A Web-Based Platform for Intelligent Instrument Design Using Improved Genetic Algorithm

Jian Pan[1, 2]
[1] College of Computer Science, Zhejiang University, Hangzhou 310027, China
[2] College of Computer of Science & Technology, Zhejiang University of Technology, Hangzhou 310023, China
Email: pj@zjut.edu.cn

Guohong Mao[2] and Jinxiang Dong[1]
[1] College of Computer Science, Zhejiang University, Hangzhou 310027, China
[2] College of Computer of Science & Technology, Zhejiang University of Technology, Hangzhou 310023, China
Email: mgh@zjut.edu.cn, djx@cs.zju.edu.cn

***Abstract*—The design of new products is a creative work based on designer's knowledge or experience. This paper develops a web-based design platform for intelligent instrument with the technology of Java and web database. It aims at assisting designers in product decision-making with the selection of modules and offering near-optimal solutions of product design scheme that meets user requirement. An improved genetic algorithm combined with simulated annealing algorithm is proposed to accomplish the selection of module more effectively. In order to avoid prematurity and convergence out of optimized point for genetic algorithm, a fitness normalization formula is introduced. In simulated annealing algorithm, the metropolis rule is used to sample product design schemes. Experimental results show that our proposed algorithm is more time-efficient than exhaustion and standard genetic algorithm, and it can guarantee the diversity of the design schemes.**

***Index Terms*—Modular design, Intelligent instrument, Genetic algorithm, Simulated annealing algorithm, Metropolis rule**

I. INTRODUCTION

At present, the trend for customers is to choose products that fulfill their needs, with the best quality and the lower price in today's highly competitive market. Improving these two global objectives is what tries to achieve, every day, engineers in enterprises, by maximizing the quality and decreasing the global cost and the time to market. Product design of intelligent instrument is particularly difficult when the objective is to meet the performance requirement of users such as power consumption, cost, precision and etc.

Intelligent instrument can be partitioned into modules that are relatively independent and connected each other. So we can design modular products that fulfill various functions through the combination of distinct modules [1]. These detachable modules are constructed both according to the maximum physical and functional relations among modules and maximizing the similarity of specifically modular driving forces. Accordingly, a non-linear programming is proposed to identify separable modules and simultaneously optimize the number of modules.

For example, an electromagnetic flow meter is comprised of preliminary amplifier, filtering, excitation, display, CPU module and etc. Its architecture is showed in Fig. 1.

Nowadays, the trend for customers is to choose products that not only fulfill functionalities, but products that also satisfy their performance requirement such as power consumption, cost and precision. Thus, the customers become an actor in the design process, giving an evaluation of the performance parameters that will guide the design rules.

So we present a systematic approach to accomplish modular product design of intelligence instrument in three major phases.

- Phase 1 is by means of functional and physical interaction analysis to generate a module-to-module correlation architecture.
- Phase 2 is the exploration of performance requirements user input to evaluate each module to select modules that meet requirements.
- In the Phase 3, an improved genetic algorithm is adopted to search for the optimal or near-optimal modular schemes.

Manuscript received October 8, 2011; revised December 10, 2011; accepted November 26, 2011.
Corresponding author: Jian Pan, pj@zjut.edu.cn.

doi:10.4304/jsw.7.10.2333-2340

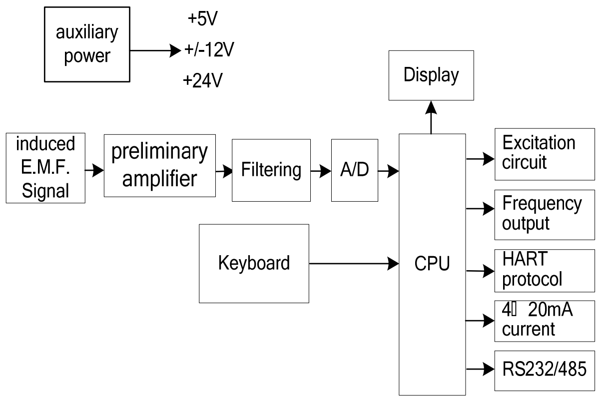

Figure1. Architecture of Electromagnetic Flow Meter.

This process is illustrated by a real case of an electromagnetic flow meter design.

The main contribution of this paper is to present a new global methodology based on an improved genetic algorithm in order to find the best or near-optimal design schemes of intelligent instrument that meets the user requirement. And we develop a web-based design platform entitled "E-design" for intelligent instrument in order to assist designers produce design schemes easily and efficiently.

The remainder of this paper is organized as follows. Section II examines previous design research related to this work. Section III describes the architecture and interface of web-based design platform. Section IV illustrates the optimization formulation of the design problem, stated as a multi-objective optimization problem. The methodology chosen to solve the design problem is presented in Section V. In Section VI, Experimental results and discussion are given. Finally, conclusions are drawn.

II. Related Work

Product design platform is the specific supporting environment and implementation tool of product design applied in enterprises [2]. At present, significant advances in product design have been achieved thanks to the introduction of several technologies, from CAD-based tools to Design For X supporting tools [3]. However, in the field of intelligent instrument oriented design tools, there is not an extensive state of the art. There are many studies on specific intelligent instrument [4][5], design platform of other product [6-9], but there is a lack of a comprehensive and integrated platform for intelligent instrument. This highlights the main challenge introduced by our E-design platform.

In the field of optimization of product design, many optimization methods are proposed and used, like calculus-based (gradient-based) [10], enumerative or heuristics methods [11, 26]. The two first schemes can be subjected to a lack of robustness if the objective function is not well defined, not continuous or not derivable and a lack of efficiency in the case of very large design spaces. In our case, the design spaces of selecting modules which comprise intelligent instrument are very large and their determination is not the scope of our study. Moreover, calculus based methods are often local in scopes which means that the optima they find are the best in the neighborhood of the current point. This will induce problem of possible local minima. In the literature, some optimization methods, based on Computational Intelligence, overcome the previous drawbacks and are able to resolve multi-objective optimization problems.

In [12], we had chosen to use a search procedure based on random choices, which does not necessitate the gradient calculation: the Genetic Algorithm (GA). The GAs, inspired by the traditional genetic, are based on the Darwin evolution theory, based on two simple postulates of survival and heritance, by using a criterion of survival of the best adapted elements in the environment and in making the genetic inheritance be propagated with a selection of the individuals.

The GAs are stochastic, non-linear optimization routines loosely based on theories of biological evolution, mechanics of natural selection and natural genetics. They are search algorithms generally used as optimization techniques to search the global optimum of a multi objective problem. However, they can also advantageously be used in other fields: instance on applications where robustness and global optimization are needed. GAs are a class of non-gradient methods: actually, in contrast to more traditional optimization methods which use gradient information to move towards better points in solution space, GAs operate on populations of solutions using models of natural selection. An advantage of the techniques is that they lead, in most of the cases, to the global optimal Pareto frontier.

This stochastic optimization algorithm provides generally a family of ''good'' solutions (Pareto set) [13] in an acceptable calculation time, and is for many fields of applications an interesting alternative to gradient-based optimization [14]. Furthermore, it enables exploration of a large design space.

Simulated Annealing (SA) another popular search algorithm utilizes the principles of statistical mechanics regarding the behavior of a large number of atoms at low temperature, for finding minimal cost solutions to large optimization problems by minimizing the associated energy. In statistical mechanics, investigating the ground states or low-energy states of matter is of fundamental importance. These states are achieved at very low temperatures. However, it is not sufficient to lower the temperature alone since these results in unstable states. In the annealing process, the temperature is first raised, and then decreased gradually to a very low value, while ensuring that one spends sufficient time at each temperature value. This process yields stable low-energy states. Being based on strong theory, SA has applications in diverse areas [15][16] by optimizing a single criterion.

In addition to the earlier aggregating approaches of multi-objective SA, there have been a few techniques that incorporate the concept of Pareto-dominance. Some such methods are proposed in [17] and [18] which use Pareto-domination-based acceptance criterion in SA. A good review of several SA algorithms and their comparative performance analysis can be found in [19]. Moreover, some optimization algorithms based on SA are applied in product design field [20-22, 27, 28].

To improve the solving precision of GA, we develop an improved genetic algorithm combined with SA [23][24] in our design platform. The essence of the algorithm is inserting SA to GA. On one hand, the result of GA restricts the forming of the random state, and on the other hand, the function, formed in Simulated Annealing according to the accepting criterion and random state, updates the population for GA.

III. Architecture of Web-Based Design Platform

The design knowledge (or experience) is a dynamic collection of knowledge, which means that the advanced knowledge in the past is the general knowledge at present and will become the knowledge to be updated in the future. In fact, this is the most important reason to construct and develop products design platform.

The architecture of design platform of intelligent instrument is showed in Fig. 2.

The E-design platform is an integrated set of software modules to support the design process of intelligent instrument, from conceptual design to optimization. The platform is designed based on Browser/Server mode with Java EE and web database technology. A user-friendly web-based interface controls all software modules and related functionalities. The main goal of the entire system is to support the designer in the rapid design of new product scheme. To achieve this scope the Knowledge Based (KB) System helps designers. The Knowledge system consists of Module base, Model base, Rule base and etc. These are generated by means of functional and physical interaction analysis in Phase 1 as stated in Section I.

Module base stores modules of intelligent instrument, including input module, signal processing module, compute module, output modules, communication modules and etc. For example, the modules of an electromagnetic flow meter including CPU, filtering, preliminary amplifier, display, excitation module and etc. can be stored in Module base. Model base stores the modules structure model. An instrument might have many of the models. As showed in Fig.1, it can be used to help model users choose the appropriate modules structure. Rule base stores inference rules of product design.

The designers can query the knowledge based System and do some work of module management very easily in the Browser. Our main goal is to support the designer in the rapid design of new product, so the designers can input user requirement such as power consumption, cost and precision to submit a design request. The interface module management and intelligent design of web-based platform is respectively showed in Fig. 3 and Fig.4.

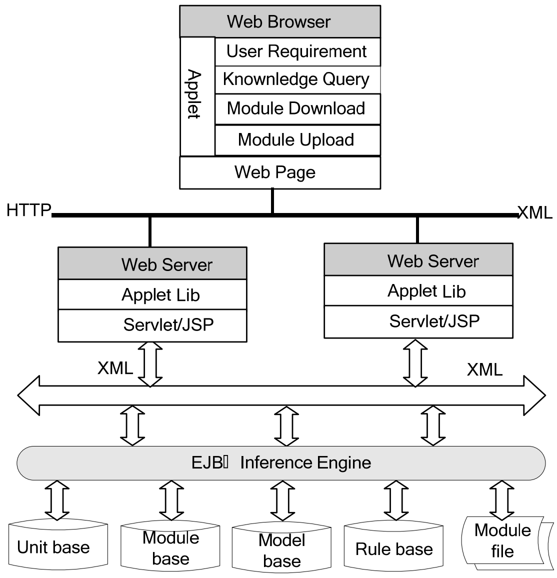

Figure 2. Architecture of Web-Based Design Platform.

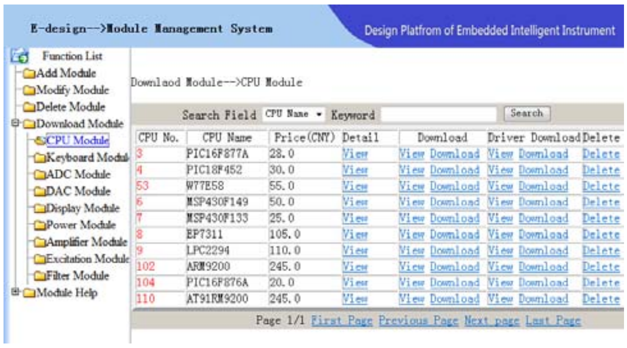

Figure 3. Interface of Module Management.

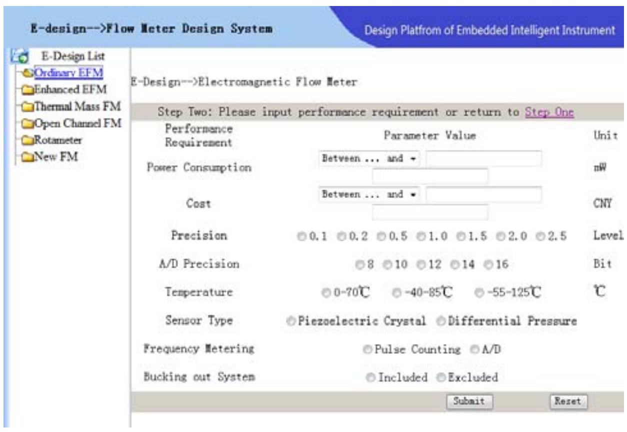

Figure 4. Interface of Intelligent Instrument Design.

The inference engine implemented with EJB component on the Server side is critical for the platform. It constitutes a combinatorial optimization problem. Conventional enumeration-based optimization techniques become inhibitive given that the number of possible combinations may be enormous. Since genetic algorithms have been proven to excel in solving combinatorial optimization problems, we develop an improved genetic algorithm. Once receiving a design request, inference engine search for the optimal or near-optimal modular design schemes using a heuristic improved genetic algorithm.

IV. Problem Formulation of Modular Design

Design requirements are clear indications to product development and design, and the accordance to evaluate final solution. In this paper, three major types of design requirements that affect product specification are presented, which are power consumption, cost and precision requirement.

We can define the product design scheme as a 2-tuple in (1).

$$S = (B, C) \tag{1}$$

B are sets of products which are combinations of modules as in (2), *C* are sets of constraints of user requirement as in (3), where Pw, Pc, and Ps denote power consumption, cost and precision, respectively.

$$B = (B_1, B_2, \cdots, B_n) \tag{2}$$

$$C = (Pw, Pc, Ps) \tag{3}$$

Supposed a product can be decomposed into N modules, such a product can also be described as follow.

$$B = M_1 \cup M_2 \cup \cdots \cup M_k \cup \cdots \cup M_N \tag{4}$$

Where M_k is the kth module, and k=1, 2, …, N.

Suppose the instance number of module M_k is n_k, and k=1, 2, …, N.

Then the design problem can be constructed as a non-linear programming to find the design schemes that meet constraints of user requirement C from the enormous number of combinations of modules ($\prod_{k=1}^{N} n_k$) . If we use method of exhaustion to search, the time of reasoning will be too long to acceptable.

V. Improved Genetic Algorithm

This improved genetic algorithm is modified as the followings compared with the genetic algorithm in [12]:

1) Set the minimum number of iterations of genetic: G_{min} and the maximal number of iterations of genetic: G_{max};

2) Set the minimum evolution rate of progeny population: $G_{minrate}$;

3) If the system has iterated k times ($G_{min} \le k \le G_{max}$) and the evolution rate of successive G_{suc} progeny population is lesser than $G_{minrate}$, It can terminate iterations of genetic because optimization of GA is winding down.

4) After the iteration of GA terminates, it uses simulated annealing algorithm to optimize the population further. Set the coefficient of annealing: λ.

The algorithm of improved GA is described in Fig. 5. The fitness function, a measurement of the quality of an individual, is used to rank the population. The ''rank'' of an individual is the number of individuals more powerful than him plus 1. To be better than the others, thus non-dominated, it is necessary to be at least as good on all the objectives and better on at least one of the objectives. An individual who is not dominated will be then of rank 1.

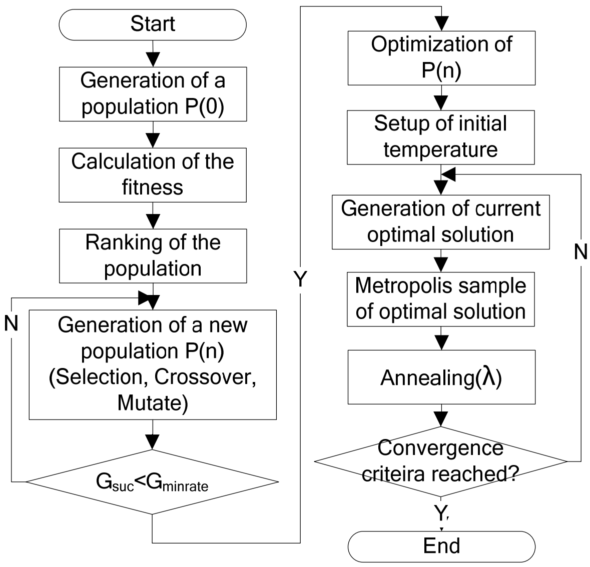

Figure 5. Flow Diagram of improved GA.

A. Encoding Schema

Falkenauer has proposed that genes can be used to represent groups [25]. The genetic operators will work with the group of chromosomes. A generic strategy for encoding the product modular design problem is illustrated in Fig. 6, with an example shown in Fig. 7. A product design scheme is represented by a chromosome consisting of a binary string. Each fragment of the chromosome (i.e., substring) represents a module contained in the product design scheme and the value represents an index of the module in the module database. Each element of the string, called gene, indicates a binary bit of the index of module. A solution (chromosome) consists of one to many modules, exhibiting a type of composition (AND) relationships.

B. Fitness Function and Selection Probability

The fitness function is constructed to evaluate an individual of design scheme meet performance requirements, which is denoted as f (B_i).

$$f(B_i) = \frac{Pw}{Tw} * Rw + \frac{Pc}{Tc} * Rc + \frac{Ps}{Ts} * Rs \tag{5}$$

Where *Pw*, *Pc*, *Ps* represent expected power consumption, cost, precision, respectively, which are from user's input. *Tw*, *Tc*, *Ts* represent practical value of the three factors, respectively, which can be calculated from each module of the individual. For example, *Tw* is sum of power consumption. The power consumption of each module can be retrieved from the module database according to his index (fragment of the chromosome). *Rw*, *Rc*, *Rs* represent the weight of the three factors in fitness function, respectively.

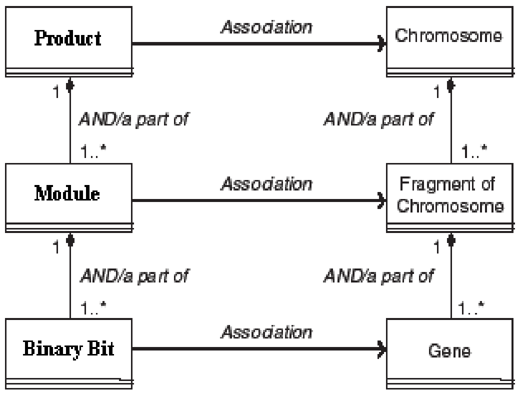

Figure 6. Generic Strategy for Encoding the Product Modular Design

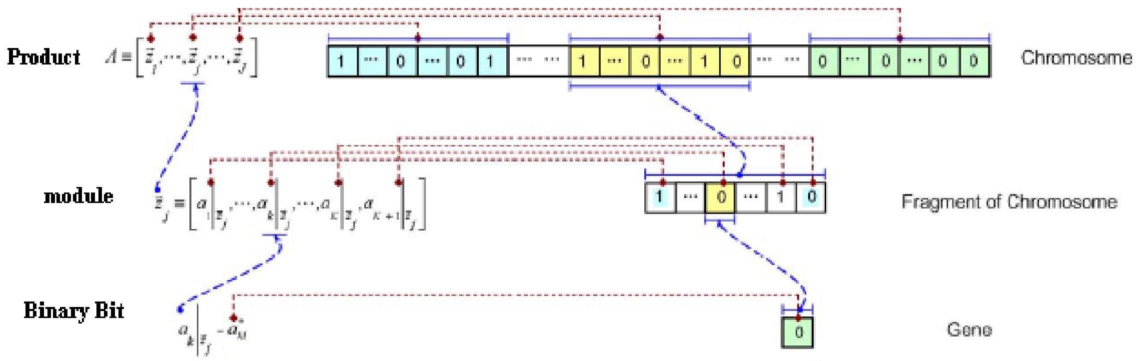

Figure 7. An example of Encoding.

And standard genetic algorithm usually uses the following probability of selection:

$$P(B_i) = \frac{f(B_i)}{\sum_{j=1}^{n} f(B_j)} * N \tag{6}$$

Obviously, the individual whose fitness is higher will reproduce more children. Thus, it can produce new generation that meet requirements. The individual whose fitness is lower reproduce less children and will even be eliminated. So this method of proportional selection will . lead to prematurity, i.e. convergence out of optimized point.

In order to avoid this shortcoming, we use chromosome classification to sort them by fitness from high to low, which denoted as the following probability formula.

$$P(B_i) = \frac{Sl - \frac{2*(Sl-1)*(Rg(B_i)-1)}{N-1}}{N} \tag{7}$$

Where N is the number of chromosomes (individuals), Rg (Bi) {1, 2, …, N} represents the number in order of chromosome Bi, $Sl \in [1, 2]$ is an adjustable parameter.

C. Metropolis rule

We use Metropolis rule to sample product design schemes in SAA. The probability of acceptance is denoted as:

$$p = \begin{cases} 1, \Delta f > 0 \\ \exp(\Delta f/T), \text{else} \end{cases} \tag{8}$$

Where $\Delta f = f(B_{new}) - f(B_{old})$, represents the difference between of fitness value of new solution and old solution.

As suggested in [19], the value of the number of iterations should be chosen depending on the nature of the problem. Several criteria for termination of an SA process have been developed. In some of them, the total number of iterations that the SA procedure must execute is given, where as in some others, the minimum value of the temperature is specified. Detailed discussion on this issue can be found in [19].

In order to reduce the numbers of iterations of SA algorithm, a dynamic terminative criterion to test the equilibrium of Metropolis process is used in this paper. If the number of new configurations generated consecutively, whose values of objective function are worse compared to the current ones, exceeds a positive number given in advance, the algorithm will terminate the Metropolis process in some fixed control parameter automatically, described as:

$$f(B_{i-p+1}) - f(B_{i-p}) > \varepsilon\left(p = 1, 2, \cdots, N\right) \tag{9}$$

D. Crossover

The crucial idea of the GA's crossover is that a child could inherit, or combine meaningful features of two individuals. Falkenauer proposed the following five-steps to creep to the optimal or near optimal value [25].

Step 1 Randomly select two crossing sites, delimiting the crossing section, in each of the parent.

Step 2 Insert the contents of the crossing section of the second parent at the first crossing site of the first parent.

Step 3 Eliminate all items now occurring twice in the groups, as they belong to the first parent.

Step 4 If necessary, adapt the resulting groups according to the hard constraints and the cost function. At this stage, problem-dependent heuristics can be applied.

Step 5 Apply Step 2 through 4, inverting the roles of the two parents in order to generate the second child.

In this study, Step 4 is specified to develop a heuristic approach to deal with modular design problem and meet performance requirements that user input such as power consumption, cost and precision. The combination and generating of new generations is performed according to the fitness value of each individual. The following steps construct a mechanism to improve the grouping result.

Step 4-1 evaluating the fitness value of each individual: The newly generated individual were formatted after eliminating redundant items. The fitness values should be calculated.

Step 4-2 Checking the number of individuals (K): The number should not be bigger than the maximum number (U); nor smaller than the minimum number (L) of individuals. The rules for rearranging the individuals are:

- If $K>U$: eliminate the worst individual and re-insert the remainder items.
- If $K\leq L$: execute individual generation and module insertion.
- If $L<K\leq U$: execute individual generation or module insertion.

Step 4-3 Calculate the fitness value of each individual when each module inserted into each individual: This is critical in determining the creeping direction.

Principle 1. Individual generation principle: If there is no positive effect on insertion, then, new individual may be generated according to the best combination results. Alternatively, if no positive result can be obtained through any combination, then, randomly generate new individual.

E. Initial Population Generation

The performance of GA is often sensitive to the quality of its initial population. More diversity and high average fitness in the initial population are viewed more suitable for maximum problems. This study uses a randomized approach to group individual.

F. Mutation

The GA's mutation operator works with groups rather than objects. In this study, implementation details of an operator depend on the modular design problem. The mutation operator uses the same procedure as initialization to mutate chromosomes with probability less then mutation rate. In each epoch, the probability to mutate chromosomes is assigned using uniform random variable on the real interval [0, 1].

VI. Experimental Results and Discussion

Experimental environment as follows: AMD Athlon (tm) 64 Processor; 4 GB memory; Windows Server 2008 operating system; VC++.Net 2010 programming environment.

The improved GA is employed to produce product design schemes of electromagnetic flow meter. An electromagnetic flow meter is comprised of k modules (k=10). There are several instances for each module and the instance number can be greater than 10.

According to many experiments test, the crossover rate and mutation rate are set to be 0.8 and 0.2, respectively. G_{max} is set to be 100 to generate a wide variety of combinations without spending too much memory space. G_{min}, $G_{minrate}$ and λ are set to 10, 0.03 and 0.6.

To select the best solutions in this population, they have been ranked according to a single criterion which aggregates the performances over all the objectives.

If the standard GA is employed in [12], the near-optimal value can reach 0.850692 in the 50th generation, where the instance number of each module is 10. Under the same conditions, the near-optimal value with the improved GA can reach 0.910875 in the 40th generation (terminated if $G_{suc}<G_{minrate}$). It can be seen from the upper and middle part of Fig. 8 that the improved GA can reach higher fit value than standard GA.

The time of reasoning is showed in Table 1 compared with method of exhaustion and standard GA. As Table 1 showed, when the instance number of each module is n, the complexity of method of exhaustion is O (k^n). If n=30, the process of compute cannot stop and return results.

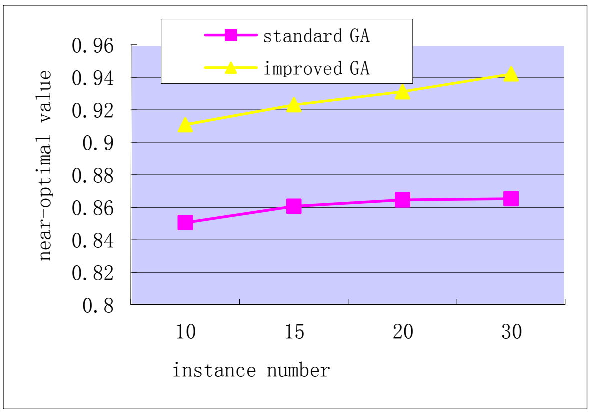

Figure 8. Near-optimal Value vs. Instance Number

The complexity of genetic algorithm depends on diversity of individual, initial population size and stopping criterion. Once initial population size and stopping criterion are determined, the time of reasoning only decide by diversity of individual. Therefore, the reasoning time of genetic algorithm rises slowly. As shown in Fig. 9, the reasoning time of improved GA is shorter than standard GA. Since G_{max} and $G_{minrate}$ are set to 100 and 0.6 respectively, the reasoning time of improved GA grows slower than standard GA when the instance number is greater than 40. The algorithm of improved GA speeds up the convergence.

It can be seen from Fig. 10 that the algorithm of improved GA can overcome the premature convergence of genetic algorithms, and guarantee to produce the diversity of design schemes generated by the design platform.

An example of design scheme of intelligent instrument produced by the design platform is shown in Table 2. In two schemes generated by standard GA six modules are the same, but in two schemes generated by improved GA only four modules are the same.

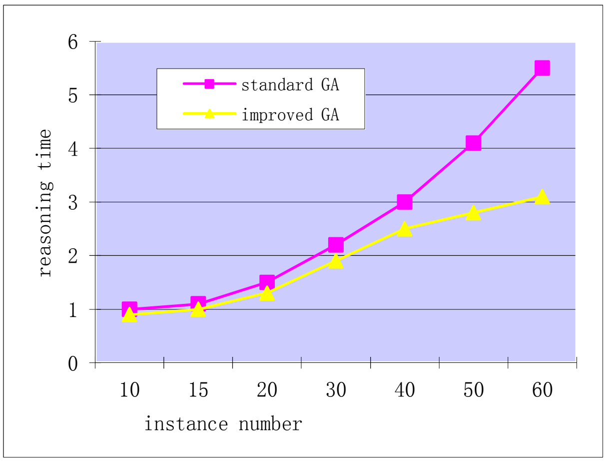

Figure 9. Reasoning Time vs. Instance Number

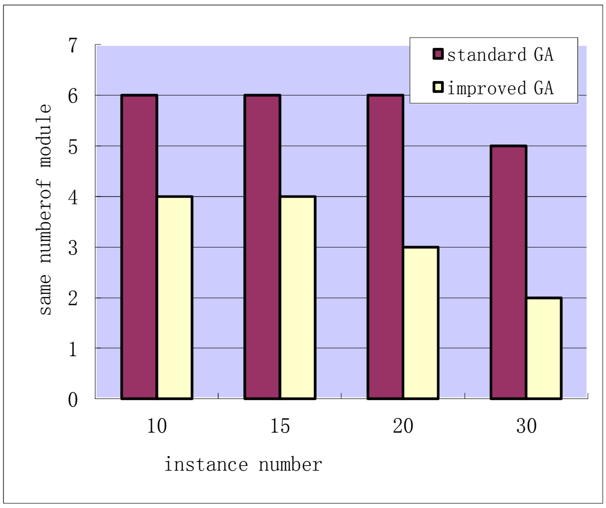

Figure 10. Same Number of Module vs. Instance Number

TABLE I.
TIME OF REASONING COMPARED WITH METHOD OF EXHAUSTION AND STANDARD GA

instance number	Time of reasoning (s)		
	exhaustion	Standard GA	Improved GA
10	30	1	0.9
15	6000	1.1	1.0
20	About 2 days	1.5	1.3
30	Cannot stop	2.2	1.9

TABLE II.
DESIGN SCHEME OF REASONING COMPARED WITH STANDARD GA

Category of Module	Scheme I (Standard GA)	Scheme II (Standard GA)	Scheme I (Improved GA)	Scheme II (Improved GA)
Power	Linear Power I	Linear Power I	Linear Power I	Linear Power I
Keyboard	Four key keyboard	Four key keyboard	ZLG 7289A	ZLG 7289A
CPU	MSP 430F133	MSP 430F133	PIC 18F452	MSP 430F149
Excitation	Rectangle wave	Rectangle wave	Rectangle wave	Rectangle wave
	100 mA	100 mA	100 mA	100 mA
Display	LCM141	LCM141	LCM151	LCM141
ADC	ADS 7871	ADC 71JG	MAX 111ACWE	TLC 2578IDW
Filter	Butterworth low-pass filter	Second band active filter	Chebyshev low-pass filter	MAX 291
Amplifier	LM 324N2	LM 324N2	INA 111PA1	INA 155U1

VII. CONCLUSION

In this paper, a web-based design platform entitled "E-design" for intelligent instrument is developed which can puts the user's requirements in the center of the product design. The E-design platform is designed based on Browser/Server mode with Java EE and web database technology. Compared with traditional approach, this study progressively exploits the interactions and uses modular design to reflect design requirements according to the competitive situation. To support the designer in the rapid configuration of a new product, a Knowledge Based (KB) System is constructed to helps designers, which consists of Module base, Model base, Rule base and etc.

The main contribution of this paper is to present a new global methodology to find the best or near-optimal design schemes of intelligent instrument that meets the performance requirement, such as power consumption, cost, precision and etc. We propose an improved genetic algorithm combined with simulation annealing algorithm, which is applied to selection of modules for the design of electromagnetic flow meter.

Experimental results show that the near-optimal value generated by improved GA is higher than that by standard GA, and the time of reasoning is shorter than the method of exhaustion and standard GA. Moreover, the improved genetic algorithm can overcome the premature convergence of genetic algorithms and guarantee the diversity of the design schemes of intelligent instrument.

REFERENCES

[1] Ericsson, A., Erixon, G., Controlling design variants: modular product platforms. New York: ASME. , 1999.

[2] Xie You-bai, "Some basic concepts in modern design theory", Journal of Mechanical Engineering.vol.43, NoV. 2007, pp.7-16

[3] M. Germani, F. Mandorli, M. Mengoni, R. Raffaeli, "An approach to support the implementation of product configuration tools", Proceedings of the ASME 2009 International Design Engineering Technical Conference & Computers and Information in Engineering Conference, San Diego, California, 2009.

[4] Zhang Ying, Yin Gang, "An intelligent instrument based on controller area network (CAN), "2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), Chengdu, China, pp.352-355, 2010.

[5] Roy, B.K., Santhosh, K.V, "An Intelligent Instrument for Measuring Liquid Level", 2011 International Conference on Process Automation, Control and Computing (PACC), Coimbatore, India, pp.1-5, 2011.

[6] Ding Hua, at el., "Design and Implementation of Modern Design Platform for Mechanical Products, " 2009 Second International Conference on Intelligent Computation Technology and Automation, pp.317-322, 2009.

[7] Yin Chenglong, Teng Youshui, Zhu Lin, "Research on expert system for conceptual design of sensor, "Journal of Electronic Measurement and Instrument, Vol.23 No.1, 2009, 85-90.

[8] Bing Li, Qiwei He, el al., "Development and Research of Virtual Design Platform of Steel Wheel Series Products, "2010 International Conference on Management and Service Science (MASS), pp.1-4, 2010.

[9] Schuh G., Arnoscht J., Rudolf S, "Integrated development of modular product platforms, "2010 Proceedings of Technology Management for Global Economic Growth (PICMET), pp.1-13, 2010.

[10] Andersson, J., Multi-objective optimization in engineering design, applications to fluid power systems, PhD thesis. Institute of Technology, Linko¨ pings Universitet. 2001.

[11] Goldberg, D.E., Genetic Algorithms in search, optimization & machine learning. Addison-Wesley Publishing Company, 1989.

[12] Jian Pan, Jinxiang Dong, Zhongdong Huang, "User-oriented product conceptual design of intelligent instrument based on genetic algorithm, "International Conference on Computer Application and System Modeling (ICCASM2010), Hangzhou, China, pp.578-581, 2010.

[13] Augusto O.B., Rabeau S. Multi-objective genetic algorithms: a way to improve the convergence rate, Engineering Application of Artificial Intelligence, Elsevier Science, 2005.

[14] Andersson J., Krus P. "Metamodel representations for robustness assessment in multiobjective optimisation, "proceedings of ICED 2001, Glasgow, UK, 2001.

[15] S. Bandyopadhyay, U. Maulik, and M. K. Pakhira, "Clustering using simulated annealing with probabilistic redistribution, "Int. J. Pattern Recogn. Artif. Intell, vol. 15, no. 2, pp. 269–285, 2001.

[16] U. Maulik, S. Bandyopadhyay, and J. Trinder, "SAFE: An efficient feature extraction technique, "J. Knowl. Inf. Syst., vol. 3, pp. 374–387, 2001.

[17] B. Suman, "Study of self-stopping PDMOSA and performance measure in multiobjective optimization, "Comput. Chem. Eng., vol. 29, no. 5, pp. 1131–1147, Apr. 15, 2005.

[18] K. Smith, R. Everson, and J. Fieldsend, "Dominance measures for multi-objective simulated annealing, "in Proc. 2004 IEEE Congr. Evol. Comput., 2004, pp. 23–30.

[19] B. Suman and P. Kumar, "A survey of simulated annealing as a tool for single and multiobjective optimization, "J. Oper. Res. Soc., vol. 57, no. 10, pp. 1143–1160, 2006.

[20] Sanghamitra B., at el., "A Simulated Annealing-Based Multiobjective Optimization Algorithm: AMOSA, "IEEE Trans. On Evolutionary Computation, Vol.12, No.3, 2008, 269-283.

[21] Quan Shan, Yan Chen, "Product modular platform design using SA_PSO, "2009 2nd IEEE International Conference on Computer Science and Information Technology (ICCSIT 2009), pp.176-179, 2009.

[22] Lanying Li, Yanbo Song, Ming Gao, "A New Genetic Simulated Annealing Algorithm for Hardware-Software Partitioning, "2010 2nd International Conference on Information Science and Engineering (ICISE2010), pp.1-4, 2010.

[23] Juanyan Fang, Zhihua Cui, Xingjuan Cai and Jianchao Zeng, "A Hybrid Group Search Optimizer With Metropolis Rule, "Proceedings of the 2010 International Conference on Modelling, Identification and Control, Okayama, Japan, pp.578-581 , 2010.

[24] Burcin Cakir, Fulya Altiparmak, Berna Dengiz, "Multi-objective optimization of a stochastic assembly line balancing: A hybrid simulated annealing algorithm, "Computers & Industrial Engineering, 60, Issue 3, 376-384, 2011.

[25] Lit, P. D., Falkenauer, E., & Delchambre, "A. Grouping genetic algorithms: an efficient method to solve the cell formation problem, "Mathematics and Computers in simulation, 51, 257–271, 2000.

[26] Rongbo Zhu, Jiangqing Wang, "Power-Efficient Spatial Reusable Channel Assignment Scheme in WLAN Mesh Networks," *Mobile Networks and Applications*, vol. 17, no. 1. pp. 1-11, 2012.

[27] Rongbo Zhu, "Intelligent Rate Control for Supporting Real-time Traffic in WLAN Mesh Networks," *Journal of Network and Computer Applications*, vol. 34, no. 5, pp. 1449-1458, 2011.

[28] Rongbo Zhu, Yingying Qin and Chin-Feng Lai, "Adaptive Packet Scheduling Scheme to Support Real-time Traffic in WLAN Mesh Networks," *KSII Transactions on Internet and Information Systems*, vol. 5, no. 9, pp. 1492-1512, 2011.

Jian Pan (corresponding author) is a candidate for Ph.D. degree in computer science from Zhejiang University and an associate professor with Zhejiang University of Technology, Hangzhou, China. He received B.S. degree in computer science from Zhejiang University of Technology, China, in 1998, M.S. degree in computer science from Zhejiang University, China, in 2004. His current research area includes embedded system design, CAD, AI and context-based computing. He has published more than 20 research papers in proceedings and journals. (Email: pj@zjut.edu.cn)

Guohong Mao is a lecturer with Zhejiang University of Technology, Hangzhou, China. She received B.S. degree in computer science from Zhejiang University of Technology, China, in 1999, M.S. degree in computer science from Zhejiang University, China, in 2005. Her current research area includes embedded system design, CAD and machine vision.

Jingxiang Dong is a professor with Zhejiang University, Hangzhou, China. He received B.S. degree in mathematics from Zhejiang University, China, in 1969. His current research area includes CAD, AI and database system. He has published more than 200 research papers in proceedings and journals.