Economic Research-Ekonomska Istraživanja

ISSN: 1331-677X (Print) 1848-9664 (Online) Journal homepage: https://www.tandfonline.com/loi/rero20

An efficient hybrid differential evolutionary algorithm for zbilevel optimisation problems

Xing Bao, Titing Cui, Zhongliang Zheng & Haiyun Liu

To cite this article: Xing Bao, Titing Cui, Zhongliang Zheng & Haiyun Liu (2019) An efficient hybrid differential evolutionary algorithm for zbilevel optimisation problems, Economic Research-Ekonomska Istraživanja, 32:1, 3016-3033, DOI: 10.1080/1331677X.2019.1656097

To link to this article: https://doi.org/10.1080/1331677X.2019.1656097

ECONOMIC RESEARCH-EKONOMSKA ISTRAŽIVANJA
2019, VOL. 32, NO. 1, 3016–3033
https://doi.org/10.1080/1331677X.2019.1656097

An efficient hybrid differential evolutionary algorithm for zbilevel optimisation problems

Xing Bao[a], Titing Cui[b], Zhongliang Zheng[c] and Haiyun Liu[d]

[a]College of Business Administration, Zhejiang University of Finance and Economics, Hangzhou, China; [b]Applied and Computational Mathematics, KTH Royal Institute of Technology, Stockholm, Sweden; [c]College of Economic and Management, China Agricultural University, Beijing, China; [d]Department of Economics, School of Liberal Arts, Tulane University, New Orleans, LA, USA

ABSTRACT
Bilevel problems are widely used to describe the decision problems with hierarchical upper–lower-level structures in many economic fields. The bilevel optimisation problem (BLOP) is intrinsically NP-hard when its objectives and constraints are complex and the decision variables are large in scale at both levels. An efficient hybrid differential evolutionary algorithm for BLOP (HDEAB) is proposed where the optimal lower level value function mapping method, the differential evolutionary algorithm, k-nearest neighbours (KNN) and a nested local search are hybridised to improve the computational accuracy and efficiency. To show the performance of the HDEAB, numerical studies were conducted on SMD (Sinha, Maro and Deb) instances and an application example of optimising a venture capital staged-financing contract. The results demonstrate that the HDEAB outperforms the BLEAQ (bilevel evolutionary algorithm based on quadratic approximations) greatly in solving the BLOPs with different scales.

ARTICLE HISTORY
Received 6 November 2018
Accepted 10 June 2019
Published online 9 September 2019

KEYWORDS
Bilevel optimisation problem; differential evolutionary algorithm; KNN; nested local search

JEL CLASSIFICATIONS
C10; C63; C18

1. Introduction

Many decision-making problems in the economic, public and private sectors could be described as bilevel optimisation problems (BLOPs), such as the taxing strategy (Wei, Liang, Liu, Mei, & Tian, 2014), environment economics (Sinha, Malo, & Deb, 2013), homeland security (Wein, 2009), the toll-setting problem (Brotcorne, Labbé, Marcotte, & Savard, 2001), operational decision-making problem (Haghighat & Kennedy, 2012), transportation policy formulation (Sinha, Malo, & Deb, 2015), spatial targeting of agri-environmental policy (Whittaker et al., 2017), and so on. The main characteristic of the BLOP is the hierarchical upper–lower-level structure (i.e., leader–follower structure), where the private profit-seeking follower at the lower level could well be in conflict with the leader's objective at the upper level. Typically, there

CONTACT Zhongliang Zheng profzzl@foxmail.com
Supplemental data for this article can be accessed online at https://doi.org/10.1080/1331677X.2019.1656097

are three features in a bilevel decision-making process: (1) decision-makers at both levels give their decisions in sequence: first for the leader, second for the follower, and both of them aim to optimise their own objectives with constraints; (2) the information powers are asymmetric at different levels: the leader at the upper level has complete knowledge of the follower at the lower level, while the follower only observes the leader's decisions; and (3) decision-making tasks at both levels are interlinked: the follower optimises the objectives according to the leader's decisions, while the leader has to incorporate the follower's response into the procedure of optimising his objectives.

From the point of view of mathematical programming, a nested strategy is required to solve the BLOP, where the lower level optimisation problem is served as a constraint of the upper level optimisation problem. However, it is not an easy task to solve the BLOP in a fast and precise way because the lower level optimisation problem is required to be solved for each decision at the upper level. Non-convexity and disconnectedness might rise easily at the lower level even for a simple bilevel problem (e.g., functions at both levels are linear, convex and differentiable), not to mention that the functions at both levels are non-convex and non-differentiable in real-world applications. Furthermore, with the number of decision variables increasing at both upper and lower level problems, the task of solving the BLOP could be turned into a disaster: it is impossible to solve the BLOP mathematically even when the bilevel problems can be reduced into single-level ones (Chen & Florian, 1995). It is still unlikely the global optimality for the BLOP will be obtained when the number of its decision variables is large in scale (e.g., the number at both levels is more than 10, otherwise the scale is defined as small) (Sinha et al., 2014). Researchers have proved the BLOP to be strong NP-hard in nature, which is similar to the scheduling, production planning and routing problems (Brajković, Pernić, & Ikonić, 2018; Pérez-Rodríguez, Hernandez-Aguirre, & Jöns, 2017; Zhang, Wen, Zhu, & Hu, 2017). It is more applicable to have the satisficing solutions instead of global optimums for the BLOP. More and more researchers have been focusing on using the meta-modelling method with heuristic algorithms (e.g., evolutionary algorithm (EA)) to solve the BLOPs in recent decades. Sinha and his co-authors are the notable ones who have successfully improved the computational accuracy and efficiency of solving different BLOPs by a bilevel evolutionary algorithm based on quadratic approximations (BLEAQ) (Sinha, Malo, & Deb, 2018). However, the BLEAQ still performs poorly when the BLOP is not sufficiently regular or simple (e.g., non-linear, non-convex, non-differentiable, etc.), thus room has been left for further improvements.

This study proposes a hybrid differential evolutionary algorithm for BLOP (HDEAB), which greatly outperforms the BLEAQ and contributes in two aspects. (i) It greatly improves the computational accuracies for both levels. First, to avoid the multi-valued (or set-valued) phenomenon in BLEAQ, the optimal lower level value function mapping is adopted as the meta-modelling method to reduce the BLOP into a single-level problem. Second, the differential evolutionary algorithm (DEA) is adopted as the optimisation engine at both levels, which avoids the discontinuous, non-convex, non-differential phenomenon on the feasible regions and objectives. (ii) The HDEAB also highlights the computational efficiency. First, the k-nearest neighbours (KNN) technique is used to save the calls of function evaluations (FEs) at the lower level. Second, a nested local search method is used to further save the calls of FEs (i.e., the times to invoke the evaluation of FEs) at the lower level by providing the upper level candidate decision variables with higher quality. Third, the DEA is applied to boost further the computational efficiency.

The remainder of this study is organised as follows. Section 2 reviews the related literature in BLOP in the recent decades. Section 3 analyses the performances of three meta-modelling methods in reducing the bilevel problem into a single-level one. Section 4 introduces the principles and pseudo-codes of the HDEAB in solving the BLOP. Section 5 provides detailed numerical studies for comparing the performances of the HDEAB and BLEAQ on SMD (Sinha, Maro, & Deb 2018) instances and an application example of optimising a venture capital stage-financing contract with different scales. Section 6 gives the conclusions.

2. Literature review

For recent decades, much literature has presented various algorithms to solve the BLOPs, which can be categorised into two streams: (1) using the classical algorithms; and (2) adopting the evolutionary algorithms.

2.1. The classical algorithms for solving the BLOP

The most attractive merit of the classical algorithms lies in the fact that the optimal results can be theoretically guaranteed. When the lower level problem is convex and sufficiently regular, it can reduce the bilevel problem into a single-level one by using the Karush–Kuhn–Tucker (KKT) method, where the Lagrangian multipliers and complementarity constraints are introduced to incorporate the lower level problem into the upper level one. However, this method increases the total number of decision variables, which gives rise to three undesired phenomena: (1) non-convexity happens easily even when the lower level problem is sufficiently regular; (2) the computational burden could rise exponentially, because more decision variables have to be optimised (Sinha et al., 2018); and (3) it could not easily handle the situation when the objectives and constraints of BLOP are non-linear or non-convex.

To overcome the shortcomings of the KKT method, many researchers resorted to other approaches. For example, a descent algorithm was developed for solving a non-linear BLOP (Savard & Gauvin, 1994). A trust region method was adopted to solve the generalised BLOP (Marcotte, Savard, & Zhu, 2001), where the bilevel problem was locally approximated with a model involving a linear program at the upper level and linear variational inequality at the lower level. In Colson, Marcotte, and Savard (2005), the BLOP was solved in two steps: first, a bilevel linear quadratic model was used to approximate the BLOP, thus it was reduced into a single-level problem; second, the reduced problem could be solved with a mixed integer programming method.

The above classical methods might be restricted in their applications in the real world. For a more general BLOP where the objectives and constraints (i.e., functions) involved are non-convex and non-differentiable, the classical methods are unable to solve the BLOP exactly. It is computation-expensive to obtain a solution when the numbers of decision variables at both levels are large in scale, because the BLOP is strongly NP-hard in nature (Benayed & Blair, 1990; Vicente, Savard, & Judice, 1994).

2.2. The evolutionary algorithms for solving the BLOP

Since most BLOPs are intrinsically NP-hard, it is realistic to have a satisficing solution rather than the optimal one. The EA requires few on the types of objectives and constraints, so that it complements the shortcomings of the classical methods. The studies with the EA contributed almost 10% of all studies in solving BLOPs (Sinha et al., 2018).

The work in Mathieu, Pittard, and Anandalingam (1994) might be the first study to solve the BLOP with an EA. In this work, the upper level problem was solved by a genetic algorithm (GA), while the lower level problem was solved by a linear programming method. This pure nested strategy solved the BLOP in a nested manner: the lower level optimisation problem was solved for every given upper level decision. Recently, more researchers considered the EA to be better than the classical methods in computational efficiency and accuracy for solving the BLOP with more complex conditions and constraints (Yin, 2000). Based on a constraint-handling scheme, an EA with a specifically-designed crossover operator was proposed to solve the non-linear BLOP (Wang, Jiao, & Li, 2005). When the objectives were linear and the constraint regions were polyhedrons at both levels, an algorithm was developed by combining the classical extreme point enumeration techniques with a genetic search, which was able to solve the quasi-concave BLOP with the linear lower level function (Calvete, Gale, & Mateo, 2008). The work proposed by Wang, Li, and Dang (2011) moved a great step forward, where a new EA was proposed and could handle a non-differentiable upper level objective and non-convex lower level problem. A much simpler approach was proposed in Angelo, Krempser, and Barbosa (2013), where two differential evolutionary techniques were applied on both levels with the continuous decision variables. Recently, an EA embedded with the KKT proximity was proposed in Sinha, Malo, and Deb (2017), which demonstrated a promising future of solving the BLOPs by hybridising EA and other approximate approaches.

There is room for improving the pure nested strategies mentioned above in solving BLOPs. It is often unnecessary to solve the lower level optimisation problem for every upper level decision (Sinha et al., 2018). There will be a great improvement in boosting the computational efficiency, when the lower level calls of the FEs could be saved. Sinha and his co-authors showed that the lower level calls of the FEs could be saved greatly through the meta-modelling methods. For example, they introduced a BLEAQ where the reaction set mapping method was adopted (Sinha et al., 2013). This approach demonstrated its capability of handling the BLOPs with different kinds of complexities by using a smaller number of FEs. Later, they improved the BLEAQ by archiving and local search techniques (Sinha et al. 2014). This improved BLEAQ offered a significant improvement in reducing the calls of FEs at both levels. Recently, they proposed a meta-modelling-based strategy to iteratively approximate the optimal lower level value function (Sinha, Malo, & Deb, 2016). It not only freed the high restrictive class of bilevel problems, but also saved the computational burden for some complex BLOPs. Angelo, Krempser, and Barbosa (2014) reported a DEA assisted with a similarity-based surrogate model (i.e., KNN) to reduce greatly the FEs on the lower level problem by the KNN techniques.

2.3. Distinctiveness of this research

This study was motivated by Angelo et al. (2014) and Sinha et al. (2017). Although these two works did great jobs in solving different BLOPs, there were shortcomings that hampered the computational accuracy and efficiency.

First, the reaction set mapping adopted in BLEAQ is not an ideal meta-modelling method to save the lower level calls in Sinha et al. (2017) because the quadratic approximation method used in BLEAQ requires calculating the inverse of the coefficient matrix. When the number of upper level decision variables (i.e., n) is on a large scale, the computational complexity could reach $O(n^6)$ to solve the inverse once for each variable at the lower level. Furthermore, the quadratic approximation method might perform poorly on the boundary of the original approximation region because the discontinuity could rise due to the constraints of the decision variables at both levels. Besides, the multi-valued phenomenon would raise the discontinuity in the reaction set mapping method used in Sinha et al. (2017), thus causing the low computational accuracy.

Second, compared with the BLEAQ in Sinha et al. (2017), there is an obvious advantage of the algorithm proposed in Angelo et al. (2014), which could make it easier to use the KNN candidate as an approximation of the lower level optimal solution. However, the disadvantage of this method is that it randomly selects only one KNN candidate of the lower level optimal decisions for the given upper level decision variables. This method might pick the undesired KNN candidate while ignoring the well-behaved one due to the random selection. Therefore, low computational efficiency and accuracy would occur simultaneously in Angelo et al. (2014).

Third, the algorithms proposed in Angelo et al. (2014) and Sinha et al. (2017) often converge rather slowly for some BLOPs where the multi-valued phenomenon might exist at lower level problems. Once the meta-modelling method fails to find the real optimum for the upper level problem, the convergence of approximation will be rather time-consuming.

To overcome the shortcomings of the algorithms in Angelo et al. (2014) and Sinha et al. (2017), we propose a more efficient algorithm called HDEAB. First, to ensure a higher computational accuracy, the optimal lower level value function mapping method is utilised to reduce the BLOPs into single-level ones. Second, the KNN method is adopted to save the lower level calls of FEs. Third, a nested local search is used for boosting the computational efficiency by providing the upper level candidate with higher quality. By numerical studies on the SMD instances, we demonstrate that the HDEAB has the same robustness as the BLEAQ presented in Sinha et al. (2017). What is more, the HDEAB performs withhigher computational accuracy and efficiency, and could solve the BLOP with larger scale of decision variables than the BLEAQ.

3. Reduction methods for BLOPs

Consider a general BLOP with n and m decision variables at the upper and lower level problems, respectively (i.e., $x_u \in X_u \subset \mathbb{R}^n$ and $x_l \in X_l \subset \mathbb{R}^m$, where the subscripts u and l denote the upper and lower levels, respectively). Decision-makers at both levels are private-profit maximisers with different objectives (i.e., $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ and $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$). A general BLOP is given in Equation (1):

$$\begin{aligned} & min\ F(x_u, x_l) \\ & s.t.\ x_l \in \underset{x_l \in X_l}{\operatorname{argmin}}\{f(x_u, x_l) : g_j(x_u, x_l) \le 0, j = 1, 2, \ldots, J\} \\ & G_k(x_u, x_l) \le 0, k = 1, 2, \ldots, K \\ & x_u \in X_u, x_l \in X_l \end{aligned} \tag{1}$$

where G_k and g_j denote the constraints for the upper and lower level problems.

For the given upper level decision variables x_u, the follower optimises the lower level decision variables x_l to maximise his/her own objective. While for the leader, he/she has to incorporate x_l into his/her own objective before optimising x_u. The procedures for solving the BLOP are highly nested, hence it is computation-expensive when the numbers of decision variables at both levels increase. If the bilevel problem is reduced into a single-level one, then the computational efficiency of solving the BLOP could be greatly improved.

The reaction set mapping and the optimal lower level value function mapping method are meta-modelling methods, which are most related to this study, to reduce the BLOP. They focus on the same reduction principle: incorporating the lower level problem into the upper level by various mapping methods. Once the reduction work is done, various optimisation techniques could be utilised to approximate the optimal solution of the BLOP. Note that even the bilevel problem could be reduced into a single-level one, and the task of solving the BLOP is still NP-hard in nature as the number of decision variables increases.

3.1. Reaction set mapping method

In this method, the constraints defined at the lower level problem can be represented by the reaction set mapping (denoted as the ψ-mapping), which is given by

$$\psi(x_u) = \underset{x_l \in X_l}{\operatorname{argmin}}\{f(x_u, x_l) : g_j(x_u, x_l) \le 0, j = 1, 2, \ldots, J\} \tag{2}$$

Then the BLOP can be reduced into a single-level constrained optimisation problem, which is given by

$$\begin{aligned} & min\ F(x_u, x_l) \\ & s.t.\ x_l \in \psi(x_u) \\ & G_k(x_u, x_l) \le 0, k = 1, 2, \ldots, K \\ & x_u \in X_u \end{aligned} \tag{3}$$

Once the value of $\psi(x_u)$ is determined, a smaller search space of x_l could be obtained. Particularly when the $\psi(x_u)$ is single-valued, there is no need to optimise x_l because $x_l = \psi(x_u)$ always holds. Only x_u needs to be optimised in the X_u space, which is given by Equation (4):

$$\begin{gathered} min\ \hat{F}(x_u) \\ s.t.\ \hat{G}_k(x_u) \leq 0, k = 1, \ldots, K \\ x_u \in X_u \end{gathered} \tag{4}$$

where $\hat{F}(x_u) = F(x_u, \psi(x_u))$ and $\hat{G}_k(x_u) = G_k(x_u, \psi(x_u))$

When $\psi(x_u)$ is set-valued, the optimisation of x_u will face serious problems of discontinuity or local optimums. No optimisation technique can promise that it is easy and straightforward to approximate $\psi(x_u)$. For example, in Figure 1 the trace could be discontinuous and non-differentiable (e.g., traces 1 and 2) in the grey shaded area where $\psi(x_{u,0})$ is multi-valued for a certain $x_{u,0}$. The optimisation technique could not predict the $\psi(x_{u,0})$ whether it is attributed to trace 1 or to trace 2.

Three shortcomings might counteract the advantages of the ψ-mapping.

First, the widely used quadratic approximation method in the ψ-mapping always requires calculation of the inverse of the coefficient matrix. When the number of upper level decision variables (i.e., n) is large, it is a rather computationally expensive task to solve the inverse of coefficients matrix; because the computational complexity could reach $O(n^6)$ to solve the inverse once at a time. The complexity of approximation for the optimal solution of the lower level with m decision variables will reach $O(n^6 \times m)$ once at a time. It is extremely time-consuming to iteratively approximate the lower level optimum. We could not promise to get the approximation within the given time (i.e., 5 hours) under certain computational environments (e.g., given the CPU and memories).

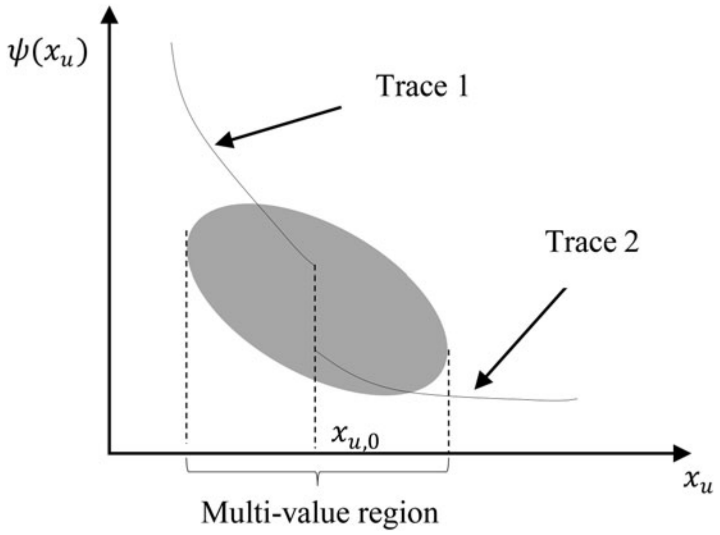

Figure 1. Multi-valued ψ-mapping and the possible traces in the multi-value region. Source: The authors.

Second, the decision variables at both levels might be always constrained. Discontinuity frequently rises on the boundaries of the decision variables at both levels. The assumption embedded in the quadratic approximation method is that the small samples come from a family of smooth or continuous functions. Therefore, it might perform poorly on the boundaries where discontinuity takes place.

Third, the algorithms proposed in Angelo et al. (2014) and Sinha et al. (2017) reported only one feasible solution of the lower level problem for each iteration. However, when there are many local optimums for the lower level problem, the quadratic approximation method could fail to give a suitable lower level solution for the given upper level decision vector, thus the convergence of the algorithms might be rather slow and the computational accuracy might be doubtful.

3.2. Optimal lower level value function mapping method

This method (denoted as the φ-mapping) is another meta-modelling method to reduce the bilevel problem. For a given x_u, the minimum lower level function value $\varphi(x_u)$ could be given as follows:

$$\varphi(x_u) = \min_{x_l \in X_l}\{f(x_u, x_l) : g_j(x_u, x_l) \le 0, j = 1, 2, \ldots, J\} \tag{5}$$

By Equation (5), the BLOP could be reduced into a single-level problem which is shown as follows:

$$\begin{aligned} & min \;\; F(x_u, x_l) \\ & s.t. \;\;\; f(x_u, x_l) \le \varphi(x_u) \\ & \qquad G_k(x_u, x_l) \le 0, k = 1, 2, \ldots, K \\ & \qquad g_j(x_u, x_l) \le 0, j = 1, 2, \ldots, J \\ & \qquad x_u \in X_u, x_l \in X_l \end{aligned} \tag{6}$$

Both x_l and x_u in Equation (6) should be predicted in the φ-mapping. It is different from the ψ-mapping where only the prediction of x_l is needed. Therefore, more computational efforts are needed to solve the BLOP by the φ-mapping than by the ψ-mapping. Interestingly, predicting x_l and x_u could overcome the shortcoming of the discontinuity in the ψ-mapping where the $\psi(x_u)$ is multi-valued. For example, as shown in Figure 2, the discontinuity phenomenon in the ψ-mapping has gone at $x_{u,0}$ in the φ-mapping. This is because the φ-mapping is always single-valued by Equation (5). Therefore, the φ-mapping could provide a relatively higher accuracy than the ψ-mapping in solving BLOP. Note that the value function $\varphi(x_u)$ is seldom known and should be still obtained by the approximation tool.

Now, let us think of approximating the $\varphi(x_u)$, and let $\hat{\varphi}(x_u)$ be the approximation of $\varphi(x_u)$. Note that there will be errors in approximating $\hat{\varphi}(x_u)$. These errors might lead $\hat{\varphi}(x_u)$ to be less than the true $\varphi(x_u)$, thus could exclude the true solution of the BLOP. To avoid this phenomenon, an error term (i.e., ε) is added on to the $\hat{\varphi}(x_u)$, thus Equation (6) can be reformulated as:

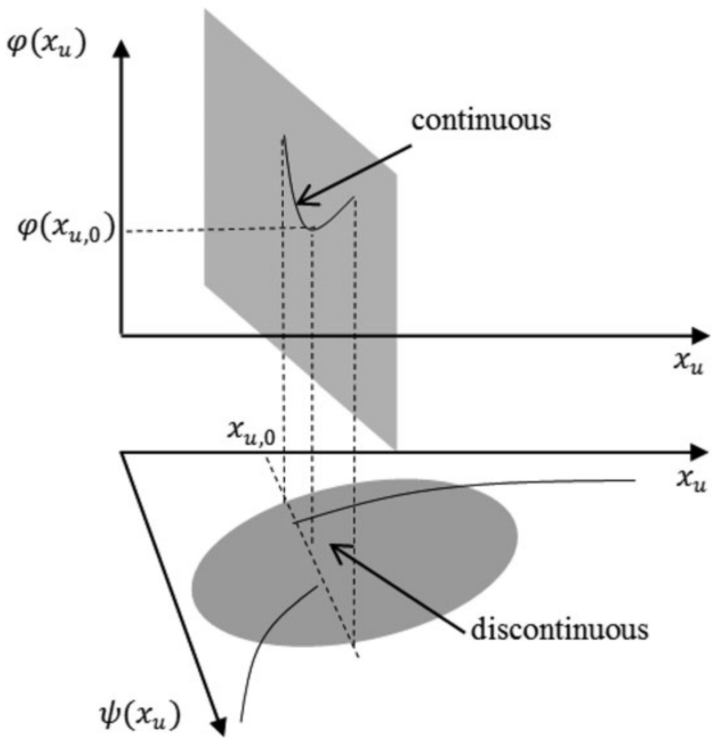

Figure 2. The advantage of the φ-mapping over the ψ-mapping in the multi-valued region. Source: The authors.

$$\begin{aligned}
&\min\ F(x_u, x_l)\\
&s.t.\ f(x_u, x_l) \le \hat{\varphi}(x_u) + \varepsilon\\
&G_k(x_u, x_l) \le 0, k = 1, 2, \ldots, K\\
&g_j(x_u, x_l) \le 0, j = 1, 2, \ldots, J\\
&x_u \in X_u, x_l \in X_l
\end{aligned} \tag{7}$$

The φ-mapping has its advantages in avoiding the discontinuity and providing a more continuous trace for approximation than the ψ-mapping, by which a higher computational accuracy in solving BLOP can be achieved. However, the computational efficiency might be hampered if no method can be used to accelerate the approximation of the φ-mapping.

4. Algorithm description

In this study, the φ-mapping is chosen to solve the BLOP for its higher computational accuracy. To improve the computational efficiency, the KNN method is first adopted to save the calls of FEs at the lower level. Second, a nested local search is used to provide the upper level candidate with higher quality, which further saves the calls of FEs at the lower level. Third, the steps of the HDEAB are outlined, which hybridises the DEA, KNN and the nested local search. The HDEAB's pseudo-codes are given in the online supplement.

4.1. Using KNN to save the calls of the FEs at lower level

As shown in Equation (7), the evaluation of $\hat{\varphi}(x_u)$ requires immense approximations at the lower level for the given x_u. Hold the fact in mind that if the approximation tasks at the lower level problem (i.e., $f(x_u, x_l) \leq \hat{\varphi}(x_u) + \varepsilon$) can be save, then the computational efficiency of the φ-mapping can be greatly improved. In order to speed up the approximation procedures for the optimal lower level solution (x_l), the KNN is used to construct a candidate $\hat{x}_l$. The candidate $\hat{x}_l$ could be iteratively estimated by Equation (8) for arbitrarily given upper level decision variables $x_{u,0}$:

$$\begin{aligned} \hat{x}_l &= \sum_{j=1}^{K} w_j x_{l,j} \\ w_j &= d^{-1}(x_{u,0}, x_{u,j}) \Big/ \sum_{i=1}^{K} d^{-1}(x_{u,0}, x_{u,j}) \end{aligned} \tag{8}$$

where $x_{u,j}$ is the archived jth-nearest upper level decision variable, $x_{l,j}$ is the corresponding jth-nearest lower level optimal solution, and $d(\cdot)$ is the Euclidean distance, which measures $x_{u,0}$ and $x_{u,j}$.

For a given $x_{u,j}$, the candidate $\hat{x}_l$ could be accepted as the true lower level optimal solution if $f(x_{u,j}, \hat{x}_l)$ satisfies the constraints in Equation (7). Otherwise, if $\hat{x}_l$ is not satisfied the neighbours of $\hat{x}_l$ should be searched to approximate the true x_l until $f(x_{u,j}, \hat{x}_l)$ satisfies the constraints.

4.2. Using a nested local search to boost the computational efficiency

In principle, if the upper level candidate $\hat{x}_u$ with a relatively high quality could be obtained, then the calls of FEs at the lower level could be further saved and the computational efficiency of the φ-mapping could be improved. However, this task should be solved in a nested way: for the lower level candidate $\hat{x}_l$ given by Equation (7), it should first be made sure that the upper level candidate $\hat{x}_u$ falls into the feasible region which is given by Equation (8). Otherwise, it cannot be hoped that $\hat{x}_l$ is the satisfied candidate.

Recall the reduced BLOP in Equation (7). If $\hat{x}_u \in X_u$ is chosen as the feasible region, then many infeasible candidates of the upper level ($\hat{x}_u$) might be brought in due to the constraints. Generally, the feasible region of $\hat{x}_u$ is no larger than X_u due to $G_k(\cdot)$ and $g_j(\cdot)$. It will be computation-expensive to exclude the infeasible candidate $\hat{x}_u$. Fortunately, a $\hat{x}_u$ with a higher quality could be obtained by solving Equation (9), which are the constraints in Equation (7):

$$\begin{aligned} &min\ 0 \\ &s.t.\ g_j(\hat{x}_u, x_l) \leq 0, j = 1, 2, \ldots, J \\ &G_k(\hat{x}_u, x_l) \leq 0, k = 1, 2, \ldots, K \\ &\hat{x}_u \in X_u, x_l \in X_l \end{aligned} \tag{9}$$

where zero denotes that the objective is constant.

The $\hat{x}_u$ that satisfies the constraints in Equation (9) will be considered as the feasible candidate for the upper level problem. Once the $\hat{x}_u$ is ready, the approximation tools (e.g., the sequential quadratic programming (SQP) or the DEA) can be utilised to predict the $\hat{x}_l$. By solving Equation (9), the total calls of FEs at the lower level could be greatly saved by using $\hat{x}_u$ with a relatively higher quality.

4.3. Hybridise the DEA, KNN and the nested local search

In this section the steps of the HDEAB are introduced, in which the DEA is adopted as the optimisation engine for both levels. The main reason is that the DEA could, astonishingly, handle the BLOP where the objectives are non-convex, non-differential or have many local optimums (Tripathy & Panda, 2017).

The HDEAB contains five steps, which are initialisation, mutation, crossover, selection and the termination criterion. To keep the consistency of the notations, let X without subscript denote the vector of the decision variables at both levels. There are M decision variables in the X. Let $\min\Pi(X)$ be the objective (e.g., $\Pi(\cdot)$ can be either $F(\cdot)$ or $f(\cdot)$ in Equation (7)).

- Step 1: Initialisation

Before using the DEA to optimise the $\Pi(X)$, the population (i.e., $X_{i,G}$, $i = 1, \ldots, N$) should first be constructed, which has the form:

$$X_{i,G} = \left[x_{1,i,G}, \ldots, x_{j,i,G}, \ldots, x_{M,i,G}\right], i = 1, \ldots, N \tag{10}$$

where $X_{i,G}$ denotes the ith individual vector at the Gth generation ($X_{i,G}$ is called the gene), N denotes the size of the population, and $x_{j,i,G}$ denotes the jth variable of the ith individual vector at the Gth generation.

Suppose x_j^L and x_j^U are the lower and upper boundaries for the j th variable, respectively. The initial population $X_{i,0}$ is randomly selected from $[x_j^L, x_j^U]$.

- Step 2: Mutation

For each $X_{i,G}$, a mutant vector $V_{i,G+1}$ (also called the donor individual) can be generated at the $G+1$th generation by the following formulation:

$$V_{i,G+1} = X_{best,G} + \mu(X_{r_1,G} - X_{r_2,G}) \tag{11}$$

where $V_{i,G+1} = [v_{1,i,G+1}, \ldots, v_{j,i,G+1}, \ldots, v_{M,i,G+1}]$, r_1 and r_2 are the indexes (which are integers) randomly chosen in $\{1, \ldots, N\}$, μ is the mutation factor, and $\mu \in [0, 2]$. r_1 and r_2 should be different, thus the size of the population (N) should be no less than three. $X_{best,G}$ is the best individual in the population.

- Step 3: Crossover

To increase the diversity of the individuals in the population, the crossover is essential in the DEA. A new trail individual $U_{i,G+1} = [u_{1,i,G+1}, \ldots, u_{j,i,G+1}, \ldots, u_{M,i,G+1}]$ could be generated by the following crossover procedure:

$$u_{j,i,G+1} = \begin{cases} v_{j,i,G+1} if \ \ rand_{j,i} \leq CR, \ or \ j = I_{rand} \\ x_{j,i,G} else \end{cases} \tag{12}$$

where $i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, M$; CR is the probability for crossover, which is a constant between $[0, 1]$; $rand_{j,i}$ is uniformly distributed in $[0, 1]$; and I_{rand} is a random integer between 1 and N, thus I_{rand} ensures that $U_{i,G+1} \neq X_{i,G}$. Each new generated trial individual $U_{i,G+1}$ should fall between the boundaries given by Equation (7).

- Step 4: Selection

After the crossover, the new trial individual $U_{i,G+1}$ is treated as the candidate solution of the individual at the $G+1$th generation. Comparison between $X_{i,G}$ and $U_{i,G+1}$ should be made to select the best candidate. The lowest function value measured by Equation (13) is used as the selection criterion for the one which will enter the next generation.

$$X_{i,G+1} = \begin{cases} U_{i,G+1}, if \Pi(U_{i,G+1}) \leq \Pi(X_{i,G}) \\ X_{i,G}, otherwise \end{cases} \tag{13}$$

First, $U_{i,G+1}$ enters the population if it satisfies Equation (14):

$$X_{i,G+1} = \begin{cases} U_{i,G+1}, if \Pi(U_{i,G+1}) \leq \max_{1 \leq j \leq N} \Pi(X_{j,G}) \\ X_{i,G}, otherwise \end{cases} \tag{14}$$

Second, if the termination criterion is unsatisfied, then go back to the mutation step.

- Step 5: Termination criterion

In this study, the variance-based termination criterion is used for both levels. This termination criterion at the G th generation for $X_{G,i}$ could be given by α_G, which is shown below:

$$\alpha_G = \sum_{i=1}^{M} \frac{\sigma_{G,i}^2}{\sigma_{0,i}^2} \tag{15}$$

where $\sigma_{G,i}^2$ and $\sigma_{0,i}^2$ denote the variances for $X_{G,i}$ and $X_{0,i}$, respectively, and M is the number of decision variables in vector $X_{G,i}$. The algorithm is terminated when $\alpha_G \leq \alpha_{stop}$. The value of α_G usually lies between zero and one. By the definition of α_G, the value of α_G is closely related to $\sigma_{0,i}^2$. If the value of $\sigma_{0,i}^2$ is very small (i.e., 1×10^{-3}), then $\alpha_G = \sum_{i=1}^{M} \sigma_{G,i}^2$ is applied. In this study, the local search is employed when $\alpha_G \leq (\alpha_{stop})^{0.1}$, otherwise the local search does not have to be employed because $X_{G,i}$ is always far from its optimal solution when $\alpha_G > (\alpha_{stop})^{0.1}$.

5. Numerical study

In this section, to compare the performance of the HDEAB and BLEAQ, numerical studies are conducted on the SMD instances and an application example given of optimising a venture capital staged-financing contract. Both approaches are run on each SMD instance 31 times, which is also the number of times that had been done in Sinha et al. (2017). All numerical studies are conducted on Matlab2016a with the hardware CPU i5 @3.20 GHz and 8 G RAM.

5.1. Performance on the small-scale SMD instances

The SMD instances contain 12 problems, of which the first eight instances are the unconstrained BLOPs and the rest are constrained. To compare the performance of the HDEAB and BLEAQ, the same parameters are used as those used in Sinha et al. (2017) to generate the small-scale SMD instances (i.e., $p = 1$, $q = 2$, $r = 1$), where the numbers of decision variables at the upper and lower levels are two and three, respectively (i.e., the scale is 2×3). For SMD6, $s = 1$ is used to generate the instance with a scale of 2×3. In this study, the scale of 2×3 is defined as small.

The parameters for the HDEAB in this study are as follows: for the unconstrained SMD instances, the population sizes for upper and lower levels are $N = 20$ and $n = 20$, while for the constrained SMD instances, $N = 30$ and $n = 30$. The rest of parameters are: $\alpha^{u}_{stop} = 10^{-6}$, $\alpha^{l}_{stop} = 10^{-6}$, $\mu = 0.9$, $CR = 0.9$ and $\varepsilon = 10^{-10}$.

Both approaches could provide a 100% success rate in approximating the benchmark solution for every SMD instance with small scale. Compared with the best-known solution of each SMD instance, there are 31 absolute differences and the calls of FEs for upper level (UL) and lower level (LL) problems, respectively. The median absolute differences (MADs) and median function evaluations (MFEs) are used to measure the computational accuracies and efficiencies at both levels.

The numerical results of the MADs and MFEs at both levels of 12 SMD instances are given in Tables 1 and 2. The results show that the HDEAB performs with much higher computational accuracies and efficiencies than the BLEAQ on each SMD instance. The average MADs at UL and LL given by the HDEAB are 10.72% and 5.33% of those given by the BLEAQ. The average MFEs at UL and LL given by the HDEAB are only 24.8% and 18.6% of those given by the BLEAQ.

5.2. Performance on the large-scale SMD instances

To our best knowledge, the largest scale of the SMD instances is 10×10, and the BLEAQ could only successfully solve the first eight unconstrained instances (Sinha et al., 2014). To investigate the largest scale that could be successfully solved by HDEAB and BLEAQ, the numbers of decision variables are gradually doubled at both levels (i.e., 5×5, 10×10 and 20×20). $\alpha^{u}_{stop} = 10^{-6}$ and $\alpha^{l}_{stop} = 10^{-6}$ are the same for both approaches. The computational time is restricted to within 5 hours for each SMD instance because it will be extremely time-consuming when the scale reaches 20×20, which might be unbearable for the decision-makers. However, one has to note that different computational environments will lead to different computational

Table 1. MADs at the UL and LL on small-scale SMD instances.

	MADs at UL			MADs at LL		
Instance	HDEAB	BLEAQ	Ratio	HDEAB	BLEAQ	Ratio
SMD1	2.27E-13	1.16E-09	1.96E-04	2.67E-13	7.13E-10	3.74E-04
SMD2	2.21E-11	5.44E-06	4.06E-06	2.54E-11	5.50E-06	4.62E-06
SMD3	3.27E-14	7.55E-06	4.33E-09	2.32E-13	5.50E-06	4.22E-08
SMD4	1.33E-08	1.15E-07	1.16E-01	1.33E-08	1.86E-06	7.15E-03
SMD5	4.68E-12	2.00E-07	2.34E-05	4.68E-12	2.50E-07	1.87E-05
SMD6	8.61E-16	1.34E-07	6.43E-09	1.29E-16	9.82E-09	1.31E-08
SMD7	6.00E-11	5.81E-06	1.03E-05	5.22E-11	9.23E-06	5.66E-06
SMD8	4.25E-06	2.21E-04	1.92E-02	7.69E-07	5.53E-05	1.39E-02
SMD9	4.30E-11	4.22E-06	1.02E-05	7.06E-11	1.16E-05	6.09E-06
SMD10	2.70E-04	1.02E-03	2.65E-01	4.68E-05	8.55E-04	5.47E-02
SMD11	1.13E-03	1.28E-03	8.83E-01	1.16E-01	2.08E-03	5.58E-01
SMD12	1.53E-04	4.56E-02	3.36E-03	1.09E-04	2.00E-02	5.45E-03

Source: Given by simulations.

consumptions. In this study, the iteration is stopped when either the stop criterion or the time restriction is reached.

Tables 3 and 4 give the numerical results of the performance of these two approaches on the same SMD instances. The largest scale for the BLEAQ is 10 × 10, which is the same as in Sinha et al. (2014). The BLEAQ cannot provide the numerical solutions for any SMD instance within 5 hours when the MFEs at the lower level exceed 1E + 07. However, the HDEAB could successfully obtain the numerical results for the scale up to 20 × 20 even when MFEs at lower level reach 1E + 09. No numerical solution will be obtained by the HDEAB and BLEAQ within 5 hours when the scale of SMD instance is larger than 20 × 20 because the MFEs at the lower level increase exponentially. The HDEAB also provides higher computational accuracy and efficiency than the BLEAQ. Take the 10 × 10 SMD instances, for example, the average MADs at UL (LL) given by the HDEAB are 0.73% (4.04%) of those by the BLEAQ, and the average MFEs at UL (LL) given by the HDEAB are 6.2% (14.8%) of those by the BLEAQ.

5.3. An application example: optimal venture capital staged-financing contract

In this section, we consider an application example of BLOP where the entrepreneur (EN) and venture capitalist (VC) are entering a staged-financing contract.

In this example, the return on investment (ROI) of the start-up is r, which is a random variable, with μ and σ^2 being the mean and variance, respectively. EN is the leader who decides to invest an amount of the owner's capital (y) and the proportion of revenue shared with the VC's equity investment (γ). VC is the follower who invests the EN with a mixture of equity and debt in M stages for mitigating the risk of investment. VC's decision variables are the investment amount on equity in stage i ($x_{i,1}$, which brings the revenue $\gamma r x_{i,1}$), and the investment amount on debt in stage i ($x_{i,2}$, which brings the revenue $r_d x_{i,2}$; r_d is the interest rate). The VC's total amount of investment is $x = (x_{1,1} \ldots x_{M1}) + (x_{1,2} \ldots x_{M2})$.

The total revenues of EN and VC are given by

$$\begin{cases} R_{EN} = r(x+y) - \gamma r(x_{1,1} + \ldots + x_{M,1}) - r_d(x_{1,2} + \ldots + x_{M,2}) - d(y) \\ R_{VC} = \gamma r(x_{1,1} + \ldots + x_{M,1}) + r_d(x_{1,2} + \ldots + x_{M,2}) - c(x) \end{cases} \tag{16}$$

Table 2. MFEs at the UL and LL on small-scale SMD instances.

	MFEs at UL			MFEs at LL		
Instance	HDEAB	BLEAQ	Ratio	HDEAB	BLEAQ	Ratio
SMD1	273	1.19E + 03	0.23	33,248	2.37E + 05	0.14
SMD2	234	1.20E + 03	0.20	26,197	4.06E + 05	0.06
SMD3	274	1.29E + 03	0.21	31,659	2.83E + 05	0.11
SMD4	274	1.32E + 03	0.21	29,528	3.84E + 05	0.07
SMD5	253	2.06E + 03	0.12	33,492	8.42E + 05	0.04
SMD6	377	4.08E + 03	0.09	4408	6.04E + 03	0.73
SMD7	334	1.27E + 03	0.26	38,507	3.82E + 05	0.10
SMD8	484	3.54E + 03	0.14	65,202	1.73E + 06	0.04
SMD9	268	1.26E + 03	0.21	54,031	4.03E + 05	0.13
SMD10	718	1.92E + 03	0.37	190,314	5.45E + 05	0.35
SMD11	1237	2.39E + 03	0.51	301,179	4.63E + 06	0.07
SMD12	632	1.50E + 03	0.42	185,997	4.79E + 05	0.39

Source: Given by simulations.

Table 3. MADs at the UL and LL when the scale of the SMD instances varies.

		5×5		10×10		20×20	
Scale Instance		HDEAB	BLEAQ	HDEAB	BLEAQ	HDEAB	BLEAQ
SMD1	UL	1.29E-07	4.23E-03	3.12E-05	9.96E-03	6.12E-07	–
	LL	1.22E-07	4.14E-03	3.10E-05	4.01E-03	3.39E-06	–
SMD2	UL	1.58E-04	2.17E-03	9.34E-04	7.95E-03	1.69E-06	–
	LL	1.58E-04	3.05E-03	3.80E-04	5.20E-03	1.69E-06	–
SMD3	UL	6.38E-08	2.27E-06	4.76E-06	9.85E-03	4.64E-08	–
	LL	6.16E-08	2.79E-07	2.93E-05	4.49E-03	7.45E-08	–
SMD4	UL	3.93E-05	4.81E-03	6.12E-06	9.14E-03	4.32E-05	–
	LL	7.45E-08	5.95E-03	3.39E-05	3.29E-04	4.69E-05	
SMD5	UL	4.56E-08	2.94E-04	7.16E-05	6.27E-03	9.18E-12	–
	LL	3.66E-08	7.09E-04	7.25E-06	4.17E-03	3.97E-11	–
SMD6	UL	1.18E-11	1.37E-03	4.39E-08	5.15E-03	5.46E-09	–
	LL	7.24E-16	3.65E-16	6.95E-11	5.25E-11	2.32E-13	–
SMD7	UL	1.63E-06	9.03E-03	5.45E-04	5.23E-03	3.41E-07	–
	LL	1.68E-06	6.39E-05	6.97E-06	1.32E-03	1.98E-06	–
SMD8	UL	7.22E-05	1.37E-02	1.24E-03	4.23E-03	7.58E-05	–
	LL	3.93E-05	7.56E-03	9.64E-04	5.19E-03	7.84E-05	–

Source: Given by simulations.

where $d(y)$ and $c(x)$ are the EN and VC's opportunity costs, and $d'(\cdot)>0$, $d''(\cdot)>0$, $c'(\cdot)>0$ and $c''(\cdot)>0$.

Both EN's and VC's objectives (F_{EN} and F_{VC}) are given in Equation (17), which are the classical investment portfolio optimisations. Both players aim to maximise their revenues for given risks.

$$\begin{cases} \max F_{EN} = \beta_{EN} E[R_{EN}] + (1-\beta_{EN}) Var[R_{EN}] \\ \max F_{VC} = \beta_{VC} E[R_{VC}] + (1-\beta_{VC}) Var[R_{VC}] \end{cases} \tag{17}$$

where $E[\cdot]$ and $Var[\cdot]$ are the mean and variance operators, and β_{EN} and β_{VC} are the players' attitude factors on the revenues and risks. Theoretically, F_{EN} and F_{VC} reach their maximal values at the global optimums.

When x and y are unconstrained, Equation (17) can be solved analytically through backward induction, while it cannot be solved analytically when x and y are constrained. The scale of Equation (17) is $2M \times 2$. When the value of M increases, non-convexity rises easily.

Table 4. MFEs at the UL and LL when the scale of the SMD instances varies.

Scale Instance		5×5 HDEAB	5×5 BLEAQ	10×10 HDEAB	10×10 BLEAQ	20×20 HDEAB	20×20 BLEAQ
SMD1	UL	202	3208	774	202,144	2371	–
	LL	48,838	562,071	578,562	1,839,412	7,586,864	–
SMD2	UL	242	3812	699	181,833	2246	–
	LL	30,658	645,960	436,976	2,538,009	6,097,331	–
SMD3	UL	286	4876	859	216,891	3266	–
	LL	46,287	740,598	641,077	2,709,139	14,279,949	–
SMD4	UL	251	5412	721	262,246	3272	–
	LL	23,941	835,286	158,864	4,193,010	9,366,748	
SMD5	UL	272	8802	745	199,568	2524	–
	LL	48,321	785,518	569,842	5,735,669	8,739,080	–
SMD6	UL	373	10,222	1081	321,707	3643	–
	LL	6795	887,076	37,093	6,546,023	550,721	–
SMD7	UL	520	23,766	1257	2,312,093	4204	–
	LL	73,775	102,0613	808,414	1,047,442	12,240,037	–
SMD8	UL	972	63,434	3267	218,965	12,542	–
	LL	166,985	1,245,362	485,554	3,908,141	1.113E + 08	–

Source: Given by simulations.

Table 5. Performances when the scale of the application example varies.

Scale	HDEAB (MFEs) UL	HDEAB (MFEs) LL	BLEAQ (MFEs) UL	BLEAQ (MFEs) LL	$F_{VC}^{HDEAB}/F_{VC}^{BLEAQ}$	$F_{EN}^{HDEAB}/F_{EN}^{BLEAQ}$
10×2 ($M = 5$)	78	1462	138	7679	1.000	1.000
20×2 ($M = 10$)	241	30,238	1123	138,009	1.051	1.023
40×2 ($M = 20$)	956	88,039	6689	738,009	1.085	1.045
100×2 (M = 50)	1859	938,009	15,698	32,288,009	1.156	1.026

Source: Given by simulations.

To compare the performances of HDEAB and BLEAQ, we set the parameters as follows: $\mu = 10$, $\sigma = 5$, $r_d = 0.1$, $\beta_{EN} = \beta_{VC} = 0.5$, $c(x) = x^2$, and $d(y) = y^2$. M varies from 5 to 25. To save space, we only conduct numerical studies on the constrained case, where $x \in [0, 100]$ and $y \in [0, 10]$. Parameters for the HDEAB are the same as in Section 5.2.

Table 5 gives the numerical results, where $F_{VC}^{HDEAB}/F_{VC}^{BLEAQ}$ and $F_{EN}^{HDEAB}/F_{EN}^{BLEAQ}$ are the average ratios of objective values obtained by HDEAB and BLEAQ, respectively. For the application example, we have two findings:

i. Regardless of the scale, the HDEAB outperforms the BLEAQ greatly on the computational efficiencies on both levels.
ii. When the scale of the application example is small, two approaches provide the same computational precisions as the $F_{VC}^{HDEAB}/F_{VC}^{BLEAQ}$ and $F_{EN}^{HDEAB}/F_{EN}^{BLEAQ}$ equals one. However, The HDEAB provides higher values of F_{VC}^{HDEAB} and F_{EN}^{HDEAB} than the BLEAQ does when the scale increases. The reason is that the BLEAQ could easily fall into the local optimum due to the non-convexity introduced by a larger value of M and the constraints of x and y.

6. Conclusion

In this study, an efficient approach (HDEAB) was proposed to solve the BLOPs. In HDEAB: (i) the optimal lower level value function mapping method was adopted to provide higher computational accuracy; (ii) the KNN and a nested local search were hybridised to boost the computational efficiency; and (iii) the DEA was utilised as the optimisation engine for both levels. From the numerical studies on the SMD instances, the HDEAB demonstrated higher performances than the BLEAQ on both small- and large-scale SMD instances. Specifically, on the small-scale SMD instances (i.e., 2×3), the computational accuracies at upper and lower levels given by the HDEAB were 9.3 and 18.9 times higher than those by the BLEAQ while costing only 24.8% and 18.6% of the calls of MFEs required by the BLEAQ. Given the computational environment, the HDEAB could solve the largest scale SMD instances up to 20×20, while the BLEAQ could only solve 10×10 SMD instances in maximal. On 10×10 SMD instances, the HDEAB provided 137.4 and 24.8 times higher computational accuracy at upper and lower levels than the BLEAQ, while costing only 6.2% and 14.8% of the calls of MFEs required by the BLEAQ. By an application example in optimal VC staged-financing contract, the HDEAB outperformed the BLEAQ in providing higher objective values for both entrepreneurs and venture capitalists.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Angelo, J. S., Krempser, E., & Barbosa, H. J. C. (2013). Differential evolution for bilevel programming. IEEE Congress on Evolutionary Computation. doi:10.1109/CEC.2013.6557606

Angelo, J. S., Krempser, E., & Barbosa, H. J. C. (2014). Differential evolution assisted by a surrogate model for bilevel programming problems. IEEE Congress on Evolutionary Computation. doi:10.1109/CEC.2014.6900529

Benayed, O., & Blair, C. E. (1990). Computational difficulties of bilevel linear programming. *Operations Research*, *38*(3), 556–560. doi:10.1287/opre.38.3.556

Brajković, T., Pernić, M., & Ikonić, M. (2018). Production Planning and optimization of work launch orders using genetic algorithm. *Technical Gazette*, *25*(5), 1278–1285. doi:10.17559/TV-20161207195125

Brotcorne, L., Labbé, M., Marcotte, P., & Savard, G. (2001). A bilevel model for toll optimization on a multicommodity transportation network. *Transportation Science*, *35*(4), 345–358. doi:10.1287/trsc.35.4.345.10433

Calvete, H. I., Gale, C., & Mateo, P. M. (2008). A new approach for solving linear bilevel problems using genetic algorithms. *European Journal of Operational Research*, *188*(1), 14–28. doi:10.1016/j.ejor.2007.03.034

Chen, Y., & Florian, M. (1995). The nonlinear bilevel programming problem: Formulations, regularity and optimality conditions. *Optimization*, *32*(3), 193–209. doi:10.1080/02331939508844048

Colson, B., Marcotte, P., & Savard, G. (2005). A trust-region method for nonlinear bilevel programming: Algorithm and computational experience. *Computational Optimization and Applications*, *30*(3), 211–227. doi:10.1007/s10589-005-4612-4

Haghighat, H., & Kennedy, S. (2012). A bilevel approach to operational decision making of a distribution company in competitive environments. *IEEE Transactions on Power Systems*, *27*(4), 1797–1807. doi:10.1109/TPWRS.2011.2182214

Marcotte, P., Savard, G., & Zhu, D. (2001). A trust region algorithm for nonlinear bilevel programming. *Operations Research Letters*, *29*(4), 171–179. doi:10.1016/S0167-6377(01)00092-X

Mathieu, R., Pittard, L., & Anandalingam, G. (1994). Genetic algorithm based approach to bilevel linear programming. *Rairo - Operations Research*, *28*(1), 1–21. doi:10.1051/ro/1994280100011

Pérez-Rodríguez, R., Hernandez-Aguirre, A., & Jöns, S. (2017). A probability model for the school bus routing problem with bus stop selection. *Dyna (Bilbao)*, *92*(2), 138–142. doi:10.6036/8204

Savard, G., & Gauvin, J. (1994). The steepest descent direction for the nonlinear bilevel programming problem. *Operations Research Letters*, *15*(5), 265–282. doi:10.1016/0167-6377(94)90086-8

Sinha, A., Malo, P., & Deb, K. (2013). Efficient evolutionary algorithm for single-objective bilevel optimization. arXiv: Neural and Evolutionary Computing.

Sinha, A., Malo, P., & Deb, K. (2014). An improved bilevel evolutionary algorithm based on quadratic approximations. IEEE Congress on Evolutionary Computation. doi:10.1109/CEC.2014.6900391

Sinha, A., Malo, P., & Deb, K. (2014). Test problem construction for single-objective bilevel optimization. *Evolutionary Computation*, *22*(3), 439–477. doi:10.1162/EVCO_a_00116

Sinha, A., Malo, P., & Deb, K. (2015). Transportation policy formulation as a multi-objective bilevel optimization problem. IEEE Congress on Evolutionary Computation. doi:10.1109/CEC.2015.7257085

Sinha, A., Malo, P., & Deb, K. (2016). Solving optimistic bilevel programs by iteratively approximating lower level optimal value function. IEEE Congress on Evolutionary Computation. doi:10.1109/CEC.2016.7744017

Sinha, A., Malo, P., & Deb, K. (2017). Evolutionary algorithm for bilevel optimization using approximations of the lower level optimal solution mapping. *European Journal of Operational Research*, *257*(2), 395–411. doi:10.1016/j.ejor.2016.08.027

Sinha, A., Malo, P., & Deb, K. (2018). A review on bilevel optimization: From classical to evolutionary approaches and applications. *IEEE Transactions on Evolutionary Computation*, *22*(2), 276–295. doi:10.1109/TEVC.2017.2712906

Sinha, A., Malo, P., & Frantsev, A. (2013). Multi-objective Stackelberg game between a regulating authority and a mining company: A case study in environmental economics. IEEE Congress on Evolutionary Computation. doi:10.1109/CEC.2013.6557607

Tripathy, M., & Panda, A. (2017). A study of algorithm selection in data mining using meta-learning. *Journal of Engineering Science and Technology Review*, *10*(2), 51–64. doi:10.25103/jestr.102.06

Vicente, L., Savard, G., & Judice, J. (1994). Descent approaches for quadratic bilevel programming. *Journal of Optimization Theory and Applications*, *81*(2), 379–399. doi:10.1007/BF02191670

Wang, Y., Jiao, Y., & Li, H. (2005). 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), 221–232. doi:10.1109/TSMCC.2004.841908

Wang, Y., Li, H., & Dang, C. (2011). A new evolutionary algorithm for a class of nonlinear bilevel programming problems and its global convergence. *Informs Journal on Computing*, *23*(4), 618–629. doi:10.1287/ijoc.1100.0430

Wei, W., Liang, Y., Liu, F., Mei, S., & Tian, F. (2014). Taxing strategies for carbon emissions: A bilevel optimization approach. *Energies*, *7*(4), 2228–2245. doi:10.3390/en7042228

Wein, L. M. (2009). Homeland security: From mathematical models to policy implementation: The 2008 Philip McCord Morse Lecture. *Operations Research*, *54*(7), 801–811. doi:10.1287/opre.1090.0695

Whittaker, G., Färe, R., Grosskopf, S., Barnhart, B., Bostian, M., Mueller-Warrant, G., & Griffith, S. (2017). Spatial targeting of agri-environmental policy using bilevel evolutionary optimization. *Omega*, *66*, 15–27. doi:10.2139/ssrn.2497134

Yin, Y. (2000). Genetic-algorithms-based approach for bilevel programming models. *Journal of Transportation Engineering*, *126*(2), 115–120. doi:10.1061/(ASCE)0733-947X(2000)126:2(115)

Zhang, W., Wen, J. B., Zhu, Y. C., & Hu, Y. (2017). Multi-objective scheduling simulation of flexible job-shop based on multi-population genetic algorithm. *International Journal of Simulation Modelling*, *16*(2), 313–321. doi:10.2507/IJSIMM16(2)CO6