IOSR Journal of Mathematics (IOSR-JM)
e-ISSN: 2278-5728, p-ISSN: 2319-765X. Volume 14, Issue 3 Ver. II (May - June 2018), PP 12-15
www.iosrjournals.org

A coadapted coevolutionary particle swarm optimization algorithm based on bilevel programming for inverse optimal problem

Tao Zhang[1,*]

[1] *School of Information and Mathematics, Yangtze University, Jingzhou 434023, China*

Abstract: *In this paper, the inverse optimal value problem is equivalently reformulated as a corresponding bilevel programming (BLP) problem. A coadapted coevolutionary particle swarm optimization (CCPSO) is proposed for solving the BLP, in which the evolutionary paradigm can efficiently prevent the premature convergence of the swarm.*

Keywords*: Inverse optimal problem, bilevel programming, particle swarm optimization*

Date of Submission: 07-05-2018 Date of acceptance: 22-05-2018

I. Introduction

Geophysical scientists firstly study inverse problems. It gives a comprehensive discussion of the theory of inverse problems in the geophysical sciences [1]. From then on, this challenge topic has attracted more and more attention, which leads to a rapid development in theories, algorithms and applications. For detailed expositions, the reader may consult [2-5].

A standard inverse optimization problem is as follows: given an optimization problem with a linear objective $P: \min\{c^T x \mid x \in X\}$ and a desired optimal solution $x^* \in X$, find a cost vector c^* such that $x^* \in X$ is an optimal solution of P, and at the same time c^* is required to satisfy some additional conditions. Such that, given a preferred cost $c^{'}$, the deviation $\left\|c^* - c^{'}\right\|_p$ is to be minimum under some l_p norm.

In this paper, the inverse optimal value problem is equivalently reformulated as a corresponding bilevel programming (BLP) problem. A coadapted coevolutionary particle swarm optimization (CCPSO) is proposed for solving the BLP, in which the evolutionary paradigm can efficiently prevent the premature convergence of the swarm.

II. The bi-level model for inverse problem

In this section, we firstly introduce the inverse optimal value problem, then we describe how to consider this problem using BLP method. Consider the optimal value function of a linear programming in terms of its cost vector

$$Q(c) = \min_x \left\{c^T x : Ax \le b, x \ge 0\right\} \tag{1}$$

where $x \in R^n$, $A \in R^{p\times n}$, $b \in R^p$. Given a set $c \subseteq R^n$ of the objective cost vectors and a real number z^*, then the inverse optimal value problem is to find a cost vector from the set C such that the optimal objective value of the linear programming (1) is “close” to z^*, thus the inverse optimal value problem can be written as [6].

$$\min_c \left\{f(c) : c \in C\right\}, \text{ where } f(c) = \left|Q(c) - z^*\right| \tag{2}$$

As $Q(c)$ is the optimal value of the linear programming, then problem (2) can be written as,

$$\min_c \left[\left|Q(c) - z^*\right| : c \in C, Q(c) = , \min_x \{c^T x \le b, x \ge 0\}\right].$$

Go one step further, it can be written a bilevel programming problem as follows:

$$\min_c \left| Q(c) - z^* \right| \qquad s.t. \quad c \in C, \qquad (3)$$
$$Q(c) = \min_x \left\{ c^T x : Ax \le b, x \ge 0 \right\}$$

In fact, model (3) is called the BLP problem with the optimal value of the lower level problem feeding back to the upper level [7]. From the above description, we can find that the inverse optimal value problem is equivalent to a special class of BLP problem. Then it provides us an alternative approach to consider the inverse optimal value problem.

III. The CCPSO for the inverse optimization problem

3.1 The CCPSO Algorithm

Now we discuss the uses of cooperative coevolution particle swarm optimization (CCPSO) to minimize a function $f(x)$ of n independent variables. The problem was decomposed into n species and each assigned to one of the independent variables. Each species consisted of a population of alternative values for its assigned variable. To evaluate an individual from one of the species, we first selected the current best individual from every one of the other species and combined them, along with the individual being evaluated, into a vector of variable values. This vector was then applied to the target function. An individual was rewarded based on how well it minimized the function within the context of the variable values selected from the other species. The details of the proposed algorithm are given as follows:

Algorithm 1

Step 1. Initialization scheme. Initialized the n species randomly and each species represents a variable of a complete solution. Then, go to step 2 (a).

Step 2. Generate the complete solution.

(a) Select the individual from each of the other species randomly and combined them, along with the individual being evaluated, into a vector. Then, go to step 3.

(b) Select the current best individual from every one of the other species and combined them, along with the individual being evaluated, into a vector. Then, go to step 3.

Step 3. Credit evaluation. The credit assignment at the species level is defined in terms of the fitness value of the complete solutions in which the species members participate.

Step 4. Species coevolutionary. Each of the species is coevolved in a round-robin fashion using a standard PSO.

Step 5. Evolutionary stagnation detection. If the evolutionary stagnation condition of a species is false, then, go to step 7. Otherwise, go to step 6.

Step 6. Reinitialization the stagnation species. Keep the member with the best credit assignment, the remaining members are reinitialized randomly and the credit evaluation of these members is computed as the step 3.

Step 7. Termination check. If termination condition is false, go to step 2(b). Otherwise, output the optimal solution.

3.2 The CCPSO Algorithm for inverse optimization problem based on BLP

The process of the proposed algorithm for solving the BLP is an interactive coevolutionary process. We first initialize population, and then the BLP is transformed to solve single level optimization problems in the upper level and the lower level interactively by the CCPSO. For each iteration, an approximate optimal solution for problem 1 is obtained and this interactive procedure is repeated until the accurate optimal solutions of the original problem are found. The details of the proposed algorithm are given as follows:

Algorithm 2

Step 1. *Initialization scheme*. Initialize a random population (N_u) of the upper level variables. For each upper level member, initialize a random population (N_l) of the lower level variables and perform a lower level optimization procedure to determine the corresponding optimal lower level variables using **Algorithm 1.**

Step 2. Combine the upper level variables and the corresponding optimal lower level variables to generate the complete upper level solution (Z^u). Evaluate the fitness value of the complete upper level solutions based on the upper level function and constraints.

Step 3. Fixed the lower level variables of the complete upper level solution, execute an upper level optimization procedure to update the upper level variables of the complete upper level solution using **Algorithm 1**.

Step 4. Reinitialize the lower level members. For the updated upper level variables, determine the individual closest to the member in the complete upper level solutions produced by the step 2, then, the corresponding lower level optimal variables is selected as a member of the lower level population. The remaining $N_l - 1$

members are generated randomly.

Step 5. Combine the updated upper level variables and the corresponding lower level variables to generate the complete lower level solution (Z^l). Then, evaluate the fitness value of the complete lower level solutions based on the lower level function and constraints.

Step 6. Fixed the upper level variables of the complete lower level solution, perform a lower level optimization procedure to produce the corresponding optimal lower level variables using **Algorithm 1**. Then, go to step 2.

At step 3 or step 6, the algorithm uses a variance based termination criteria at both levels. When the value of α_j, described in the following equation becomes less than α_{stop}, the optimization task terminates. In the following, we state the termination criteria at the lower level, which can be similarly extended to the upper level. Let the variance of the lower level population members at generation j for each lower level variable i be v_j^i. If the number of lower level variables is m_l, then α is computed as:

$$\alpha_j = \sum_{i=1}^{m_l} \frac{v_j^i}{v_0^i} \tag{4}$$

where v_0^i denotes the variance for the variable i in the initial lower level population, the value of α_j usually lies between 0 and 1 in (4). In this paper, the value of α_{stop} is set as 10^{-5} for the lower level and the value of α_{stop} is set as 10^{-4} for the upper level.

IV. Numerical Experiment

In this section, the parameters are set as follows: The PSO parameters are set as follows: $r_1, r_2 \in random(0,1),$ the inertia weight $w = 0.7298$ and acceleration coefficients with $c_1 = c_2 = 1.49618$. To illustrate the algorithm, we solve the following inverse optimal value problem.

Example Let $c \in R^2, x \in R^2$:

$$\min_{c} \left\{ \left| Q(c) - z^* \right| : 10 \le c_1^2 + c_2^2 \le 13, c_1 \ge 0, c_2 \ge 0 \right\}$$

where $z^* = 14$ and $Q(c) = \max_{x} \left\{ c_1x_1 + c_2x_2 : x_1 + 2x_2 \le 8, x_1 \le 3, x_2 \le 4, x \ge 0 \right\}$

Following problem (3), we can write the above inverse optimal value problem as the following nonlinear bilevel programming:

$$\begin{aligned} &\min_{c \ge 0} \ (c_1x_1 + c_2x_2 - 14)^2 \\ &\quad s.t.\ 10 \le c_1^2 + c_2^2 \le 13 \\ &\qquad Q(c) = \max_{x \ge 0} \left\{ c_1x_1 + c_2x_2 : x_1 + 2x_2 \le 8, x_1 \le 3, x_2 \le 4 \right\} \end{aligned} \tag{5}$$

For the CCPSO, the population sizes of each species were set as 10 at the upper level and lower level. All results presented in this paper have been obtained on a personal computer (CPU:AMD Phenon(tm) Ⅱ X6 1055T 2.80GHz; RAM:3.25GB) using a C# implementation of the proposed algorithm. We can get the optimal solutions of problem (5): $c^* = (2,3), x^* = (4,2)$. Through some simple validating calculations, it is shown that the algorithm proposed in this work is feasible for the inverse optimal value problem.

V. Conclusions

In this paper, the inverse optimal value problem is equivalently reformulated as a corresponding bilevel programming problem. A coadapted coevolutionary particle swarm optimization is proposed for solving the BLP, in which the evolutionary paradigm can efficiently prevent the premature convergence of the swarm. Through some simple validating calculations, it is shown that the algorithm proposed in this work is feasible for the inverse optimal value problem.

Reference

[1]. A. Tarantola, Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation, Elsevier, Amsterdam, The Netherlands, 1987.

[2]. R.K. Ahuja, J.B. Orlin, Inverse optimization, Oper. Res. 49 (2001) 771 – 783.

[3]. D. Burton, Ph.L. Toint, On an instance of the inverse shortest paths problem, Math. Programming 53 (1992) 45 – 61.
[4]. D. Burton, Ph.L. Toint, On the use of the inverse shortest paths algorithm for recovering linearly correlated costs, Math. Programming 63 (1994) 1 – 22.
[5]. J. Zhang, Z. Liu, Calculating some inverse linear programming problems, J. Comput. Appl. Math. 72 (1996) 261 – 273.
[6]. S. Ahmed, Y. Guan, The inverse optimal value problem, Math. Programming 102 (2005) 91 – 110.
[7]. X.J. Wang, S.Y. Feng, The Optimality Theory of Bilevel System, Science Publishers, Beijing, 1995.

Tao Zhang "A coadapted coevolutionary particle swarm optimization algorithm based on bilevel programming for inverse optimal problem. "IOSR Journal of Mathematics (IOSR-JM) 14.3 (2018): 12-15.