Research Project on Robust Optimization

PROJECT REPORT

Anish Pokhrel
Robin White
Project 2
Introduction to Engineering Design Optimization
MSE 726

Submitted to—
Dr. Gary Wang
Mechatronics Systems Engineering
Simon Fraser University

Date: 4/9/2015

Contents

Contributions:
Robin White created the robust model using GA implementation, ran calculations and compared to literature values. Anish Pokhrel performed literature search and ran calculations using the model. Both contributed to the report, sections 1 by Anish and 3,4 by Robin. Both contributed to introduction and conclusions.

1. Abstract

The usefulness of optimization approach in solving systems has been demonstrated extensively in various fields such as science, engineering, and business. One problem however, is that often times the optimized solution may not be physically realistic or that it may be too sensitive to real world fluctuations and error, making it an idealized solution, that is a normal optimization point is not always applicable to the real world, it is only suitable for an ideal situation with no fluctuation. This was soon realized in the optimization community and since then, robust optimization has been a growing field of study for its usefulness in application to problems in the field. This project focuses on summarizing various current robust optimization methodology techniques. However, implementation of this useful methodology is sometimes extremely difficult for a user with limited optimization background. A simple robust optimization method is presented here using only Matlab's genetic algorithm search routine. A detailed literature comparison on various different mathematical problems is discussed and the usefulness as well limitations of this simple routine are described in detail.

2. Introduction

Robust Optimization is an approach for modeling optimization problems under uncertainty, where the modeler aims to find decisions that are optimal for the uncertainties within a given set [1]. Usually the uncertain optimization models first alter the objective function and the constraints such that the feasible area is reduced and then solves using a standard optimization algorithm. The following figure is a graphical representation of a robust design vs a non-robust design. As shown in the figure, a robust design has less variance than a non-robust design, which has larger uncertainty and higher line slope. In order for an optimization algorithm to determine the robust solution, generally the 'robustness' needs to be defined in some way.

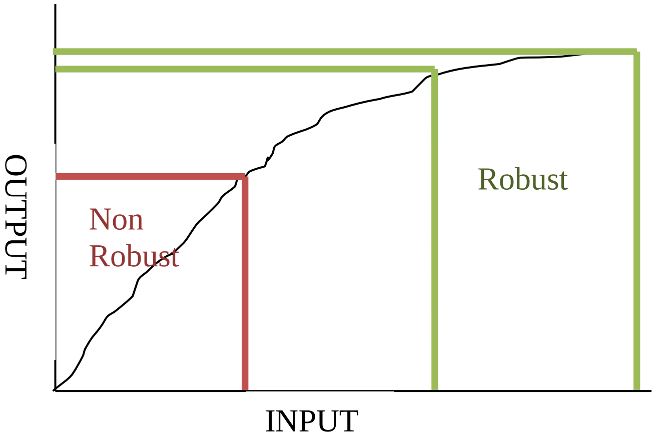

Fig 1: A Robust Design vs a non-Robust Design

Here is another example of a robust Solution. As you can see in the figure, an initial deterministic solution is graphed against a final robust solution. The y axis values on the graphs have less variance on the robust solution graph than the deterministic solution graph.

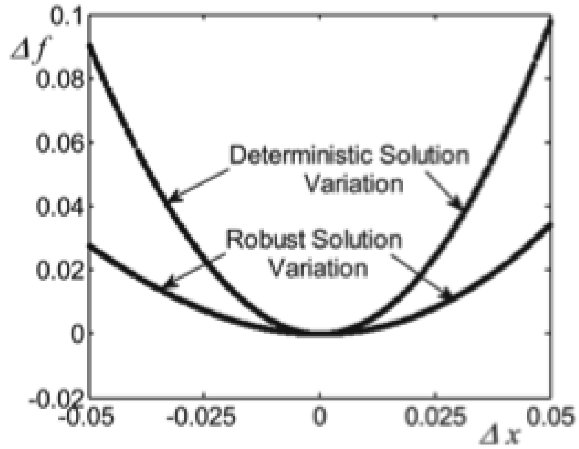

Fig 2: A Deterministic solution vs a Robust Solution

In the paper about evolutionary multi objective robust optimization by Ferreira, Fonseca, Covas, and Cunha they use expectation measure function to replace the original single objective function by a measure of performance and expectation in the vicinity of the solution. As shown in the figure below, graph A has five different peaks. A standard optimization algorithm will pick peak one to be the maximum optimized solution. However, it is clear from the graph that peak three is more robust solution. After an expectation measure has been used on graph A, a new graph B is formed. Now a standard optimization algorithm will choose peak 3 to be the maximum optimized solution, which is also the robust solution. Another criterion is added to the objective function that measures the deviation around the vicinity of the design point. So in case of a single objective function the algorithm runs a performance and robustness optimization simultaneously.

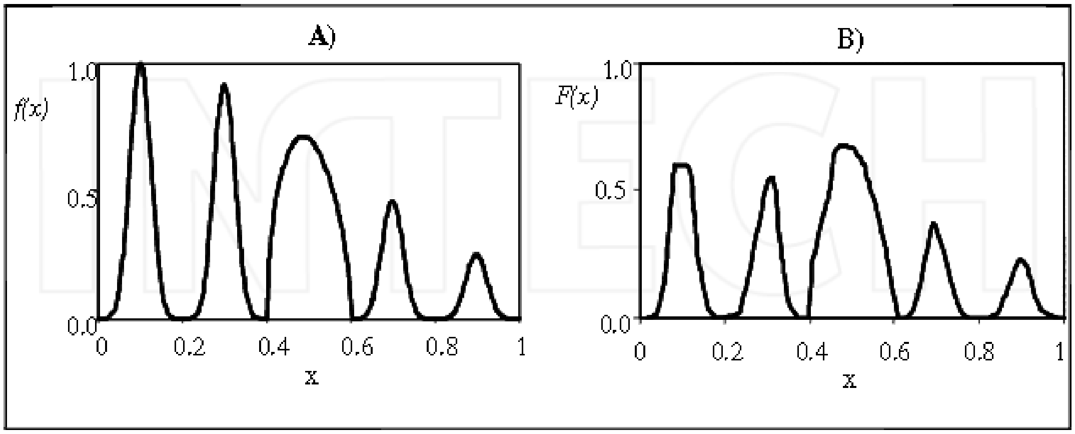

Figure 3: Single objective optimization

However, Messac and Yahaya use physical programming method to solve a robust multi objective optimization problem. Physical programming is an optimization method that addresses important optimization issues in physical designs or systems that lend themselves to a qualitative and quantitative physical description of the designer's preferences [9]. There is a five step process in physical programming. They are choosing design metrics, choose design parameters, develop mapping between design parameters and design metric's, develop aggregate objective function, and perform computational optimization. This optimization process is better when the designer needs to know the robustness separately for each design metric and design parameter

On the other hand, in a paper on Multi-Objective Genetic Algorithm for Robust Design Optimization by Li, Azarm, and Aute they present a new deterministic approach to investigate the tradeoffs between the performance and robustness of optimum solutions, based on a Multi objective Genetic Algorithm (MOGA)[3]. Their approach simultaneously optimizes the performance by calculating the fitness value, which accounts for the objective and constraint values and measures the solution's robustness by calculating the robustness index. They use Robust MOGA (RMOGA) Approach which defines two objective functions. The first objective function defines the performance and the second objective function defines the robustness index (η) of the design solution. Following is the example of a formulation of a RMOGA problem .

$$\min_{x} \; f_v(f_1, \cdots, f_M, g_1, \cdots, g_G)$$

$$\max_{x} \; \eta = \frac{R}{R_E} \quad [2]$$

The RMOGA gives a set of solutions that are Pareto optimum in terms of the performance (i.e. minimization of fitness value) and the robustness (i.e. maximization of the robustness index [3]. The fitness value and robustness index both accounts for the variation in objectives and constraints of the problem. Hence the tradeoff between the performance and the robustness of a design can be made by the user using the Pareto optimum. This optimization technique will be useful in manufacturing because sometimes higher cost and simpler design or vice versa would be needed depending on the requirements.

In a Paper about "Manufacturing System Robustness through Integrated Modeling" by Suri and Otto, they use the entire process variable simultaneously in the design unlike the traditional parameter design method where each process operation is optimized separately. This process allows for a concurrent change in operating points and can result in lower end of line variations [4]. A sheet stretch forming system which is used in aerospace manufacturing has been used as an example to demonstrate the difference in the traditional robust design approach vs the integrated modeling approach. The traditional approach resulted in a 14 percent decrease in the end of line variation whereas this new approach resulted in 24 percent decrease in the end of line variation with minor process variable changes. This process is a good optimization technique in manufacturing as well because sometimes there is several factors interaction affecting the output variation. It would be a good tool in production line with high mix of products.

Beyer and Sendhoff's paper on "Robust Optimization-A comprehensive survey" talks about Taguchi's method of Robust Optimization. Taguchi, who is recognized as the father of robust optimization defines a three staged design methodology. They are System design, Parameter design, and Tolerance design. System design determines the basic performance parameters of the product and its general structure, Parameter design optimizes the design parameters in order to meet the qualification, and Tolerance design fine tunes the design parameters obtained in the second stage [5].Taguchi's method accounts for the performance variation due to noise factors which are beyond the control of the designer. The objective function has a control parameter and a noise factor. Taguchi defined an optimization technique that maximizes the signal to noise ratio. He doesn't use the standard optimization algorithm instead uses design of experiments in order to assess design variables. A statistical analysis is performed on a special data array which identifies the variable producing the best performance. However, Trosset points out that "the Taguchi approach violates a fundamental tenet of numerical optimization –that one should avoid doing too much work until one nears a solution."[6]. Taguchi's Loss function was used by Ramakrischnan and Rao in their paper with statistical concepts and nonlinear optimization techniques. On their problem model the deviation of the variables are given and on the same problem they use the deviation as a variable which is used in the optimization model. The numerical evaluation of total loss was designed so that it required sensitivity information. The total loss was the variation of all the statistical quantities used [7]. According to the paper by Welch, he first combined the control and noise factors into a single array. Then he modeled the response by making an estimated prediction model based on a fitted response model [8].

In the paper on "A Robust optimization Approach to Inventory theory", Bertsimas and Thiele propose a supply chain optimal control robust optimization with respect to stochastic random demands. They use the deterministic and numerically tractable methodology. An equivalent model without uncertainty is built using the robust optimization ideas. The model is a linear programming problem if there are no fixed costs throughout the supply chain and it is a mixed integer programming problem if fixed costs are present [10]. This model uses very little information on the demand distribution and is therefore widely applicable. This approach provides good framework to analyze complex supply chains and is not only limited to benchmark problems where the optimal stochastic policy is known.

The use of Matlab as a computational engine has found its use into all aspects of science and engineering, and there is no exception for the case of robust optimization. Various implementations of interfaces to use Matlab's Toolboxes have been formulated and available for use. Goh and Sim developed a so-called Robust Optimization Made Easy (ROME), an algebraic modeling toolbox for modeling robust optimization problems [1].

ROME is built in within MATLAB which gives users the advantage of using MATLAB's numerical and graphical capabilities. ROME can also be used as a subroutine within a MATLAB model. ROME is built with similar syntax and construct as MATLAB which helps new ROME users to learn quickly. A distinctive advantage of ROME than any other toolbox is that robust optimization model cannot be directly modeled in any other toolbox with an exception of YALMIP. However, the scope of ROME is limited in terms of the different types of deterministic optimization models it can model.

Another integration toolbox using Matlab is YALMIP, developed by Löfberg [11], which uses a significant list of toolboxes developed by various other parties, but combines them into a simple to use search interface to find the best solver for a given problem. The issue with this of course that 99% of the code is hidden and customization as well as understanding becomes increasing difficult as well as a significant number of free or commercial toolboxes need to downloaded onto the users system. This kind of tool is useful if a significant number of various optimization problems are solved are to be solved by the user however it is felt that there is significant disconnect between the output and how it was determined, unless significant knowledge of various higher order optimization methods is known.

In this report, a simple, easy to incorporate robust optimization routine is presented as a way to clearly illustrate robust optimization in a simplistic manner which could be used and modified by a user with only an introductory knowledge of optimization routines, as was presented in this course.

3. Variance Genetic Algorithm model

After extensive research in current topics in the field of robust optimization, it was decided that an application of the knowledge gained could be put to some use in creating a very simple but effective algorithm for robust optimization that is easy to understand, implement and requires only Matlab optimization toolbox. The simplicity in the algorithm draws on the fact that many of the topics being used in robust optimization are beyond the scope of the context of the course and thus implementation became difficult to say the least. The model presented here uses Matlab Genetic Algorithm as its base and some statistical knowledge. Since we know that there is more than one optimum point in a robust optimization problem, a global optimizer was required, although as will be seen in the next section, some other robust optimization techniques in current literature used fmincon [13] which is surprising since we know from this course that the starting position has a very strong influence on the optimization point for any multi-modal problem. Another extremely useful feature of using GA is that it requires no knowledge of the gradient, which is helpful for more real-world problems which robust optimization is designed to address.

Since GA is designed to find the global optimum (minimum) the objective function needs to be changed in a way that the robust optimum is the most tantalizing optimum. To do this we employed the knowledge of the “penalty method” [14], whereby the objective function is altered based on a trait that would make robust peaks easier to find than non-robust peaks. The trait that we used in this algorithm design was the variance. This algorithm design only considers variations in the input (independent) variables for the objective, many other robust optimization methods consider variations in the output as well as constraint variations, however for this simple model we only consider the input variables and leave implementation of variance in the objective as well as the constraints for further iterations.

Consider a generic optimization problem $\begin{array}{l} \min\limits_{x} f(x) \\ s.t. \quad g_j(x) \le 0, j = 1..J \end{array}$

The variance is defined as a non-negative measure of the extent at which a set of values is spread out. A large spread has high variance and the smaller the spread the smaller the variance, and more robust, as shown in figure 1. The variance is defined mathematically as:

$$V = \frac{1}{N-1} \sum_{i=1}^{N} \left| A_i - \mu \right|^2$$

for a vector A made up of N observations, where μ is the mean defined as,

$$\mu = \frac{1}{N} \sum_{i=1}^{N} A_i$$

In order to project the variance onto the objective function, N random observations needed to be made within a given allowed error of the input parameter (independent variable) x. To do this multiple random observation points of f were taken within the domain of $x \pm \Delta x$, where Δx is the worst-case error in x, using these points the variance could then be calculated.

Since the robustness of an optimization point is best for the smallest variance, the robust objective was defined as $f(x)/V^q$, where q is typically set to 1. This is illustrated in the example function below where the nominal point and the robust point are indicated on the objective function and the altered objective function due to the variance is shown above.

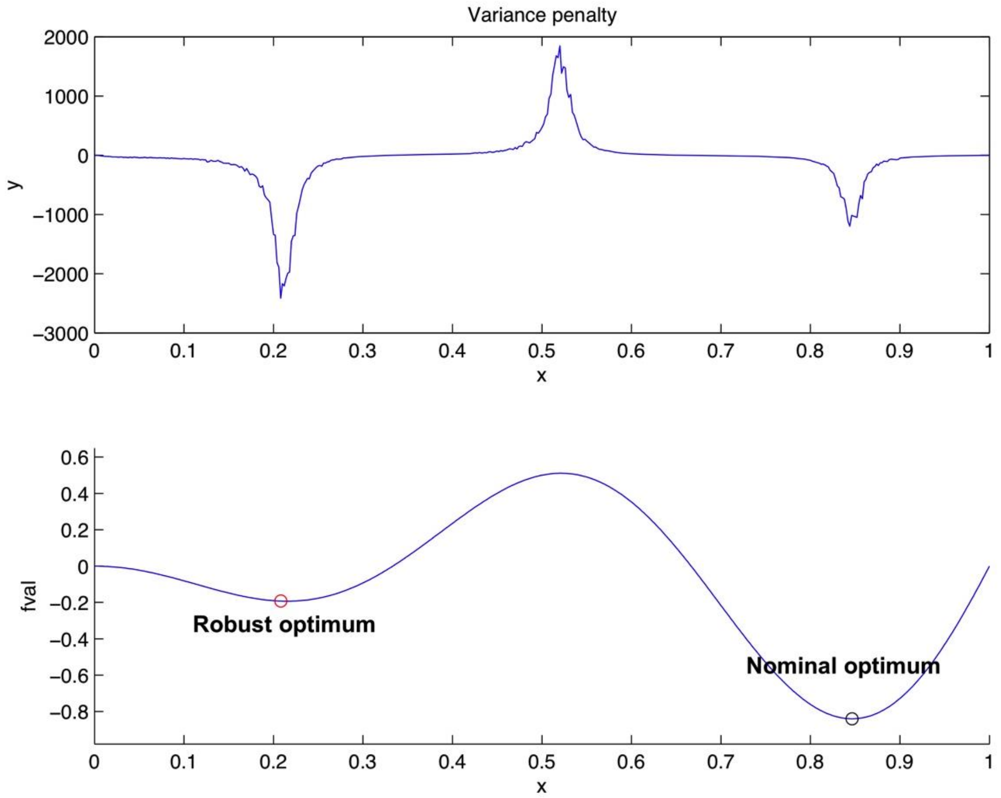

Figure 3: Variance-penalty GA robust optimization for f = -x*sin(3*pi*x), see appendix for code

Specific values and comparisons are shown in table 1, other more complex examples are also discussed in the next section; however this example clearly illustrates the effectiveness of the simple algorithm designed here.

4. Literature comparisons

In this section we will investigate the application of this algorithm further. There are some limitations in the simplicity of the design, however the results are also difficult to fully compare as there is even variation in the robust optimum between different literature sources. So really, who is to say what the ‘best’ robust optimum is. Since the robust optimum is used to provide a more realistic application of optimization problems, then we can say the best robust solution will be the one that best suits the objective in question to be decided by the decision maker. Therefore several robust optimization solutions should be provided to yield the best decision and understanding of each limitation should be known. Let us then first look at some examples of its application.

Consider the problem presented above in figure 3. For this simple problem very good agreement is reached between an algorithm presented by Zhou et al that uses a robust index and solve by sequential quadratic programming (SQP) [12] and the variance GA presented here (see table 1)

Table 1

F = -x*sin(3*pi*x)	***x****	***F****
Nominal [12]	**0.8466**	
Nominal (GA)	**0.8465**	**-0.8400**
Robust [12]	**0.2153**	
Robust (V. GA)	**0.2145**	**-0.1931**

Let us now consider a problem of higher dimension as well as with non-linear constraints. The problem in question is described as [12]:

$$\min_{x} x_1^3\sin\left(x_1+4\right)+10x_1^2+22x_1+5x_1x_2+2x_2^2+3x_2+12$$

$$\text{s.t. } x_1^2+3x_1-x_1\sin\left(x_1\right)+x_2-2.75\le 0$$

$$-\log\left(0.1x_1+0.41\right)+x_2\mathrm{e}^{-x_1+3x_2-4}+x_2-3\le 0$$

Results for the optimization and robust analysis are shown in table 2 below.

Table 2

	x*	***F****	***Variance***
Nominal [13]	**[-1.825, 0.741]**	**-3.287**	
Nominal (GA)	**[2.0579, 0.8276]**	**-3.7083**	**0.9836**
Robust [12]	**[-1.440, 0.3369]**	**-1.773**	**0.4755**
Robust [13]	**[-1.557, 0.477]**	**-2.266**	
Robust (V. GA)	**[-1.6211, 0.7017]**	**-2.9255**	**0.2953**

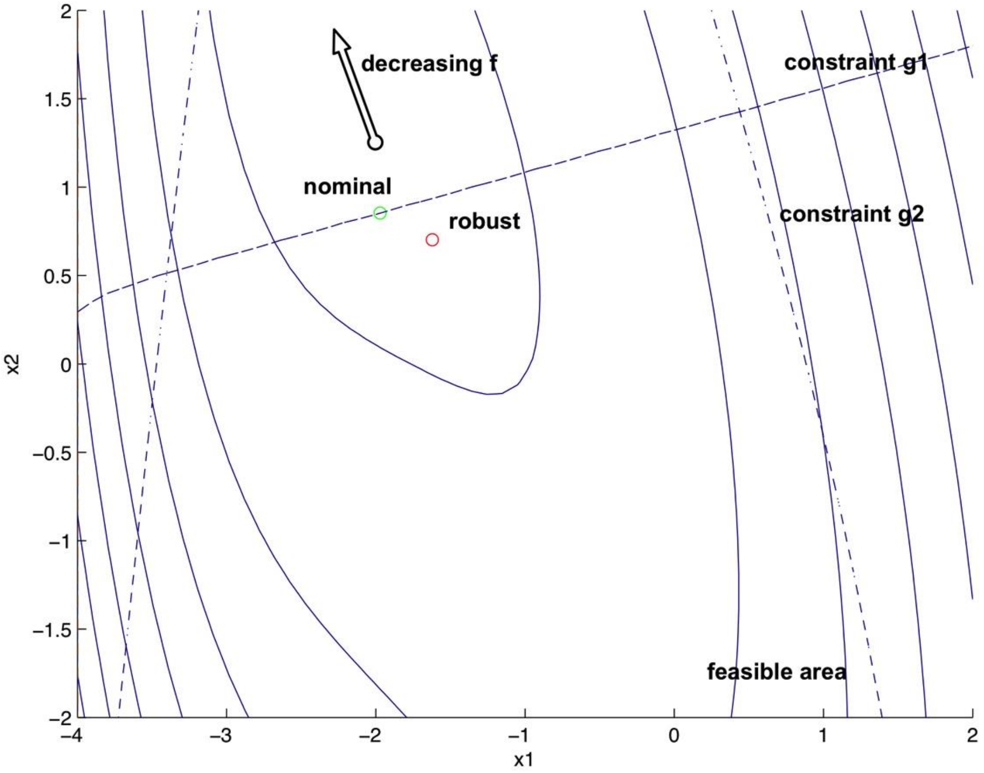

Figure 4: Contour for non-linear constraint problem showing nominal optimum and robust optimum by Variance GA method

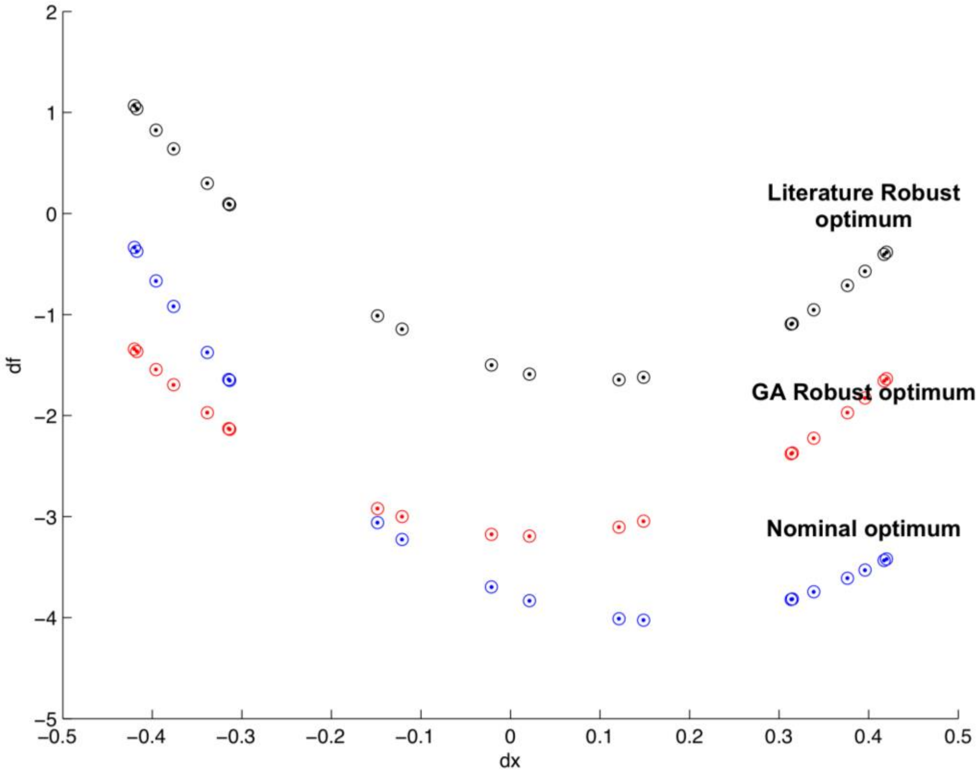

Figure 5: Variations around optimum point for nominal optimum and robust optimum from literature and using the variance GA method. The variance GA robust optimum shows the smallest variance in f over the range of dx as well as a better optimum

As can be seen in table 2, and figure 4, there is some difference between both robust optimum from literature and the variance GA method presented here. For all robust optimum presented, including the variance GA method, there is room for variation where as the nominal has no room as it lies right at boundary of active constraint g1 = 0. It should be noted that the literature robust optimum algorithms do not consider variation in the constraint either so that the comparison is fair. Also shown in table 2 is the variance value around the optimum point presented for the nominal solution, a robust solution from literature and the variance GA method. The variance is smallest for the GA method presented here indicating the most robust solution however, the allowed variation before the constraint boundary is not met. For the worst-case (WC) scenario [15] of dx = 0.4, the GA robust solution would go beyond the constraint boundary, this is the difference between the literature solution and the variance GA method. A constraint that the robust optimum must not lie within the WC radius would need to be implemented. However as can be seen in another literature robust solution, this also lies close enough to the constraint boundary that in a WC the optimum would be outside the feasible area. If we consider only the WC scenario, then using the variance GA method the optimum solution is improved to fall closer to other literature values, but still with better variance indicated as v in figure 6. This requires more specific knowledge of the variation in the input variables however and may be difficult to obtain in some cases.

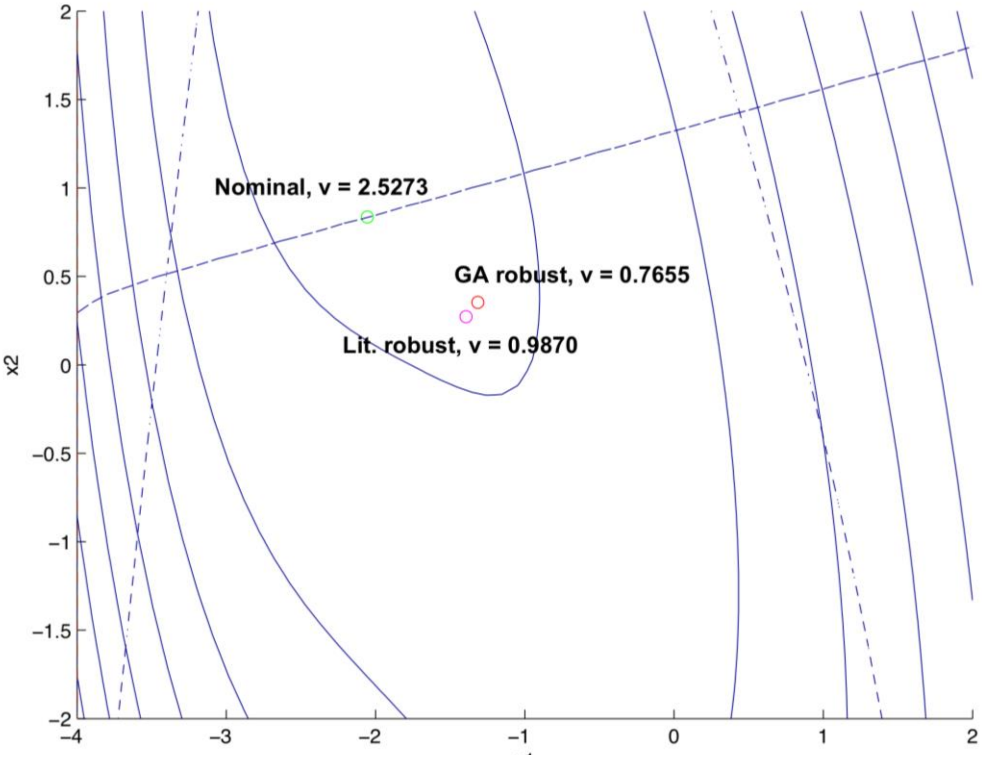

Figure 6: Worst-case scenario variance GA where the robust optimum is now away far enough from the constraint boundary with better variance than lit. robust optimum

It should be noted as well that in the case where the nominal optimum has a very large difference between the robust optimum or the variance of the robust and nominal are similar the penalty in the objective due to variance may not be sufficient enough for the GA search algorithm to properly find the robust optimum. In this case an adjustment power factor can be changed such that q > 1.

In the final consideration of the performance of the simple variance GA presented here it can be noticed that if not all input parameters are in the objective function, the variance will not penalize the objective function enough. This can be shown in the example function below, which is a consideration of a two-truss problem:

$$\min_{x} \quad f\left(x\right)=\frac{20\left(16+x_3^2\right)^{1/2}}{10^3 x_1 x_3}$$

$$\text{s.t.} \quad g_1 = f - 100 \le 0$$

$$g_2 = f_2 - 100 \le 0$$

$$g_3 = f_3 - 100 \le 0$$

$$f_2 = \frac{80\left(1+x_3^2\right)^{1/2}}{10^3 x_1 x_3}$$

$$f_3 = 10^3 \left[x_1 \left(16+x_3^2\right)^{1/2} + x_2 \left(16+x_3^2\right)^{1/2} \right]$$

Now x2 lies only in the constraint leading to difference in the robust optimum position shown in table 3

Table 3

	x*	***F****
Nominal [12]	**[0.0199, 0.0001, 3]**	
Nominal (GA)	**[0.0201, 0, 2.9548]**	**1.6749**
Robust [12]	**[0.0197, 0.0002, 2.9250]**	
Robust (V. GA)	**[0.0103, 0.0019, 1.1156]**	**7.2462**

Unfortunately, simply adding variation to the input variables does not work with Matlab’s GA algorithm and no solution in the feasible area can be found. This is the limitation with the current simple set-up. In order for this to be fixed a new method of allowing for constraint variation needs to be included into the GA search algorithm. Worst-case scenario could be easily implemented as additional constraint functions could be added to cover the limited number of possibilities in these cases.

5. CONCLUSION

A detail literature review was done to see what kind of robust optimization process has been used so far. Several robust optimization methodologies are an extension of Taguchi’s method. Taguchi’s robust optimization method accounts for the performance variation and noise factor in the model, which is the base for every robust optimization problems. It was found out that for several optimization processes there is not a specific robust optimization algorithm used. Instead the optimization problem has been modified to take account the different environmental uncertainties which cannot be controlled by the designer and then a standard optimization algorithm has been used to find the optimum solution. However, there are Matlab based toolbox like ROME and YALMIP, which allows a direct methodology to solve robust optimization problems. This methodology however implements an interface to run robust optimization problems using a large set of other commercial or freely distributed solver toolboxes. This makes the understanding of the process unclear and any variation to customize is difficult. For this reason, a simple robust optimization routine was constructed using a variance penalty method to alter the objective function and allow Matlab genetic algorithm search method to find the most robust solution described by the functions variance. This was compared to other robust optimization methods for three different examples showing its limitations as well as its usefulness in ease of use and accuracy. From this model an introduction to robust optimization is more easily implemented and can be expanded to more complex problems.

Work cited

1. Goh, J., M. Sim. 2009. Robust optimization made easy with ROME. Working paper, National University of Singapore
2. Ferreira, J., Fonseca, C.M., Covas, J.A., & Gaspar-Cunha, A. (2008). *Evolutionary multi objective robust optimization. Advances in evolutionary algorithms* (pp.261-278). Vienna, Austria: I-Tech Education and Publishing.
3. M. Li, A. Azarm, and V. Aute, “A multiobjective genetic algorithm for robust design optimization,” in Proc. Genetic Evol. Comput. Conf.,New York: ACM, 2005, pp. 771–778
4. R. Suri, K. Otto, Manufacturing system robustness through integrated modeling, Journal of Mechanical Design 123 (4) (2001) 630– 636
5. H.-G. Beyer, B. Sendhoff, Robust optimization – a comprehensive survey, Comput. Methods Appl. Mech. Engrg. 196 (33–34) (2007) 3190–3218.
6. M. Trosset, Taguchi and robust optimization, Tech. Rep. 96-31, Department of Computational & Applied Mathematics, Rice University (1996)
7. Ramakrishnan, B.; Rao, S.S. 1991: A robust optimizationapproach using Taguchi’s loss function for solving nonlinear optimization problems. Advances in Design Automation 32,241–248
8. Welch, W.J.; Yu, T.K.; Kang, S.M.; Wu, J. 1990: Computer experiments for quality control by parameter design. Quality Technology 22, 15–22
9. Messac, A., and Yahaya, A. I., 2002, “Multiobjective Robust Design Using Physical Programming,” Struct. Multidiscip. Optim., 235, pp. 357–371.
10. Bertsimas, D.,D A. Thiele. 2006. A robust optimization approach to inventory theory. Oper. Res. 54 150-168.
11. J. Löfberg. YALMIP : A Toolbox for Modeling and Optimization in MATLAB. In Proceedings of the CACSD Conference, Taipei, Taiwan, 2004.
12. Zhou, J., Cheng, S., Li, M. Sequential Quadratic Programming for Robust Optimization With Interval Uncertainty. *J. Mech Design.* 134 (2012)
13. Gunawan. S. Parameter sensitivity measures for single objective, multi-

objective and feasibility robust design optimization. PhD dissertation. 2004

14. Mulvey, J. Vanderbei, R. and Zenios, S. Robust Optimization of Large-Scale Systems. *Operations Research* 43 (1995)

15. Pflugfelder, D. Wilkens, J.J. and Oelfke, U. Worst case optimization: a method to account for uncertainties in the optimization of intensity modulated proton therapy. *Phys. Med. Biol.* 53 (2008)

16. 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