Struct Multidisc Optim (2011) 44:259–275
DOI 10.1007/s00158-011-0631-1

RESEARCH PAPER

A modified Benders decomposition method for efficient robust optimization under interval uncertainty

Sauleh Siddiqui · Shapour Azarm · Steven Gabriel

Received: 27 December 2010 / Revised: 28 January 2011 / Accepted: 29 January 2011 / Published online: 3 March 2011

Abstract The goal of robust optimization problems is to find an optimal solution that is minimally sensitive to uncertain factors. Uncertain factors can include inputs to the problem such as parameters, decision variables, or both. Given any combination of possible uncertain factors, a solution is said to be robust if it is feasible and the variation in its objective function value is acceptable within a given user-specified range. Previous approaches for general nonlinear robust optimization problems under interval uncertainty involve nested optimization and are not computationally tractable. The overall objective in this paper is to develop an efficient robust optimization method that is scalable and does not contain nested optimization. The proposed method is applied to a variety of numerical and engineering examples to test its applicability. Current results show that the approach is able to numerically obtain a locally optimal robust solution to problems with quasi-convex constraints ($\leq$ type) and an approximate locally optimal robust solution to general nonlinear optimization problems.

Keywords Robust optimization · Nonlinear optimization · Benders decomposition · Interval uncertainty · Quasi-convex function

S. Siddiqui (✉)
Department of Applied Mathematics, University of Maryland,
College Park, MD 20742, USA
e-mail: siddiqui@umd.edu

S. Azarm
Department of Mechanical Engineering, University of Maryland,
College Park, MD 20742, USA

S. Gabriel
Department of Civil Engineering, University of Maryland,
College Park, MD 20742, USA

1 Introduction

Engineering optimization problems often involve uncontrollable variations or uncertainties in both decision variables and parameters. Optimal solutions that might be deterministically feasible often end up being infeasible due to uncertainty. Additionally, even small levels of variations can cause large degradations in the objective function value. Manufacturing errors, measurement problems, and uncertainty in inputs are examples of sources for these variations.

In this paper, an approach for robust optimization, e.g., Ben-Tal et al. (2009), for linear, quadratic, convex, and non-convex programs is developed by applying a worst-case analysis using a decomposition method. No probability distribution is presumed, but only intervals with a nominal point (user or problem-defined) are used to represent the uncertainty in decision variables and/or parameters. The problem structure in this paper reflects a real-world design situation, e.g., when information about uncertain factors during the early stages of a design process is often limited.

This paper's approach (hereafter referred to as the *modified Benders method*) has been tested and verified with 14 optimization problem examples. A comprehensive review of the literature was conducted and the main distinctions between the proposed modified Benders method and previous works are as follows:

1. In addition to considering nonlinear constraints, the robust optimization problems in the proposed modified Benders method also involve a nonlinear objective function.[1] This is more general than only considering a

[1] In some cases, a slightly stricter condition with convexity in the lower-level of the Benders decomposition method is needed. However, we did not encounter this in any of our test problems. A workaround to this problem is available in Gabriel et al. (2009).

linear objective function in the problem as reported in the literature (e.g., Balling et al. 1986; Ben-Tal and Nemirovski 2002; Bertsimas and Sim 2006; Soyster 1973) or quadratic (e.g., Li et al. 2010) as well as other versions involving convex programs (e.g., Ganzerli and Pantelides 1999) or linearization to solve the problem (e.g., Balling et al. 1986). The modified Benders method is able to obtain exact locally optimal robust solutions to problems with quasi-convex constraints as well as non-convex quadratic programs, which no one method in the reported literature is able to achieve. Other approaches also consider distributions for uncertainty (e.g., Lee et al. 2009; Lagaros and Papadrakakis 2007) while the approach of this paper looks at a worst-case analysis for interval uncertainty without any explicit probability distribution or a nested optimization structure. Moreover, the modified Benders method is able to handle large uncertainties which earlier methods (e.g., Balling et al. 1986; Soyster 1973) were not able to tackle.

2. The proposed approach preserves the computational tractability, theoretically and practically, of the deterministic (i.e., nominal) problems. By contrast, under interval uncertainty, the computational effort for previous methods (e.g., Gunawan and Azarm 2004; Li et al. 2006) to obtain robust solutions is much higher than their deterministic counterparts. However, results from a variety of numerical experiments show that the computational effort of solving the robust optimization problems is not much greater than that of their deterministic counterparts for the modified Benders method. Moreover, the modified Benders method is scalable, in that by numerical tests, the number of function calls per iteration increases at most linearly with an increase in the number of variables, variables and parameters with uncertainty, and constraints.
3. Since this paper's approach is based on gradient-based methods, a globally optimal robust solution can never be guaranteed for non-convex problems. However, this paper defines the idea of a locally optimal robust solution, and shows that this approach can obtain a locally optimal robust solution for nonlinear robust optimization problems.
4. In addition to the uncertainty in the data of the problems (i.e., the parameters), interval uncertainty in the decision variables corresponding to manufacturing tolerances, implementation errors, etc. where optimized values cannot be achieved exactly, which is very common in practical engineering applications, is considered. For the current robust optimization formulations in the literature (e.g., Ben-Tal and Nemirovski 2008; Lu et al. 2010; Qiu and Wang 2010; Zhu and Ting 2001), considering uncertainty in the decision variables may considerably change those formulations or increase the complexity of the problem. The approach in this paper, however, keeps the same formulation and obtains locally optimal robust solutions to these problems with not much greater computational effort than the deterministic problem.
5. There has been an abundance of literature modifying Benders decomposition (Benders 1962) that enables the algorithm to solve various types of optimization problems. However, to the authors' knowledge, there have not been any modifications to Benders method that solve nonlinear robust optimization problems with interval uncertainty. Velarde and Laguna (2004) provided a Benders-based heuristic to solve the international source allocation problem. In this problem, a subset of international suppliers needs to be selected to meet local demand. The uncertainty is in the demand function parameters and exchange rates. However, their approach did not consider uncertainty in variables. For their approach to work, they needed to include control variables, which change depending on the uncertainty scenario to provide an easier route to solution. The approach in our paper does not require the introduction of such variables. Also, their methodology cannot be extended to general nonlinear robust optimization problems. Saito and Murota (2007) described a method to apply Benders decomposition to solve linear, mixed-integer, robust optimization problems with ellipsoidal uncertainty. However, this approach only works for linear problems. Finally, Montemanni (2006) applied a Benders algorithm to a specific robust spanning tree problem, while Ng et al. (2010) applied it to a specific semiconductor allocation problem that had uncertainty. Again, both approaches are not applicable to general robust optimization problems and have not modified Benders decomposition in the way this paper does.
6. There are related topics to robust optimization such as anti optimization (e.g., Qiu and Wang 2010) and reliability-based design optimization (e.g., Zou and Mahadevan 2006) that run into the same problems as described above of not being computationally tractable or only working for a certain simple type of problems. For example, Youn and Xi (2009) modified a double loop problem (like robust optimization) into a single loop so that it becomes computationally easier. This work involved using an eigenvector dimension reduction method, and probability distributions, which may not be applicable to general non-linear robust optimization problems. Also, neither of these papers has techniques that include interval uncertainty in parameters, in decision variables, along with being computationally efficient. The modified Benders method of this paper is not only directly relevant to robust

optimization, but it handles the specific two-level structure of robust optimization in a less computationally intensive way.

7. The modified Benders method provides a computationally tractable way of solving combined optimization and anti-optimization problems as well. Note that anti-optimization is a generalization of interval analysis and involves finding a worst-case design given non-stochastic uncertainty (Elishakoff and Ohsaki 2010). The lower-level problem of a robust optimization problem can then be described as an anti-optimization problem with the only slight difference being that robust optimization considers uncertainty in both decision variables as well as parameters, while anti-optimization is focused on parameter uncertainty. Indeed, the two-level combination of optimization and anti-optimization as first outlined by Elishakoff et al. (1994) and later described (Elishakoff and Ohsaki 2010) is equivalent to solving a robust optimization problem with uncertainty in parameters.

In the next two sections, the background terminology and problem definition are described. Then, in Sections 4 and 5, a detailed formulation of the proposed modified Bender's method and its tractability is given. Section 6 provides numerical and engineering examples from the literature that highlight the differentiating characteristics of this method. Some concluding remarks will be provided in Section 7.

Table 1 Definition of terms

Symbol	Interpretation
x	Vector of decision variables
p	Vector of parameters
f	Objective function to be minimized
$g_j(\tilde{x}, \tilde{p})$	Constraint functions of the form "≤ 0"
$\bar{g}_j(x, p, \hat{x}, \hat{p})$	Constraint functions of the form "≤ 0" modified
h_j	Constraint functions of the form "≤ 0" after modification to binary values
$\Delta x, \Delta p$	Maximum deviations of variables and parameters from nominal values
$\hat{x}, \hat{p}$	Deviations from nominal values of uncertain variables and parameters, respectively: $\hat{x} \in [-\Delta x, +\Delta x]$, $\hat{p} \in [-\Delta p, +\Delta p]$
$\tilde{x}, \tilde{p}$	Uncertain values of variables and parameters, respectively: $\tilde{x} \in [x - \Delta x, x + \Delta x]$, $\tilde{p} \in [p - \Delta p, p + \Delta p]$
Δf_0	User-specified tolerance for acceptable variation in objective function under uncertainty
v_c	Complicating variable to explain standard Benders decomposition
v_u	Uncomplicating variable to explain standard Benders decomposition
$c(v_c)$	Constraint of complicating variable
$d(v_c, v_u)$	Constraint of uncomplicating and complicating variables

2 Background terminology

Table 1 describes the terminology used in this paper.

First, an introduction to Benders Decomposition (Benders 1962) will be provided. Benders decomposition is used to efficiently solve nonlinear programs (Conejo et al. 2006) that can be decomposed into both complicating and uncomplicating decision variables. Normally, integer variables are defined as *complicating* variables as fixing their values allows the problem to have a structure that provides an easy solution (assuming the rest of the problem is relatively easier to solve). However, in general, the complicating variables need not be integer and can be real numbers which, if fixed, render a simple or decomposable problem. Using notation provided above, Benders decomposition seeks to solve an optimization problem of the following form:

$$
\begin{aligned}
&\min_{v_c, v_u} f(v_c, v_u) \\
&s.t. \\
&c(v_c) \leq 0 \\
&d(v_c, v_u) \leq 0
\end{aligned}
$$

where

$$
\begin{aligned}
&v_c \in U \subseteq \Re^n \\
&v_u \in \Re^m \\
&f(v_c, v_u) : \Re^{n\times m} \to \Re \\
&c(v_c) : \Re^n \to \Re^p \\
&d(v_c, v_u) : \Re^{n\times m} \to \Re^q
\end{aligned} \tag{1}
$$

To solve this problem, the Benders decomposition technique fixes values of v_c (which are part of a set U that can be integers or other sets that are subsets of $\Re^n$) and solves the simplified problem after decomposing into a master problem and sub-problem (Conejo et al. 2006). First, define an auxiliary function $\alpha(v_c)$ as follows which expresses the objective function of the original problem as a function solely of the complicating variables.

$$
\begin{aligned}
&\alpha(v_c) = \min_{v_u} f(v_c, v_u) \\
&s.t. \\
&d(v_c, v_u) \leq 0
\end{aligned} \tag{2}
$$

Using the definition of $\alpha(v_c)$, the original problem (1) can be expressed as follows.

$$\begin{aligned}
&\min_{v_c} \alpha(v_c) \\
&s.t. \\
&c(v_c) \leq 0
\end{aligned} \tag{3}$$

Iteratively, a subproblem (4) is solved to approximate $\alpha(v_c)$ from above by fixing values of the complicating variables $\left(v_c = v_c^{fixed}\right)$ and obtaining the dual variables λ to these constraints as shown in (4).

$$\begin{aligned}
&\min_{v_u} f(v_c, v_u) \\
&s.t. \\
&v_c = v_c^{fixed} \quad (dual : \lambda) \\
&d(v_c, v_u) \leq 0
\end{aligned} \tag{4}$$

Then, the solution to the above problem, $\left(v_u^{sol}\right)$ and the dual variables (λ^{sol}) are used to construct "Benders cuts" in the master problem to approximate the function[2] $\alpha(v_c)$ from below.

$$\begin{aligned}
&\min_{\alpha, v_c} \alpha \\
&s.t. \\
&c(v_c) \leq 0 \\
&\alpha \geq f\left(v_c^{fixed}, v_u^{sol}\right) + \lambda^{sol}\left(v_c - v_c^{fixed}\right)
\end{aligned} \tag{5}$$

Since the master problem (5) has a larger feasible region, it provides a lower bound (z_{lo}) while the more restricted subproblem (4) provides an upper bound (z_{up}) for the solution objective function value. These problems are solved iteratively until $(z_{up} - z_{lo})/(z_{lo})$ is less than some tolerance. As long as the function $\alpha(v_c)$ is convex, Benders decomposition provides an optimal solution. The next section defines the specific robust optimization problem that is considered in this paper.

3 Definitions and goal

The goal in robust optimization is to optimize the objective function with respect to uncertain decision variables x, satisfying all constraints and ensuring the objective variation is kept within an acceptable range Δf_0, while accounting for uncertainty in decision variables and parameters p. A *parameter* has a fixed deterministic value, which is not changeable in the problem, but has an uncertainty range. Specifically, this paper considers robust optimization problems of the form:

$$\begin{aligned}
&\min_{x} f(x, p) \\
&s.t. \\
&g_j(\tilde{x}, \tilde{p}) \leq 0 \quad j = 1, ..., J \\
&\left| \frac{f(\tilde{x}, \tilde{p}) - f(x, p)}{\Delta f_0} \right| \leq 1 \\
&\forall \tilde{x} \in [x - \Delta x, x + \Delta x] \\
&\forall \tilde{p} \in [p - \Delta p, p + \Delta p]
\end{aligned} \tag{6}$$

where f and g are continuous and first-order differentiable in both x and p. In the next few paragraphs, we define a few terms used in the paper.

Objective robustness For a candidate point x^c objective robustness holds if the inequality

$$\left| \frac{f(\tilde{x}, \tilde{p}) - f(x^c, p)}{\Delta f_0} \right| \leq 1 \tag{7}$$

is satisfied for all $\tilde{x} \in [x^c - \Delta x, x^c + \Delta x]$ and for all $\tilde{p} \in [p - \Delta p, p + \Delta p]$. Thus, this inequality ensures that the maximum objective function variation stays below a certain predetermined maximal amount Δf_0 when presented with deviations in uncertain variables and parameters.

Feasibility robustness For a candidate solution x^c if

$$g_j(\tilde{x}, \tilde{p}) \leq 0 \quad j = 1, ..., J \tag{8}$$

is satisfied for all $\tilde{x} \in \left[x^c - \Delta x, x^c + \Delta x\right]$ and for all $\tilde{p} \in \left[p - \Delta p, p + \Delta p\right]$, then feasibility robustness holds. Note that (7) is just another constraint, so it can be easily incorporated into inequality (8) when stating formulation (6). From this point on, inequality (7) will not be stated separately in any formulation but will be assumed to be incorporated in inequality (8). For a more detailed description on objective robustness, please refer to Li et al. (2006).

Robust point A robust point of a robust optimization problem is one that is feasible for all realizations of uncertainty and satisfies objective robustness as outlined in (7). In other words, a robust point is both objectively and feasibly robust.

Locally optimal robust For a robust optimization problem, a *locally optimal robust* solution x^*, is a robust point that is optimal $(f(x^*) \leq f(x), \forall x \in U')$ within a neighboring set U' of robust points. It is essential that the neighboring set be made up of only robust points otherwise the term is ill-defined.

[2] A sufficient condition for convergence is that the objective function in formulation (4) needs to be convex. However, the modification this paper presents, from the experimental results, does not need this convexity as the Benders cuts are modified. For more information, please refer to Conejo et al. (2006) and Gabriel et al. (2009).

Next, a description of the notation used and the goal of the proposed method are provided. Write the uncertainty elements as the sum of a nominal value x plus the uncertain deviation $\hat{x}$, similarly for p. Hence,

$$\tilde{x} = x + \hat{x}, \quad \tilde{p} = p + \hat{p}. \tag{9}$$

Observe that since $g(\tilde{x}, \tilde{p})$ is continuous in both x and p, (9) can be substituted into the constraint functions (8). Hence, $g_j(\tilde{x}, \tilde{p}) = g_j(x + \hat{x}, p + \hat{p}) = \bar{g}_j(x, \hat{x}, p, \hat{p})$ for all $j = 1, \ldots, J$. Note here that $\bar{g}$ now depends on four variables. The 1st and the 3rd variables represent the nominal values and the 2nd and the 4th the variations under uncertainty. This change in notation is due to the fact that the variables with ˆ on top will be used as complicating variables in the proposed modified Benders method. Clearly, $\bar{g}_j(x, \hat{x}, p, \hat{p})$ are continuous and differentiable in x, p, $\hat{x}$ and $\hat{p}$.

The goal is to choose values of x such that the formulation (6) gives an optimal solution regardless of the values of $\hat{x}$ and $\hat{p}$. Since this paper only considers the worst-case analysis, the method aims to get the "worst" values of $\hat{x}$ and $\hat{p}$ for this problem. These are called "interval-optimal" values, as defined next.

Interval-optimal An *interval-optimal* value of $\hat{x}$ for a particular candidate solution x^c is defined as $\hat{x}^* \in [-\Delta x, +\Delta x]$ such that $g(x^c, \hat{x}, p) \leq g(x^c, \hat{x}^*, p)$ for all realizations of $\hat{x} \in [-\Delta x, +\Delta x]$. The $\hat{x}^* \in [-\Delta x, +\Delta x]$ is a particular value of the $\hat{x}$ such that the constraint attains its maximum value at that particular value of uncertainty. A similar definition is for the interval-optimal value of the uncertainty in a parameter p. However, finding these interval-optimal values can be computationally intensive, as that would entail solving the robust optimization problem (6). Note that since $g(\tilde{x}, \tilde{p})$ is a continuous function of $\hat{x} \in [-\Delta x, +\Delta x]$ (and $\hat{p} \in [-\Delta p, +\Delta p]$) over a closed and bounded interval set $[-\Delta x, +\Delta x]$ ($[-\Delta p, +\Delta p]$), by the Weierstrass theorem (Bazaraa et al. 1993) over compact sets, there always exists at least one interval-optimal point in every uncertainty interval for each candidate solution x^c.

4 Formulation of approach

The strategy will be to partition uncertainty intervals for variables and parameters and attempt to find the interval-optimal points. Essentially, one can think of this as attempting to check the robustness of a candidate solution by partitioning the uncertainty interval by points and checking feasibility at each point. Clearly, if all constraints are feasible when fixed with the interval-optimal values for uncertainty elements, then the candidate solution is robust. Partitions can be selected depending on which type of constraint functions have uncertainty. In particular, quasi-convex constraint functions are simplest to consider. A function $g(x, \hat{x}, p)$ is said to be quasi-convex in $\hat{x} \in [-\Delta x, +\Delta x]$ if for all $\hat{x} \in [-\Delta x, +\Delta x]$, $g(x, \hat{x}, p) \leq \max\{g(x, \Delta x, p), g(x, -\Delta x, p)\}$ (Bazaraa et al. 1993). Hence, under uncertainty in x, the maximum value of the function g lies on one of the endpoints of uncertainty when g is quasi-convex. So finding the interval-optimal point for quasi-convex constraint functions under uncertainty entails checking one of the two endpoints of uncertainty. A similar argument applies for uncertainty in parameters.

If all constraint functions, including the constraint relating to objective robustness, are quasi-convex, the interval-optimal values will lie on one of the endpoints of the uncertainty interval. Note that all convex functions are quasi-convex as well, so this result extends to convex constraints. Now, for quasi-convex constraints, the problem has a different form from (6) as described in below:

$$\begin{aligned}
&\min_x f(x, p)\\
&s.t.\\
&g(x, p, \hat{x}, \hat{p}) \leq 0\\
&where\\
&\forall \hat{x} \in \{-\Delta x, \Delta x\}\\
&\forall \hat{p} \in \{-\Delta p, \Delta p\}
\end{aligned} \tag{10}$$

Essentially, the proposed method modifies Benders decomposition to obtain the interval-optimal values and uses them to solve the robust optimization problem. This is most efficiently done by considering binary values for uncertainty values at two endpoints (left endpoint signified by 0, right endpoint signified by 1).

However, for nonlinear, not necessarily quasi-convex constraint functions, the interval-optimal values need not lie on the endpoints. For this, the constraint functions in the uncertainty interval range need to be checked at intermediary points to find the interval-optimal points. Figure 1 shows how checking further points helps. For quasi-convex constraints, Fig. 1a, only the endpoints need to be checked. Since quadratic constraints are symmetric, for concave quadratic constraints in particular (that are not quasi-convex) checking three points (endpoints plus central point of uncertainty interval) is enough to guarantee a robust solution (in general true for all symmetric concave constraints), Fig. 1b. For general nonlinear constraints, however, problems might be encountered if enough points within the uncertainty interval are not checked. This is shown in Fig. 1c where the method fails if enough points are not selected. Selecting more points, Fig. 1d, solves this

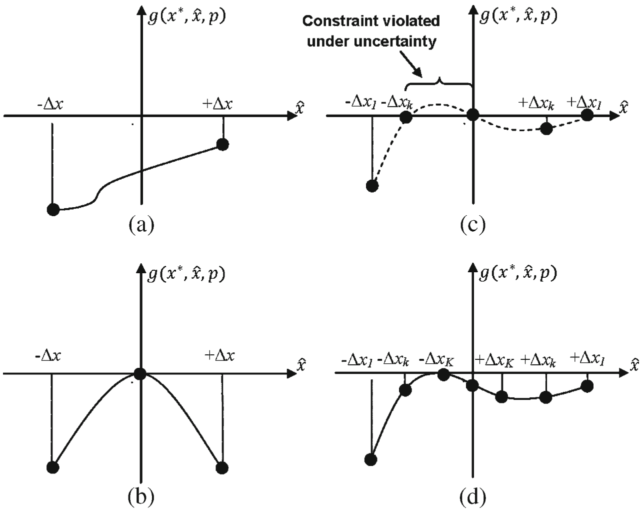

Fig. 1 Checking feasibility by interval-optimal points for constraints: **a** quasi-convex: successful check by endpoints; **b** symmetric concave: successful check by middle point and endpoints; **c** non-convex: failed check due to insufficient number of points; **d** non-convex: successful check by sufficient number of intermediary points

problem. Points maybe selected according to the accuracy desired for a locally optimal robust solution.

Checking additional points entails adding additional constraints that have different uncertainty variables $\hat{x}_k \in [-\Delta x_k, \Delta x_k]$ and $\hat{p}_k \in [-\Delta p_k, \Delta p_k]$, $k = 1, \ldots, K$, with uncertainty ranges that are subsets of the original uncertainty ranges, i.e., $[-\Delta x_k, \Delta x_k] \subset [-\Delta x, \Delta x]$ and $[-\Delta p_k, \Delta p_k] \subset [-\Delta p, \Delta p]$ for all $k = 1, \ldots, K$. In particular, the modified Benders method considers a uniform distribution of these points to be checked with $\Delta x_k = \Delta x/k$, $\Delta p_k = \Delta p/k$, $k = 1, \ldots, K$. To check for center points, the constraints with $\hat{x} = 0$, $\hat{p} = 0$ need to be added.

So, the formulation changes from (10) by adding further sample points ($2K$ sample points as every pair of checked points needs just one uncertainty range):

$$
\begin{aligned}
&\min_x f(x, p) \\
&s.t. \\
&g(x, p, \hat{x}, \hat{p}) \le 0 \\
&g(x, p, 0, 0) \le 0 \\
&g(x, p, \hat{x}_k, \hat{p}_k) \le 0 \quad k = 1, ..., K \\
&where \\
&\forall \hat{x} \in \{-\Delta x, \Delta x\} \\
&\forall \hat{x}_k \in \{-\Delta x_k, \Delta x_k\} \quad k = 1, ..., K \\
&\forall \hat{p} \in \{-\Delta p, \Delta p\} \\
&\forall \hat{p}_k \in \{-\Delta p_k, \Delta p_k\} \quad k = 1, ..., K
\end{aligned} \tag{11}
$$

Here, Δx_k, Δp_k, $k = 1, \ldots, K$ are the different sample points for the uncertainty interval as described by the black dots in Fig. 1. Note that $0 \le \Delta x_k \le \Delta x$ and $0 \le \Delta p_k \le \Delta p$. The uncertainty elements (with ˆ) only take on two values (per element of vector) each. Without loss of generality, by simple linear transformations, the endpoints of the uncertainty intervals can be mapped to take on binary values of 0 or 1. An example of this linear transformation is given of a simple example later (see Section 6.1). These binary variables need to be introduced for each parameter or variable or constraint with uncertainty. For simplicity, as stated before, the objective robustness constraint (7) is included within the feasibility constraints (8) represented by g. However, for ease of notation, since all the uncertainty has been included in binary variables, the constraint functions will now be signified by h as they have binary arguments. They are essentially the same function as g, but h signifies the change from continuous uncertainty elements to binary uncertainty which also incorporates the multiple checking. The reformulated problem then becomes for N variables and parameters with uncertainty (accounting for partition of uncertainty sets):

$$
\begin{aligned}
&\min_x f(x, p) \\
&s.t. \\
&h_j(x, p, b) \le 0 \quad j = 1, ..., J \\
&where \\
&\forall b_{ijk} \in \{0, 1\} \quad i = 1, ..., N, j = 1, ..., J, k = 1, ..., K
\end{aligned} \tag{12}
$$

Note that with this formulation, if the interval-optimal points as defined above are obtained, they can be fixed to help obtain robust solutions. The interval-optimal values input into constraints create a new restricted feasible region in which all points are robust. Hence, solving the optimization problems with the interval-optimal values in the constraints will help obtain the robust solution. In other words, if for a candidate solution x^* the interval-optimal values $\hat{x}^*$, $\hat{p}^*$, $\hat{x}_k^*$, $\hat{p}_k^*$ are obtained, they can be input into the constraint g to get $g(x^*, p, \hat{x}_k^*, \hat{p}_k^*) \le 0$, $k = 1, \ldots, K$. By the definition of the interval-optimal value, $g(\tilde{x}^*, \tilde{p}^*) \le 0$ for all $\tilde{x} \in [x - \Delta x, x + \Delta x]$, $\tilde{p} \in [p - \Delta p, p + \Delta p]$ where $\tilde{x} = x^* + \hat{x}^*, \tilde{x} = p^* + \hat{p}^*$.

To speed up computation of the proposed approach, gradient-based optimization algorithms (Bazaraa et al. 1993) are used as opposed to population-based optimization ones (such as Genetic Algorithms or Simulated Annealing). In particular, the nonlinear solvers CONOPT in GAMS (GAMS 2009) were used for all test problems except the last one on heat exchanger design. Since the code for the heat exchanger design problem was already available and coded in MATLAB, *fmincon* (MATLAB 2008) was used for that particular problem.

5 Modified Benders method: algorithm and tractability

The following are the steps for the algorithm to solve for locally optimal robust solutions based on the standard Benders decomposition algorithm (1)–(5) when applied to problem (12) but with modified robust Benders cuts.

Set iteration counter (*it*) to 0. Pick a small positive constant for tolerance (*tol*).

Step 1: Set iteration counter (*it*) to $it = it + 1$. The variable b is a *complicating* variable since it takes on binary values. Hence, the original master problem will be:[3]

$$\begin{aligned}&\min_{\alpha,b} \alpha\\&s.t.\\&0 \le b \le 1\\&\alpha \ge \alpha_{\min}\\&b \in \Re^{NJK}\end{aligned} \tag{13}$$

The lower bound on α is user-defined depending on the problem. Solving the above problem gives $\alpha = \alpha_{it}$ and $b = b_{fixed,it}$ (This is the solution b) where

$$\begin{aligned}&\alpha = \min_x f(x, p)\\&s.t.\\&h_j(x, p, b) \le 0 \quad j = 1, \ldots, J\\&where\\&0 \le b_{ijk} \le 1 \quad i = 1, \ldots, N,\ j = 1, \ldots, J,\ k = 1, \ldots, K\end{aligned} \tag{14}$$

Step 2: Fix the value of b, and then solve the following subproblem as in the standard Benders decomposition method.

$$\begin{aligned}&w = \min_x f(x, p)\\&s.t.\\&h_j(x, p, b) \le 0 \quad j = 1, \ldots, J\\&where\\&b_{ijk} = b_{fixed,it} \quad (Dual : \lambda_{ijk})\\&\qquad i = 1, \ldots, N,\ j = 1, \ldots, J,\ k = 1, \ldots, K\end{aligned} \tag{15}$$

Step 3: Check for convergence. Set $z_{up} = w$ and $z_{low} = \alpha_{it}$. If the difference $(z_{up} - z_{low})/z_{low} \le tol$ then stop. Note that in the first iteration, $\alpha_{\min}$ is usually set low enough to not get convergence. This is important because up to that point, the algorithm will not have deviated from the standard Benders decomposition approach. If the problem does not converge, proceed to Step 4. Strictly speaking, a locally optimal robust solution is obtained only if the difference between the upper and lower bound is zero.

Step 4: Add modified robust Benders cuts to take into account the robustness aspects of the problem. In Benders decomposition, the Benders cut that would have been added would be:

$$w_{it} + \lambda_{ijk}(b_{ijk} - b_{it,fixed}) \le \alpha \tag{16}$$

However, this value of λ, a dual variable to the constraint $b_{ijk} = b_{it,fixed}$, gives the direction which will be favorable for the optimization.[4] However, the method is modified to go towards the interval-optimal direction obtaining a robust solution that will be feasible for all realizations of uncertainty. Hence, this dual variable is used to get a "robust Benders cut" as follows:

$$w_{it} - \lambda_{ijk}(b_{ijk} - b_{it,fixed}) \le \alpha \tag{17}$$

Add this constraint to the original master problem and return to Step 1.

Step 1 (returned): Solve the following master problem after adding the robust Benders cut:

$$\begin{aligned}&\min_{b} \alpha\\&s.t.\\&0 \le b \le 1\\&\alpha \ge 0\\&w_{it} - \lambda_{ijk}(b_{ijk} - b_{it,fixed}) \le \alpha\end{aligned} \tag{18}$$

Step 2: is next with a new value of $b_{it,fixed}$. The algorithm proceeds in this manner until convergence is met.

Step 4 helps get a value of $b = b^*$ such that $h(x^*, b, p) \le h(x^*, b^*, p)$ for all possible values of b. In other words, this method obtains an approximation of the interval-optimal value of b which is a representation of the interval-optimal values of $\hat{x}$ and $\hat{p}$. This way, the method obtains locally

[3] Note that the complicating variable b does not appear in the objective function for the master problem (13).

[4] Again, as discussed before, convexity of the sub-problem is a sufficient condition to guarantee convergence. The dual variable λ can have slightly different interpretation for a general benders decomposition method. However, numerical evidence in this paper shows that with the modified Benders cuts this condition can be relaxed.

Table 2 Analysis of function calls for one iteration

Operation	Number of assignments	Function calls
Objective function	1	1
Iteration counter	1	1
Constraints	$J+1$ (*Extra 1 for obj. robust*)	$N(J+1)$
Fixing uncertainty at lower level	N	N
Slope of modified Benders cut	N	$N(J+1)$
Sample points for nonlinear	$2K$	$2KN+4KN-2N$
Total maximum expected	–	$2+N+JN+6KN$

optimal robust solutions as the modified Benders cuts only select robust values. Benders decomposition provides optimal values when the auxiliary function α is convex (Conejo et al. 2006). It is beyond the scope of this paper to prove theoretically that a convex α will always enable us to obtain locally robust solutions. However, the numerical results indicate that this assertion is true. There are ways to detect and overcome the problem of a non-convex function α (Gabriel et al. 2009). However, none of the test examples had this difficulty.

Consider a single-objective robust optimization problem with V variables, J constraints, P parameters, and N variables and parameters with uncertainty. One function call is defined as any instance where the solver calls an objective function, constraint, or other value or assignment in the optimization problem. Table 2 gives the maximum number of function calls possible through one iteration of the modified Benders method. This analysis is only based on numerical evidence that the method finds a locally optimal robust solution.

K is in most cases not more than 10. Hence, the number of binary variables introduced to the formulation is NJK. Note that this is the maximum theoretically possible function calls and actual function calls (as shown by examples) are much less. For example, in the heat exchanger problem (see Section 6.5), $N=8$, $J=17$, $K=10$ and 6 iterations were used to solve the problem so the maximum expected number of function calls is $6(2+8+(17)(8)+6(10)(8)) =$ 3,756. According to the results, it took only 984 function calls to solve this particular problem.

6 Numerical results

The following numerical and engineering examples serve to demonstrate applicability of the algorithm, compare the proposed algorithm to a previous algorithm from the literature, and show the different types of problems that can be solved. The first example is a simple numerical quadratic program to show the algorithm steps in detail. The next three examples then increase the complexity (number of variables, nonlinearity) of this quadratic program and show how the number of function calls changes. The next two examples are similar linear programs except that Example 6 has been shown to need significantly higher number of function calls to solve for the locally robust optimal solution than Example 5 (Li et al. 2010). Example 7 and 8 are examples of robust optimization problems with quasi-convex constraints which the modified Benders method is shown to solve exactly. The next four problems are scalable versions of an engineering example with quasi-convex constraints. This example shows that the modified Benders method is scalable and can be applied to large problems without a drastic increase in function calls. The final two examples are engineering design examples that are nonlinear (non-convex) programs. Of the two engineering examples, the first one (Welded Beam Design) considers objective robustness and the second (Heat Exchanger Design) considers feasibility robustness. All optimization problems correspond to minimizing a single objective function with a set of constraints. Problems labeled as "self" have been designed by the authors to use as test problems; detailed formulations as well as further characteristics of the solution are in the Appendix. Solutions were checked by a simple uniform discretization of the uncertainty range (each point separated by 0.01) to see if any of the constraints were violated under uncertainty. Tolerance (*tol*) was set to 0.00001 for all examples.

6.1 Numerical example (Example 1) to show methodology

A simple numerical example is presented to show how Benders algorithm is modified to obtain robust solutions. In the following robust problem, uncertainty is only in the constraint (without loss of generality).

$$\begin{aligned}
&\min_x \left((x_1-0.6)^2+(x_2-0.6)^2\right)\\
&s.t.\\
&(q+\hat{q})+x_1+x_2\le 0\\
&-x_1\le 0\\
&-x_2\le 0\\
&where\\
&\forall \hat{q}\in\{-\Delta q,\Delta q\}
\end{aligned} \tag{19}$$

Take $q=-1$, $\Delta q=0.1$. The robust solution to this problem (verified as unique algebraically) is $x_1=0.45$, $x_2=0.45$ with the interval-optimal value $\hat{q}=0.1$. Reformulate this problem to have 0, 1 binary variables as described in the formulation in Section 4.

$$\min_x \left((x_1-0.6)^2+(x_2-0.6)^2\right)$$
$$s.t.$$
$$(-1+(2b-1)(0.1))+x_1+x_2\le 0$$
$$-x_1\le 0$$
$$-x_2\le 0$$
$$where$$
$$\forall b\in\{0,1\} \tag{20}$$

Proceed according to the modified Benders method described in Section 2.

Step 1: The master problem is the following:

$$\min_{\alpha,b}\alpha$$
$$0\le b\le 1$$
$$\alpha\ge 0 \tag{21}$$

The lower bound on α is obviously zero given the form of (21). Solving the above problem gives $\alpha=0$ and $b=0$. Here

$$\alpha=\min_x\left((x_1-0.6)^2+(x_2-0.6)^2\right)$$
$$s.t.$$
$$(-1+(2b-1)(0.1))+x_1+x_2\le 0$$
$$-x_1\le 0$$
$$-x_2\le 0$$
$$0\le b\le 1 \tag{22}$$

Step 2: Fix b and then solve the following subproblem

$$w=\min_x\left((x_1-0.6)^2+(x_2-0.6)^2\right)$$
$$s.t.$$
$$(-1+(2b-1)(0.1))+x_1+x_2\le 0$$
$$-x_1\le 0$$
$$-x_2\le 0$$
$$b=0\qquad(\lambda) \tag{23}$$

This gives: $x_1=0.55$, $x_2=0.55$ with the dual variable value $\lambda=0.02$.

Step 3: Check for convergence with $z_{up}-z_{low}=0.005-0=0.005$ where $z_{up}=((0.55-0.6)^2+(0.55-0.6)^2)$ and $z_{low}=\alpha$. Since this is not good enough for convergence, a modified Benders cut is added.

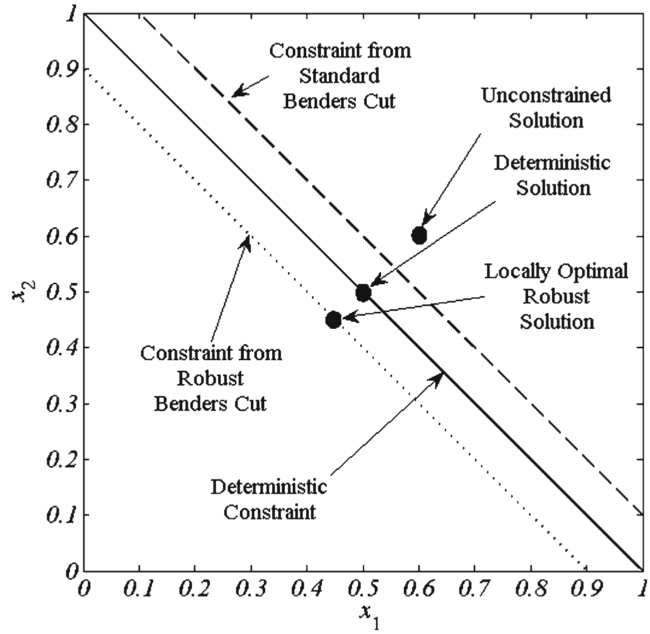

Fig. 2 Adding a modified (Robust) Benders cut

Step 4: ***Here is where the method differs from Benders decomposition*** Add the following *robust Benders cut*.

$$0.005-\lambda(b-b_{fixed})\le\alpha \tag{24}$$

The following graph (Fig. 2) shows what happens when this cut is added. The standard Benders decomposition method would have taken a cut that would have forced $b=0$ and that would have given the constraint with the dashed line. However, the robust Benders cut (24) generates the cut signified by the dotted line, which is in fact the constraint that forms the border of the robust feasible region.

Step 1 (returned): Solve the following master problem after adding the robust Benders cut:

$$\min_b\alpha$$
$$s.t.$$
$$0\le b\le 1$$
$$\alpha\ge 0$$
$$0.005-\lambda(b-b_{fixed})\le\alpha \tag{25}$$

Table 3 Solution steps for modified Benders approach

Iteration	b	x_1	x_2	z_{low}	z_{up}	λ
1	0	0.55	0.55	0	0.005	0.02
2	1	0.45	0.45	0	0.045	0.02
3	1	0.45	0.45	0.045	0.045	0.06

Table 4 Detailed solution for simple problem

Information	Nominal solution	Robust solution
x_1	0.5	0.45
x_2	0.5	0.45
$f(x)$	0.02	0.045
Function calls	5	11

Step 2 is next with the new value of b. The algorithm proceeds in this manner until convergence. Table 3 summarizes these results.

This simple problem was solved in three iterations. The final row corresponds to the locally optimal robust solution, which can be verified algebraically. The details are shown in Table 4.

6.2 Results with numerical test problems

Table 5 describes the results obtained from the numerical test problems. The first two examples have only uncertainty in the parameters while the rest have uncertainty in the parameters and the decision variables.

The same test problems were solved using the method of Li et al. (2006) for robust optimization and the results are displayed in Table 5. Not only does the method of Li et al. (2006) use a lot of function calls, but often the robust function value is higher than the modified Benders value. Once the number of function calls exceeded 10^9, the run was stopped for the method of Li et al. (2006). These solutions have not been reported in Table 5 as well as in the rest of the paper.

6.3 Fleury's weight minimization like problem (Engineering example)

This is a modified example from the literature (Groenwold and Etman 2010) so that interval uncertainty is present in all the decision variables. This example supports the approach for feasibility robustness as well as corroborates the fact that the modified Benders method is able to tackle problems with large number of variables and constraints without being computationally expensive. The problem is for a given number of variables N as follows:

$$\min_x f(x) = \sum_{i=1}^{N} x_i$$

$$s.t.$$

$$\sum_{i=1}^{0.95N} \frac{1}{\tilde{x}_i} + \frac{1}{N^2} \sum_{i=0.95N+1}^{N} \frac{1}{\tilde{x}_i} - N \leq 0$$

$$\sum_{i=1}^{0.95N} \frac{1}{\tilde{x}_i} - \frac{1}{N^2} \sum_{i=0.95N+1}^{N} \frac{1}{\tilde{x}_i} - 0.9N \leq 0$$

$$\frac{1}{N^2} \leq \tilde{x}_i \leq N^2 \quad i = 1, 2, \ldots, N$$

$$\tilde{x}_i \in [x_i - 0.1, x_i + 0.1] \tag{26}$$

The modified Benders method solved this problem with the results shown in Table 6 for $N = 10^2, 10^3, 10^4$, and 10^5. Note that the number of function calls increase linearly with the complexity of the problem. All variables have uncertainty. Again, this example was compared to Li et al. (2006) as shown in Table 6. However, the results for Li et al. (2006) are not reported as the problem was stopped after a certain number of function calls. Here, the method of Li et al. (2006) could, conceptually, solve this problem but would

Table 5 Description of test problems

Source	# of variables	# of constraints	Determ. optimal function value	Robust optimal function value	Li et al. (2006) function value	# of function calls (determ.)	# of function calls (robust)	Li et al. (2006) function calls
Example 1 (Self)	2	3	0.02	0.045	0.045	5	11	540
Example 2 (Self)	4	6	9.02	9.145	9.268	7	19	2,592
Example 3 (Self)	4	6	9.02	9.145	9.268	7	21	2,808
Example 4 (Self)	4	6	9.77	9.885	9.920	7	21	2,916
Example 5 (Self)	5	12	−23.00	−21.50	−20.75	5	17	7,856
Example 6 (Self)	5	12	−31.21	−29.79	−28.36	5	17	11,099
Hock 100 (Hock 1980)	7	18	680.6	692.4	–	7	19	$>10^9$
Hock 106 (Hock 1980)	8	22	7049	7219	–	5	17	$>10^9$

Table 6 Results for Fleury's weight minimization like problem

Number of variables	$N = 10^2$	$N = 10^3$	$N = 10^4$	$N = 10^5$
Tolerance	10^{-6}	10^{-8}	10^{-10}	10^{-12}
x_1 to $x_{0.95N}$ (Determ.)	1	1	1	1
x_1 to $x_{0.95N}$ (Robust)	1.1556	1.1556	1.1556	1.1556
$x_{0.95N+1}$ to x_N (Determ.)	10^{-2}	10^{-3}	10^{-4}	10^{-5}
$x_{0.95N+1}$ to x_N (Robust)	$0.1{+}10^{-2}$	$0.1{+}10^{-3}$	$0.1{+}10^{-4}$	$0.1{+}10^{-5}$
Function value (Determ.)	95.00005	950.0005	9500.005	95000.05
Function value (Robust)	110.2820	1102.820	11028.20	110282.0
Function calls (Determ.)	506	2.0×10^3	2.0×10^4	2.0×10^5
Function calls (Robust)	744	2.3×10^4	1.9×10^5	1.9×10^6
Fn. calls (Li et al. 2006)	$>10^9$	$>10^9$	$>10^{12}$	$>10^{12}$

have taken a lot of computation time. However, the modified Benders method solved all cases and produced only a linear increase in computational effort as the number of variables increased.

6.4 Welded beam design problem (Engineering example)

This example is a well-known welded beam problem from Ragsdell and Phillips (1976). In this problem, a beam A is to be welded to a rigid support member B. The beam has a rectangular cross-section and is made out of steel. The beam is designed to support a force F = 6000 lbf acting at the tip of the beam, and there are constraints on the shear stress, normal stress, deflection, and buckling load on the beam. The problem has four continuous design variables, and they are: thickness of the weld (h), length of the weld (l), thickness of the beam (t), and width of the beam (b). All variables are in inches. The objective of the problem is to minimize the total cost $f(x)$ of making such an assembly. For complete formulation of the robust optimization problem including specific values of the parameters, please refer to Gunawan and Azarm (2004).

The solution from the modified Benders method is different from the solution of Gunawan and Azarm (2004). First, the modified Benders method's nominal solution is closer to an actual solution from an earlier paper by Ragsdell and Phillips (1976) who provided an optimal objective function value of $f = 2.38$ while Gunawan and Azarm (2004) provided $f = 2.39$. Second, the robust solution is also lower in function value but still feasible. The robust solution is also feasible for all realizations of uncertainty.

This example highlights the strength of the modified Benders method over previous methods. The method of Gunawan and Azarm (2004) involves a backward mapping approach, which is known to omit solutions. The modified Benders method, while giving a better solution (lower in function value), is also computationally less expensive. The solution is displayed in Table 7.

6.5 Heat exchanger design problem (Engineering example)

For this example (Magrab et al. 2004), the energy balance on a heat exchanger can be written as

$$Q = UAF\Delta T_m = (mc_p)_h(T_{h1} - T_{h2}) = (mc_p)_c(T_{c2} - T_{c1}) \tag{27}$$

Several equations govern the above heat transfer. The above (27) will be used as an objective function that is to be maximized, as well as constraints that restrict the structure, in particular constraints on the pressure drop on the tube side (Δp_t) and shell side (Δp_s). Subscript 1 denotes the fluid

Table 7 Results of welded beam example

Info	Gunawan and Azarm (2004) nominal sol.	Modified Benders nominal sol.	Gunawan and Azarm (2004) robust sol.	Modified Benders robust sol.
h	0.241	0.2444	0.246	0.2392
l	6.158	6.2186	5.461	5.6753
t	8.5	8.2915	9.138	9.1225
b	0.243	0.2444	0.248	0.2392
$f(x)$	2.39	2.3807	2.48	2.4236
Function calls	N/A	8	250	38

Table 8 Design variables and parameters with uncertainty

Symbol	Unit	Description	Uncertainty
d_i	m	Tube inside diameter (variable)	$d_i \pm 0.001$
m_t	kg/s	Tube-side mass flow rate (variable)	$m_t \pm 1$
m_s	kg/s	Shell-side mass flow rate (variable)	$m_s \pm 1$
D_s	m	Shell inside diameter (variable)	$D_s \pm 0.01$
P_T	m	Pitch size (variable)	$P_T \pm 0.01$
T_{h1}	K	Inlet temperature of hot fluid	65 ± 1
T_{c1}	K	Inlet temperature of cold fluid	18 ± 1
ktube	W/(m K)	Thermal conductivity of tubes	60 ± 1

Table 9 Results for heat exchanger design

Information	Nominal solution	Robust solution
Q	1006.77	906.09
d_i	0.0160	0.0149
m_t	10	9
m_s	14	14
D_s	0.3900	0.3900
P_T	0.0240	0.0311
Function calls	49	984

entering while subscript 2 denotes it leaving; c denotes the cold fluid and h the hot fluid. In this example, cold water is in the tubes and hot water is on the shell side and the problem has been setup in a counterflow arrangement for a 124 tubes and two-pass heat exchanger. Table 8 lists the design variables and parameters with uncertainty (Fig. 3).

The optimization problem is the following:

$$
\begin{aligned}
&\max Q\\
&s.t.\\
&\Delta p_s \leq 35000\\
&\Delta p_t \leq 50000\\
&T_{h2} \leq 60\\
&N_P A_T \leq A_S\\
&-C \leq 0\\
&L \leq 40\\
&7 \leq \tilde{m}_t \leq 10\\
&13 \leq \tilde{m}_s \leq 16\\
&0.001 \leq \tilde{D}_s \leq 2\\
&0.002 \leq \tilde{P}_T \leq 0.1\\
&0.0001 \leq \tilde{d}_i \leq 0.1
\end{aligned}
\tag{28}
$$

Table 9 shows the results.

The thing to note about this example is that with less than an average of 1% uncertainty, the objective function value decreases by almost 10%. Hence, in the design of any model, it is important to consider the uncertainty in the problem, which can lead to different designs as well. This problem was also tried with the method of Li et al. (2006) however after 10^9 function calls, the approach was stopped. For the complete formulation of this problem, see Magrab et al. (2004) and the Appendix for definition of terms in the heat exchanger design.

7 Concluding remarks

This paper presents an efficient robust optimization approach to solve problems that have parameters and/or decision variables with interval uncertainty. The proposed modified Benders method obtains robust optimal solutions to linear programming, quadratic programming, convex and non-convex programming problems. The approach is computationally tractable and is tested with 14 numerical and engineering examples with the most general being nonlinear (non-convex) objective function and nonlinear (non-convex) constraint robust optimization problems. The modified Benders method provides an approximate locally optimal robust solution to general nonlinear robust optimization problems, with a way to improve this approximation if desired.

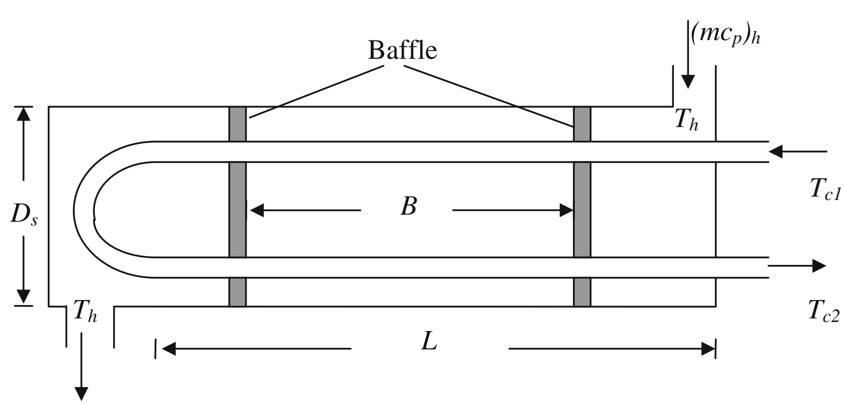

Fig. 3 Heat exchanger schematic (Magrab et al. 2004)

The test examples show the strength of this method when compared to two previous approaches. Not only is the method computationally efficient, but also obtains better solutions when compared to these previous methods. The method is scalable, that is, number of function calls increases at most linearly with an increase in number of variables.

Acknowledgements The authors wish to thank Dr. Mian Li and Mr. Amir Mortazavi for their constructive ideas to improve the paper. The work presented in this paper was supported in part by the Office of Naval Research Contract N000140810384. Such support does not constitute an endorsement by the funding agency of the opinions expressed in this paper.

Appendix: Test problems formulations

For examples 2 to 4: $q_1 = -1, q_2 = -1, \Delta q_1 = 0.1$, $\Delta q_2 = 0.1$, $\Delta x_3 = 0.1$.

(Example 2)

$$\begin{aligned}
&\min_x \left((x_1 - 0.6)^2 + (x_2 - 0.6)^2 - x_3 - x_4 + 10\right) \\
&s.t. \\
&(q_1 + \hat{q}_1) + x_1 + x_2 \le 0 \\
&(q_2 + \hat{q}_2) + x_3 + x_4 \le 0 \\
&-x_1 \le 0,\ -x_2 \le 0,\ -x_3 \le 0,\ -x_4 \le 0 \\
&where: \ \forall \hat{q}_1 \in \{-\Delta q_1, \Delta q_1\},\ \forall \hat{q}_2 \in \{-\Delta q_2, \Delta q_2\}
\end{aligned} \tag{29}$$

(Example 3)

$$\begin{aligned}
&\min_x \left((x_1 - 0.6)^2 + (x_2 - 0.6)^2 - \tilde{x}_3 - x_4 + 10\right) \\
&s.t. \\
&(q_1 + \hat{q}_1) + x_1 + x_2 \le 0 \\
&(q_2 + \hat{q}_2) + \tilde{x}_3 + x_4 \le 0 \\
&-x_1 \le 0,\ -x_2 \le 0,\ -\tilde{x}_3 \le 0,\ -x_4 \le 0 \\
&where: \ \forall \hat{q}_1 \in \{-\Delta q_1, \Delta q_1\},\ \forall \hat{q}_2 \in \{-\Delta q_2, \Delta q_2\}, \\
&\qquad \forall \tilde{x}_3 \in \{x_3 - \Delta x_3, x_3 + \Delta x_3\}
\end{aligned} \tag{30}$$

Table 10 Solution to Example 2

Information	Nominal solution	Robust solution	Solution of Li et al. (2006)
x_1	0.5	0.45	0.375
x_2	0.5	0.45	0.375
x_3	1	0.90	0.416
x_4	0	0	0.416
$f(x)$	9.02	9.145	9.268
Function calls	7	19	2592

Table 11 Solution to Example 3

Information	Nominal solution	Robust solution	Solution of Li et al. (2006)
x_1	0.5	0.45	0.375
x_2	0.5	0.45	0.375
x_3	1	0.10	0.416
x_4	0	0.70	0.416
$f(x)$	9.02	9.145	9.268
Function calls	7	21	2808

(Example 4)

$$\begin{aligned}
&\min_x \left((x_1 - 0.6)^2 + (x_2 - 0.6)^2 - \tilde{x}_3 x_4 + 10\right) \\
&s.t. \\
&(q_1 + \hat{q}_1) + x_1 + x_2 \le 0 \\
&(q_2 + \hat{q}_2) + \tilde{x}_3 + x_4 \le 0 \\
&-x_1 \le 0, -x_2 \le 0, -\tilde{x}_3 \le 0, -x_4 \le 0 \\
&where \quad \forall \hat{q}_1 \in \{-\Delta q_1, \Delta q_1\}, \forall \hat{q}_2 \in \{-\Delta q_2, \Delta q_2\}, \\
&\qquad \forall \tilde{x}_3 \in \{x_3 - \Delta x_3, x_3 + \Delta x_3\}
\end{aligned} \tag{31}$$

(Example 5)

$\Delta x = \Delta p = 0.1$

$$\begin{aligned}
&\min_x (2x_1 + 3x_2 - 5x_3 - 2x_4 + 3x_5) \\
&s.t. \\
&\tilde{x}_1 + \tilde{x}_2 - \tilde{2}x_3 - x_4 + 3x_5 - 1 \ge 0 \\
&2\tilde{x}_1 - 2\tilde{x}_2 + 3x_3 - x_4 + x_5 - 3 \ge 0 \\
&-5 \le \tilde{x}_i \le 5, \quad i = 1, 2 \\
&-5 \le x_i \le 5, \quad i = 3, 4, 5
\end{aligned} \tag{32}$$

Table 12 Solution to Example 4

Information	Nominal solution	Robust solution	Solution of Li et al. (2006)
x_1	0.5	0.45	0.40
x_2	0.5	0.45	0.40
x_3	1	0.90	0.40
x_4	0	0	0.40
$f(x)$	9.77	9.8850	9.92
Function calls	7	21	2592

Table 13 Solution to Example 5

Information	Nominal solution	Robust solution	Solution of Li et al. (2006)
x_1	−4.00	−3.40	1.04
x_2	−5.00	−4.90	−4.53
x_3	5.00	5.00	5.00
x_4	−5.00	−5.00	−0.37
x_5	5.00	5.00	5.00
$f(x)$	−23.00	−21.50	−20.75
Function calls	5	17	7856

(Example 6)

$\Delta x = \Delta p = 0.1$

$$\begin{aligned}
&\min_x (2.1x_1 + 3.07x_2 - 5x_3 - 2x_4 + 2.4x_5)\\
&s.t.\\
&0.9\tilde{x}_1 + \tilde{x}_2 - \tilde{2.2}x_3 - 1.1x_4 + 3.5x_5 - 1.2 \geq 0\\
&2\tilde{x}_1 - 2\tilde{x}_2 + 3x_3 - x_4 + x_5 - 10 \geq 0\\
&-5 \leq \tilde{x}_i \leq 5, \quad i = 1, 2\\
&-5 \leq x_i \leq 5, \quad i = 3, 4, 5
\end{aligned} \tag{33}$$

(Hock 100)

This is problem 100 modified from Hock and Schittkowski (1980). $\Delta x = \Delta p = 0.1$

$$\begin{aligned}
&\min_x \big((\tilde{x}_1 - 10)^2 + 5(\tilde{x}_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2\\
&\qquad + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7\big)\\
&s.t.\\
&\tilde{127} - 2\tilde{x}_1^2 - 3\tilde{x}_2^4 - x_3 - 4x_4^2 - 5x_5 \geq 0\\
&282 - 7\tilde{x}_1 - 3\tilde{x}_2 - 10x_3^2 - x_4 + x_5 \geq 0\\
&196 - 23\tilde{x}_1 - \tilde{x}_2^2 - 6x_6^2 + 8x_7 \geq 0\\
&-4\tilde{x}_1^2 - \tilde{x}_2^2 + 3\tilde{x}_1\tilde{x}_2 - 2x_3^2 - 5x_6 + 11x_7 \geq 0
\end{aligned} \tag{34}$$

Table 14 Solution to Example 6

Information	Nominal solution	Robust solution	Solution of Li et al. (2006)
x_1	−5.00	−4.90	−4.23
x_2	−5.00	−4.90	−4.52
x_3	5.00	5.00	5.00
x_4	−3.82	−4.27	−3.67
x_5	5.00	5.00	5.00
$f(x)$	−31.21	−29.79	−28.36
Function calls	5	17	11099

Table 15 Solution to Hock 100

Information	Nominal solution	Robust solution	Solution of Li et al. (2006)
x_1	2.3304	2.2350	–
x_2	1.9514	1.8546	–
x_3	−0.4775	−0.4749	–
x_4	4.3657	4.3533	–
x_5	−0.6245	−0.6251	–
x_6	1.0381	1.0359	–
x_7	1.5942	1.5970	–
$f(x)$	680.6301	692.3847	–
Function calls	7	19	$>10^9$

(Hock 106)

$\Delta x = \Delta p = 0.1$

$$\begin{aligned}
&\min_x ((\tilde{x}_1) + (\tilde{x}_2) + (x_3))\\
&s.t.\\
&\tilde{1} - 0.0025(x_4 + x_6) \geq 0\\
&1 - 0.0025(x_5 + x_7 - x_4) \geq 0\\
&1 - 0.01(x_8 - x_5) \geq 0\\
&\tilde{x}_1x_6 - 833.33252x_4 - 100\tilde{x}_1 + 83333.333 \geq 0\\
&\tilde{x}_2x_7 - 1250x_5 - \tilde{x}_2x_4 + 1250x_4 \geq 0\\
&\tilde{x}_3x_8 - 1250000 - \tilde{x}_3x_5 + 2500x_5 \geq 0\\
&100 \leq \tilde{x}_1 \leq 10000,\ 1000 \leq \tilde{x}_2 \leq 10000,\\
&1000 \leq \tilde{x}_3 \leq 10000,\ 10 \leq x_i \leq 1000, \quad i = 4, \ldots, 8
\end{aligned} \tag{35}$$

Table 16 Solution to Hock 106

Information	Nominal solution	Robust solution	Solution of Li et al. (2006)
x_1	579.32	388.73	–
x_2	1359.94	1540.21	–
x_3	5110.07	5290.11	–
x_4	182.02	150.89	–
x_5	295.60	288.40	–
x_6	217.98	209.11	–
x_7	286.42	262.49	–
x_8	395.60	388.40	–
$f(x)$	7049.33	7219.06	–
Function calls	5	17	$>10^9$

(6.5: Heat Exchanger Design)

Table 17 Additional details for heat exchanger formulation

Symbol	Units	Description	Value
c_p	J/(kg K)	Specific heat at constant pressure	Variable
d_0	m	Tube outside diameter	Variable
f		Tube flow friction factor	Variable
f_s		Friction factor shell side	Variable
h_0	W/(m^2 K)	Heat-transfer coefficient outside tube	Variable
h_i	W/(m^2 K)	Heat-transfer coefficient inside tube	Variable
k	W/(m K)	Thermal conductivity of fluids	Variable
Δp_s	Pa	Shell-side pressure drop	Variable
Δp_t	Pa	Tube-side pressure drop	Variable
u_t	M/s	Mean axial velocity of fluid in tube	Variable
A_0	m^2	Tube outside surface area per pass	Variable
A_i	m^2	Tube inside surface	Variable
A_s	m^2	Cross-flow area at or near shell centerline	Variable
A_t	m^2	Total cross-sectional area of tubes per pass	Variable
B	m	Baffle spacing	0.5
C	m	Clearance between adjacent tubes	Variable
C_L		Tube layout constant	1 (for 90°)
C_{TP}		Tube count calculation constant	0.90
D_e	m	Equivalent diameter of shell	Variable
F		LMTD correction factor	Variable
L	m	Tube length	Variable
N_b		Number of baffles (Integer = B/L)	4
N_T		Number of tubes	124
N_p		Number of tube passes	2
P_P	W	Pumping power of fluid in tubes	Variable
Pr		Prandtl number	Variable
Q	W	Heat-transfer rate	Variable
R_{fo}	(m^2 K)/W	Fouling resistance on outside of tube	0.00015
R_{fi}	(m^2 K)/W	Fouling resistance on inside of tube	0.00015
Re_b		Reynolds number at T_b	Variable
Re_s		Shell-side Reynolds number at T_b	Variable
ΔT_m	K	LMTD	Variable
T_{h2}	K	Outlet temperature of hot fluid	Variable
T_{c2}	K	Outlet temperature of cold fluid	315
T_b	K	Bulk temperature	Variable
T_w	K	Wall temperature	Variable
U	W/(m^2 K)	Average overall heat transfer coefficient based on A	Variable
φ_s		Viscosity correction factor	Variable
μ	kg/(s m)	Dynamic viscosity	Variable
μ_b	kg/(s m)	Dynamic viscosity at T_b	Variable
μ_w	kg/(s m)	Dynamic viscosity at T_w	Variable
ρ	kg/m^3	Density	Variable

"Variable" in the last column of Table 17 implies non-constant terms in the model and not decision variables which can be seen in Table 8

Table 18 Number of iterations and CPU time to solve problems

Test problem	Number of iterations	CPU (2.0 GHz, 4 GB RAM) time (s)
Example 1	3	0.560
Example 2	3	0.787
Example 3	3	1.654
Example 4	3	1.435
Example 5	3	3.821
Example 6	3	3.494
Hock 100	4	30.552
Hock 106	4	25.645
Fleury ($N = 10^2$)	6	9.328
Fleury ($N = 10^3$)	9	324.532
Fleury ($N = 10^4$)	12	886.321
Fleury ($N = 10^5$)	15	2123.453
Welded beam	4	1.606
Heat exchanger	6	45.234

The following equations are the ones coded into MATLAB and are selected from the whole formulation to provide further insight. For complete formulation and all equations used, please refer to Magrab et al. (2004).

$$T_{h2} = T_{h1} - \frac{(mc_p)_t}{(mc_p)_s}(T_{c2} - T_{c1}) \tag{36}$$

$$\Delta p_t = C_0 \frac{f m_t^3}{UF\Delta T_m} \tag{37}$$

$$\Delta p_s = \frac{f_s m_s^2 (N_b + 1) D_s}{2A_s^2 \rho D_e \phi_s} \tag{38}$$

$$L = \frac{(mc_p)_t (T_{c2} - T_{c1})}{\pi d_0 N_T F \Delta T_m} \tag{39}$$

$$C_0 = \frac{N_p (c_p)_t (T_{c2} - T_{c1})}{2\pi d_i d_0 N_T \rho_t A_t^2} \tag{40}$$

$$A_t = \frac{\pi d_i^2 N_T}{4N_P} \tag{41}$$

References

Balling R, Free J, Parkinson A (1986) Consideration of worst-case manufacturing tolerances in design optimization. J Mech Transm Autom Des 108:438–441

Bazaraa M, Sherali H, Shetty C (1993) Nonlinear programming: theory and algorithms, 2nd edn. Wiley, New York, NY, pp 438–448

Ben-Tal A, Nemirovski A (2002) Robust optimization-methodology and applications. Math Program, Ser B 92:453–480

Ben-Tal A, Nemirovski A (2008) Robust solutions to conic quadratic problems and their applications. Optim Eng 9:1–18

Ben-Tal A, El Ghaoui L, Nemirovski A (2009) Robust optimization. Princeton University Press, Princeton, NJ

Benders J (1962) Partitioning procedures for solving mixed-variables programming problems. Numer Math 4:238–252

Bertsimas D, Sim M (2006) Tractable approximations to robust conic optimization problems. Math Program, Ser B 107:5–36

Conejo AJ, Castillo E, Minguez R, Garcia-Bertrand R (2006) Decomposition techniques in mathematical programming. Springer, New York, NY

Elishakoff I, Haftka R, Fang J (1994) Structural design under bounded uncertainty-optimization with anti-optimization. Comput Struct 53:1401–1405

Elishakoff I, Ohsaki M (2010) Optimization and anti-optimization of structures under uncertainty. Imperial College Press, London, UK

Gabriel S, Shim Y, Conejo A, de la Torre S, Garcia-Bertrand R (2009) A benders decomposition method for discretely-constrained mathematical programs with equilibrium constraints. J Oper Res Soc September(61):1–16

GAMS (2009) General algebraic modeling system. GAMS Version 22.9.

Ganzerli S, Pantelides CP (1999) Load and resistance convex models for optimum design. Struct Optim 17:259–268

Groenwold A, Etman L (2010) On the conditional acceptance of iterates in SAO algorithms based on convex separable approximations. Struct Multidisc Optim 42:165–178

Gunawan S, Azarm S (2004) Non-gradient based parameter sensitivity estimation for single objective robust design optimization. J Mech Des 126(3):395–402

Hock W, Schittkowski K (1980) Test examples for nonlinear programming codes. Springer, New York, NY, USA

Lagaros N, Papadrakakis M (2007) Robust seismic design optimization of steel structures. Struct Multidisc Optim 33:457–469

Lee SH, Chen W, Kwak BM (2009) Robust design with arbitrary distributions using gauss-type quadrature formula. Struct Multidisc Optim 39:227–243

Li M, Azarm S, Boyars A (2006) A new deterministic approach using sensitivity region measures for multi-objective and feasibility robust design optimization. J Mech Des 128(4):874–883

Li M, Gabriel S, Shim Y, Azarm S (2010) Interval uncertainty-based robust optimization for convex and non-convex quadratic programs with applications in network infrastructure planning. Netw Spatial Econ (accepted)

Lu X, Li H, Chen C (2010) Variable sensitivity-based deterministic robust design for nonlinear system. J Mech Des 132:064502-1

Magrab E, Azarm S, Balachandran B, Duncan J, Herold K, Walsh G (2004) An engineer's guide to Matlab. Prentice Hall, New York, NY

MATLAB (2008) MATLAB and simulink for technical computing. Mathworks, Version 2008b

Montemanni R (2006) A benders decomposition approach for the robust spanning tree problem with interval data. Discrete Optim 174:1479–1490

Ng TS, Sun Y, Fowler J (2010) Semiconductor lot allocation using robust optimization. Eur J Oper Res 205:557–570

Qiu Z, Wang X (2010) Structural anti-optimization with interval design parameters. Struct Multidisc Optim 41:397–406

Ragsdell KM, Phillips DT (1976) Optimal design of a class of welded structures using geometric programming. J Eng Ind Ser B 98(3):1021–1025

Saito H, Murota K (2007) Benders decomposition approach to robust mixed integer programming. Pacific J Optim 3(1):99–112

Soyster AL (1973) Convex programming with set-inclusive constraints and applications to inexact linear programming. Oper Res 21:1154–1157

Velarde J, Laguna M (2004) A benders-based heuristic for the robust capacitated international sourcing problem. IIE Trans 36:1125–1133

Youn B, Xi Z (2009) Reliability-based robust design optimization using the Eigenvector Dimension Reduction (EDR) method. Struct Multidisc Optim 37:475–492

Zhu J, Ting K (2001) Performance distribution analysis and robust design. J Mech Des 123(1):11–17

Zou T, Mahadevan S (2006) A direct decoupling approach for efficient reliability-based design optimization. Struct Multidisc Optim 31:190–200