

# An Optimization Approach for Nonlinear Programming Problems with Interval Coefficients

Mei Ma and Hecheng Li

Department of Mathematics, Qinghai Normal University, Xining 810008, China

1450670013@qq.com

**Keywords:** Nonlinear programming problem, interval coefficients, bilevel program.

**Abstract.** When some coefficients involved in nonlinear programming problems are taken as interval numbers, the problem becomes more challengeable than the original one. In the paper, a nonlinear programming problem with interval objective coefficients is considered, and a genetic algorithm is presented. Firstly, the original problem is transformed into a bilevel programming model without interval coefficients. In addition, a genetic algorithm is designed and used to deal with the transformed problem, in which the search space is decided by these intervals given in the original problem. Based on the proposed approach, the best and the worst optimal solutions as well as some intermediate solutions can be obtained. Finally, a computational example is solved and the results show that the proposed algorithm is efficient and robust.

## 1. Introduction

The methodology for solving optimization problems has widely applied to many research fields. If the coefficients of optimization problem are taken as closed intervals, they will be categorized as interval-valued optimization problems. In recent years, the interval analysis method was developed to model the uncertainty in uncertain optimization problems, in which the bounds of the uncertain coefficients are only required.

The linear programming problem with interval coefficients in objective function was discussed by Tanaka et al and Rommelfanger<sup>[1,2]</sup>. Using an order relation of the intervals, a linear interval programming problem was converted to a deterministic problem<sup>[3,4]</sup>. Levin<sup>[6]</sup> justified a more general approach to the optimization of systems with interval parameters. Lai et al<sup>[7]</sup> discussed a class of linear programming problems with interval coefficients in both the objective functions and constraints.

But for most of the engineering problems, the objective function and constraints are nonlinear, and they are always obtained through numerical algorithms. The reference [8] seems the first publication on nonlinear interval number programming (NINP). However, only the uncertain objective function is considered, and no approach is proposed to deal with the nonlinear constraints with uncertainty. So the reference [5] is suggested to solve the nonlinear interval number programming problem with uncertain coefficients both in nonlinear objective function and nonlinear constraints.

Multiobjective linear programming (MOLP) under uncertainty has gained great interest in the past decades. A number of Multiobjective linear programming (MOLP) under uncertainty methods and their improvements have been proposed. An interval-parameter fuzzy linear programming method (IFMOLP) has been proposed.<sup>[9]</sup> Carla and Carlos<sup>[10]</sup> provided an illustrated overview of the state of the art of Interval Programming in the context of multiple objective linear programming models.

This paper is organized as follows. The discussed problem and some notations are presented in Section 2, and a bilevel programming model is investigated in Section 3. A genetic algorithm is given based on the transform procedure in Section 4, and this is followed by the simulation as Section 5. We finally conclude our paper in Section 6.

## 2. Discussed Problem and Some Definitions

The nonlinear programming problems with interval coefficients can be formulated as:

$$\begin{cases} \min_{x \in R^n} f(U, x) \\ \text{s.t. } G(U, x) \leq 0. \end{cases} \quad (1)$$

Here,  $x$  is the decision-making vector,  $U$  is the interval coefficient vector and  $G(x)$  is the constraint function vector.

Since  $U$  is the interval vector, Eq. (1) is uncertain. For any  $u \in U$  fixed, Eq. (1) can be taken as an exact nonlinear programming problem, and by solving the problem, an optimal solution can be obtained. When all  $u$  in  $U$  is taken into account, a set of the optimal values of Eq. (1) are gotten. As two extreme values, the upper and lower bounds of the set are important for the decision maker to understand the risk involved better. Some definitions are presented as follows:

**Definition 1.** (Optimal solution) For some  $u \in U$  fixed, if  $x$  solves Eq. (1), then  $x$  is called an optimal solution of Eq. (1).

**Definition 2.** (Best optimal solution) For any  $u \in U$  with the optimal solution  $x$ , if there exists a  $u^b \in U$ , the corresponding optimal solution  $x^b$  satisfies:

$$f(u^b, x^b) \leq f(u, x), \quad \forall u \in U. \quad (3)$$

then  $x^b$  is called the best optimal solution of Eq. (1).

**Definition 3.** (Worst optimal solution) For any  $u \in U$  with the optimal solution  $x$ , if there exists a  $u^w \in U$ , the corresponding optimal solution  $x^w$  satisfies:

$$f(u^w, x^w) \geq f(u, x), \quad \forall u \in U. \quad (4)$$

then  $x^w$  is called the worst optimal solution of Eq. (1).

The set of all optimal solutions is denoted by  $\Omega$ , that is,

$$\Omega = \{x \mid \exists u \in U, \text{ such that } x \text{ is optimal}\}. \quad (5)$$

$$\text{Also, the set of all optimal values } F = \{f(u, x) \mid x \in \Omega\} \quad (6)$$

In order to avoid too much theoretic analysis, we assume that for any  $u \in U$ , there exists at least one optimal solution. The upper bound is the best optimal value of Eq. (1), and the lower bound is the worst optimal value of Eq. (1).

### 3. Bilevel Optimization Model

In order to solve Eq. (1), we solve the two extreme solutions in two steps. For the best optimal solution and the worst optimal solution, we solve the Eqs. (7) and (8) separately as follows:

$$\begin{cases} \min_{u \in U} f(U, x) \\ \min_{x \in R^n} f(U, x) \\ \text{s.t. } G(U, x) \leq 0. \end{cases} \quad (7) \quad \text{and} \quad \begin{cases} \max_{u \in U} f(U, x) \\ \min_{x \in R^n} f(U, x) \\ \text{s.t. } G(U, x) \leq 0. \end{cases} \quad (8)$$

In Eqs. (7) and (8),  $U$  can be seen as the leader's vector, and  $x$  is the follower's vector. One can obtain the best and worst solutions by solving these bilevel programming problems.

If Eqs. (7) and (8) are separately solved, the computation is expensive. In the next section, an efficient genetic algorithm is designed, which can simultaneously deal with these two problems and provides some other solutions between the best and the worst ones according to the requirement of the decision maker.

#### 4. Proposed Algorithm

In the section, we present a genetic algorithm using a special archive technique, which can provide the best and the worst solutions as well as some expected intermediate solutions.

**Chromosome Encoding.** We take vectors in  $U$  as individuals, and the real coding scheme is adopted. Set the number of the intervals in the objective function be  $q$ , then each individual is a real vector with dimensions  $q$ . One can denote these individuals by  $u_1, u_2, \dots, u_i, \dots$ .

**Fitness Evaluation.** In order to obtain the best and the worst optimal solution in one run of the algorithm, we design a special evaluation scheme. Firstly, for each individual, the follower's problem of Eqs. (7) and (8) are solved to obtain the optimal solution  $x$ . The objective is taken as evaluation criterion. In addition, the spread of objective values is also taken as the other criterion to evaluate individuals.

**Selection.** According the objective value, the best and the worst solutions in the present population and offspring set are directly put into the next population. Meanwhile, some equal diversion points are taken between the two points, and the nearest individuals or offspring to these equal diversion points are put into the next population. If the individual number in the next population is less than the population size, other individuals are randomly chosen from the rest of the present population and its offspring.

**Crossover Operator.** Set  $u^b$  be the best individual found so far with regarding to the best optimal solution, and  $u^w$  be the best individual found so far with regarding to the worst optimal solution. Let  $u$  be a crossover parent individual. For  $\alpha_i$  taken randomly in  $[0,2]$ ,  $i = 1,2$ , crossover offspring can be generated as follows:

$$o_1 = u + \alpha_1(u^b - u) \quad (9)$$

$$o_2 = u + \alpha_2(u^w - u) \quad (10)$$

**Mutation Operator.** Gaussian mutation is adopted.

**Proposed Algorithm.** In the subsection, we propose a genetic algorithm using a special archive technique (GA-SAT).

Step1. (Initial population)  $N$  initial points are generated randomly in search space  $U$  to form the initial population  $\text{pop}(0)$ . Let  $g = 0$ ,  $\text{Arc} = \phi$  and  $N_a = r$ ;

Step2. (Fitness) Evaluate each point in  $\text{pop}(g)$  by solving objective values, and record the best solution  $u^b$  and  $u^w$ , and all individuals are put into Arc;

Step3. (Crossover and mutation) The designed crossover operator and the Gaussian mutation are adopted to generate all genetic offspring. The set of all offspring is denoted by  $O$ ;

Step4. (Offspring evaluation) Evaluate the fitness values of all offspring, and update  $u^b$  and  $u^w$ . all offspring individuals are put into Arc. If the element number of Arc is larger than  $r$ , then the crowding scheme is used to delete redundant elements such that the size of Arc is equal to  $r$ ;

Step5. (Next generation of population) The selection scheme is applied to generate the next generation of population  $\text{pop}(g+1)$ ;

Step6. (Stopping criterion) If the termination condition is satisfied, then the algorithm is stopped and output Arc ; otherwise, let  $g=g+1$ , go to Step 3.

In Step4, when the crowding scheme is used, the best and the worst solutions found so far are always kept in Arc.

#### 5. Simulation

We select the following example to illustrate the efficiency of the proposed algorithm.

$$\begin{cases} \min_u f(U, x) = U_3x_1 + 7U_2x_2 - U_1x_1x_2 + 100, \\ \text{s.t. } U_2U_3x_2 + U_1x_1x_2 \leq U_4, \\ U_1U_2(x_1 - 15)^2 + U_3(x_2 - 20)^2 \leq U_5, \\ x_i \in [0, 20], i = 1, 2. \end{cases}$$

Here,  $U_1 = [1.5, 2.5], U_2 = [5.5, 6.5], U_3 = [3.5, 4.5], U_4 = [790, 810], U_5 = [690, 710]$ .

Genetic parameters are taken as: the population size pop\_size=50, the crossover probability pc=0.8, the mutation probability pm=0.01, and the maximum number of generations Max G=200. The proposed GA-SAT is executed 10 independent runs. The best optimal solution is denoted by Best\_opt., whereas the worst one is denoted by Worst\_opt. In 10 runs, the best ones are represented as F\_b, and F\_w means the worst ones. Also, 15 intermediate objectives are provided, which are between the best and the worst optimal objectives.

		Best_opt.	Worst_opt.
F	F_b	41.20	327.85
	F_w	41.21	327.85
	mean	41.20	327.85
U		(2.49,5.53,3.51,800.94,690.81 )	(1.52,6.47,4.45,799.82,702.46)
x_b		(20.00,11.55)	(17.12,7.84)
Intermediate points		41.2,64.9,85.6,106.3,127.1,147.8,168.5,189.2,209.9,230.7, 251.4,272.1,292.8,313.6,327.9	

## 6. Conclusion

Based on the proposed approach, this paper solved a nonlinear programming problem with interval objective coefficients. We transformed the original problem into a bilevel programming model without interval coefficients, and presented a genetic algorithm which was used to obtain the best and the worst optimal solutions as well as some other intermediate solutions. For the best optimal solution and the worst optimal solution, we solved the bilevel programming problem in two steps. From the computational example and the results, we can see the proposed algorithm is efficient and robust.

## Acknowledgements

This work was financially supported by the National Natural Science Foundation of China (61463045) and the Natural Science Foundation of Qinghai Provincial (2013-Z-937Q).

## References

- [1] H.Tanaka, T. Ukuda , K. Asal: Journal of Cybernetics Vol. 3 (1984), p. 37
- [2] H.Rommelfanger: Fuzzy Sets and Systems Vol. 29 (1989), p. 31
- [3] S.Chanas &D. Kuchta: Fuzzy Sets and Systems Vol. 82 (1996), p. 299
- [4] S. Chanas & D. Kuchta: European Journal of Operational Research Vol. 94 (1996),p. 594
- [5] C.Jiang, X. Han & G. R. Liu: European Journal of Operational Research, Vol. 188(2007), p. 113
- [6] V. I. Levin: Cybernetics and Systems Analysis, Vol. 40 (2004), No. 3
- [7] K. K. Lai, S. Y.Wang, J. P. Xu, S. S.Zhu &Y. Fang: IEEE Transactions on Fuzzy Systems Vol. 10 (2002), No. 6
- [8] L. H. Ma. Research on method and application of robust optimization for uncertain system. (Ph.D. thesis, Zhejiang University, China 2002)

[9] Wu • Guo Simon, H. Huang: J Math Model Algor Vol. 6 (2007), p. 195

[10] Carla Oliveira , Carlos Henggeler Antunes: European Journal of Operational Research Vol. 181 (2007), p. 1434