

## Comparison of some commercial software systems for structural optimization

Wook-han Choi<sup>1</sup>, Cheng-guo Huang<sup>2</sup>, Jong-moon Kim<sup>3</sup>, Gyung-Jin Park<sup>3\*</sup>

<sup>1</sup> Hanyang University, Seoul, Republic of Korea, dreamybird@hanyang.ac.kr

<sup>2</sup> Hanyang University, Seoul, Republic of Korea, huangcg@hanyang.ac.kr

<sup>3</sup> Hanyang University, Seoul, Republic of Korea, thcbdie@hanyang.ac.kr

<sup>4</sup> Hanyang University, Ansan, Republic of Korea, gjpark@hanyang.ac.kr

### 1. Abstract

Mathematical optimization theories are employed for the design of structures in structural optimization. These days, structural optimization is widely utilized for practical problems due to well-developed commercial software systems. Three representative general-purpose structural optimization systems such as Genesis, MSC.Nastran and OptiStruct are investigated and evaluated by solving various test examples in different scales. The performance of structural optimization depends on the quality of the optimum solution and the computational time, and these aspects are compared from the application viewpoint. For fair comparison, the same formulations are defined, and the same optimization methods are adopted for each example. Also, the same system environment is prepared, and the same optimization parameters are used. Linear static response size, shape and topology optimizations are applied to the examples and the results are compared. No system seems to be the best in all the cases and each system has advantages and drawbacks depending on the application. In general, Genesis is excellent in the computational time while OptiStruct gives excellent solutions, especially in topology optimization. Meanwhile, MSC.Nastran presents excellent solutions in shape optimization.

**2. Keywords:** Structural Optimization, Design Software, Performance comparison

### 3. Introduction

Optimization generally finds design variables to maximize/minimize an objective function, while design constraints are simultaneously satisfied. In structural optimization, the optimization problem is defined for the design of a structure. Nowadays, structural optimization is widely accepted due to the development of the finite element method (FEM) [1-3]. Structural analysis and optimization techniques have been continuously developed based on FEM. Recently, commercial design software systems are also well-developed to solve structural design problems, and they have their own FE analysis modules. They are MSC.Nastran [4], Genesis [5], OptiStruct [6], ANSYS [7], TOSCA [8], etc.

As the FE models become more complex and larger, designers are interested in the decision on the choice of a structural optimization software system. One of the reasons is that the systems give different performances. It is natural that a designer wants to use an appropriate system for her/his applications. However, there is no comparison study for the structural optimization systems. Some researches were performed with regard to the performance of structural optimization methods. Structural optimization is classified into the direct method and the approximation method. The direct method directly applies nonlinear programming (NLP) to a structural optimization problem; therefore, an NLP algorithm controls the overall process. On the other hand, the functions are approximated to explicit functions of design variables and an NLP algorithm is applied to the approximated problem. An approximation method is generally employed by commercial systems for structural optimization while the direct method is generally used in the academic sites. The performances of the direct method and the approximation method were compared [9]. The efficiency of an NLP algorithm is not very critical in the approximation method while it is crucial in the direct method. A comparative study of the optimization software systems, which have various NLP algorithms, was performed in [10].

In this research, three popularly used commercial structural optimization software systems such as Genesis, MSC.Nastran 2013.1 [4], Genesis 13.1 [5] and OptiStruct 13.0 [6] are investigated and evaluated. These software systems provide optimum solutions based on a gradient based optimization method and have their own FE analysis modules. Also, they have all the capabilities for size, shape, topology optimizations. The systems that do not have these conditions are excluded in this study. Linear static response size, shape and topology optimizations are explored and compared by various test examples. The examples cover small, medium and large scale problems. Evaluations of the systems are made for the quality of the optima and the computational cost.

### 4. Environment for comparison

Various factors can have influence on the performance of the structural optimization software system. They can be classified into the optimization environment and the system environment. The optimization environment consists of optimization formulation, convergence criteria, utilized optimization method, move limit strategy, constraint screening method, etc. Meanwhile, the system environment is determined by the performances of the computer such as the operating system, CPU, amount of memory usage, etc. It should be assured that the same conditions are used for a fair comparison. Some conditions are controllable in one software system but uncontrollable in another system. In that case, unification of the variables is made as much as possible. If the unification is impossible, default values of the software systems are used.

#### 4.1 Optimization formulation

The formulation of linear static response structural optimization is represented as follows [11-12]:

$$\begin{aligned}
 &\text{Find} && \mathbf{b} \in R^n, \mathbf{z} \in R^l, \zeta \in R^l \\
 &\text{to minimize} && f(\mathbf{b}, \mathbf{z}, \zeta) \\
 &\text{subject to} && \mathbf{K}(\mathbf{b})\mathbf{z} = \mathbf{f} \\
 &&& \mathbf{K}(\mathbf{b})\mathbf{y} - \zeta\mathbf{M}(\mathbf{b})\mathbf{y} = 0 \\
 &&& \mathbf{g}_j(\mathbf{b}, \mathbf{z}, \zeta) \leq 0, j = 1, \dots, m \\
 &&& \mathbf{b}_L \leq \mathbf{b} \leq \mathbf{b}_U
 \end{aligned} \tag{1}$$

where  $\mathbf{b}$  is the design variable vector,  $\mathbf{z}$  are the state variable vector, and  $\xi$  and  $\mathbf{y}$  are the eigenvalue and eigenvector, respectively.  $n$  is the number of the design variables,  $l$  is the number of the state variables, and  $m$  is the number of the constraints, respectively.  $\mathbf{b}_L$  and  $\mathbf{b}_U$  are the lower bound and the upper bound of the design variable vector  $\mathbf{b}$ , respectively.  $f$  is the objective function and  $\mathbf{g}_j$  is the  $j$ th constraint.  $\mathbf{K}$  is the stiffness matrix,  $\mathbf{M}$  can be either the mass matrix or geometry matrix, respectively. Each example uses an identical optimization formulation for all the software systems for a fair comparison.

#### 4.2 Optimization method

An optimization algorithm has to be selected, but this selection is not very important because an approximation is used for the software systems. Modified method of feasible directions (MMFD), sequential linear programming (SLP) and sequential quadratic programming (SQP) are commonly supported by the three software systems. In MSC.Nastran, an optimization method can be automatically selected for a better performance. However, the utilized optimization method is unified by MMFD [13-14]. Each software system supports a separate optimizer for topology optimization and large scale problems. IPOPT [15], BIG-DOT [5] and Dual-Optimizer [16] are the optimizers in Genesis, MSC.Nastran and OptiStruct, respectively. These are based on the sequential unconstrained minimization technique (SUMT) [14].

#### 4.3 Others

The convergence criteria of the optimization may affect the performance. In this research, the relative change of the objective function is used for a convergence criterion and the bound of the value is set by 0.001. Meanwhile, the normalized maximum constraint violation is used as another convergence criterion and set by 0.01. Because OptiStruct does not allow changing this criterion, the default value of OptiStruct is adopted. The move limit strategy is considered. A move limit is defined for the change of the design variables in an approximation problem. The value of the move limit can be defined in the three software systems. The lower and upper bounds of the move limit can be defined by a user in MSC.Nastran and Genesis; however, they cannot be defined by a user in OptiStruct. Therefore, the default values of the bounds are used. The same active set strategy for constraints is considered for large scale problems.

The system environments are also unified. The utilized operating system is MS Windows x64 Ultimate (version 6.1, Build 7601) that is commonly supported by the three software systems. The hardware system is 16.0GB Memory, 8 CPU and Intel core i7-3770 at 3.40GHz. The amount of memory usage has a significant impact on the performance of a software system. Three software systems support memory control options in different ways; however, the total amount of memory usage can be commonly determined. The amount of memory usage is unified as 8GB.

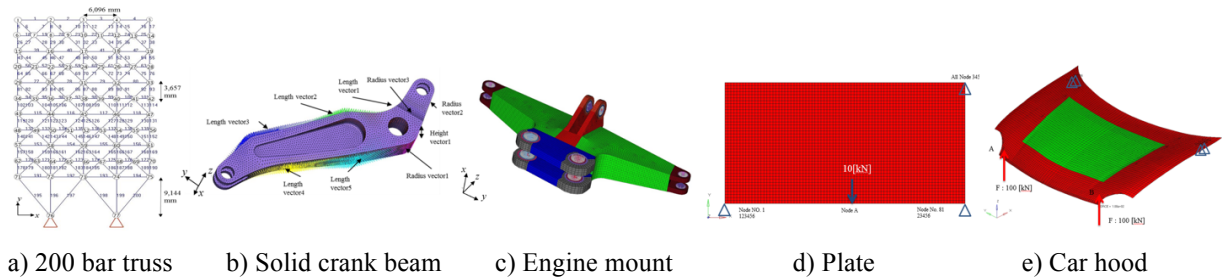


Figure 1: Finite element model of the structural optimization examples

## 5. Structural optimization

Structural optimization is classified into size, shape, topology, topometry and topography optimizations [4-6]. Size, shape and topology optimizations are classical optimization methods. On the other hand, topometry, topography optimizations are non-classical optimization methods. Structural optimization is classified based on the characteristics of the design variables [12]. The domain of FE analysis is not changed by design variables in size and topometry optimization. However, the domain of shape and topography optimization can be changed by the design variables. In the case of topology optimization, the distribution or existence of materials is determined by the design variables. In topometry optimization, each designable element has an independent design variable. Likewise, normal perturbation vectors are defined at all designable grids in topography optimization. Structural optimization examples are explored regarding each optimization.

### 5.1 Structural optimization examples

Many structural optimization examples are solved for the comparison of performance. However, only the representative examples, which show distinct characteristics of the software systems, are presented here. They are a 200 bar truss example [17], a solid crank example [18], an engine mount example [19], a plate example [20], and a car hood example [5] for size, shape, topology, topometry and topography optimization, respectively. The examples are illustrated in Figure 1. The detailed optimization formulation of each example is in [21]. The characteristics of each problem are summarized in Table 1.

### 5.2 Results of optimization

The optimization results for each example are shown in Table 2. The initial and final objective function values are compared. In topology optimization, the final objective function value, which is evaluated after an additional post-processing, is shown. The initial objective function values of the plate example are different for each software system, because the initial plate thickness is automatically determined by using the lower and upper bounds of the design variables. CPU time means the computational time for the optimization process and the elapsed time means the total time that includes the time for writing results and checking the license, etc.

Table 1: Structural optimization problems

Example	Design variables	Objective function	Constraints	Loading conditions	Remark
200 bar truss	96	Mass	Stress and displacement constraints	Multiple loading conditions	Size optimization
Solid crank beam	9	Mass	Stress constraints	A single loading condition	Shape optimization
Engine mount	57,280	Compliance	Fraction mass constraint	Multiple loading conditions	Topology optimization
Plate	3,200	Compliance	Volume constraint	A single loading condition	Topometry optimization
Car hood	2,069	Compliance	.	A single loading condition	Topography optimization

Table 2: Results of structural optimization problems

		Objective Function value		Iterations	Func. calls	CPU Time (s)	Elapsed time (s)
		Initial	Final [Final after post processing]				
200 bar truss	MSC.Nastran	5.082E+04	1.380E+04	18	18	10.84	31.20
	Genesis	5.082E+04	1.390E+04	13	13	0.82	1.00
	OptiStruct	5.082E+04	1.354E+04	25	25	1.40	21.56
Solid crank beam	MSC.Nastran	1.396E-02	1.439E-02	25	25	182.10	387.48
	Genesis	1.396E-02	1.440E-02	12	12	32.00	45.00
	OptiStruct	1.396E-02	1.448E-02	8	8	22.90	28.22
Engine mount	MSC.Nastran	3.560E+06	4.033E+05 [4.986E+05]	64	64	2232.00	4329.27
	Genesis	3.561E+06	4.839E+05 [3.517E+05]	21	21	1449.00	1869.00
	OptiStruct	3.561E+06	3.455E+05 [3.456E+05]	82	82	2304.01	3277.05
Plate	MSC.Nastran	1.229E+01	1.186E+01	16	16	8.47	23.65
	Genesis	1.096E+01	1.032E+01	11	11	4.76	5.00
	OptiStruct	1.799E+01	7.898E+00	17	17	6.46	14.57
Car hood	MSC.Nastran	3.6071E+04	2.822E+04	8	8	131.96	311.961
	Genesis	3.6054E+04	2.876E+04	15	15	15.00	24.00
	OptiStruct	3.5336E+04	2.975E+04	7	7	5.87	10.55

In size, topology and topometry optimizations, it seems that OptiStruct gives the best objective function value although more CPU time is necessary. Genesis gives solutions with the shortest CPU time. In shape and topography optimization, the optimum value from MSC.Nastran looks the best. However, the computational time is quite long compared to the other software systems. Figure 2 shows the results of shape, topology and topography optimization, respectively. There is no big difference in shape and topography optimization results among the three software systems. However, the final shapes from MSC.Nastran are different from those of other software systems, especially in topology optimization.

## 6. Structural optimization of a large scale structure

As the FE models of the structures become more complex and larger, the performance comparison of large scale structures is quite important. The scale of the problem depends on the size of the FE model and the number of design variables. The number of constraints is not considered in the decision of the scale if an active set strategy for constraints is used. Actually, topology, topometry and topography optimizations can be considered as large scale problems, because they are using many design variables. In this study, a large scale problem means an optimization problem with a large FE structure and many design variables.

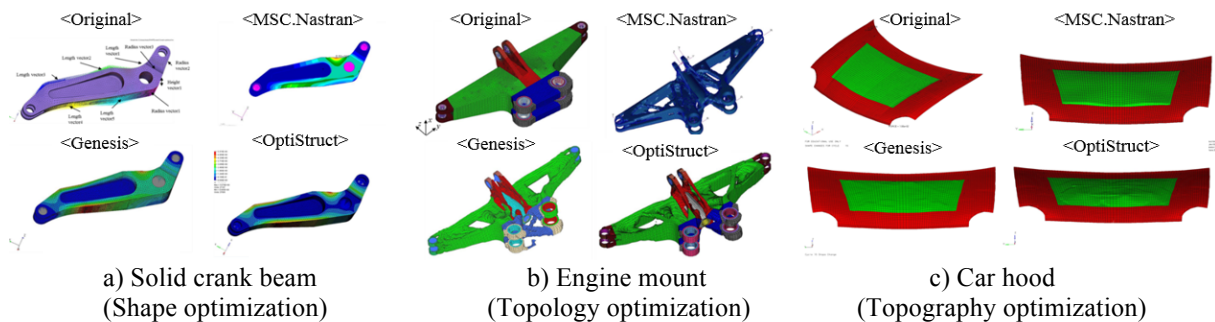


Figure 2: Results of optimization examples

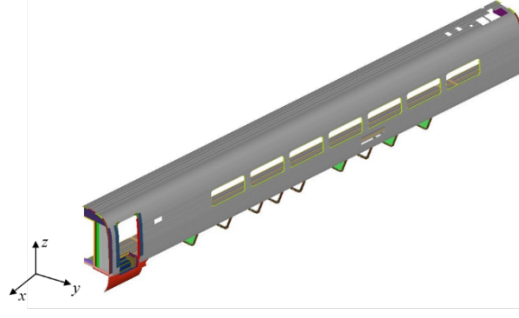


Figure 3: Finite element model of the passenger train

### 6.1 Structural optimization of a passenger train

An optimization of a passenger train is solved [22]. As illustrated in Fig. 3, the width, height and length are 1.5m, 3.0m, and 23.5m, respectively. The passenger train model consists of shell and solid elements. The total number of FE elements is 239,020 and the number of design variables is 3,410. Five loading conditions are applied as multiple loading conditions. The design formulation is as follows:

$$\begin{aligned}
 &\text{Find} && b_i, b_j \in R^n \\
 &\text{to minimize} && \text{Mass} \\
 &\text{subject to} && \sigma_{\text{von Mises}} \leq \sigma_{\text{allow}} \quad (2) \\
 &&& \delta_{z, \text{initial}} \leq \delta_{z, \text{current}} \\
 &&& b_{i, \text{lower}} \leq b_i \leq b_{i, \text{upper}}, \quad (i=1, \dots, 3398) \\
 &&& b_{j, \text{lower}} \leq b_j \leq b_{j, \text{upper}}, \quad (j=1, \dots, 12)
 \end{aligned}$$

where  $b_i$  is the  $i$ th size variable,  $b_j$  is  $j$ th the shape variable, and  $b_{\text{lower}}$  and  $b_{\text{upper}}$  are the lower bound and the upper bound of the design variable, respectively.  $\sigma_{\text{von Mises}}$  is the von Mises stress,  $\sigma_{\text{allow}}$  is the allowable stress,  $\delta_{z, \text{initial}}$  is the initial displacement in the  $z$ -axis, and  $\delta_{z, \text{current}}$  is the displacement in the  $z$ -axis of the currently existing model, respectively. The objective function to be minimized is the mass of the structure while the displacement and stress constraints are satisfied. For this example, the amount of memory usage is extended to 64GB, because OptiStruct did not work with 16GB memory.

### 6.2 Results of the large scale example

The optimization results are shown in Table 2. In a large scale problem, the elapsed time is larger than the computational time, because there are many other processes in addition to the pure optimization process. It is noted that the optimum values are similar. That means the three systems can handle large scale optimization problems. OptiStruct is excellent in the computational time, even though it has a memory control issue. Genesis shows a good performance in the computational time. The computational time of MSC.Nastran is quite long compared to the other software systems. But it has no memory control issue. Actually, this example can be solved with 8GB memory by MSC.Nastran and Genesis.

Table 2: Results of the large scale example

		Objective Function value		Iterations	Func. Calls	CPU time	Elapsed time
		Initial	Final				
Passenger train	MSC.Nastran (IPOPT optimizer)	4.2739	3.5026	11	11	17h 39min	5d 9h 30min
	Genesis (BIGDOT optimizer)	4.2739	3.4972	11	11	1h 16min	3h 28min
	OptiStruct (Dual optimizer)	4.2739	3.4449	9	9	26min	46min

## 7. Conclusions

Linear static response optimizations are explored, and three commercial structural optimization systems are compared through various structural examples. Only representative examples are demonstrated here. We could see some performance distinctions out of the three software systems. No system is the best in all the cases and each system has advantages and disadvantages depending on the application. In general, Genesis is excellent in computational time while OptiStruct provides excellent solutions, especially in topology optimization. MSC.Nastran presents good solutions in shape optimization. In the case of a large scale example, the three systems give similar objective function values. OptiStruct is excellent in computational time; however, there is a memory control issue that is not found in Genesis and MSC.Nastran.

There can be various reasons for performance distinction, because performance is determined by a combination of many factors. The method for approximation is different in each of the three software systems. The move limit strategy is slightly different. Some parameters can be controllable in one system but uncontrollable in other system. This aspect should be theoretically investigated.

It is noted that the three systems never fail in any examples. That means that all of them are quite reliable. Also, even with one system, different results can be obtained when different decisions are made for the selection of the parameters. The authors hope that this paper helps practitioners in the decision of a structural optimization system.

## 8. Acknowledgements

This research was supported by the Commercialization Promotion Agency for R&D Outcomes through the Ministry of Science (No. 2014K00042). The authors are thankful to Mrs. MiSun Park for her English correction of the manuscript.

## 9. References

- [1] R.D. Cook, D.S. Malkus, M.E. Plesha, and R.J. Witt, *Concepts and applications of finite element analysis*, fourth edition, John Wiley and Sons. Inc., NY, USA, 2001.
- [2] K.J. Bathe, *Finite element procedures in engineering analysis*, Prentice Hall, Englewood Cliffs, NJ, USA, 1996.
- [3] D.L. Logan, *A first course in the finite element method*, PWS Publishing Company, Boston, MA, USA, 1993.
- [4] MSC NASTRAN User's Guide, Ver. 2013.1, MSC Software Corporation, Santa Ana, CA, USA, 2013.
- [5] GENESIS User's Manual, Ver. 13.1, Vanderplaats Research and Development. Inc., Colorado Springs, CO, USA, 2014.
- [6] ALTAIR OPTISTRUCT User's Manual, Ver. 13.0, Altair Engineering. Inc., Troy, MI, 2014.
- [7] P. Kohnke, ANSYS theory reference manual, ANSYS. Inc., Canonsburg, PA, USA, 1994.
- [8] <http://www.3ds.com/products-services/simulia/products/tosca/>, Dassault systemes.
- [9] Y.S. Park, S.H. Lee and G.J. Park, A study of direct vs. approximation methods in structural optimization. *Structural Optimization*, Springer-Verlag, 10, 64-66, 1995.
- [10] U.P. Hong, K.H. Hwang and G.J. Park, A comparative study of software systems from the optimization viewpoint, *Structural and Multidisciplinary Optimization*, 27(6), 460-468, 2004.
- [11] R.T. Haftka and Z. Gurdal, *Elements of structural optimization*, Kluwer Academic Publishers, Dordrecht, Netherlands, 1992.
- [12] G.J. Park, *Analytical Methods in Design Practice*, Springer, Germany, 2007.
- [13] G.N. Vanderplaats, *Numerical optimization techniques for engineering design*, Vanderplaats Research & Dvelopment. Inc., Colorado Springs, CO, USA, 1999.
- [14] J.S. Arora, *Introduction to optimum design*, Elsevier, Waltham, MA, USA, 2012.
- [15] MSC NASTRAN Design Sensitivity and Optimization User's Guide, Ver. 2013.1, MSC Software Corporation, Santa Ana, CA, USA, 2013.
- [16] C. Fleury and V. Braibant, A new dual method using mixed variables, *International Journal for Numerical Methods in Engineering*, 23(3), 409-428, 1986.
- [17] E.J. Haug and J.S. Arora, *Applied Optimal Design*, New York, U.S.A, 1979.
- [18] ALTAIR HyperStrudy Tutorials, Ver. 13.0, Altair Engineering. Inc., Troy, U.S.A, 2014.
- [19] MD NASTRAN Demonstration Problems, Ver.2013.1, MSC Software Corporation, Santa Ana, CA, USA, 2013.
- [20] MSC NASTRAN Automated structural optimization in MSC Natran., MSC Software Corporation, Santa Ana, CA, USA, 2014.
- [21] C.G. Huang, A comparative study of performance of structural optimization software, Master's theses, Hanyang University, Korea, 2014. (In Korean)
- [22] Y.M. Lee, H.A. Lee, Y.W. Yoon and G.J. Park, Design of aluminium extrusion panels in a railway-car body using orthogonal arrays, *The seventh China-Japan-Korea joint symposium on optimization of structural and mechanical systems*, Huangshan, China, 2012.