 CMES, vol.22, no.2, pp.97-118, 2007

A New Uncertain Optimization Method Based on Intervals and An Approximation Management Model

C. Jiang[1] **and X. Han**[1,2]

Abstract: A new uncertain optimization method is developed based on intervals and an approximation management model. A general uncertain optimization problem is considered in which the objective function and constraints are both nonlinear and uncertain, and intervals are used to model the uncertainty existing in the system. Based on a *possibility degree of interval*, a *nonlinear interval number programming* (NINP) method is proposed. A deterministic objective function is constructed to maximize the possibility degree of the uncertain objective function, and the uncertain constraints are changed into deterministic ones by introducing some possibility degree levels. If the optimal possibility degree of the objective function reaches 1.0, a robustness criterion is introduced and a corresponding robustness optimization is performed for the uncertain objective function. To improve the optimization efficiency, the NINP method is combined with an approximation management model to form an efficient uncertain optimization method. The trust region method is employed to manage a sequence of NINP problems which are based on the approximation models of the uncertain objective function and constraints within the uncertainty space and current design space. Two numerical examples are investigated to demonstrate the effectiveness of the present method.

Keyword: uncertain optimization; interval; approximation model; trust region; possibility degree of interval

[1] State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, College of Mechanical and Automotive Engineering, Hunan University, Changsha City, P. R. China 410082

[2] Corresponding author. Tel: 86-731-8823993; fax: 86-731-8821445. E-mail address: hanxu@hnu.cn

1 Introduction

In the traditional engineering optimization (Haftka and Gurdal, 1992; Sedaghati et al., 2001; Mathur et al., 2003; Tapp et al., 2004; Fedelinski and Gorski, 2006; Amirante et al., 2007; Lamberti and Pappalettere, 2007), the system parameters are always treated as deterministic values. However, uncertainties in geometric dimensions, material properties, loads, boundary conditions and etc widely exist in practical engineering problems. In order to obtain a reliable analysis or design, the uncertainty in the system must be considered. The probability method has been widely and successfully used to model the uncertainty, and based on it various kinds of stochastic programming methods have been developed (Charnes and Cooper, 1959; Kall, 1982; Liu and Iwamura, 1997; Liu et al., 2003; Gyeong-Mi, 2005; Abbas and Bellahcene, 2006). In these methods, the uncertain parameters are treated as random variables, and the uncertain optimization problem is transformed into a deterministic optimization problem based on the statistics theory. Using the stochastic programming method, a precondition should be satisfied that the sufficient information on the uncertainty is available for constructing the precise probability distributions of the uncertain parameters. Unfortunately, for many engineering problems, the information on the uncertainty is deficient and sometimes very expensive. For this class of problems, the stochastic programming will encounter difficulties. Additionally, there is research indicating that even a small deviation of the probability distribution may lead to a very large error of the reliability analysis (Ben-Haim and Elishakoff, 1990).

In recent years, many researchers intend to develop some new methods to model the uncertainty and whereby overcome the deficiency of the probability method, among which the interval method seems a very promising and inspiring one. *Interval* represents a closed bounded set of real numbers, and in interval mathematics (Moore, 1979), it is regarded as a *number*, namely *interval number*. Using the interval method, the lower and upper bounds of an uncertain parameter are only needed, not necessarily knowing its precise probability distribution. Thus the uncertainty description can be accomplished through a small quantity of information on the uncertainty. Based on the interval method, a kind of uncertain optimization method named *interval number programming* has been developed. The references (Tanaka et al., 1984; Rommelfanger, 1989; Ishibuchi and Tanaka, 1990) investigated the linear problems with interval coefficients in the objective function. Problem with uncertainties in both of the objective function and constraints was studied, and the possible interval of the solution was obtained by taking the maximum value range and minimum value range inequalities as constraint conditions (Tong, 1994). A fuzzy satisfactory degree of interval was constructed to deal with the uncertain constraints with interval coefficients (Liu and Da, 1999). Based on a comparative study on ordering interval numbers, a *linear interval number programming* method was developed (Sengupta et al., 2001). A new possibility degree of interval was constructed based on the probability method and whereby a multi-criteria decision problem was solved (Zhang et al., 1999). These methods all belong to the linear interval number programming method, namely the objective function and constraints are linear functions with respect to the design variables and uncertain parameters. To make the interval number programming applicable for practical engineering problems, the research on the *nonlinear interval number programming* (NINP) has been attracting more and more attentions and so far some NINP methods have been also developed. The reference (Ma, 2002) seems the first attempt to study the NINP problem, in which only the uncertain objective function was considered and the uncertain optimization was transformed into a deterministic three-objective optimization. This work was improved by the reference (Jiang et al., 2007a), in which the uncertainty in the nonlinear constraints was also considered. A neural network was employed to create an approximation model between the design variables and the bounds of the uncertain objective function and whereby an efficient NINP method was constructed (Jiang et al., 2007b). The interval analysis method was used to compute the bounds of the objective function and constraints at each iterate, and hence the time-consuming nesting optimization of the NINP was avoided (Jiang et al., 2007c and 2007d). The approximate management framework was firstly introduced into the NINP method, and through a sequential optimization process the optimal design vector could be achieved by a small amount of evaluations of the actual simulation model (Jiang et al., 2007e). In the above linear interval number programming and NINP methods, the uncertain objective function is generally changed into a deterministic multi-objective optimization problem based on an *order relation of interval*. However, solving the multi-objective optimization problem is a complicated and time-consuming task. Especially for NINP problems, the nesting optimization is generally involved, in which the outer layer optimization is used to optimize the design vector and the inner layer optimization is used to obtain the intervals of the objective function and constraints. Once the nesting optimization is coupled with the multi-objective optimization, the uncertain optimization process will be made much more complicated and difficultly treated. On the other hand, in the existing interval number programming methods we usually use two different mathematical tools to deal with the uncertain objective function and constraints, namely order relation of interval for the objective function and possibility degree of interval for the constraints. A uniform treatment method which is effective for both of the uncertain objective function and constraints is still unavailable. Thus if we could develop a new NINP method which can avoid the multi-objective problem and furthermore provide a uniform treatment form for the uncertain objective function and constraints, the uncertain optimization can be treated much more conveniently and this NINP method will be more applicable for practical engineering problems. This paper just aims to develop such an NINP method, and simultaneously ensure a high optimization efficiency for this NINP method.

In this paper, an uncertain optimization method is developed based on a new NINP method and an approximation management model. It should be declared that the research work in this paper concerning with the approximation management model is partly based on the authors' previous work (Jiang et al., 2007e). In the NINP method, the uncertain objective function is transformed into a deterministic single-objective optimization problem to maximize the possibility degree that the interval of the uncertain objective function is larger than a performance interval. The uncertain constraints are changed into deterministic constraints by giving a possibility degree level for each constraint. Then using the penalty function method, a deterministic non-constraint optimization problem can be finally obtained, which is a nesting optimization. If the optimal possibility degree of the uncertain objective function reaches 1.0, a robustness optimization is then carried out, in which the radius of the uncertain objective function is minimized and a new constraint that the possibility degree of the uncertain objective function is equal to 1.0 is added. To improve the optimization efficiency, an approximation management model is combined with the NINP method. A sequence of sub-optimization problems are generated, and for each sub-optimization the approximation models are constructed for the uncertain objective function and constraints. The NINP method is used to solve each sub-optimization problem based on the nesting optimization of the efficient approximation models. The trust region method is employed to manage the approximation models and make the iterative sequence converge at an optimal design vector. Finally, the present method is applied to a benchmark test and a practical engineering problem, and the computation results show the effectiveness of the present method.

2 Statement of the problem

A general NINP problem can be given in the following form:

$$\max_{\mathbf{X}} f(\mathbf{X},\mathbf{U})$$

subject to

$$\begin{aligned}
&g_i(\mathbf{X},\mathbf{U}) \le V_i^I = \left[V_i^L, V_i^R\right], \quad i=1,\ldots,l \\
&\mathbf{U} \in \mathbf{U}^I = \left[\mathbf{U}^L, \mathbf{U}^R\right], \\
&U_i \in U_i^I = \left[U_i^L, U_i^R\right], \quad i=1,2,\ldots,q \\
&\mathbf{X}_l \le \mathbf{X} \le \mathbf{X}_r
\end{aligned} \tag{1}$$

where f and g_i denote the objective function and the ith constraint, respectively, and they are generally computed based on the simulation models in practical applications. l is the number of the constraints. $\mathbf{X}$ is an n-dimensional design vector, and $\mathbf{X}_l$ and $\mathbf{X}_r$ denote the allowable minimum and maximum vectors of $\mathbf{X}$, respectively. $\mathbf{U}$ is a q-dimensional uncertain vector which collects all of the uncertain parameters in the system, and its uncertainty is modeled by an interval vector $\mathbf{U}^I$. The superscript I denotes an interval, and L and R denote the lower and upper bounds of the interval, respectively. V_i^I represents an allowable interval of the ith constraint. Here, the objective function f and constraints g are both nonlinear functions of $\mathbf{X}$ and $\mathbf{U}$, and continuous with respect to $\mathbf{X}$ and $\mathbf{U}$. In our problem, the uncertainty level, namely the interval of each uncertain parameter is assumed to be relatively small, which is often true for practical engineering problems.

For a specific design vector $\mathbf{X}$, the possible values of f or g_i caused by the uncertainty will form an interval, as the uncertain parameters are all intervals, and f and g are continuous with respect to $\mathbf{U}$. However, in a deterministic optimization problem, a specific design vector always corresponds to a specific value of the objective function or constraint. Thus the above NINP problem is much more difficultly treated than the deterministic optimization problems. In the following section, an uncertain optimization method will be suggested to solve the above problem based on a new NINP method and an approximation management model.

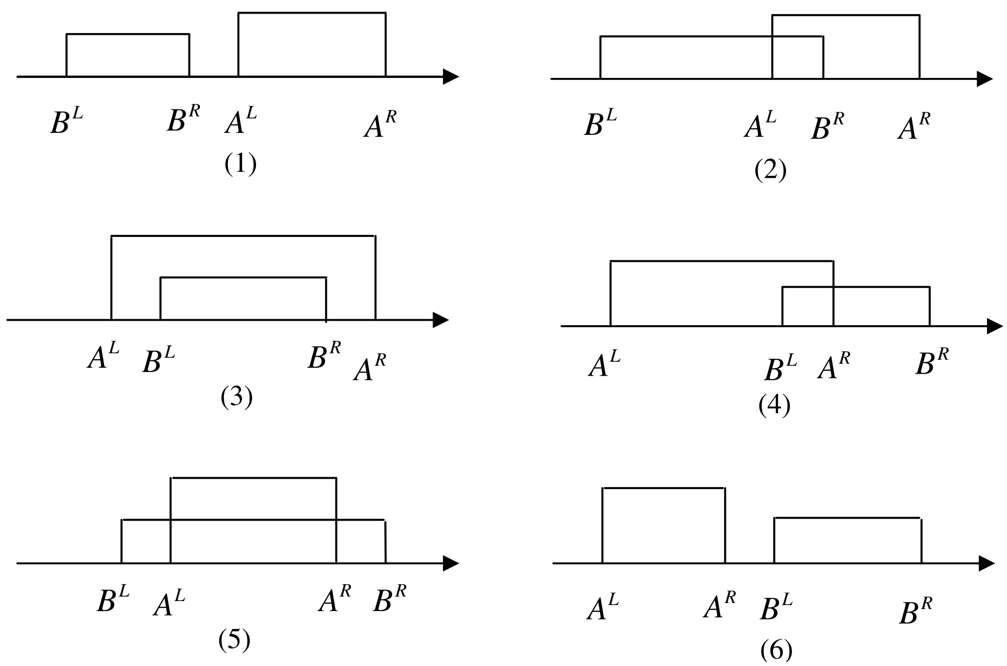

Figure 1: Six possible positional relations between intervals A^I and B^I

3 A new nonlinear interval number programming (NINP) method

3.1 A possibility degree of interval

In deterministic optimization problems, we can appraise the design vectors through the real number values of the objective function and constraints at these design vectors. However, in an NINP problem, the values of the uncertain objective function and constraints at a specific design vector are all intervals, instead of real numbers. Thus to evaluate the design vectors, a mathematical tool should be used to compare the intervals. The possibility degree of interval is such a mathematical tool, which represents an extent that one interval is larger than another. Zhang et al. (1999) proposed a construction method of possibility degree based on the probability method, and this work was improved by Jiang et al. (2007a). Comparing with the widely used possibility degrees based on the fuzzy sets (e.g. Sengupta et al., 2001), this kind of construction method can give a more intuitive and stricter mathematical explanation for the possibility degree of interval. Between intervals A^I and B^I, there exist six possible positional relations as shown in Fig. 1, and based on these relations a possibility degree $P\left(A^I \geq B^I\right)$ can be constructed (Jiang et al., 2007a):

$$P\left(A^I \geq B^I\right) = \begin{cases} 1 & A^L \geq B^R \\ \frac{A^R-B^R}{A^R-A^L} + \frac{B^R-A^L}{A^R-A^L} \cdot \frac{A^L-B^L}{B^R-B^L} & \\ \quad +0.5 \cdot \frac{B^R-A^L}{A^R-A^L} \cdot \frac{B^R-A^L}{B^R-B^L} & B^L \leq A^L < B^R \leq A^R \\ \frac{A^R-B^R}{A^R-A^L} + 0.5 \cdot \frac{B^R-B^L}{A^R-A^L} & A^L < B^L < B^R \leq A^R \\ 0.5 \cdot \frac{A^R-B^L}{A^R-A^L} \cdot \frac{A^R-B^L}{B^R-B^L} & A^L < B^L \leq A^R < B^R \\ \frac{A^L-B^L}{B^R-B^L} + 0.5 \cdot \frac{A^R-A^L}{B^R-B^L} & B^L \leq A^L < A^R < B^R \\ 0 & A^R < B^L \end{cases} \tag{2}$$

Here intervals A^I and B^I are treated as random variables $\tilde{A}$ and $\tilde{B}$ with uniform distributions, and the probability for random variable $\tilde{A}$ larger than $\tilde{B}$ is regarded as $P\left(A^I \geq B^I\right)$. In Eq. (2), $P\left(A^I \geq B^I\right) = 0$ or 1 means that interval A^I is absolutely smaller or larger than B^I. When B^I is degenerated into a real number b or A^I is degenerated into a real number a, Eq. (2) can be rewritten:

$$P\left(A^I \geq b\right) = \begin{cases} 0 & b > A^R \\ \frac{A^R-b}{A^R-A^L} & A^L < b \leq A^R \\ 1 & b \leq A^L \end{cases} \tag{3}$$

$$P\left(a \geq B^I\right)=\begin{cases}0 & a<B^L \\ \frac{a-B^L}{B^R-B^L} & B^L \leq a<B^R \\ 1 & a \geq B^R\end{cases} \tag{4}$$

In these two cases, only A^I or B^I is regarded as random variable. It can be found that $P\left(A^I \geq b\right)$ behaves a linear relation with respect to b when b is inside A^I, and similarly $P\left(a \geq B^I\right)$ behaves a linear relation with respect to a when a is inside B^I.

3.2 *Treatment of the uncertain objective function and constraints*

In stochastic optimization (e.g. Liu et al., 2003), we often make the constraints satisfied with a confidence level and transform the uncertain constraints into deterministic ones. Additionally, through maximizing the probability that the objective function satisfies the performance requirement of the system, the uncertain objective function can be also transformed into a deterministic one. In the same way, for an NINP problem, we can maximize the possibility degree of the uncertain objective function and simultaneously make each uncertain constraint satisfied with a possibility degree level, and hence Eq. (1) can be changed into the following optimization problem:

$$\max_{\mathbf{X}} P\left(F^I \geq G^I\right)$$

subject to

$$\begin{aligned} & P\left(C_i^I \leq V_i^I\right) \geq \lambda_i, \quad i=1,2,\ldots,l \\ & \mathbf{X}_l \leq \mathbf{X} \leq \mathbf{X}_r \end{aligned} \tag{5}$$

where

$$\begin{aligned} & F^I=\left[f^L(\mathbf{X}), f^R(\mathbf{X})\right], \quad G^I=\left[G^L, G^R\right] \\ & C_i^I=\left[g_i^L(\mathbf{X}), g_i^R(\mathbf{X})\right], \quad V_i^I=\left[V_i^L, V_i^R\right] \end{aligned} \tag{6}$$

In Eq. (5), F^I and C_i^I are the intervals of the objective function f and g_i caused by the uncertainty at a specific $\mathbf{X}$. G^I is an interval (termed as "performance interval") which represents a performance requirement that the objective function is expected to achieve. G^I can be also a real number, and it should be determined based on the practical problem. $P\left(F^I \geq G^I\right)$ represents the possibility degree that the objective function satisfies the performance requirement of the system under the uncertainty circumstance, and $P\left(C_i^I \leq V_i^I\right)$ represents the possibility degree that the ith constraint is satisfied. The values of $P\left(F^I \geq G^I\right)$ and $P\left(C_i^I \leq V_i^I\right)$ can be calculated through Eq. (2), (3) or (4) based on the different forms of G^I and V_i^I. $0 \leq \lambda_i \leq 1.0$ is a predetermined possibility degree level of the ith constraint. Obviously, λ can be used to adjust the feasible field of the design vector $\mathbf{X}$. A larger λ means a stricter constraint and whereby a smaller feasible field of $\mathbf{X}$.

Bounds of the intervals F^I and C_i^I can be expressed in the following form:

$$\begin{aligned} & f^L(\mathbf{X})=\min_{U\in\Gamma} f(\mathbf{X},\mathbf{U}), \qquad f^R(\mathbf{X})=\max_{U\in\Gamma} f(\mathbf{X},\mathbf{U}) \\ & g_i^L(\mathbf{X})=\min_{U\in\Gamma} g_i(\mathbf{X},\mathbf{U}), \qquad g_i^R(\mathbf{X})=\max_{U\in\Gamma} g_i(\mathbf{X},\mathbf{U}) \\ & \mathbf{U}\in\Gamma=\left\{\mathbf{U}\middle|\mathbf{U}^L\leq\mathbf{U}\leq\mathbf{U}^R\right\} \end{aligned} \tag{7}$$

Thus it can be seen that to obtain the intervals of the objective function and constraints caused by the uncertainty should involve some sub-optimization processes in the uncertainty space.

Through Eq. (7), the uncertain vector $\mathbf{U}$ is eliminated, and Eq. (5) actually becomes a deterministic optimization problem. Using the penalty function method (Chen, 2002), a non-constraint optimization problem can be obtained in terms of a penalty function f_p:

$$\max_{\mathbf{X}} f_p(\mathbf{X})= P\left(F^I \geq G^I\right)-\sigma\sum_{i=1}^{l}\varphi\left(P\left(C_i^I \leq V_i^I\right)-\lambda_i\right) \tag{8}$$

where σ is a penalty factor which is always specified as a large value, and φ is a function in the following form:

$$\varphi\left(P\left(C_i^I \leq V_i^I\right)-\lambda_i\right)= \left(\max\left(0,-\left(P\left(C_i^I \leq V_i^I\right)-\lambda_i\right)\right)\right)^2 \tag{9}$$

Equation (8) can be solved by the traditional deterministic optimization methods.

3.3 *A robustness criterion for the uncertain objective function*

Through optimizing Eq. (8), an optimal $\mathbf{X}$ can be obtained which makes $P\left(F^I \geq G^I\right)$ reach a maximal value P_m. We are likely to achieve a mostly expected value of P_m, namely $P_m = 1.0$. It means that the objective function absolutely satisfies the performance requirement, regardless of the variation of the uncertain parameters. In this case, generally, there exist a set of optimal designs which lead to $P_m = 1.0$. In theory, each one of these optimal designs satisfies our design requirement and can be selected as the final design. However, among these designs it is still possible to select a best one through introducing a second optimality criterion. In practical engineering problems, we generally expect to find a design which is insensitive to the uncertainty and whereby guarantee the robustness of the system. Therefore, here a robustness criterion is introduced, and Eq. (5) can be transformed into the following optimization problem:

$$\min_{\mathbf{X}} F^w$$

subject to

$$\begin{aligned} & P\left(F^I \geq G^I\right) = 1.0 \\ & P\left(C_i^I \leq V_i^I\right) \geq \lambda_i, \quad i = 1,2,\ldots,l \\ & \mathbf{X}_l \leq \mathbf{X} \leq \mathbf{X}_r \end{aligned} \tag{10}$$

where

$$\begin{aligned} F^w &= \frac{f^R(\mathbf{X}) - f^L(\mathbf{X})}{2} \\ &= \frac{\max\limits_{U\in\Gamma} f(\mathbf{X},\mathbf{U}) - \min\limits_{U\in\Gamma} f(\mathbf{X},\mathbf{U})}{2} \end{aligned} \tag{11}$$

F^w denotes the radius of the interval of the objective function. Through minimizing F^w, the variation of the objective function caused by the uncertainty will be decreased, namely, the optimal design can make the objective function insensitive to the fluctuation of the uncertain parameters. Thus a robust design can be ensured.

Eq. (10) can be also transformed into a non-constraint optimization problem in terms of a penalty function f'_p:

$$\begin{aligned} \min_{\mathbf{X}} f'_p(\mathbf{X}) = F^w + \sigma \sum_{i=1}^{l} \varphi\left(P\left(C_i^I \leq V_i^I\right) - \lambda_i\right) \\ + \sigma\left(P\left(F^I \geq G^I\right) - 1.0\right)^2 \end{aligned} \tag{12}$$

3.4 Difficulty of the conventional optimization methods

The transformed problems defined by Eqs. (8) and (12) are both deterministic non-constraint optimization problems. If we use a conventional optimization method to solve these two problems, the optimization process can be outlined as shown in Fig. 2. It can be found that the optimization consists of two layers in which the outer layer is used to optimize the design vector $\mathbf{X}$ and the inner layer is used to compute the bounds of the objective function and constraints caused by the uncertainty. At each iterate of $\mathbf{X}$, the inner layer optimization operator will be called 2 times to compute the lower and upper bounds of the objective function or each constraint. Furthermore each time of inner layer optimization needs an amount of evaluations of the objective function and constraints which are generally computation-intensive simulation models. Thus the nesting optimization of the actual simulation models will be caused inevitably, and it will lead to an extremely low computation efficiency which is unacceptable for most practical engineering problems.

4 Uncertain optimization based on an approximation management model

In this section, an approximation management model will be combined with the NINP to construct an efficient uncertain optimization method. In the optimization process, a series of sub-optimization problems based on the approximation models and the NINP method are created, and at the sth iterate the sub-optimization problem has the following form:

$$\max_{\mathbf{X}} P\left(\tilde{F}^I \geq G^I\right)$$

subject to

$$\begin{aligned} & P\left(\tilde{C}_i^I \leq V_i^I\right) \geq \lambda_i, \quad i = 1,2,\ldots,l \\ & \max\left[\mathbf{X}_l, \mathbf{X}^{(s)} - \mathbf{\Delta}^{(s)}\right] \leq \mathbf{X} \leq \min\left[\mathbf{X}_r, \mathbf{X}^{(s)} + \mathbf{\Delta}^{(s)}\right] \end{aligned} \tag{13}$$

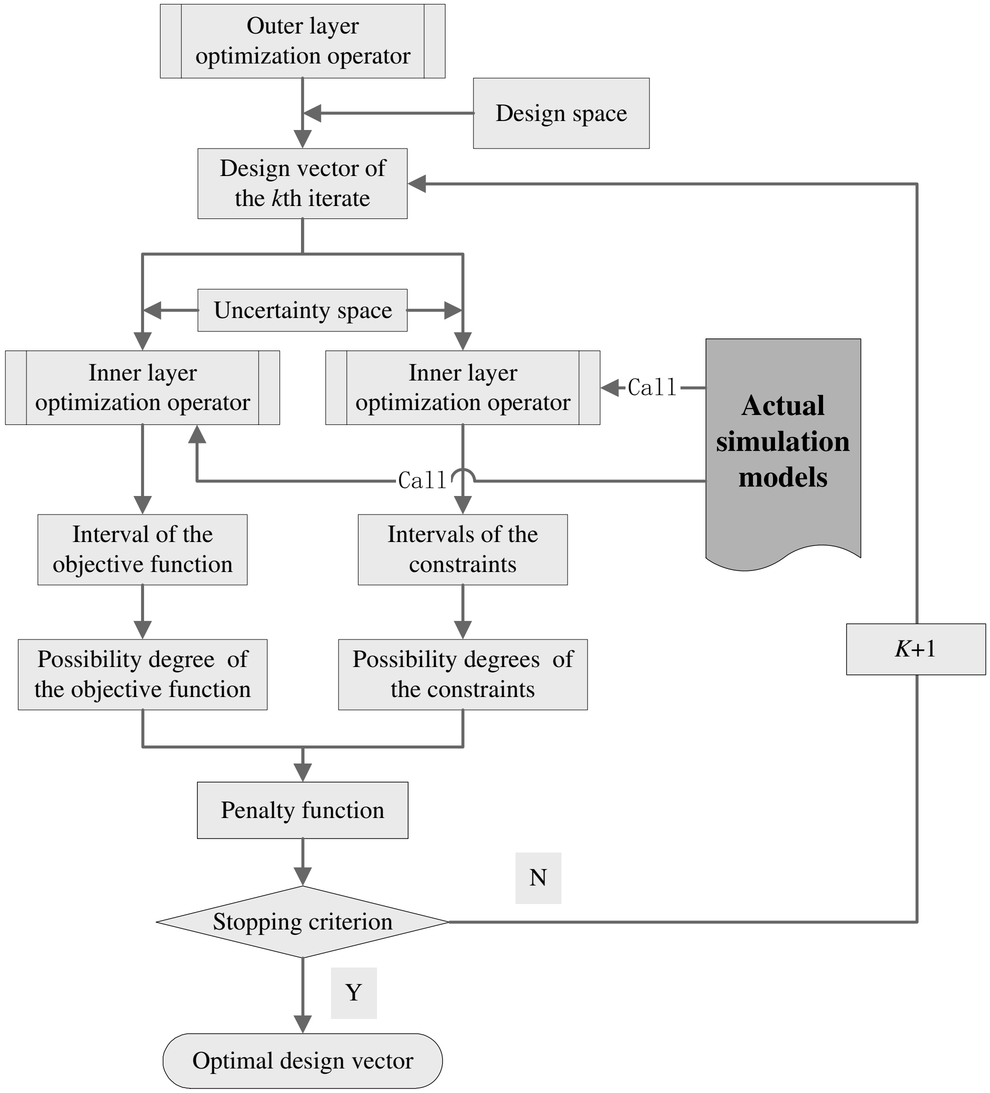

Figure 2: Nesting optimization based on the actual simulation models

where

$$
\begin{aligned}
\tilde{F}^I &= \left[\tilde{f}^L(\mathbf{X}), \tilde{f}^R(\mathbf{X})\right] \\
&= \left[\min_{U\in\Gamma} \tilde{f}(\mathbf{X},\mathbf{U}), \max_{U\in\Gamma} \tilde{f}(\mathbf{X},\mathbf{U})\right] \\
\tilde{C}_i^I &= \left[\tilde{g}_i^L(\mathbf{X}), \tilde{g}_i^R(\mathbf{X})\right] \\
&= \left[\min_{U\in\Gamma} \tilde{g}_i(\mathbf{X},\mathbf{U}), \max_{U\in\Gamma} \tilde{g}_i(\mathbf{X},\mathbf{U})\right] \\
\mathbf{U} &\in \Gamma = \left\{\mathbf{U} \middle| \mathbf{U}^L \le \mathbf{U} \le \mathbf{U}^R\right\}
\end{aligned}
\tag{14}
$$

where $\tilde{f}$ and $\tilde{g}_i$ are approximation models of the objective function and the ith constraint, respectively, and they are both explicit functions with respect to $\mathbf{X}$ and $\mathbf{U}$. $\tilde{F}^I$ and $\tilde{C}_i^I$ are intervals of the approximate objective function and ith approximate constraint, respectively. $\mathbf{\Delta}^{(s)}$ is a move limit vector which is placed on the current design vector $\mathbf{X}^{(s)}$ to ensure the approximation accuracy. $\mathbf{\Delta}^{(s)}$ varies with the proceeding of the optimization, and thus the design spaces at the iterates will not be fixed. Based on Eq. (8), Eq. (13) can be transformed into the following optimization problem:

$$
\begin{aligned}
&\max_{\mathbf{X}} \tilde{f}_p(\mathbf{X}) \\
&\qquad = P\left(\tilde{F}^I \ge G^I\right) - \sigma \sum_{i=1}^{l} \varphi\left(P\left(\tilde{C}_i^I \le V_i^I\right) - \lambda_i\right) \\
&\max\left[\mathbf{X}_l, \mathbf{X}^{(s)} - \mathbf{\Delta}^{(s)}\right] \le \mathbf{X} \le \min\left[\mathbf{X}_r, \mathbf{X}^{(s)} + \mathbf{\Delta}^{(s)}\right]
\end{aligned}
\tag{15}
$$

where $\tilde{f}_p$ is the penalty function based on the approximation models (termed as "approximate penalty function").

4.1 Construction of the approximation models

The quadratic polynomial response surface (RS) (Rodriguez et al., 2000) is used to construct the approximation models of the uncertain objective function and constraints. Considering a function $h(\mathbf{Y})$ with n_d input variables and n_s design samples, a quadratic polynomial model $\tilde{h}(\mathbf{Y})$ of the function $h(\mathbf{Y})$ at the samples is given as follows:

$$h^{(k)} = c_0 + \sum_{i=1}^{n_d} c_i Y_i^{(k)} + \sum_{i=1}^{n_d}\sum_{j=1}^{n_d} c_{ij} Y_i^{(k)} Y_j^{(k)}, \quad k = 1,2,\ldots,n_s \qquad (16)$$

where c_0, c_i and c_{ij} are coefficients of the constant term, linear term and quadratic term, respectively. $h^{(k)}$ is the actual value of h at the kth sample. $Y_i^{(k)}$ and $Y_j^{(k)}$ are values of the ith and jth input variables at the kth sample, respectively. In Eq. (16), the total number of the unknown coefficients is $n_t = (n_d+1)(n_d+2)/2$ if $c_{ij} = c_{ji}$, and hence to guarantee the proper characterization of Eq. (16) n_s should be not less than n_t. Equation (16) can be rewritten in the matrix form (Rodriguez et al., 2000):

$$\mathbf{h} = \mathbf{Mc} \qquad (17)$$

where $\mathbf{h}$ is an n_s-dimensional actual value vector of the function h, and $\mathbf{c}$ is an n_t-dimensional coefficient vector. $\mathbf{M}$ is a following $n_s \times n_t$ matrix:

$$\mathbf{M} = \begin{bmatrix} 1 & Y_1^{(1)} & Y_2^{(1)} & \cdots & \left(Y_{n_d}^{(1)}\right)^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & Y_1^{(n_s)} & Y_2^{(n_s)} & \cdots & \left(Y_{n_d}^{(n_s)}\right)^2 \end{bmatrix} \qquad (18)$$

A least-squares estimation $\tilde{\mathbf{c}}$ to $\mathbf{c}$ can be obtained:

$$\tilde{\mathbf{c}} = \left(\mathbf{M}^T\mathbf{M}\right)^{-1}\mathbf{M}^T\mathbf{h} \qquad (19)$$

Then substituting Eq. (19) into Eq. (16), the approximate value of h can be obtained at any input vector $\mathbf{Y}$. Generally, the value of $\tilde{h}$ is not equal to the true value of h at a sample point, as the coefficient vector $\tilde{\mathbf{c}}$ is achieved through the least-squares method instead of the interpolation method.

In Eq. (13), we use the above RS to construct the approximation models $\tilde{f}(\mathbf{X},\mathbf{U})$ and $\tilde{g}(\mathbf{X},\mathbf{U})$ for the objective function and constraints. In the construction process, $\mathbf{X}$ and $\mathbf{U}$ are both used as the input variables of the RS, namely here $n_d = n+q$. Therefore, $\tilde{f}(\mathbf{X},\mathbf{U})$ and $\tilde{g}(\mathbf{X},\mathbf{U})$ are both explicit functions with respect to the design vector $\mathbf{X}$ and the uncertain vector $\mathbf{U}$, instead of only $\mathbf{X}$ as we usually do for deterministic optimization problems.

In this paper, the design samples used to construct the approximation models are generated by Latin Hypercube Design (LHD) (McKay et al., 1979; Stephen et al., 2003). LHD is more accurate than random sampling and stratified sampling in estimating the means, variances and distribution functions of an output. Each input variable is ensured to have all portions of its range represented, and many input variables can be treated with a relatively low cost. Furthermore, the size of an LHD sample set can be controlled arbitrarily by the designer according to the budget, time and other conditions, and hence LHD has a capability to generate the saturated design samples.

4.2 Optimization based on the approximation models at each iterate

In our formulation, the intergeneration projection genetic algorithm (IP-GA) (Liu and Han, 2003) is used to solve the optimization problem Eq. (13), and the flowchart is shown in Fig. 3. In the uncertainty space and current design space, one set of sampling points are obtained by LHD, and each point is a vector with size $n_d = n+q$. After inputting the sampling points into the actual simulation models one by one, the samples can be obtained to construct the approximation models of the objective function and constraints using the above RS method. Then the actual simulation models can be discarded temporally, and the optimization process at current iterate can be performed only based on these approximation models. In the outer layer IP-GA, an amount of individuals of the design vector $\mathbf{X}$ are generated,

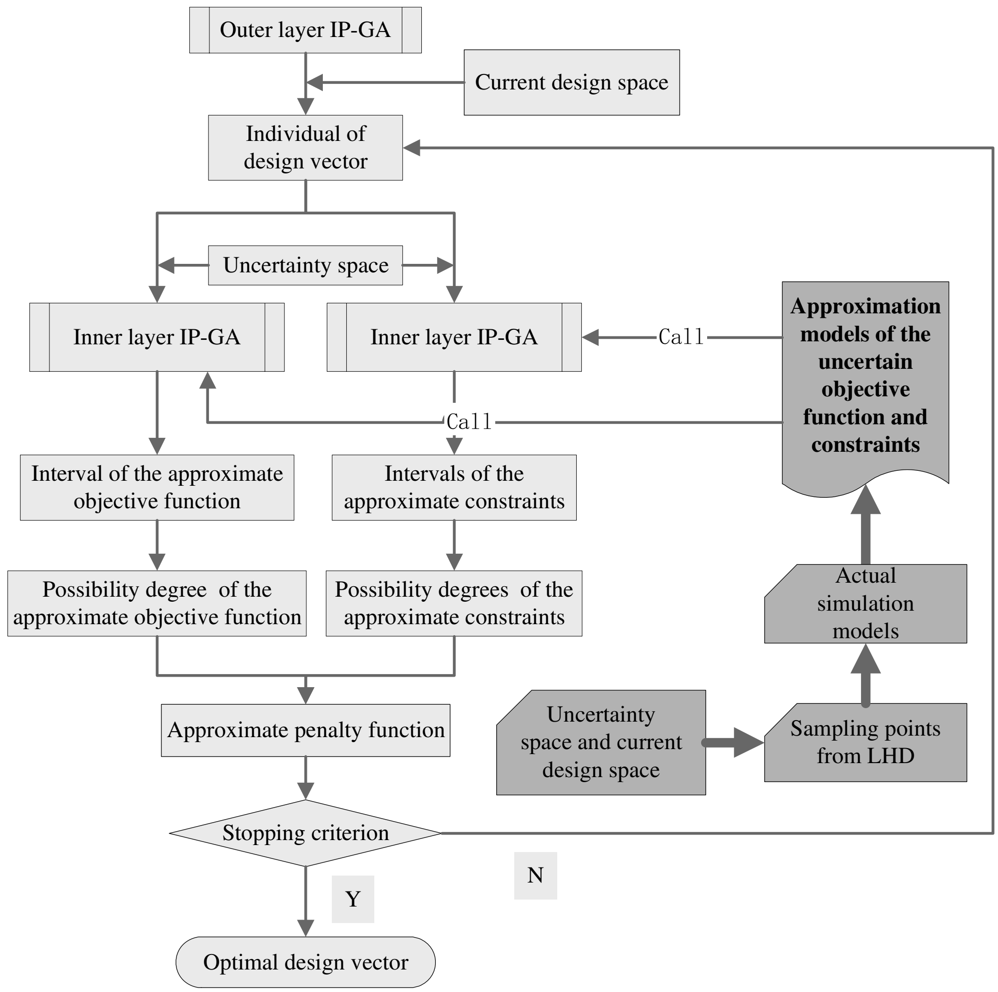

Figure 3: Nesting optimization based on the approximation models and IP-GA

and for each individual the inner layer IP-GA will be called two times to compute the lower and upper bounds of the approximate objective function or each constraint. In Fig. 3, the nesting optimization is still found to exist. However, here, the nesting optimization is based on the approximation models which are explicit and simple functions with respect to **X** and **U** instead of the computation-intensive actual simulation models, and therefore the optimization efficiency is still very high.

4.3 Approximation model management based on trust region method

The trust region method (Rodriguez and Renaud, 1998; Rodriguez et al., 2001) is used to manage the approximation models and make the optimization sequence converge at an optimal design vector. For each iterate defined by Eq. (13), a trust region test is needed to appraise the validity of the approximation models:

$$\rho^{(s)} = \frac{f_p\left(\mathbf{X}^{(s)}\right) - f_p\left(\mathbf{X}^{(s)*}\right)}{f_p\left(\mathbf{X}^{(s)}\right) - \tilde{f}_p\left(\mathbf{X}^{(s)*}\right)} \tag{20}$$

where $\mathbf{X}^{(s)*}$ is the optimum of Eq. (13) which is obtained through the method in section 4.2. $\rho^{(s)}$ is a reliability index which is a ratio of the actual change of the penalty function to the predictive change based on the approximation models. Thus $\rho^{(s)}$ can be used to monitor how well the current approximation models represent the actual simulation models.

According to the value of $\rho^{(s)}$, the move limit vector $\boldsymbol{\Delta}^{(s)}$, namely the trust region radius vector can

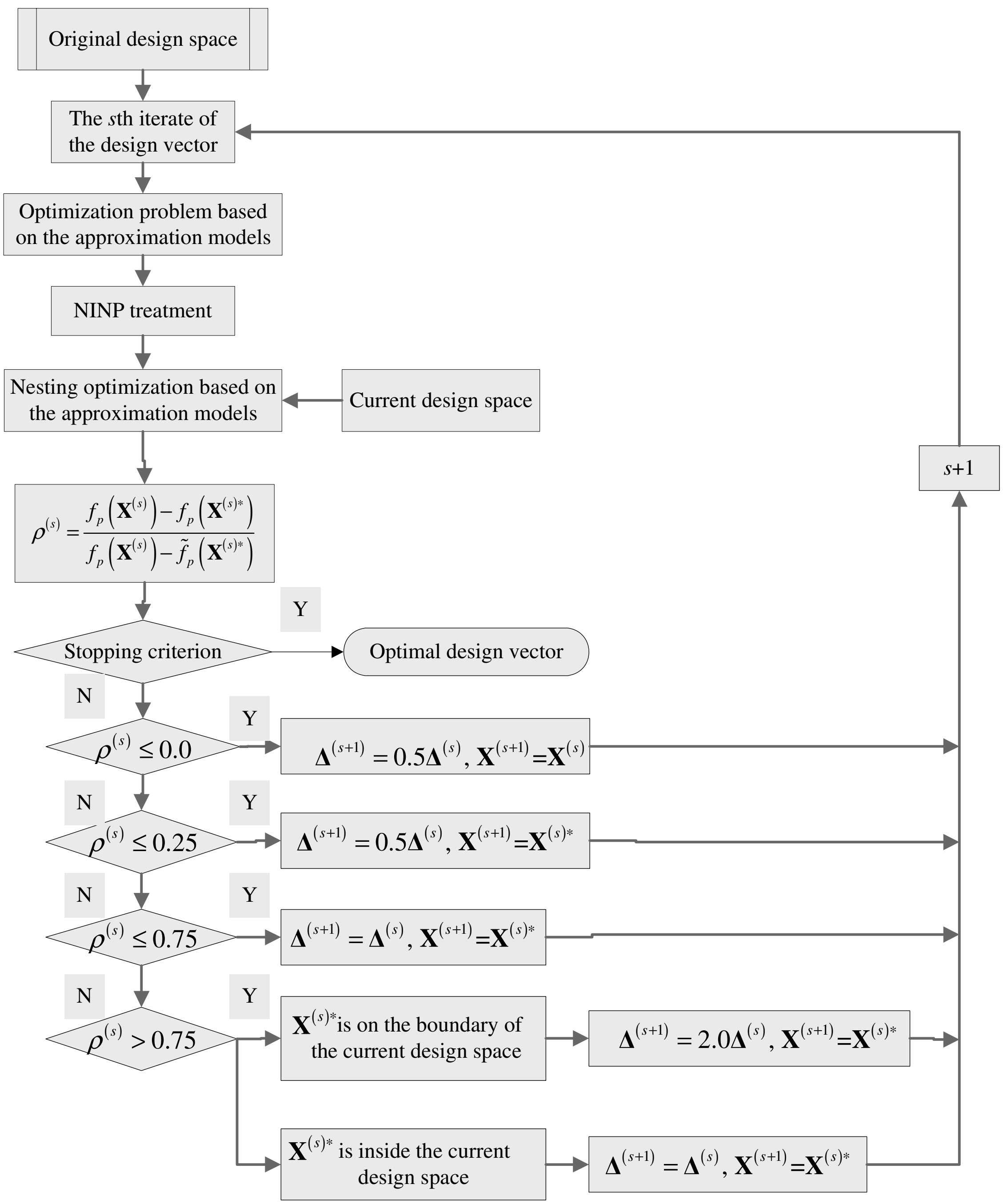

Figure 4: Uncertain optimization based on the NINP method and the trust region management

be updated: $\rho^{(s)} \leq 0.0$ implies that the approximation models have a very poor precision, and $\mathbf{\Delta}^{(s)}$ should be reduced to improve the approximation precision at the next iterate; $\rho^{(s)} \approx 1.0$ implies that the approximation models are precise, and the trust region radius vector $\mathbf{\Delta}^{(s)}$ should be expanded if $\mathbf{X}^{(s)*}$ is just located on the boundary of the current design space; For $0.0 < \rho^{(s)} < 1.0$, whether $\mathbf{\Delta}^{(s)}$ needs to be unchanged, reduced or expanded depends on how close $\rho^{(s)}$ is away from 0.0 or 1.0; $\rho^{(s)} \gg 1.0$ also means a poor approximation precision, however, a favorable search direction has been achieved. Generally, we can select many sets of constants to reduce or expand the trust region radius vector. In this research, the following updating procedure is used (Rodriguez and Renaud, 1998):

$$\begin{cases} \mathbf{\Delta}^{(s+1)} = 0.5\mathbf{\Delta}^{(s)}, & \\ \mathbf{X}^{(s+1)} = \mathbf{X}^{(s)} & \text{if } \rho^{(s)} \leq 0.0 \\ \mathbf{\Delta}^{(s+1)} = 0.5\mathbf{\Delta}^{(s)}, & \\ \mathbf{X}^{(s+1)} = \mathbf{X}^{(s)*} & \text{if } 0.0<\rho^{(s)} \leq 0.25 \\ \mathbf{\Delta}^{(s+1)} = \mathbf{\Delta}^{(s)}, & \\ \mathbf{X}^{(s+1)} = \mathbf{X}^{(s)*} & \text{if } 0.25 < \rho^{(s)} \leq 0.75 \\ \mathbf{\Delta}^{(s+1)} = 2.0\mathbf{\Delta}^{(s)}, & \\ \mathbf{X}^{(s+1)} = \mathbf{X}^{(s)*} & \text{if } \rho^{(s)} > 0.75, \text{ and } \mathbf{X}^{(s)*} \text{ is on the boundary of the current design space} \\ \mathbf{\Delta}^{(s+1)} = \mathbf{\Delta}^{(s)}, & \\ \mathbf{X}^{(s+1)} = \mathbf{X}^{(s)*} & \text{if } \rho^{(s)} > 0.75, \text{ and } \mathbf{X}^{(s)*} \text{ is inside the current design space} \end{cases} \tag{21}$$

The flowchart of the uncertain optimization based on the trust region management is shown in Fig. 4. The whole optimization process consists of a sequence of iterates. At each iterate, an uncertain sub-optimization problem based on the NINP method and the approximation models is performed, and a predicative optimal design vector can be obtained. Then according to the value of the reliability index, the design space at the next iterate can be determined. The iterative process will be terminated until a stopping criterion is satisfied. In this paper, the maximum iterate number is used as stopping criterion.

As shown in Eq. (20), when computing the reliability index $\rho^{(s)}$, the values of the actual penalty function and the approximate penalty function at $\mathbf{X}^{(s)}$ and $\mathbf{X}^{(s)*}$ need to been known. Being different from the deterministic optimization problems, here to compute the actual and approximate penalty functions will involve several optimization processes which are employed to compute the intervals of the objective function and constraints. Obviously, the approximate penalty function can be computed very efficiently by calling the approximation models based on Eq. (14). However, if we use the optimization processes based on the actual simulation models to compute the actual penalty function, the computation efficiency will be very low. Thus we provide an efficient approach to compute the actual penalty function $f_p\left(x^{(s)}\right)$ and $f_p\left(x^{(s)*}\right)$. Firstly, a set of sampling points of $\mathbf{U}$ are selected through LHD in the uncertainty space. Inputting each sampling point of $\mathbf{U}$ into the actual simulation models in which $\mathbf{X}$ is specified as $x^{(s)}$ or $x^{(s)*}$, several sets of samples can be obtained to construct the approximation models of the objective function and constraints using the RS. All of the approximation models are explicit functions of the uncertain vector $\mathbf{U}$. In the investigated optimization problem Eq. (1), the uncertainty level, namely the interval of each uncertain parameter is assumed to be small, thus these obtained approximation models which are defined within the small uncertainty space can be ensured to be precise. Generally, we can construct an infinitely precise approximation model using quadratic polynomial RS within an infinitely small uncertainty space. Thus as long as the uncertainty space is small enough, the enough precise approximation models can be obtained for the objective function and constraints. Fortunately, this condition can commonly be satisfied in practical applications as the uncertainty of the parameters always behaves a small disturbance around the nominal values. Then based on these approximation models, the intervals of the objective function and constraints $x^{(s)}$ or $x^{(s)*}$ can be obtained very precisely using also the IP-GA, and whereby the penalty function can be also obtained with a very fine precision. Thus in a rigorous sense, this obtained penalty function is still an approximate value. Because it can be guaranteed to be very precise, it is addressed as "actual penalty function" to distinguish from the approximate penalty function.

4.4 Robustness optimization based on the approximation management model

Through the above uncertain optimization based on the approximation management model, an optimal possibility degree P_m can be obtained for the uncertain objective function. If $P_m = 1.0$, a robustness criterion should be considered as addressed in section 3.3. Then a robustness optimization can be performed based on the approximation management model. The optimization process also consists of a sequence of sub-optimization problems based on the approximation models, and at the sth iterate a following sub-optimization problem can be formulated based on Eq. (10):

$$\min_{\mathbf{X}} \tilde{F}^w$$

subject to

$$\begin{aligned}
&P\left(\tilde{F}^I \geq G^I\right) = 1.0 \\
&P\left(\tilde{C}_i^I \leq V_i^I\right) \geq \lambda_i, \quad i = 1,2,\ldots,l \\
&\max\left[\mathbf{X}_l, \mathbf{X}^{(s)} - \mathbf{\Delta}^{(s)}\right] \leq \mathbf{X} \leq \min\left[\mathbf{X}_r, \mathbf{X}^{(s)} + \mathbf{\Delta}^{(s)}\right]
\end{aligned} \tag{22}$$

where

$$\begin{aligned}
\tilde{F}^w &= \frac{\tilde{f}^R(\mathbf{X}) - \tilde{f}^L(\mathbf{X})}{2} \\
&= \frac{\max\limits_{U\in\Gamma} \tilde{f}(\mathbf{X},\mathbf{U}) - \min\limits_{U\in\Gamma} \tilde{f}(\mathbf{X},\mathbf{U})}{2} \\
\tilde{F}^I &= \left[\tilde{f}^L(\mathbf{X}), \tilde{f}^R(\mathbf{X})\right] \\
&= \left[\min_{U\in\Gamma} \tilde{f}(\mathbf{X},\mathbf{U}), \max_{U\in\Gamma} \tilde{f}(\mathbf{X},\mathbf{U})\right] \\
\tilde{C}_i^I &= \left[\tilde{g}_i^L(\mathbf{X}), \tilde{g}_i^R(\mathbf{X})\right] \\
&= \left[\min_{U\in\Gamma} \tilde{g}_i(\mathbf{X},\mathbf{U}), \max_{U\in\Gamma} \tilde{g}_i(\mathbf{X},\mathbf{U})\right]
\end{aligned} \tag{23}$$

where $\tilde{F}^w$ is the radius of the approximate objective function. Then based on Eq. (12), Eq. (22) can be transformed into the following optimization problem in terms of an approximate penalty function $\tilde{f}'_p$:

$$\begin{aligned}
\min_{\mathbf{X}} \tilde{f}'_p(\mathbf{X}) = \tilde{F}^w &+ \sigma \sum_{i=1}^{l} \varphi\left(P\left(\tilde{C}_i^I \leq V_i^I\right) - \lambda_i\right) \\
&+ \sigma\left(P\left(\tilde{F}^I \geq G^I\right) - 1.0\right)^2 \\
\max\left[\mathbf{X}_l, \mathbf{X}^{(s)} - \mathbf{\Delta}^{(s)}\right] &\leq \mathbf{X} \leq \min\left[\mathbf{X}_r, \mathbf{X}^{(s)} + \mathbf{\Delta}^{(s)}\right]
\end{aligned} \tag{24}$$

The above sequential optimization problem can be solved by nearly the same way as the problem Eq. (13), namely, using the RS to construct the approximation models of the objective function and constraints and obtaining the predicative design based on the nesting optimization of IP-GA at each iterate, and then the trust region method is used to make the optimization sequence converge at an optimal design vector.

5 Numerical examples and discussion

5.1 Benchmark test

A benchmark test is presented as follows:

$$\max_{\mathbf{X}} f[\mathbf{X},\mathbf{U}] = U_1(X_1+2)^2 + U_2^3(X_2+1) + X_3^2$$

subject to

$$\begin{aligned}
&U_1^2(X_1+X_3) + U_2(X_2-4)^2 \leq V^I \\
&2 \leq X_1 \leq 14, \quad 2 \leq X_2 \leq 14, \quad 2 \leq X_3 \leq 14
\end{aligned} \tag{25}$$

The performance interval G^I is specified as $[45,68]$ for the uncertain objective function. The nominal values of the uncertain parameters U_1 and U_2 are both 1.0.

The population size and crossover probability are set to 5 and 0.5 for the IP-GA, respectively. The maximum generation number is set to 200 and 100 for the outer layer IP-GA and inner layer IP-GA, respectively. The maximum iterate number is specified as 10 for the optimization sequence. At each iterate, the uncertain objective function and constraint both need 30 samples to create their approximation models, and 8 samples to compute the actual penalty function. The possibility degree level λ of the constraint and the penalty factor σ are set to 0.8 and 1000, respectively. In the following text, the above problem will be analyzed based on three cases.

5.1.1 Case I

In this case, the uncertainty level is set to $\pm 10\%$ off from the nominal values for U_1 and U_2, namely $U_1 \in [0.9, 1.1]$ and $U_2 \in [0.9, 1.1]$. The interval V^I in the constraint is specified as $V^I = [8,9]$. The original design vector $\mathbf{X}^{(1)}$ and the original trust

Table 1: Optimization results of case I

<table>
<tr><th>Iterate s</th><th>$\mathbf{X}^{(s)}$</th><th>Penalty function $f_p\left(\mathbf{X}^{(s)}\right)$</th><th>Precise optimum $\mathbf{X}$ (conventional method)</th></tr>
<tr><td>1</td><td>(8.00, 8.00, 8.00)</td><td>-639.00</td><td rowspan="12">(5.45, 4.06, 2.05)</td></tr>
<tr><td>2</td><td>(3.49, 3.88, 2.36)</td><td>0.00</td></tr>
<tr><td>3</td><td>(3.87, 4.21, 3.29)</td><td>0.24</td></tr>
<tr><td>4</td><td>(4.79, 3.80, 2.39)</td><td>0.51</td></tr>
<tr><td>5</td><td>(4.79, 3.80, 2.39)</td><td>0.51</td></tr>
<tr><td>6</td><td>(4.89, 4.55, 2.24)</td><td>0.58</td></tr>
<tr><td>7</td><td>(4.94, 4.39, 2.20)</td><td>0.59</td></tr>
<tr><td>8</td><td>(5.44, 4.08, 2.05)</td><td>0.84</td></tr>
<tr><td>9</td><td>(5.44, 4.08, 2.05)</td><td>0.84</td></tr>
<tr><td rowspan="2">10</td><td>(5.44, 4.08, 2.07)</td><td>0.84</td></tr>
<tr><td colspan="2">Possibility degree of the uncertain objective function: 0.84
Possibility degree of the uncertain constraint: 0.80
Deviations from the precise optimum: (0.2%, 0.5%, 1.0%)</td></tr>
</table>

region radius vector are set to $(8.00, 8.00, 8.00)$ and $(6.00, 6.00, 6.00)$, respectively. The optimization results at all of the iterates are listed in Table 1. Additionally, we also use a conventional optimization method illustrated in Fig. 2 to obtain an optimum, in which the outer layer and inner layer optimization operators are both IP-GA. This optimum is regarded as a precise one to test the precision of the present method, and it is also listed in Table 1. It can be found that the penalty function and whereby the design vector become better and better as the iterate proceeds. At the 8^{th} iterate, the penalty function converges at a stationary value 0.84. After 10 iterates, an optimal design vector $\mathbf{X}^{(10)} = (5.44, 4.08, 2.07)$ is finally obtained which is very close to the precise optimum, and the maximum deviation from the precise optimum is only 1.0% which occurs at X_3. At this optimal design vector, the possibility degree of the uncertain objective function is 0.84, and the possibility degree of the uncertain constraint is 0.8 which is just equal to the possibility degree level. The convergence curve is shown in Fig. 5, and it can be found that the present method has a high convergence velocity.

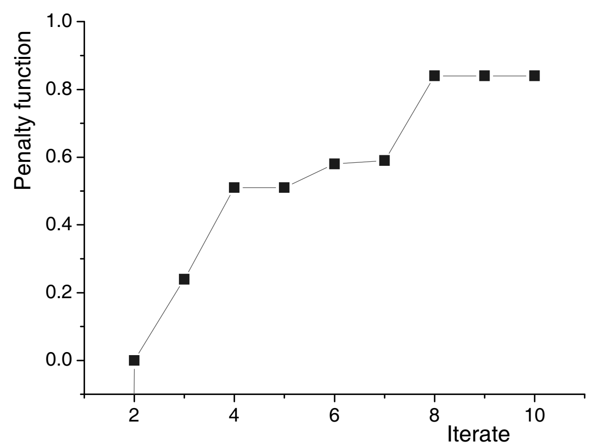

Figure 5: Convergence curve of case I

In this numerical example, the objective function in Eq. (25) is assumed as a computation-intensive simulation model, and the evaluation number of the objective function is our concern. At each iterate except the first one, 38 evaluations of the simulation model are needed among which 30 ones are used to construct the approximation optimization problem and 8 ones are used to calculate the actual penalty function $f_p\left(\mathbf{X}^{(s)*}\right)$, and the actual penalty function $f_p\left(\mathbf{X}^{(s)}\right)$ can be inherited from the preceding iterate. At the first iterate, 46 evaluations are needed, as 8 more evaluations are required to compute the actual penalty function $f_p\left(\mathbf{X}^{(1)}\right)$. Thus for 10 iterates, our method needs a total of 388 evaluations. On the other hand, using the conventional method based on the nesting optimization of IP-GA, we need a total of 1.0×10^6 evaluations. Obviously, comparing with the conventional method, the present method has a much higher optimization efficiency.

5.1.2 Case II

In this case, the uncertainty level is also set to $\pm 10\%$ off from the nominal values of the uncertain parameters, while the interval V^I in the constraint is specified as $V^I = [8, 11]$. The original design vector and trust region radius vector are set as same as case I. The optimization results at each iterate are listed in Table 2, and the convergence curve is shown in Fig. 6. It is found that at the 9^{th} iterate the penalty function converges at a maximum value 1.0. At the optimal design vector $(6.29, 4.04, 2.00)$, the possibility degrees of the uncertain objective function and constraint are 1.00 and 0.81, respectively. Thus as analyzed in section 4.4, a robustness optimization based on the approximation management model is then performed for the numerical example. The corresponding optimization results are listed in Table 3. An optimal design vector $(5.89, 3.92, 2.35)$ is obtained and the corresponding penalty function is 7.84. At this optimum, the radius of the uncertain objective function is 7.84. The possibility degrees of the uncertain objective function and constraint are 1.00 and 0.80, respectively, and they both satisfy the requirements in Eq. (10), namely the possibility degree of the objective function is equal to 1.0 and the one of the constraint is not less than the possibility degree level. The convergence curve is shown in Fig. 7, and it can be found that in the beginning phase a relatively fine design vector can be obtained very quickly and as the optimization proceeds the convergence velocity becomes relatively slow.

5.1.3 Case III

As mentioned before, the present method is based on a precondition that the uncertainty level is relatively small. Here the influences of the uncertainty levels on the optimization results will be investigated. In this case, the uncertainty level is set to $\pm 30\%$ off from the nominal values for U_1 and U_2, namely $U_1 \in [0.7, 1.3]$ and $U_2 \in [0.7, 1.3]$. The interval V^I is set to $V^I = [8, 9]$. The original design vector and trust region radius vector are set as same as case I. The optimization results

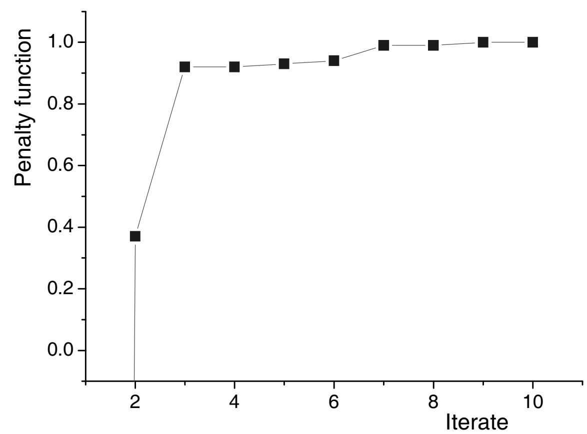

Figure 6: Convergence curve of case II (possibility degree optimization)

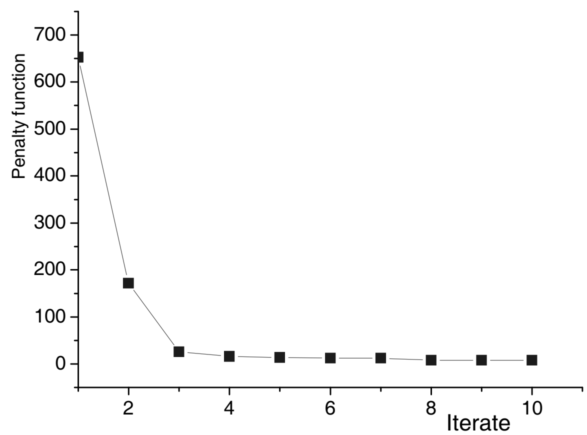

Figure 7: Convergence curve of case II (robustness optimization)

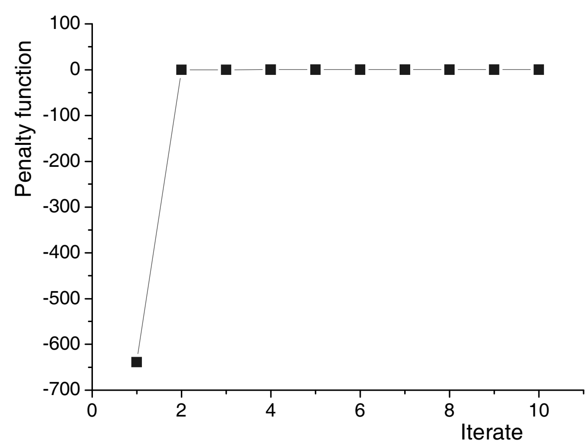

Figure 8: Convergence curve of case III

Table 2: Optimization results of case II (possibility degree optimization)

Iterate s	$\mathbf{X}^{(s)}$	Penalty function $f_p\left(\mathbf{X}^{(s)}\right)$
1	(8.00, 8.00, 8.00)	-639.00
2	(4.59, 3.92, 2.21)	0.37
3	(5.65, 3.99, 2.29)	0.92
4	(5.65, 3.99, 2.29)	0.92
5	(5.65, 4.27, 2.16)	0.93
6	(5.65, 4.46, 2.26)	0.94
7	(6.02, 4.41, 2.00)	0.99
8	(6.02, 4.41, 2.00)	0.99
9	(6.02, 4.10, 2.21)	1.00
10	(6.20, 4.04, 2.00)	1.00
	Possibility degree of the uncertain objective function: 1.00 Possibility degree of the uncertain constraint: 0.81	

Table 3: Optimization results of case II (robustness optimization)

Iterate s	$\mathbf{X}^{(s)}$	Penalty function $f'_p\left(\mathbf{X}^{(s)}\right)$
1	(8.00, 8.00, 8.00)	652.71
2	(4.96, 4.26, 2.20)	171.75
3	(5.50, 4.25, 2.00)	25.74
4	(5.56, 3.81, 2.28)	16.23
5	(5.57, 3.64, 2.40)	13.68
6	(5.58, 3.91, 2.38)	12.63
7	(5.58, 3.91, 2.38)	12.63
8	5.96, 3.54, 2.00)	8.07
9	5.96, 3.54, 2.00)	8.07
10	(5.89, 3.92, 2.35)	7.84
	Radius of the uncertain objective function: 7.84 Possibility degree of the uncertain objective function: 1.00 Possibility degree of the uncertain constraint: 0.80	

are listed in Table 4, and the convergence curve is shown in Fig. 8. It can be found that at only the 4^{th} iterate the penalty function converges at a stationary value 0.10, and the corresponding optimal design vector is $(3.57, 4.19, 2.14)$. Comparing with the precise optimum $(3.47, 4.26, 2.34)$ from the conventional method, the maximum deviation of this optimal design vector reaches 8.5% which occurs at X_3. As analyzed in case I, for uncertainty level $\pm 10\%$ the maximum deviation of the optimization results from the present method and the conventional optimization method is only 1.0%. Thus, with the increasing of the uncertainty level, the optimization precision of the present method declines. This phenomenon can be explained from two aspects. Firstly, for a larger uncertainty level, the intervals of the uncertain parameters are wider and hence the uncertainty space is larger. Thus the precision of the approximation models at each iterate will be worse, as the approximation models are created within the uncertainty space and current design space. Furthermore, the uncertainty space is not updated in the sequential optimization, and hence even for a very small design space a fine approximation precision is still difficult to achieve for a relatively large uncertainty space. Secondly, when computing the actual penalty function, some approximation models are also created within the uncertainty space. Thus for a larger uncertainty level,

Table 4: Optimization results of case III

<table>
<tr><th>Iterate s</th><th>$\mathbf{X}^{(s)}$</th><th>Penalty function $f_p\left(\mathbf{X}^{(s)}\right)$</th><th>Precise optimum $\mathbf{X}$ (conventional method)</th></tr>
<tr><td>1</td><td>(8.00, 8.00, 8.00)</td><td>-639.00</td><td rowspan="11">(3.47, 4.26, 2.34)</td></tr>
<tr><td>2</td><td>(2.00, 4.24, 2.00)</td><td>0.00</td></tr>
<tr><td>3</td><td>(2.74, 3.80, 2.55)</td><td>0.00</td></tr>
<tr><td>4</td><td>(3.57, 4.19, 2.14)</td><td>0.10</td></tr>
<tr><td>5</td><td>(3.57, 4.19, 2.14)</td><td>0.10</td></tr>
<tr><td>6</td><td>(3.57, 4.19, 2.14)</td><td>0.10</td></tr>
<tr><td>7</td><td>(3.57, 4.19, 2.14)</td><td>0.10</td></tr>
<tr><td>8</td><td>(3.57, 4.19, 2.14)</td><td>0.10</td></tr>
<tr><td>9</td><td>(3.57, 4.19, 2.14)</td><td>0.10</td></tr>
<tr><td rowspan="2">10</td><td>(3.57, 4.19, 2.14)</td><td>0.10</td></tr>
<tr><td colspan="2">Possibility degree of the uncertain objective function: 0.10
Possibility degree of the uncertain constraint: 0.83
Deviations from the precise optimum: (2.9%, 1.6%, 8.5%)</td></tr>
</table>

these approximation models and whereby the actual penalty function have a larger error. Then the precision of the reliability index will be also decreased and therefore the updating of the design space will be influenced. As a result, to ensure the optimization precision of the present method, the uncertainty level should be guaranteed to be relatively small.

5.2 *Application*

Thin-walled beams connected by spot welding are major structures of an automotive body for load-support and energy-absorption. Optimizing the thin-walled beam structures to improve their crashworthiness performance is very important to the security design of vehicles. The closed-hat beam is a kind of typical thin-walled beam structure in automotive body, and in this application the optimization of a closed-hat beam impacting a rigid wall with an initial velocity of 10 m/s is investigated. As shown in Fig. 9, the closed-hat beam is formed by a hat beam and a web plate which are connected through some uniformly distributed spot-welding points along the two rims of the hat beam. The time duration of the impacting process is 20ms. Based on the reference (Kurtaran et al., 2002), the closed-hat beam will be optimized to maximize the absorbed energy subjected to an axial impact force (average normal impact force on the rigid wall). The research (Wang, 2002) indicated that the plate thickness t, round radius R of the hat beam and space length d of each two neighboring spot-welding points have prominent effects on the crashworthiness performance of a closed-hat beam, and hence these three parameters are employed as design variables in our study. The finite element method (FEM) is used to simulate the impacting process, and an elasto-plasticity material model of bilinear kinematic hardening is used for the closed-hat beam, as given in Table 5. The nominal values of the yield stress σ_s and tangent Modulus E_t are 310Mpa and 763Mpa, respectively. Due to the measuring and manufacturing errors, σ_s and E_t are treated as uncertain parameters, and the uncertainty level is $\pm 5\%$ off from their nominal values, namely $\sigma_s \in [294.5\text{Mpa}, 325.5\text{Mpa}]$ and $E_t \in [724.85\text{Mpa}, 801.15\text{Mpa}]$. As a result, a following optimization problem can be formulated:

$$\max_{t,R,d} f_e(t,R,d,\sigma_s,E_t)$$

subject to

$$\begin{aligned}
&g_f(t,R,d,\sigma_s,E_t) \le [65\text{KN}, 70\text{KN}] \\
&\sigma_s \in [294.5\text{Mpa}, 325.5\text{Mpa}], \\
&E_t \in [724.85\text{Mpa}, 801.15\text{Mpa}] \\
&0.5\text{mm} \le t \le 2.5\text{mm}, \\
&1\text{mm} \le R \le 8\text{mm}, \\
&10\text{mm} \le d \le 60\text{mm}
\end{aligned} \tag{26}$$

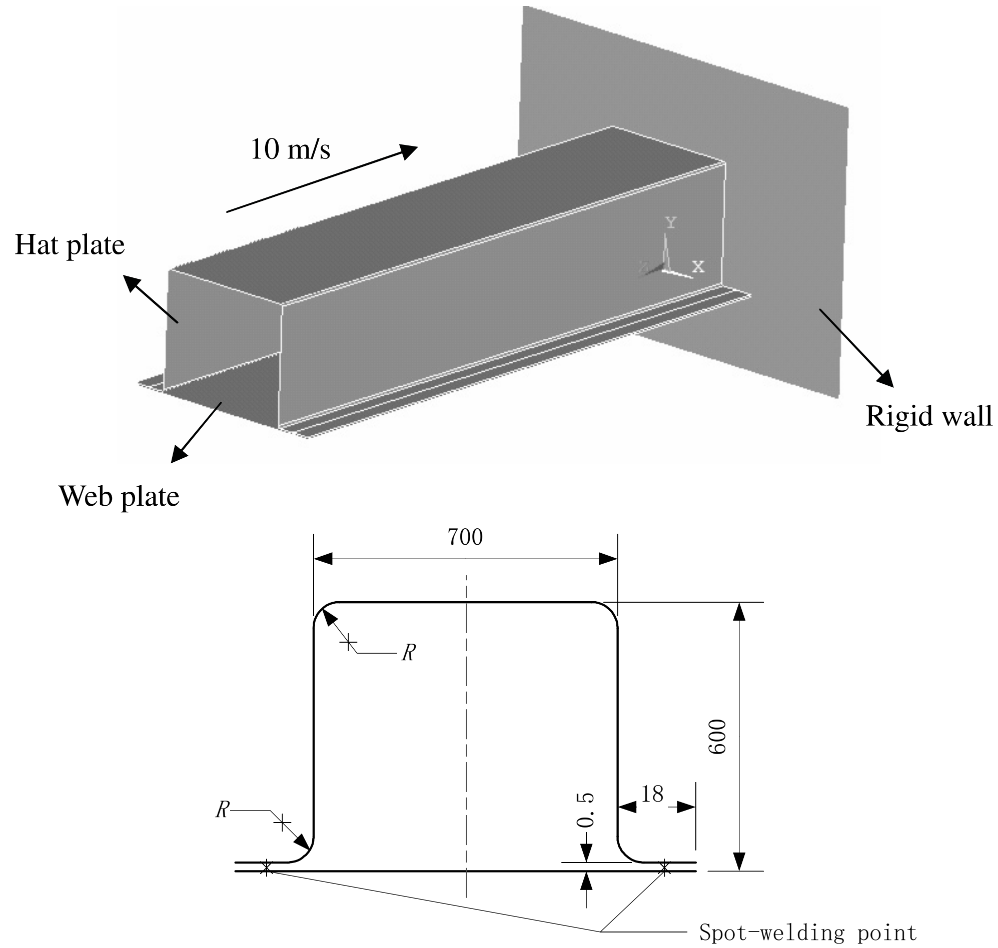

Figure 9: A closed-hat beam impacting the rigid wall and its cross-sectional dimensions (mm)

Table 5: Material properties of the closed-hat beam

Young's Modulus E	Poisson's ratio ν	Density ρ	Yield stress σ_s	Tangent Modulus E_t
2.0×10^5Mpa	0.27	7.85×10^{-3}Kg/mm^{-3}	310Mpa	763Mpa

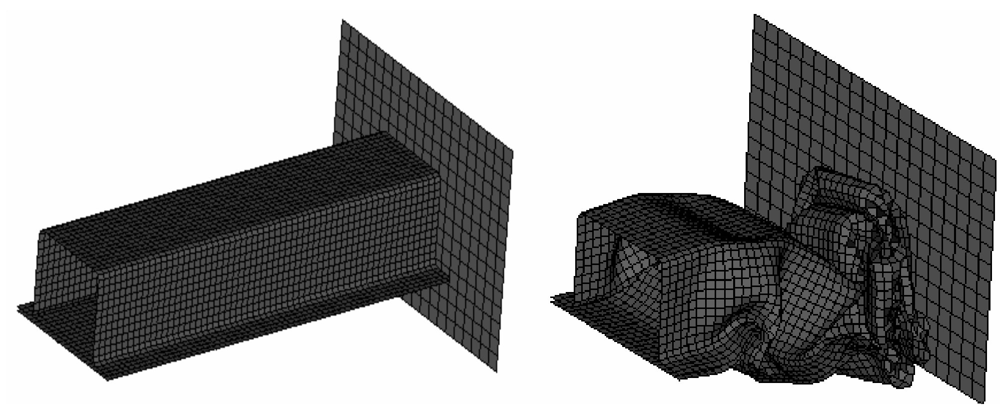

Figure 10: FEM mesh and a possible deformation of the closed-hat beam impacting a rigid wall

where the objective function f_e and constraint g_f represent the absorbed energy (i.e. internal energy) of the closed-hat beam and the axial impact force, respectively, and they are both obtained through the FEM.

The FEM simulation is carried out on the commercial software ANSYS/LS-DYNA. The Belytschko-Tsay shell element is used to create the FEM mesh of the impacting system, and the total number of the elements is 4200. A concentrated mass as weight as 250Kg is attached to the end of the closed-hat beam in order to supply enough crushing energy. The FEM mesh and a possible deformation of the impacting system are shown in Fig. 10.

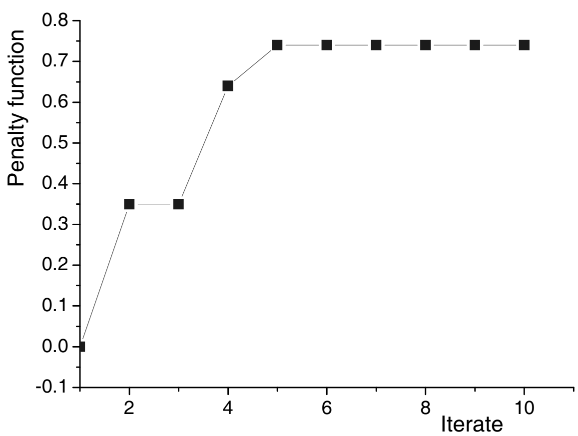

Figure 11: Convergence curve of the close-hat beam optimization

Firstly, the performance interval G_e^I of the objective function is set to $[8\text{KJ}, 10\text{KJ}]$, namely we will maximize the possibility degree that the interval of the absorbed energy of the closed-hat beam is larger than $[8\text{KJ}, 10\text{KJ}]$. The possibility degree level λ of the constraint g_f and the penalty factor σ are set to 0.8 and 1000, respectively. The original design vector is $(1.5\text{mm}, 4.5\text{mm}, 35.0\text{mm})$, and the original trust region radius vector is $(1.0\text{mm}, 3.5\text{mm}, 25.0\text{mm})$. The maximum iterate number is also set to 10. At each iterate, 30 samples are used to create the approximation models for the uncertain objective function and constraint, and 8 samples are used for calculation of the actual penalty function. The optimization results at all iterates are listed in Table 6, and the corresponding convergence curve is shown in Fig. 11. It can be found that the convergence velocity of the iterative process is very high, as at only the 5^{th} iterate a stationary design vector $(2.10\text{mm}, 2.45\text{mm}, 35.41\text{mm})$ is achieved with a penalty function value 0.74. At this design vector, the possibility degree of the uncertain constraint is 0.83 which is larger than the predetermined possibility degree level 0.8.

Secondly, the performance interval G_e^I is set to $[7\text{KJ}, 8\text{KJ}]$ which is a lower performance requirement for the objective function than the preceding one. A new uncertain optimization based on this G_e^I is performed to maximize the possibility degree of the uncertain objective function, and the optimization results are listed in Table 7. Here only three iterates are provided, as at the third iterate the possibility degree of the uncertain objective function reaches 1.00. Thus according to the aforementioned analysis, a robustness criterion is introduced, and an uncertain optimization is then carried out to minimize the radius of the uncertain objective function. In the optimization, the penalty factor σ is set to 10000, and the other concerned parameters are kept same. The optimization results are listed in Table 8. The design vector converges at a stationary point $(2.00\text{mm}, 4.50\text{mm}, 33.20\text{mm})$ at only the 4^{th} iterate, and the corresponding penalty function is 0.17. At this design vector, the possibility degrees of the uncertain objective function and constraint are both 1.00 which is just what we expect. The radius of the objective function is 0.17, namely under this optimal design vector, the uncertainty of the absorbed energy of the closed-hat beam caused by the uncertain plasticity parameters is only $0.17 \times 2 = 0.34$KJ. Thus in practical application, this closed-hat beam can be ensured to possess a robust crashworthiness performance.

6 Conclusion

In this paper, an efficient uncertain optimization method is developed by combining a new NINP method with an approximation management model. Intervals are used to model the parameter uncertainty, and hence the problems with

Table 6: Optimization results of the closed-hat beam ($G_e^I = [8\text{KJ}, 10\text{KJ}]$)

Iterate s	$\mathbf{X}^{(s)}$ (mm)	Penalty function $f_p\left(\mathbf{X}^{(s)}\right)$
1	(1.50, 4.50, 35.00)	0.00
2	(2.06, 3.66, 35.02)	0.35
3	(2.06, 3.66, 35.02)	0.35
4	(2.09, 3.11, 35.40)	0.64
5	(2.10, 2.45, 35.41)	0.74
6	(2.10, 2.45, 35.41)	0.74
7	(2.10, 2.45, 35.41)	0.74
8	(2.10, 2.45, 35.41)	0.74
9	(2.10, 2.45, 35.41)	0.74
10	(2.10, 2.45, 35.41)	0.74
	Possibility degree of the objective function: 0.74 Possibility degree of the constraint: 0.83	

Table 7: Optimization results of the closed-hat beam ($G_e^I = [7\text{KJ}, 8\text{KJ}]$, possibility degree optimization)

Iterate s	$\mathbf{X}^{(s)}$ (mm)	Penalty function $f_p\left(\mathbf{X}^{(s)}\right)$
1	(1.50, 4.50, 35.00)	0.39
2	(1.50, 4.50, 35.00)	0.39
3	(1.70, 3.60, 28.75)	1.00
	Possibility degree of the objective function: 1.00 Possibility degree of the constraint: 1.00	

Table 8: Optimization results of the closed-hat beam ($G_e^I = [7\text{KJ}, 8\text{KJ}]$, robustness optimization)

Iterate s	$\mathbf{X}^{(s)}$ (mm)	Penalty function $f'_p\left(\mathbf{X}^{(s)}\right)$
1	(1.50, 4.50, 35.00)	10000.26
2	(2.02, 6.59,20.80)	0.20
3	(2.02, 6.59, 20.80)	0.20
4	(2.00, 4.50, 33.20)	0.17
5	(2.00, 4.50, 33.20)	0.17
6	(2.00, 4.50, 33.20)	0.17
7	(2.00, 4.50, 33.20)	0.17
8	(2.00, 4.50, 33.20)	0.17
9	(2.00, 4.50, 33.20)	0.17
10	(2.00, 4.50, 33.20)	0.17
	Radius of the objective function: 0.17 Possibility degree of the objective function: 1.00 Possibility degree of the constraint: 1.00	

a small amount of information on the uncertainty can be treated effectively. In the suggested NINP method, the uncertain objective function is transformed into a deterministic single-objective optimization problem, instead of a multi-objective optimization as usually done in the other NINP methods. Additionally, the possibility degree of interval is used as a mathematical tool to deal with not only the uncertain objective function but also the uncertain constraints. Thus a uniform treatment form has been created for the uncertain objective function and constraints. Therefore, the uncertain optimization can be treated and performed more easily and conveniently. An approximation management model based on approximation models and trust region method is introduced into the NINP, and the nesting optimization of the actual simulation models is changed into the iterative nesting optimization of the explicit approximation models. Thus the optimization efficiency can be improved greatly. The optimization results of the benchmark test indicate that the present method has a fine convergence performance and high optimization efficiency. Additionally, through investigating different uncertainty levels, the present method is found to have a fine precision for a small uncertainty level. However, for a larger uncertainty level, the precision of the present method will decline. Thus to obtain an effective design, the uncertainty level of the parameters should be ensured to be relatively small. The present method is also applied to the crashworthiness design of a closed-hat beam with uncertain plasticity parameters. The fine optimization results exhibit the applicability of the present method to practical engineering problems.

Acknowledgement: This work is supported by the national 973 program under the Grant No. 2004CB719402, the National Natural Science Foundation of China (10572048) and the National Science Fund for Distinguished Young Scholars (50625519).

References

Abbas, M.; Bellahcene, F. (2006): Cutting plane method for multiple objective stochastic integer linear programming, *Eur. J. Oper. Res.*, 168(3):967-984.

Amirante, R.; Catalano, L.A.; Dadone, A.; Daloiso, V.S.E. (2007): Design optimization of the intake of a small-scale turbojet engine, *CMES: Computer Modeling in Engineering & Sciences*, 18(1):17-30.

Ben-Haim, Y.; Elishakoff, I. (1990): *Convex Models of Uncertainties in Applied Mechanics*, Elsevier Science Publisher, Amsterdam.

Charnes, A.; Cooper, W.W. (1959): Chance-constrained programming, *Manage. Sci.*, 6:73-79.

Chen, B.L. (2002): *Optimization theories and algorithms*, Tsinghua University Press, Beijing, China.

Fedelinski, P.; Gorski, R. (2006): Analysis and optimization of dynamically loaded reinforced plates by the coupled boundary and finite element method, *CMES: Computer Modeling in Engineering & Sciences*, 15(1):31-40.

Gyeong-Mi, C. (2005): Log-barrier method for two-stage quadratic stochastic programming, *Appl. Math. Comput.*, 164(1):45-69.

Haftka, R.T.; Gurdal, Z. (1992): *Elements of structural optimization*, Kluwer Academic Publishers, Netherlands.

Ishibuchi, H.; Tanaka, H. (1990): Multiobjective programming in optimization of the interval objective function, *Eur. J. Oper. Res.*, 48:219-225.

Jiang, C.; Han, X.; Liu, G.R. (2007a): A nonlinear interval number programming method for uncertain optimization problems, *Eur. J. Oper. Res.*, doi:10.1016/j.ejor.2007.03.031.

Jiang, C.; Han, X.; Liu, G.R.; Li, G.Y. (2007b): The optimization of the variable binder force in U-shaped forming with uncertain friction coefficient, *J. Mater. Process. Tech.*, 182:262-267.

Jiang, C.; Han, X.; Guan, F.J.; Li, Y.H. (2007c): An uncertain structural optimization method based on nonlinear interval number programming and interval analysis method, *Eng. Struct.*, Doi:10.1016/j.engstruct.2007.01.020.

Jiang, C.; Han, X.; Liu, G.R. (2007d): Optimization of structures with uncertain constraints based on convex model and satisfaction degree of interval, *Comput. Method. Appl. M.*, Doi:10.1016/j.cma.2007.03.024.

Jiang, C.; Han, X.; Liu, G.P. (2007e): A nonlinear interval number programming method for uncertain structures based on an approximation management framework, submitted.

Kall, P. (1982): Stochastic programming, *Eur. J. Oper. Res.*, 10,125-130.

Kurtaran, H.; Eskandarian, A.; Marzougui, D.; Bedewi, N.E. (2002): Crashworthiness design optimization using successive response surface approximations, *Comput. Mech.*, 29:409-421.

Lamberti, L.; Pappalettere, C. (2007): Weight optimization of skeletal structures with multipoint simulated annealing, *CMES: Computer Modeling in Engineering & Sciences*, 18(3):183-221.

Liu, B.D.; Zhao, R.Q.; Wang, G. (2003): *Uncertain programming with applications*, Tsinghua University Press, Beijing, China.

Liu, B.; Iwamura, K. (1997): Modelling stochastic decision systems using dependent-chance programming, *Eur. J. Oper. Res.*, 101(1):193-203.

Liu, G.R.; Han, X. (2003): *Computational inverse techniques in nondestructive evaluation*, CRC Press, Florida.

Liu, X.W.; Da, Q.L. (1999): A satisfactory solution for interval number linear programming, *Journal of Systems Engineering*, 14:123-128, China.

Ma, L.H. (2002): Research on method and application of robust optimization for uncertain system, Ph.D. thesis, Zhejiang University, China.

Mathur, R.; Advani, S.G.; Fink, B.K. (2003): A real-coded hybrid genetic algorithm to determine optimal resin injection locations in the resin transfer molding process, *CMES: Computer Modeling in Engineering & Sciences*, 4(5):587-601.

McKay, M.D.; Bechman, R.J.; Conover, W.J. (1979): A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, *Technometrics*, 21(2):439-452.

Moore, R.E. (1979): *Methods and applications of interval analysis*, Prentice-Hall, Inc., London.

Rodriguez, J.F.; Perez, V.M.; Padmanabhan, D.; Renaud, J.E. (2001): Sequential approximate optimization using variable fidelity response surface approximations, *Struct. Multidisc. Optim.*, 22:24-34.

Rodriguez, J.F.; Renaud, J.E. (1998): Convergence of trust region augmented Lagrangian methods using variable fidelity approximation data, *Structural Optimization*, 15:141-156.

Rodriguez, J.F.; Renaud, J.E.; Wujek, B.A.; Tappeta, R.V. (2000): Trust region model management in multidisciplinary design optimization, *J. Comput. Appl. Math.*, 124:139-154.

Rommelfanger, H. (1989): Linear programming with fuzzy objective. *Fuzzy set. Syst.*, 29:31-48.

Sedaghati, R.; Suleman, A.; Dost, S.; Tabarrok, B. (2001): Optimum design of adaptive truss structures using the integrated force method, *CMES: Computer Modeling in Engineering & Sciences*, 2(2):259-271.

Sengupta, A.; Pal, T.K.; Chakraborty, D. (2001): Interpretation of inequality constraints involving interval coefficients and a solution to interval linear programming, *Fuzzy set. Syst.*, 119:129-138.

Stephen, L.; Bhaskar, A.; Keane, A. (2003): Optimal orthogonal-array-based latin hypercubes, *J. Appl. Stat.*, 30(5):585-598.

Tanaka, H.; Ukuda, T.; Asal, K. (1984): On fuzzy mathematical programming, *Journal of Cybernetics*, 3:37-46.

Tapp, C.; Hansel, W.; Mittelstedt, C.; Becker, W. (2004): Weight-minimization of sandwich structures by a heuristic topology optimization algorithm, *CMES: Computer Modeling in Engineering & Sciences*, 5(6):563-573.

Tong, S.C. (1994): Interval number and fuzzy number linear programming, *Fuzzy set. Syst.*, 66:301-306.

Wang, H.L. (2002): Study on optimal design of auto-body structure based on crashworthiness numerical simulation, Ph.D. thesis, Shanghai Jiao

Tong University, China.

Zhang, Q.; Fan, Z.P.; Pan, D.H. (1999): A ranking approach for interval numbers in uncertain multiple attribute decision making problems, *Systems Engineering – Theory & Practice*, 5:129-133, China.