CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

| | |
|---|---|
| Title | A fully Bayesian approach to bilevel problems |
| Authors | Dogan, Vedat;Prestwich, Steven;O'Sullivan, Barry |
| Publication date | 2024-10-16 |
| Original Citation | Dogan, V., Prestwich, S. and O'Sullivan, B. (2024) 'A fully Bayesian approach to bilevel problems', in Freeman, R. and Mattei, N. (eds) Algorithmic Decision Theory. ADT 2024. Lecture Notes in Computer Science, 15248, pp. 144-159. Springer, Cham. https://doi.org/10.1007/978-3-031-73903-3_10 |
| Type of publication | Article (peer-reviewed);Book chapter |
| Link to publisher's version | 10.1007/978-3-031-73903-3_10 |
| Rights | © 2024 The Author(s). Version of Record published under exclusive license to Springer Nature Switzerland. For Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. - https://creativecommons.org/licenses/by/4.0/ |
| Download date | 2026-11-14 03:37:59 |
| Item downloaded from | https://hdl.handle.net/10468/16681 |



UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh



# A Fully Bayesian Approach to Bilevel Problems

Vedat Dogan(✉), Steven Prestwich, and Barry O'Sullivan

Insight SFI Research Centre for Data Analytics, School of Computer Science and IT, University College Cork, Cork, Ireland
{vedat.dogan,steven.prestwich,barry.osullivan}@insight-centre.org

**Abstract.** The mathematical models of many real-world decision-making problems contain two levels of optimization. In these models, one of the optimization problems appears as a constraint of the other one, called follower and leader, respectively. These problems are known as bilevel optimization problems (BOPs) in mathematical programming and are widely studied by both classical and evolutionary optimization communities. The nested nature of these problems causes many difficulties such as non-convexity and disconnectedness for traditional methods, and requires a huge number of function evaluations for evolutionary algorithms. This paper proposes a fully Bayesian optimization approach, called FB-BLO. We aim to reduce the necessary function evaluations for both upper and lower level problems by iteratively approximating promising solutions with Gaussian process surrogate models at both levels. The proposed FB-BLO algorithm uses the other decision-makers' observations in its Gaussian process model to leverage the correlation between decisions and objective values. This allows us to extract knowledge from previous decisions for each level. The algorithm has been evaluated on numerous benchmark problems and compared with existing state-of-the-art algorithms. Our evaluation demonstrates the success of our proposed FB-BLO algorithm in terms of both effectiveness and efficiency.

**Keywords:** Bilevel Decision-Making · Bayesian Optimization · Gaussian Process · Stackelberg Games

## 1 Introduction

The hierarchical and nested nature of real-world decision-making problems has increased the interest of researchers in bilevel optimization. Various applications in different domains are implemented as bilevel optimization problems (BOPs). One of the recent applications is in the domain of operational research for solving the retailer-consumer problem considering multiple strategic retailer involvement and customers' switching behaviours [19]. Another application of BOPs is for modelling the multiple decision-makers behaviour. For instance, in the context of homeland security the objective is to ensure that the country is secure, safe

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
R. Freeman and N. Mattei (Eds.): ADT 2024, LNAI 15248, pp. 144–159, 2025.
https://doi.org/10.1007/978-3-031-73903-3_10

and resilient against terrorism and other possible attacks. Therefore, a government needs to model its own decision-making model to prevent terrorist attacks, considering cost efficiency without causing any harm to the public during operations. The government also needs to take into account the potential outcome of its decisions, such as terrorists' possible reactions after the operation. So, the nested decision-making problem must be modelled considering both levels' objectives and constraints for ease of government. In these kinds of applications, one decision-maker acts as a leader and the other one acts as a follower where follower objectives and constraints appear as a constraint of leaders' objective problems. The government acts as the leader in our example but it could be the contrary in some situations. Mathematically BOPs can be formulated as follows:

$$\begin{aligned} \underset{x_u, x_l}{\text{Minimize}} \;\; & F(x_u, x_l') \\ \text{s.t.} \;\; & x_l' \in \underset{x_l}{\operatorname{argmin}} \{ f(x_u, x_l) : g_j(x_u, x_l) \leq 0, \; j = 1, 2, \ldots, J \} \\ & G_k(x_u, x_l') \leq 0, \; k = 1, 2, \ldots, K \end{aligned} \tag{1}$$

where $x_u$ and $x_l'$ represent the upper level (leader) decisions and optimal lower level (follower) reactions. $F$ and $f$ represent the leader's and follower's objective values, respectively. $G$ and $g$ are leader's and follower's constraints. As we can see from Eq. 1, the follower's optimization problem is nested within the leader's. This structure causes computational challenges compared with single-level optimization problems. For instance, it is proven that BOPs are NP-hard even with linear objective functions [22]. Also, it is difficult to ensure lower level optimality while solving the bilevel problem, which is generally vital as non-optimal lower level solutions may mislead the algorithm. Further, the necessary function evaluation is expected to be extremely high as each upper level solution requires solving the lower level optimization problem. Such problems are relevant to different domains such as environmental economics literature [40] which commonly appears as Stackelberg games in Game Theory [33]. Over the years, many applications have been modelled as BOPs such as machine learning [43], supply chain management [42], security [15], defence industry [17], etc.

BOPs have been widely investigated over the years by mathematical programming literature. For instance, BOPs are simplified using Karush-Kuhn-Tucker (KKT) conditions [3] to reduce the two levels into a single level and solve the resulting problem using standard methods. The disadvantage of these kind of classical approaches is that they require strong assumptions such as smoothness, nonlinearity or convexity of one or both levels. However, these assumptions do not always correspond to real-world problems. For this reason the literature has focused on evolutionary algorithms (EA) to solve BOPs [28] due to their ability to handle the discreteness and nonlinearity of the optimization problems at both levels [31]. Nevertheless, EAs require a large number of function evaluations because of their nature of population-based search strategies to obtain a competitive result. To tackle this problem, EAs have been improved with new technologies that have developed over the years [18,26]. Surrogate-assisted EAs are developed to reduce the number of function evaluations required by EAs and

they achieved competitive performance for solving BOPs [6,37]. The majority of them focused on modelling the lower level problem to decrease the necessary evaluation, such as [8]. The authors proposed a method to model with surrogate-assisted EAs for both levels in [20] and they achieved competitive results compared with others.

Even though both proposed EAs and surrogate-assisted EAs at both levels decrease the computational cost, the nature of the population-based search strategy still requires a high number of function evaluations. Regarding that, a few Bayesian approach-based black-box optimization algorithms have been applied to solve BOPs. In [21], standard Bayesian optimization (BO) is used to solve upper level problems and EA is used to solve lower level problems. However, they only used the upper level decisions and objective values to train the upper level model. [12] improved the algorithm by a conditional BO approach to decrease the computational cost by conditioning lower level decisions while approximating upper level objectives. Even though they both showed that the necessary function evaluation decreased while approximating the bilevel global optima, the computational cost at the lower level was still high because of the population-based search strategy at the lower level. To this end, in this paper we propose an algorithm that uses a fully Bayesian optimization approach to solve bilevel optimization problems, called FB-BLO. In contrast to the proposed black-box approaches methods above, we developed a Gaussian process ($\mathcal{GP}$) based surrogate model for both upper and lower level optimization problems. The combination of using $\mathcal{GP}$ models at both upper and lower levels and applying BO significantly reduces the necessary function evaluations and improves the optimization performance. The FB-BLO algorithm is analysed and compared to the well-established algorithm in the literature. Our contributions are as follows:

- The proposed FB-BLO method for bilevel optimization models both upper and lower level problems as black-box. It eliminates the necessity of assumptions generally required by classical approaches and reduces the necessary function evaluations which are high in general by population-based search strategies.
- Assuming both upper and lower level optimization problems as black-box and modelling by a surrogate-assisted model makes the proposed approach promising when upper level objective function $F$ or lower level objective function $f$ or both are unavailable, unreliable or impractical to obtain.
- We also provide the theoretical development of the proposed FB-BLO algorithm with competitive results in terms of accuracy and function evaluations after the experiment by two popular benchmarks for BOPs in the literature.

The rest of this paper is organized as follows. In Sect. 2, we provide a brief literature review by analysing existing methods for solving BOPs and the motivation of this study. Then in Sect. 3, we propose the FB-BLO algorithm and provide the theoretical details. Empirical studies are presented and discussed with results in Sect. 4. Finally, Sect. 5 concludes this article by discussing limitations and future works.

## 2 Related Work and Motivation

Over the years, many approaches have been developed to tackle BOPs. H. von Stackelberg introduced Stackelberg games [33] connected to Game Theory in 1952 in mathematical programming literature, and the first formulation was proposed in [4] by J. Bracken and J. McGill in 1973. They also proposed the defence application formulation in the following year [5] and used "mathematical programs with optimization problems in the constraints" instead of bilevel problems. BOPs were modelled as mathematical programs at this time and are difficult to handle mathematically because of the hierarchical optimization structure. It may introduce difficulties such as non-convexity and disconnectedness between the upper level and lower level problems, even for simple instances. It has been shown that bilevel programming is strongly NP-hard [16], and it has been proven that just evaluating a solution is also an NP-hard task [35].

Even though exact approaches are not efficient and require strong assumptions, they are the main focus in the bilevel optimization literature. One of the oldest is reformulation from bilevel to single level by using KKT conditions [3]. This has been widely used to solve linear [38] and quadratic [13] bilevel problems. The other classical approaches to BOPs are gradient descent [35], trust region [9] and penalty methods [39]. Because of the necessity of strong assumptions for applying classical approaches, the literature focused on EAs to solve BOPs, and the first EA proposed to solve BOPs was in [41]. They used EAs to solve the upper level and classical approaches for solving the lower level problem. Later, several works applied KKT conditions to reduce the bilevel problem into a single level and apply EAs to solve the reduced version of the problem [32]. Several nested algorithms were also proposed which use EAs at both levels with different approaches. For instance [1] applied a differential evolution algorithm at both levels and [23] implemented a particle swarm optimization approach to solve BOPs. EAs reached competitive results compared with classical approaches, but the required number of function evaluations is still huge because of the nature of the population-based search strategy. Meta-modelling approaches have been developed over the years to handle this problem [29]. This is a commonly used method for expensive-to-evaluate optimization problems [36]. It uses a surrogate model to approximate the actual model to reduce the function evaluations required to solve the actual problem. Even though using a surrogate model for the lower level problem decreases the function evaluation significantly, still EAs at the upper level require a high number of function evaluations to achieve competitive results. For further details about theoretical developments of classical and evolutionary algorithms for BOPs and more applications, the readers may refer to [7,31].

Recently, [21] proposed a nested approach to decrease function evaluations by approximating the optimal values. They applied a general Bayesian approach at the upper level by using upper level decisions and objective values, and an evolutionary algorithm to solve the lower level problem. [11] presented a multi-objective acquisition optimization approach to improve BO at the upper level. Following that, [12] presented a conditional BO-based approach to solve the

bilevel problems by conditioning lower level decisions at the upper level, and they presented competitive results. They decreased the upper level function evaluations significantly, but it was still high at the lower level.

Based on the above observations, we develop a fully BO algorithm for BOPs, called FB-BLO. Unlike existing approaches, the FB-BLO algorithm uses $\mathcal{GP}$ surrogate model for both upper and lower levels to decrease at both upper and lower level function evaluations while approximating the optimal objective values. It selects a small number of candidates where the uncertainty is high during the optimization at both levels by the BO process and uses both decisions and its own objective values to model. Therefore, the number of necessary function evaluations can be reduced significantly while ensuring approximation to the global optima. Because both levels' surrogate models use both levels' decisions, the upper and lower level models cooperate to improve the optimization process. Moreover, the proposed FB-BLO algorithm can be implemented in several fields where the objective functions are expensive to evaluate, unavailable and impractical to model because of complexity or other reasons. It makes it easier to model practical problems as the proposed algorithm works only on decisions and the outcomes of the decisions, without requiring function characteristics such as nonconvexity, differentiability, etc.

## 3 Proposed Method

In this section, we first briefly describe BO and $\mathcal{GP}$, and then we propose the main steps of the FB-BLO by explaining how upper and lower level optimizations are implemented with the Bayesian method.

BO is a global optimization method to optimize expensive-to-evaluate black-box functions. It is guided by a statistical model of the objective function, which is called a surrogate model, generally the $\mathcal{GP}$ [14] is selected. The $\mathcal{GP}$ model takes a prior distribution to be a multivariate normal with a mean function and covariance function, represented by $\mu_0(\cdot)$ and $\sigma_0^2(\cdot)$ respectively. Let us assume that we have $n$ decisions $x_{1:n} = \{x_i\}_{i=0}^n$ and observations $f(x_{1:n}) = \{f(x_i)\}_{i=0}^n$, then the $\mathcal{GP}$ posterior distribution is calculated as in Eq. 2;

$$
\begin{aligned}
f(x')|f(x_{1:n}) &\sim \mathcal{N}(\mu_n(x'), \sigma_n^2(x')) \\
\mu_n(x') &= \Sigma_0(x', x_{1:n})\Sigma_0(x_{1:n}, x_{1:n})^{-1}(f(x_{1:n}) - \mu_0(x_{1:n}))) + \mu_0(x') \quad (2) \\
\sigma_n^2(x') &= \Sigma_0(x', x') - \Sigma_0(x', x_{1:n})\Sigma_0(x_{1:n}, x_{1:n})^{-1}\Sigma_0(x_{1:n}, x')
\end{aligned}
$$

where $x'$ is a new candidate and $\Sigma_0(\cdot, \cdot)$ is a covariance matrix. Because it is expensive to evaluate the true objective function, the $\mathcal{GP}$ uses acquisition functions which are much cheaper and easier to optimize. Acquisition functions are designed to evaluate the trade-off between exploration and exploitation of search space and the current promising area. If we represent the acquisition function by $\alpha(\cdot)$ then the $\mathcal{GP}$ selects the next candidate by optimizing the predefined acquisition function, $x_{n+1} = \arg\max \alpha_n(x)$. For further details, the readers may refer to [25].

**Algorithm 1:** Procedure of the FB-BLO Algorithm

**Input:** $\{x_{u_i}\}_{i=0}^{n}$ (the upper level random decision vectors with the size of $n$)
**Output:** $x_u^{best}, x_l^{best}, F^{best}, f^{best}$ (the best objective values and decision set)

```
/* Initialization of the Algorithm */
```

Set the bilevel data set $\mathcal{D} = \emptyset$;
Conduct lower level optimization for each upper level decision;
Evaluate the objective function results for each decision pair;
Add the decisions and related solutions to the bilevel dataset $\mathcal{D}$;
Initialize the bilevel dataset $\mathcal{D} = \{x_{u_i}, x_{l_i}, F_i(x_{u_i}, x_{l_i}), f_i(x_{u_i}, x_{l_i})\}_{i=0}^{n}$ with random decision vectors and related objective values;
**while** *termination criteria not fulfilled* **do**
    `/* upper level Optimization */`
    $\mathcal{GP}_u \leftarrow$ Built upper level Gaussian process model based on all different *upper level* decisions and related upper level and lower level objective values in $\mathcal{D}$;
    $\mathbf{X_u^*} \leftarrow$ Select the best upper level decision candidate $x_u^*$ by evaluating upper level acquisition optimization;
    **for** $x_u^* \in \mathbf{X_u^*}$ **do**
        `/* lower level Optimization */`
        $\mathcal{GP}_l \leftarrow$ Built lower level Gaussian process model based on *lower level* solutions and upper level and lower level objective values in $\mathcal{D}$;
        $\mathbf{X_l^*} \leftarrow$ Select the best lower level decision candidate $x_l^*$ by evaluating lower level acquisition optimization;
    Evaluate upper level $F(\mathbf{X_u^*}, \mathbf{X_l^*})$ and lower level $f(\mathbf{X_u^*}, \mathbf{X_l^*})$ objective results;
    Insert solutions in into bilevel dataset $\mathcal{D}$;
    Update upper level and lower level $\mathcal{GP}$ models with new observations;
**return** $x_u^{best}, x_l^{best}, F^{best}, f^{best}$ (the best objective values and decision set) in the bilevel dataset $\mathcal{D}$;

We present the main steps of the proposed FB-BLO in Algorithm 1 and share the flowchart in Fig. 1. The FB-BLO first generates $n$ upper level decision samples $\{x_{u_i}\}_{i=0}^{n}$ using the Latin hypercube sampling [34] to begin. After that, the FB-BLO performs the lower level optimization by BO and uses the $\mathcal{GP}$ as a surrogate model for each initialized upper level decision. Consecutively, the FB-BLO evaluates the upper and lower level objective values and updates the data set with objective values along with optimal lower level decision $x_l^*$ considering upper level value $x_u$. During upper level optimization, the proposed algorithm initializes the upper level Gaussian process model $\mathcal{GP}_u$ with upper and lower level decision vectors, and upper level objective values $x_u, x_l^*$ and $F(x_u, x_l^*)$. Therefore, the $\mathcal{GP}_u$ is built based on the previous observations to search the following decision candidate for the upper level optimization problem. For each upper level decision candidate, the FB-BLO performs the lower level optimization to approximate the optimal value. During lower level optimization, a $\mathcal{GP}$ model is trained to approximate lower level problem, called $\mathcal{GP}_l$, with both decision vectors and lower level objective values $x_u, x_l^*$ and $f(x_u, x_l^*)$. By

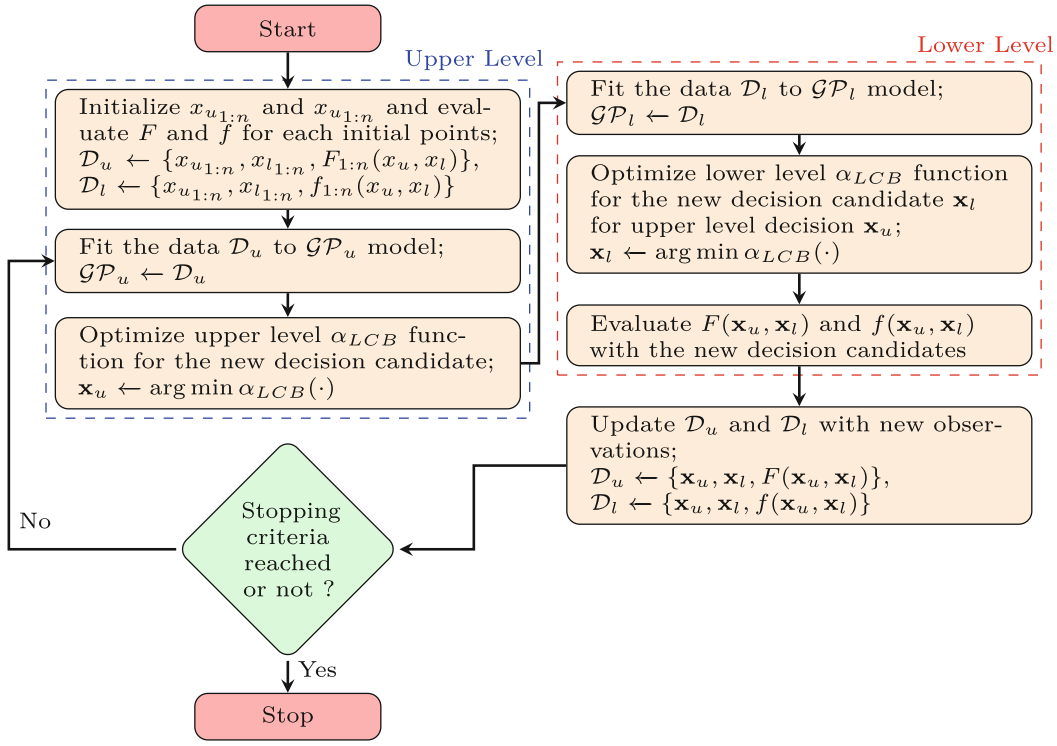



**Fig. 1.** Flowchart of the FB-BLO

using both decision vectors of the upper and lower level optimization problems to train $\mathcal{GP}$ models, the algorithm takes leverage to use the correlation between decisions at both levels and objective values to converge faster and to perform a smarter search. After lower level optimization, the FB-BLO adds the obtained results to the data set to update upper and lower level $\mathcal{GP}$ models.

We can observe that during upper level optimization, the FB-BLO algorithm selects the promising candidate by approximating the problem considering the previous observations. Then, it approximates the lower level problem considering the upper level decision variable. Using the BO approach and approximating both levels decreases the number of function evaluations significantly while converging the optimal value at both levels.

As we described above, the FB-BLO algorithm creates a Gaussian process surrogate model and it performs BO to select the most promising candidate during the upper level optimization, called $\mathcal{GP}_u$. Assume that we have the bilevel data set $\mathcal{D} = \{x_{u_i}, x^*_{l_i}, F_i(x_{u_i}, x^*_{l_i})\}_{i=0}^n$ contains $n$ observations and upper level objective value where $x_{u_i}$ is upper level decision vector and $x^*_{l_i}$ best lower level responses. Then, the FB-BLO algorithm initializes the upper level surrogate model $\mathcal{GP}_u$ with a prior mean function and variance defined by $\mu_u$ and $\sigma_u^2$ respectively. The standard deviation can also be used to measure the uncertainty of predicted results. Let us assume that the decision vector represented by $\mathbf{x_i} = \{(x_{u_i}, x^*_{l_i})\}_{i=0}^n$. After that, we construct the mean vector by calculating $\mu_{0_u}(\cdot)$ at each $\mathbf{x}_i$ and covariance matrix by calculating $\Sigma_{0_u}$ at each pair of decision points

$\mathbf{x}_i$ and $\mathbf{x}_j$. Then, the posterior mean and variance are formulated as follows:

$$\begin{aligned} F(\mathbf{x}')|F(\mathbf{x}_{1:n}) &\sim \mathcal{GP}_u(\mu_{n_u}(\mathbf{x}'), \sigma^2_{n_u}(\mathbf{x}')) \\ \mu_{n_u}(\mathbf{x}') &= \Sigma_0(\mathbf{x}', \mathbf{x}_{1:n})\Sigma_0(\mathbf{x}_{1:n}, \mathbf{x}_{1:n})^{-1}(F(\mathbf{x}_{1:n}) - \mu_0(\mathbf{x}_{1:n}))) + \mu_0(\mathbf{x}') \\ \sigma^2_{n_u}(\mathbf{x}') &= \Sigma_0(\mathbf{x}', \mathbf{x}') - \Sigma_0(\mathbf{x}', \mathbf{x}_{1:n})\Sigma_0(\mathbf{x}_{1:n}, \mathbf{x}_{1:n})^{-1}\Sigma_0(\mathbf{x}_{1:n}, \mathbf{x}') \end{aligned} \tag{3}$$

where $F(\cdot)$ is the upper level objective function and $\mathbf{x}'$ is the next candidate to be estimated at the upper level. Then the algorithm obtains the next promising point by minimizing the lower confidence bound (LCB) acquisition function [10], $\mathbf{x}' = \arg\min \alpha_{LCB}(\mathbf{x})$, as defined in Eq. 4.

$$\alpha_{LCB}(\mathbf{x}') = \mu_{n_u}(\mathbf{x}') - \beta^{\frac{1}{2}}\sigma_{n_u}(\mathbf{x}') \tag{4}$$

The FB-BLO trains the $\mathcal{GP}_u$ model based on the entire dataset $\mathcal{D}$. So the upper level model in FB-BLO considers upper and lower level decision variables as inputs to predict the upper level objective values. Therefore, it initializes the $\mathcal{GP}_u$ to approximate the relationship between $x_u, x_l^*$ and $F(x_u, x_l^*)$. In this way, the proposed algorithm can leverage the correlation of both upper and lower level decision sets with the upper level objective values. Then it optimizes the predefined acquisition function which is much cheaper in terms of computation than the true objective value, then selects the next promising candidate. It dramatically reduces the necessary number of function evaluations at the upper level. The optimal previous lower level objective value could also be used to train $\mathcal{GP}_u$ but it could affect the efficiency.

Similar upper level solutions likely lead to similar lower level optima as indicated in [30]. Considering this fact, for lower level optimization problems, FB-BLO initializes another Gaussian process surrogate model called $\mathcal{GP}_l$. The main reason behind this idea is selecting the decision candidates from an area in the search space that has the highest uncertainty. During the lower level optimization, considering a given upper level decision variable, the FB-BLO first creates the model with previous observations $x_u, x_l^*$ and lower level objective $f(x_u, x_l^*)$. Then, the $\mathcal{GP}_l$ model uses $x_u, x_l^*$ and $f(x_u, x_l^*)$ to approximate the lower level objective value. Subsequently, for each upper level decision, the proposed algorithm uses the $\mathcal{GP}_l$ model to select lower level optimum $x_l^{'}$ then updates the bilevel dataset with the set of decisions along with objective values $x_u, x_l^{'}, F(x_u, x_l^{'})$ and $f(x_u, x_l^{'})$ respectively. It should be noted that both $\mathcal{GP}_u$ and $\mathcal{GP}_l$ share the same bilevel data set for building the model. The only difference is that each model uses its own objective values to predict. Also, note that the proposed FB-BLO algorithm is developed for optimistic BOPs and tested only with continuous variables.

## 4 Empirical Studies

This section presents empirical results and compares the FB-BLO algorithm with six other algorithms. First, we briefly describe the compared algorithms along

with the test problems. Then, we share the parameter settings for the experiments and performance metrics respectively. We also assess the performance of the proposed FB-BLO algorithm to verify the effectiveness and efficiency of both $\mathcal{GP}$ models and investigate the performance on selected benchmark problems.

### 4.1 Experimental Settings

The performance of the proposed algorithm is compared with six state-of-the-art algorithms selected from the bilevel optimization literature. These are ConBaBo [12], BOBP [21], MOTEA [8], BL-CMA-ES [18] and BLEAQ2 [26]. The results for the comparison are extracted from the related references in the literature. The ConBaBo uses a conditional Bayesian approach for solving upper level problems during the optimization process. An evolutionary algorithm is implemented to solve the lower level problems and they focused on decreasing function evaluations by surrogating *only* upper level optimization problems by conditioning lower level decisions. The BOBP algorithm applies a standard Bayesian approach to solve bilevel problems without conditioning and they used only upper level variables and objective values to solve the upper level problems. The MOTEA uses a multi-objective transformation-based evolutionary algorithm to solve multiple lower level optimization in a parallel and collaborative way. In the BL-CMA-ES algorithm, a bilevel covariance matrix adaptation is used to solve BOPs and a specific sharing mechanism is developed to use prior knowledge for lower level problems. The NBLEA algorithm is designed as a nested algorithm that solves both level problems with genetic algorithms. The BLEAQ2 algorithm uses a quadratic approximation to obtain both levels' optimum values to reduce the number of necessary function evaluations.

We evaluate the performance of the proposed FB-BLO algorithm on sixteen popular benchmark problems including linear, nonlinear, constrained and unconstrained minimization problems, called TP [28] and SMD [27] problems. Each problem has different convexity considering the interaction between the upper level and lower level decision-makers. The TP benchmark contains one three-dimensional, six four-dimensional, one five-dimensional and two ten-dimensional bilevel problems, and the SMD benchmark has six unconstrained bilevel problems. For more details about the benchmarks and the mathematical formulations, please see Sect. 1 in the supplementary material.

All experiments were conducted on a single core of 1.4 GHz Quad Core i5, 8 Gb 2133 Mhz LPDDR3 RAM. The algorithm is executed 31 times for each test function and the results shown are medians. The termination criteria are selected as the accuracy difference $\epsilon = 10^{-6}$ for each level and for not converging for a certain iteration, the closest accuracy and step are chosen as a result. Later, the best decision is selected after convergence. All experiments were conducted on a single core of 1.4 GHz Quad BO and $\mathcal{GP}$ are implemented in Python via the GPyOpt library [2] and *optimize_restarts* selected as 10 with the parameters of *verbose=False*, and *exact_feval=True*. The method initialized $n$ Sobol points to construct the initial GP model. The LCB acquisition function has been selected for the $\mathcal{GP}$ and the Radial Basis Function (RBF) kernel is used.

**Table 1.** Median accuracy table for both TP and SMD benchmarks.

| | | FB-BLO | CONBABO | BOBP | MOTEA | BL-CMA-ES | NBLEA | BLEAQ2 |
|---|---|---|---|---|---|---|---|---|
| TP1 | ULFAcc | 0.465 | 1.02 | 23.6 | 0.185 | 0.000001 | 0.426 | 0.000002 |
| | LLFAcc | 0.277 | 1.99 | 29.6 | 0.298 | 0.00002 | 0.702 | 0.000001 |
| TP2 | ULFAcc | 0.000032 | 0.000001 | 0.0007 | 0.000227 | 0.000001 | 0.000001 | 0.000001 |
| | LLFAcc | 10.0 | 10.0 | 83.8 | 0.00643 | 0.00113 | 100.0 | 100.0 |
| TP3 | ULFAcc | 0.000124 | 0.663 | 0.120 | 1.011 | 0.364 | 0.00652 | 0.000011 |
| | LLFAcc | 0.000417 | 0.0001 | 0.663 | 0.00216 | 0.219 | 0.00132 | 0.000025 |
| TP4 | ULFAcc | 0.000016 | 1.76 | 1.57 | 0.000001 | 0.000001 | 0.293 | 0.000008 |
| | LLFAcc | 0.00002 | 0.528 | 0.101 | 0.000001 | 0.000001 | 0.00568 | 0.000001 |
| TP5 | ULFAcc | 0.944 | 0.710 | 0.251 | 0.854 | 1.12 | 0.000349 | 0.00571 |
| | LLFAcc | 0.214 | 0.445 | 0.231 | 0.877 | 5.45 | 0.00140 | 0.265 |
| TP6 | ULFAcc | 0.000016 | 0.132 | 0.0004 | 0.000001 | 0.000001 | 0.000006 | 0.00544 |
| | LLFAcc | 0.000045 | 0.606 | 0.00230 | 0.000026 | 0.000028 | 0.0000532 | 0.193 |
| TP7 | ULFAcc | 0.000262 | 0.6 | 0.285 | 0.000679 | 0.00066 | 0.00157 | 0.00136 |
| | LLFAcc | 0.000262 | 0.6 | 0.285 | 0.000679 | 0.00066 | 0.00157 | 0.00136 |
| TP8 | ULFAcc | 0.000001 | 0.000001 | 0.0008 | 0.000177 | 0.000002 | 0.000001 | 0.00254 |
| | LLFAcc | 10.0 | 10.0 | 80.6 | 0.00245 | 0.000406 | 100.0 | 100.0 |
| TP9 | ULFAcc | 0.000001 | 0.000001 | 0.000001 | 0.000001 | 0.000001 | 0.000032 | 0.000007 |
| | LLFAcc | 0.000001 | 0.000001 | 0.000001 | 0.000001 | 0.000001 | 0.000001 | 0.000001 |
| TP10 | ULFAcc | 0.000001 | 0.000001 | 0.000001 | 0.000001 | 0.000001 | 3.02 | 0.00186 |
| | LLFAcc | 0.000001 | 0.000001 | 0.000001 | 0.000001 | 0.000001 | 0.000001 | 0.000001 |
| SMD1 | ULFAcc | 0.000059 | 0.0000018 | – | 0.000001 | 0.000001 | 0.000033 | 0.00018 |
| | LLFAcc | 0.000001 | 0.000001 | – | 0.000001 | 0.000001 | 0.000015 | 0.00018 |
| SMD2 | ULFAcc | 0.000038 | 0.000009 | – | 0.000001 | 0.000001 | 0.000021 | 0.000119 |
| | LLFAcc | 0.000001 | 0.000001 | – | 0.000001 | 0.000001 | 0.000013 | 0.000117 |
| SMD3 | ULFAcc | 0.000063 | 0.000002 | – | 0.000001 | 0.000001 | 0.000043 | 0.00647 |
| | LLFAcc | 0.000001 | 0.000001 | – | 0.000001 | 0.000001 | 0.000014 | 0.00297 |
| SMD4 | ULFAcc | 0.000027 | 0.000006 | – | 0.000001 | 0.000001 | 0.000016 | 0.000024 |
| | LLFAcc | 0.000001 | 0.000001 | – | 0.000004 | 0.000005 | 0.000011 | 0.000333 |
| SMD5 | ULFAcc | 0.000003 | 0.000003 | – | 0.000001 | 0.000001 | 0.000039 | 0.000724 |
| | LLFAcc | 0.000001 | 0.000001 | – | 0.000002 | 0.000002 | 0.000014 | 0.000657 |
| SMD6 | ULFAcc | 0.000007 | 0.000002 | – | 0.000001 | 0.000001 | 0.000653 | 0.000001 |
| | LLFAcc | 0.000001 | 0.000001 | – | 0.000001 | 0.000001 | 0.000126 | 0.000001 |

For optimizing the acquisition function, the L-BFGS method [24] has been used. The exploration-exploitation parameter for LCB, w is set to 2.

### 4.2 Performance Comparison

Table 1 shows the accuracy of the different algorithms on both TP and SMD benchmarks. For BOPs, the absolute difference between the true optimum value

**Table 2.** Median function evaluation table for both TP and SMD benchmarks.

| | | FB-BLO | CONBABO | BOBP | MOTEA | BL-CMA-ES | NBLEA | BLEAQ2 |
|---|---|---|---|---|---|---|---|---|
| TP1 | ULFEval | 33 | 18 | 211 | 320 | 214 | 523 | 136 |
| | LLFEval | 64 | 190 | 1558 | 2265 | 2084 | 2568 | 242 |
| TP2 | ULFEval | 28 | 17 | 35 | 293 | 224 | 763 | 255 |
| | LLFEval | 52 | 148 | 383 | 2186 | 1986 | 4237 | 440 |
| TP3 | ULFEval | 61 | 10 | 89 | 307 | 333 | 468 | 158 |
| | LLFEval | 88 | 111 | 1128 | 1838 | 2124 | 5472 | 224 |
| TP4 | ULFEval | 54 | 14 | 16 | 294 | 367 | 746 | 198 |
| | LLFEval | 92 | 1063 | 334 | 1733 | 2165 | 8649 | 788 |
| TP5 | ULFEval | 44 | 23 | 57 | 339 | 430 | 1021 | 272 |
| | LLFEval | 80 | 222 | 319 | 2135 | 2406 | 9842 | 967 |
| TP6 | ULFEval | 43 | 8 | 12 | 414 | 330 | 624 | 161 |
| | LLFEval | 71 | 323 | 182 | 19300 | 2178 | 6736 | 323 |
| TP7 | ULFEval | 57 | 13 | 72 | 356 | 134 | 748 | 112 |
| | LLFEval | 99 | 274 | 320 | 4608 | 6445 | 5532 | 287 |
| TP8 | ULFEval | 20 | 14 | 37 | 134 | 314 | 364 | 241 |
| | LLFEval | 42 | 114 | 413 | 2389 | 2018 | 9638 | 467 |
| TP9 | ULFEval | 36 | 8 | 16 | 313 | 2508 | 2262 | 1512 |
| | LLFEval | 128 | 1020 | 396 | 13800 | 12800 | 32460 | 14130 |
| TP10 | ULFEval | 42 | 15 | 21 | 187 | 9122 | 2458 | 1847 |
| | LLFEval | 264 | 3689 | 974 | 24400 | 46410 | 4868 | 24510 |
| SMD1 | ULFEval | 65 | 11 | – | 3200 | 321 | 839 | 266 |
| | LLFEval | 162 | 380 | – | 22600 | 20800 | 546000 | 27500 |
| SMD2 | ULFEval | 73 | 14 | – | 293 | 297 | 1010 | 290 |
| | LLFEval | 174 | 381 | – | 21800 | 19800 | 610000 | 25100 |
| SMD3 | ULFEval | 53 | 23 | – | 307 | 333 | 938 | 311 |
| | LLFEval | 187 | 770 | – | 18300 | 21200 | 589000 | 35700 |
| SMD4 | ULFEval | 73 | 11 | – | 294 | 334 | 1350 | 410 |
| | LLFEval | 168 | 258 | – | 17300 | 21800 | 604000 | 26400 |
| SMD5 | ULFEval | 61 | 24 | – | 339 | 367 | 1040 | 286 |
| | LLFEval | 139 | 905 | – | 21300 | 21600 | 584000 | 36200 |
| SMD6 | ULFEval | 82 | 25 | – | 414 | 430 | 1310 | 274 |
| | LLFEval | 197 | 769 | – | 21000 | 24000 | 36700 | 92900 |

and the best-obtained value by the proposed algorithm is used for accuracy comparison. The upper and lower level accuracy, called ULAcc and LLAcc respectively, can be calculated by $ULAcc = |F^{best}F^{*}|$ and $LLAcc = |f^{best}f^{*}|$ where $F^{*}$ and $f^{*}$ represent the true optimum value of upper and lower level optimization

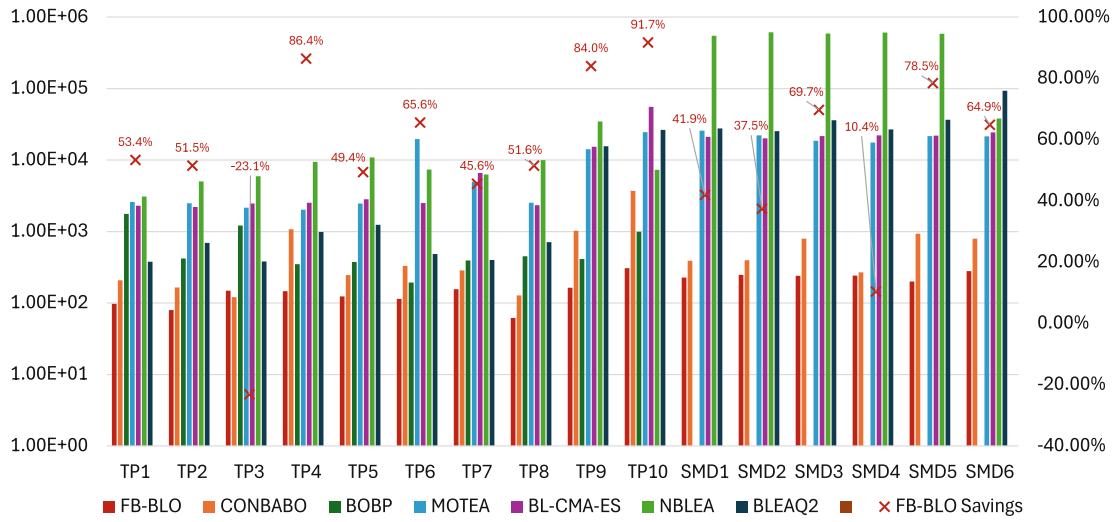



**Fig. 2.** The number of total function evaluation comparisons in logarithmic scale (left y-axis) and the FB-BLO savings comparing with the CONBABO (right y-axis).

problems, and $F^{best}$ and $f^{best}$ are the best obtained values from the proposed algorithm, respectively. Also, according to the nested structure of the BOPs, it is hard to determine the computational resources of other experiments. Therefore we share the necessary number of function evaluations for upper and lower level problems in Table 2, denoted by $ULFEval$ and $LLFEval$ respectively.

As shown in Table 1, the FB-BLO algorithm achieved competitive accuracy scores compared with the others. Both CONBABO and BOBP are based on the Bayesian approach to bilevel problems and the proposed FB-BLO algorithm reached better accuracy on the majority of the TP problems. The BLEAQ2 and MOTEA algorithms achieved $1E-06$ accuracy in several instances of the TP benchmark which is fairly better than the proposed FB-BLO algorithm. TP9 and TP10 problems are relatively higher dimension problems and the FB-BLO algorithm achieved competitive accuracy on these problems. The SMD benchmark problems are unconstrained problems and we can see from Table 1 our proposed algorithm performed well with the other compared methods.

As shown in Table 2, the FB-BLO algorithm surpasses all other algorithms in terms of function evaluations by requiring the minimum number of function evaluations for all problems in both benchmarks. We can observe that the MOTEA, BL-CMA-ES, NBLEA and BLEAQ2 algorithms consume much more function evaluations compared with the Bayesian-based approaches. The main reason behind this is the integrated population-based search strategy of EAs. The CONBABO and BOBP algorithms require more function evaluations compared with the proposed FB-BLO algorithm because they both used BO at the upper level and EAs at the lower level problem. The FB-BLO algorithm achieved competitive results with much more fewer function evaluations compared with the other Bayesian-based algorithms, CONBABO and BOBP because of applying a fully Bayesian approach at both levels. From the experimental results in Tables 1 and

2, we conclude that the combination of using BO at both upper and lower level significantly reduces the necessary function evaluations when solving BOPs.

Solving BOPs effectively requires good performance at both levels, therefore we also consider the total function evaluations in the whole process. Figure 2 compares the total function evaluations in logarithmic scale on the left y-axis considering the sum of upper and lower level optimization. It also contains the FB-BLO savings compared with the second-best algorithm CONBABO on the right y-axis. The saving of the total function evaluation is calculated as follows:

$$\text{FB-BLO Savings}\,(\%) = \frac{\text{TotalFEval}_{CONBABO} - \text{TotalFEval}_{FB-BLO}}{\text{TotalFEval}_{CONBABO}} \tag{5}$$

As we can see from Fig. 2, the FB-BLO algorithm reached competitive accuracies in all test problems but TP3. The best improvement is TP10 with 91.7% compared with the CONBABO. We also run the Wilcoxon rank-sum test at a significance level of 0.05 to verify if there is a significant difference between the total function evaluation of the FB-BLO algorithm and the CONBABO algorithm. The obtained p-value from the test is 0.00058 ($p < 0.05$) which is smaller than the significance level, and it shows that the performance of the proposed algorithm is significant. We can observe that the proposed algorithm achieved competitive results with significantly fewer total function evaluations and is successful in solving both constraint and unconstrant bilevel problems.

## 5 Conclusion and Future Works

BOPs are a specific kind of optimization problem that features two or more decision-makers and models a hierarchical decision-making process. In this study, we developed a fully BO approach to solve BOPs, called FB-BLO. We model the upper and lower level problems with two different $\mathcal{GP}s$ and used both upper and lower level decisions with their objectives to use the cooperation between both decisions and objective values. Also, the black-box approach at both levels made it possible to model expensive-to-evaluate functions that may be unavailable or impractical to real-world applications. Based on the experimental results, it was demonstrated that the proposed algorithm can improve the performance in terms of computational cost while obtaining competitive results. It significantly reduced the number of function evaluations and handled the complexities without any assumptions such as nonconvexity, differentiability, etc. Hence, we believe that the proposed fully Bayesian approach is a promising way to improve the ability to solve BOPs and model real-world problems more easily and effectively.

In this study, we investigated and experimented with the presented approach only with the continuous BOPs. It would be interesting to explore and verify the performance of this approach in combinatorial search space. We used the basic acquisition function during BO and it would also be interesting to see more advanced acquisition function performances at both levels. Many practical applications have been modelled as BOPs as we mentioned above, so there are

numerous practical problems to solve and explore the performance without any assumptions about the characteristics of objective functions.

**Acknowledgments.** This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 12/RC/2289-P2 at Insight the SFI Research Centre for Data Analytics at UCC, which is co-funded under the European Regional Development Fund. For Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. The authors have no competing interests to declare that are relevant to the content of this article.

## References

1. Angelo, J.S., Krempser, E., Barbosa, H.J.: Differential evolution for bilevel programming. In: IEEE Congress on Evolutionary Computation, pp. 470–477 (2013)
2. Authors, T.G.: GPyOpt: a Bayesian optimization framework in python (2016)
3. Bard, J.F., Falk, J.E.: An explicit solution to the multi-level programming problem. Comput. Oper. Res. **9**(1), 77–100 (1982)
4. Bracken, J., McGill, J.T.: Mathematical programs with optimization problems in the constraints. Oper. Res. **21**(1), 37–44 (1973)
5. Bracken, J., McGill, J.T.: Defense applications of mathematical programs with optimization problems in the constraints. Oper. Res. **22**(5), 1086–1096 (1974)
6. Cai, X., Gao, L., Li, X.: Efficient generalized surrogate-assisted evolutionary algorithm for high-dimensional expensive problems. IEEE Trans. Evol. Comput. **24**(2), 365–379 (2020)
7. Camacho, J.F., Corpus, C., Villegas, J.G.: Metaheuristics for bilevel optimization: a comprehensive review. Comput. Oper. Res. **161**, 106410 (2024)
8. Chen, L., Liu, H.L., Li, K., Tan, K.C.: Evolutionary bi-level optimization via multi-objective transformation-based lower level search. IEEE Trans. Evol. Comput. 1 (2023)
9. Colson, B., Marcotte, P., Savard, G.: A trust-region method for nonlinear bilevel programming: algorithm and computational experience. Comput. Optim. Appl. **30**, 211–227 (2005)
10. Cox, D.D., John, S.: SDO: a statistical method for global optimization. In: Multidisciplinary Design Optimization: State-of-the-Art, pp. 315–329 (1997)
11. Dogan, V., Prestwich, S.: Bayesian optimization with multi-objective acquisition function for bilevel problems. In: Longo, L., O'Reilly, R. (eds.) AICS 2022. CCIS, vol. 1662, pp. 409–422. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-26438-2_32
12. Dogan, V., Prestwich, S.: Bilevel optimization by conditional Bayesian optimization. In: Nicosia, G., Ojha, V., La Malfa, E., La Malfa, G., Pardalos, P.M., Umeton, R. (eds.) LOD 2023, Part I. LNCS, vol. 14505, pp. 243–258. Springer, Heidelberg (2024). https://doi.org/10.1007/978-3-031-53969-5_19
13. Edmunds, T., Bard, J.: Algorithms for nonlinear bilevel mathematical programs. IEEE Trans. Syst. Man Cybern. **21**(1), 83–89 (1991)
14. Frazier, P.: A tutorial on Bayesian optimization. ArXiv abs/1807.02811 (2018)
15. Girigoudar, K., Roald, L.A.: Identifying secure operating ranges for der control using bilevel optimization. IEEE Trans. Smart Grid **15**(3), 2921–2933 (2024)

16. Hansen, P., Jaumard, B., Savard, G.: New branch-and-bound rules for linear bilevel programming. SIAM J. Sci. Stat. Comput. **13**(5), 1194–1217 (1992)
17. Haywood, A.B., Lunday, B.J., Robbins, M.J.: Intruder detection and interdiction modeling: a bilevel programming approach for ballistic missile defense asset location. Omega **110**, 102640 (2022)
18. He, X., Zhou, Y., Chen, Z.: Evolutionary bilevel optimization based on covariance matrix adaptation. IEEE Tran. Evol. Comput. **23**(2), 258–272 (2019)
19. Hong, Q., Meng, F., Liu, J., Bo, R.: A bilevel game-theoretic decision-making framework for strategic retailers in both local and wholesale electricity markets. Appl. Energy **330**, 120311 (2023)
20. Jiang, H., Chou, K., Tian, Y., Zhang, X., Jin, Y.: Efficient surrogate modeling method for evolutionary algorithm to solve bilevel optimization problems. IEEE Trans. Cybern. 1–13 (2023)
21. Kieffer, E., Danoy, G., Bouvry, P., Nagih, A.: Bayesian optimization approach of general bi-level problems. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO 2017, pp. 1614–1621. Association for Computing Machinery, New York (2017)
22. Kleinert, T., Labbé, M., Plein, F., Schmidt, M.: Technical note—there's no free lunch: on the hardness of choosing a correct big-M in bilevel optimization. Oper. Res. **68** (2020)
23. Li, X., Tian, P., Min, X.: A hierarchical particle swarm optimization for solving bilevel programming problems. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 1169–1178. Springer, Heidelberg (2006). https://doi.org/10.1007/11785231_122
24. Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization. Math. Program. **45**(1), 503–528 (1989)
25. Rasmussen, C.E.: Gaussian processes in machine learning. In: Bousquet, O., von Luxburg, U., Rätsch, G. (eds.) ML -2003. LNCS (LNAI), vol. 3176, pp. 63–71. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28650-9_4
26. Sinha, A., Lu, Z., Deb, K., Malo, P.: Bilevel optimization based on iterative approximation of mappings. J. Heuristics **26** (2020)
27. Sinha, A., Malo, P., Deb, K.: Test problem construction for single-objective bilevel optimization. Evol. Comput. **22** (2013)
28. Sinha, A., Malo, P., Deb, K.: Efficient evolutionary algorithm for single-objective bilevel optimization (2013)
29. Sinha, A., Malo, P., Deb, K.: Evolutionary algorithm for bilevel optimization using approximations of the lower level optimal solution mapping. Eur. J. Oper. Res. **257**(2), 395–411 (2017)
30. Sinha, A., Malo, P., Deb, K.: Evolutionary algorithm for bilevel optimization using approximations of the lower level optimal solution mapping. Eur. J. Oper. Res. **257**, 395–411 (2017)
31. Sinha, A., Malo, P., Deb, K.: A review on bilevel optimization: from classical to evolutionary approaches and applications. IEEE Trans. Evol. Comput. **22**(2), 276–295 (2018)
32. Sinha, A., Soun, T., Deb, K.: Evolutionary bilevel optimization using KKT proximity measure. In: IEEE Congress on Evolutionary Computation, pp. 2412–2419 (2017)
33. von Stackelberg, H.: Marktform und Gleichgewicht. Die Handelsblatt-Bibliothek "Klassiker der Nationalökonomie" (1934)
34. Stein, M.: Large sample properties of simulations using Latin hypercube sampling. Technometrics **29**(2), 143–151 (1987)

35. Vicente, L., Savard, G., Júdice, J.: Descent approaches for quadratic bilevel programming. J. Optim. Theory Appl. **81**(2), 379–399 (1994)
36. Wang, G., Shan, S.: Review of metamodeling techniques in support of engineering design optimization. J. Mech. Design **129** (2007)
37. Wang, H., Feng, L., Jin, Y., Doherty, J.: Surrogate-assisted evolutionary multitasking for expensive minimax optimization in multiple scenarios. IEEE Comput. Intell. Mag. **16**(1), 34–48 (2021)
38. Wein, L.: Or forum—homeland security: from mathematical models to policy implementation. Oper. Res. **57**, 801–811 (2009)
39. White, D.J., Anandalingam, G.: A penalty function approach for solving bi-level linear programs. J. Global Optim. **3**, 397–419 (1993)
40. Yan, S., Wang, W., Li, X., Lv, H., Fan, T., Aikepaer, S.: Stochastic optimal scheduling strategy of cross-regional carbon emissions trading and green certificate trading market based on stackelberg game. Renew. Energy **219**, 119268 (2023)
41. Yin, Y.: Genetic-algorithms-based approach for bilevel programming models. J. Transp. Eng.-ASCE **126** (2000)
42. Zhang, Q., Liu, S.Q., D'Ariano, A., Chung, S.H., Masoud, M., Li, X.: A bi-level programming methodology for decentralized mining supply chain network design. Expert Syst. Appl. **250**, 123904 (2024)
43. Zheng, A.Y., He, T., Qiu, Y., Wang, M., Wipf, D.: BloomGML: graph machine learning through the lens of bilevel optimization (2024)