Struct Multidisc Optim 27, 460–468 (2004)
DOI 10.1007/s00158-004-0407-y

A comparative study of software systems from the optimization viewpoint

U.P. Hong, K.H. Hwang and G.J. Park

Abstract Analysis technology is widely accepted and quite popular these days. Incorporation of the analysis result into a design process is a key factor for the success of an analysis area. A few design software products have been commercialized. Generally, they are trying to make an interface between various design methods and analysis software. Optimization is a typical automatic design method. The software products of optimization are investigated and compared for user convenience and algorithm performance. A few popular products are selected. A graphical user interface (GUI) is compared for capability and efficiency. The performances of the optimization algorithms are tested by mathematical and engineering examples, and the results are discussed.

Key words design software, design environment, optimization software

1 Introduction

As the performance of analysis software systems improve and stabilize, they are becoming more widely used in many engineering disciplines. Detailed design for products is one of the most popular applications of the analysis software. Conventional design has only been performed with analysis results of the analysis software. The conventional design process depends on the designer's intuition, experience, skill, etc. This human element can sometimes lead to dangerous and erroneous results in the synthesis of complex systems (Arora 1989). These shortcomings can be overcome by well-established design theories such as optimization.

Some analysis software systems have their own design modules that have functions for systematic design. Sometimes, a designer may want to use analysis software that does not have a design module. In this case, the designer must develop a communication code that has a function for exchange data between the analysis software and design software. Even though the analysis software has design modules, the designer needs to develop a communication code if s/he wants to use sophisticated design methods such as multidisciplinary design optimization (MDO). MDO is a design methodology for complex systems that is composed of many disciplines (Sobieszcznski-Sobieski and Haftka 1996). A designer can make a design module for the complex systems. However, it is extremely difficult and costly.

Design software systems have been made to overcome these difficulties and inefficiencies. They provide design capabilities for the designer to control the analysis software systems. Design software systems have functions of design methods such as optimization, design of experiment (DOE), and so forth. They furnish design environment by integrating design algorithms and analysis software systems. Design environment consists of a set of necessary information in the design process. For example, design data, analysis software system, and input/output files are defined. By definition, designers can work without the development of interface codes. Various commercial codes have been developed.

In this research, the capabilities of typical design software systems are investigated. First, the systems must be selected. There are many available systems from software industries, research centers, universities, etc. Two criteria are adopted for the selection: (1) The system should be a commercialized one so that an engineer can use it upon purchase; and (2) The system should be available in Korea where this research has been conducted. Four systems satisfied both criteria and they are VisualDOC (Vanderplaats 1998b), iSIGHT (Engineous 1999), OPTIMUS (LMS 1999), and ModelCenter (Phoenix 1999). Since they are general-purpose and popular in

Received: 8 January 2002
Published online: 13 May 2004

U.P. Hong[1], K.H. Hwang[2] and G.J. Park[2,✉]

[1] Mando Co., Development 2 Team, 5-22 Bangye-ri, Munmak-eup, Wonju-si, Gangwon-do 220-805, Korea
e-mail: P13217@mando.com
[2] Department of Mechanical Design and Production Engineering, Hanyang University, 1271 Sa 1-dong, Sangnok-gu, Ansan-si, Gyeonggi-do 425-791, Korea
e-mail: khhwang@ihanyang.ac.kr, gjpark@hanyang.ac.kr

the worldwide optimization community, this research can provide general insight.

A comparative study is performed on the characteristics of the design environment control, performances of optimization and DOE modules. At first, a designer must specify design information via a design environment control. This information is utilized throughout the designing process. The characteristics of the design environment control are compared. Design software systems commonly have optimization and DOE modules. The optimization result depends on the performance of the optimization algorithm. The number of function calculations is dominant in the evaluation of the efficiency of an optimization. Various problems are selected from standard structural optimization examples. A practical problem is solved for an optimization with occupant analysis.

2
Basic directions of the comparison

Design software systems are constituted with a design environment control, design process, and output-monitoring as illustrated in Fig. 1. The design environment control and the design process are the most important in the design software systems.

At first, comparisons are carried out for the design environment control. The design environment control has functions that execute analysis software systems and exchange design data. Data for design variables, constraints, and cost functions are exchanged between design software and analysis software. Design data and analysis software systems must be specified in the design software. First of all, design data are specified from input/output files of analysis software. This might be inefficient, but it is quite flexible regardless of the existence of the source codes for the analysis software (Ghosh *et al.* 1999, 2000). Design software can access and control design data with specified location of the data in input/output files of the analysis software. If a wrong location is specified, the design software makes wrong design results, or does not perform the design because of input errors. This communication technique is called file wrapping. Different design software has different characteristics, and their characteristics of the design environment control are compared.

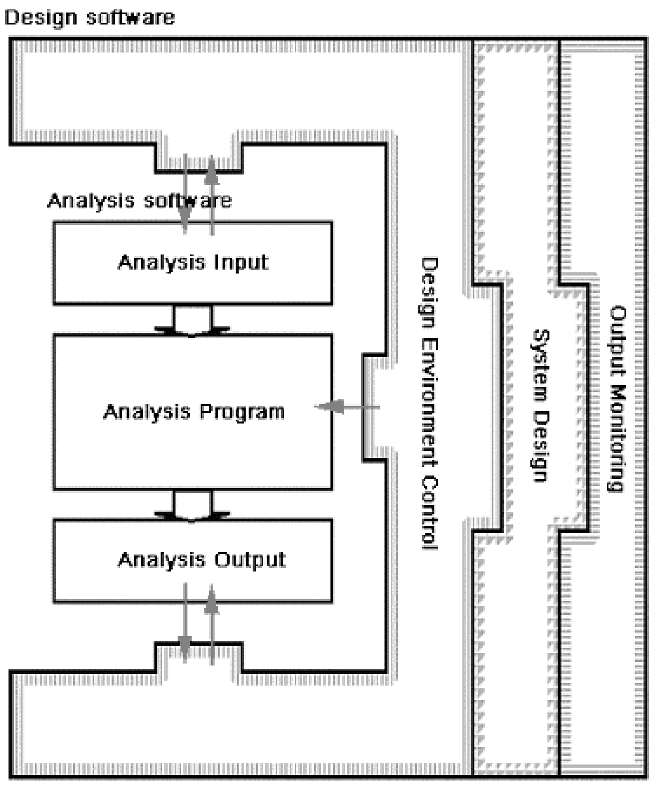

Fig. 1 Functions of the design software systems

Table 1 Software release

Software	Version	Developer
VisualDOC	1.2	VR&D
iSIGHT	5.0	Engineous
OPTIMUS	2.2	LMS International
ModelCenter	2.0	Phoenix Integration

Secondly, comparisons are carried out for the performances of design methods, especially for optimization algorithms. After the design environment is fully defined, the design software performs design with design modules. The software usually has well-established design algorithms, such as optimization and DOE. Design methods vary, but the software systems commonly have optimization and a DOE module. Optimization is an iterative method that updates design points using mathematical algorithms and finds a better design. The optimization algorithms are directly related to the accuracy of the results and efficiency of the design process. Optimization algorithms are compared by some mathematical and engineering problems. The characteristics and strong/weak points of the DOE are also compared.

When the design is finished, the results must be displayed by text or in graphical format. Although it is important to users, it does not affect the design performances. Therefore, the characteristics of output monitoring are not compared, because it is not easy to keep objectiveness and fairness in this comparison. Table 1 shows the specifications of the systems. New versions might be released, but the versions in Table 1 are those at the time of this research.

3
Basic directions of the comparison

The design data must be defined. Design software commonly provides many functions to define the design environment. Aspects of the definitions are compared.

3.1
Control functions for analysis software

At first, comparisons are conducted for the functions that control analysis software. Analysis modes are defined in the input files of an analysis software system and design modes are defined in the output files. The control is defined in the design environmental control in Fig. 1 by changing the data in the input file. All the design software systems commonly have graphical user interface (GUI) and file parsing functions, but they have different characteristics. The characteristics are briefly summarized in Table 2.

VisualDOC provides a source code that contains analysis software control functions and design data exchange functions. The code is written in user-friendly languages, such as FORTRAN or C. It may be difficult for some designers, because the code must be compiled before it is used. If the designer can use the languages, s/he can change the codes to adapt to their purposes. Therefore, it can be quite flexible for experts. In this research, the code is modified for specified design problems. There is no compiling process in iSIGHT, OPTIMUS, and ModelCenter. They communicate with analysis software through their internal process automatically. Therefore, easiness is enhanced but flexibility is deteriorated.

iSIGHT and OPTIMUS have their own script languages. They are a set of commands and furnish more flexibility for complex problems. The script language of iSIGHT, MDOL provides many useful functions. OPTIMUS provides a relatively simple script language.

ModelCenter has a module called an analysis server to control analysis software. The analysis server operates in any operating system (OS) because the software was developed with the JAVA language (Phoenix 1999). Designers can specify a design problem in the ModelCenter with a specified analysis module. It is built on an analysis server in a network environment. Therefore, it can be defined on any computer throughout a network. When analysis software systems exist in many computers with various operating systems, this capability can be very helpful.

Table 2 Characteristics of selected software systems to control the analysis program

No.	VisualDOC	iSIGHT	OPTIMUS	ModelCenter
1	○	○	○	○
2	×	○	○	○
3	○	○	○	○
4	×	○	○	×
5	○	×	×	×
6	×	○	×	○
7	×	○	○	○

1: GUI to define design problem
2: Multiple input/output files
3: Definition of user supplied equation
4: Script to define user defined problem
5: Compile process to integrate an analysis software
6: Real time output monitoring
7: Remote access to the analysis software

3.2
Functions specifying design data

Design software systems exchange design data via input/output files using a file wrapping technique. Design data are managed directly in this process. Most input files of analysis software have fixed formats. However, the formats of output files of analysis software vary in many cases. Therefore, special attention is needed in handling output files. These characteristics are common to all the design software systems.

There are two methods to exchange design data. One uses a template file that specifies data with special characters. It is very effective when the format of input/output files are fixed. The other uses locations of data on input/output files. In this method, design software systems exchange data using the row and column positions of the data from a special character or word that is specified as the origin. It can be used when the format of input/output files are changed during the design process. The characteristics to exchange data are compared in Table 3 for the fixed format input files.

iSIGHT has many strong capabilities for data exchange compared to other systems. It retains many useful functions. For example, it has a function that uses row and column information of data and specifies formats of data.

OPTIMUS uses a template to specify data of input files and uses locations of data to specify data of output files. Usually, input file formats are not changed for a specific design problem. Thus, it is one of the effective methods to exchange design data with input files. However, OPTIMUS has some problems when two or more design data exist in one line. Some improvements are needed for this problem. ModelCenter also uses locations of data. It also has a problem when two or more data exist in one line.

All systems except VisualDOC use relative locations of data. They exchange data with the row position of the

Table 3 File parsing characteristics of input files with fixed format

No.	VisualDOC	iSIGHT	OPTIMUS	ModelCenter
1	×	○	○	○
2	○	○	×	×
3	○	○	×	○
4	○	○	×	×

1: Relative position handling in files
2: Column position handling in files
3: Format defining for the variables
4: Multiple variable handling in a line

data from a special character or word that is specified as the origin. This method makes it possible to exchange design data in changing output files. VisualDOC should improve this capability. If the locations of data in output files change drastically, no system can trace them. This should be improved later.

4 Comparative study for performances of design modules

After the design environment is fully specified, the design is performed. The main purpose of design software is effectiveness of design capability. Each system provides some different design modules, but they commonly provide optimization and DOE modules. Design modules are represented in Table 4. In this research, performances of optimization and DOE modules are compared. Default settings are utilized in solving problems.

4.1 Comparison of optimization algorithms

Optimization is one design method that find a better design through an iterative process. Various algorithms have been developed (Haftka 1991; Haug 1979; Rao 1996; Bradley 1977; Fletcher 1970; Vanderplaats and Sugimoto 1985; Zoutendijk 1960; Vanderplaats 1984b; Fletcher *et al.* 1964; Thanedar *et al.* 1987). Generally, commercial software systems have algorithms that are known to be excellent. Supplied algorithms are shown in Table 5.

VisualDOC and ModelCenter have separate algorithms for the constrained problem and the unconstrained problem. The designer can select an algorithm from the category for each case. iSIGHT and OPTIMUS have no difference between constrained and unconstrained algorithms. The designers must select one suitable algorithm with their knowledge. iSIGHT provides advisor functions for the choice.

Function calculations are performed repeatedly during the optimization process. The number of the calculations can be a measure for efficiency of an algorithm since a function calculation needs much computational time for a large-scale problem. Thus, the number of function calculations is compared.

Table 4 Design methods in software systems

No.	VisualDOC	iSIGHT	OPTIMUS	ModelCenter
1	○	○	○	○
2	○	○	○	○
3	○	○	○	○
4	×	○	×	×

1: Optimization
2: Design of experiments
3: Parameter study
4: Quality engineering

Table 5 Optimization algorithms of design software systems

No.	VisualDOC	iSIGHT	OPTIMUS	ModelCenter
1	BFGS	MFD	SQP	VMM
2	FR	MMFD	GRG	CGM
3	MMFD	SLP		MFD
4	SLP	SQP		SLP
5	SQP	SAM		SQP
6		DHS		
7		EP		
8		GA		
9		HJ		
10		SA		
11		GRG		

BFGS: Broyden–Fletcher–Goldfarb–Shannon algorithm
CGM: Conjugate gradient method
DHS: Direct heuristic search
EP: Exterior penalty
FR: Fletcher–Reeves algorithm
GRG: Generalized reduced gradient
GA: Genetic algorithm
HJ: Hooke–Jeeves algorithm
MFD: Method of feasible directions
MMFD: Modified method of feasible directions
SLP: Sequential linear programming
SQP: Sequential quadratic programming
SA: Simulated aAnnealing
SAM: Successive approximation method
VMM: Variable metric method

At first, mathematical examples are solved. Equations (1) and (2) are the formulations of the examples. Equation (1) is an unconstrained problem and (2) is a constrained problem. VisualDOC provides BFGS and FR algorithms, and ModelCenter provides VMM and CGM algorithms for unconstrained problems. These algorithms are used for the Rosenbrock's valley problem. For constrained problems, MMFD and SQP algorithms are used. The optima and numbers of function calculations are represented in Figs. 2 and 3. Figure 2 is the optimization result of Rosenbrock's valley problem. Figure 2 shows that relatively many function calculations were performed in the SQP algorithm of iSIGHT. Figure 3 is the result of spring design problem. All systems find almost the same optimum. The SQP algorithm of iSIGHT and OPTIMUS need more function calculations.

1. Rosenbrock's valley problem (Kroo *et al.* 1994)

   Find x_1, x_2
   To minimize

   $$f(x) = 100\left(x_2 - x_1^2\right)^2 + (1 - x_1)^2 \tag{1}$$

2. Spring design problem (Haug 1979)

   Find d, D, N

To minimize $f = (N+2)Dd^2$
Subject to

$$g_1 = 1.0 - \frac{D^3 N}{71\,875 d^4} \leq 0.0$$

$$g_2 = \frac{D(4D-d)}{12\,566 d^3(D-d)} + \frac{2.46}{12\,566 d^2} - 1.0 \leq 0.0$$

$$g_3 = 1.0 - \frac{140.54 d}{D^2 N} \leq 0.0$$

$$g_4 = \frac{D+d}{1.5} - 1.0 \leq 0.0 \qquad (2)$$

3. Optimization problem for three-bar truss structure (Haug 1979; Vanderplaats 1998a)

Find A_1, A_2, A_3
To minimize Mass
Subject to

$$-15\,000 \leq \sigma_i \leq 15\,000 \quad i = 1,2,3$$

$$-2.0 \leq \delta_{4,j} \leq 2.0 \quad j = x, y \qquad (3)$$

where
σ_i are the stresses of each element
δ_4 is the displacement in the x- and y- direction at grid point 4

4. Optimization problem for portal frame structure (Haug 1979; Vanderplaats 1998a)

Find Shape of cross sections of the frame
To minimize Mass
Subject to

$$-20\,000 < \sigma_{ij} < 20\,000 \quad i = 1,2,3, \quad j = 1,2$$

$$-4.0 < \delta_3 < 4.0$$

$$-0.015 < \theta_3 < 0.015 \qquad (4)$$

where
σ_{ij} are the axial stresses on the top and bottom at each end of the elements
δ_3 is the displacement in the x-direction at grid point 3
θ_3 is the rotation in the z-direction at grid point 3

Engineering examples are chosen from standard structural optimization problems. The three-bar truss and the portal frame are selected as examples. They are illustrated in Figs. 4 and 5, respectively. Equations (3) and (4) are the formulations of the problems. It is noted that the portal frame example does not satisfy the constraints at the initial design point. GENESIS 6.0 (Vanderplaats 1998a) is interfaced to the design systems and used for the finite element analysis. The results are illustrated in Figs. 6 and 7. The trends are quite similar to that of mathematical problems. In these cases, iSIGHT needs more iterations compared to other systems.

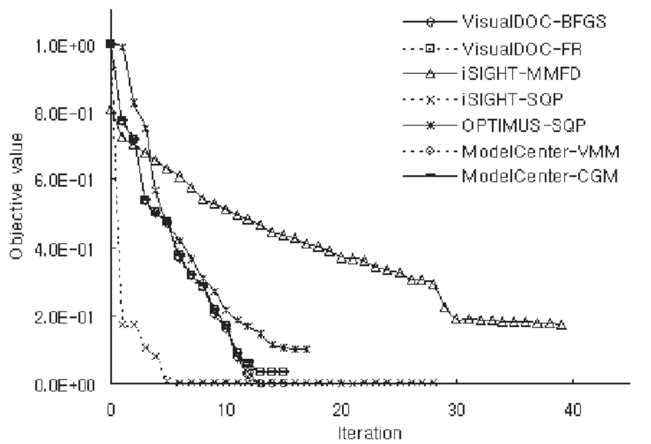

Fig. 2 Optimization results of the Rosenbrock's valley problem

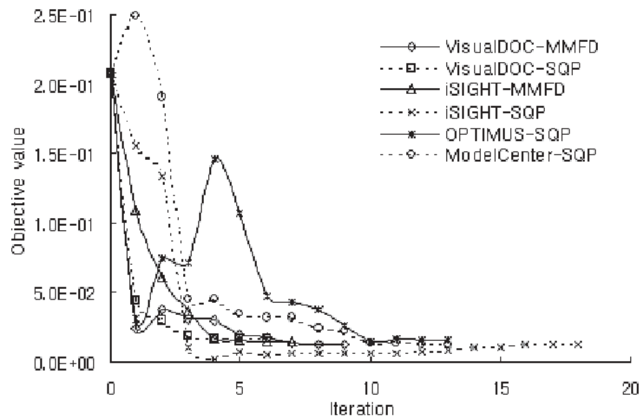

Fig. 3 Optimization results of the spring design problem

A practical problem is selected to see how the systems work for real problems. Also, since a separate analysis software system is utilized, the interface capability with other systems can be compared. Automobile components related to safety are designed. Occupant analysis is carried out for the automotive crash. Multi-body dynamics

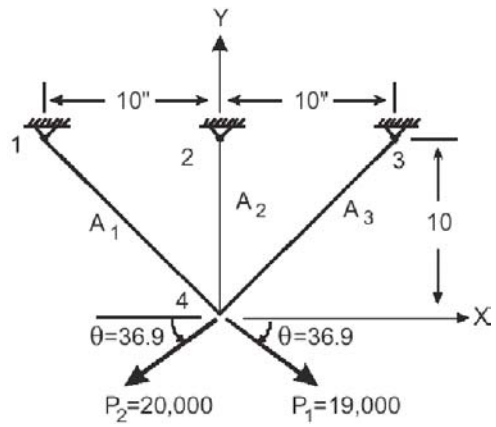

Fig. 4 Three-bar truss

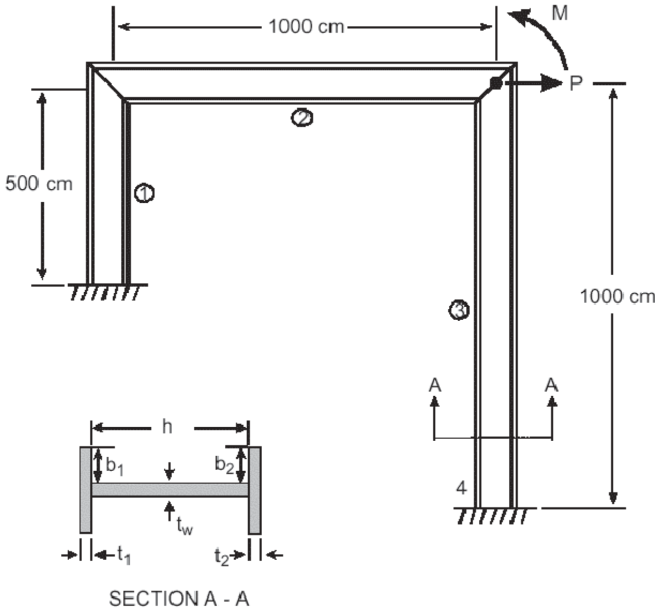

Fig. 5 Portal frame structure

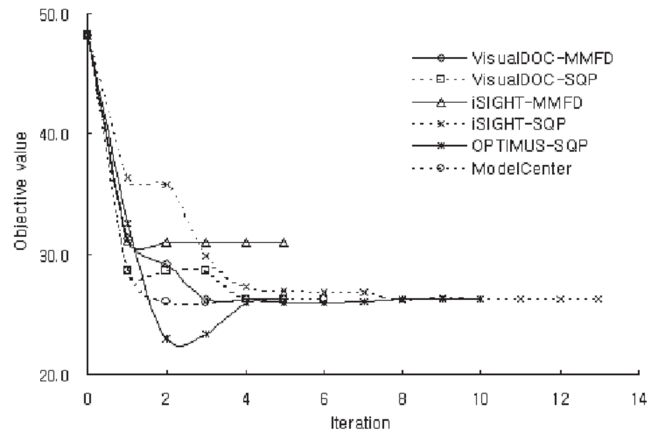

Fig. 6 Optimization results of the three-bar truss

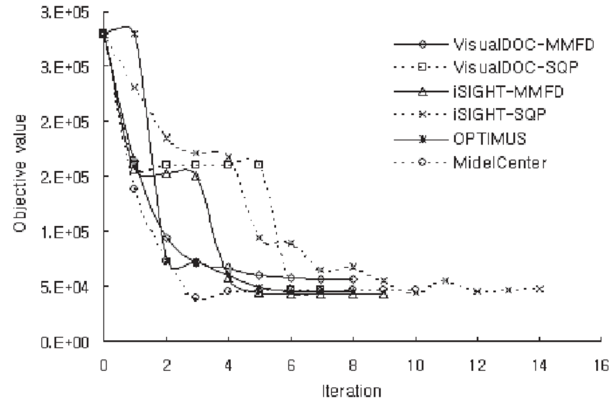

Fig. 7 Optimization results of the portal frame

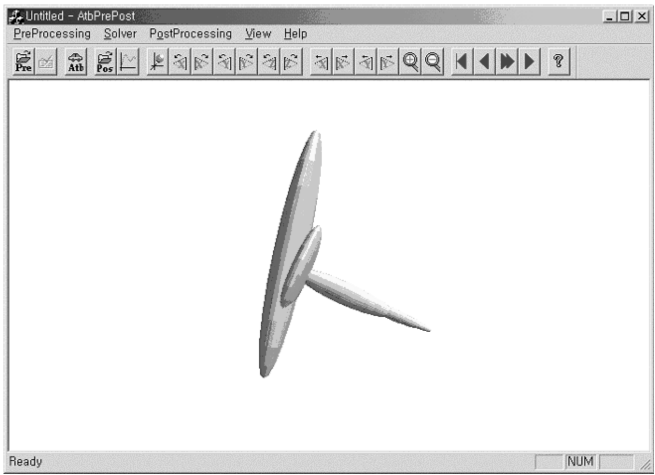

Fig. 8 The simulation model of the body block test with SAFE

is utilized for the analysis. The goal of the occupant analysis is the evaluation of occupant injuries in the crash environment. An in-house program called SAFE (Safety Analysis For crash Environment) (Lim *et al.* 2000) is used to calculate the injuries. SAFE is interfaced to the design software systems.

In this research, the steering column block is optimized. The steering system plays an important role in vehicle crashes since the system impacts directly on the occupant. The steering system has the greatest potential for injuring the driver out of any part of the vehicle interior when the seat belt and airbag are not utilized. Stiffness of molding, plate, column, and rubber flange are defined as design variables to improve energy absorption characteristics of the steering system. The cost function is defined by the impact load on the body block. Detailed formulation and description are beyond the scope of this research and are explained elsewhere (Park *et al.* 1995, 1996; Shin *et al.* 2001).

A simulation model is illustrated in Fig. 8. SAFE uses a fixed format input file that must be written in the specified format of the software. Two or more pieces of data in one line of input/output files are utilized in the optimization process. This is one of the most difficult cases in which to control input/output data. OPTIMUS and ModelCenter cannot control the input file when two pieces of data in a line are used. Therefore, the performances of OPTIMUS and ModelCenter could not be compared.

Occupant analysis has highly nonlinear characteristics. Therefore, gradient-based optimization methods are not efficient because they are quite costly for highly nonlinear or noisy problems. Sometimes, they do not find a solution of good quality. Thus, optimization methods without gradients would be a better choice. A modified response surface method (RSM) of VisualDOC is utilized. The detailed algorithm is explained elsewhere (Vanderplaats 1984a). Also, iSIGHT has a method that finds the optimum through an iterative RSM method without gradient information. The problem is solved by this method as well. iSIGHT also has another algorithm

Table 6 Optimization result of the steering column problem with SAFE

		Initial	VisualDOC	iSIGHT
Design Variables	S03	1.0000	0.4320	0.7438
	S06	1.0000	0.3000	0.6888
	S11	1.0000	0.3600	0.4731
	S12	1.0000	0.3000	0.7010
Objective		1.000	0.883	0.921
Constraint		1.000	0.989	0.990
Iteration		–	33	21

that does not use gradient information such as genetic algorithms (GAs). However, only RSMs are utilized for fair comparison. The results are shown in Table 6 where the results are normalized by nominal value. The table shows that the design is improved in both cases. VisualDOC performs function calculations more than iSIGHT but finds a better design.

All of the results show that VisualDOC has the best optimization algorithms. Better optima are found by the algorithms of VisualDOC and the SQP algorithm of iSIGHT in all mathematical and engineering examples. However, the SQP algorithm performs many more function calculations.

4.2 Comparison of DOE modules

Recently, DOE has been receiving more attention because it is easy to use. DOE (Park 1991; Phadke 1989) is a design method that specifies a design matrix to find the characteristic of the model and analyzes results with a statistical method. All the design software systems have a DOE design module.

Table 7 Supplied types of experiments

No.	VisualDOC	iSIGHT	OPTIMUS	ModelCenter
1	○	○	○	×
2	○	○	○	○
3	○	○	○	×
4	×	○	○	×
5	○	×	○	×
6	○	×	○	×
7	○	○	○	○
8	○	○	○	×

1: Full/fractional factorial
2: Central composite design
3: Orthogonal arrays
4: Latin hypercubes
5: Box–Behnken
6: Plackatt–Burman
7: User define
8: Etc.

Table 8 Supplied output results of DOE

No.	VisualDOC	iSIGHT	OPTIMUS	ModelCenter
1	○	○	×	×
2	×	○	×	○
3	×	○	×	×
4	○	○	○	×
5	×	○	×	×
6	○	○	○	×

1: ANOVA
2: Main effects
3: Interaction effects
4: Scatter
5: Pareto graph
6: Contribution

Design is performed with the same statistical analysis in the design process. If input data are the same, the design results are exactly the same for all the systems. Therefore, comparison of the design results is meaningless. However, it is important to have various design matrices and analysis capability for the design results. Supplied design matrices and analysis capability are shown in Tables 7 and 8. VisualDOC, iSIGHT, and OPTIMUS provide many design matrices. iSIGHT has various analy-

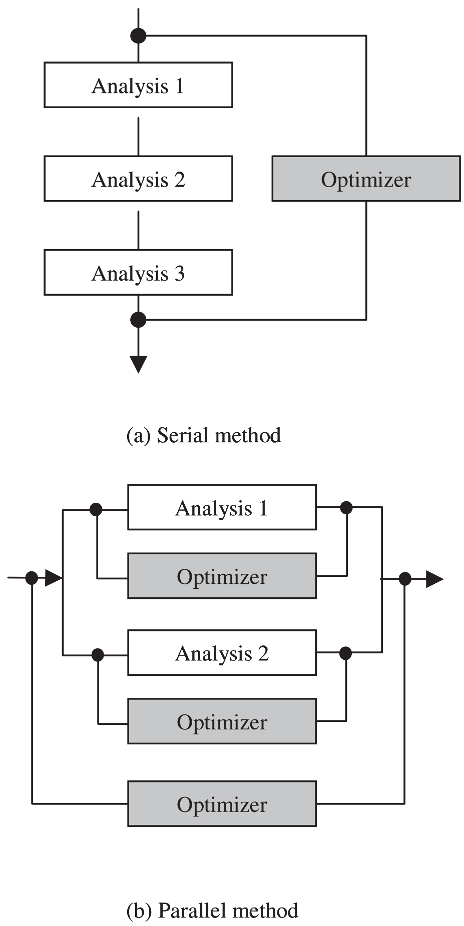

Fig. 9 Serial and parallel usages of the optimizers

sis capabilities for design results. It seems that iSIGHT has many advantages in the DOE module.

5
Limitations of design software packages

Recently, many analysis software systems have been developed in many disciplines. Also, various design methods have been developed. One of the design methods is multidisciplinary design optimization (MDO). MDO is a design method that concerns two or more disciplines simultaneously. For example, a product should be designed by considering the structure and fluid theories, or structure and control theories at the same time in MDO problems. In these cases, two or more analysis software systems may be used in the design process.

The selected systems can control more than two analysis systems in the design process. However, they serially use analysis systems as illustrated in Fig. 9a. Some MDO methods require parallel usage as illustrated in Fig. 9b. None of the selected systems can be used in the paradigm of Fig. 9b. Although some systems emphasize that they can be used in the MDO environment, the flexibility is still lacking. This aspect should be considered for future development.

6
Conclusions

Characteristics and performance of the design software systems are compared. The comparisons are made based on the characteristics of design environment specifications and the performances of optimization modules. Mathematical and engineering examples are solved for the evaluation.

1. VisualDOC has a powerful optimization module. Some unstable functions exist in the design environment specification, especially for interface functions with input/output files. Therefore, some improvements are needed for those functions.
2. iSIGHT is excellent in the design environment specification. The designer can easily specify the design environment. It has various capabilities in the DOE module compared to others. However, performances of the optimization algorithms are not as efficient as those of others.
3. OPTIMUS has different characteristics from others. The GUI displays the relationship between input/output variables and analysis software with a diagram. However, there are some unstable functions in the interface functions between the input/output files.
4. ModelCenter separates the design environment specification and system design software with an analysis server and ModelCenter. Therefore, ModelCenter can be the most powerful in the network environment. However, ModelCenter also has some instability in interface with input/output files.

The common deficiencies of all systems are pointed out in the previous section. They cannot control multiple design modules simultaneously. Some MDO methods need multiple design modules in one design process. Selected systems are not adequate for these cases. Therefore, the systems should improve flexibilities for diverse usages.

Acknowledgements This research was supported by the Center of Innovative Design Optimization Technology, Korea Science and Engineering Foundation. The authors thank Mrs. MiSun Park for correcting the manuscript.

References

Arora, J.S. 1989: *Introduction to optimum design*, int. edn. New York: McGraw-Hill

Bradley, S.P.; Hax, A.C.; Magnanti, T.L. 1977: *Applied mathematical programming*. MA: Addison-Wesley

Engineous Software 1999: *iSIGHT designer's guide version 5.0*. North Carolina: Engineous Software

Fletcher, R. 1970: A new approach to variable metric algorithms. *Comput. J.* **13**, 317–322

Fletcher, R.; Powell, M.J.D. 1964: Function minimization by conjugate gradients. *Comput. J.* **7**, 149–160

Ghosh, D.K.; Garcelon, J.H.; Balabanov, V.O.; Vanderplaats, G.N. 1999: Development of a flexible design optimization study tool. *Proc. 7-th AIAA/USAF/NASA/ISSMO Symp. on Multidisciplinary and Structural Optimization*. (St. Louis, MO), pp. 2–4

Ghosh, D.K.; Garcelon, J.H.; Balabanov, V.O.; Venter, G.; Vanderplaats, G.N. 2000: VisualDOC – A flexible design optimization software system. *Proc. 5-th AIAA/USAF/NASA/ISSMO Symp. on Multidisciplinary and Structural Optimization*. (Long Beach, CA), pp. 6–8

Haftka, R.T.; Gurdal, Z. 1991: *Elements of structural optimization*. The Netherlands: Kluwer

Haug, E.J.; Arora, J.S. 1979: *Applied optimal design*. New York: Wiley

Kroo, I.; Altus, S.; Braun, R.; Gage, P.; Sobieski, I. 1994: Multidisciplinary optimization methods for aircraft preliminary design. *AIAA Paper 94-4325*, 697–707

Lim, J.M.; Wu, D.H.; Park, G.J. 2000: Analysis and design consideration of an energy absorbing steering system using orthogonal arrays. *Int. J. Crashworthiness* **5**, 271–278

LMS 1999: *OPTIMUS users manual version 2.1*. Leuven, Belgium: LMS International

Park, S.H. 1991: *Modern design of experiment*. Seoul: Minyoungsa (in Korean)

Park, Y.S.; Lee, J.Y.; Park, G.J. 1995: An optimal deign of a steering column to minimize the injury of a passenger. *Transactions of the Korea Society of Automotive Engineers* **3**, 33–44 (in Korean)

Park, Y.S.; Lee, J.Y.; Lim, J.M.; Park, G.J. 1996: Optimum design of a steering column to minimize the injury of passenger. *Int. J. Veh. Des.* **17**, 398–414

Phadke, M.S. 1989: *Quality engineering using robust design*. New Jersey: Prentice Hall

Phoenix Integration 1999: *ModelCenter training and help*. Virginia: Phoenix Integration

Rao, S.S. 1996: *Engineering optimization theory and practice*. New York: Wiley

Shin, M.K.; Hong, S.W.; Park, G.J. 2001: Axiomatic design of the motor driven tilt/telescopic steering system for safety and vibration. *Proc. Inst. Mech. Eng. Part D-J. Automob. Eng.* **215**, 179–187

Sobieszcznski-Sobieski, J.; Haftka, R.T. 1996: Multidisciplinary aerospace design optimization: survey of recent developments. *AIAA Paper 96-0711*, 1–32

Thanedar, P.B.; Arora, J.S.; Tseng, C.H.; Lim, O.K.; Park, G.J. 1987: Performance of some SQP algorithms on structural design problems. *Int. J. Numer. Methods Eng.* **23**, 2187–2203

Vanderplaats, G.N. 1984a: *Numerical optimization techniques for engineering design*. New York: McGraw-Hill

Vanderplaats, G.N. 1984b: An efficient feasible direction algorithm for design synthesis. *AIAA J.* **22**, 1798–1803

Vanderplaats, G.N. 1998a: *Genesis user manual version 5.0*. Colorado Springs: Vanderplaats Research & Development

Vanderplaats, G.N. 1998b: *VisualDOC manual version 1.0*. Colorado Springs: Vanderplaats Research & Development

Vanderplaats, G.N.; Sugimoto, H. 1985: Application of variable metric methods to structural synthesis. *Eng. Comput.* **2**, 96–100

Zoutendijk, K.G. 1960: *Methods of feasible directions*. Amsterdam: Elsevier