

A globally convergent sequential convex programming using an enhanced two-point diagonal quadratic approximation for structural optimization

Seonho Park · Seung Hyun Jeong · Gil Ho Yoon ·
Albert A. Groenwold · Dong-Hoon Choi

Received: 1 May 2013 / Revised: 6 February 2014 / Accepted: 18 March 2014 / Published online: 1 June 2014
© Springer-Verlag Berlin Heidelberg 2014

Abstract In this study, we propose a sequential convex programming (SCP) method that uses an enhanced two-point diagonal quadratic approximation (eTDQA) to generate diagonal Hessian terms of approximate functions. In addition, we use nonlinear programming (NLP) filtering, conservatism, and trust region reduction to enforce global convergence. By using the diagonal Hessian terms of a highly accurate two-point approximation, eTDQA, the efficiency of SCP can be improved. Moreover, by using an appropriate procedure using NLP filtering, conservatism, and trust region reduction, the convergence can be improved without worsening the efficiency. To investigate the performance of the proposed algorithm, several benchmark

numerical examples and a structural topology optimization problem are solved. Numerical tests show that the proposed algorithm is generally more efficient than competing algorithms. In particular, in the case of the topology optimization problem of minimizing compliance subject to a volume constraint with a penalization parameter of three, the proposed algorithm is found to converge well to the optimum solution while the other algorithms tested do not converge in the maximum number of iterations specified.

Keywords Sequential convex programming (SCP) · Diagonal quadratic approximation (DQA) · Filter method · Conservatism · Enhanced two-point diagonal quadratic approximation (eTDQA)

This paper is based on previous papers entitled “A new convex separable approximation based on two-point diagonal quadratic approximation for large-scale structural design optimization,” presented at the 9th World Congress on Structural and Multidisciplinary Optimization, 13–17 June 2011, Shizuoka, Japan, and “A filtered sequential approximate optimization algorithm based on dual subproblems using an enhanced two-point diagonal quadratic approximation for structural optimization,” presented at the 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 17–19 September 2012, Indianapolis, Indiana, USA.

S. Park · S. H. Jeong
Graduate School of Mechanical Engineering, Hanyang University,
Seoul, Republic of Korea

G. H. Yoon · D.-H. Choi (✉)
School of Mechanical Engineering, Hanyang University,
Seoul, Republic of Korea
e-mail: dhchoi@hanyang.ac.kr

A. A. Groenwold
Department of Mechanical Engineering, University of
Stellenbosch, Stellenbosch South Africa

1 Introduction

Sequential approximate optimization (SAO) can be used to solve nonlinear optimization problems when the objective and/or constraint functions involved are costly to evaluate (Bruyneel et al. 2002; Fleury and Braibant 1986). Let us consider a nonlinear optimization problem with inequality constraints expressed as follows:

$$\begin{aligned} & \text{minimize } f_0(\mathbf{x}) \\ & \text{subject to } f_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, m \\ & \text{and } x_i \in [x_{i,L}, x_{i,U}], \quad i = 1, 2, \dots, n, \end{aligned} \quad (1)$$

where the objective function and constraint functions are denoted by $f_0(\mathbf{x})$ and $f_j(\mathbf{x})$, $j = 1, 2, \dots, m$, respectively. The number of constraints and design variables in the range of $x_{i,L}$ to $x_{i,U}$, $i = 1, 2, \dots, n$ are m and n , respectively. To reduce the computational cost of solving this optimization problem, a primal approximate subproblem is

sequentially constructed by using approximate functions at the k th iteration as follows:

$$\begin{aligned} & \text{minimize} \quad \tilde{f}_0^{(k)}(\mathbf{x}) \\ & \text{subject to} \quad \tilde{f}_j^{(k)}(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, m \\ & \text{and} \quad x_i \in [x_{i,L}^{(k)}, x_{i,U}^{(k)}], \quad i = 1, 2, \dots, n, \end{aligned} \quad (2)$$

where $\tilde{f}_j^{(k)}(x)$ ($j = 0, 1, \dots, m$) is the j th approximate function at the k th iteration. The i th design variable at the k th iteration ranges from $x_{i,L}^{(k)}$ to $x_{i,U}^{(k)}$.

Usually, for structural optimization problems, the function value of $f_j(\mathbf{x})$ in (1) is obtained by the finite element method (FEM), and the derivative is obtained by a design sensitivity analysis method such as the adjoint variable method. These function and design sensitivity values can be used to construct the approximate functions in (2). To efficiently find the optimal solution, it is very important to take into account the characteristics of the optimization problem when selecting an appropriate method for constructing the approximate functions. Structural optimization problems usually have more design variables than constraints. For such problems, SAO using duality is known to be a good methodology because the dimensionality of the dual space is much smaller than that of the primal design space (Fleury 1989; Svanberg 1987; Zillober 2001; Zillober et al. 2004). This methodology is especially useful if approximate functions are convex and separable, in which case the relationship between primal and dual variables can be explicitly expressed and a dual approximate subproblem can be effectively constructed by employing a convex separable approximation (CSA):

$$\begin{aligned} & \text{maximize} \quad \tilde{\gamma}^{(k)}(\lambda) = \tilde{f}_0^{(k)}(\mathbf{x}(\lambda)) + \sum_{j=1}^m \lambda_j \tilde{f}_j^{(k)}(\mathbf{x}(\lambda)) \\ & \text{subject to} \quad \lambda_j \geq 0, \quad j = 1, 2, \dots, m, \end{aligned} \quad (3)$$

where $\tilde{\gamma}^{(k)}(\lambda)$ is an approximate dual function at the k th iteration formulated by dual variables λ_j , $j = 1, 2, \dots, m$. $\mathbf{x}(\lambda)$ denotes a primal variable vector explicitly expressed in terms of dual variables using the CSA. Using SAO to sequentially solve (3) to obtain the optimum solution is called sequential convex programming (SCP).

The performance of SCP depends strongly on the accuracy of the CSA's diagonal Hessian terms. As the accuracy of the diagonal Hessian terms increases, this decreases the number of function evaluation and sensitivity analysis calls required for the SCP to converge to the true optimum point. Therefore, many methods have been proposed to approximate the diagonal Hessian terms accurately. The methods usually use either reciprocal-like intervening variables or exponential intervening variables. For structural optimization, reciprocal-like intervening variables have usually been used because of the reciprocal relationship between design

variables and structural responses (Fadel et al. 1990; Fleury 1989; Svanberg 1987). Popular algorithms using reciprocal-like intervening variables are convex linearization (CONLIN) (Fleury 1989) and the method of moving asymptotes (MMA) (Svanberg 1987). CONLIN uses a conservative approximation that adopts a hybrid form of the linear and the reciprocal approximations (Fleury 1989). The MMA also uses the reciprocal-like approximation, but adopts the concept of moving asymptotes to allow variable curvatures in the approximate functions (Svanberg 1987).

The aforementioned methods are all one-point approximation methods. To make full use of the known information at two consecutive points generated in the process of SAO, Fadel et al. (1990) proposed the first two-point approximation method: a two-point exponential approximation (TPEA) that employed exponential intervening variables. This method can provide curvatures of approximate functions over a range wider than that of the reciprocal approximation. Based on TPEA, a series of improved two-point approximations were developed (Chickermane and Gea 1996; Duysinx et al. 2001; Zhang and Fleury 1997); as one of these approximations, two-point adaptive nonlinearity approximations (TANAs) enhanced approximation quality by matching not only the gradient values but also the function value of an approximate function at the previous point (Wang and Grandhi 1994, 1995; Xu and Grandhi 1998). However, TPEA and TANAs have difficulty in deriving the analytical relationship between primal and dual variables (Groenwold et al. 2007, 2010). Xu et al. (2000) have suggested linear combination of linear and reciprocal variables with coefficients to utilize two consecutive design points. Also, there are other approaches using multi-point information for approximation (Wang and Grandhi 1996; Wang et al. 1996).

To overcome this difficulty, diagonal quadratic approximation (DQA) has been used for structural optimization. Because the DQA is separable and can become the convex separable approximation (CSA) with just enforcing strict convexity, many researchers have studied the DQA for decades. Zhang and Fleury (1997) suggested a "two-point fitting scheme" which decided diagonal Hessian terms to fit the function values of two consecutive points. Duysinx et al. (2001) suggested a quasi-Cauchy update of diagonal Hessian terms of DQA. In 2010, Groenwold, Etman, and Wood used the notion of "approximated approximations" and presented formulations for obtaining the diagonal Hessian terms of DQA based on TPEA and TANA-3.

In this study, we propose a SCP that improves on TPEA and TANA-3 by introducing a better two-point approximation method, including a new scheme to ensure that approximate functions are convex, and a new method for enforcing global convergence. The former is discussed in this paragraph and the latter in the next paragraph. The two-point

approximation method introduced in the proposed SCP is the enhanced two-point diagonal quadratic approximation (eTDQA) proposed by Kim and Choi (2008), which they demonstrated to generate accurate approximation results whether the derivatives at two consecutive design points have the same sign or not and regardless of the move limit strategy. How to generate highly accurate diagonal Hessian terms of DQAs using eTDQA will be described in detail in Section 2.1. Because the off-diagonal terms of the Hessian matrix of eTDQA turned out to be all zeros, eTDQA is already separable. We also propose a scheme to ensure the DQA using eTDQA is convex. The scheme generates a convex approximation, taking into account the function values and sensitivity values at two consecutive design points. The details of this scheme are described in Section 2.2.

It is desirable for an SAO algorithm that its global convergence be guaranteed. However, many popular algorithms for structural optimization, such as MMA and CONLIN, are not guaranteed to be globally convergent (Groenwold and Etman 2010a). Guaranteeing global convergence requires some additional procedure, but this may increase computational cost. Therefore, it is important to improve the convergence property without worsening the efficiency. For example, to improve the convergence property, a line search has been added to MMA (Zillober 1993) and the conservative, convex, and separable approximation (CCSA) has been implemented in MMA (Svanberg 2002). Furthermore, the concept of the trust region, originally proposed for sequential quadratic programming (SQP), was applied to approximate optimization (Alexandrov et al. 1998). Recently, a nonlinear programming (NLP) filter for SQP was proposed (Fletcher et al. 2003, 2002) and applied to SCP (Groenwold and Etman 2010a). In this study, we adopt the techniques of NLP filtering, conservatism, and trust region reduction to improve the convergence property without worsening the efficiency. Our scheme for enforcing global convergence will be described in detail in Section 3.

Our paper is organized as follows: Section 2 describes the proposed DQA using eTDQA. Section 3 describes our method for enforcing global convergence. Section 4 presents the overall procedure of the proposed globally convergent SCP. Section 5 demonstrates the effectiveness of the proposed algorithm through comparisons of its numerical performance in five structural optimization problems versus the performances of competing algorithms. Section 6 gives our concluding remarks.

2 Proposed diagonal quadratic approximation

In this section, we describe in detail a new DQA proposed in this study for SCP. In Section 2.1, we derive the diagonal

Hessian terms of the suggested DQA by twice differentiating eTDQA analytically. In Section 2.2, we present a scheme of enforcing our DQA to be convex.

2.1 Diagonal hessian terms of DQA obtained using eTDQA

The diagonal quadratic approximation for a function $f_j(\mathbf{x})$ at the k th iteration of SAO can be expressed as follows (Groenwold and Etman 2010; Groenwold et al. 2010; Park and Choi 2011):

$$\tilde{f}_j^{(k)}(\mathbf{x}) = f_j(\mathbf{x}^{(k)}) + \sum_{i=1}^n \left(\frac{\partial f_j}{\partial x_i} \right)^{(k)} (x_i - x_i^{(k)}) + \frac{1}{2} \sum_{i=1}^n h_{i,j}^{(k)} (x_i - x_i^{(k)})^2, \quad j = 0, \dots, m, \quad (4)$$

where $h_{i,j}^{(k)}$ denotes the i th approximate diagonal Hessian term. Note that the first and second terms of the right-hand side of (4) can be obtained by function evaluation and design sensitivity analysis. In this study, we employ the second derivative of eTDQA for $h_{i,j}^{(k)}$ as defined in (5).

$$h_{i,j}^{(k)} = \frac{\partial^2 \tilde{f}_j^{eTDQA}(\mathbf{x}^{(k)})}{\partial x_i^2} \quad (5)$$

The eTDQA for a function $f_j(\mathbf{x})$ at the k th iteration of SAO can be mathematically expressed as follows (Kim and Choi 2008):

$$\begin{aligned} \tilde{f}_j^{eTDQA}(\mathbf{x}) = & f_j(\mathbf{x}^{(k)}) + \sum_{i=1}^n \frac{\partial f_j(\mathbf{x}^{(k)})}{\partial y_i} (y_i - y_i^{(k)}) \\ & + \frac{1}{2} \sum_{i=1}^n G_i (y_i - y_i^{(k)})^2 \\ & + \frac{1}{2} \frac{\eta_e \sum_{i=1}^n H_i (y_i - y_i^{(k)})^2}{\sum_{i=1}^n H_i (y_i - y_i^{(k-1)})^2 + \sum_{i=1}^n H_i (y_i - y_i^{(k)})^2} \end{aligned} \quad (6)$$

The i th intervening variable y_i in (6) is defined as

$$y_i = (x_i + c_i)^{p_i}, \quad (7)$$

where c_i is a shifting constant for the i th design variable, included to avoid the fundamental difficulty of defining the exponential intervening variable encountered when x_i has a negative value. In this study, the shifting constant is specified as

$$c_i = \begin{cases} |x_{i,L}| + 0.001 & \text{if } x_{i,L} \leq 0.001 \\ 0 & \text{if } x_{i,L} > 0.001 \end{cases} \quad (8)$$

The exponent p_i in (7) is determined to match the first derivative values of eTDQA with those of the exact func-

tion at the previous design point in the intervening variable space as follows:

$$p_i = 1 + \ln \left[\frac{\partial f_j(\mathbf{x}^{(k-1)})}{\partial x_i} / \frac{\partial f_j(\mathbf{x}^{(k)})}{\partial x_i} \right] / \ln \left[\frac{(x_i^{(k-1)} + c_i)}{(x_i^{(k)} + c_i)} \right]. \quad (9)$$

If the magnitude of p_i becomes too large, the consequent approximate function may become ill conditioned and excessively nonlinear. Thus, in this study, we restrict p_i to the range of $[-2, 4]$. When the signs of the i th first derivative values at two design points are different, the numerator of the second term in the right-hand side of (9) cannot be calculated. In such case, p_i is set to a value of -1 , 1 , or 3 according to the local behavior of f_j ; details on which value is chosen are described in the next section. G_i in eTDQA expressed in (6) is defined as

$$G_i = \begin{cases} \frac{1}{(y_i^{(k-1)} - y_i^{(k)})} \left(\frac{\partial f_j(\mathbf{x}^{(k-1)})}{\partial y_i} - \frac{\partial f_j(\mathbf{x}^{(k)})}{\partial y_i} \right) & \text{if } [\partial f_j(\mathbf{x}^{(k-1)})/\partial x_i] \cdot [\partial f_j(\mathbf{x}^{(k)})/\partial x_i] \leq 0 \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

The term G_i is introduced to ensure that the first derivative values of the approximate function generated by eTDQA are equivalent to those of the exact function at the previous design point even when the value of p_i is arbitrarily set to a preset value because of different signs of the first derivative values at two design points. We also define the term H_i in (6) as

$$H_i = \begin{cases} G_i & \text{if } [\partial f_j(\mathbf{x}^{(k-1)})/\partial x_i] \cdot [\partial f_j(\mathbf{x}^{(k)})/\partial x_i] \leq 0 \\ 1 & \text{otherwise.} \end{cases} \quad (11)$$

We set $H_i = 1$ when the first derivative values at two points have the same sign, in which case $G_i = 0$ as defined in (10) to prevent a zero denominator in the last term in (6). To match the approximate function value with the exact function value at the previous design point, the correction coefficient η_e in (6) is determined as

$$\eta_e = 2 \left[f_j(\mathbf{x}^{(k-1)}) - f_j(\mathbf{x}^{(k)}) - \sum_{i=1}^n \frac{\partial f_j(\mathbf{x}^{(k)})}{\partial y_i} (y_i^{(k-1)} - y_i^{(k)}) - \frac{1}{2} \sum_{i=1}^n G_i (y_i^{(k-1)} - y_i^{(k)})^2 \right] \quad (12)$$

eTDQA is given in the paper by Kim and Choi (2008).

From the brief explanation of eTDQA described above, we know that the Hessian terms of eTDQA can be analytically obtained by twice differentiating eTDQA expressed in (6). The Hessian terms are mathematically expressed as

$$\frac{\partial^2 \tilde{f}_j^{eTDQA}(\mathbf{x}^{(k)})}{\partial x_i^2} = \frac{p_i - 1}{x_i^{(k)} + c_i} \left(\frac{\partial f_j}{\partial x_i} \right)^{(k)} + \left[\frac{\eta_e H_i}{\sum_{l=1}^n H_l (y_l^{(k)} - y_l^{(k-1)})^2} + G_i \right] \times \left(p_i \cdot (x_i^{(k)} + c_i)^{p_i-1} \right)^2, \quad (13)$$

$$\frac{\partial^2 \tilde{f}_j^{eTDQA}(\mathbf{x}^{(k)})}{\partial x_i \partial x_l} = 0 \quad \text{if } i \neq l. \quad (14)$$

As shown in (14), the off-diagonal terms of the Hessian matrix of eTDQA are all zero. Thus, the DQA obtained by substituting (13) for $h_{i,j}^{(k)}$ of (4) is separable.

2.2 Proposed scheme for enforcing the proposed approximation to be convex

A sufficient condition to invoke the efficient dual of Falk (Falk 1967) is the strict convexity of the separable approximate primal subproblem formulated in (2). If the approximate objective function $\tilde{f}_0^{(k)}$ is strictly convex and the approximate constraint functions $\tilde{f}_j^{(k)}$, $j = 1, 2, \dots, m$ are convex, the approximate primal subproblem, (2), is strictly convex. To enforce such convexity with as little artificial modification as possible, a diagonal Hessian term $h_{i,j}^{(k)}$ of the proposed approximation expressed in (13) is divided into two parts as

$$h_{i,j}^{(k)} = P_1 + P_2, \quad (15)$$

$$\text{where } P_1 = \frac{(p_i - 1)}{x_i^{(k)} + c_i} \left(\frac{\partial f_j}{\partial x_i} \right)^{(k)}, \quad (16)$$

$$\text{and } P_2 = \max \left(\varepsilon, \left[\frac{\eta_e H_i}{\left(\sum_{l=1}^n H_l (y_l^{(k)} - y_l^{(k-1)})^2 \right)} + G_i \right] \times \left(p_i \cdot (x_i^{(k)} + c_i)^{p_i-1} \right)^2 \right). \quad (17)$$

We render the second part, P_2 , strictly convex by setting ε to a small positive number. Now, the first part, P_1 , should be

equal to or larger than zero to make this part positive semi-definite. For P_1 to be non-negative, the following condition must be satisfied:

$$If \begin{cases} \left(\frac{\partial f}{\partial x_i}\right)^{(k)} < 0 \Rightarrow p_i < 1, \\ \left(\frac{\partial f}{\partial x_i}\right)^{(k)} \geq 0 \Rightarrow p_i > 1 \end{cases} \quad then \quad P_1 \geq 0. \quad (18)$$

Figure 1 shows eight convex or concave functional behaviors near two consecutive design points when the derivative values of two consecutive points have the same sign. If the function f is locally convex (four unshaded cases in Fig. 1), then p_i as calculated by (9) satisfies condition (18). If the function f is locally concave (four shaded cases in Fig. 1), then p_i as calculated by (9) does not satisfy the condition (18) and we have to force p_i to obey the condition (18). Equation (9) in Fig. 1 denotes the equation number from which the p_i value is calculated for locally convex cases, and the numbers 3 and -1 in Fig. 1 denote p_i values for locally concave cases. How to determine the p_i value will be elaborated in the next paragraph.

In case that f is locally concave, we set $p_i = 3$ if the condition in (19) is satisfied, and set $p_i = -1$ if the condition in (20) is satisfied, so that the absolute value of $p_i - 1$ in (16) equals 2 based on our numerical experience.

$$If \left(\frac{\partial f}{\partial x_i}\right)^{(k)} > 0 \text{ and } \left(\left(\frac{\partial f}{\partial x_i}\right)^{(k)} - \left(\frac{\partial f}{\partial x_i}\right)^{(k-1)}\right) \left((x_i^{(k)} + c_i) - (x_i^{(k-1)} + c_i)\right) < 0, \quad \text{then } p_i = 3 \quad (19)$$

$$If \left(\frac{\partial f}{\partial x_i}\right)^{(k)} < 0 \text{ and } \left(\left(\frac{\partial f}{\partial x_i}\right)^{(k)} - \left(\frac{\partial f}{\partial x_i}\right)^{(k-1)}\right) \left((x_i^{(k)} + c_i) - (x_i^{(k-1)} + c_i)\right) > 0, \quad \text{then } p_i = -1 \quad (20)$$

In case that f is locally convex, the value is determined by (9) as mentioned in the previous paragraph. However, when p_i is too large, this may result in very slow convergence, and when it is too small, this may result in oscillatory behavior (Groenwold and Etman 2010). Therefore, based on our numerical experience, we restricted the p_i value to be between 2 and 4 if $\frac{\partial f(x^{(k)})}{\partial x_i} \geq 0$, and between -2 and 0 if $\frac{\partial f(x^{(k)})}{\partial x_i} < 0$.

In case that f is neither locally convex nor concave, i.e. $\left(\frac{\partial f}{\partial x}\right)^{(k)}$ is close to zero, p_i is set to 1. Also, when the derivative values of two consecutive points are almost same, f can be interpreted to be linear and we set $p_i = 1$. These cases

are mathematically expressed in (21). In this study, we set $\varepsilon = 10^{-7}$

$$If \left|\left(\frac{\partial f}{\partial x_i}\right)^{(k)}\right| \leq \varepsilon \text{ or } \left|\left(\frac{\partial f}{\partial x_i}\right)^{(k)} - \left(\frac{\partial f}{\partial x_i}\right)^{(k-1)}\right| \leq \varepsilon, \text{ then } p_i = 1 \quad (21)$$

If the signs of the design sensitivity values of two consecutive design points are different (i.e., if $\left[\frac{\partial f_j(x^{(k-1)})}{\partial x} \cdot \frac{\partial f_j(x^{(k)})}{\partial x}\right] < 0$), then the numerator of the second term in (9) cannot be calculated because the argument of the logarithm becomes negative. In this case, we set the p_i value to 1.

In the case of a linear function, such as the volume fraction of a structural topology optimization problem, the value of $h_{i,j}^{(k)}$ is set to zero by following the rule of mix-and-match (Groenwold and Etman 2010, 2010a, Groenwold et al. 2010).

The proposed CSA of the j th function at the k th iteration can now be expressed as follows:

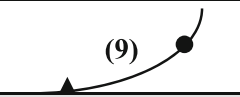
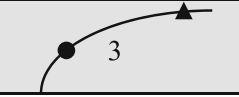
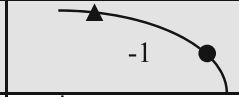
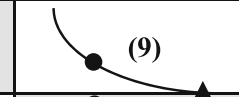
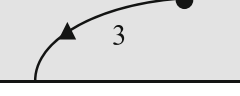
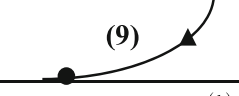
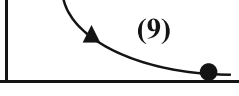
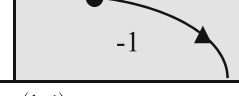
$$\begin{aligned} \tilde{f}_j^{(k)}(x) = & f_j(x^{(k)}) + \sum_{i=1}^n \left(\frac{\partial f_j}{\partial x_i}\right)^{(k)} (x_i - x_i^{(k)}) \\ & + \frac{1}{2} \sum_{i=1}^n \left[\frac{p_i - 1}{x_i^{(k)} + c_i} \left(\frac{\partial f_j}{\partial x_i}\right)^{(k)} + \max \left(\varepsilon, \left\{ \frac{\eta_e H_i}{\left(\sum_{l=1}^n H_l (y_l^{(k)} - y_l^{(k-1)})^2\right)} + G_i \right\} \right) \right. \\ & \left. \times \left(p_i \cdot (x_i^{(k)} + c_i)^{p_i - 1}\right)^2 \right] (x_i - x_i^{(k)})^2, \end{aligned} \quad (22)$$

where the p_i is determined as shown in Fig. 1. Hereafter, the proposed CSA is named T2:eTDQA.

3 Proposed method for enforcing global convergence of T2:eTDQA

A SCP using T2:eTDQA in its rudimentary form may proceed as follows:

- 1) Set $k = 0$ and select constants and parameters required for the CSA.
- 2) Perform function evaluations to compute $f_j(x^{(k)})$ and design sensitivity analyses to compute $\nabla f_j(x^{(k)})$, $j = 0, \dots, m$.
- 3) Construct the CSAs, (22), of the objective and constraint functions at the current design point.
- 4) Construct the local approximate dual subproblem, (13), and solve it to find the approximate optimum, $x^{(k*)}$.

	$(\partial f(\mathbf{x}^{(k)}) / \partial x_i) > 0$		$(\partial f(\mathbf{x}^{(k)}) / \partial x_i) < 0$	
	$x_i^{(k)} + c_i \geq x_i^{(k-1)} + c_i$	$x_i^{(k)} + c_i < x_i^{(k-1)} + c_i$	$x_i^{(k)} + c_i \geq x_i^{(k-1)} + c_i$	$x_i^{(k)} + c_i < x_i^{(k-1)} + c_i$
$\left(\frac{\partial f}{\partial x_i}\right)^{(k)} > \left(\frac{\partial f}{\partial x_i}\right)^{(k-1)}$				
$\left(\frac{\partial f}{\partial x_i}\right)^{(k)} < \left(\frac{\partial f}{\partial x_i}\right)^{(k-1)}$				

● $x_i^{(k)}$ Current point ▲ $x_i^{(k-1)}$ Previous point

Fig. 1 p_i value for eight convex or concave behaviors when the derivative values of two consecutive points have the same sign

- 5) Perform function evaluations to compute $f_j(\mathbf{x}^{(k*)})$, $j = 0, \dots, m$.
- 6) Set $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k*)}$.
- 7) If $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| \leq \varepsilon_x$, then STOP.
Otherwise (not converged), compute $\nabla f_j(\mathbf{x}^{(k+1)})$, $j = 0, \dots, m$.
- 8) Set $k := k + 1$ and GOTO step 3.

It is well known that the SAO methods mentioned in Section 1, including our T2:eTDQA, may not be globally convergent in their rudimentary forms (Groenwold and Etman 2010a; Svanberg 2002; Zillober 1993). One obvious way to force global convergence is to employ a trust region method (Alexandrov et al. 1998; Conn et al. 2000) in combination with a merit function to decide whether a design point is accepted. Fletcher et al. (2002) replaced the (problematic) merit function with an NLP filter in a trust region framework. As an alternative to the trust region method, Svanberg (2002) introduced the concept of conservative, convex and separable approximations, and numerically implemented this concept in the MMA. The filtered conservatism proposed by Groenwold and Etman (2010a) combined the salient features of both trust region methods with NLP filtering and conservatism. In their method, the NLP filter was used to decide whether an approximate optimum point was accepted or rejected; if rejected, the filtered conservatism did not decrease the trust region but increased the conservatism of only unconservative approximations in the constant trust region until the approximate optimum point became acceptable to the filter.

To guarantee global convergence of T2:eTDQA, we adopt the NLP filter proposed by Fletcher et al. (2002) to judge the acceptance of $\mathbf{x}^{(k*)}$ obtained in Step 4 of the rudimentary algorithm. If $\mathbf{x}^{(k*)}$ is not acceptable, we adopt both conservatism and trust region reduction in inner loop iteration to force global convergence efficiently. The conservatism we employ in this study is different from that used in Svanberg (2002) and Groenwold et al. (2010a). In Section 3.1, the concept of the NLP filtering is briefly

presented to explain our method, and our new method implemented to force global convergence efficiently in inner loop iteration is described in detail in Section 3.2.

3.1 Nonlinear programming filter

Let $f = f_0$ and $h = \max\{0, f_j\}$, $j = 1, \dots, m$. A pair $(h^{(i)}, f^{(i)})$ obtained at iteration i , is said to dominate another pair $(h^{(j)}, f^{(j)})$ if and only if both $h^{(i)} \leq h^{(j)}$ and $f^{(i)} \leq f^{(j)}$. The NLP filter is defined as a list of pairs $(h^{(i)}, f^{(i)})$ such that no pair dominates any other. In other words, the NLP filter represents Pareto optima of a multi-objective optimization problem of minimizing both h and f . To enforce global convergence, we employ the NLP filter to test if $\mathbf{x}^{(k*)}$ obtained in Step 4 is acceptable as the optimum point at the k th iteration, instead of unconditionally accepting it as in the rudimentary algorithm. Denoting the pair obtained at $\mathbf{x}^{(k*)}$ as (h, f) , $\mathbf{x}^{(k*)}$ is judged to be acceptable if either $h < h^{(i)}$ or $f < f^{(i)}$ for all i in the current filter. This condition was slightly modified and the following slanting envelope test was proposed by Fletcher et al. (2002) to prove convergence: a pair (h, f) is acceptable if

$$\text{either } h \leq \beta h^{(i)} \text{ or } f + \gamma h \leq f^{(i)}, \text{ for all } i \text{ in the current filter} \quad (23)$$

where β and γ are preset parameters such that $1 > \beta > \gamma > 0$, with β close to 1 and γ close to zero. In this study, we set $\gamma = 10^{-5}$ and $\beta = 1 - \gamma$. If $\mathbf{x}^{(k*)}$ does not pass the slanting envelope test, it is not accepted as $\mathbf{x}^{(k+1)}$ but an inner loop is invoked to iterate until the test is passed. Otherwise, a sufficient reduction condition is tested as follows:

The sufficient reduction condition tests if the reduction of the real objective value is sufficient compared to that of the approximate objective value. Let $\Delta f = f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k*)})$ and $\Delta q = \tilde{f}(\mathbf{x}^{(k)}) - \tilde{f}(\mathbf{x}^{(k*)})$. Then the sufficient reduction condition is not satisfied if

$$\nabla f < \sigma \Delta q \text{ and } \Delta q > 0, \quad (24)$$

where σ is a preset positive parameter. In this study, we set $\sigma = 0.1$. If the $\mathbf{x}^{(k*)}$ does not pass the sufficient reduction test, it is not accepted as $\mathbf{x}^{(k+1)}$ but an inner loop is invoked to iterate until the test is passed.

In this study, we follow the work of Fletcher et al. (2002): if Δq is positive, then an f -type iteration is said to have occurred, and if $\Delta q \leq 0$, then the primary aim of iteration is to reduce h (possibly allowing an increase in f) and the resulting iteration is referred to as an h -type iteration. Only in an h -type iteration, the pair $(h^{(k)}, f^{(k)})$ is included in the filter. The case of no feasible solutions in the trust region at the current approximate subproblem is called as “incompatible” and also included in the h -type iteration. If there are no feasible solutions in whole design space, the algorithm cannot converge due to non-existence of feasible solutions.

This test procedure can be summarized as follows:

If $h \leq \beta h^{(i)}$ or $f + \gamma h \leq f^{(i)}$, for all i in the current filter is not satisfied, then go to the inner loop.

If $\Delta f < \sigma \Delta q$ and $\Delta q > 0$, then go to the inner loop.

Otherwise, accept $\mathbf{x}^{(k*)}$ as $\mathbf{x}^{(k+1)}$, and include $(h^{(k)}, f^{(k)})$ in the filter if $\Delta q \leq 0$.

A more detailed description of the test and a discussion of its meaning are provided in (Fletcher et al. 2002; Groenwold and Etman 2010a). Section 3.2 describes the implementation of our method for effectively forcing global convergence in the inner loop iteration.

3.2 Enforcing global convergence in inner loop iteration

In the inner loop iteration, we adopt both conservatism and trust region reduction. In this study, we propose to use an adaptive conservatism instead of the fixed conservatism as used in Svanberg (2002) and Groenwold et al. (2010a).

3.2.1 Conservatism

Svanberg (2002) developed the concept of conservative, convex and separable approximations (CCSA) that enforce conservatism by multiplying the diagonal Hessian term of the DQA (4) by a constant χ larger than one as follows:

$$\begin{aligned} \tilde{f}_j^{(k)}(\mathbf{x}) &= f_j(\mathbf{x}^{(k)}) + \sum_{i=1}^n \left(\frac{\partial f_j}{\partial x_i} \right)^{(k)} \\ &\quad \left(x_i - x_i^{(k)} \right) + \frac{1}{2} \chi \sum_{i=1}^n h_{i,j}^{(k)} \left(x_i - x_i^{(k)} \right)^2, \quad j = 0, \dots, m. \end{aligned} \quad (25)$$

He implemented it in the MMA and demonstrated the global convergence of this ‘globally convergent version of MMA’. Rather than enforcing conservatism unconditionally as done by Svanberg, Groenwold et al. (2009) suggested “relaxed conservatism” in which conservatism was enforced only if

a feasible descent step could not be made. In their study, the diagonal Hessian terms of the CSA were also multiplied by a constant larger than one to enforce conservatism.

In this study, instead of using a constant multiplier, we propose to enforce conservatism adaptively by utilizing the real function value $f_j(\mathbf{x}^{(k*)})$ at the approximate optimum $\mathbf{x}^{(k*)}$ already calculated before conducting the test procedure using the NLP filter. By matching the real function value $f_j(\mathbf{x}^{(k*)})$ and the conservative CSA value at the approximate optimum $\mathbf{x}^{(k*)}$, we obtain

$$\begin{aligned} f_j(\mathbf{x}^{(k*)}) &= f_j(\mathbf{x}^{(k)}) + \sum_{i=1}^n \left(\frac{\partial f_j}{\partial x_i} \right)^{(k)} \left(x_i^{(k*)} - x_i^{(k)} \right) \\ &\quad + \frac{1}{2} \Psi_j^{(l)} \sum_{i=1}^n h_{i,j}^{(k)} \left(x_i^{(k*)} - x_i^{(k)} \right)^2, \quad j = 0, \dots, m, \end{aligned} \quad (26)$$

where $\Psi_j^{(l)}$ is a multiplier for the l th inner iteration used instead of χ in (25). From (26), an adaptive multiplier $\Psi_j^{(l)}$ can be explicitly determined as

$$\Psi_j^{(l)} = \frac{2 \left(f_j(\mathbf{x}^{(k*)}) - f_j(\mathbf{x}^{(k)}) - \sum_{i=1}^n \left(\frac{\partial f_j}{\partial x_i} \right)^{(k)} \left(x_i^{(k*)} - x_i^{(k)} \right) \right)}{\sum_{i=1}^n h_{i,j}^{(k)} \left(x_i^{(k*)} - x_i^{(k)} \right)^2}. \quad (27)$$

In this study, we adjust $\Psi_j^{(l)}$ as in (28) to make a slightly more conservative CSA.

$$\Psi_j^{(l)} = \max \left\{ 1.1, 1.1 \Psi_j^{(l)} \right\} \quad (28)$$

3.2.2 Trust region reduction

In addition to enforcing conservatism, we decrease the trust region in inner loop iteration as (Fletcher et al. 2002; Groenwold and Etman 2010a) did. In this study, the trust region is decreased by halving the move limit ratio mlr (defined as the ratio of the move limit length to the range of each design variable) as

$$mlr^{(k,l+1)} = \frac{mlr^{(k,l)}}{2}. \quad (29)$$

Then, the lower and upper bounds of the i th design variable at the $(l+1)$ th inner iteration can be determined as follows:

$$\begin{aligned} x_{i,L}^{(k,l+1)} &= \max \left\{ x_{i,L}, x_i^{(k,0)} - mlr^{(k,l+1)} \cdot (x_{i,U} - x_{i,L}) \right\} \\ x_{i,U}^{(k,l+1)} &= \min \left\{ x_{i,U}, x_i^{(k,0)} + mlr^{(k,l+1)} \cdot (x_{i,U} - x_{i,L}) \right\}, \end{aligned} \quad (30)$$

where $x_{i,L}^{(k,l+1)}$ and $x_{i,U}^{(k,l+1)}$ denote the lower and upper bounds of the $(l+1)$ th inner iteration, respectively.

Table 1 Five structural optimization test problems, their number of design variables n , and their number of constraints m

No.	Option	Problem	n	m
1		Two bar shape and size design problem	2	2
2	a	Vanderplaats' cantilever beam problem	20	21
	b		200	201
3		Svanberg's 5-variate cantilever beam problem	5	1
4		Fleury's weight minimization like problem	1000	1001
5	a	MBB beam topology optimization problem ($p = 1, 75$ by 25)	1875	1
	b	MBB beam topology optimization problem ($p = 3, 75$ by 25)	1875	1

4 Proposed globally convergent sequential convex programming

By combining the proposed method for enforcing global convergence (described in Section 3) with the proposed CSA called T2:eTDQA (described in Section 2), we propose a globally convergent SCP, named *gc-T2:eTDQA*, which proceeds as follows:

- 1) Set $k := 0$ and specify constants and parameters.
- 2) Perform function evaluations and sensitivity analyses to compute $f_j(\mathbf{x}^{(k)})$ and $\nabla f_j(\mathbf{x}^{(k)})$, $j = 0, \dots, m$, respectively.
- 3) Construct the CSAs, (22), of the objective and constraint functions at the current design point.
- 4) Construct the local approximate dual subproblem, (3), and solve it to find the approximate optimum, $\mathbf{x}^{(k*)}$.
- 5) Perform function evaluations to compute $f_j(\mathbf{x}^{(k*)})$, $j = 0, \dots, m$.
- 6) If $h \leq \beta h^{(i)}$ or $f + \gamma h \leq f^{(i)}$, for all i in the current filter is not satisfied, then GOTO Step 9 (inner loop).
- 7) Otherwise, if $\Delta f < \sigma \Delta q$ and $\Delta q > 0$, then GOTO Step 9 (inner loop).
- 8) Otherwise, accept $\mathbf{x}^{(k*)}$ as $\mathbf{x}^{(k+1)}$, and include $(h^{(k)}, f^{(k)})$ in the filter if $\Delta q \leq 0$, then GOTO Step 11.
- 9) Enforce conservatism in the constructed CSA by using an adaptive multiplier $\Psi_j^{(l)}$ in (28).

- 10) Reduce the trust region, set the lower and upper bounds of design variables for the next inner iteration as in (30), and GOTO Step 4.
- 11) If $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| \leq \varepsilon_x$, then STOP. Otherwise (not converged), compute $\nabla f_j(\mathbf{x}^{(k+1)})$, $j = 0, \dots, m$.
- 12) $k := k+1$ and GOTO Step 3.

Note that Steps 6 through 10 are added to the rudimentary algorithm described in the first paragraph of Section 3 to implement *gc-T2:eTDQA*.

5 Numerical examples

To demonstrate the effectiveness of the *gc-T2:eTDQA* proposed for structural optimization, we solve five structural optimization test problems. These test problems are listed in Table 1, along with their corresponding numbers of design variables and constraints. The upper bounds, lower bounds, and initial values of design variables for each problem are given in Table 2. A detailed description of each problem is given in the Appendix. Note that we use the same initial values of design variables as used in the previous studies presented in the Appendix. Table 2 also specifies the values of convergence tolerance in Step 11 for each problem. Note that we use the same convergence tolerance values

Table 2 Lower bounds (x_L), upper bounds (x_U), and initial values (x^0) of design variables and convergence tolerances (ε_x)

No.				x_U	ε_x
1	x_1	0.2	1	4	10^{-5}
	x_2	0.1	1	1.6	10^{-5}
2	$x_1 \sim x_{n/2}$	1	5	80	10^{-5}
	$x_{n/2+1} \sim x_n$	5	60	80	10^{-5}
3	$x_1 \sim x_n$	1	5	10	10^{-4}
4	$x_1 \sim x_n$	10^{-6}	10^{-5}	10^6	10^{-5}
5	$x_1 \sim x_n$	0.01	0.5	1	10^{-5}

as Groenwold and Etman (2010a) and Groenwold et al. (2010) to fairly compare the performance of gc -T2:eTDQA with that of competing algorithms. To solve dual subproblems, the projected BFGS (Kelley 1999) is employed in this study. First, we investigate the effect of our method for enforcing global convergence in inner loop iteration as proposed in Section 3.2. Then, we assess the performance of the proposed gc -T2:eTDQA using competing algorithms.

5.1 Effect of our method for enforcing global convergence in inner loop iteration

To investigate the effect of the proposed method for enforcing both adaptive conservatism and trust region reduction in inner loop iteration, we compare our enforcement method with two enforcement alternatives. One alternative, instead of using the adaptive conservatism expressed in (28), employs a fixed conservatism that uses a constant value of 2 for the multiplier in (25), as Svanberg (2002) and Groenwold and Etman (2010a) did. The globally convergent SCP using this alternative is named gc -T2:eTDQA- a . The performance of gc -T2:eTDQA- a is compared with that of the proposed gc -T2:eTDQA to investigate the effects of adaptive versus fixed conservatism. The second alternative is to employ adaptive conservatism only, without employing trust region reduction. The globally convergent SCP using

this alternative is named gc -T2:eTDQA- b . The performance of the gc -T2:eTDQA- b will be compared with that of the proposed gc -T2:eTDQA to investigate the effect of adding trust region reduction to the adaptive conservatism.

Table 3 lists the numerical results for all test problems obtained using gc -T2:eTDQA, gc -T2:eTDQA- a , and gc -T2:eTDQA- b . In Table 3, k^* and l^* denote the numbers of outer and inner iterations, respectively, that are used in obtaining the optimum objective value f_0^* . $Max(f_j)$ denotes the maximum constraint value from which we can confirm satisfaction of all constraints. Problems 2a, 3 and 4 are found to converge well to the optimum solution without resorting to inner loop iteration. In case of Problems 1 and 2b, the three methods perform equally well. The effect of the proposed method of enforcing global convergence is demonstrated well in Problem 5, a topology optimization problem. The three methods are similarly accurate for Problem 5, but gc -T2:eTDQA is much more efficient than gc -T2:eTDQA- a and gc -T2:eTDQA- b for Problem 5a, and slightly more efficient for Problem 5b. Note that the computational cost of an outer iteration, which requires function and design sensitivity evaluations, is slightly more expensive than that of an inner iteration, which requires only function evaluation. Thus, the proposed method of enforcing global convergence can be said to be equally efficient or more so than the two alternatives.

Table 3 Comparison of methods of enforcing global convergence in inner loop iteration

No.	Method				
1	gc -T2:eTDQA	9	1	1.508656	-2.33×10^{-6}
	gc -T2:eTDQA- a	9	1	1.508656	-2.33×10^{-6}
	gc -T2:eTDQA- b	9	1	1.508656	-2.33×10^{-6}
2a	gc -T2:eTDQA	10	0	64244.83	5.35×10^{-6}
	gc -T2:eTDQA- a	—	—	—	—
	gc -T2:eTDQA- b	—	—	—	—
2b	gc -T2:eTDQA	10	1	63678.1	1.66×10^{-6}
	gc -T2:eTDQA- a	10	1	63678.1	6.80×10^{-6}
	gc -T2:eTDQA- b	10	1	63678.1	1.66×10^{-6}
3	gc -T2:eTDQA	8	0	1.339957	-1.26×10^{-6}
	gc -T2:eTDQA- a	—	—	—	—
	gc -T2:eTDQA- b	—	—	—	—
4	gc -T2:eTDQA	25	0	950	-5.04×10^{-7}
	gc -T2:eTDQA- a	—	—	—	—
	gc -T2:eTDQA- b	—	—	—	—
5a	gc -T2:eTDQA	39	16	165.494	-2.70×10^{-6}
	gc -T2:eTDQA- a	77	91	165.611	-1.00×10^{-5}
	gc -T2:eTDQA- b	39	1000	165.494	-3.41×10^{-13}
5b	gc -T2:eTDQA	34	47	204.7286	-3.43×10^{-7}
	gc -T2:eTDQA- a	31	58	204.7738	-1.43×10^{-7}
	gc -T2:eTDQA- b	34	50	204.7158	-9.20×10^{-8}

Table 4 Comparison of numerical results from various SCP algorithms

No.	Method				
1	SAO-A†	9	–	1.508652	1.98×10^{-12}
	SAO-B	6	5	1.508652	1.16×10^{-10}
	SAO-C	9	0	1.508652	1.98×10^{-12}
	SAO-D	9	0	1.508652	1.98×10^{-12}
	T2:εTDQA†	9	–	1.508653	-4.11×10^{-7}
	gc-T2:εTDQA	9	0	1.508656	-2.33×10^{-6}
2a	SAO-A†	38	–	64244.83	1.68×10^{-6}
	SAO-B	101	261	64244.83	1.11×10^{-6}
	SAO-C	40	94	64244.83	3.21×10^{-6}
	SAO-D	29	2	64244.83	2.41×10^{-5}
	T2:εTDQA†	10	–	64244.83	5.35×10^{-6}
	gc-T2:εTDQA	10	0	64244.83	5.35×10^{-6}
2b	SAO-A†	34	–	63678.1	1.78×10^{-6}
	SAO-B	457	2535	63678.1	4.90×10^{-6}
	SAO-C	30	25	63678.1	2.98×10^{-7}
	SAO-D	29	1	63678.1	4.06×10^{-5}
	T2:εTDQA†	11	–	63678.1	8.66×10^{-6}
	gc-T2:εTDQA	10	1	63678.1	1.66×10^{-6}
3	T2:R	10	8	1.339956	–
	T2:E	13	7	1.339956	–
	T2:MMA	20	15	1.339956	–
	T2:TANA-3	10	4	1.339956	–
	GCMMA	19	20	1.339956	–
	T2:εTDQA†	8	–	1.339957	-1.26×10^{-6}
	gc-T2:εTDQA	8	0	1.339957	-1.26×10^{-6}
	SAO-A†	not converged			
4	SAO-B	40	51	950.0001	-3.85×10^{-6}
	SAO-C	25	39	950.0001	6.82×10^{-11}
	SAO-D	35	23	950.0001	6.54×10^{-8}
	T2:R	34	0	950.0001	–
	T2:E	34	0	950.0001	–
	T2:C	32	2	950.0001	–
	T2:MMA	30	593	950.0001	–
	T2:TANA-3	48	0	950.0001	–
	T2:εTDQA†	25	–	950.0001	-5.04×10^{-7}
	gc-T2:εTDQA	25	0	950.0001	-5.04×10^{-7}
	T2:R	58	0	165.8839	-1.11×10^{-13}
	T2:E	35	0	165.8838	-5.46×10^{-16}
5a	T2:MMA	43	11	165.8838	-6.06×10^{-16}
	GCMMA	37	103	165.9624	1.90×10^{-8}
	T2:εTDQA †	56	–	165.4939	-1.02×10^{-11}
	gc-T2:εTDQA	39	16	165.4939	-2.70×10^{-6}
	T2:R ‡	83	23	204.27	–
	T2:E‡	92	28	203.84	–
5b	T2:MMA‡	107	90	205.93	–
	MMA	not converged	–	–	–
	GCMMA	not converged	–	–	–

Table 4 (continued)

No.	Method				
	T2:eTDQA [†]	not converged	–	–	–
	<i>gc</i> -T2:eTDQA	34	47	204.73	-3.43×10^{-7}

[†]SAO-A and T2:eTDQA are rudimentary algorithms without any enforcement of global convergence

[‡]Note that competing algorithms T2:R, T2:E, and T2:MMA solved the ‘discrete’ MBB beam problem using a continuation method on the penalization parameter rather than using a constant parameter of three

5.2 Performance of the proposed *gc*-T2:eTDQA compared with competing algorithms

To assess the performance of the proposed *gc*-T2:eTDQA using previous studies, we compare our numerical results for Problems 1, 2a, 2b, and 4 with those of the structural optimization problems presented in Groenwold and Etman (2010a), and our numerical results for Problems 3, 4, and 5a with those of structural optimization problems presented in Groenwold et al. (2010). We include Problem 5b (which has not been solved in the two aforementioned references) because the penalization parameter value of three ($p = 3$) is usually employed in topology optimization.

Table 4 lists the numerical results of *gc*-T2:eTDQA and T2:eTDQA and the results of competing algorithms. Groenwold and Etman (2010a) studied the convergence and termination of SCP using two recently proposed strategies, namely, a trust region with filtered acceptance, and conservatism. They compared four methods of enforcing global convergence: SAO-A (no enforcement; rudimentary), SAO-B (which enforces conservatism), SAO-C (which enforces filtered trust region), and SAO-D (which enforces filtered conservatism); these are described in more detail in Groenwold and Etman (2010a). The numerical results of SAO-A, SAO-B, SAO-C, and SAO-D for Problems 1, 2a, 2b, and 4 listed in Table 4 are copied from Groenwold and Etman (2010a).

Groenwold et al. (2010) proposed to replace a number of popular approximations, such as reciprocal approximation (R), exponential approximation (E), conservative approximation (C), MMA, and TANA-3, by their diagonal quadratic Taylor series expansions; the replacements were respectively named T2:R, T2:E, T2:C, T2:MMA, and T2:TANA-3. The resulting quadratic approximations are easily convexified and well suited for use in SCP. They enforced global convergence using a fixed conservatism by simply increasing the approximate diagonal curvatures in inner loop iteration. Please refer to Groenwold et al. (2010) for a more detailed description of their proposed diagonal quadratic approximations. The numerical results of various approximations for Problems 3, 4, and 5a listed in Table 4 are copied from Groenwold et al. (2010). In the case of Problem 5b, we solved the Messerschmitt-Bolkow-Blohm (MBB)

beam topology optimization problem with the penalization parameter of three, using the MMA and the globally convergent version of MMA (GCMMA) for comparison. The linear constraint on volume is modeled exactly. Groenwold et al. (2010) applied T2:E, T2:R, and T2:MMA to solve the ‘discrete’ MBB problem using a continuation strategy for the penalization parameter rather than using a constant penalization parameter of three. Even though they solved the problem using a different strategy for the penalization parameter, their numerical results are also copied in the first three rows of Problem 5b in Table 4 for comparison. (Strictly speaking, this comparison may not be fair.)

As shown in the f_0^* and $\max(f_j)$ columns in Table 4, all algorithms converge to the test problems’ optimum solutions, except three algorithms for Problem 5b and SAO-A for Problem 4. In the case of Problem 5b with $p = 3$, only *gc*-T2:eTDQA converges to the optimum solution among the four algorithms tested, demonstrating the robustness of our proposed *gc*-T2:eTDQA. Figure 2 depicts the layout of the MBB beam after topology optimization with $p = 3$ using *gc*-T2:eTDQA. Comparing the performance of *gc*-T2:eTDQA with the performances of the first three algorithms using a continuation strategy for the penalization parameter in Problem 5b (see Table 4) clearly shows that all four algorithms converge to similar optimum solutions, but that *gc*-T2:eTDQA does so much more efficiently.

Typically, a convergence tolerance for maximum change of design variable, ε_x , for topology optimization is 10^{-2} or 10^{-3} . However, in our practice, we used 10^{-5} same with the tolerance value for example 5a and 5b to verify the effectiveness of optimization algorithm. The numerical results of example 5b with different tolerance are illustrated in Table 5. According to those results, we found that a small oscillation is occurred in problem 5b nearby the



Fig. 2 Topology optimization result of Problem 5b using *gc*-T2:eTDQA

Table 5 Comparison of numerical results with different convergence tolerance values for Example 5b

No.	Method				
5b ($\varepsilon_x = 10^{-2}$)	MMA	74	—	205.657	-3.52×10^{-3}
	GCMMA	58	175	236.552	-3.92×10^{-4}
	T2:eTDQA†	145	—	203.959	-6.76×10^{-11}
	gc-T2:eTDQA	34	47	204.729	-3.43×10^{-7}
5b ($\varepsilon_x = 10^{-3}$)	MMA	301	—	205.784	-5.22×10^{-5}
	GCMMA	212	928	205.165	-4.13×10^{-6}
	T2:eTDQA†	956	—	205.692	6.87×10^{-11}
	gc-T2:eTDQA	34	47	204.729	-3.43×10^{-7}
5b ($\varepsilon_x = 10^{-4}$)	MMA	not converged	—	—	—
	GCMMA	880	4258	205.553	-2.51×10^{-6}
	T2:eTDQA†	not converged	—	—	—
	gc-T2:eTDQA	34	47	204.729	-3.43×10^{-7}

optimum point for the MMA and GCMMA. According to the decrease of tolerance value, the number of iterations for convergence increases except for the gc-T2:eTDQA. When the tolerance value is set to 10^{-4} , the MMA and T2:eTDQA cannot be converged. When the tolerance value is set to 10^{-2} , all of the algorithms are converged but the GCMMA converges to the wrong solution which has larger objective value. When the tolerance is set to 10^{-3} , all of the algorithms are converged with the larger number of iteration than that of the gc-T2:eTDQA. Therefore, the proposed gc-T2:eTDQA is more efficient than other three algorithms for problem 5b. It seems that the NLP filter and the procedures in inner iteration proposed by our research can prevent small oscillation nearby the optimum point.

The efficiency of gc-T2:eTDQA is equivalent to that of T2:eTDQA for Problems 1, 2a, 3, and 4, and very slightly better for Problems 2b and 5a. (Note that the T2:eTDQA is a rudimentary algorithm.) This illustrates that our proposed method for enforcing global convergence of the T2:eTDQA guarantees convergence without worsening efficiency.

In the case of Problem 1, gc-T2:eTDQA is equally efficient as or slightly more efficient than the competing algorithms. In the case of Problems 2a, 2b, 3, and 4, the gc-T2:eTDQA is considerably more efficient than the competing algorithms. In the case of Problem 5a, gc-T2:eTDQA is less efficient than T2:E and very similar to T2:MMA, though it is slightly better than T2:R and far better than GCMMA.

For Example 5b, we also compared numerical results with different convergence tolerance (ε_x) values. The comparison results are shown in Table 5. As shown in Table 5, MMA converged to the optimum solution when ε_x is set to 10^{-2} or 10^{-3} , but did not converge when ε_x is equal to or tighter than 10^{-4} . GCMMA converged to a wrong solution when ε_x is set to 10^{-2} , converged to the optimum

solution when ε_x is set to 10^{-3} and 10^{-4} , and did not converge when ε_x is set to 10^{-5} . We investigated the cases that MMA and GCMMA did not converge, and found that a small oscillation occurred near the optimum point. The proposed algorithm with the NLP filter and the procedures in inner iteration, however, can prevent such a small oscillation near the optimum point and converge to the optimum solution with any convergence tolerance values employed in this study.

From the comparisons of numerical results for all the test problems described above, we can conclude that the proposed gc-T2:eTDQA generally performs better than all the competing algorithms. We believe that the proposed CSA described in Section 2 enhances efficiency, and that the proposed method of enforcing global convergence described in Section 3 enhances robustness without worsening efficiency. Recall that features of the proposed CSA are employing the diagonal Hessian terms of a highly accurate two-point approximation (eTDQA) for DQA, as described in Section 2.1, and enforcing convexity with as little artificial modification as possible, as described in Section 2.2.

6 Conclusions

In this study, we have proposed a new globally convergent SCP, called gc-T2:eTDQA, that employs a new CSA as well as NLP filtering, conservatism, and trust region reduction for enforcing global convergence. The diagonal Hessian terms of the proposed DQA are obtained by twice analytically differentiating an eTDQA, which has been demonstrated to generate highly accurate approximation results. To enforce the proposed DQA to be convex with as little artificial modification of the original diagonal Hessian terms of eTDQA as possible, we also proposed a scheme for

generating a convex approximation that takes into account the function and design sensitivity values at two consecutive design points. To guarantee global convergence, we adopted the NLP filter, and proposed to use both adaptive conservatism and trust region reduction for enforcing global convergence without worsening efficiency.

To demonstrate *gc-T2:eTDQA*'s effectiveness in structural optimization, we solved five test problems. First, we investigated the effect of the proposed method in employing both adaptive conservatism and trust region reduction to enforce global convergence by comparing its numerical results with those of two other alternatives. One alternative was to employ a fixed conservatism, and the other was to employ adaptive conservatism only without including trust region reduction. This comparison revealed that the numerical performance of the proposed method for enforcing global convergence was equivalent to or better than that of the two alternatives.

We then compared the numerical performance of *gc-T2:eTDQA* with the performances of competing algorithms for all the test problems. From this comparison, we conclude that the proposed *gc-T2:eTDQA* generally performed better than all competing algorithms to which it was compared. We believe that the proposed CSA enhances efficiency, and that the proposed method of enforcing global convergence enhances robustness without worsening efficiency.

Acknowledgments This work was supported by the National Space Lab (NSL) program through the National Research Foundation (NRF) of Korea funded by the Ministry of Science, ICT and Future Planning (No. 2013042240) and by the NRF grant funded by the Korean government (No. 2013031530).

Appendix: The test problems

A.1 Shape and size design of a two-bar truss

This problem was proposed by Svanberg (1987). The problem can be mathematically stated as

$$\begin{aligned} & \underset{x}{\text{minimize}} && f_0(x) = c_1 x_1 \sqrt{1 + x_2^2} \\ & \text{subject to} && f_1(x) = c_2 \sqrt{1 + x_2^2} \left(\frac{8}{x_1} + \frac{1}{x_1 x_2} \right) - 1 \leq 0 \\ & && f_2(x) = c_2 \sqrt{1 + x_2^2} \left(\frac{8}{x_1} - \frac{1}{x_1 x_2} \right) - 1 \leq 0 \\ & && 0.2 \leq x_1 \leq 4.0, \\ & && 0.1 \leq x_2 \leq 1.6, \end{aligned}$$

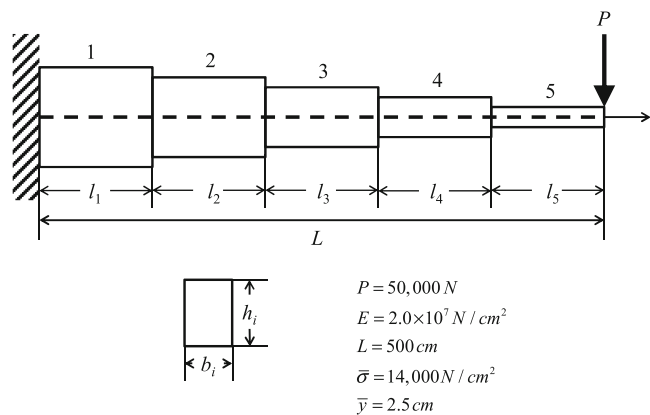
where $c_1 = 1.0$ and $c_2 = 0.124$.

The initial point is $x^0 = (1.0, 1.0)^T$.

A.2 Vanderplaats' cantilever beam

This problem was proposed by Vanderplaats (1984). The optimization problem is to minimize the volume of a p -segmented cantilever beam with constraints on the maximum stress of each segment and the tip deflection. A cantilever beam with five segments is depicted below. The cantilever beam has a rectangular cross-section, and the width and height of each segment are selected as design variables. The optimization problem is formulated as

$$\begin{aligned} & \underset{\mathbf{b}, \mathbf{h}}{\text{minimize}} && f_0(\mathbf{b}, \mathbf{h}) = \sum_{i=1}^p b_i h_i l_i \\ & \text{subject to} && f_j(\mathbf{b}, \mathbf{h}) = \frac{\sigma_i}{\bar{\sigma}} - 1 \leq 0 \quad j = 1, \dots, p, \\ & && f_{p+j}(\mathbf{b}, \mathbf{h}) = h_i - 20b_i \leq 0 \quad j = 1, \dots, p, \\ & && f_{2p+1}(\mathbf{b}, \mathbf{h}) = \frac{y_p}{\bar{y}} - 1 \leq 0, \\ & && 1.0 \leq b_i \leq 100, \\ & && 5.0 \leq h_i \leq 100. \end{aligned}$$



A.3 Svanberg's five-variate cantilever beam

This optimization problem was proposed by Svanberg (1987). The sizing design variables are considered to minimize weight subject to a single displacement constraint. The optimization problem is formulated as

$$\begin{aligned} & \underset{x}{\text{minimize}} && f_0(x) = c_1 (x_1 + x_2 + x_3 + x_4 + x_5) \\ & \text{subject to} && f_1(x) = 61/x_1^3 + 37/x_2^3 + 19/x_3^3 \\ & && \quad + 7/x_4^3 + 1/x_5^3 - c_2 \leq 0, \\ & && 0.001 \leq x_i \leq 10, \quad i = 1, 2, 3, 4, 5 \end{aligned}$$

The initial point is $x^0 = (5.0, 5.0, 5.0, 5.0, 5.0)^T$.

A.4 Fleury's weight-minimization-like problem

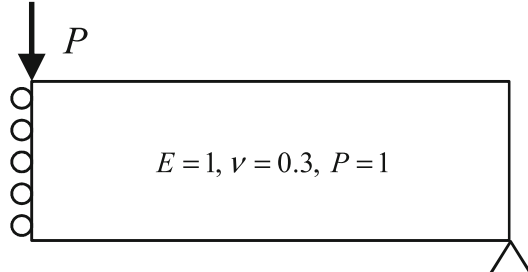
This problem was proposed by Fleury (1979). The optimization problem is formulated as

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} \quad f_0(\mathbf{x}) = \sum_{i=1}^{1,000} x_i, \\ & \text{subject to} \quad f_1(\mathbf{x}) = \sum_{i=1}^{950} \frac{1}{x_i} + 10^{-6} \sum_{i=951}^{1,000} \frac{1}{x_i} - 1,000 \leq 0, \\ & \quad f_2(\mathbf{x}) = \sum_{i=1}^{950} \frac{1}{x_i} - 10^{-6} \sum_{i=951}^{1,000} \frac{1}{x_i} - 900 \leq 0, \\ & \quad 10^{-6} \leq x_i \leq 10^6, \quad i = 1, \dots, 1000. \end{aligned}$$

The initial point is $x_i^0 = 10^{-5}$. The optimum point is known to be $x_i^* = 1$ for $i = 1, 2, \dots, 950$ and $x_i^* = 10^{-6}$ for $i = 951, 952, \dots, 1000$ with $f_0(\mathbf{x}^*) = 950.0005$. We set the move limit of the first outer iteration to 0.01 as Groenwold and Etman (2010a) did.

A.5 MBB beam topology optimization

The classical topology optimization problem for the Messerschmitt-Bolkow-Blöhm (MBB) beam to minimize compliance with a volume constraint is shown below. The design domain is discretized by the plane stress element, and the popular solid isotropic material with penalization (SIMP) scheme is used for this problem. A detailed description of the MBB problem is given by Bendsøe and Sigmund (2003).



The optimization problem can be stated as

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} \quad f_0(\mathbf{x}) = \mathbf{u}^T \mathbf{K} \mathbf{u} = \sum_{e=1}^n x_e^p \mathbf{u}_e^T \mathbf{k}_e \mathbf{u}_e \\ & \text{subject to} \quad f_1(\mathbf{x}) = \sum_{e=1}^n x_e - f V_0 \leq 0, \\ & \quad \mathbf{K} \mathbf{u} = \mathbf{f}, \\ & \quad 0.001 \leq x_e \leq 1, \end{aligned}$$

where x_e , \mathbf{u} , \mathbf{K} , and \mathbf{f} represent the e th design variable representing density, the displacement vector, the global stiffness matrix assembled by an interpolated element stiffness matrix $x_e^p \mathbf{k}_e$, and the external force applied, respectively.

The e th element displacement vector and desired volume fraction are denoted as \mathbf{u}_e and f , respectively. The sensitivity of the objective function with respect to the e th design variable can be represented as

$$\frac{df_0}{dx_e} = -\frac{p}{\rho_e} \mathbf{u}_e^T \mathbf{k}_e \mathbf{u}_e$$

For finite element analysis, the standard element known as 'Q4' is used, a four-node membrane based on isoparametric displacement. Furthermore, to avoid the checkerboard problem, the sensitivity filter proposed by Sigmund (1997) is used with the filter radius set to 8 % of the beam height.

References

- Alexandrov NM, Dennis JE, Lewis RM, Torczon V (1998) A trust-region framework for managing the use of approximation models in optimization. *Struct Optim* 15:16–23
- Bendsøe MP, Sigmund O (2003) *Topology optimization: theory, methods and applications*. Springer, Berlin; New York
- Bruyneel M, Duysinx P, Fleury C (2002) A family of MMA approximations for structural optimization. *Struct Multidiscip O* 24:263–276
- Chickermane H, Gea HC (1996) Structural optimization using a new local approximation method. *Int J Numer Meth Eng* 39:829–846
- Conn AR, Gould NIM, Toint PL (2000) *Trust-region methods*. PA, Philadelphia
- Duysinx P, Nguyen VH, Bruyneel M, Fleury C (2001) Estimating diagonal second order terms in structural approximations with quasi-Cauchy techniques. In: 4th World congress of structural and multidisciplinary optimization, Dalian China
- Fadel GM, Riley MF, Barthelemy JM (1990) Two point exponential approximation method for structural optimization. *Struct Optim* 2:117–124
- Falk JE (1967) Lagrange Multipliers and Nonlinear Programming. *J Math Anal Appl* 19:141–&
- Fletcher R, Gould NIM, Leyffer S, Toint PL, Wachter A (2003) Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming. *Siam J Optimiz* 13:635–659
- Fletcher R, Leyffer S, Toint PL (2002) On the global convergence of a filter SQP algorithm. *Siam J Optimiz* 13:44–59
- Fleury C (1979) Structural weight optimization by dual methods of convex programming. *Int J Numer Meth Eng* 14:1761–1783
- Fleury C (1989) CONLIN: an efficient dual optimizer based on convex approximation concepts. *Struct Optim* 1:81–89
- Fleury C, Braibant V (1986) Structural Optimization - a New Dual Method Using Mixed Variables. *Int J Numer Meth Eng* 23:409–428
- Groenwold AA, Etman LFP (2010) A quadratic approximation for structural topology optimization. *Int J Numer Meth Eng* 82:505–524
- Groenwold AA, Etman LFP (2010a) On the conditional acceptance of iterates in SAO algorithms based on convex separable approximations. *Struct Multidiscip O* 42:165–178
- Groenwold AA, Etman LFP, Snyman JA, Rooda JE (2007) Incomplete series expansion for function approximation. *Struct Multidiscip O* 34:21–40

- Groenwold AA, Etman LFP, Wood DW (2010) Approximated approximations for SAO. *Struct Multidiscip O* 41:39–56
- Groenwold AA, Wood DW, Etman LFP, Tosserams S (2009) Globally convergent optimization algorithm using conservative convex separable diagonal quadratic approximations. *Aiaa J* 47:2649–2657
- Kelley CT (1999) *Iterative methods for optimization*. SIAM, Philadelphia
- Kim JR, Choi DH (2008) Enhanced two-point diagonal quadratic approximation methods for design optimization. *Comput Method Appl M* 197:846–856
- Park SH, Choi DH (2011) A new convex separable approximation based on two-point diagonal quadratic approximation for large-scale structural design optimization. In: 9th World congress on structural and multidisciplinary optimization. Shizuoka, Japan
- Sigmund O (1997) On the design of compliant mechanisms using topology optimization. *Mech Struct Mach* 25:493–524
- Svanberg K (1987) The Method of moving asymptotes - a new method for structural optimization. *Int J Numer Meth Eng* 24:359–373
- Svanberg K (2002) A class of globally convergent optimization methods based on conservative convex separable approximations. *Siam J Optimiz* 12:555–573
- Vanderplaats GN (1984) *Numerical optimization techniques for engineering design: with applications*. McGraw-Hill, New York
- Wang LP, Grandhi RV (1994) Efficient safety index calculation for structural reliability-analysis. *Comput Struct* 52:103–111
- Wang LP, Grandhi RV (1995) Improved two-point function approximation for design optimization. *Aiaa J* 33:1720–1727
- Wang LP, Grandhi RV (1996) Multipoint approximations: Comparisons using structural size, configuration and shape design. *Struct Optimization* 12:177–185
- Wang LP, Grandhi RV, Canfield RA (1996) Multivariate hermite approximation for design optimization. *Int J Numer Meth Eng* 39:787–803
- Xu G, Yamazaki K, Cheng GD (2000) A new two-point approximation approach for structural optimization. *Struct Multidiscip O* 20:22–28
- Xu SQ, Grandhi RV (1998) Effective two-point function approximation for design optimization. *Aiaa J* 36:2269–2275
- Zhang WH, Fleury C (1997) A modification of convex approximation methods for structural optimization. *Comput Struct* 64:89–95
- Zillober C (1993) A globally convergent version of the method of moving asymptotes. *Struct Optimization* 6:166–174
- Zillober C (2001) A combined convex approximation-interior point approach for large scale nonlinear programming. *Optim Eng* 2:51–73
- Zillober C, Schittkowski K, Moritzen K (2004) Very large scale optimization by sequential convex programming. *Optim Method Softw* 19:103–120