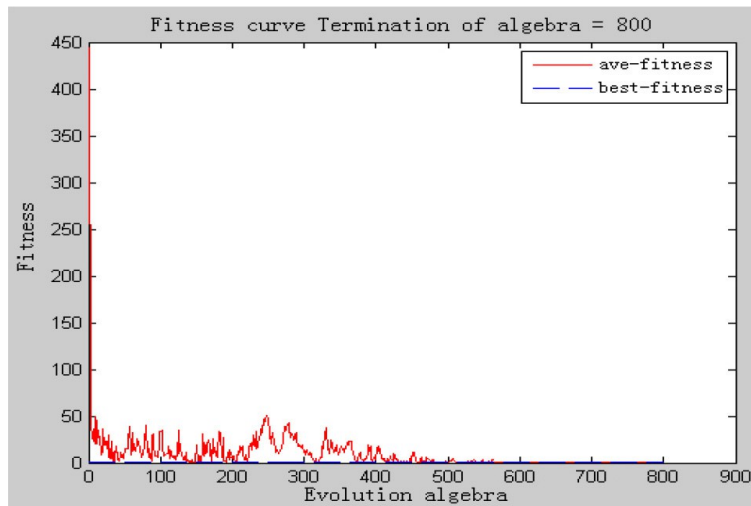


Figure 5. Genetic annealing algorithm of f_2

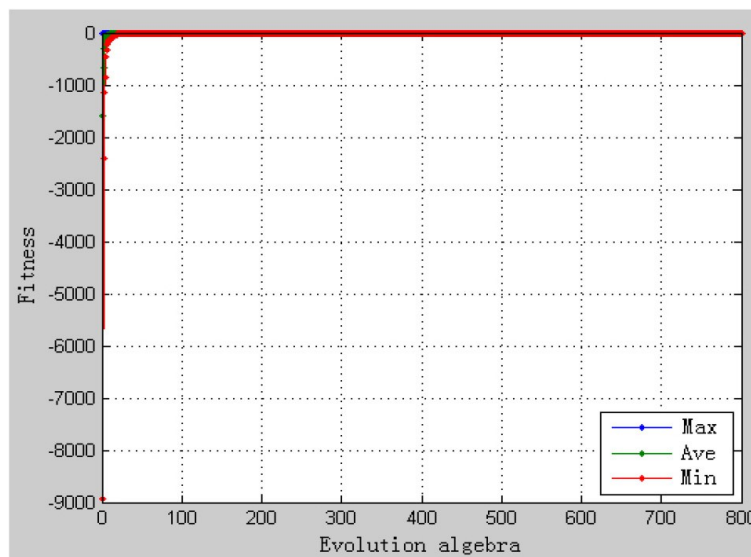


Figure 6. Niche genetic annealing algorithm f_2

The analyses from Table 4, Figure 5 and Figure 6 show that:

- (1) In optimizing function f_1 , under the same population scale and evolution algebra, the niche genetic algorithm in the convergence speed is faster than genetic algorithm, and it achieves a more optimal solution, with the search process showing robustness;
- (2) In optimizing function f_2 , the niche genetic algorithm having the fastest convergence speed, which could search for better solutions and get the minimum mean and deviation, but its optimization time was about 34 times, which substantially increased the consumption of operational resources. The contrast analysis in Fig5 and Fig6 shows that the niche genetic annealing algorithm has a higher stability computing process and lower volatility to mean;
- (3) In optimizing function f_3 , compared with the hybrid quantum genetic algorithm, the niche genetic annealing algorithm shows faster convergence speed and more strong searching capability.

The numerical experiment proves that NGS hybrid algorithm has obvious advantages in terms of searching capability and optimization efficiency.

5. Conclusions

In order to improve the efficiency of processing large amounts of complex information in customer collaborative product innovation, the paper presents a time optimization based on improved genetic annealing algorithm. The main contributions are as follows.

- (1) The defects that traditional genetic annealing algorithm is easy lost of the effective gene in early search was analyzed, and niche genetic annealing algorithm was proposed.
- (2) Adaptive two-point crossover and swap mutation strategy were introduced during solving process in order to shorten the calculation process and improve the accuracy of the convergence value.
- (3) The elite strategy was introduced to speed up the evolution and ensure the reliability of the optimal solution.

References

- Andresen, M., Brasel, H., & Morig, M. (2008). Simulated annealing and genetic algorithms for minimizing mean flow time in an open shop. *Mathematical and Computer Modeling*, 48(7-8), 1279-1293. <http://dx.doi.org/10.1016/j.mcm.2008.01.002>

- Cai-Mao, S.U., & Ke, Y.L. (2006). Task planning and decoupling strategy for collaborative design. *Computer Integrated Manufacturing Systems*, 12(1), 22-26.
- Chunqiang, J., Ling, Y., & Jun, S. (2007). Optimal design of hydraulic manifold blocks based on niching genetic simulated annealing algorithm. *High Technology Letters (English Language Edition)*, 13(4), 363-368.
- Hou, L., Chen, F., & Wen, Z.J. (2007). Design task decomposition and allocation for inter-firm product collaborative development. *Journal of Zhejiang University (Engineering Science)*, 41(12), 1976-1981.
- Hou, L., Han, D.H., & Lin, Z.S. (2006). Research on the Architecture of Product Collaborative Development Chains. *Science & Technology Progress and Policy*, 23(4), 133-135.
- Joglekar, N.R., & Ford, D.N. (2005). Product development resource allocation with foresight. *European Journal of Operational Research*, 160 (1), 72-87.
<http://dx.doi.org/10.1016/j.ejor.2003.06.021>
- Liang, X., Ge, P., & Yang, Y. (2011). Task Sequencing for Single Product Collaborative Designer Based on Dependency Logic. *China Mechanical Engineering*, 22(2), 227-231.
- Ming, X.G., & Yan, J.Q. (2008). Collaborative process planning and manufacturing in product lifecycle management. *Computers in Industry*, 59, 154-166.
<http://dx.doi.org/10.1016/j.compind.2007.06.012>
- Pang, H., Fang, Z.D., & Guo, H. (2008). Research on task scheduling method of cooperative design. *Systems Engineering and Electronics*, 30(10), 1899-1903.
- Sheng, B.Y., Lin, Z.J., & Ding, Y.F. (2006). Rough set-based conflict resolution case reasoning in collaborative design. *Computer Integrated Manufacturing Systems*, 12(12), 1952-1956.
- Song, L.J., Yang, Y., & Yang, J. (2008). Study on Task Scheduling in Collaborative Product Development. *China Mechanical Engineering*, 19(7), 798-803.
- Zhang, W.J., Liu, W., & Zhang, Z.J. (2009). Task assignment for suppliers' participation in collaborative product development. *Computer Integrated Manufacturing Systems*, 15(6), 1231-1236.

