

A peer-to-peer distributed collaborative optimization system¹

Caijun Xue^{a,*}, Hong Nie^a, Qingying Qiu^b and Peien Feng^b

^a*College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, P.R. China*

^b*State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, 310027, P.R. China*

Abstract. It is difficult to solve design optimization problems of complex systems by using a traditional computing method because complex simulation processes usually lead to large-scale computation. Therefore the distributed computing technology based on decomposition-coordination theory has received much attention by design engineers. This paper studies a peer-to-peer collaborative optimization method based on distributed computing technology in order to examine flexible optimization. A new distributed collaborative optimization framework is proposed, and a coordination method is developed and used to deal with the conflict of related variables among sub-optimization problems. A multi-agent based distributed computing environment is implemented. The implementation of an optimization agent, in which CORBA technology is used to implement communication between the components of the optimization agent, is discussed in detail. Two examples are used to demonstrate the efficiency of the computing method and the reliability and flexibility of the multi-agent system.

1. Introduction

Products development is in essence an optimization process. Although optimization techniques have been successfully used in many engineering fields with promising results, solving optimization problems of complex engineering systems is still a grand challenge that must be faced. The state-of-the-art optimization techniques have not yet meet the demand of modern engineering design problems due to large-scale computation and complex simulation, in particular the design of an entire system and/or an overall design process [1].

Large scale optimization problems may be tackled only if they are somehow decomposed. Decomposing a large optimization problem into several smaller sub-problems also allows many optimisation prob-

lems to be solved in parallel. Thus, efficiency is increased. The distributed computing technology based on decomposition-coordination theory is very promising and attractive. For our discussion, collaborative optimization technology means a systemic approach to the optimisation of complex and coupled engineering systems by decomposing large design problems into smaller design problems, which can be assigned to different groups of engineers. Here, collaborative optimization technology is, in some sense, very much the same as Multidisciplinary Design Optimization (MDO), but its application must not be confined to the multidisciplinary fields.

This paper studies a peer-to-peer collaborative optimization method based on distributed computing technology in order to understand flexible optimisation. A distributed computing framework is proposed and a multi-agent based distributed collaborative optimisation system is constructed. The paper is organized as follows. Section 2 introduces some related work in this research area. A distributed computing framework is presented and a coordination strategy among sub-optimization problems is derived in Section 3. System

*Present address: College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, P.R. China. E-mail: cjsxue@nuaa.edu.cn.

¹This paper is based on the presentation made at the International Symposium on Distributed Computing and Applications to Business, Engineering and Science, Wuxi, China, November 16–21, 2002.

implementation is discussed in section 4. In section 5, two examples are used to demonstrate the application of the framework. Finally, conclusions are drawn.

2. Related work

An intense research area in recent years is to provide engineering support and solutions to design optimizations of complex engineering systems. The requirements of MDO accelerate the development of collaborative optimization technology [2]. For hierarchic problems, a linear decomposition method was proposed in [3] and successfully applied to structural optimization [4]. Since most practical problems are non-hierarchic, the present research focuses on a collaborative optimization technique for non-hierarchic problems.

The method for hierarchic problems was improved and adapted to suit non-hierarchic problems [5]. Other methods for non-hierarchic problems include, the compatibility constrained optimization (CCO) method, the concurrent sub-space optimization (CSSO) method and the collaborative optimization (CO) method. The CCO method treats non-hierarchic problems as hierarchic problems after including design constraints and coupling constraints at the discipline level [6]. The CSSO method provides multidisciplinary feasibility at each CSSO cycle; it separates optimizations within the sub-systems and deals with all of the design variables simultaneously at the system level [7]. Bolebaum introduced an expert system technology into CSSO [8]. Renaud improved coordination methods and studied an approximation technology for CSSO [9,10]. The SYSOPT system, based on CSSO, has been successfully applied to a commercial aircraft design [11]. Sellar used response surface technology to increase the computing efficiency of the CSSO method [12]. The CO method preserved disciplinary groupings by allowing parallel development of design [13]. Braun studied CO method at length and applied CO method to aircraft design [14, 15]. Tappeta solved multi-objective optimization by using the CO method [16]. Parallel to the development of the above methodology, a number of software packages have been developed to facilitate integration of codes, data, and user interfaces, such as FIDO [17], iSIGHT [18].

3. A new distributed collaborative optimization framework

Although sub-problems have different levels of autonomy within a decomposed problem when applying the above methods [19], none of these methods is a peer-to-peer method. They are mainly focused on integrating developed modules, lack of flexibility and do not support product development process. This is due to the fact that a systemic optimization model is difficult to build and information of analytic gradients is difficult to obtain. The aim of this paper is to develop practical design optimization tools for engineers enabling them to systematically improve designs processes.

3.1. The collaborative optimisation framework

Products development is a collaborative process in which each design group deals with a sub-system. As a result, the sub-optimization models are independent to a certain extent. On the other hand, the correlation of the sub-systems results in the coupling relationship of the sub-optimization models. A good collaborative optimisation framework should meet the following demands:

- (1) It does not require modifications to any sub-optimization models because modifying any optimization model not only demands design engineers to grasp collaborative optimization algorithm, but also may lead to loss of technology security.
- (2) Sub-optimization models can be computed independently for some purpose. This is also an important way to validate sub-optimization models.
- (3) Sub-optimization program runs in parallel mode and the coordination operation can be finished intelligently under the framework.

Based on the above considerations this paper proposes the Collaborative Subspace Optimization Framework (CSOF) as depicted in Fig. 1. Sub-optimizations run in mutually independent sub-spaces and the coordination module deal with conflict of sub-optimizations.

The advantage of this framework is that sub-optimizations are mutually independent and, thus, may be run in parallel. One can freely experience the flexible optimization technology for the generalized optimal design under this framework [20]. Moreover, technology security of design groups is guaranteed since it is not necessary to modify sub-optimization models.

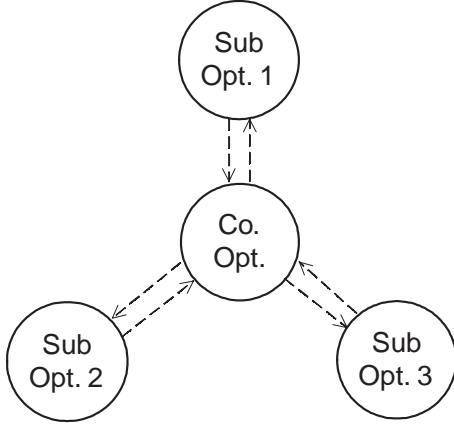


Fig. 1. Collaborative Subspace Optimization Framework (CSOF).

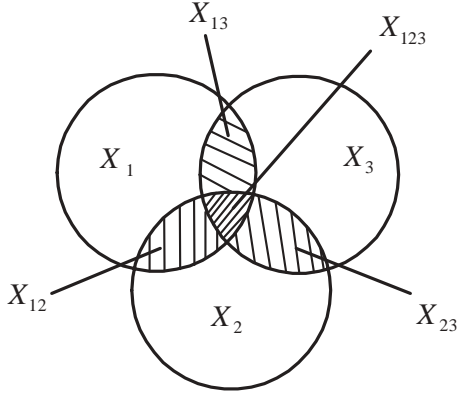


Fig. 2. Coupling relation of the related variables.

3.2. Coordination strategy

The complete autonomy of sub-optimization problems makes the conflict among them more serious, which must be resolved by a coordination strategy. A great challenge for CSOF is to develop an efficient coordination strategy in order to solve the conflict among sub-optimization problems.

3.2.1. Problem description

An optimization problem including three sub-optimizations is studied as an example. Since state variables can be treated as special design variables, only coupling design variables are considered. The design variables are classified into free variables and related variables. The coupling relation of the related variables is depicted in Fig. 2.

The variables of sub-optimization problems can be formulated as Eqs (1)–(3):

$$X_1 = \{X_{1f} \ X_{12} \ X_{13}\} \quad (1)$$

$$X_2 = \{X_{2f} \ X_{12} \ X_{23}\} \quad (2)$$

$$X_3 = \{X_{3f} \ X_{13} \ X_{23}\} \quad (3)$$

Where, X_{1f} , X_{2f} and X_{3f} are free variables vectors, and X_{12} , X_{13} and X_{23} are related variables vectors, And $X_{123} = X_{12} \cap X_{13} \cap X_{23}$. In general models of sub-optimization problems are formulated as Eqs (4)–(5):

$$\text{Min } F_i(X_i) \quad (4)$$

$$\text{s.t. } G_i(X_i) \leq 0 \quad (5)$$

3.2.2. The coordination model

When a sub-optimization puts forward a coordination proposal, the related sub-optimizations stop their optimization processes and the coordination process starts. The present values of the related variables for every sub-optimization can be formulated as Eqs (6)–(8):

$$X_1^0 = \{X_{1f} \ X_{12}^1 \ X_{13}^1\} \quad (6)$$

$$X_2^0 = \{X_{2f} \ X_{12}^3 \ X_{23}^2\} \quad (7)$$

$$X_3^0 = \{X_{3f} \ X_{13}^3 \ X_{23}^3\} \quad (8)$$

The present values of the objective functions are denoted as $F_1(X_1^0)$, $F_2(X_2^0)$ and $F_3(X_3^0)$. Since the purpose of the coordination of the related variables is to find a variable scheme that satisfies every sub-optimization problem, the essence of the coordination is an optimization problem itself and is given by Eqs (9)–(13):

$$\text{Min } (F_1(X_1) - F_1(X_1^0))^2 + F_2(X_2) - F_2(X_2^0))^2 + (F_3(X_3) - (F_3(X_3^0))^2) \quad (9)$$

$$\text{S.t. } X_{12}^1 = X_{12}^2 \quad (10)$$

$$X_{13}^1 = X_{13}^3 \quad (11)$$

$$X_{23}^2 = X_{23}^3 \quad (12)$$

$$G_i(X_i) \leq 0 \quad (i = 1, 2, 3) \quad (13)$$

The above coordination model includes all variables, all objectives functions and restrictions of all sub-optimization problems, and has unambiguous meaning and good integrity.

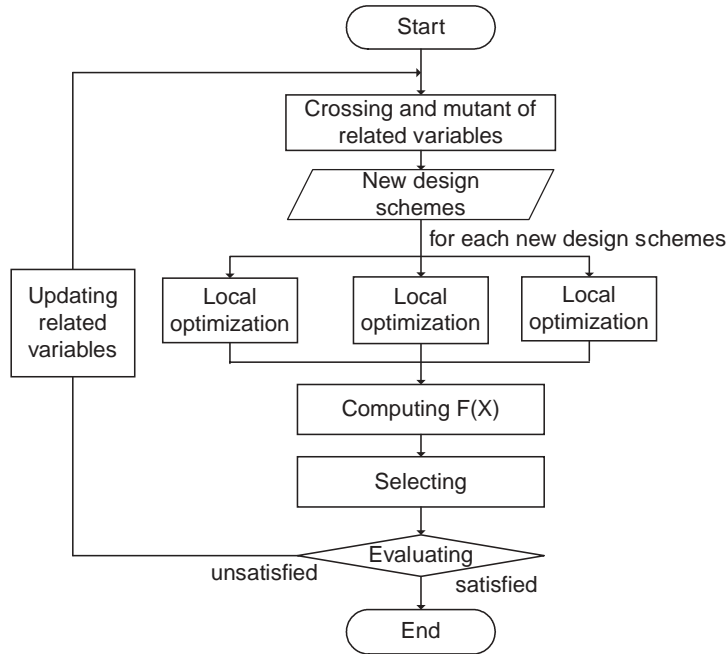


Fig. 3. Computational method of the coordination model.

3.2.3. A simplified computational method

It is nearly impossible to calculate directly the above coordination in spite of its unambiguous meaning and good integrity. To overcome the computational difficulty, it is necessary to study simplified computational methods of the coordination model. The simplified method simulates biological evolution processes to produce new design schemes by crossing and mutating of the related variables. Descendants differ from their ancestors by the use of natural selection based on valuating the objective functions of the coordination model. The above ideas come from genetic algorithm [21]. The theory of the simplified computational method is shown in Fig. 3, which includes three main steps.

The first step is to form new design schemes by crossing and/or mutating of the related variables. The crossing operation of the related variables is defined by,

$$X^I = X^1 \otimes X^2 \tag{14}$$

where \otimes is a crossing operator. If

$$X^1 = \{x_1^1 \ x_2^1 \ x_3^1 \ \dots \ x_n^1\} \tag{15}$$

$$X^2 = \{x_1^2 \ x_2^2 \ x_3^2 \ \dots \ x_n^2\} \tag{16}$$

The theory of crossing operations is demonstrated in Fig. 4. The crossing place can be selected at random.

The mutating operation of the related variables is defined by,

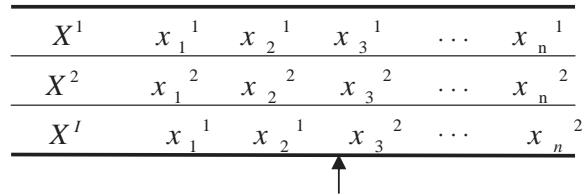


Fig. 4. Theory of crossing operations.

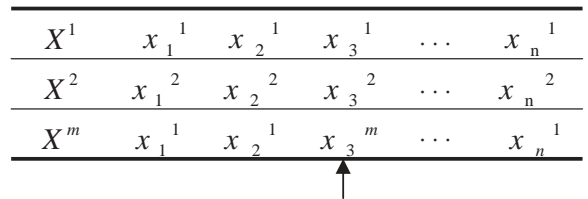


Fig. 5. Theory of mutating operations.

$$X^I = X^1 \odot X^2 \tag{17}$$

where \odot is a mutating operator and $x_3^m = (x_3^1 + x_3^2)/2$. The theory of mutating operations is demonstrated as in Fig. 5. The mutant place can be selected at random.

The second step is a local optimization process for each new scheme applied to the related variables coming from the first step, where the related variables do not participate in optimization. Every sub-optimization problem finishes the process in order to obtain local optimal solutions.

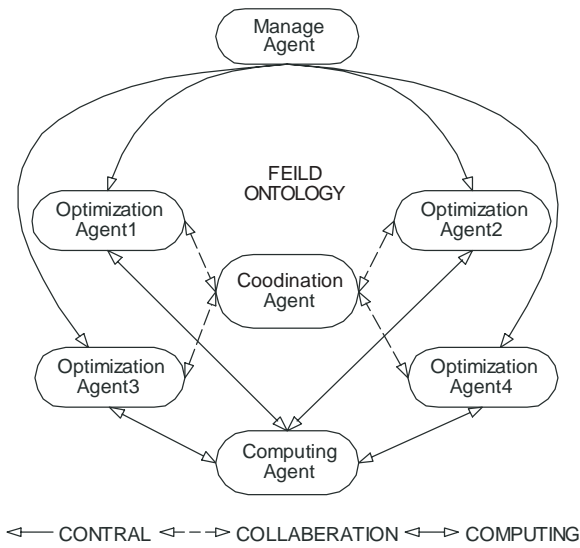


Fig. 6. The architecture of a multi-agent based collaborative optimization system.

Finally, the objective functions of the coordination model are calculated and the optimal solution is selected. The convergence condition is defined as $\varepsilon = (Obj_k - Obj_{k-1}) / Obj_k$. When $\varepsilon \leq \varepsilon_0$, the coordination process stops and the sub-optimization problems restart based on the coordination solution. When $\varepsilon > \varepsilon_0$, the solution with its collaborative objective function being smallest is selected to produce new design schemes by crossing and/or mutating.

The above process is a loop process until the criterion $\varepsilon \leq \varepsilon_0$ is satisfied. It is clear that all sub-optimization problems obtained the shared value preference by crossing or/and mutating of the related design variables. So, the above coordination process is an evolution process for obtaining a satisfactory solution of all sub-optimization problems.

4. A multi-agent based collaborative optimization system

Multi-agent technology requires the coordination in group with intelligence in some degree, and is an imitation to human society on rational level. The notion of heterogeneous autonomous agents collaborating to solve problems is a powerful metaphor for the engineering of distributed and interoperable software systems. Using multi-agent technology for distributed collaborative optimization system is motivated by its following advantages: distributed architecture, autonomy, sharable [22]. Distributed architecture makes it

convenient to solve large-scale optimization problem, autonomy suits peer-to-peer optimization, and sharable promises to collaborate.

4.1. System architecture

As a peer-to-peer computing circumstance, distributed collaborative optimization system must meet the following requirements: easy linking of CAE tools, support for geographically distributed optimization, support for database management, extensible for optimization solvers and ability to perform trade-off between different design responses to assist exploration of Pareto solutions.

A multi-agent architecture for collaborative optimization is designed as shown in Fig. 6, including a management agent, a coordination agent, a computing agent, several optimization agents and some utility agents (not shown in the figure). The whole system works together and is based on the field ontology. The management agent facilitates exploring optimization models, planning optimization scheme, changing coordination parameters, observing coordination activities and so on. The coordination agent is responsible for coordinating related variables among optimization agents according to the coordination strategy discussed in section 3. The computing agent responds to computing requests (for example, FEA) coming from an optimization agent, and allocates computing tasks to a suitable computer on the INTERNET according to the reliable computing resources.

4.2. Field ontology creation

In the area of artificial intelligence, ontology belongs to the theory of content, studies the object classification, object properties and relationship among objects on special field knowledge, and provides terminology for description of field knowledge [23]. Collaborative optimization system gains strong power for its internal parallelity, flexibility and robustness based on the distributed technology. However, an agent is absorbed in the task on the field of itself, which cause the difficulty in cooperation and knowledge sharing. Hence, the methodology of ontology is introduced to set up the conceptual schema of collaborative optimization. Based on ontology principle, the conceptual schema of collaborative optimization is build up. Abstract gradation structure method is introduced to abstract collaborative optimization to be objects as Fig. 7.

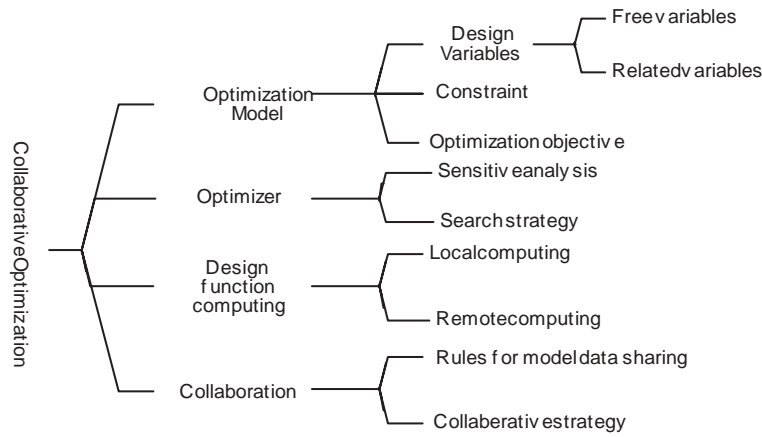


Fig. 7. Objective sort tree for collaborative optimization.

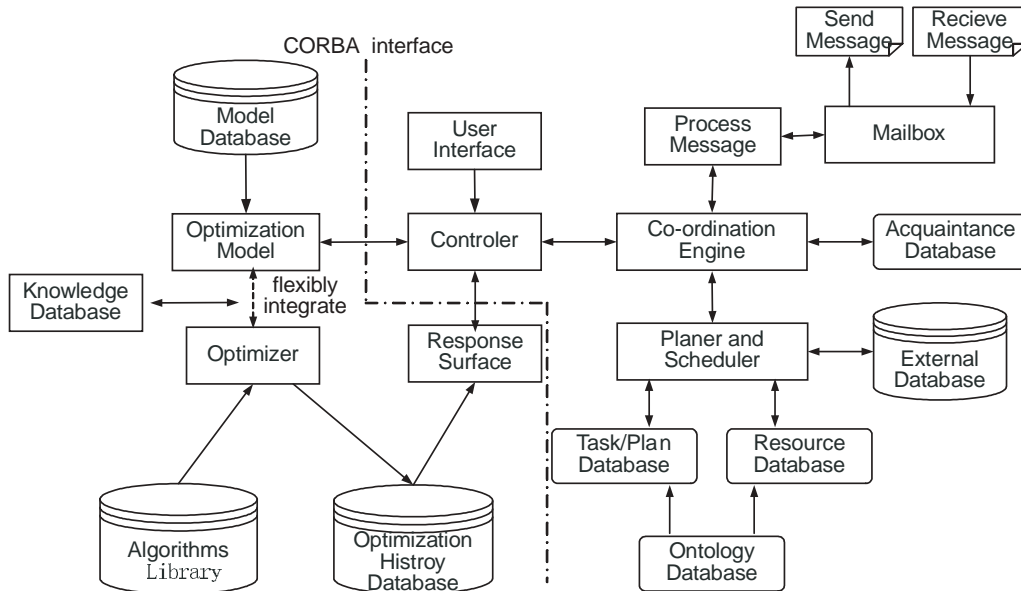


Fig. 8. Model of optimization agent.

4.3. Agent generation

A toolkit for constructing collaborative multi-agent applications named as ZEUS is used to facilitate developing the distributed collaborative optimization system. ZEUS is a culmination of a careful synthesis of established agent technologies to provide an integrated environment for the rapid development of multi-agent systems [24]. After ontology is created using ontology editor according to field ontology discussed in section 4.2, the generic agents are configured to fulfil its application-specific responsibilities during the agent creation stage. These agents' responsibilities are listed in Table 1.

4.4. Implementing an optimization agent

An optimization agent should be an integration of an optimization program, an optimization model and an agent. Moreover, in order to increase flexibility, an optimization model and an optimization program should implement flexible integration based on the characters of the optimization model. So, an optimization agent is designed as Fig. 8 shows.

Because the optimization program is written in C++ language and the agent is implemented in java language in our work, CORBA technology is used here to realize the communication between them. CORBA is a

Table 1
Agents' responsibilities

No.	Agent	Task	
		Task name	Describe
1	Optimization agents	Optimization	Running sub-optimization problems
		SendCoDemand	Sending coordination demands
		SendFeaDemand	Sending FEA computing demands
		ReportOptResult	Reporting sub-optimization results
2	Coordination agent	ResponseCoDemand	Starting coordination process
		Coordination	Coordinating the related variables
		ReportCoResult	Reporting coordinating results
2	Computing agent	PlanFeaTask	Planning FEA computing scheme
		AssignFeaTask	Assigning FEA computing tasks
		ReportFeaResult	Reporting FEA computing results

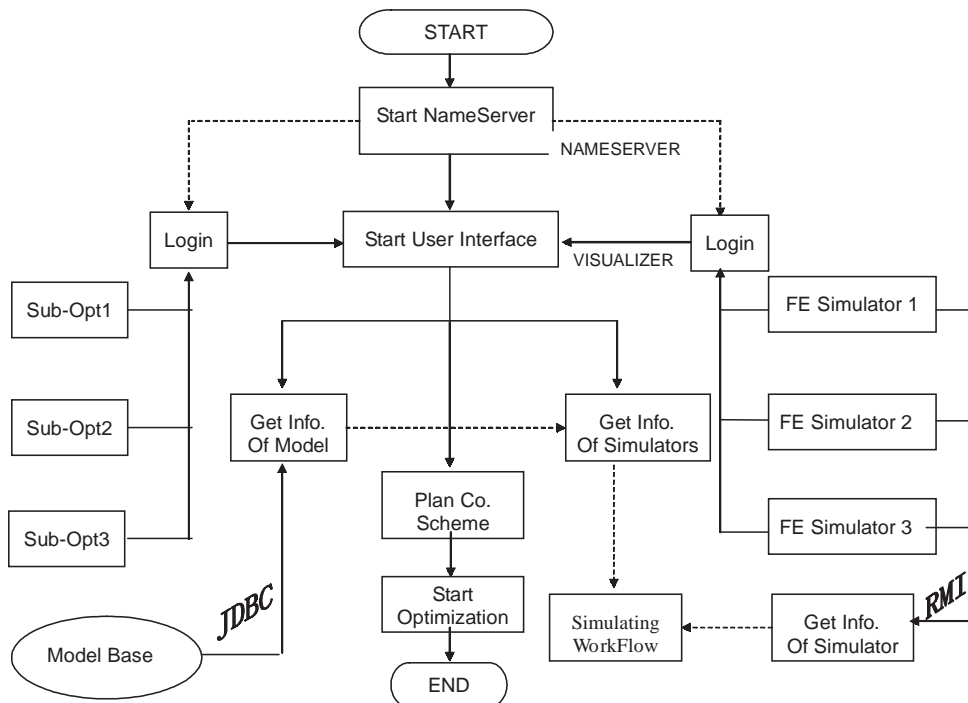


Fig. 9. Flowchart of the multi-agent system optimization system.

transit structure, which can let multi-component communicate with each other on the network. Its characteristics include access transparency, network and location transparency, programming language independence, and hardware and operating system independence. The interfaces defined in the object of CORBA are listed in Table 2.

4.5. System application

The multi-agent based collaborative optimization system, written in the Java programming language, is running on the network and the running flowchart is

Table 2
Interfaces defined in the object of CORBA

No.	Interface name	function
1	JSetModelName	Setting optimization model to optimizer
2	JStartOpt	Starting an optimization process
3	JPauseOpt	Pausing an optimization process
4	JStopOpt	Stopping an optimization process
5	JGetVariables	Getting the values of design variables
6	JGetFunctions	Getting the values of functions
7	JGetOptState	Getting optimization state
8	JQueryName	Getting the used algorithm name

demonstrated as Fig. 9. First, the NameServer agent is started which facilitates information discovery. Then

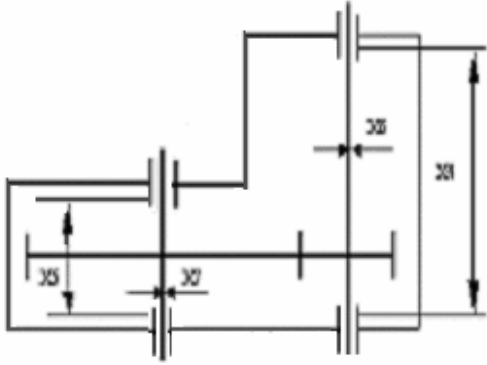


Fig. 10. Optimal design of a gear reducer.

the user interface agent is started which allows user to control the while system. When the NameServer agent and user interface agent have been started, the remote optimization agents and computing agents can start and join the system. After all agents have started, the user can browse or modify the related optimization models by JDBC programming, and can get information of computing resource by java RMI programming. A collaborative optimization scheme is planned based on above information of the models and computing resource. Finally, the collaborative optimization process can be started.

5. Optimization examples

5.1. Design optimization of a gear reducer

A gear reducer as shown in Fig. 10 is considered. The design objective is to minimize the weight of the gear reducer under 11 constraints (see the sub-optimization models). The meanings and the bounds of the design variables are:

- x_1 tooth width, $2.6 = x_1 = 3.6$,
- x_2 module of gear, $0.7 = x_2 = 0.8$,
- x_3 number of teeth of the smaller gear, $17 = x_3 = 28$,
- x_4 distance of the first shaft bearing, $7.3 = x_4 = 8.3$,
- x_5 distance of the second shaft bearing, $7.3 = x_5 = 8.3$,
- x_6 diameter of the first shaft, $2.9 = x_6 = 3.9$,
- x_7 diameter of the second shaft, $5.0 = x_7 = 5.5$.

The solution obtained by the integral optimization is listed as Table 3.

Table 3
Solution obtained by the integral optimization

X1	X2	X3	X4	X5	X6	X7	Obj. f
3.50	0.70	17.00	7.32	7.75	3.36	5.29	2997.9

For collaborative optimization, the original problem is decomposed into three sub-problems as follows.

Problem 1:

$$X^1 = \{x_1 \ x_2 \ x_3\} \quad (18)$$

$$\text{Min } f_1 = 0.7854x_1x_2^2(3.333x_3^2 + 14.933x_3 - 43.0934) \quad (19)$$

$$S.t. \ g_1 = 27/(x_1x_2^2x_3) - 1 \leq 0 \quad (20)$$

$$g_2 = 397.5/(x_1x_2^2x_3^2) - 1 \leq 0 \quad (21)$$

$$g_7 = x_2x_3/4 - 1 \leq 0 \quad (22)$$

$$g_8 = x_1/x_2 - 12 \leq 0 \quad (23)$$

$$g_9 = 5 - x_1/x_2 \leq 0 \quad (24)$$

Problem 2:

$$X^2 = \{x_1 \ x_2 \ x_3 \ x_4 \ x_6\} \quad (25)$$

$$\text{Min } f_2 = -1.508x_1x_6^2 + 7.477x_6^3 + 0.7854x_4x_6^2 \quad (26)$$

$$S.t. \ g_3 = 1.93x_4^3/(x_2x_3x_6^4) - 1 \leq 0 \quad (27)$$

$$g_5 = \sqrt{745^2x_4^2/x_2^2x_3^2 + 16.9 \times 10^6} / 110x_6^3 - 1 \leq 0 \quad (28)$$

$$g_{10} = (1.5x_6 + 1.9)/x_4 - 1 \leq 0 \quad (29)$$

Problem 3:

$$X^3 = \{x_1 \ x_2 \ x_3 \ x_5 \ x_7\} \quad (30)$$

$$\text{Min } f_3 = -1.508x_1x_7^3 + 7.477x_7^3 + 0.7854x_5x_7^2 \quad (31)$$

$$S.t. \ g_4 = 1.93x_5^3/(x_2x_3x_7^4) - 1 \leq 0 \quad (32)$$

$$g_6 = \sqrt{745^2x_5^2/x_2^2x_3^2 + 157.5 \times 10^6} / 85x_7^3 - 1 \leq 0 \quad (33)$$

$$g_{11} = (1.1x_7 + 1.9)/x_5 - 1 \leq 0 \quad (34)$$

The optimization result obtained by using the multi-agent based collaborative optimization system is listed in Table 4. The result is very close to the result obtained by using the integral optimization. Figure 11 shows the histogram of the related variables and the coordination objective.

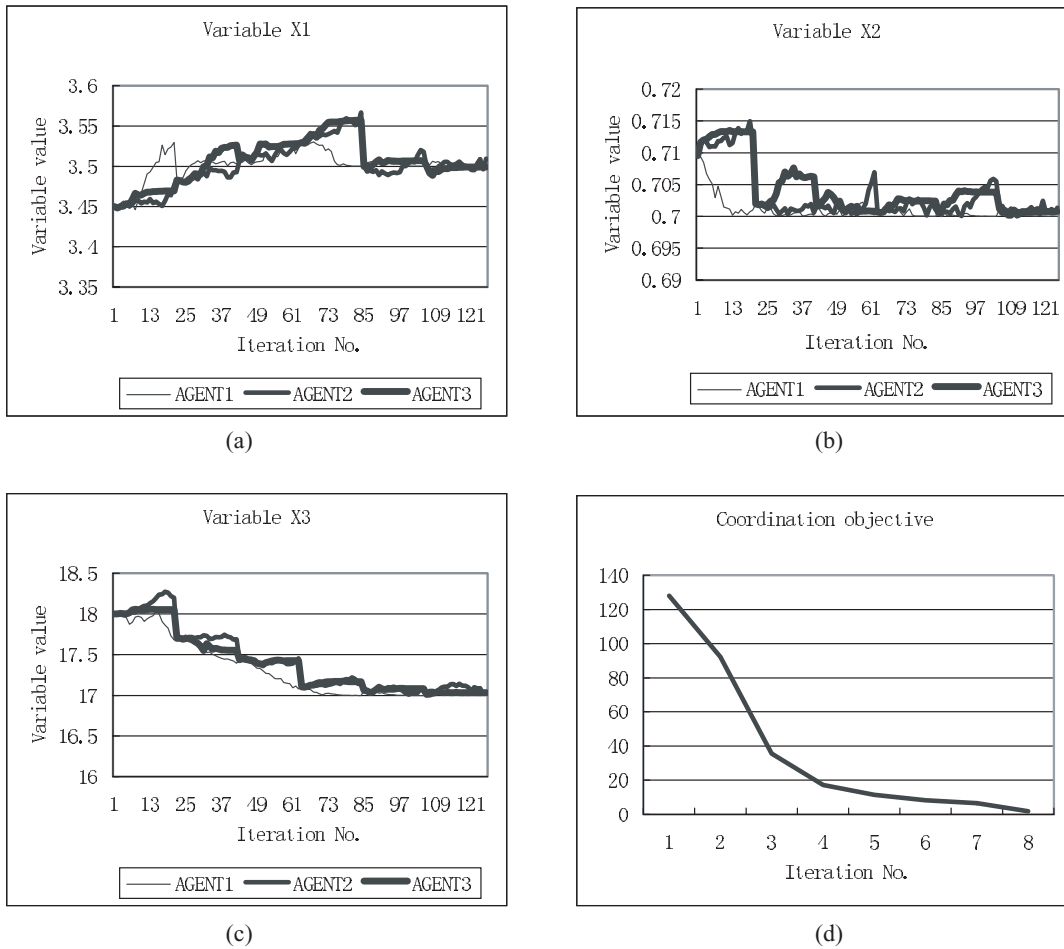


Fig. 11. Histogram of related variables and coordination objective.

5.2. Design optimization of hydraulic excavator working equipment

The structures of the hydraulic excavator (Fig. 12) tend to be larger and heavier than the current type. This causes increasing complex design requirements, such as greater digging range, lower vibration and noise, and higher durability, etc. The design process needs the coordination of engineers with different disciplinary knowledge. Here, two sub-optimization problems, the working equipment sub-optimization problem and boom sub-optimization problem, are considered. The information of sub-optimization models is listed in Table 5. There are twelve coupling variables between the two sub-optimization problems.

Table 6 shows the results of collaborative optimization for a hydraulic excavator working equipment. From this table we can find that the first natural frequency, maximize digging force by arm, maximize dig-

ging force by bucket, maximize digging depth, maximize digging radius and maximize digging height raise. The weight of the boom reduces. All of these results are expected.

6. Conclusions

The decomposition-coordination method has been used in complex engineering optimization for several decades. However, the successful application is limited due to lack of a flexible collaborative optimization system. This paper proposes a multi-agent based collaborative optimization system, in which a large optimization problem is decomposed into several smaller optimization problems. This would allow many optimization agents working in parallel for the sub-problems. Two engineering optimization examples are implemented by using this system and some promising results are obtained.

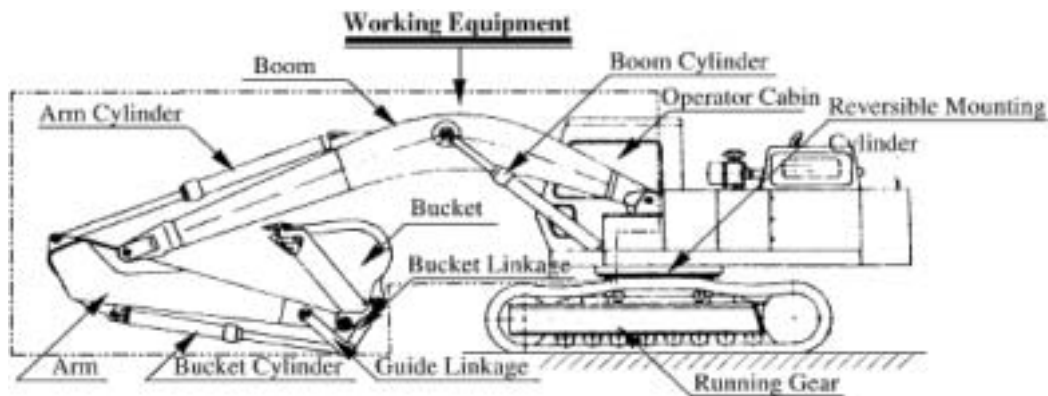


Fig. 12. A general view of a hydraulic excavator.

Table 4
Result obtained by collaborative optimization

X1	X2	X3	X4	X5	X6	X7	f1	f2	f3	f
3.50	0.70	17.00	7.32	7.97	3.35	5.29	1580.35	286.28	1131.45	2998.08

Table 5
The information of sub-optimization models

No.	Sub-optimization	Number of variables	Number of objectives	Number of constrains
1	working equipment	43	6	105
2	boom	35	1	19

Table 6
The results of collaborative optimization

Sub-optimization	Objective name	Initial solution	Final solution
Working equipment	First natural frequency Hz	8.45	9.13
	Max. digging force by arm (kN)	16.62	16.72
	Max. digging force by bucket (kN)	24.93	26.35
	Max. digging depth (m)	3.42	3.52
	Max. digging radius (m)	5.30	5.67
	Max. digging height (m)	4.38	4.74
Boom	Weight (T)	0.126	0.113

Acknowledgements

This work is supported by the National 863 Program of China under Grant Number No. 2001AA412112. The authors wish to thank the enormous effort of the referees and the guest editor for their useful comments and suggestions.

References

- [1] Feng Peien, Qiu Qingying and Pan Shuangxia, et al., Research on generalized optimization process for mechanical product, *Science in China (Series E)* **42**(4) (1999), 407–417.
- [2] Yu Xiongqing, Ding Yunliang, Multidisciplinary design optimization—a survey of its algorithms and application to aircraft design, *Acta Aeronautica ET Astronautica Sinica* **21**(1) (2000), 1–6.
- [3] Sobieszcanski-Sobieski J. A linear decomposition method for large optimization problems—blueprint for development. NASA TM-83248, 1982.
- [4] J. Sobieszcanski-Sobieski, B.B. James and A.R. Dovi, Structural Optimization by Multilevel Decomposition, *AIAA Journal* **23**(11) (November 1985), 1775–1782.
- [5] J. Sobieszcanski-Sobieski, Two alternative ways for solving the coordination problem in multilevel optimization, *Structural Optimization* **6** (1993), 205–215.
- [6] R. Balling and J.J. Sobieszcanski-Sobieski, An algorithm for solving the system-level problem in multilevel optimization, *Structural Optimization* **9** (1995), 168–177.
- [7] Sobieszcanski-Sobieski J. Optimization, by Decomposition: A Step from Hierarchic to Non-Hierarchic Systems, Proceedings, 2nd NASA/USAF Symposium on Recent Advances in Multidisciplinary Analysis and Optimization, Hampton, Virginia, 1988. NASA CP-3031.

- [8] C.L. Bolebaum, Formal and heuristic system decomposition in structural optimization. NASA-CR-4413, Virginia: NASA, 1991.
- [9] J.E. Renaud and G.A. Gabriele, A improved coordination in non-hierarchical system optimization, *AIAA Journal* **31** (1993), 2367–2373.
- [10] J.E. Renaud and G. Gabriele, A. Approximation in non-hierarchical system optimization, *AIAA Journal* **32** (1994), 198–205.
- [11] E.D. Eason, G.A. Nystrom and A. Burlingham, et al., Non-hierarchical multidisciplinary design of a commercial aircraft. AIAA-94-4302, Panama City Beach, FL: AIAA, 1994.
- [12] R.S. Sellar, S.M. Batill and J.E. Renaud, Response surface based concurrent subspace optimization for multidisciplinary system design. AIAA-96-0714, Nevada: AIAA, 1996.
- [13] I. Kroo, S. Altus and R. Braun, et al., Multidisciplinary optimization methods for aircraft preliminary design. AIAA-94-4325, FL: AIAA, 1994.
- [14] R.D. Braun, Development and application of the collaborative optimization architecture in a multidisciplinary design environment. NASA, International Congress on Industrial and Applied Mathematics, Hamburg, Germany, 1995.
- [15] R.D. Braun, Collaborative Optimization: An architecture for large-scale distributed design. Ph.D. Thesis, Stanford University, 1996.
- [16] R.V. Tappeta, Multiobjective collaborative optimization, *ASME J. Mech. Des.* **119**(3) (1997), 403–411.
- [17] R.P. Townsend, J.C. Weston and T.M. Eidsen, A programming environment for distributed complex computing. An overview of the framework for interdisciplinary design optimization (FIDO) project. Technical report, NASA TM 109058, 1993.
- [18] URL: <http://www.engineous.com/index.htm>.
- [19] R. Balling and M.R. Rawlings, Collaborative optimization with disciplinary conceptual design. *Journal of the International Society for Structural and Multidisciplinary Optimization (ISSMO)* **20**(3) (2000), 232–241.
- [20] Qiu Qingying, Feng Peien and Wu Jianwei, et al., Product model structure for generalized optimal design, *Progress in Natural Science* **2**(4) (2002), 294–300.
- [21] Chen Guoliang, Wang Xunfa and Zhuang Zhenque, et al., *Genetic algorithms and its applications*, Post & press, China, 1996.
- [22] M. Wooldridge and N.R. Jennings, Agent theories, architectures, and languages: a survey, in: *Intelligent Agents*, Wooldridge and Jennings, eds, Berlin: Springer-Verlag, 1995, pp. 1–22.
- [23] B. Chandrasekaran, J.R. Josephson and V.R. Benjamins, What are ontologies, and why do we need them? *IEEE Intelligent Systems & Their Applications* **14**(1) (1999), 20–26.
- [24] URL: <http://more.btexact.com/projects/agents/zeus/index.htm>.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

