1996年度日本オペレーションズ・リサーチ学会
春季研究発表会

遺伝的アルゴリズムによる区間計画

01103510 足利工業大学 *玄　光男　GEN Mitsuo
01006330 足利工業大学　井田憲一　IDA Kenichi
宇都宮大学　程　潤偉　CHENG Runwei

1 Introduction

In this paper, we investigate how to solve interval programming problem with genetic algorithms. The basic idea of the proposed approach is firstly to transform interval programming model into an equivalent bicriteria programming model and then to find out the Pareto solutions of the bicriteria programming problem using genetic algorithms. Numerical example is given to demonstrate the efficiency of the proposed approach.

2 Interval Programming

An interval is defined as an ordered pair of real numbers as follows:

$$\begin{aligned} A &= [a^L, a^R] \\ &= \{x \mid a^L \le x \le a^R; x \in R\} \quad (1) \\ &= \langle a^C, a^W \rangle \\ &= \{x \mid a^C - a^W \le x \le a^C + a^W; x \in R\} \quad (2) \end{aligned}$$

where a^L, a^R, a^C, and a^W are the left bound bound, right bound, center, and width of interval A, respectively. The center and width are calculated as follows:

$$a^C = \frac{1}{2}(a^R + a^L), \quad a^W = \frac{1}{2}(a^R - a^L) \quad (3)$$

There are two key steps when transforming interval programming to bicriteria programming: (1) Using the definition of the degree of inequality-holding-true for two intervals, transform interval constraints into equivalent crisp constraints, (2) Using the definition of the order relation between intervals, transform interval objective into equivalent crisp two objectives. Let us examine the following problem:

$$\max \quad Z(\boldsymbol{x}) = \sum_{j=1}^{n} C_j x_j \quad (4)$$

$$\text{s. t.} \quad G_i(\boldsymbol{x}) = \sum_{j=1}^{n} A_{ij} x_j \le B_i, \; i = 1, 2, \ldots, m \quad (5)$$

$$x_j^L \le \; x_j \le x_j^U : \text{integer}, \; j = 1, 2, \ldots, n \quad (6)$$

where $C = [c^L, c^R]$, $A_{ij} = [a_{ij}^L, a_{ij}^R]$, and $B = [b^L, b^R]$, respectively, and x_j^L and x_j^U are the lower and upper bounds for x_j, respectively. Problem (4)-(6) can be transformed into the following problem:

$$\max \quad z^L(\boldsymbol{x}) = \sum_{j=1}^{n} c_j^L x_j \quad (7)$$

$$\max \quad z^C(\boldsymbol{x}) = \sum_{j=1}^{n} \frac{1}{2}(c_j^L + c_j^R) x_j \quad (8)$$

$$\text{s. t.} \quad g_i(\boldsymbol{x}) = \sum_{j=1}^{n} a_{ij} x_j \le b_i, \; i = 1, 2, \ldots, m \quad (9)$$

$$x_j^L \le \; x_j \le x_j^U : \text{integer}, \; j = 1, 2, \ldots, n \quad (10)$$

where $a_{ij} = q a_{ij}^R + (1-q) a_{ij}^L$ and $b_i = (1-q) b_i^R + q b_i^L$.

3 Genetic Algorithm

Now we discuss how to solve the problem (7)-(10) with GA.

Chromosome representation and initial population: A chromosome is defined as follows:

$$\boldsymbol{x}^k \;=\; [\; x_1^k \; x_2^k \; \cdots \; x_n^k \;]$$

where subscript k is the index of chromosome. Initial population is randomly generated within the range $[x_j^L, x_j^U]$ for all x_j.

Crossover and mutation: Crossover is implemented with uniform crossover operator [3] and mutation is performed as random perturbation within the permissive range of integer variables.

Selection: Deterministic selection is used, that is, delete all duplicate among parents and offspring, and then sort them in descending order and select the first *pop_size* chromosomes as the new population.

Evaluation: There are two main tasks involved in this phase: (1) how to handle infeasible chromosomes and (2) how to determine fitness value of chromosomes according to bicriteria. Let $\boldsymbol{x}^k$ be the k-th chromosome in current generation and E the Pareto solutions set found so far. The fitness function is given as follows:

$$eval(\boldsymbol{x}^k) = (w_1 z^L(\boldsymbol{x}^k) + w_2 z^C(\boldsymbol{x}^k))p(\boldsymbol{x}^k)$$

where $w_1 = z^C_{\max} - z^C_{\min}, w_2 = z^L_{\max} - z^L_{\min}$,

$$\begin{aligned} z^C_{\min} &= \min\{z^C(\boldsymbol{x}^k) \mid \boldsymbol{x}^k \in E\}, \\ z^C_{\max} &= \max\{z^C(\boldsymbol{x}^k) \mid \boldsymbol{x}^k \in E\}, \\ z^L_{\min} &= \min\{z^L(\boldsymbol{x}^k) \mid \boldsymbol{x}^k \in E\}, \\ z^L_{\max} &= \max\{z^L(\boldsymbol{x}^k) \mid \boldsymbol{x}^k \in E\} \end{aligned}$$

and the penalty term for the k-th chromosome is defined as follows:

$$p(\boldsymbol{x}^k) = 1 - \frac{1}{m}\sum_{j=1}^{m}\left(\frac{\Delta b_j(\boldsymbol{x}^k)}{\Delta b_j^{\max}}\right)$$

where

$$\begin{aligned} \Delta b_j(\boldsymbol{x}^k) &= \max\{0, g_j(\boldsymbol{x}^k) - b_j\}, \\ \Delta b_j^{\max} &= \max\{\epsilon, \Delta b_j(\boldsymbol{x}^k),\ k = 1, \ldots, pop\_size\} \end{aligned}$$

Along with evolutionary process, Pareto set E is updated, the two special points may be renewed and the line will move gradually in the direction from negative ideal point to positive ideal point. It means that the fitness function gives the selection pressure to force genetic search towards to exploiting the nondominated points in the criteria space.

4 Example

Let us consider following example with the interval objective function:

$$\begin{aligned} \max\quad & Z(\boldsymbol{x}) = [15,17]x_1 + [15,20]x_2 + [10,30]x_3 \\ \text{s. t.}\quad & g_1(\boldsymbol{x}) = x_1 + x_2 + x_3 \le 30 && (11) \\ & g_2(\boldsymbol{x}) = x_1 + 2x_2 + x_3 \le 40 && (12) \\ & g_3(\boldsymbol{x}) = x_2 + 4x_3 \le 60 && (13) \\ & x_j \ge 0 : \text{ interger}, j = 1,2,3 && (14) \end{aligned}$$

This problem can be transformed into the following bicriteria problem:

$$\begin{aligned} \max\quad & z^L(\boldsymbol{x}) = 15x_1 + 15x_2 + 10x_3 \\ \max\quad & z^C(\boldsymbol{x}) = 16x_1 + 17.5x_2 + 20x_3 \\ \text{s. t.}\quad & \text{constraints (11)–(14)} \end{aligned}$$

In this problem, total 13 Pareto solutions were found by our GA. The corresponding interval objectives are given in Figure 1. From the figure we can know that these intervals can not be compared with each other under the order relation.

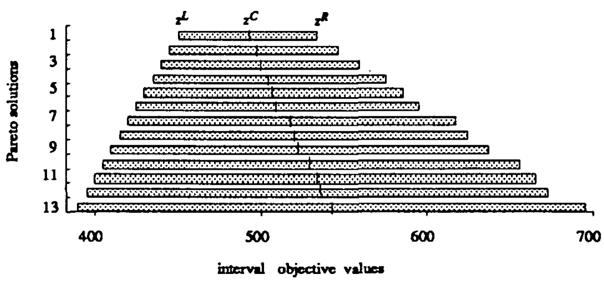

Figure 1: Interval objectives for Pareto solutions

5 Conclusion

In this paper, we have investigated how to solve interval programming with genetic algorithms. While the results are preliminary, numerical example demonstrated that rich Pareto solutions can be found by the proposed approach. The proposed GA can also be directly applied to other bicriteria programming problems.

References

[1] Hansen, E., *Global Optimization Using Interval Analysis*, Marcel dekker Inc., New York, 1992.

[2] Ishibuchi, H. and H. Tanaka, Formulation and analysis of linear programming problem with interval coefficients, *Journal of Japan Industrial Management Association*, vol.40, no.5, pp.320–329, 1989, in Japanese.

[3] Syswerda, G., Uniform crossover in genetic algorithms, In Schaffer, J., editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pp. 2–9, Morgan Kaufmann Publishers, San Mateo, California, 1989.