## Advances in Engineering Software 64 (2013) 1-10

Contents lists available at SciVerse ScienceDirect

# Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

## An object-oriented graphics interface design and optimization software for cross-sectional shape of automobile body

## Wenjie Zuo\*

College of Mechanical Science and Engineering, Jilin University, Changchun 130025, PR China

#### ARTICLE INFO

Article history: Received 11 December 2012 Received in revised form 18 March 2013 Accepted 13 April 2013 Available online 8 June 2013

Keywords: Automobile body Thin-walled beam Cross section Shape optimization Genetic algorithm Object-oriented programming

## ABSTRACT

At the conceptual design stage, automobile body is evaluated by simplified frame structure, consisting of thin-walled beams (TWBs). In the automobile practice, design engineers mostly rely on their experience and intuition when making decisions on cross-sectional shape of TWBs. So this paper presents a cross-sectional shape optimization method in order to achieve a high stiffness and lightweight TWB. Firstly, cross-sectional property formulations is summarized and reviewed. Secondly, we build up a shape optimization model to minimize the cross-sectional area and satisfy the stiffness and manufacturing demands. The objective and constraints are nonlinear polynomial functions of the point coordinates defining the cross-sectional shape. Genetic algorithm is introduced to solve this nonlinear optimization problem. Thirdly, object-oriented programming and design patterns are adopted to design and implement the software framework. Lastly, numerical example is used to verify the presented method. This software, "SuperBeam" for short, is released for free and does speed up the conceptual design of automobile body.

© 2013 Elsevier Ltd. All rights reserved.

## 1. Introduction

Thin-walled beams (TWBs) belong to a kind of lightweight structures, owing to the high stiffness to mass ratio. At the conceptual design stage of automobile body, no sufficient computeraided-design (CAD) geometry data is available; moreover, the level of detail is not yet necessary in the design process. Therefore, the body-in-white (BIW) structure is usually evaluated by a simplified thin-walled frame consisting of TWBs with complex shape section [1], as shown in Fig. 1. The simplified model concentrates on its specific merit to offer early-stage predictions about the global performances of mass and stiffness for BIW.

TWBs of automotive body are fabricated from several pressformed metal sheets, which are spot-welded together along two edges [2]. These TWBs mainly carry on bending and torsional loadings so automobile design engineers are eager to maximize the second and torsional moments of area of cross section, respectively. On the other hand, the cross-sectional area should be minimized to reduce the mass of TWBs. In the automobile practice, design engineers mostly rely on their experience and intuition when making decisions on cross-sectional shape. So the most important issues in the above conceptual design stage are to determine the optimal cross-sectional shapes of these TWBs. Mathematically, it is a typically optimization problem.

E-mail address: zuowenjie@jlu.edu.cn

To date, the subject of cross-sectional optimization has achieved some progresses. Two kinds of optimization methods, i.e. topology and shape optimization are developed, respectively.

Kim and Kim [3] presented the first successful applications of the topology optimization for the design of TWB's section. In succession, they took into account a distortional rigidity to optimize the topological shape of cross section [4]. Griffiths and Miles [5] adopted genetic algorithm (GA) to solve the topology optimization of cross section. Liu et al. [6] formulated a section topology optimization method based on an anisotropic beam theory considering sectional warping of and coupling among deformations. Vinot [7] optimized the shape of thin-walled structures having a beam-like dynamic behavior. Apostol [8] developed a general approach for detailed analysis and design optimization of arbitrary cross-sections of truss/beam built up structures. Previous studies have mostly focused only on the reduction of cross-sectional area and little attention has been paid to the issue of whether an effective method is able to consider the fabricating demands of metal sheet, such as extrusion or stamping constraints.

Much effort, Yoshimura et al. [2] have been devoted to include more manufacture demands. They proposed a cross-sectional shape generating method to achieve the cross-sectional properties preassigned by design engineers. The cross-sectional shape-generating problem for stamped metal sheets is formulated as a multiobjective optimization problem. Cross-sectional properties (crosssectional area A, second moments of area  $I_y$  and  $I_z$ , torsional rigidity J, etc.) were all regarded as objective functions. However, in the





CrossMark

ENGINEEBING

<sup>\*</sup> Tel./fax: +86 431 85095834.

<sup>0965-9978/\$ -</sup> see front matter @ 2013 Elsevier Ltd. All rights reserved. http://dx.doi.org/10.1016/j.advengsoft.2013.04.003



Fig. 1. BIW frame structure and its thin-walled cross section.

automobile practice, the demands of second moments of area and torsional rigidity are known. Design engineers commonly change the cross-sectional shape to minimize the cross-sectional area under those demands. The method proposed in this paper fulfills these needs. In addition, another aim is to develop an engineering software, named "SuperBeam", which will rapidly design and optimize the cross-sectional shape of TWB.

The outline of this paper is organized as follows: firstly, crosssectional property formulations are summarized and reviewed; secondly, optimization model with fabricating constraints is built up and solved by GA; thirdly, the framework of this software is designed using object-oriented programming technology and design patterns; lastly, numerical example is introduced to verify the presented method.

## 2. Summary of cross-sectional property formulations

Fig. 2 shows a sample of the cross section of a TWB. As demonstrated in the figure, three press-formed sheet metal parts are spotwelded together along two edges. Every sheet can be regarded as consisting of rectangle segments. So the cross-sectional area is

$$A = \sum_{i=1}^{n} A_i = \sum_{i=1}^{n} l_i t_i$$
(1)

where *n* is the number of segment;  $l_i$  and  $t_i$  are, respectively, the length and thickness of the *i*th segment.

The centroid of the section is the weighted average of the centroid of segments, i.e.

$$c_y = \frac{1}{A} \sum_{i=1}^{n} y_{ci} A_i$$
 and  $c_z = \frac{1}{A} \sum_{i=1}^{n} z_{ci} A_i$  (2)

where  $(y_{ci}, z_{ci})$  represents the coordinate of the middle center of the *i*th segment, as shown in Fig. 2.

Then, second moments of area  $I_y$ ,  $I_z$  and product moment of area  $I_{yz}$  with respect to the centroid may be, respectively, defined as



Fig. 2. A sketch of shape optimization model of cross section.

$$I_y = \sum_{i=1}^n \left[ \left( \frac{l_i t_i^3}{12} \right) \cos^2 \theta_i + \left( \frac{l_i^3 t_i}{12} \right) \sin^2 \theta_i + l_i t_i y_{ci}^2 \right]$$
(3)

$$I_z = \sum_{i=1}^n \left[ \left( \frac{l_i t_i^3}{12} \right) \sin^2 \theta_i + \left( \frac{l_i^3 t_i}{12} \right) \cos^2 \theta_i + l_i t_i z_{ci}^2 \right]$$
(4)

$$I_{yz} = \sum_{i=1}^{n} \left[ \left( \frac{l_i^3 t_i - l_i t_i^3}{24} \right) \sin 2\theta_i + l_i t_i y_i z_{ci} \right]$$
(5)

where  $\theta_i$  is angle between the positive *y* axis and the *i*th segment. From  $I_y$ ,  $I_z$  and  $I_{yz}$ , principal moments of area are derived that

$$I_{\max} = \frac{1}{2}(I_y + I_z) + \sqrt{\frac{1}{2}(I_y - I_z)^2 + I_{yz}^2}$$
(6)

$$I_{\min} = \frac{1}{2}(I_y + I_z) - \sqrt{\frac{1}{2}(I_y - I_z)^2 + I_{yz}^2}$$
(7)

The procedure for calculating the torsional rigidity depends on the cross-sectional shape. The torsional rigidity of open section and single cell are given by

$$J_{open} = \sum_{i=1}^{o} l_i t_i^3 \tag{8}$$

$$J_{closed} = \frac{4A_0^2}{\sum_{i=1}^c \frac{l_i}{t_i}} \tag{9}$$

where o and c are the number of segments for open section and single cell, respectively;  $A_0$  is area enclosed by the mean perimeter.

When a section consists of  $n_0$  open sections and one single cell, the torsional rigidity can be written as

$$J = n_0 J_{open} + J_{closed} \tag{10}$$

On the other hand, when a reinforcement segment exists as shown in Fig. 2, the torsional rigidity of double cell is calculated as follows [2]:

$$J_{double} = \frac{4\left\{A_{u}^{2}(L_{l}/t_{l} + L_{r}/t_{r}) + 2A_{u}A_{l}L_{r}/t_{r} + A_{l}^{2}(L_{u}/t_{u} + L_{r}/t_{r})\right\}}{L_{u}L_{l}/t_{u}t_{l} + L_{l}L_{r}/t_{l}t_{r} + L_{r}L_{u}/t_{r}t_{u}}$$
(11)

where  $A_u$  is the area enclosed by the upper sheet and reinforcement,  $A_l$  is the area enclosed by the lower sheet and reinforcement,  $L_u$ ,  $L_l$ , and  $L_r$  are, respectively, the length of the upper sheet, lower sheet and reinforcement;  $t_u$ ,  $t_l$  and  $t_r$  are, respectively, the thickness of the upper sheet, lower sheet and reinforcement.

Furthermore, when a section consists of  $n_0$  open sections and one double cells, the torsional rigidity can be expressed as

$$J = n_0 J_{open} + J_{double} \tag{12}$$



Fig. 3. Five types of cross sections and their formulations of torsional rigidity J.

In short, the formulations of second moments of area  $I_y$  and  $I_z$  for the five types of cross sections are the same, i.e., Eqs. (3) and (4), respectively. However, the formulations of the torsional rigidity for the five types of cross sections are totally different, as shown in Fig. 3.

#### 3. Cross-sectional shape optimization model using GA

In the structural design of automobile body, when the stiffness of automobile body meets design requirements, the weight of that should be as light as possible. Thus, second moment of area and torsional rigidity of every beam section should be greater than specified value; besides, the cross-sectional area is minimal.

To formulate the shape generation problem as an optimization problem the cross-sectional shape of the metal sheets is represented using a number of shape control points, as shown in Fig. 2. Points on section are classified as moving points and fixed points. The coordinates of moving points are represented as  $(y_i, z_i)$ for i = 1, 2, 3, ..., m, where *m* indicates the number of moving points. Moving points are the design variables of the optimization problem, but fixed points, determined by geometric layout requirements, are constant during the optimization process. Take the Apillar of a vehicle as an example to explain the definition of fixed points. The A-pillar is the first pillar of the passenger compartment, generally enclosing the windscreen. The A-pillars play an important part in protecting the occupants in a roll-over incident. Hence the width of A-pillars has been increased over the past years. However this has caused the opposite effect of increasing the size of blind spots in the driver's vision. Consequently, the width of A-pillar is usually controlled by two fixed points. In addition, the number of moving points is also fixed and not a variable during the optimization process.

Moreover, difficulty arises from manufacture demands. Since the sheet metal is formed using an extrusion process, draft angle must be larger than 90°, as shown in Fig. 4a; when assembling sheets for spot welding, the intersection of sheets has no practicability as shown in Fig. 4b. These two constraints must not be violated during the optimization process.

So, the mathematical form of the shape optimization problem for cross section may be defined as

$$\min A = \sum_{i=1}^{n} l_i t_i \tag{13}$$

$$s.t.\begin{cases} y_{i\min} \leqslant y_i \leqslant y_{i\max} \\ z_{i\min} \leqslant z_i \leqslant z_{i\max} & i = 1, 2, 3, \cdots, n_{mp} \\ I_y \geqslant I_{y,allowable} \\ I_z \geqslant I_{z,allowable} \\ J \geqslant J_{allowable} \\ \phi_j \geqslant \phi_{allowable} = \pi/2j = 1, 2, 3, \cdots, n_{da} \\ n_{isp} = 0 \end{cases}$$

$$(14)$$

where  $y_{imin}$  and  $y_{imax}$  are, respectively, the lower and upper bounds of *y* coordinate for the *i*th point;  $z_{imin}$  and  $z_{imax}$  are, respectively, the lower and upper bounds of *z* coordinate for the *i*th point;  $n_{mp}$  is the number of moveable points;  $I_{y,allowable}$ ,  $I_{z,allowable}$  and  $J_{allowable}$  are, respectively, the specified lower bound of second moment of area and torsional rigidity;  $\varphi_j$  is the *j*th draft angle and  $n_{da}$  is number of draft angles that can potentially be less than 90°;  $n_{isp}$  is the number of intersection points.

Eqs. (13) and (14) are nonlinear optimization problem. So far, the effective strategies for this problem are the sequential linear programming (SLP) and evolutionary algorithms (EAs). EA includes GA, particle swarm optimization and ant colony algorithm, etc. SLP needs the following key steps: linear approximation, sensitivity analysis, move limit adjustment [9]. However, the sensitivity analysis of responses  $I_{y}$ ,  $I_{z}$  and J with respect to design variables governed by Eqs. (2)-(11) is too complicated to derive the explicit expressions. More importantly, manufacture constraints, as shown in Fig. 4, may not be explicitly expressed by formulas, so sensitivity analysis from responses of draft angle and intersection with respect to design variables cannot be obtained. In addition, the oscillations of objective function may occur if the move limit is not appropriately controlled. In general, SLP is more efficient and faster for nonlinear optimization than EA but sensitive to the initial design variables and easily trapped into local and infeasible solution. GA is a classical EA, inspired by natural evolution, hereditary and survival of the fittest. It is particularly effective for discontinuous, non-linear, global and paralleled optimization problems [10]. Consequently, we choose GA to overcome the above difficulties in this







Fig. 5. Static class diagram of the solver for cross-sectional property.



Fig. 6. Static class diagram of the cross-sectional design.

paper. Although GA requires numerous iterative analyses in achieving convergence, Eqs. (1)–(11) are all polynomial whose computational cost is very small. Therefore, GA is appropriate for this optimization problem.

To evaluate the fitness of GA, the constrained problem should be converted to an unconstrained problem. Hence, a penalty function is added to the objective function, which considers the normalization of violated constraints with respect to the allowable constraints. The penalty functions can be written as

$$p(I_{y}) = \begin{cases} c_{ly}(I_{y,allowable} - I_{y})/I_{y,allowable} & \text{for } I_{y} < I_{y,allowable} \\ 0 & \text{otherwise} \end{cases}$$
(15)

where  $c_{ly}$  is a penalty coefficient, assigned the value  $10^3$ . The mathematical expressions of penalty functions for  $I_z$ , J, and  $\varphi_j$  constraints are similar to Eq. (15). And it is should be noted that the penalty function of the number of intersection points  $n_{isp}$  can be defined as

$$p(n_{isp}) = \begin{cases} c_{isp} \times n_{isp} & \text{for } n_{isp} > 0\\ 0 & \text{for } n_{isp} = 0 \end{cases}$$
(16)

where  $c_{isp}$  is a penalty coefficient, assigned the value 10<sup>4</sup>.  $c_{isp}$  is larger than  $c_{ly}$ , it is because intersection points must be avoided in extrusion process. Consequently, the fitness of GA is evaluated using roulette section method [10] by the following equation

Fitness = 
$$\frac{1}{f} = \frac{1}{A/A_0 + p(I_y) + p(I_z) + p(J) + \sum_{j=1}^{n_{da}} p(\phi_j) + p(n_{isp})}$$
 (17)

where  $A_0$  is the initial cross-sectional area to normalize A.

When  $A/A_0 < 1$ ,  $p(I_y) = 0$ ,  $p(I_z) = 0$ , p(J) = 0,  $p(\varphi_j) = 0$  and  $p(n_{isp}) = 0$ , an improved solution is obtained. In this case, f < 1. So the sufficient condition to obtain improved solution is



Fig. 7. The main form of cross-sectional design.



Fig. 8. Static class diagram of the cross-sectional shape optimization.

$$Fitness = \frac{1}{f} > 1$$
 (18)

This advantage is verified in the following numerical example.

## 4. Object-oriented software development

The object-oriented programming (OOP) may be regarded as a collection of interacting objects, as opposed to the conventionally procedure-oriented programming. In OOP, each object is capable of receiving messages, processing data, and sending messages to

other objects, such as applications in engineering software development [11–13]. The object-oriented paradigm is composed of the three major virtues: encapsulation, inheritance and polymorphism. These advantages help improving the software code reusability, modifiability, maintainability. In this paper, .NET framework is selected as the development platform.

SuperBeam is an integrated graphics design, analysis and optimization environment that consists of three main parts: the solver of cross-sectional property, the cross-sectional shape design form and the cross-sectional shape optimization. The graphical user interface (GUI) has been mostly developed using Visual Basic.NET



Fig. 9. The configurations of the cross-sectional optimization software.



Fig. 10. Initial cross section with variable boundary.

2008 for the design of form, menu, tool bar, dialog boxes and so on. Graphics display of cross section is implemented using Microsoft's graphics device interface (GDI+). Solver and GA optimization are programmed by OOP. The Unified Modeling Language (UML) is used to draw the classes and their relationships such as generalization, dependency, aggregation and association. Appropriate design patterns are identified and introduced in the framework design to promote communication and system expansion. In order to present class diagrams clear, the following simplifications are used: (1) Only the most important member variables and methods (functions) of a class are shown; (2) The arguments and return value types of a method and the constructor of a class are omitted. (3) Virtual base classes are shown in italics. The system components making full use of .NET features are demonstrated as follows.

#### 4.1. Solver for cross-sectional property

All the formulations in Section 2 are calculated according to the different types of cross sections. Five types of cross sections are inherited from the virtual base class Cell, and the strategy design pattern [14] is introduced to define a family of algorithms solving cross-sectional properties, encapsulates each one, and dispatch polymorphic behavior, as shown in Fig. 5. Every cross section has its own torsional rigidity Eqs. 8, 10, and 11. If we do not adopt polymorphism programming of OOP, we have to instantiate these derived classes at client. However, strategy design pattern lets the torsional rigidity vary independently from derived classes. Therefore, users are only familiar with the CrossSection class and the reusability of codes is improved. Moreover, the relationship of *Cell*, Sheet, Segment and Point classes are also described in Fig. 4. Sheet class presents the upper sheet, lower sheet or reinforcement, depicted in Fig. 2. Every sheet aggregates several segment objects, which consists of two point objects, respectively. CrossSection class is next instantiated in the cross-sectional design windows form.

#### 4.2. Cross-sectional design based GDI+

GDI+ is applied to develop a 2D CAD system, so points and lines are drawn to enclose the cross-sectional cells. GDI+ allows us to create device-independent applications without worrying about the hardware. Lots of the GDI+ classes, including Pen, Brush, Point, Rectangle, Ellipse, Graphics and Bitmap, are referenced in this software. Fig. 6 illustrates the architecture of cross-sectional design. The GUI class is an aggregation of the Render2D class and Cross-Section class: the former is responsible for drawing points and lines to render the cross-sectional shape; the latter receives the data from drawing board, identify the cross-section types and call the corresponding solver in Fig. 5. The GUI for cross-sectional design, data management and results report are integrated into one

| Table 1                                 |       |
|---|-------|
| Design variables and optimal solution ( | (mm). |

| Points No. | Style | Y coordinate |         |             | Z coordinate |         |             | Optimal solution |              |
|------------|-------|--------------|---------|-------------|--------------|---------|-------------|------------------|--------------|
|            |       | Lower bound  | Initial | Upper bound | Lower bound  | Initial | Upper bound | Y coordinate     | Z coordinate |
| 1          | F     | -            | -47     | -           | -            | -2      | -           | -47              | -2           |
| 2          | F     | -            | -40     | -           | -            | -2      | -           | -40              | -2           |
| 3          | М     | -45          | -36     | -27         | 2            | 10      | 18          | -36.3            | 14.9         |
| 4          | Μ     | -37          | -28     | -19         | 20           | 28      | 36          | -34.1            | 24.6         |
| 5          | Μ     | -17          | -8      | 1           | 24           | 32      | 40          | 1.0              | 33.8         |
| 6          | Μ     | 7            | 16      | 25          | 22           | 30      | 38          | 19.2             | 32.3         |
| 7          | Μ     | 24           | 33      | 42          | 8            | 16      | 24          | 34.5             | 18.8         |
| 8          | Μ     | 25           | 34      | 43          | -10          | -2      | 6           | 40.7             | -3.8         |
| 9          | F     | -            | 41      | -           | -            | -7      | -           | 41               | -7           |
| 10         | F     | -            | 46      | -           | -            | -7      | -           | 46               | -7           |
| 11         | F     | -            | -40     | -           | -            | -2      | -           | -40              | -2           |
| 12         | Μ     | -47          | -38     | -29         | -37          | -29     | -21         | -38.9            | -24.6        |
| 13         | Μ     | -22          | -13     | -4          | -43          | -35     | -27         | -15.6            | -39.9        |
| 14         | М     | 10           | 19      | 28          | -41          | -33     | -25         | 15.2             | -36.4        |
| 15         | М     | 22           | 31      | 40          | -24          | -16     | -8          | 37.1             | -13.2        |
| 16         | F     | -            | 41      | -           | -            | -7      | -           | 41               | -7           |
| 17         | F     | -            | -40     | -           | -            | -2      | -           | -40              | -2           |
| 18         | Μ     | -34          | -25     | -16         | 3            | 11      | 19          | -29.9            | 11.8         |
| 19         | Μ     | -23          | -14     | -5          | 3            | 11      | 19          | -16.0            | 11.3         |
| 20         | Μ     | -20          | -11     | -2          | -19          | -11     | -3          | -2.6             | -4.0         |
| 21         | Μ     | -11          | -2      | 7           | -35          | -27     | -19         | 5.3              | -19.0        |
| 22         | M     | 10           | 19      | 28          | -28          | -20     | -12         | 21.6             | -18.2        |
| 23         | Μ     | 10           | 19      | 28          | -16          | -8      | 0           | 23.4             | -13.9        |
| 24         | М     | 22           | 31      | 40          | -18          | -10     | -2          | 32.5             | -7.2         |
| 25         | F     | -            | 41      | -           | -            | -7      | -           | 41               | -7           |

F indicates fixed point which cannot be optimized; M is moveable point for shape optimization; "-" is null.





windows form, exhibited in Fig. 7. Cross-sectional shape can be created by mouse click or text input of dialog box.

## 4.3. Cross-sectional shape optimization

Aggregating the above classes, GA implements cross-sectional shape optimization, whose class diagram is shown in Fig. 8. The OptimizationGUI is the top-level class and composites crosssectional solver and GAOptimizer. During the optimization process, OptimizationGUI transfers the design variables to the crosssectional solver, which employs these values to calculate the objective functions and constraints and then returns the values to the GAOptimizer for fitness evaluation. The GA optimizer has been designed according to its solution steps. As well, GA should be easily extended to solve different types of problems such as discrete or continuous variables, single- or multi-objective, elitist and parallel optimization problems [10,15–18]. So, five classes Population, Selection, Crossover, Mutation, and Elitist are created as an instance in GAOptimizer. Population stores a list of design variables which include several members and methods. The initial method creates



Fig. 12. Dynamic views of cross-sectional shape and properties at some key generations.

the binary population using random generator in the class library of the .net framework. The decoding method decodes a variable from a binary string to a double value. The BinaryLength member defines the necessary length of a binary string to acquire the specified precision for continuous variables. Upper bound, lower bound, and precision are the three necessary members to declare a continuous variable. The singleton design pattern [19] is introduced here to guarantee the uniqueness of variable list for a given optimization problem and the variable list must be called at any place in the optimization process. Roulette section, uniform crossover and simple mutation are employed to select and alter the population. Elitist class selects the highest fitness of individuals for the next generation, which can avoid the removal of optimal individual during GA operations. Finally, the main form of cross-sectional optimization software is shown in Fig. 9.

#### 5. Numerical example

TWB of double cell with two opens is commonly used in automobile structure, so the initial cross section in Fig. 10 is optimized as follows. Firstly, the initial cross section is drawn by the designer. After that, lower and upper bounds of design variables are proportionally created by dragging the "Track Bar" button, as shown in Fig. 9. The initial value, lower and upper bounds of design variables are listed in Table 1. Lower and upper bounds are rectangular

 Table 2

 Comparison of cross-sectional properties between initial and optimal cross section.

| Cross<br>section    | Objective |         |         | Constraints | Number of acute |
|---------------------|-----------|---------|---------|-------------|-----------------|
|                     | A         | Iy      | Iz      | J           | angle           |
| Allowable<br>values | -         | 512,000 | 778,000 | 920,000     | 0               |
| Initial             | 1026      | 486,930 | 739,950 | 875,431     | 1               |
| Optimal             | 1003      | 543,281 | 782,694 | 1,041,272   | 0               |
| Improved<br>rate    | 2.24%     | 11.57%  | 5.78%   | 18.94%      | 100%            |

boundary as shown in Fig. 10, in which moveable point successively changes its (y, z) coordinate during GA optimization. Points 2, 11 and 17 are spot-welded together as the first welding spot; meanwhile, points 9, 16 and 25 become the second one. The two welding spots are fixed points. Moreover, points 1 and 10 are also fixed to keep the length of flanging so as to satisfy the demand of welding manufacture. The thickness of upper sheet, lower sheet and reinforcement are 3.0, 3.0 and 2.0, respectively.

In the following, the constraints are planned to increase by 5% with respect to those of initial cross section, i.e.,  $I_{y,allowable} = 512,000$ ,  $I_{z,allowable} = 778,000$  and  $J_{allowable} = 920,000$ , respectively. In practice, these three allowable values are known, generally originated from benchmarking experiment of automobile body. The draft angle at point 23 is less than 90°, which must be improved in the optimization process. The configurations of GA, i.e., the number of population (NoP), the number of generations (NoG), crossover probability, and mutation probability are, respectively, 40, 20, 0.7, 0.1, input by the GUI of the solver, as shown in Fig. 9.

After 20 generations evolved, fitness in Eq. (17) is gradually enhanced and converged, as demonstrated in Fig. 11. It is important to note that turning point occurs at the fifth generation, where fitness exceeds one, so after that all constraints are satisfied and the cross-sectional area continues to reduction. During the optimization process, outstanding individuals are gradually elite-preserved and the cross-sectional properties are enhanced. The cross-sectional shape and properties at the 1, 3, 4, 5, 12 and 20 generation are described in Fig. 12. As can be seen from Fig. 12 that: (1) In figure (a–c), the number of draft angle less than 90° (point 12 in figure (a), point 21 in figure (b)) is penalized from one to zero. (2) In figure (d–f), all the constraints in Eq. (14) are satisfied and objective function in Eq. (13) is decreased generation by generation.

Optimal solution of (y, z) coordinates is listed in Table 1 and the corresponding cross-sectional properties compared with those of initial cross section are presented in Table 2. In experience, there is a positive correlation between the objective of cross-sectional



Fig. 13. Probability distribution of fitness for different NoP and NoG.

area and constraints of  $I_y$ ,  $I_z$ , J and  $n_{da}$  in engineering of automobile body design. However, the proposed method successfully deals with this problem. Table 2 shows that constraints are largely improved; meanwhile, the objective is reduced. Therefore, designer can develop high stiffness and lightweight section of TWB using this method.

At last, the statistical analysis of this cross section is conducted by one hundred optimizations for different NoP and NoG on ordinary laptop with 2.5 GHz Intel Core i5 CPU and 3.25 GB memory, running under Windows XP system. The probability distribution of the fitness is shown in Fig. 13. When NoP and NoG are, respectively, assigned the value 80 and 40, 93% of the fitness is larger than one. So 93% of the optimization tests obtain the improved solution, which is proved by Eq. (18). Furthermore, the computational time, spent on the 40 generations evolution, is only 246 s, which can be completely accepted in the automobile practice. Consequently, the statistical results demonstrate that the cross-sectional optimization method is reliable and efficient.

## 6. Conclusion

The aim of this study is to develop an object-oriented graphics interface design and GA-based optimization software for cross-sectional shape of TWB in automobile body. The following conclusions can be summarized from this study:

- The mathematical model of cross-sectional shape optimization is appropriate to describe the demands of designer in automobile engineering. Stiffness and manufacture constraints can easily be adapted to the problem using a normalized penalty function, which is effective to evaluate the fitness of GA.
- The object-oriented programming and design patterns are identified and introduced in the framework design to improve the software code reusability, modifiability and maintainability. Microsoft GDI+ functions and classes implement fluently graphics display of thin-walled section. Users only need mouse click to design cross section and optimize it.
- The solutions reached by the GA are reliable and efficient. Generally, the solver takes 4 min or so to obtain an improved solution on an ordinary laptop. It is unnecessary to manually modify the optimal cross-sectional shape because all the manufacture demands are considered in the optimization model and programmed in the software.
- Additionally, my "SuperBeam" software is free for academic research and has been integrated into the "Vehicle Body – FDO" software which can be downloaded at the end of my personal website: http://mach.jlu.edu.cn/hb\_images/xygk/xssz\_ sz\_js.php?id=395.

#### Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 51205159) and the Fundamental Research Funds for the Central Universities.

#### References

- Zuo W, Li W, Xu T, Xuan S, Na J. A complete development process of finite element software for body-in-white structure with semi-rigid beam in .NET framework. Adv Eng Softw 2012;45:261–71.
- [2] Yoshimura M, Nishiwaki S, Izui K. A multiple cross-sectional shape optimization method for automotive body frames. J Mech Des 2005;127:49–57.
- [3] Kim YY, Kim TS. Topology optimization of beam cross sections. Int J Solids Struct 2000;37:477–93.
- [4] Kim TS, Kim YY. Multiobjective topology optimization of a beam under torsion and distortion. AIAA J 2002;40:376–81.

- [5] Griffiths DR, Miles JC. Determining the optimal cross-section of beams. Adv Eng Inform 2003;17:59–76.
- [6] Liu S, An S, Jia H. Topology optimization of beam cross-section considering warping deformation. Struct Multidisc Optim 2008;35:403–11.
- [7] Vinot P, Cogan S, Piranda J. Shape optimization of thin-walled beam-like structures. Thin Wall Struct 2001;39:611–30.
- [8] Apostol V, Santos JLT, Paiva M. Sensitivity analysis and optimization of truss/ beam components of arbitrary cross-section II. Shear stresses. Comput Struct 2002;80:391–401.
- [9] Lamberti L, Pappalettere C. Move limits definition in structural optimization with sequential linear programming. Part I: Optimization algorithm. Comput Struct 2003;81:197–213.
- [10] Goldberg DE. Genetic Algorithms in Search, Optimization and Machine Learning. first ed. Reading MA: Addison Wesley; 1989.
- [11] Rypl D, Patzak B. Object oriented implementation of the T-spline based isogeometric analysis. Adv Eng Softw 2012;50:137–49.
- [12] Patzak B, Rypl D. Object-oriented, parallel finite element framework with dynamic load balancing. Adv Eng Softw 2012;47:35–50.
- [13] Lejeune A, Bechet F, Boudaoud H, Mathieu N, Potier-Ferry M. Object-oriented design to automate a high order non-linear solver based on asymptotic numerical method. Adv Eng Softw 2012;48:70–88.

- [14] Gamma E, Helm R, Johnson R, Vlissides J. Design Patterns: Element of Reusable Object-Oriented Software. MA: Addison-Wesley; 1995.
- [15] Villalba JD, Laier JE. Localising and quantifying damage by means of a multichromosome genetic algorithm. Adv Eng Softw 2012;50:150–7.
- [16] Alcan P, Basligil H. A genetic algorithm application using fuzzy processing times in non-identical parallel machine scheduling problem. Adv Eng Softw 2012;45:272–80.
- [17] Zuo W, Xu T, Zhang H, Xu T. Fast structural optimization with frequency constraints by genetic algorithm using adaptive eigenvalue reanalysis methods. Struct Multidisc Optim 2011;43:799–810.
- [18] Xu T, Zuo WJ, Xu TS, Li RC. An adaptive reanalysis method for genetic algorithm with application to fast truss optimization. Acta Mechanica Sinica 2010;26:225–34.
- [19] Wang W, Rivard H, Zmeureanu R. An object-oriented framework for simulation-based green building design optimization with genetic algorithms. Adv Eng Inform 2005;19:5–23.