Information Sciences 256 (2014) 184–196

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Estimation of distribution algorithm for a class of nonlinear bilevel programming problems

Zhongping Wan [a], Lijun Mao [a], Guangmin Wang [b,*]

[a] School of Mathematics and Statistics, Wuhan University, Wuhan 430072, PR China
[b] School of Economics and Management, China University of Geosciences, Wuhan 430072, PR China

ARTICLE INFO

Article history:
Received 2 September 2012
Received in revised form 4 September 2013
Accepted 7 September 2013
Available online 18 September 2013

Keywords:
Nonlinear bilevel programming problem
Karush–Kuhn–Tucker (KKT) condition
Estimation of distribution algorithm (EDA)

ABSTRACT

In this paper, a novel evolutionary algorithm called estimation of distribution algorithm (EDA) is proposed for solving a special class of nonlinear bilevel programming problems (BLPPs) in which the lower level problem is a convex programming problem for each given upper level decision. This special type of BLPP is transformed into a equivalent single-level constrained optimization problem using the Karush-Kuhn-er conditions of the lower level problem. Then, we propose an EDA based on the statistical information of the superior candidate solutions to solve the transformed problem. We stress that the new population of individuals is sampled from the probabilistic distribution of those superior solutions. Thus, one of the main advantages of EDA over most other meta-heuristics is its ability to adapt the operators to the structure of the problem, although adaptation in EDA is usually limited by the initial choice of the probabilistic model. In addition, two specific rules are established in the initialization procedure to make use of the hierarchical structure of BLPPs and to handle the constraints. Moreover, without requiring the differentiability of the objective function, or the convexity of the search space of the equivalent problem, the proposed algorithm can address nonlinear BLPPs with non-differentiable or non-convex upper level objective function and upper level constraint functions. Finally, the proposed algorithm has been applied to 16 benchmark problem; in five of these problems, all of the upper level variables and lower level variables are 10-dimensional. The numerical results compared with those of other methods reveal the feasibility and effectiveness of the proposed algorithm.

1. Introduction

The bilevel programming problem (BLPP) addresses an optimization problem whose constraint region is determined implicitly by another (parametric) mathematical programming problem [15]. This nested system arises when two decision makers, ordered within a hierarchical structure, may have conflicting objectives. Considering the possible reactions of the decision maker at the lower level (the follower), the decision maker at the upper level (the leader) optimizes his/her objective function first. Afterward, the follower selects his/her decision under the given decision of the leader. Because plentiful real-life problems, such as transportation, management and economics, engineering design, supply chain planning, principal-agent problems, and health insurance, can be transformed into BLPPs [11–13,19,38,52,54], it is significant to solve these problems in an efficient way [7,14,29,39,44,53].

* Corresponding author. Tel.: +86 18971049799.
E-mail addresses: mathwanzhp@whu.edu.cn (Z. Wan), Stephenmao@whu.edu.cn (L. Mao), wgm97@163.com (G. Wang).

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.09.021

However, the bilevel programming problem is non-convex and non-differentiable. Even the linear bilevel programming problem, the simplest version of BLPPs, was shown to be NP-hard by Jeroslow [21]. This result was later proved by Bard [6] and Ben-Ayed and Blair [8]. Afterward, Vicente et al. [40] confirmed that even searching for the local optima of the linear BLPPs is NP-hard. See Ref. [17] for more detailed discussion of the complexity issues in the linear bilevel programming.

Over the past thirty years, much progress has been made in developing solution methods for BLPPs. Such methods can generally be classified into the following categories [14,44]: extreme-point search approaches for the linear case, branch-and-bound approaches, complementary pivoting methods, decent methods, penalty function methods and trust-region methods. Because methods that are based on meta-heuristics do not require differentiability of the objective functions, gradient information, or convexity of the search space, they have attracted considerable attention for their potential as alternative methods for bilevel programming problems. Heuristic algorithms (including simulated annealing, neural network, particle swarm optimization, genetic algorithms, tabu search, bee colony algorithm, cuckoo optimization algorithm and chaotic annealing) and their applications have received a large amount of attention over the past few years [18,22,23,31,36,37,41,44–50]. This attention has motivated the use of heuristic algorithms for bilevel programming problems [44].

As a new meta-heuristic, an estimation of distribution algorithm (EDA) is one of the stochastic optimization techniques that explores the space of potential solutions by building and sampling explicit probabilistic models of promising candidate solutions. This model-based approach to optimization has allowed EDAs to solve many large and complex problems, such as multi-objective knapsack, military antenna design, identification of clusters of genes with similar expression profiles, economic dispatch, forest management, portfolio management, cancer chemotherapy optimization, and environmental monitoring network design (see [20] and the references therein). It is important to stress that no other techniques were shown to be capable of achieving better performance or solving problems of comparable size and complexity than EDAs in most of these applications. In this paper, we extend the application of EDA to address bilevel programming problems as an attempt. Notably, there are no reports on solving BLPPs through EDA-type methods. Through building an explicit probabilistic model for the selected candidate solutions that have been obtained so far, EDAs extract the global statistical information of the promising solutions. Then, an EDA method improves the quality of the candidate solutions by generating the new generation of solutions via sampling based on the model, instead of via the GA-type operators of crossover and mutation. Thereby, EDA surpasses other intelligent algorithms for efficiency because of its avoidance of the complex parameter selection [26]. Similar to other meta-heuristics methods, EDA is implemented without requiring the differentiability of the objective function or the convexity of the search space.

In this paper, we mainly concentrate on the nonlinear bilevel programming problem, in which both the upper level objective function and the lower level objective function are all real-valued convex functions. First, we transform the nonlinear bilevel programming problem into a single-level mathematical programming problem by replacing the lower level problem with its Karush–Kuhn–Tucker (KKT) conditions. To handle the original upper level constraints and the KKT conditions, a new fitness function with a penalty scheme is constructed to evaluate the quality of each candidate solution. In solving constrained optimization, addressing an infeasible solution is very difficult. Hence, we initially generate the first generation, while obeying two preset rules: (i) the upper level decision of each individual belongs to the projection of the feasible solution region onto the solution space of the upper level problem, (ii) the lower level decision of each candidate solves the lower level problem under the corresponding given upper level decision. Through the above rules, we can initially generate the solution in the inducible region of the BLPP to improve the efficiency of the algorithm. Then, in each generation, we select the superior solutions based on their fitness function values and build up a probabilistic model for the selected feasible solutions. Afterward, the algorithm generates new populations based on the model and regularizes new individuals to satisfy the ordinary restrictions. Note that the proposed EDA calls for no special characters of the upper level objective function and constraint functions. Thus, it can effectively address the nonlinear bilevel programming problem with a non-differentiable upper level objective function and non-differentiable or non-convex upper level constraint functions, only if the lower level problem is convex for each given upper level decision and the solution to the problem (BLPP) exists.

The organization of the remainder of this paper is as follows. The general formulation and basic concepts of BLPP are presented in Section 2. After a brief recall of the estimation of distribution algorithm, the proposed algorithm for the nonlinear BLPP is given after the reformulation of BLPP in Section 3. Experimental results regarding 16 benchmark problems are presented in Section 4 and the paper concludes with a summary in Section 5.

2. The bilevel programming problem

2.1. The general formulation and basic concepts for BLPP

The general formulation of a bilevel programming problem can be stated as follows:

$$(BLPP)\ \min_{x\in X, y\in Y} F(x,y) \tag{1}$$

$$s.t.\quad G(x,y)\leqslant 0, \tag{2}$$

where y, for each vector x, belongs to the solution set of the so-called lower level problem:

$$\min_{y\in Y} \quad f(x,y) \tag{3}$$

$$s.t. \quad g(x,y) \leqslant 0, \tag{4}$$

where F, f: $R^n \times R^m \to R$ are the objective functions of the upper and lower level problems, respectively. While the vector-valued functions G: $R^n \times R^m \to R^q$ and g: $R^n \times R^m \to R^p$ are called the upper and lower level constraints, respectively. $x \in X \subset R^n$ and $y \in Y \subset R^m$ are the decision variables under the control of the upper and lower level decision makers, respectively. The sets X and Y set additional restrictions for the involved variables.

The basic definitions that pertain to the bilevel programming problem are the following:

(i) The constraint region of the bilevel programming problem:
$S = \{(x, y) \in X \times Y: G(x, y) \leqslant 0, g(x, y) \leqslant 0\}$.
(ii) The projection of S onto the leader's decision space:
$S(X) = \{x \in X: \exists y \in Y, \text{such that } (x, y) \in S\}$.
(iii) The set of feasible solutions for the lower level for each fixed $x \in S(X)$:
$S(x) = \{y \in Y: g(x, y) \leqslant 0\}$.
(iv) The lower level reaction set for each fixed $x \in S(X)$:
$P(x) = \{y \in Y: y \in \text{argmin}\{f(x, y): y \in S(x)\}\}$.
(v) The inducible region of the bilevel programming problem:
$IR = \{(x, y) \in X \times Y: (x, y) \in S, y \in P(x)\}$.

In terms of this notation, the bilevel programming problem is to optimize the upper level objective function $F(x, y)$ over the inducible region. Consequently, the bilevel programming problem can be restated as

$$\min_{x,y}\{F(x,y) : (x,y) \in IR\}. \tag{5}$$

Thus, the induced region IR can be regarded as the feasible region of BLPP. This set is usually non-convex and may be disconnected or even empty in the presence of upper level constraints [15].

To ensure that the nonlinear bilevel programming problem is well posed, it is common to make the following assumptions.

Assumption 1. The polyhedron S is nonempty and compact.

Assumption 2. $F(x, y)$, $G(x, y)$, $f(x, y)$ and $g(x, y)$ are all continuous functions, while $f(x, y)$ and $g(x, y)$ are continuously differentiable.

Assumption 3. For each decision x that is selected by the leader, the follower has some responses, i.e., $P(x) \neq \emptyset$.

Note that the lower level reaction set for some given upper level decision might not be a singleton. There are two possible modeling approaches when considering the bilevel programming problem, namely, the *optimistic* or *weak* approach and the *pessimistic* or *strong* approach [15]. It is assumed that the leader is allowed to select the element in the lower level feasible set that suits him/her best whenever the follower's reaction set $P(x)$ is not a singleton for the bilevel programming problem in the *optimistic* case. Conversely, in the case of a *pessimistic* bilevel programming problem, the cooperation of the leader and follower fails to be accepted, and the follower chooses the behavior, which is least favorable for the leader.

The algorithm presented in this paper is designed for the *optimistic* bilevel programming problem. Based on this problem, the feasible solution and optimal solution for the bilevel programming problem are described as follows:

Definition 1. The vector $(\bar{x}, \bar{y})$ is called the feasible solution of the bilevel programming problem if $(\bar{x}, \bar{y}) \in IR$.

Definition 2. The vector (x^*, y^*) is called the optimal solution of the bilevel programming problem if $(x^*, y^*) \in IR$ and $F(x^*, y^*) \leqslant F(\bar{x}, \bar{y})\ \forall (\bar{x}, \bar{y}) \in IR$.

2.2. The development of the BLPP

In this subsection, we transform the nonlinear BLPP into an equivalent single-level programming problem with the above notation and assumptions.

With the assumption that the constraint region is nonempty and compact, the follower has his/her room for seeking the corresponding responses for each decision that is made by the leader. In other words, the feasible region of the lower level problem is nonempty. Thus, the follower's rational reaction set and the inducible region are guaranteed to be nonempty. Therefore, the existence of the optimal solution for BLPP is ensured.

Note that, if the lower level problem is a convex parametric programming problem that satisfies the Manasarian-Fromowitz constraint qualification (MFCQ) [30] for each point $x \in S(X)$, then it is equivalent to the following Karush–Kuhn–Tucker conditions [9]:

$$\nabla_y L(x,y,\lambda) = \nabla_y f(x,y) + \lambda^T \nabla_y g(x,y) = 0, \tag{6}$$
$$g(x,y) \leqslant 0, \tag{7}$$
$$\lambda \geqslant 0, \tag{8}$$
$$\lambda^T g(x,y) = 0, \tag{9}$$

where the Lagrangian function of the lower level problem is formed as

$$L(x,y,\lambda) = f(x,y) + \lambda^T g(x,y), \tag{10}$$

$\nabla_y(\cdot)$ denotes the gradient of the function $(\cdot)$ with respect to y, and λ is the vector of Lagrangian multipliers.

Then, we have the following theorem [15]:

Theorem 1. *Let $(x, y) \in S$; then, a necessary and sufficient condition that $(x, y) \in IR$ is that there exists a $\lambda \geqslant 0$ such that (x, y, λ) satisfies Eqs. (6)–(9)*

To this end, by replacing the lower level problem with its KKT conditions (6)–(9), we can transform the nonlinear BLPP (1)–(4) into the following equivalent single-level mathematical programming problem (SLP) [16,51]:

$$(SLP) \min_{x,y,\lambda} F(x,y) \tag{11}$$
$$s.t. \quad G(x,y) \leqslant 0, \tag{12}$$
$$\nabla_y f(x,y) + \lambda^T \nabla_y g(x,y) = 0, \tag{13}$$
$$g(x,y) \leqslant 0, \tag{14}$$
$$\lambda \geqslant 0, \tag{15}$$
$$\lambda^T g(x,y) = 0, \tag{16}$$
$$x \in X, \quad y \in Y. \tag{17}$$

Therefore, we have the following theorem [5]:

Theorem 2. *A necessary and sufficient condition that (x^*, y^*) solves BLPP is that there exists $\lambda^* \geqslant 0$ such that (x^*, y^*, λ^*) is the optimal solution of SLP.*

3. The proposed algorithm for BLPPs

3.1. The description of estimation of the distribution algorithm

Estimation of distribution algorithm (EDA), introduced by Mühlenbein and Paaβ [32], is a new optimization technique that has been successfully applied to optimization, engineering, cluster analysis, machine learning and design problems (see [1,2] and the references therein). Inspired by the natural evolution of the species, genetic algorithms (GAs) perform well and improve the results that are obtained by previous algorithms in addressing different problems. However, GAs call for complex investigation by adjusting a large number of parameters in order to achieve good performance. Furthermore, GAs receive poor performance when addressing certain problems, such as deceptive and separable problems, because of the disruption of the building blocks. EDA is a combination of GA and statistical learning for employing probabilistic models to identify relationships among variables. Abandoning the traditional GA-type evolutionary operators, EDA captures global statistical information in the search space through constructing a probabilistic model for the selected superior solutions obtained so far and generates new candidates via sampling the established probabilistic distribution; this approach is an important step that distinguishes EDAs from most of the other meta-heuristics. The goal is not to perfectly represent the population of promising solutions but instead to represent a more general distribution that captures the features of the selected solutions. Hence, this approac can make these solutions better than other candidate solutions. In general, EDAs iterate the following three phases until the termination criteria are satisfied [2,35]:

Step 1 Select good individuals from a population.
Step 2 Estimate the probability distribution from the selected individuals.
Step 3 Generate new individuals (i.e., offspring) from the estimated distribution.

The last phase replaces the traditional recombination and mutation operators of the genetic and evolutionary algorithms.

Similar to the development of GAs, research on EDAs has been extended from the discrete domain to continuous optimization during the past two decades [20]. In those former attempts, EDAs are applied to optimization problems that have feasible solutions and that are encoded by binary strings that have a fixed length. The representative algorithms proposed include the Univariate Marginal Distribution Algorithm (UMDA), the Bivariate Marginal Distribution Algorithm (BMDA), and the Bayesian Optimization Algorithm (BOA). In recent years, more researchers have turned to developing EDAs for optimization problems in the continuous case, mainly because it is significant to investigate those problems that have real-valued domains in fields of practical engineering and scientific research. There are two main approaches to applying EDAs into searching the optima of those problems that are in a continuous domain:

(1) Map the real-valued vectors to discrete variables and solve the discrete problem that is obtained through EDAs, as discussed above, and finally, map the solutions back to the real-valued domain of the original problem.
(2) Admit the variables to be real-valued and use a probability model for the continuous domain under the basic formation of EDAs.

Regardless of whether the interactions among the variables are accounted for or not, different types of EDAs arise with the development of various types of statistical learning. For example, the Estimation of Multivariate Normal Algorithm (EMNA) was constructed based on the estimation of a multivariate Gaussian distribution for probabilistic characteristics, namely, the vector of means and the variance–covariance matrix of the candidate solutions [27]. Readers can also refer to [10,28,34] for more details about EDA.

3.2. The estimation of distribution algorithm for BLPPs

To handle the constraints in SLP, a new fitness function is provided by using a penalty method to evaluate the quality of each candidate solution. Hence, those better candidates receive higher fitness values, while the relatively worse solutions show lower fitness values. Moreover, the constraints of the original BLPP are also applied to keep individuals from the new population in the feasible region. Based on these processes, the estimation of distribution algorithm uses real encoding for both the upper and lower level variables, and the Lagrangian multipliers are designed as follows.

3.2.1. Coding and initialization rules

First, the floating vector coding method is adopted for its faster convergence rate and higher computing precision. To keep each individual of the initial population in the feasible region and to make full use of the hierarchical structure of BLPP, the following two rules for the initialization step are set in advance:

(1) the upper level decision of each individual belongs to the projection of the feasible solution region onto the solution space of the upper level problem.
(2) the lower level decision of each candidate solves the lower level problem for the corresponding fixed upper level problem.

The population size of each generation is denoted by *PopSize*. We first randomly generate *PopSize* vectors, $x^i \in S(X)$, $i = 1, 2, \ldots, PopSize$, as the upper level components of the initial points. Then, for each fixed x^i, the vector y^i is obtained by solving the following lower level parametric problem (using the MATLAB Optimization Toolbox):

$$\begin{aligned} &\min_{y \in Y} f(x^i, y) \\ &s.t. \quad g(x^i, y) \leqslant 0, \end{aligned} \tag{18}$$

Afterward, with the fixed upper and lower level variables (x^i, y^i), the corresponding KKT multipliers λ^i can be gained through solving the lower level problem's KKT conditions (6)–(9). Hence, the individuals generated in this way satisfy the original constraints and the KKT conditions of the follower's problem. All of the points $z^i = (x^i, y^i, \lambda^i)$ form the initial population $Pop(0)$ with the population size *PopSize*.

3.2.2. Fitness assignment

In this section, the penalty method is applied to address the constraints of SLP. We denote the fitness function at the point $z = (x, y, \lambda)$ with $H(z) = H(x, y, \lambda)$, which is defined by the following formula:

$$H(x,y,\lambda) = -F(x,y) - p_1 \sum_{i=1}^{q} \max\{0, G_i(x,y)\} - p_2 \sum_{j=1}^{m} \max\{0, |\nabla_{y_j} L(x,y,\lambda)|\} - p_3 \sum_{t=1}^{p} \max\{0, |\phi(-g_t(x,y), \lambda_t)|\}, \tag{19}$$

where m, p, q are defined as those in Section 2, and the NCP function ϕ: $R \times R \to R$ is defined for the complementary conditions from the lower level problem's KKT conditions as

$$\phi(a,b) = \min\{a,b\}. \tag{20}$$

Note that the NCP function has the following property, which guarantees its efficiency in reformulating the complementary conditions:

$$\phi(a,b)=0 \Longleftrightarrow a \geqslant 0, b \geqslant 0, ab=0. \tag{21}$$

Let p_1, p_2, p_3 denote the sufficiently large penalty parameters. Then, the fitness function that is defined in this way has the following property: the fitness value at any point z, which violates the upper level constraints or KKT conditions, is smaller than that at any feasible point. In this way, feasible solutions can be distinguished from infeasible solutions, and better individuals can be selected effectively. To be noticed, the optimal solution of the BLPP will achieve the highest fitness value.

3.2.3. The selection of the most promising solutions

From the above ranked population, a subset of the most promising solutions are selected by the selection operator. In this paper, the selection operator is truncation selection with a threshold τ. For example, $\tau = 50\%$ means selecting the 50% best solutions.

3.2.4. The construction of the probabilistic model

Throughout this paper, the approach that we proposed for BLPPs is primarily based on one of the simplest estimation of distribution algorithms, which is called the Univariate Marginal Distribution Algorithm for continuous domains (UMDAc) [24,25], which is motivated by UMDA in the discrete case. With the assumption that all of the univariate distributions are normal, UMDAc employs a product of independent one-dimensional normal densities to factorize the joint probability density function.

$$\mathrm{F}(Y|\mu,\sigma)=\prod_{i=0}^{n-1}N(Y_i;\mu_i,\sigma_i)=\prod_{i=0}^{n-1}\frac{1}{\sqrt{2\pi}\sigma_i}e^{-\frac{1}{2}\left(\frac{Y_i-\mu_i}{\sigma_i}\right)^2}, \tag{22}$$

where n is the dimension of the vector, Y_i denotes the i-th component of the vector, and the mean value and standard deviation of the i-th component are denoted by μ_i and σ_i, respectively. For each component, these two parameters are estimated from the previous population by employing the following two equations:

$$\mu_i=\frac{1}{N}\sum_{j=0}^{N-1}x_i^j, \tag{23}$$

$$\sigma_i=\sqrt{\frac{1}{N-1}\sum_{j=0}^{N-1}\left(x_i^j-\mu_i\right)^2}, \tag{24}$$

where N is the number of samples.

To estimate the distribution of the superior solution group, the proposed algorithm employs a joint density function that is factorized as the following equation:

$$\mathrm{F}(z,\mu,\Sigma)=\prod_{k=0}^{n+m+p}f(z_k;\mu_k,\sigma_k)=\prod_{k=0}^{n+m+p}\frac{1}{\sqrt{2\pi}\sigma_k}e^{-\frac{1}{2}\left(\frac{z_k-\mu_k}{\sigma_k}\right)^2}. \tag{25}$$

With the assumption of independence, the estimation step is simplified into calculating the mean and standard deviation of each component of z under the guidance of Eqs. (23) and (24). In each generation, these parameters are estimated according to the selected superior candidates, in such way as to extract the global statistical information of the previous population. Hence, the parameters are updated adaptively and accordingly in each iteration.

3.2.5. Offspring generation

Based on the probabilistic model built for the parent population, the proposed algorithm takes samples to generate individuals randomly. To make these newly generated candidates appropriate and to make the algorithm efficient, we set up a regulation for them. In other words, only the new individual $z=(x, y, \lambda)$ satisfies the conditions that $x \in X$, $y \in Y$ and $\lambda \geqslant 0$, and it can be selected as a candidate solution. Otherwise, the algorithm should abandon it and make another sample operation. Repeat the generating and "check-up" procedures until *PopSize* new proper individuals are generated. Calculate the fitness function values of the offspring candidates and sort them based on the values of the fitness function. Therefore, the new population can be obtained by using the selected offspring individuals and the superior part of the parent population, while maintaining a fixed population size *PopSize*.

3.2.6. Termination criteria

The termination criteria are used to decide when to stop the iteration and return the solution results. In this algorithm, if the iteration is executed to the preset maximal number of generations *Gen_Max* or the best solution's fitness function value has not been improved in the last consequential *Gen_Min* generations, then stop the algorithm and take the best solution that is present (as judged by the fitness function values) as the approximate global optimum of the BLPP.

3.2.7. The steps of the proposed EDA

The steps of the estimation of distribution algorithm for nonlinear BLPPs is presented as follows:

Step 1. (Initialization) Randomly generate *PopSize* initial points $z^i = (x^i, y^i, \lambda^i)$, $i = 1, 2, \ldots, PopSize$, obeying the initialization rules. All of the points z^i form the initial population $Pop(0)$, and let $g = 0$;
Step 2. (Evaluation) Evaluate the fitness $H(z)$ at each point in $Pop(g)$ according to the formula in (19).
Step 3. (Selection) Use the truncation selection with the threshold τ to select $N(<PopSize)$ superior candidate solutions from $Pop(g)$ to form the parent population $S(g)$ based on the population's fitness function values.
Step 4. (Estimation) Estimate the distribution of the selected parent population $S(g)$ to establish a probabilistic model $M(g)$.
Step 5. (Offspring Generating) Sample *PopSize* proper candidates according to the established model $M(g)$ and choose $(PopSize - N)$ relatively better candidates based on their fitness values. The chosen best offspring candidates and the parent population $S(g)$ make up the next population $Pop(g + 1)$.
Step 6. (Termination Criteria) Make a judgment on the condition of termination. When the maximal number of generations *Gen_Max* is attained or the best result (using the evaluation method in Step 2) has not been improved for *Gen_Min* straight iterations, then stop the algorithm and return the optimal solution. Otherwise, let $g = g + 1$ and go to Step 3.

4. Numerical experiments

In this section, we tested the estimation of distribution algorithm on selected examples from the literature [42,43,45]. These examples involve linear, quadratic, and other nonlinear cases. Exs. 7–16 involve non-convex and non-differentiable objective functions, whereas Exs. 12–16 are bilevel programming problems that have 10 dimensions for both the leader's and follower's variables, respectively. Some characteristics of these examples are shown in Table 1.

Some of these examples are often solved to demonstrate the performance of the algorithms in the literature. In those examples, we present the known best solutions that are found by some existing algorithms to make a comparison with our computational results, when a theoretically optimal solution cannot be determined. The best solution (x^*, y^*) and the upper levels objective function $F(x^*, y^*)$ as well as the lower levels objective function $f(x^*, y^*)$ at the best solution (x^*, y^*) are recorded. The best solution in the references is denoted by $(\bar{x}, \bar{y})$, and the upper levels objective function $F(\bar{x}, \bar{y})$ as well as the lower levels objective function $f(\bar{x}, \bar{y})$ at the best solution $(\bar{x}, \bar{y})$ are recorded.

The parameters are chosen as follows: the population size of each generation is fixed with *PopSize* = 1000, the size of the selected population is $N = 400$, the penalty parameters are uniformly set at $p_1 = p_2 = p_3 = 10{,}000$, and the threshold of the truncation selection is $\tau = 50\%$. The algorithm will stop when one of the termination criteria has been reached, where *Gen_Max* = 20 for Exs. 1–11, *Gen_Max* = 100 for Exs. 12–16, and *Gen_Min* = 5 for all of the problems.

First, we execute the proposed EDA with 50 independent runs on each problem. A comparison of the results in our paper with the results in the references is given in Tables 2 and 3. In Tables 2 and 3, Column "Ref." sequentially lists the results obtained by Wang et al. for Ex. 1 [45], Bard for Ex. 2 [7], Aiyoshi and Shimuzu for Ex. 3 [3], Amouzegar for Ex. 4 [4], Oduguwa and Roy for Exs. 5–6 [33] and Wang et al. for Exs. 7–12 [42].

It can be seen from Tables 2 and 3 that, for Exs. 3, 5, and 9, the solutions that are found by our proposed EDA are better than or equal to the solutions that are found by the compared algorithms in the references. Although the solutions found by EDA for Exs. 1, 2, 4 and 11 are worse than the solutions obtained by the compared algorithms, the former are almost as good as the latter. Note that the solutions found by the algorithm provided in [3] for Ex. 3 are not the global optima, and the

Table 1
Types of the test examples.

No.	Type	Scale	Functions
Ex.1	Linear	$n = 3$; $m = 1$	Convex, differential
Ex.2	Linear	$n = 2$; $m = 3$	Convex, differential
Ex.3	Linear	$n = 2$; $m = 2$	Convex, differential
Ex.4	Quadratic	$n = 2$; $m = 2$	Convex, differential
Ex.5	Quadratic	$n = 1$; $m = 1$	Convex, differential
Ex.6	Quadratic	$n = 1$; $m = 1$	Convex, differential
Ex.7	Nonlinear	$n = 2$; $m = 2$	Non-convex, non-differential
Ex.8	Nonlinear	$n = 2$; $m = 2$	Non-convex, non-differential
Ex.9	Nonlinear	$n = 2$; $m = 2$	Non-convex, non-differential
Ex.10	Nonlinear	$n = 2$; $m = 2$	Non-convex, non-differential
Ex.11	Nonlinear	$n = 2$; $m = 2$	Non-convex, non-differential
Ex.12	Nonlinear	$n = 10$; $m = 10$	Non-convex, non-differential
Ex.13	Nonlinear	$n = 10$; $m = 10$	Non-convex, non-differential
Ex.14	Nonlinear	$n = 10$; $m = 10$	Non-convex, non-differential
Ex.15	Nonlinear	$n = 10$; $m = 10$	Non-convex, non-differential
Ex.16	Nonlinear	$n = 10$; $m = 10$	Non-convex, non-differential

Table 2
Comparison of the solutions by EDA with those in the related references for Exs. 1–11.

No.	Our proposed EDA (x^*, y^*)	Ref. $(\bar{x}, \bar{y})$
Ex. 1	(4.000517, 14.999931, 9.199862, 2)	(4, 15, 9.2, 2)
Ex. 2	(2e−6, 0.899997, 4e−6, 0.6, 0.400005)	(0, 0.9, 0, 0.6, 0.4)
Ex. 3	(0, 30, −10, 10)	(25, 30, 5, 10)
Ex. 4	(0, 2, 1.875, 0.90625)	(0, 2, 1.875, 0.9063)
Ex. 5	(10, 0)	(10.04, 0.1429)
Ex. 6	(10.0005, 9.9995)	(10.03, 9.969)
Ex. 7	(0, 30, −10, 10)	(0, 30, −10, 10)
Ex. 8	(0, 30, −10, 10)	(0, 30, −10, 10)
Ex. 9	(0, 30, −10, 10)	(0, 30, −10, 10)
Ex. 10	(15.29735, 8.245, 10, 8.245)	(19.563, 5.272, 10, 5.272)
Ex. 11	(15.29735, 8.245, 10, 8.245)	(12.860, 6.205, 10, 6.205)

Table 3
Comparison of the upper and lower level objective function values at the best solution by EDA with those in the references for Exs. 1–11.

No.	Our proposed EDA		Ref.	
	$F(x^*, y^*)$	$f(x^*, y^*)$	$F(\bar{x}, \bar{y})$	$f(\bar{x}, \bar{y})$
Ex. 1	41.199207	−9.198828	41.2	−9.2
Ex. 2	−29.200009	3.200009	−29.2	3.2
Ex. 3	0	100	5	0
Ex. 4	−12.678711	−1.015625	−12.68	−1.016
Ex. 5	82	0	82.44	0.271
Ex. 6	100.01	2.5e-07	100.58	0.001
Ex. 7	0	100	0	100
Ex. 8	0	100	0	100
Ex. 9	0	100	0	100
Ex. 10	1.6241e-04	28.0619	6.86e-15	91.45
Ex. 11	1.6241e-04	28.0619	1.47e-14	8.18

Table 4
Comparison of the upper and lower level objective function values at the best solution from EDA with those in Ref. [43] for Exs. 12–16.

No.	Our proposed EDA x^*	Ref. [43] $\bar{x}$
Ex. 12	(1, 1, 1, 1, 1, 1, 1, 1, 1, 1)	(0.99999998, 0.99999999, 1.00000006,0.99999999, 1.00000000, 1.00000001, 0.99999999,0.99999992, 0.99999998, 1.00000001)
Ex. 13	(1, 1, 1, 1, 1, 1, 1, 1, 1, 1)	(1.00000000, 1.00000000, 1.00000000, 1.00000000, 1.00000000, 0.99999999, 0.99999999, 1.00000000, 0.99999999, 0.99999999)
Ex. 14	(1.149034, 0.08833383, 1.254797, 1.182997, 2.130051, 1.742112, 0.3082794, 1.591319, 1.409942, −0.2195419)	(0.21650445, 0.70129941, 0.73361882, 0.27478271, −0.39242837, 0.18513273, 0.69041802, 0.46409579, 0.07205230, 0.77427495)
Ex. 15	(−1.275612, 0.4240169, −1.292204, −0.57017, 1.238698, 2.83057, 1.313386, 0.65589,-2.799304, −1.467915)	(1.55838477, −0.64856587, 1.13718010, 0.10569137, −1.76868487, 0.58635685, 0.74817824, −0.84311991, 0.59501691, 0.49591424)
Ex. 16	(1, 1, 1, 1, 1, 1, 1, 1, 1, 1)	(1.00347786, 1.00046181, 0.99991034, 0.99996156, 1.00034019, 1.00020616, 0.99985678, 0.9990869, 1.00016481, 0.99989127)
	y^*	$\bar{y}$
Ex. 12	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0)	10e−3(−0.04845, −0.03471, 0.11674, 0.09264, 0.2121, 0.09969, 0.07125, 0.05798, 0.04344, 0.03512)
Ex. 13	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0)	10e−9(−0.89425, 0.85196, 0.85196, −0.89425, −0.89425, −0.89425, −0.67166, 0.85196, 0.85196, 0.85196)
Ex. 14	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0)	10e−3(−0.04398, −0.00529, 0.02614, −0.41031, −0.08806, 0.52992, 0.29883, −0.39756, −0.57348, 0.58285)
Ex. 15	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0)	10e−9(0.90664, −0.75161, −0.33431, −0.54023, 0.90664, −0.54023, 0.90663, −0.75161, −0.75161, 0.90664)
Ex. 16	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0)	10e−3(−0.03757, −0.15561, 0.03938, 0.06108, −0.27616, 0.35876, −0.29831, 0.02883, 0.00558, −0.02131)

solution in this paper is the certain answer. Note that, within the acceptable precision, our results for Ex. 10 are much better than those obtained by Wang et al. [42]. In other words, both values of the upper level objective function are acceptable and close enough to the optimal value 0, while the value of the lower level objective function obtained by Wang et al. is worse

Table 5
Comparison of the upper and lower level objective function values at the best solution by EDA with those in Ref. [43] for Exs. 12–16.

No.	Our proposed EDA		Ref. [43]		Conv. rate
	$F(x^*, y^*)$	$f(x^*, y^*)$	$F(\bar{x}, \bar{y})$	$f(\bar{x}, \bar{y})$	
Ex. 12	0	1	6.21498e−04	1	59
Ex. 13	0	1	8.50866e−09	1	56
Ex. 14	4.64e−06	1	2.03246e−05	1	79
Ex. 15	1.12e−05	1	6.17254e−08	1	22
Ex. 16	0	1	7.22651e−03	1	56

than ours. Hence, it can be concluded that the solutions found by EDA for these former 11 problems are global optimal solutions or close-to-optimal solutions.

For a further test, we solve the high-dimensional nonlinear BLPPs (10 dimensions for both the leader's and follower's variables, respectively) from Ref. [43]. Similarly, we execute the proposed EDA with 50 independent runs on Exs. 12–16. The comparison of the results in our paper with those in Ref. [43] are listed in Tables 4 and 5. The best results found by the EDA and algorithms in the corresponding references for those problems are provided in Tables 4 and 5. The Column "Conv. Rate" (which represents the convergence rate of the algorithm) in Table 5 shows the counts of the generation when obtaining the optimal solution to each problem from Ex. 12 to Ex. 16.

For the high-dimensional Exs. 12–16, because EDA fails to obtain the optimal solutions within 20 generations, the greatest generation to stop the algorithm is set at "*Gen_Max* = 100". It can be seen from Tables 4 and 5 that EDA finds the optimal or close-to-optimal solutions of these problems within no more than 79 iterations, and it is especially notable that only 22 iterations were required for Ex. 15. Furthermore, the solutions found by EDA for Exs. 12, 13 and 16 are exactly the global optimal solutions, which are obviously better than those found by the compared algorithm. These two tables show that the solutions found by EDA and the compared algorithm are close-to-optimal solutions. It should also be noted that, due to the two pre-set rules in the procedure for initialization, all of the solutions found by EDA solve the corresponding lower level problems well.

To summarize, the solution results and the comparison shown in these four tables reveal the feasibility of the proposed algorithm for addressing both low-dimensional and high-dimensional (10 dimensions for both the leader's and follower's variables, respectively) nonlinear BLPPs, regardless of the differentiability of the upper level objective functions or the convexity of the decision space.

5. Conclusions

In this paper, the EDA is proposed for solving a class of nonlinear bilevel programming problems. First, we transform the nonlinear BLPP into an equivalent single-level optimization problem by replacing the lower level problem with its KKT conditions. Hence, we can handle the BLPPs in a much easier way. To solve the equivalent problem, we construct the estimation of distribution algorithm. In contrast to GA, special steps for estimating the distribution of the superior solution population and sampling new candidates are introduced instead of applying the crossover and mutation operators. Through this specific scheme, the quality of each generation becomes improved in the long run, and the solution or an approximate solution is achieved. The proposed EDA is executed on 16 benchmark problems, and the experimental results have been compared with those found in the related references. The comparison reveals that the proposed algorithm can also be used as a competitive approach for addressing BLPPs, even for the case in which the upper level objective functions are non-differentiable or the variables of both the upper level problem and the lower level problem are all high-dimensional. Thus, the evolutionary algorithm herein can be considered to be an effective tool for solving nonlinear BLPPs.

In our future work, the following will be researched:

(1) Research to demonstrate the efficiency of the proposed algorithm by solving more and larger-scale examples generated as the references.
(2) Comparison with other algorithms by solving more examples.

Acknowledgments

The authors thank the anonymous referees. This research was partially funded by the National Natural Science Foundation of China (Nos. 71171150, 71201146), the Social Science Foundation of the Ministry of Education (No. 10YJC630233) and the Fundamental Research Funds for the Central Universities (No. CUG120410).

Appendix A. Test problems

Problem Ex 1. [45]:

$$\max_{x,y} \quad F(x,y) = -x_1 + 2x_2 + x_3 + 3y,$$

where y solves:

$$\begin{aligned}
&\max_{y} f(x,y) = 2x_1 - x_3 - 4y,\\
&s.t.\ 0.2x_1 + x_3 + y \leqslant 12, \quad -2x_2 + y \leqslant 10,\\
&\quad -3x_1 - x_2 + x_3 \leqslant 12, \quad -x_1 + y \leqslant -2,\\
&\quad -2x_1 - x_3 \leqslant -2, \quad x_2 \leqslant 15, \quad x \geqslant 0, \quad y \geqslant 2.
\end{aligned}$$

Problem Ex 2. [7]:

$$\begin{aligned}
&\min_{x,y} F(x,y) = -8x_1 - 4x_2 + 4y_1 - 40y_2 - 4y_3,\\
&s.t.\ x \geqslant 0,
\end{aligned}$$

where y solves:

$$\begin{aligned}
&\min_{y} f(x,y) = x_1 + 2x_2 + y_1 + y_2 + 2y_3,\\
&s.t.\ -y_1 + y_2 + y_3 \leqslant 1, \quad 2x_1 - y_1 + 2y_2 - 0.5y_3 \leqslant 1,\\
&\quad 2x_2 + 2y_1 - y_2 - 0.5y_3 \leqslant 1, \quad y \geqslant 0.
\end{aligned}$$

Problem Ex 3. [3]:

$$\begin{aligned}
&\min_{x,y} F(x,y) = 2x_1 + 2x_2 - 3y_1 - 3y_2 - 60,\\
&s.t.\ x_1 + x_2 + y_1 - 2y_2 \leqslant 40 \quad 0 \leqslant x \leqslant 50,
\end{aligned}$$

where y solves:

$$\begin{aligned}
&\min_{y} f(x,y) = (y_1 - x_1 + 20)^2 + (y_2 - x_2 + 20)^2,\\
&s.t.\ x_1 - 2y_1 \geqslant 10, \quad x_2 - 2y_2 \geqslant 10,\\
&\quad -10 \leqslant y \leqslant 20.
\end{aligned}$$

Problem Ex 4. [4]:

$$\begin{aligned}
&\min_{x,y} F(x,y) = -x_1^2 - 3x_2^2 - 4y_1 + y_2^2,\\
&s.t.\ x_1^2 + 2x_2 \leqslant 4, \quad x \geqslant 0,
\end{aligned}$$

where y solves:

$$\begin{aligned}
&\min_{y} f(x,y) = 2x_1^2 + y_1^2 - 5y_2,\\
&s.t.\ x_1^2 - 2x_1 + x_2^2 - 2y_1 + y_2 \geqslant -3,\\
&\quad x_2 + 3y_1 - 4y_2 \geqslant 4, \quad y \geqslant 0.
\end{aligned}$$

Problem Ex 5. [33]:

$$\min_{x,y} \quad F(x,y) = (x-1)^2 + (y-1)^2,$$

where y solves:

$$\min_{y} \quad f(x,y) = 0.5y^2 + 500y - 50xy.$$

Problem Ex 6. [33]:

$$\min_{x,y} F(x,y) = x^2 + (y-10)^2,$$
$$s.t.\ -x+y \leqslant 0, \quad 0 \leqslant x \leqslant 15,$$

where y solves:

$$\min_{y} f(x,y) = (x+2y-30)^2,$$
$$s.t.\ x+y \leqslant 20, \quad 0 \leqslant y \leqslant 20.$$

Problem Ex 7. [42]:

$$\min_{x,y} F(x,y) = |2x_1 + 2x_2 - 3y_1 - 3y_2 - 60|,$$
$$s.t.\ x_1 + x_2 + y_1 - 2y_2 \leqslant 40, \quad 0 \leqslant x \leqslant 50,$$

where y solves:

$$\min_{y} f(x,y) = (y_1 - x_1 + 20)^2 + (y_2 - x_2 + 20)^2,$$
$$s.t.\ 2y_1 - x_1 + 10 \leqslant 0, \quad 2y_2 - x_2 + 10 \leqslant 0,$$
$$-10 \leqslant y \leqslant 20.$$

Problem Ex 8. [42]: The problem is the same as $Q7$ except for

$$F(x,y) = |\sin(2x_1 + 2x_2 - 3y_1 - 3y_2 - 60)|.$$

Problem Ex 9. [42]: The problem is the same as $Q7$ except for

$$F(x,y) = |\tan(2x_1 + 2x_2 - 3y_1 - 3y_2 - 60)|.$$

Problem Ex 10. [42]:

$$\min_{x,y} F(x,y) = |\sin((x_1-30)^2 + (x_2-20)^2 - 20y_1 + 20y_2 - 225)|,$$
$$s.t.\ 30 - x_1 - 2x_2 \leqslant 0, \quad x_1 + x_2 - 25 \leqslant 0, \quad x_2 \leqslant 15,$$

where y solves:

$$\min_{y} f(x,y) = (y_1 - x_1)^2 + (y_2 - x_2)^2,$$
$$s.t.\ 0 \leqslant y \leqslant 10.$$

Problem Ex 11. [42]: The problem is the same as $Q10$ except for

$$F(x,y) = |\tan((x_1-30)^2 + (x_2-20)^2 - 20y_1 + 20y_2 - 225)|.$$

Problem Ex 12. [43]:

$$\min_{x,y} \quad F(x,y) = \sum_{i=1}^{10} [|x_i - 1| + |y_i|],$$

where y solves:

$$\min_{y} f(x,y) = exp\{[1 + \sum_{i=1}^{10}(y_i^2/4000) - \prod_{i=1}^{10}\cos(y_i/\sqrt{i})]\sum_{i=1}^{10}x_i^2\},$$
$$s.t.\ -\pi \leqslant y \leqslant \pi.$$

Problem Ex 13. [43]:

$$\min_{x,y} \quad F(x,y) = \sum_{i=1}^{10}[|x_i - 1| + |y_i|],$$

where y solves:

$$\min_{y} f(x,y) = exp\{[100 + \sum_{i=1}^{10}(y_i^2 - 10\cos(2\pi y_i))]\sum_{i=1}^{10}x_i^2\},$$
$$s.t. \ -3 \leqslant y \leqslant 3.$$

Problem Ex 14. [43]:

$$\min_{x,y} \quad F(x,y) = |\sin(\sum_{i=1}^{10}[|x_i - 1| + |y_i|])|,$$

where y solves:

$$\min_{y} f(x,y) = exp\{[1 + \sum_{i=1}^{10}(y_i^2/4000) - \prod_{i=1}^{10}\cos(y_i/\sqrt{i})]\sum_{i=1}^{10}x_i^2\},$$
$$s.t. \ -\pi \leqslant y \leqslant \pi.$$

Problem Ex 15. [43]:

$$\min_{x,y} \quad F(x,y) = |\sin(\sum_{i=1}^{10}[|x_i - 1| + |y_i|])|,$$

where y solves:

$$\min_{y} f(x,y) = exp\{[100 + \sum_{i=1}^{10}(y_i^2 - 10\cos(2\pi y_i))]\sum_{i=1}^{10}x_i^2\},$$
$$s.t. \ -3 \leqslant y \leqslant 3.$$

Problem Ex 16. [43]:

$$\min_{x,y} \quad F(x,y) = \sum_{i=1}^{10}[|x_i - 1| + |y_i|],$$

where y solves:

$$\min_{y} f(x,y) = exp\{[1 + \sum_{i=1}^{10}((x_i y_i)^2/4000) - \prod_{i=1}^{10}\cos(x_i y_i/\sqrt{i})]\},$$
$$s.t. \ -\pi \leqslant y \leqslant \pi.$$

References

[1] C.W. Ahn, Advances in Evolutionary Algorithms: Theory, design and practice, Springer, 2006.
[2] C.W. Ahn, J. An, J. Yoo, Estimation of particle swarm distribution algorithms: combining the benefits of PSO and EDAs, Information Sciences 192 (1) (2010) 109–119.
[3] E. Aiyoshi, K. Shimuzu, A solution method for the static constrained Stackelberg problem via penalty method, IEEE Transactions on Automatic Control 29 (12) (1984) 1111–1114.
[4] M.A. Amouzegar, A global optimization method for nonlinear bilevel programming problems, IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics 29 (6) (1999) 771–777.
[5] J.F. Bard, Convex two-level optimization, Mathematical Programming 40 (1988) 15–27.
[6] J.F. Bard, Some properties of the bilevel linear programming, Journal of Optimization Theory and Applications 68 (2) (1991) 146–164.
[7] J.F. Bard, Practical Bilevel Optimization: Algorithms and Applications, Kluwer Academic Publishers, 1998.
[8] O. Ben-Ayed, C. Blair, Computational difficulties of bilevel linear programming, Operational Research 38 (3) (1990) 556–560.
[9] M.S. Bazarar, H.D. Sherali, C.M. Shetty, Nonlinear Programming: Theory and Algorithms, second ed., John Wiley and Sons, New York, 1993.
[10] J. Ceberio, E. Irurozki, A. Mendiburu, J.A. Lozano, A Review on Estimation of Distribution Algorithms in Permutation-based Combinatorial Optimization Problems, Progress in Artificial Intelligence, Springer, 2011.

[11] M. Cecchini, J. Ecker, M. Kupferschmid, et al, Solving nonlinear principal-agent problems using bilevel programming, European Journal of Operational Research 230 (2) (2013) 364–373.
[12] S.-W. Chiou, Optimization of limited network capacity with toll settings, Information Sciences 179 (1–2) (2009) 109–119.
[13] S.-W. Chiou, A bi-level programming for logistics network design with system-optimized flows, Information Sciences 179 (14) (2009) 2434–2441.
[14] B. Colson, P. Marcotte, G. Savard, An overview of bilevel optimization, Annals of Operations Research 153 (2007) 235–256.
[15] S. Dempe, Foundation of Bilevel Programming, Kluwer Academic Publishers., 2002.
[16] S. Dempe, A.B. Zemkoho, On the Karush–Kuhn–Tucker reformulation of the bilevel optimization problem, Nonlinear Analysis 75 (2012) 1202–1218.
[17] X. Deng, Complexity issues in bilevel linear programming, in: A. Migdalas, P.M. Pardalos, P. Varbrand (Eds.), Multilevel Optimization: Algorithms and Applications, Kluwer Academic Publishers., Dordrecht, 1998, pp. 149–164.
[18] C. García-Martínez, M. Lozano, F.J. Rodríguez-Díaz, A simulated annealing method based on a specialised evolutionary algorithm, Applied Soft Computing 12 (2012) 573–588.
[19] F. Gzara, A cutting plane approach for bilevel hazardous material transport network design, Operations Research Letters 41 (1) (2013) 40–46.
[20] M. Hauschild, M. Pelikan, An introduction and survey of estimation of distribution algorithms, Swarm and Evolutionary Computation 1 (3) (2011) 111–128.
[21] R.G. Jeroslow, The polynomial hierarchy and a simple model for competitive analysis, Mathematical Programming 32 (2) (1985) 146–164.
[22] Y. Jiang et al, An augmented Lagrangian multiplier method based on a CHKS smoothing function for solving nonlinear bilevel programming problems, Knowledge-Based Systems (2013), http://dx.doi.org/10.1016/j.knosys.2013.08.017.
[23] K. Lan, U. Wen, H.S. Shih, E.S. Lee, A hybrid neural network approach to bilevel programming problems, Applied Mathematics Letters 20 (2007) 880–884.
[24] P. Larrañaga, R. Etxeberria, J.A. Lozano, J.M. Peña, Optimization by learning and simulation of Bayesian and Gaussian networks (Technical Report), Department of Computer Science and Artificial Intelligence, University of the Basque Country, 1999.
[25] P. Larrañaga, R. Etxeberria, J.A. Lozano, J.M. Peña, Optimization in continuous domains by learning and simulation of Gaussian networks, in: Proceedings of the 2000 Genetic and Evolutionary Computation Conference Workshop Program, 2000, pp. 201–204.
[26] P. Larrañaga, J.A. Lozano, Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, Kluwer Academic Publishers., 2002.
[27] P. Larrañaga, J.A. Lozano, E. Bengoetxea, Estimation of Distribution Algorithms based on multivariate normal distributions and Gaussian networks. Technical Report KZZA-IK-1-01, Department of Computer Science and Artificial Intelligence, University of the Basque Country, 2001.
[28] J.A. Lozano, P. Larrañaga, I. Iñza, E. Bengoetxea, Towards a New Evolutionary Computation: Advances in the Estimation of Distribution Algorithms, Springer, 2006.
[29] J. Lu, C.G. Shi, G.Q. Zhang, On bilevel multi-follower decision making: general framework and solutions, Information Sciences 176 (11) (2006) 1607–1627.
[30] O.L. Mangasarian, S. Fromovitz, The Fritz John necessary optimality condition in the presence of equality and inequality constraints, Journal of Mathematical Analysis and Applications 17 (1967) 37–47.
[31] M. Marić, Z. Stanimirović, N. Milenković, Metaheuristic methods for solving the bilevel uncapacitated facility location problem with clients' preferences, Electronic Notes in Discrete Mathematics 39 (2012) 43–50.
[32] H. Mühlenbein, G. Paa, From recombination of genes to the estimation of distributions I Binary parameters, in: Parallel Problem Solving from Nature – {PPSN IV} Lecture Notes in Computer Science 1141, 1996, pp. 178–187.
[33] V. Oduguwa, R. Roy, Bi-level optimisation using genetic algorithm, in: Proceedings of the 2002 IEEE International Conference on Artificial Intelligence Systems (ICAIS'02), 2002, pp. 322–327.
[34] I.J. Ocenasek, Parallel Estimation of Distribution Algorithms, PhD thesis, Brno University of Technology, 2002.
[35] M. Pelikan, D.E. Goldberg, E. Lobo, A survey of optimization by building and using probabilistic models, Computational Optimization and Applications 21 (1) (2002) 5–20.
[36] R. Rajabioun, Cuckoo optimization algorithm, Applied Soft Computing 11 (2011) 5508–5518.
[37] F.J. Rodriguez, M. Lozano, C. Garca-Martnez, et al, An artificial bee colony algorithm for the maximally diverse grouping problem, Information Sciences 230 (2013) 183–196.
[38] M. Sakawa, T. Matsui, Interactive fuzzy random two-level linear programming based on level sets and fractile criterion optimization, Information Sciences 238 (2013) 163–175.
[39] L. Vicente, P. Calamai, Bilevel and multilevel programming: a bibliography review, Journal of Global Optimization 5 (1994) 291–306.
[40] L. Vicente, G. Savard, J. Judice, Descent approaches for quadratic bilevel programming, Journal of Optimization Theory and Applications 81 (1994) 379–399.
[41] Z.P. Wan, G.M. Wang, B. Sun, A hybrid intelligent algorithm by combining particle swarm optimization with chaos searching technique for solving nonlinear bilevel programming problems, Swarm and Evolutionary Computation 8 (2013) 26–32.
[42] Y. Wang, Y. Jiao, H. Li, An evolutionary algorithm for solving nonlinear bilevel programming based on a new constraint-handling scheme, IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews 35 (2) (2005) 221–232.
[43] Y. Wang, H. Li, C. Dang, A new evolutionary algorithm for a class of nonlinear bilevel programming problems and its global convergence, INFORMS Journal on Computing 23 (4) (2011) 618–629.
[44] G. Wang, Z. Wan, X. Wan, Bibliography on bilevel programming, Advances in Mathematics 36 (5) (2007) 513–529.
[45] G. Wang, X. Wang, Z. Wan, S. Jia, Adaptive genetic algorithms for solving bilevel linear programming problem, Applied Mathematics and Mechanics 28 (12) (2007) 1433–1440.
[46] A.R. Yildiz, A comparative study of population-based optimization algorithms for turning operations, Information Sciences 210 (2012) 81–88.
[47] A.R. Yildiz, Optimization of cutting parameters in multi-pass turning using artificial bee colony-based approach, Information Sciences 220 (2013) 399–407.
[48] A.R. Yildiz, Hybrid Taguchi-differential evolution algorithm for optimization of multi-pass turning operations, Applied Soft Computing 13 (3) (2013) 1433–1439.
[49] A.R. Yildiz, A new hybrid differential evolution algorithm for the selection of optimal machining parameters in milling operations, Applied Soft Computing 13 (3) (2013) 1561–1566.
[50] A.R. Yildiz, A new hybrid artificial bee colony algorithm for robust optimal design and Manufacturing, Applied Soft Computing 13 (2013) 2906–2912.
[51] A.J. Zaslavski, Necessary optimality conditions for bilevel minimization problems, Nonlinear Analysis 75 (2012) 1655–1678.
[52] D.L. Zhang, G.H. Lin, Bilevel direct search method for leader-follower problems and application in health insurance, Computers & Operations Research (2012). http://dx.doi.org/10.1016/j.cor.2012.12.005.
[53] G.Q. Zhang, J. Lu, J. Montero, et al, and Kth-best algorithm for linear trilevel programming, Information Sciences 180 (4) (2010) 481–492.
[54] M. Zugno, J.M. Morales, P. Pinson, et al, A bilevel model for electricity retailers' participation in a demand response market environment, Energy Economics 36 (2013) 182–197.