A GENETIC ALGORITHMS FRAMEWORK FOR GREY NON-LINEAR PROGRAMMING PROBLEMS

Weihua Jin
Faculty of Engineering,
University of Regina
Regina, Saskatchewan Canada
E-mail: weiji111@uregina.ca

Christine W. Chan
Faculty of Engineering,
University of Regina
Regina, Saskatchewan Canada
E-mail: Christine.Chan@uregina.ca

Paitoon Tontiwachwuthikul
Faculty of Engineering,
University of Regina
Regina, Saskatchewan Canada

Gordon H. Huang
Faculty of Engineering,
University of Regina
Regina, Saskatchewan Canada

Abstract

This paper discusses the solution of a particular case of grey nonlinear programming, the Grey Quadratic Programming (GQP), and introduces the Genetic Algorithms (GA) approach as a feasible method for solving GQP problems. A framework using Genetic Algorithm for Grey Quadratic Programming (GAGQP) framework is designed and constructed by generalizing the common components of the GQP solutions and encapsulating the basic GA operations. This framework has been applied on a hypothetical municipal solid waste management problem and the result of the case study indicated that the GA approach is competitive with, if not superior to, other methods in solving GQP problems.

Keywords: *Grey numbers, grey quadratic programming, genetic algorithms.*

1. Introduction

Nonlinear programming is considered a powerful optimization tool capable of modeling and solving complex optimization problems in engineering. To handle uncertainty in real world data, inexact parameters and constraints are introduced and combined with various kinds of optimization techniques. Huang et al. (1994, 1995) proposed two inexact nonlinear programming methods by introducing internal and fuzzy numbers into the Quadratic Programming (QP) frameworks. The methods are Grey Quadratic Programming (GQP) and Grey-Fuzzy Quadratic Programming, both of which are applied to the planning of solid waste management systems. Detailed solution of GQP involves large numbers of direct comparisons to interactively identify the uncertain relationships between the objective function and decision variables. When applying these kinds of methods on medium sized or larger scale and complicated problems, the number of direct comparisons can become exponential.

This paper proposes feasibility of using Genetic Algorithms (GA) for solving typical GQP problems and further provides a detailed framework which renders the problem solving process easier and more error-free.

The GA approach can simulate the natural adaptation process and have been widely used in optimization problems, especially those which involve non-smooth and multimodal search spaces. For a GQP problem in an engineering domain, normally the search space is large and unsmooth. Furthermore, for many engineering problems, the global optimized solution is not required. Thus, the GA approach is suitable for GQP problems, and in fact, it has a better chance to be an effective approach compared to some traditional methods.

2. Commonly Used Modeling Process of GQP

A typical GQP problem can be expressed as follows:

$$\begin{aligned} \max \quad & f^{\pm} = \sum_{j=1}^{n}\left[c_j^{\pm}x_j^{\pm} + d_j^{\pm}\left(x_j^{\pm}\right)^2\right], \\ s.t. \quad & \sum_{j=1}^{n} a_{ij}^{\pm}x_j^{\pm} \le b_i^{\pm}, \quad i = 1,2,\ldots,m, \\ & x_j^{\pm} \ge 0, \quad j = 1,2,\ldots n, \end{aligned} \qquad [1]$$

where $a_{ij}^{\pm}, b_i^{\pm}, c_j^{\pm}, d_j^{\pm}$ are grey parameters and $x_j^{\pm}$ is a grey variable. It is assumed that an optimal solution exists. For a grey number $g^{\pm} \in [g^-, g^+]$, g^- and g^+ are the lower and upper bound respectively.

Normally, a GQP problem can be solved by best-worst case analysis when the problem is simple. To meet the requirements of engineering computation, Huang et al. (1994) proposed an interactive binary algorithm. The detailed procedures are described as follows:

Step 1. Group symbols for grey coefficients $c_{ij}^{\pm}$, let former k_1 coefficients be positive and latter k_2 be negative, $k_1 + k_2 = n$.

Step 2. Define the lower and upper bounds of the objective

CCECE/CCGEI, Saskatoon, May 2005

function as f^- and f^+.

Step 3. Define absolute values and signs for coefficients of the constraints $a_{ij}^{\pm}$.

Step 4. Define the relationships between the decision variables $x_j^{\pm}$ and the absolute value of the coefficients of the constraints $\left|a_{ij}^{\pm}\right|$.

Step 5. Formulate constraints corresponding to lower and upper bounds of objective function f^- and f^+.

Step 6. When the right-hand side of the constraints, b_i, are also grey numbers, define the relationships between $f^{\pm}$ and $b_i^{\pm}$.

Step 7. Specify the two sub-models.

Huang et al. (1995) further provided detailed explanations and examples to illustrate the modeling process. When this method is applied to a large scale optimization problem which comes with many coefficients of different signs, the modeling process becomes complicated and computationally expensive. Considering this method is based on an interactive process, the solution process can become less efficient and more error-prone when intensive interaction is necessary.

As an improvement of the algorithm presented in Huang et al, (1994), Chen et al. (2001) introduced a derivative algorithm (DAM) which simplifies the problem solving process by providing a quantitative expression for uncertain relationships between the objective function and the decision variables. The method from Chen et al. (2001) reduces the amount of computation necessary compared to the method proposed by Huang et al. (1995). This new modeling procedure can be represented briefly as follows:

Stage 1. Solve mid-value sub-model

Stage 2. Calculate $2d_j^+\left(x_j\right)_{mv\ opt}+c_j^+$ values

Stage 3. Identify optimal bound distribution for $x_j^{\pm}$

Stage 4. Formulate two sub-models corresponding to f^- and f^+

As described above, DAM was derived from the interactive binary algorithm and reduced the degree of complexity in the modeling process; it also decreases the computational complexity. However, DAM still involves formulating sub-models and depends on subjective judgments on how to combine different signed coefficients. Thus DAM cannot be automated and directly generated.

3. GA for problem solving of GQP Problems

As heuristic search algorithms, GA is adopted for solving GQP problems. In the GA approach, the whitened grey numbers of coefficients $a_{ij}^{\pm}, b_i^{\pm}, c_j^{\pm}, d_j^{\pm}$ can be determined by substituting the initial suboptimal decision variables back into the objective function. f^- and f^+ can be calculated directly without any uncertainty in the parameters.

The Genetic Algorithm for Grey Quadratic Programming (GAGQP) framework is constructed in Matlab® and is based on the GAOT (Genetic Algorithm Optimization Toolbox) from North Carolina State University (*http://www.ie.ncsu.edu/mirage/GAToolBox/gaot/*). The configuration tools which help users to customize the GAGQP according to different concrete problems are provided as part of GAGQP. The problem solving procedures implemented by the framework are grouped into three stages discussed as follows.

The objective of the first stage is to get an initial suboptimal x_j^s from the transformed problem from the GQP problem [1]:

$$\begin{aligned} \max \quad & f = \sum_{j=1}^{n}\left[c_j^r x_j + d_j^r\left(x_j\right)^2\right], \\ s.t. \quad & \sum_{j=1}^{n} a_{ij}^r x_j^r \le b_i^r, \quad i=1,2,\ldots,m, \\ & x_j \ge 0, \quad j=1,2,\ldots n, \end{aligned} \qquad [2]$$

where $a_{ij}^r, b_i^r, c_j^r, d_j^r$ are random numbers which satisfy the continuous uniform distribution in the interval $[a_{ij}^-, a_{ij}^+]$, $[b_i^-, b_i^+]$, $[c_j^-, c_j^+]$ and $[d_j^-, d_j^+]$ respectively. Then, problem [2] is solved in GAGQP's GA Non-Linear Program (GANLP) solving engine which uses the objective function in [2] as the positive term of the fitness function and the constraints of [2] as negative punishment terms of the fitness function. Thus, a suboptimal solution f^s can be identified and the corresponding decision variables x_j^s are also obtained.

In the second stage, the grey coefficients $a_{ij}^{\pm}, b_i^{\pm}, c_j^{\pm}, d_j^{\pm}$ will be whitened. To whiten a grey number means to give a determined value to this number. Let the whitened coefficients corresponding to f^+ be $\overset{+}{a_{ij}^{\pm}}, \overset{+}{b_i^{\pm}}, \overset{+}{c_j^{\pm}}, \overset{+}{d_j^{\pm}}$; and those corresponding to f^- be $\overset{-}{a_{ij}^{\pm}}, \overset{-}{b_i^{\pm}}, \overset{-}{c_j^{\pm}}, \overset{-}{d_j^{\pm}}$. These two sets of coefficients can be identified using the following method.

Substituting x_j^s into formula [1] will convert [1] into formula [3]:

$$\begin{aligned} \max \quad & f^{\pm} = \sum_{j=1}^{n}\left[c_j^{\pm} x_j^s + d_j^{\pm}\left(x_j^s\right)^2\right], \\ s.t. \quad & \sum_{j=1}^{n} a_{ij}^{\pm} x_j^s \le b_i^{\pm}, \quad i=1,2,\ldots,m, \\ & x_j^s \ge 0, \quad j=1,2,\ldots n, \end{aligned} \qquad [3]$$

To identify the coefficients $a_{ij}^{\pm}, b_i^{\pm}, c_j^{\pm}, d_j^{\pm}$ corresponding to $f^{\pm}$, a set of optimization problems need to be constructed and solve. Since x_j^s are suboptimal variables which tend to make the objective function closer to f^+, consider $a_{ij}^{\pm}, b_i^{\pm}, c_j^{\pm}, d_j^{\pm}$ being variables, then the optimization problem [s1] can be constructed to find $\overset{+}{c_j^{\pm}}, \overset{+}{d_j^{\pm}}$.

$$
\begin{aligned}
\max \quad & f^{\pm} = \sum_{j=1}^{n} \left[c_j^{\pm} x_j^s + d_j^{\pm} \left(x_j^s \right)^2 \right], \\
s.t. \quad & \sum_{j=1}^{n} a_{ij}^{\pm} x_j^s \le b_i^{\pm}, \quad i = 1,2,\ldots,m,
\end{aligned}
\qquad [s1]
$$

The coefficients $\overset{+}{c_j^{\pm}}, \overset{+}{d_j^{\pm}}$ are considered corresponding to f^+. Then the optimization problem of seeking $\overset{+}{a_j^{\pm}}, \overset{+}{b_j^{\pm}}$ which makes the constraint as loose (easy to be satisfied) as possible can be constructed as follows.

$$
\begin{aligned}
\min \quad & \sum_{j=1}^{n} a_{ij}^{\pm} x_j^s - b_i^{\pm}, \\
s.t. \quad & \sum_{j=1}^{n} a_{ij}^{\pm} x_j^s \le b_i^{\pm}, \quad i = 1,2,\ldots,m,
\end{aligned}
\qquad [s2]
$$

Hence, the $\overset{+}{a_{ij}^{\pm}}, \overset{+}{b_i^{\pm}}, \overset{+}{c_j^{\pm}}, \overset{+}{d_j^{\pm}}$ values can be calculated. Then the values of $\overset{-}{a_{ij}^{\pm}}, \overset{-}{b_i^{\pm}}, \overset{-}{c_j^{\pm}}, \overset{-}{d_j^{\pm}}$ can be identified easily by selecting the other end of the interval of each grey coefficient.

In the third stage, problem [1] has been converted into the following two sub-problems:

For f^+,

$$
\begin{aligned}
\max \quad & f^{+} = \sum_{j=1}^{n} \left[\overset{+}{c_j^{\pm}} x_j^{\pm} + \overset{+}{d_j^{\pm}} \left(x_j^{\pm} \right)^2 \right], \\
s.t. \quad & \sum_{j=1}^{n} \overset{+}{a_{ij}^{\pm}} x_j^{\pm} \le \overset{+}{b_i^{\pm}}, \quad i = 1,2,\ldots,m, \\
& x_j^{\pm} \ge 0, \quad j = 1,2,\ldots n,
\end{aligned}
\qquad [s31]
$$

For f^-,

$$
\begin{aligned}
\max \quad & f^{-} = \sum_{j=1}^{n} \left[\overset{-}{c_j^{\pm}} x_j^{\pm} + \overset{-}{d_j^{\pm}} \left(x_j^{\pm} \right)^2 \right], \\
s.t. \quad & \sum_{j=1}^{n} \overset{-}{a_{ij}^{\pm}} x_j^{\pm} \le \overset{-}{b_i^{\pm}}, \quad i = 1,2,\ldots,m, \\
& x_j^{\pm} \ge 0, \quad j = 1,2,\ldots n,
\end{aligned}
\qquad [s32]
$$

This step eliminates the grey parameters in formula [1] and generates [s31] and [s32] as typical Non-Linear Programming (NLP) problems which can be solved by the GANLP solving engine in GAGQP.

To illustrate the above algorithm more clearly, it is applied to GQP problem from Chen et al. (1999). This GQP problem can be expressed as follows,

$$
\begin{aligned}
\max \quad & f^{\pm} = [16,18] x_1^{\pm} - [12,14] \left(x_1^{\pm} \right)^2 - [4,5] x_2^{\pm} + [14,15] \left(x_2^{\pm} \right)^2, \\
s.t. \quad & [4.5,5.5] x_1^{\pm} + [1.8,2.2] x_2^{\pm} \le [1.8,2.1], \\
& x_1^{\pm} + [1.8,2.2] x_2^{\pm} \le [0.9,1.1], \\
& x_1^{\pm} \ge 0, \quad x_2^{\pm} \ge 0.
\end{aligned}
\qquad [e1]
$$

In stage one, suboptimal variables x_1^s, x_2^s can be calculated by the Genetic Algorithm for Non-Linear Programming (GANLP) engine:

$$x_1^s = 0.26497, \quad x_2^s = 0.37772, \quad f^s = 4.397.$$

In stage two, x_1^s, x_2^s were used to construct the optimization problems (s1) and (s2) in order to determine the coefficients $\overset{+}{a_{ij}^{\pm}}, \overset{+}{b_i^{\pm}}, \overset{+}{c_j^{\pm}}, \overset{+}{d_j^{\pm}}$ and $\overset{-}{a_{ij}^{\pm}}, \overset{-}{b_i^{\pm}}, \overset{-}{c_j^{\pm}}, \overset{-}{d_j^{\pm}}$. Solving [s1], $\overset{+}{c_j^{\pm}}, \overset{+}{d_j^{\pm}}$ were identified to be 18, 12, 4, and 15. The solution of problem [s2] gives the values of $\overset{+}{a_j^{\pm}}, \overset{+}{b_j^{\pm}}$ as 4.5, 1.8, 2.1, 1.8, and 1.1. In stage three, problems [s31] and [s32] were generated as follows:

$$
\begin{aligned}
\max \quad & f^{+} = 18 x_1^{\pm} - 12 \left(x_1^{\pm} \right)^2 - 4 x_2^{\pm} + 15 \left(x_2^{\pm} \right)^2, \\
s.t. \quad & 4.5 x_1^{\pm} + 1.8 x_2^{\pm} \le 2.1, \\
& x_1^{\pm} + 1.8 x_2^{\pm} \le 1.1, \\
& x_1^{\pm} \ge 0, \quad x_2^{\pm} \ge 0.
\end{aligned}
$$

$$
\begin{aligned}
\max \quad & f^{-} = 16 x_1^{\pm} - 14 \left(x_1^{\pm} \right)^2 - 5 x_2^{\pm} + 14 \left(x_2^{\pm} \right)^2, \\
s.t. \quad & 5.5 x_1^{\pm} + 2.2 x_2^{\pm} \le 1.8, \\
& x_1^{\pm} + 2.2 x_2^{\pm} \le 0.9, \\
& x_1^{\pm} \ge 0, \quad x_2^{\pm} \ge 0.
\end{aligned}
$$

Solving the above two problems, the solution of example [e1] is: $f^{\pm} = [2.6371, 5.4234]$, $x_1^{\pm} = [0.2441, 0.2857]$, and $x_2^{\pm} = [0.20792, 0.45239]$. As a comparison, the solution given by Chen et al. (1999) is: $f^{\pm} = [2.59, 5.42]$, $x_1^{\pm} = [0.238, 0.286]$, and $x_2^{\pm} = [0.224, 0.452]$.

By properly configuring the GANLP engine in stage three, such as increasing the maximum genetics generations number, the solution can enhanced. However, the above given solution is good enough for engineering problems.

4. Case Study

In this case study, the problem provided by Huang et al. (1995) was adopted and recalculated using the GAGQP framework. In this case study, the system contains three cities. The planning horizon is 15 years and is divided equally into three periods. A landfill and an incinerator are available to serve the needs of municipal solid waste disposal. The landfill has an existing capacity of $[2.05, 2.30] \times 10^6 t$, and the incinerator has a capacity of $[500, 600]\, t/d$. The incinerator generates residues of approximately 30% of the incoming waste streams, and its revenue from energy sale is $[15, 25] per tone combusted. According to Huang et al. (1995) this GQP model could be formulated as follows:

$$\min \quad f^{\pm} = \sum_{i=1}^{2} \sum_{j=1}^{3} {}^{+} \sum_{k=1}^{3} L_k^{\pm} \left(TR_{ijk}^{\pm} + OP_{ik}^{\pm} \right) x_{ijk}^{\pm},$$

$$+ \sum_{j=1}^{3} \sum_{k=1}^{3} L_k^{\pm} \left[FE \left(FT_{jk}^{\pm} + OP_{jk}^{\pm} \right) - RE_k^{\pm} \right] x_{2jk}^{\pm}$$

$$s.t. \quad \sum_{j=1}^{3} \sum_{k=1}^{3} L_k^{\pm} \left(x_{1jk}^{\pm} + FEx_{2jk}^{\pm} \right) \le TL^{\pm},$$

$$\sum_{j=1}^{3} x_{2jk}^{\pm} \le TE^{\pm} \quad \forall k,$$

$$\sum_{i=1}^{2} x_{ijk}^{\pm} \ge WG_{jk}^{\pm} \quad \forall j, k,$$

$$x_{ijk}^{\pm} \ge 0 \quad \forall i, j, k,$$

where

FE Residue flow rate from incinerator to landfill

$FT_{jk}^{\pm}$ Transportation cost for residue from incinerator to landfill during period k ($/t)

$OP_{ik}^{\pm}$ Operating cost of facility i during period k ($/t)

$RE_k^{\pm}$ Revenue from incinerator during period k ($/t)

$TE^{\pm}$ Capacity of incinerator (t/d)

$TL^{\pm}$ Capacity of landfill (t)

$TR_{ijk}^{\pm}$ Transportation cost for waste from city j to facility i during period k ($/t)

$WG_{jk}^{\pm}$ Waste generation rate in city j during period k (t/d)

Detailed data was given in Huang et al. (1995). The GAGQP framework generated the solution as follows:

$$f^{\pm} = [2.405e+08, 5.133e+08].$$

This solution is close to the result in Huang et al. (1995), which is, for comparison purposes, $f^{\pm} = [2.39e+08, 5.14e+08]$. This case study indicates that the GAGQP framework can be configured to deal with complex engineering problems and generate good solutions. If the parameters of the genetic algorithms of the GANLP engine are further tuned, a better solution can be generated.

5. Conclusions

This paper discusses a particular kind of grey nonlinear problems, called the grey quadratic programming problems. Commonly used modeling methods of GQP are reviewed and their shortcomings explained. Compared with traditional methods, the GAGQP framework can avoid the interactive process of problem solving. Instead, the whole process can be rendered automatic and stable by direct calculation of the coefficients $a_{ij}^{\pm}, b_i^{\pm}, c_j^{\pm}, d_j^{\pm}$, thereby reducing the complexity and the chance for errors. Since the algorithm is concise and programmable, it can be built into a software package. The GAGQP framework is currently implemented on Matlab® and is based on the GA toolbox GAOT. It can be further customized according to user requirements for different concrete GQP problems. This framework has been applied on a hypothetical municipal solid waste management problem and the results generated are satisfactory. Considering the prevalence of GQP problems in environmental engineering and operation research, this framework can be extended and made into a general purposed tool.

Acknowledgements

The generous support of a Research grant from the Natural Sciences and Engineering Research Council is gratefully acknowledgement.

References

[1] G. H. Huang, B. W. Baetz, G. G. Party, "Waste flow allocation planning through a grey fuzzy quadratic programming approach," *Civil Engineering System*, 11, pp. 209-243, 1994.

[2] G. H. Huang, B. W. Baetz, G. G. Party, "Grey quadratic programming and its application to municipal waste management planning under uncertainty," *Engineering Optimization*, 23, pp.201-223, 1995.

[3] G. H. Huang, "IPWA: An interval parameter water quality management model," *Engineering Optimization*, 26, pp.79-103, 1996.

[4] GAOT: Genetic Algorithm Optimization Toolbox http://www.ie.ncsu.edu/mirage/GAToolBox/gaot/

[5] J. Tang, D. Wang, "An interactive approach based on a genetics algorithm for a type of quadratic programming problems with fuzzy objective and resources," *Computers and Operations Research*, 24, pp.413-429, 1997.

[6] M. J. Chen, G. H. Huang, "A derivative algorithm for inexact quadratic program – application to environmental decision-making under uncertainty," *European Journal of Operational Research*, 128, pp. 570-586, 2001.

[7] W. L. Winston, *OPERATIONS RESEARCH: APPLICATIONS AND ALGORITHMS, THIRD EDITION*. Wadsworth Publishing Company; Belmont, California, January 13, 1997.