

Comparison study of some commercial structural optimization software systems

Wook-han Choi¹ · Jong-moon Kim¹ · Gyung-Jin Park²

Received: 6 October 2015 / Revised: 24 January 2016 / Accepted: 1 March 2016 / Published online: 1 April 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract Mathematical optimization theories are employed for the design of structures in structural optimization. Structural optimization is being widely utilized for practical problems due to well-developed commercial software systems. Three representative structural optimization systems such as Genesis, MSC Nastran and OptiStruct are investigated and evaluated by solving various test examples in different scales. The design capabilities of three software systems are explored and the performances of the systems are compared. The performance of structural optimization depends on the quality of the optimum solution and the computational time, and these aspects are compared from an application viewpoint. For a fair comparison, the same formulations are utilized, and the same optimization methods are employed for each example. Also, the same system environment is prepared, and the same optimization parameters are used. Additionally, various design options of each software system are tested for the best performance. Linear static response size, shape, topology, topometry and topography optimizations are applied to the examples and the results are compared. No system seems to be the best in all cases and each system has advantages and disadvantages depending on the application. In general, Genesis is excellent in computational time while

OptiStruct gives excellent optimum solutions, in size, topometry and topology optimizations. Meanwhile, MSC Nastran presents good solutions in shape and topography optimizations.

Keywords Structural optimization · Optimization software · Performance of optimization

1 Introduction

Structural optimization was initiated by Schmit to solve a mathematically described structural optimization problem in 1960 (Schmit 1960). 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) (Cook et al. 2001; Bathe 1996; Logan 1993) and the optimization system that uses the FEM (Haftka and Gürdal 1992; Park 2007).

A general formulation of linear response structural optimization is represented as follows: (Haftka and Gürdal 1992; Park 2007)

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

where \mathbf{b} is the design variable vector, \mathbf{z} is the state variable (displacement) vector, and ξ and \mathbf{y} are the eigenvalue and

✉ Gyung-Jin Park
gjpark@hanyang.ac.kr
Wook-han Choi
dreamybird@hanyang.ac.kr
Jong-moon Kim
thcbdie@hanyang.ac.kr

¹ Hanyang University, Seoul, Republic of Korea
² Hanyang University, Ansan, Republic of Korea

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. \mathbf{b}_L and \mathbf{b}_U are the lower bound and the upper bound vectors of the design variable vector \mathbf{b} , respectively. f is the objective function and g_j is the j th constraint. \mathbf{K} is the stiffness matrix, \mathbf{M} can be either the mass matrix or geometry matrix. In the gradient based optimization method, a nonlinear programming (NLP) algorithm is employed to solve an optimization problem.

The solving methods for structural optimization are classified into the direct method and the approximation method in size and shape optimizations (Park et al. 1995; Park 2007). The direct method directly applies an NLP algorithm to a structural optimization problem; therefore, the NLP algorithm controls the overall process. In the approximation method, the functions are approximated to explicit functions of design variables and an NLP algorithm handles the approximated functions. Commercial systems generally employ the approximation method while the direct method is frequently used in academic sites. Two methods such as the homogenization method and the density method have been developed for topology optimization, and the density method is usually adopted for topology optimization in commercial systems (Bendsøe and Sigmund 2003). A detailed explanation of each method is beyond the scope of this study.

It is natural that a practitioner wants to use an appropriate commercial optimization system for her/his applications. Some studies were performed with regard to the performance of structural optimization methods. The performances of the direct method and the approximation method were compared (Park et al. 1995). 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 (Hong et al. 2004). However, the commercial structural optimization systems are not rigorously compared yet. In this study, some commercial structural optimization systems are sophisticatedly compared for linear static response optimization.

Several criteria are defined for the choice of the commercial systems to be evaluated. They are (1) the system should have its own module for FE analysis. (2) An approximation method is used for size and shape optimizations, and sensitivity information regarding FE analysis should be used in the optimization process. (3) Exact sensitivity information is utilized in topology optimization. (4) The system should be able to handle size, shape, topology, topometry and topography optimizations. (5) The system should be easily available on the market.

Based on the above criteria, three commercial systems are selected, and they are MSC Nastran (MSC Nastran user's guide version 2013a), Genesis (Genesis user's manual version

13.1 design reference 2014) and OptiStruct (Altair OptiStruct user's manual version 13.0 2014). These systems satisfy the above requirements. There are some other commercial systems such as ANSYS (<http://www.ansys.com/About+ANSYS>), Tosca (<http://www.3ds.com/products-services/simulia/products/tosca/>), etc. Since these systems do not satisfy the requirements, they are excluded from this study.

First, the characteristics and capabilities of the selected systems are investigated. That is, theoretical as well as numerical aspects of the systems are analyzed and compared. The computer environment is set to have the same conditions for a fair comparison. Various test examples are selected for linear static response size, shape, topology, topometry and topography optimizations. Some of the examples are well known as standard problems while some of them are made for this study. The examples cover small, medium and large scale problems. A very large scale problem is defined by a train structure. The quality of the optimum solutions, CPU times, etc. are compared and analyzed.

2 Structural optimization software

2.1 Structural optimization method

An optimization theory is utilized to solve a structural optimization problem formulated as in (1) (Park 2007). As mentioned earlier, the approximation method is generally employed in a commercial structural optimization system. Structural optimization is classified into size, shape, topology, topometry and topography optimizations (MSC Nastran user's guide design sensitivity and optimization version 2013b; Genesis user's manual version 13.1 design reference 2014; Altair OptiStruct user's manual version 13.0 2014). This classification is made based on the characteristics of the design variables (Park 2007). The domain of FE analysis is not changed during the size and topometry optimization processes while the domain of shape and topography optimizations 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.

Size, shape and topology optimizations are considered as classical optimization methods. On the other hand, topometry and topography optimizations are non-classical optimization methods that have been recently developed. Topometry optimization is an element-by-element size optimization (MSC Nastran user's guide design sensitivity and optimization version 2013b). Unlike conventional size optimization where all the elements referencing a property entry are grouped as one design variable, each finite element has an independent design variable in topometry optimization. For example, the thickness of each shell element makes a design variable. Topography optimization (also called bead or stamp

Table 1 Available optimization

	MSC Nastran	Genesis	OptiStruct
Linear static response optimization	O	O	O
Linear transient dynamic response optimization	O	X	X
Nonlinear static response optimization	O	O	O
Nonlinear transient dynamic response optimization	X	O	O
Fatigue response optimization	O	X	O
Linear buckling response optimization	O	O	O
Frequency response optimization	O	O	O
Multi-body dynamics optimization	X	X	O
Heat transfer optimization	X	O	O
Multiple model optimization	O	X	O
Global optimization	O	X	O
Super-element optimization	O	O	X
Parts-element optimization	O	O	X
Aero-elastic optimization	O	X	X

optimization) is used to generate a design proposal for reinforcement bead patterns. In topography optimization, finite element grids move in the direction of the normal vectors to the shell surface or the user’s given direction.

In general, there are many design variables in topology, topometry and topography optimizations. It is well known that the scale of a structural optimization problem is determined by the size of the structure as well as the number of design variables. Therefore, although the structure to be optimized is small, the optimization problem can be a large scale problem in the cases of topology, topometry and topography optimizations. A special optimization algorithm can be used to solve such large scale optimizations.

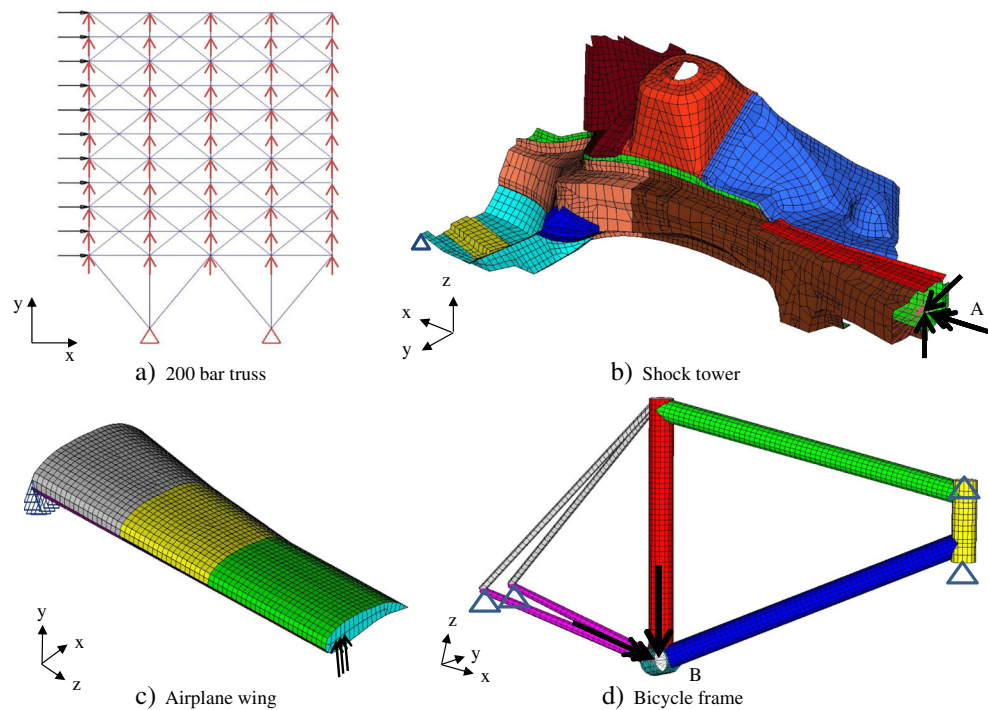
2.2 Optimization algorithm

Optimization algorithms such as the modified method of feasible directions (MMFD) and sequential quadratic programming (SQP) are commonly utilized in MSC Nastran, Genesis and OptiStruct (Arora 2012; Vanderplaats 1999). MMFD is based on the method of feasible directions (MFD). MMFD assumes that optimization begins with a feasible design and maintains feasibility. In SQP, the equality constraints are easily handled and an infeasible starting point can be well dealt with no additional strategy required. MSC Nastran and Genesis also support sequential linear programming (SLP). SLP often converges rapidly to the solution for fully

Table 2 Structural optimization problems

Example	Number of design variables	Objective function	Constraints	Loading conditions	Remark
200 bar truss	96	Mass	Stress, natural frequency and displacement constraints	Multiple loading conditions	Size optimization
Shock tower	15	Mass	Stress and natural frequency constraints	Single loading condition	Size optimization
Airplane wing	11	Mass	Stress, buckling and failure index constraints	Single loading condition	Size optimization (Composite)
Bicycle frame	5	Volume	Displacement constraint	Single loading condition	Size optimization (Composite)
Crank	9	Mass	Stress and natural frequency constraints	Single loading condition	Shape optimization
Rail joint	4	Mass	Stress constraint	Single loading condition	Shape optimization
Car frame	15237	-Torsional stiffness	Fraction mass constraint	Single loading condition	Topology optimization
Engine mount	57280	Compliance	Fraction mass constraint	Multiple loading conditions	Topology optimization
Plate	3200	Compliance	Volume constraint	Single loading condition	Topometry optimization
Airplane wing	21090	Mass	Displacement constraint	Multiple loading conditions	Topometry optimization (Composite)
Latch	13818	Compliance	.	Single loading condition	Topography optimization
Car hood	2069	Compliance	.	Single loading condition	Topography optimization

Fig. 1 Examples of size optimization



constrained problems whose true optimum lies on a vertex in the design space. The commercial structural optimization systems install the above algorithms.

MSC Nastran, Genesis and OptiStruct has MSCADS (MSC Nastran user's guide design sensitivity and optimization version 2013b), DOT (Genesis user's manual version 13.1 design reference 2014) and CONMIN (Vanderplaats 1973) as an optimizer, respectively. For size and shape optimizations of this study, the optimization algorithm MMFD is commonly utilized and it shows good convergence for small-to-medium scale problems. Meanwhile, each software system provides a separate optimizer for large scale problems such as topology topometry and topography optimizations. IPOPT (MSC Nastran user's guide design sensitivity and optimization version 2013b), BIGDOT (Genesis user's manual version 13.1 design reference 2014; Vanderplaats 2000) and Dual-Optimizer (Fleury 1989) are the optimizers for large scale problems in MSC Nastran, Genesis and OptiStruct, respectively. They can be selected for large scale size and shape optimizations.

IPOPT in MSC Nastran implements an interior point line search filter method that aims to find a local optimal solution for a large scale problem. It is one of the classical barrier methods. BIGDOT in Genesis uses a Sequential Unconstrained Minimization Technique (SUMT) to solve the constrained optimization problem as a sequence of unconstrained optimization sub-problems (Arora 2012). Dual optimizer in OptiStruct solves the explicit sub-problem generated by the Convex Linearization Method (CONLIN). The primary dual problem is replaced by a sequence of approximate quadratic sub-problems with non-negative constraints on the dual variables. For large scale problems, the same optimization algorithm is not used because each software system has its own optimizer.

2.3 Characteristics of the software systems

MSC Nastran was developed by NASA in 1965 (Lim et al. 2009). MSC Nastran, which is commercialized by the MSC Corporation, contributed to the development of technology of

Fig. 2 Examples of shape optimization

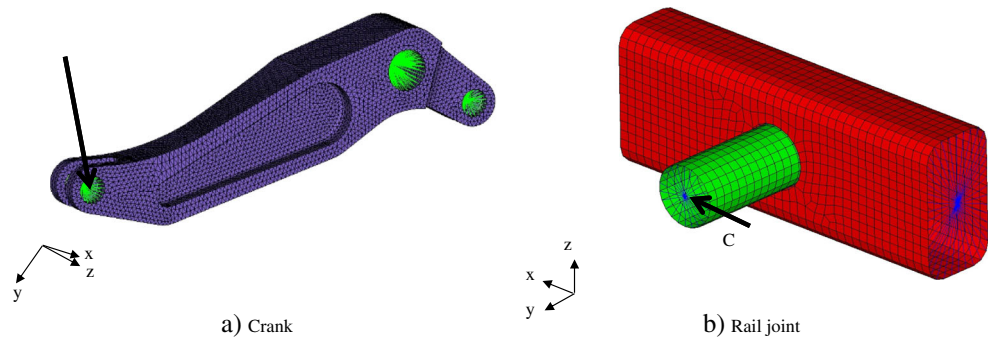
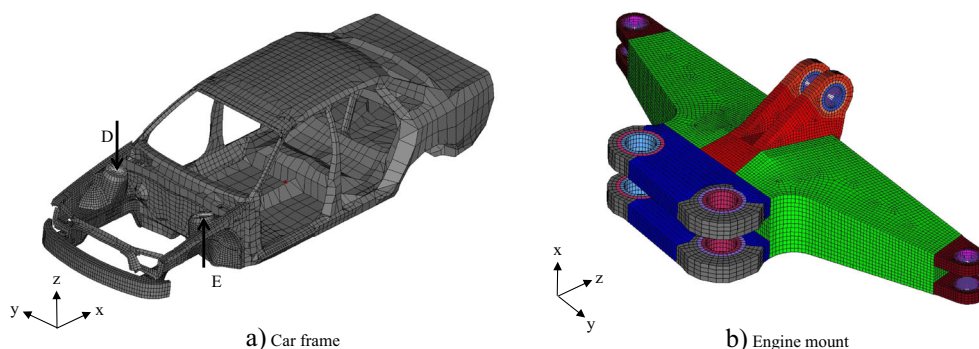


Fig. 3 Examples of topology optimization



finite element analysis in the aerospace field. MSC Nastran provides a variety of FE modules such as static/dynamic analysis of both linear and nonlinear cases. It has a module for linear static response optimization. Genesis was developed in 1984 by Vanderplaats, Miura and Associates (Genesis, <http://www.vrand.com/companyProfile.html>), and it is specialized for structural optimization. OptiStruct was developed by Altair that was founded in 1985 (OptiStruct, <http://www.altairhyperworks.com/Product,19,OptiStruct.aspx>). It is a part of a large software system called HyperWorks (HyperWorks, <http://www.altairhyperworks.com>).

Table 1 shows the characteristics of the software systems. All the software systems can conduct linear/nonlinear static response optimization, linear buckling response optimization, frequency response optimization, optimization for composite materials, etc. It is noted that MSC Nastran has the capability for linear dynamic response optimization using the beta-method (MSC Nastran user’s guide design sensitivity and optimization version 2013b). MSC Nastran does not have the capability for multi-body dynamics optimization and heat transfer optimization while Genesis does not have fatigue response optimization, multi-body dynamics optimization, multiple model optimization, global optimization and aero-elastic optimization. In the case of OptiStruct, it cannot use the super-element technique in optimization, parts-element optimization and aero-elastic optimization.

The techniques, which are common to the three systems, are tested and compared. That is, the test examples are for linear static response structural optimization, and some examples have constraints on linear buckling and frequency responses. As sophisticated nonlinear analysis software systems are well

developed, nonlinear static/dynamic structural optimization draws much attention these days. MSC Nastran can perform only nonlinear static response optimization while the other two systems have the capability for nonlinear static/dynamic response optimization. For such nonlinear response optimization, the three systems commonly use the equivalent static loads method (ESLM) (Choi and Park 2002; Shin et al. 2007; Kim and Park 2010; Park 2011). However, this technique is not a part of a standard package yet; therefore, the capability for nonlinear response optimization is not compared in this study.

The computational time consists of the finite element analysis (FEA) and optimization times. Furthermore, the optimization time is composed of the time for design sensitivity analysis (DSA) and the optimization process. The most of optimization time is spent for design sensitivity analysis (DSA). Genesis and OptiStruct provide the computational time for DSA and the optimization process. Thus, a designer can find the FEA and optimization times. However, MSC Nastran does not provide each computational time in detail. A user can only check the total computational time of the overall process in Nastran. Therefore, only the total computational time that includes FEA, DSA and the optimization process times is calculated in this research.

2.4 Computation environment

Various factors can have influence on the performance of structural optimization software systems. They are the optimization environment (software) and the system environment (hardware). The optimization environment consists of the optimization formulation, the convergence criteria, the utilized

Fig. 4 Examples of topometry optimization

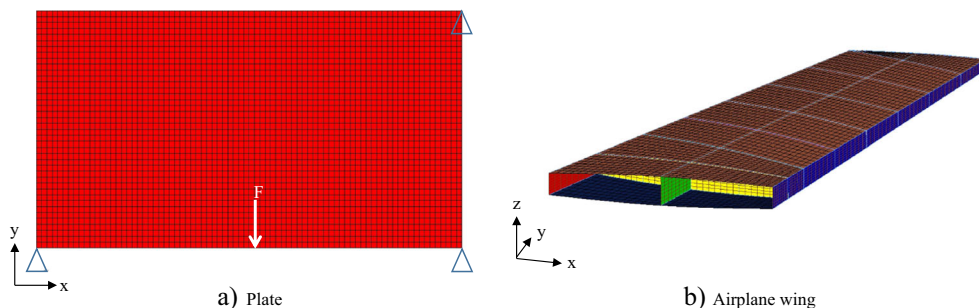
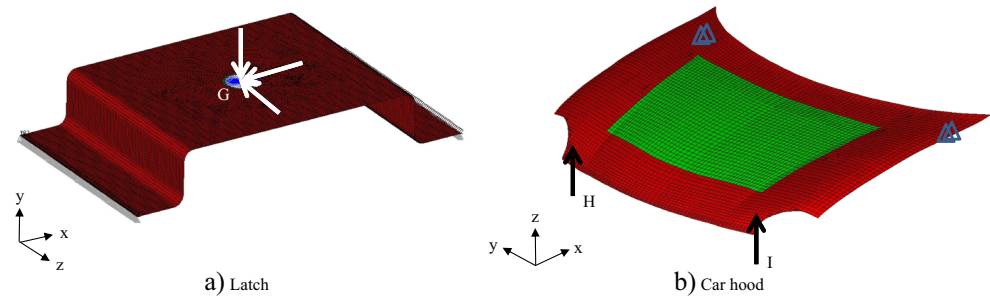
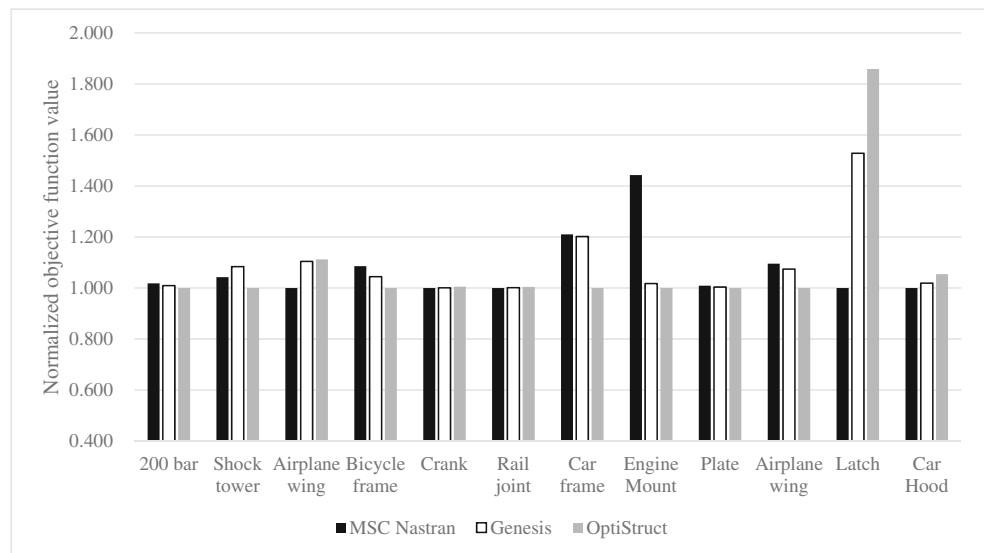


Fig. 5 Examples of topography optimization**Table 3** Results of structural optimization problems

	Example	Software systems	Objective function value		Constraint Violation (%)	Iteration	CPU time (s)	Elapsed time
			Initial	Final [After post-process]				
Size optimization	200 bar truss	MSC Nastran	5.082E+04	1.414E+04	0.296	36	21.886	45.162
		Genesis	5.082E+04	1.417E+04	0.00	23	1.000	2.000
		OptiStruct	5.082E+04	1.390E+04	0.065	46	2.060	19.130
	Shock tower	MSC Nastran	2.514E-02	2.834E-02	0.300	34	76.674	150.150
		Genesis	2.514E-02	2.926E-02	0.00	33	77.000	107.000
		OptiStruct	2.514E-02	2.709E-02	0.018	31	45.330	67.360
	Airplane wing	MSC Nastran	1.756E-02	2.860E-02	0.268	14	188.465	387.740
		Genesis	1.756E-02	2.012E-02	0.00	13	85.000	92.000
		OptiStruct	1.756E-02	1.998E-02	0.00	41	291.800	308.340
Bicycle frame	MSC Nastran	2.605E+06	1.571E+06	0.211	40	49.155	75.114	
	Genesis	2.605E+06	1.504E+06	0.00	17	14.000	19.000	
	OptiStruct	2.605E+06	1.464E+06	0.0119	38	34.800	49.110	
Shape optimization	Crank	MSC Nastran	1.396E-02	1.487E-02	0.064	20	200.227	396.661
		Genesis	1.396E-02	1.483E-02	0.00	23	62.000	73.000
		OptiStruct	1.396E-02	1.490E-02	0.00	6	26.690	31.950
	Rail joint	MSC Nastran	2.411E-03	2.475E-03	0.014	9	13.010	28.626
		Genesis	2.411E-03	2.473E-03	0.00	8	9.000	12.000
		OptiStruct	2.411E-03	2.470E-03	0.00	7	5.440	10.140
Topology optimization	Car frame	MSC Nastran	-3.859E+00	-4.315E+00 [-1.111E+04]	0.00	82	229.945	681.643
		Genesis	-3.837E+00	-2.171E+01 [-1.119E+04]	0.00	42	161.000	197.000
		OptiStruct	-3.975E+00	-9.700E+00 [-1.344E+04]	0.00	114	258.350	296.120
	Engine mount	MSC Nastran	3.560E+06	4.033E+05 [4.986E+05]	0.00	64	2232.000	4329.270
		Genesis	3.561E+06	4.839E+05 [3.517E+05]	0.300	21	1449.000	1869.000
		OptiStruct	3.561E+06	3.455E+05 [3.456E+05]	0.00	82	2304.010	3277.050
Topometry optimization	Plate	MSC Nastran	1.230E+01	1.187E+01	0.00	15	8.080	24.913
		Genesis	1.225E+01	1.179E+01	0.00	11	5.000	6.000
		OptiStruct	2.431E+01	1.175E+01	0.00	19	8.380	25.870
	Airplane wing	MSC Nastran	6.599E+02	3.473E+02	0.240	6	29.452	50.958
		Genesis	6.599E+02	3.405E+02	0.00	9	59.000	69.000
		OptiStruct	9.345E+02	3.169E+02	0.00	12	22.340	28.660
Topography optimization	Latch	MSC Nastran	5.303E+02	1.471E+01	x	32	3121.439	15375.281
		Genesis	5.277E+02	2.248E+01	x	10	50.000	57.000
		OptiStruct	5.302E+02	2.734E+01	x	27	68.840	79.840
	Car hood	MSC Nastran	3.607E+04	2.822E+04	x	8	131.960	311.961
		Genesis	3.605E+04	2.876E+04	x	15	15.000	24.000
		OptiStruct	3.534E+04	2.975E+04	x	7	5.870	10.550

Fig. 6 Normalized objective function values (The smaller value is the better design result.)



optimization method, the move limit strategy, the constraint screening method, etc. The system environment is the capacity of the utilized computer. These environments should be set to have the same value across all the software systems for a fair comparison.

An identical optimization formulation is used for each example problem. The formulation is defined as (1). The convergence criteria of optimization affect the performance. In this research, the relative change of the objective function is used as the convergence criterion with a value of 0.001. The convergence criterion for the objective function is commonly controllable by the three systems. The constraint violation is also used for a convergence criterion. The maximum constraint violation allowed at the converged optimum is set to 0.01 for all the software systems. Because OptiStruct does not support a change for this criterion it is fixed to 0.01 that is a default value

of OptiStruct. All the constraints have the form of the less than type, and they are normalized to have 1.0 for the design bound.

The move limit strategy is considered in the examples. When the approximation method is used, the approximation model is not a precise representation of the original objective and constraint functions. Therefore, move limits of design variables are used to achieve efficient convergence. The move limit can be controlled by changing the initial value, lower and upper bounds in MSC Nastran and Genesis. However, the lower and upper bounds of the move limit cannot be manually controlled in OptiStruct. Therefore, only the initial value of the move limit is the same.

The same constraint screening method is utilized for the large scale problem that is an optimization of a train structure. The train problem will be shown in Section 4. If there are many constraints, the numerical cost is high because sensitivity

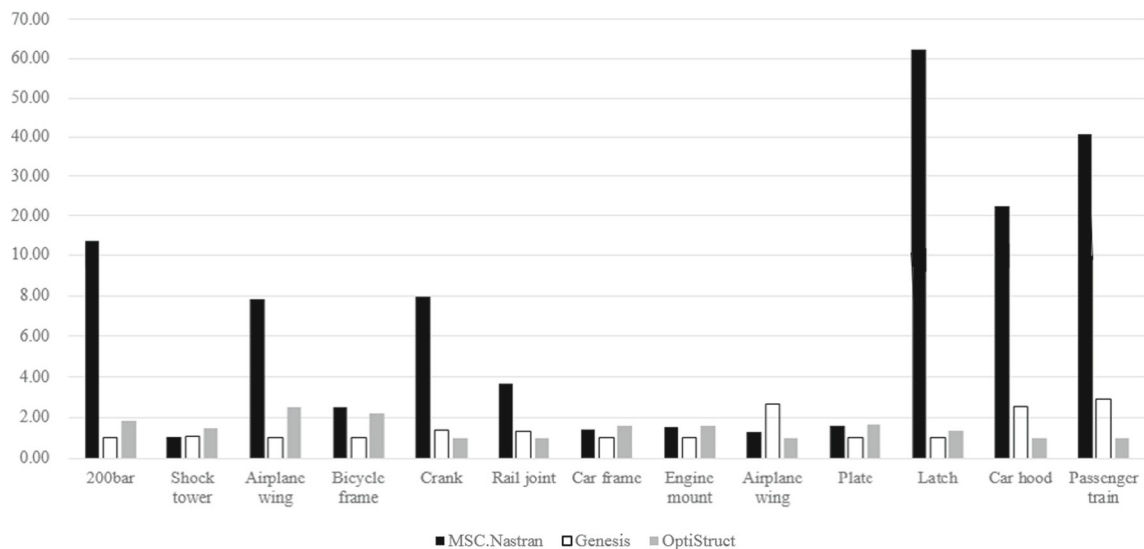
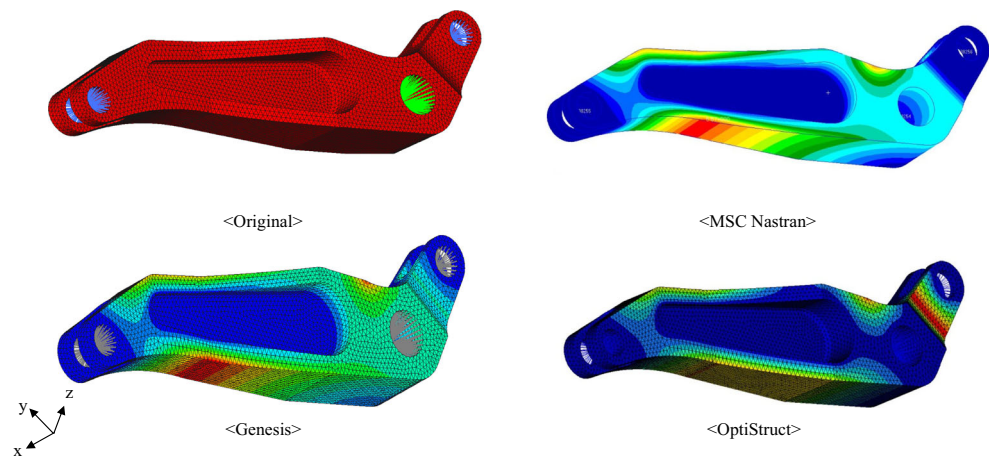


Fig. 7 Normalized calculation time

Fig. 8 Results of the crank example in shape optimization



information of the constraints is required in the optimization process. In general, the active constraints, which should be considered in each iteration (cycle) of the optimization process, are just a part of all constraints. Two steps are conducted to select the active constraints. In the first step, each constraint is normalized to be less than 1, so a constraint is violated if it is greater than 1. The constraints, which are greater than 0.5, are selected as active constraints in this study. In the second step, the active constraints are defined for a certain region of the structure. Based on the design variables, the entire structure is divided into multiple design regions that are called ‘components’ in the software systems. Out of the active constraints selected in the first step, the top 20 constraints per component are chosen for the final active constraints. Therefore, the performance of the optimization can be improved by properly adjusting the number of constraints to consider. The constraint screening method is not utilized for the other examples.

The system environment is determined by the capacity of the computer such as the operating system, the CPU, the amount of memory usage, etc. The system environments are also set to the same ones across all the software systems. The utilized operating system is MS Windows ×64 Ultimate (version 6.1, Build 7601) that is commonly supported by the three software systems. The

utilized hardware system has 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 of 8GB is commonly used. However, it is extended to 64GB in the case of the large scale problem. Some conditions are controllable in one software system but uncontrollable in another system. In this case, the same conditions are used as much as possible. When it is not possible to use the same conditions, default values of the software systems are used.

3 Small-to-medium scale structural optimization examples

3.1 Structural optimization examples

Various structural optimization examples are solved for comparison of the performances. Examples of size optimization are a 200 bar truss (Haug and Arora 1979), a shock tower (MSC Nastran user’s guide automated structural optimization in MSC Nastran version 2014c), an airplane wing (Altair

Fig. 9 Results of the rail joint example in shape optimization

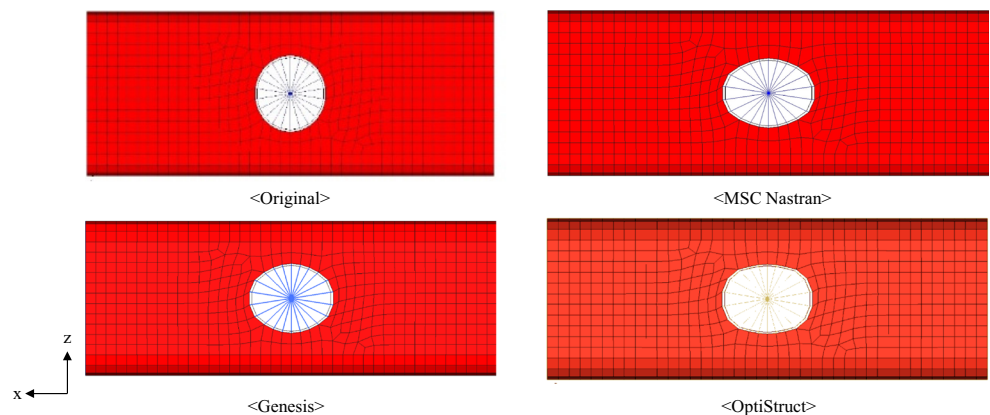
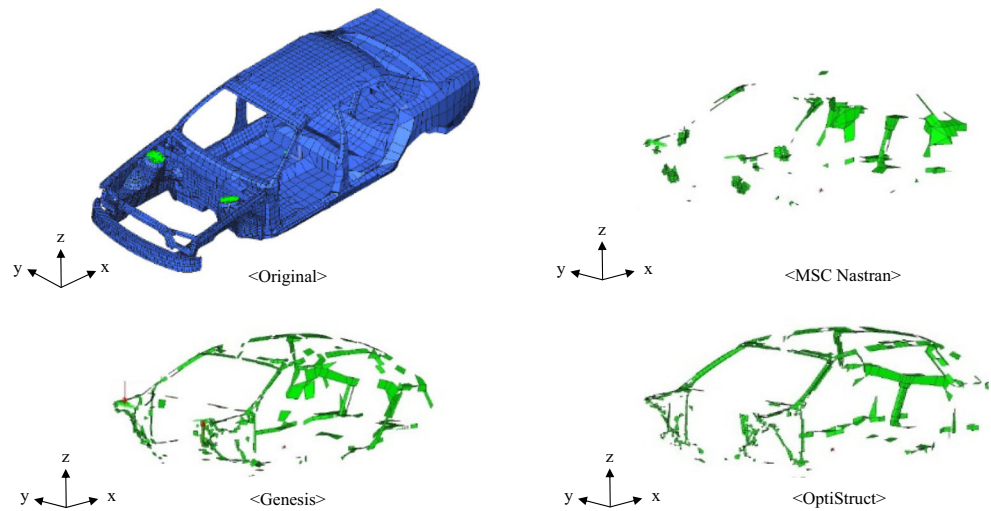


Fig. 10 Results of the car frame example in topology optimization



HyperStudy tutorials version 13.0 2014) and a bicycle frame (Genesis user’s manual version 13.1 design reference 2014). Examples of shape optimization are a crank (Altair HyperStudy tutorials version 13.0 2014) and a rail joint (Altair HyperStudy tutorials version 13.0 2014). Examples of topology optimization are a car frame (Genesis user’s manual version 13.1 design reference 2014) and an engine mount (MSC Nastran user’s guide design sensitivity and optimization version 2013b). Examples of topometry optimization are a plate (MSC Nastran user’s guide automated structural optimization in MSC Nastran version 2014c) and an airplane wing (Altair HyperStudy tutorials version 13.0 2014). Finally, examples of topography optimization are a latch (MSC Nastran user’s guide automated structural optimization in MSC Nastran version 2014c) and a car hood (Genesis user’s manual version 13.1 design reference 2014). The detailed optimization formulation of each example is in the above references. The problems for topology, topometry and topography optimizations can be considered as large scale problems because

there are many design variables. Such examples are included in this section. The characteristics of the examples are shown in Table 2.

The examples of size optimization are illustrated in Fig. 1. The 200 bar truss in Fig. 1a) has multiple loading conditions. The first loading condition is that 4.45 kN force is applied at node no. 1, 6, 15, 20, 29, 34, 43, 48, 57, 62 and 71 in the positive x direction. The second loading condition is that 44.48 kN force is applied at node no. 1, 2, 3, 4, 5, 8, 10, 12, 14, 15, 16, 17, 18, 19, 20, 22, 24, ..., 71, 72, 73, 74 and 75 in the positive y direction. The third loading condition is that loading conditions 1 and 2 are acting together. The shock tower in Fig. 1b) has a single loading condition. 1 kN force acts in the positive y and z directions and 10 kN-mm moment in the positive x direction at node A. The number of finite elements is 6,081. The airplane wing in Fig. 1c) has a single loading condition. 1.724 kPa pressure acts in the positive y direction of the bottom elements and 4.4452E+3 kN force acts in the positive y direction at node no. 7473,

Fig. 11 Results of the engine mount example in topology optimization

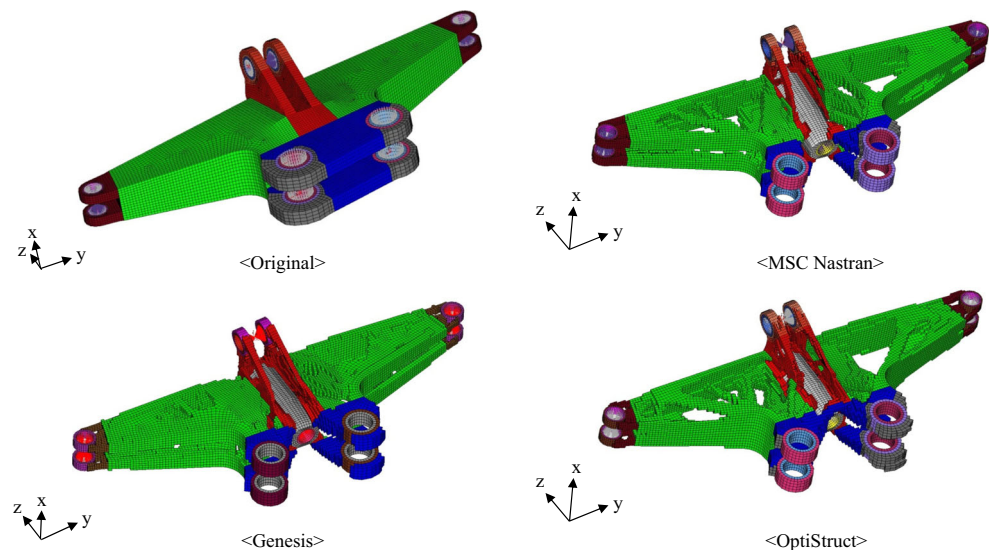
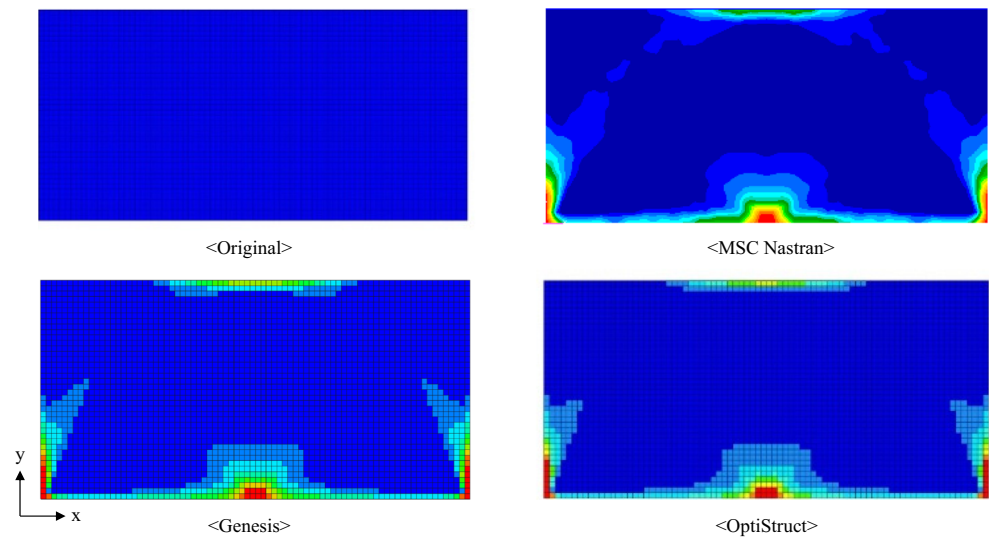


Fig. 12 Results of the plate example in topometry optimization



7474, 7475 and 7476. The number of finite elements is 2,899. This example consists of composite materials. The bicycle frame in Fig. 1d) has a single loading condition. 100 kN force acts in the negative z direction and 100 kN-mm moment acts in the positive x direction at node B. The number of finite elements is 3,935. Also composite materials are utilized.

The examples for shape optimization are illustrated in Fig. 2. The crank in Fig. 2a) has a single loading condition of $1.27E+5$ N. The number of finite elements is 56,552. This example has nine perturbation vectors. The finite elements consist of solid elements. The rail joint in Fig. 2b) has a single loading condition. 10 kN force acts in the positive x direction at node C. The number of finite elements is 2,064. This example has four perturbation vectors.

The examples of topology optimization are illustrated in Fig. 3. The car frame in Fig. 3a) has a single loading condition.

265 N negative force acts at node D and 265 N positive force acts at node E in the z direction. The number of finite elements is 15,237. The engine mount in Fig. 3b) has multiple loading conditions. The number of finite elements is 63,045 and the utilized elements are solid elements.

The examples of topometry optimization are illustrated in Fig. 4. The plate in Fig. 4a) has a single loading condition. 10 kN acts in the negative y direction at node F. The number of finite elements is 3,200. The airplane wing in Fig. 4b) has multiple loading conditions. The first loading condition is the maximum torque, and the second loading condition is the minimum torque. The number of finite elements is 21,090 and composite materials are utilized.

The examples of the topography optimization are illustrated in Fig. 5. The latch in Fig. 5a) has a single loading condition. 100 N force acts in the negative x , y and z directions at node G. The number of finite elements is 13,818. The car hood in

Fig. 13 Results of the latch example in topography optimization

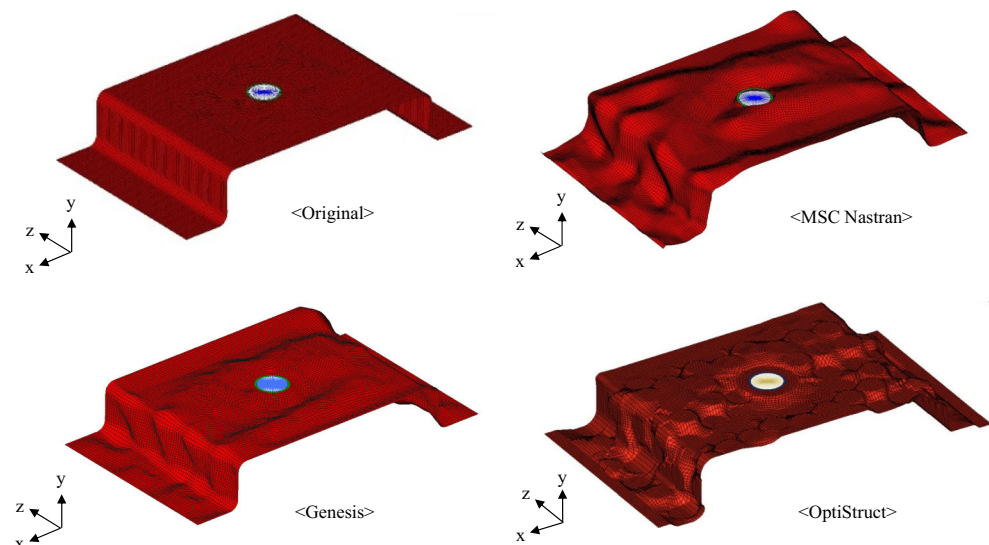


Fig. 14 Results of the car hood example in topography optimization

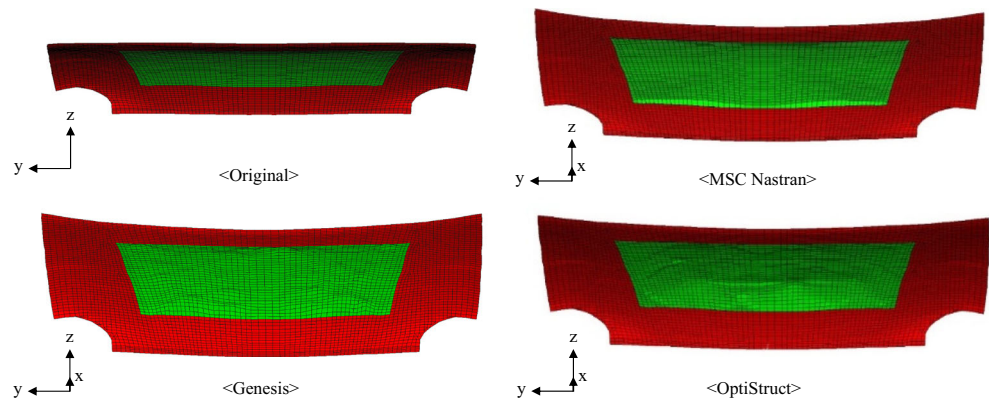


Fig. 5b) has a single loading condition. 100 N force acts in the positive z direction at nodes H and I. The number of finite elements is 5,072.

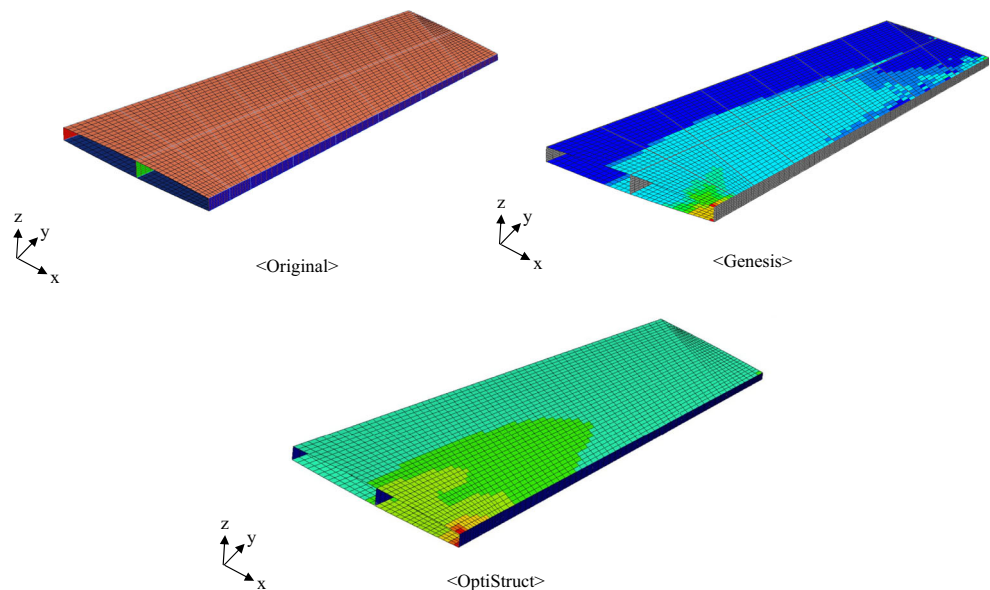
3.2 Results of structural optimization examples

The optimization results of each example are summarized in Table 3 including the objective function, constraint violation, the number of iterations, the CPU time and the elapsed time. As mentioned earlier, since the CPU times for each process such as FEA and optimization times is not provided by MSC Nastran, only the total CPU time is shown in Table 3. If the maximum violation is less than 0.01 (1 %), it is considered to be satisfied. It is noted that Genesis marks the maximum violation with 0.0 when it is less than 0.01. Figures 6 and 7 illustrate the normalized objective function and CPU, respectively. The values are normalized by the best one that is the smallest. In the case of size and shape optimizations, the three systems show similar objective function values. OptiStruct is slightly better than others in size optimization and MSC Nastran is a little better in shape optimization. However,

MSC Nastran spends a lot of CPU time for some examples. It is noted that optimum shapes are different in the three systems as illustrated in Figs. 8 and 9.

The density method is employed in topology optimization. It is difficult to make a precise comparison for the results of topology optimization. When the optimization process ends with a constraint on the volume constraints, the total volumes are the same but the design variables (densities) are quite different. That is, many elements have the values far from 0 or 1. To overcome this difficulty, the post process can be adopted to select the elements. In the post process, the density filtering method is employed. That is, the rankings of elements are made by the optimum density values and the elements of the upper rankings are remained. The number of remaining elements is the same as the volume constraints. Since there are multiple elements with the same ranking, the number of remained elements can be varied in each software system and precise comparison may not be able to be made with the results of topology optimization. The topology results in Figs. 10 and 11 are the ones after the post processes. The results of the three systems are very different because the post

Fig. 15 Results of the airplane wing example in topometry optimization



process can give quite different results. If we make a rough comparison with the results in Table 3, OptiStruct gives a better optimum after the post process and Genesis spends less CPU time.

In topometry optimization, the optimum objective functions are similar. OptiStruct gives a slightly better objective function while Genesis spends the least CPU time. This trend is similar to that in topology optimization. It is noted that the initial design is not controllable in OptiStruct, and each initial design variable is automatically determined by the lower and upper bounds. Figure 12 illustrates the optimization results of the plate example. The optimum results of MSC Nastran and Genesis are similar while OptiStruct gives slightly different results. We could not find a certain trend in the CPU time.

In topography optimization, MSC Nastran gives a little better optimum value; however, it needs a lot more CPU time as shown in Table 3. As illustrated in Fig. 13, the final results of MSC Nastran and Genesis are similar and the results of OptiStruct are very different in the case of the latch example. The three systems give different results of the car hood as illustrated in Fig. 14.

For the optimum objective function, not one out of the three systems is absolutely better than the other two systems. However, the required CPU times are quite different. Although the structure to be optimized is small, the examples for topology, topometry and topography optimizations can be considered as large scale problems because they have many design variables. The examples in this section are not very large in a general sense. A large scale problem is solved in Section 4.

3.3 Some issues of the examples and the software systems

In the case of size optimization with composite materials, MSC Nastran and Genesis do not update the neutral plane for the stacked layers. The neutral plane is updated by a formula provided by the user in MSC Nastran and Genesis, while it is automatically updated in OptiStruct. In topometry optimization with composite materials, the neutral plane is updated based on the given functions in Genesis. However, the neutral plane cannot be updated in MSC Nastran. Thus, the results of MSC Nastran may be incorrect. In Fig. 15, the results of Genesis and OptiStruct are similar but it is not possible to show the contour from MSC Nastran due to the limit of Patran ver. 2014 (Patran user's guide version 2014) that is used for the graphic user interface for MSC Nastran.

The car frame (Genesis user's manual version 13.1 design reference 2014) in Fig. 10 originally consists of two parts. Two car frame parts are duplicated at the same location. One is the main car frame that is not included in the design domain, and the other car frame, which makes the design domain, is for the reinforcement. Since OptiStruct does not normally optimize a structure that has two duplicated parts in topology optimization, only the reinforced part is optimized and the

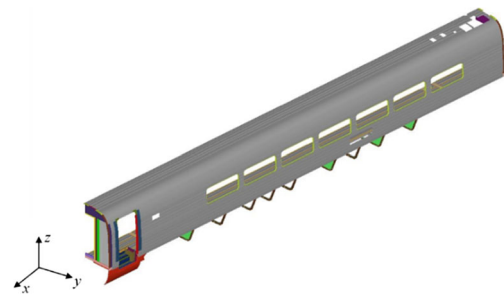


Fig. 16 Finite element model of the passenger train

results of topology optimization are added to the original part. Also, the option to prevent the checkerboard problem (Diaz and Sigmund 1995) is used. The minimum member size is defined in topology optimization (Zhou et al. 2001) so that the width of the left part should be greater than a certain value. OptiStruct recommends the usage of the minimum-member size. Therefore, the option for the minimum-member size is utilized for all the examples. MSC Nastran and OptiStruct give a raw strain energy or compliance. That is, the strain energy is evaluated for a rugged boundary. It is noted that Genesis can also calculate the strain energy for a smooth curve. The rugged boundary is used for the calculation of the strain energy to have the same conditions.

4 Large scale structural optimization example

4.1 Structural optimization of a passenger train

The performance of a software system is more important when an FE model becomes larger. The scale of an optimization problem is determined by the size of the FE model, the number of design variables and the number of the design constraints. If the constraint screening is employed, the number of considered constraints is significantly reduced from the original problem. Thus, the number of constraints is not generally considered to judge the scale. As a large scale problem, the structure of a passenger train is optimized (Lee et al. 2015). As illustrated in Fig. 16, the width, the height and the length of the train structure are 1.5, 3.0 and 23.5 m, respectively. The passenger train model consists of shell and solid elements.

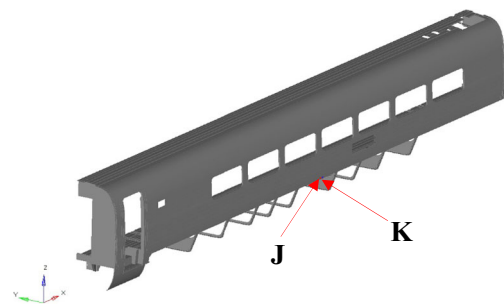


Fig. 17 Location of initial displacement nodes in the passenger train

Table 4 Results of the large scale example

	Example	Software systems	Objective function value		Constraint Violation (%)	Iteration	CPU time	Elapsed time
			Initial	Final				
			Size & Shape optimization	Passenger train				
		Genesis	4.2739	3.4972	0.00	11	1 h. 16 min.	3 h. 28 min.
		OptiStruct	4.2739	3.4449	0.241	9	26 min.	46 min.

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.

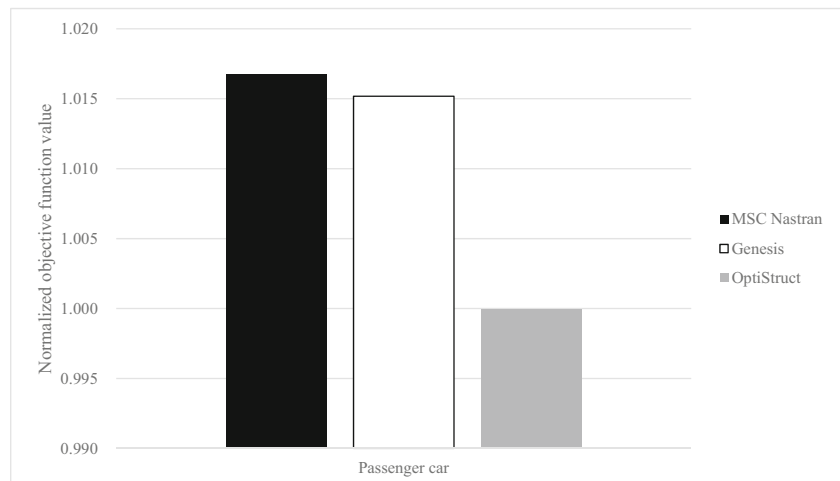
The design formulation is as follows:

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

where b_i is the i th size variable, b_j is j th the shape variable, and b_{lower} and b_{upper} are the lower bound and the upper bound of the design variable, respectively. $\sigma_{\text{von Mises}}$ is the von Mises stress, σ_{allow} is the allowable stress, $\delta_{z, \text{initial}}$ is the initial displacement in the z axis at nodes J and K in Fig. 17 and $\delta_{z, \text{reference}}$ is the displacement of the reference model. The objective function to be minimized is the mass of the structure while the displacement and stress constraints are satisfied.

For this example, OptiStruct has a problem in memory usage. 8GB of main memory is sufficient for MSC Nastran and Genesis even without the constraint screening strategy. However, the amount of the memory usage should be extended to 64GB for OptiStruct. Moreover, the constraint screening should be used in OptiStruct even with 64GB of memory. Thus, OptiStruct should be improved to use less memory.

Fig. 18 Normalized objective function (mass) of the passenger train



4.2 Results of the large scale example

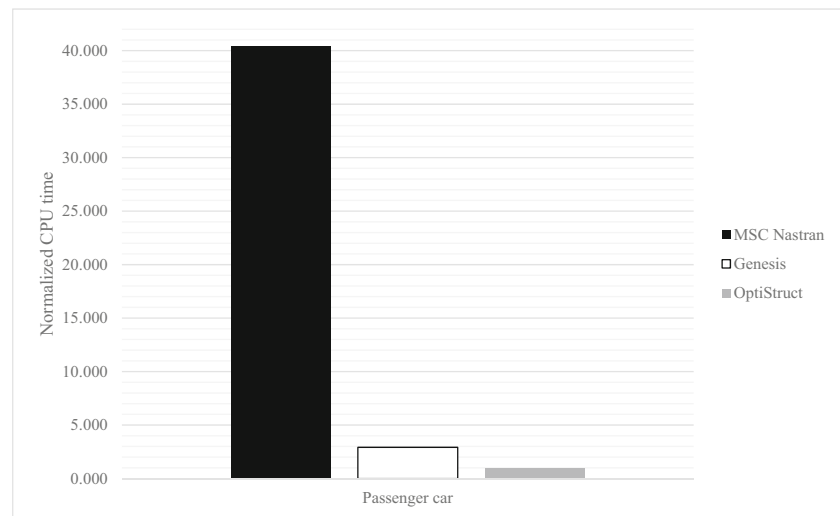
The optimization results are shown in Table 4. They are the same as those of Table 3. In this problem, the elapsed time is larger than the computational time because there are many other processes in addition to the pure optimization process. As mentioned earlier, the elapsed time includes the time for checking the license file, reading input files writing output files, etc. The elapsed time is also important because designers can finally obtain the optimization results after the elapsed time.

Figures 18 and 19 illustrate the normalized objective function and calculation time, respectively. As depicted in the figures, the normalized optima are similar; however, the calculation time of MSC Nastran is at least 14 times longer than those of other software systems. MSC Nastran should improve the CPU time in the case of a large scale problem.

5 Conclusions

Linear static response optimizations are explored, and three commercial structural optimization systems are compared using various structural examples. We could see some differences in performance for the three software systems. No system is the best in all the cases and each system has advantages

Fig. 19 Normalized calculation time in the passenger train



and disadvantages depending on the application. It should be noted that the results of this research are based on the selected examples. If different examples are solved, different conclusions can be obtained. Moreover, it is not possible to make a conclusion regarding the influence of a parameter or a method in the optimization process, because they operate in a complex manner with the characteristics of an example.

For small-to-medium scale examples, there are no big differences in the optimum results. In general, Genesis is excellent in the computational time while OptiStruct provides excellent optimum solutions, especially in size, topometry and topology optimizations. MSC Nastran presents good solutions in shape and topography optimizations, however; MSC Nastran requires a long computational time in topography optimization. In the case of a large scale example, the three systems give similar objective function values. OptiStruct is excellent in computational time; however, it has a memory control issue that is not present in Genesis and MSC Nastran. Especially in large scale problems, the elapsed time of MSC Nastran is very long compared to other software systems.

There can be various reasons for performance distinction, because performance is determined by a combination of many factors. For example, the method for approximation is different in each of the three systems. The move limit strategy is also slightly different. All the systems allow choosing an initial value of the move limit, but the size of the move limit is changed as the optimization process proceeds. Some parameters can be controllable in one system but uncontrollable in another system. This aspect should be more thoroughly investigated.

It is difficult for a user to figure out how exactly an optimization method would influence the performance. The software developers should provide more information about it. It is noted that the three systems do not fail in any examples. That means that all of them are quite reliable. Within one

software system, different results can be obtained when different values are selected for parameters. The influence of these parameters should be more clearly explained by the software developers. The authors hope that this paper helps practitioners choose a structural optimization system. Currently, comparison of linear static response structural optimization is performed.

This paper compares structural optimization performance for linear static response. The future work will be extended to structural optimization with linear/nonlinear static/dynamic response: all considered software systems have the capability to do that via the equivalent static loads method.

The authors hope that this paper helps practitioners to choose the most appropriate structural optimization software system for them.

Acknowledgments This research was supported by the Guangdong Provincial Natural Science Foundation (2015A030312008) and Guangdong Provincial Science and Technology Plan (2015B010104006). The authors are thankful to Mrs. MiSun Park for the English correction of the manuscript.

References

- Altair HyperStudy tutorials version 13.0 (2014) Altair Engineering, Inc., MI, USA
- Altair OptiStruct user's manual version 13.0 (2014) Altair Engineering, Inc., MI, USA
- ANSYS <http://www.ansys.com/About+ANSYS> ANSYS, Inc
- Arora JS (2012) Introduction to optimum design. Elsevier, Waltham
- Bathe KJ (1996) Finite element procedures in engineering analysis. Prentice Hall, Englewood Cliffs
- Bendsøe MP, Sigmund O (2003) Topology optimization theory, methods and applications Springer, Germany
- Choi WS, Park GJ (2002) Structural optimization using equivalent static loads at all the time intervals. *Comput Methods Appl Mech Eng* 191:2105–2122

- Cook R, Malkus D, Plesha M, Witt R (2001) Concepts and applications of finite element analysis. Wiley, USA
- Diaz AR, Sigmund O (1995) Checkerboard patterns in layout optimization. *Struct Optim* 10:40–45
- Fleury C (1989) CONLIN: an efficient dual optimizer based on convex approximation concepts. Mechanical, aerospace and nuclear engineering department. University of California, USA
- Fleury C, Braibant V (1986) A new dual method using mixed variables. *Int J Numer Methods Eng* 23(3):409–428
- Genesis <http://www.vrand.com/companyProfile.html> Vanderplaats Research and Development, Inc
- Genesis user's manual version 13.1 design reference (2014) Vanderplaats Research and Development. Inc. Colorado Springs, USA
- Haftka RT, Gürdal Z (1992) Elements of structural optimization. Kluwer, Dordrecht
- Haug EJ, Arora JS (1979) Applied optimal design. NY, USA
- Hong UP, Hwang KH, Park GJ (2004) A comparative study of software systems from the optimization viewpoint. *Struct Multidiscip Optim* 27(6):460–468
- HyperWorks <http://www.altairhyperworks.com/> Altair Engineering, Inc
- Kim YI, Park GJ (2010) Nonlinear dynamic response structural optimization using equivalent static loads. *Comput Methods Appl Mech Eng* 199:660–676
- Lee HA, Jung SB, Jang HH, Shin DH, Lee JW, Kim KW, Park GJ (2015) Structural-optimization-based design process for the body of a railway vehicle made from extruded aluminium panels. *Proc IMechE Part F: J Rail Rapid Transit*. doi:10.1177/0954409715593971
- Lim JH, Kim KW, Kim SW, Hwang DS (2009) Technology trends on structural analysis software in aerospace industry. *Current Industrial Tech Trends Aerospace* 7(2):59–67
- Logan DL (1993) A first course in the finite element method. PWS publishing company, Boston
- MSC Nastran user's guide automated structural optimization in MSC Nastran version 2014 (2014c) MSC Software Co., Newport Beach, CA, USA
- MSC Nastran user's guide design sensitivity and optimization version 2013 (2013b) MSC Software Co., Santa Ana, CA, USA
- MSC Nastran user's guide version 2013.1.1 (2013a) MSC Software Co., Newport Beach, CA, USA
- OptiStruct <http://www.altairhyperworks.com/Product,19,OptiStruct.aspx> Altair Engineering, Inc
- Park GJ (2007) Analytical methods in design practice. Springer, Germany
- Park GJ (2011) Technical overview of the equivalent static loads method for non-linear static response structural optimization. *Struct Multidiscip Optim* 43:319–337
- Park YS, Lee SH, Park GJ (1995) A study of direct vs. approximation methods in structural optimization. *Structural Optimization* 10: 64–66
- Patran user's guide version 2014 (2014) MSC Software Co., Newport Beach, CA, USA
- Schmit LA (1960) Structural design by systematic synthesis. Proceedings of the second ASCE conference, NY, 105–122
- Shin MK, Park KJ, Park GJ (2007) Optimization of structures with non-linear behavior using equivalent loads. *Comput Methods Appl Mech Eng* 196:1154–1167
- Tosca <http://www.3ds.com/products-services/simulia/products/tosca/> Dassault systems
- Vanderplaats GN (1973) CONMIN – a Fortran program for constrained function minimization: user's manual. NASA TM X-62282
- Vanderplaats GN (1999) Numerical optimization techniques for engineering design. Vanderplaats Research and Development. Inc, Colorado Springs
- Vanderplaats GN (2000) Very large scale optimization. The eighth AIAA/USAF/NASA/ISSMO symposium at multidisciplinary analysis and optimization, Long Beach, CA, USA
- Zhou M, Shyy YK, Thomas HL (2001) Checkerboard and minimum member size control in topology optimization. *Struct Multidiscip Optim* 21:152–158