Evolutionary Bilevel Optimization: An Introduction and Recent Advances

Ankur Sinha, Pekka Malo and Kalyanmoy Deb

Abstract Bilevel optimization involves two levels of optimization where one optimization level acts as a constraint to another optimization level. There are enormous applications that are bilevel in nature; however, given the difficulties associated with solving this difficult class of problem, the area still lacks efficient solution methods capable of handling complex application problems. Most of the available solution methods can either be applied to highly restrictive class of problems, or are computationally very expensive such that they do not scale for large scale bilevel problems. The difficulties in bilevel programming arise primarily from the nested structure of the problem. Evolutionary algorithms have been able to demonstrate its potential in solving single-level optimization problems. In this chapter, we provide an introduction to the progress made by the evolutionary computation community towards handling bilevel problems. The chapter highlights past research and future research directions both on single as well as multiobjective bilevel programming. Some of the immediate application areas of bilevel programming have also been highlighted.

Keywords Bilevel optimization · Stackelberg games · Evolutionary algorithms · Mathematical programming

---

A. Sinha (✉)
Production and Quantitative Methods, Indian Institute of Management, Ahmedabad 380015, India
e-mail: asinha@iima.ac.in

P. Malo
Department of Information and Service Economy, Aalto University School of Business, PO Box 21220, 00076 Aalto, Finland
e-mail: pekka.malo@aalto.fi

K. Deb
Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, USA
e-mail: kdeb@egr.msu.edu

S. Bechikh et al. (eds.), *Recent Advances in Evolutionary Multi-objective Optimization*, Adaptation, Learning, and Optimization 20,
DOI 10.1007/978-3-319-42978-6_3

1 Introduction

Bilevel optimization is characterized as a mathematical program with two levels of optimization. The outer optimization problem is commonly referred to as the upper level optimization problem and the inner optimization problem is commonly referred to as the lower level optimization problem. The origin of bilevel optimization can be traced to two roots: these problems were first realized by Stackelberg [1] in the area of game theory and came to be known as Stackelberg games; later these problems were realized in the area of mathematical programming by Bracken and McGill [2] as a constrained optimization task, where the lower level optimization problem acts as a constraint to the upper level optimization problem. These problems are known to be difficult due to its nested structure; therefore, it has received most attention from the mathematical community towards simple cases where the objective functions and constraints are linear [3, 4], quadratic [5–7] or convex [8]. The nested structure in bilevel introduces difficulties such as non-convexity and disconnectedness even for simpler instances of bilevel optimization like bilevel linear programming problems. Bilevel linear programming is known to be strongly NP-hard [9], and it has been proven that merely evaluating a solution for optimality is also a NP-hard task [10]. This gives us an idea about the kind of challenges offered by bilevel problems with complex (non-linear, non-convex, discontinuous etc.) objective and constraint functions.

An interest in bilevel programming has been driven by a number of new applications arising in different fields of optimization. For instance, in the context of homeland security [11–13], bilevel and even trilevel optimization models are common. In game theoretic settings, bilevel programs have been used in the context of optimal tax policies [14–16]; model production processes [17]; investigation of strategic behavior in deregulated markets [18] and optimization of retail channel structures [19], among others. Bilevel optimization applications are ubiquitous and arise in many other disciplines, like in transportation [20–22], management [23, 24], facility location [23, 25, 26], chemical engineering [27, 28], structural optimization [29, 30], and optimal control [31, 32] problems.

Evolutionary computation [33] techniques have been successfully applied to handle mathematical programming problems and applications that do not adhere to regularities like continuity, differentiability or convexities. Due to these properties of evolutionary algorithms, attempts have been made to solve bilevel optimization problems using these methods, as even simple (linear or quadratic) bilevel optimization problems are intrinsically non-convex, non-differentiable and disconnected at times. However, the advantages come with a trade-off. Most of the evolutionary bilevel techniques are nested where an outer algorithm handles the upper level optimization task and an inner algorithm handles the lower level optimization task, thereby making the overall bilevel optimization computationally very intensive. To address these problems attempts have been made to reduce the computational expense of evolutionary bilevel optimization algorithms by utilizing metamodeling-based principles. Multiobjective bilevel programming is a natural extension of bilevel optimization problems with single objectives. However, multiple objectives in bilevel optimization, along with computational challenges, brings in intricacies related to hierarchical decision making.

In this chapter, we highlight some of the past, and recent studies and results in the area of evolutionary bilevel optimization. The chapter begins with a survey on single objective bilevel optimization in Sect. 2. This is followed by single-level formulations of bilevel optimization in Sect. 3. Thereafter, in Sect. 4 we discuss and compare some recent solution methods for bilevel optimization. Section 5 introduces multiobjective bilevel optimization and provides a survey on the topic. In Sect. 6 we discuss the decision making issues in multiobjective bilevel optimization. Finally, we conclude in Sect. 7 with some ideas on future research directions.

2 A Survey on Evolutionary Bilevel Optimization

Most of the evolutionary approaches proposed to handle bilevel optimization problems are nested in nature. As the name suggests, these approaches rely on two optimization algorithms, where one algorithm is executed within the other. Based on the complexity of the optimization tasks at each level, researchers have chosen to use either evolutionary algorithms at both levels or evolutionary algorithm at one level and classical optimization algorithm at the other level. One of the earliest evolutionary algorithms for solving bilevel optimization problems was proposed in the early 1990s by Mathieu et al. [34] who used a nested approach with genetic algorithm at the upper level, and linear programming at the lower level. Later, Yin [35] used genetic algorithm at the upper level and Frank–Wolfe algorithm (reduced gradient method) at the lower level. In both these approaches a lower level optimization task was executed for every upper level member that emphasizes the nested structure of these approaches. Along similar lines, nested procedures were used in [36–39]. Approaches with evolutionary algorithms at both levels are also common; for instance, in [40] authors used differential evolution at both levels, and in [41] authors nested differential evolution within an ant colony optimization.

In a number of studies, where lower level problem adhered to certain regularity conditions, researchers have used the KKT conditions for the lower level problem to reduce the bilevel problem into a single-level problem. The reduced single-level problem is then solved with an evolutionary algorithm. For instance, Hejazi et al. [42], reduced the linear bilevel problem to single-level and then used a genetic algorithm, where chromosomes emulate the vertex points, to solve the problem. Wang et al. [43] used KKT conditions to reduce the bilevel problem into single-level, and then utilized a constraint handling scheme to successfully solve a number of standard test problems. A later study by Wang et al. [44] introduced an improved algorithm that performed better than the previous approach [43]. Recently, Jiang et al. [45] reduced the bilevel optimization problem into a non-linear optimization problem with complementarity constraints, which is sequentially smoothed and solved with a PSO algorithm. Other studies using similar ideas are [46, 47].

It is noteworthy that utilization of KKT conditions restricts the algorithm's applicability to only a special class of bilevel problems. To overcome this drawback, researchers are looking into metamodeling based approaches where the lower level optimal reaction set is approximated over generations of the evolutionary algorithm. Studies in this direction are [48, 49]. Along similar lines, attempts have been made to metamodel the lower level optimal value function [50] to solve bilevel optimization problems. Approximating the lower level optimal value function may offer a few advantages over approximating the lower level reaction set that has been highlighted in this chapter.

3 Bilevel Formulation and Single-Level Reductions

In this section, we provide a general formulation for bilevel optimization, and different ways people have used to reduce bilevel optimization problems to single-level problems. Bilevel problems contain two levels, upper and lower, where lower level is nested within the upper level problem. The two levels have their own objectives, constraints and variables. In the context of game theory, the two problems are also referred to as the leader's (upper) and follower's problems (lower). The lower level optimization problem is a parametric optimization problem that is solved with respect to the lower level variables while the upper level variables act as parameters. The difficulty in bilevel optimization arises from the fact that only lower level optimal solutions can be considered as feasible members, if they also satisfy the upper level constraints. Below we provide a general bilevel formulation:

Definition 1 For the upper-level objective function $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ and lower-level objective function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, the bilevel optimization problem is given by

$$\begin{aligned} \underset{x_u \in X_U, x_l \in X_L}{\text{“min”}} \quad & F(x_u, x_l) \text{ subject to} \\ & x_l \in \underset{x_l \in X_L}{\operatorname{argmin}} \{ f(x_u, x_l) : g_j(x_u, x_l) \leq 0, \ j = 1, \ldots, J \} \\ & G_k(x_u, x_l) \leq 0, k = 1, \ldots, K \end{aligned}$$

where $G_k : X_U \times X_L \to \mathbb{R}$, $k = 1, \ldots, K$ denotes the upper level constraints, and $g_j : X_U \times X_L \to \mathbb{R}$ represents the lower level constraints, respectively.

3.1 *Optimistic Versus Pessimistic*

Quotes have been used while specifying the upper level minimization problem in Definition 1 because the problem is ill-posed for cases where the lower level has multiple optimal solutions. For instance, Fig. 1 shows the case where the lower level

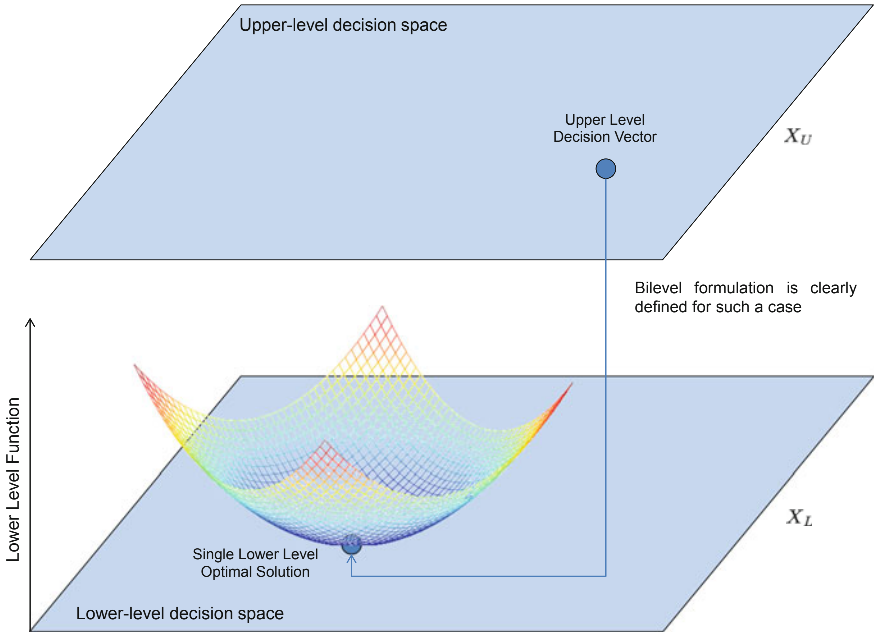

Fig. 1 A scenario where there is a single lower level optimal solution corresponding to an upper level decision vector. The bilevel optimization problem in Definition 1 is clearly defined for this case

problem has a single optimal solution corresponding to an upper level decision. Therefore, it is clear that for the upper level decision, the only rational lower level decision would be the single optimal solution at the lower level. However, there is lack of clarity in the situation shown in Fig. 2, as it is not clear that, out of multiple lower level optimal solutions, which solution will actually be chosen by the lower level decision maker. If the selection of the lower level decision maker is unknown, the bilevel formulation remains ill-defined. It is common to assume either of the two positions, i.e., optimistic or pessimistic, to sort out this ambiguity. In optimistic position some form of cooperation is assumed between the leader and the follower. For any given leader's decision vector that has multiple optimal solutions for the follower, the follower is expected to choose that optimal solution that leads to the best objective function value for the leader. On the other hand, in a pessimistic position the leader optimizes for the worst case, i.e. the follower may choose that solution from the optimal set which leads to the worst objective function value for the leader. Optimistic position being more tractable is commonly studied in the literature, and we also consider the optimistic position in this chapter.

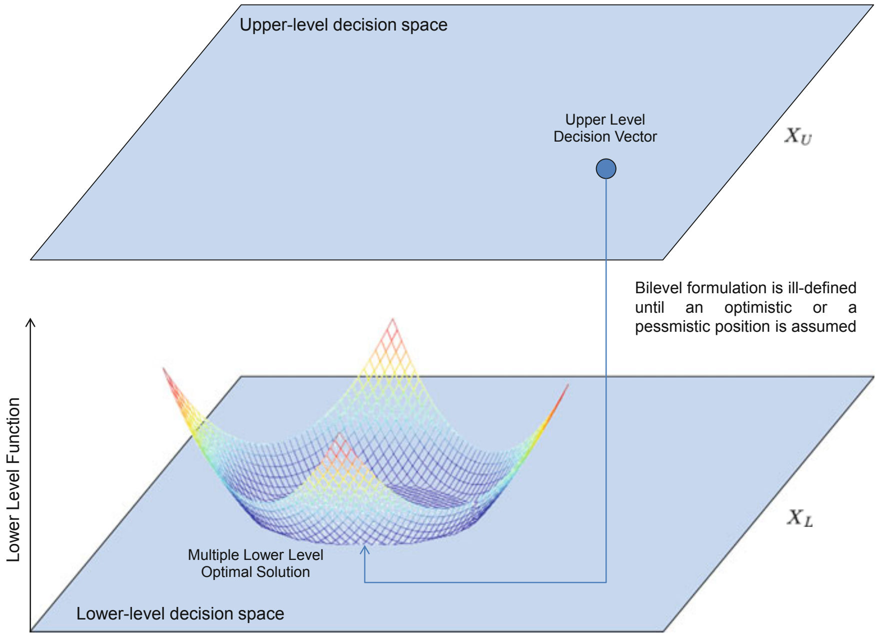

Fig. 2 A scenario where there is a multiple lower level optimal solution corresponding to an upper level decision vector. The bilevel optimization problem in Definition 1 is ill-defined for this case if the lower level's selection is not known or assumed

3.2 KKT Reduction

When the lower level problem in Definition 1 adheres to certain convexity and regularity conditions, it is possible to replace the lower level optimization task with its KKT conditions.

Definition 2 The KKT conditions appear as Lagrangian and complementarity constraints in the single-level formulation provide below:

$$
\begin{aligned}
&\min_{x_u \in X_U, x_l \in X_L, \lambda} \quad F(x_u, x_l) \\
&\text{subject to} \\
&\qquad G_k(x_u, x_l) \leq 0, k = 1, \ldots, K, \\
&\qquad g_j(x_u, x_l) \leq 0, \ j = 1, \ldots, J, \\
&\qquad \lambda_j g_j(x_u, x_l) = 0, \ j = 1, \ldots, J, \\
&\qquad \lambda_j \geq 0, \ j = 1, \ldots, J, \\
&\qquad \nabla_{x_l} L(x_u, x_l, \lambda) = 0,
\end{aligned}
$$

where

$$L(x_u, x_l, \lambda) = f(x_u, x_l) + \sum_{j=1}^{J} \lambda_j g_j(x_u, x_l).$$

The above formulation might not be simple to handle, as the Lagrangian constraints often lead to non-convexities, and the complementarity condition being combinatorial, make the overall problem a mixed integer problem. In case of linear bilevel optimization problems, the Lagrangian constraint is also linear. Therefore, the single-level reduced problem becomes a mixed integer linear program. Approaches based on vertex enumeration [51–53], as well as branch-and-bound [54, 55] have been proposed to solve these problems.

3.3 Reaction Set Mapping

An equivalent formulation of the problem given in Definition 1 can be stated in terms of set-valued mappings as follows:

Definition 3 Let $\Psi : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ be the reaction set mapping,

$$\Psi(x_u) = \underset{x_l \in X_L}{\text{argmin}}\{f(x_u, x_l) : g_j(x_u, x_l) \leq 0, j = 1, \ldots, J\},$$

which represents the constraint defined by the lower-level optimization problem, i.e. $\Psi(x_u) \subset X_L$ for every $x_u \in X_U$. Then the bilevel optimization problem can be expressed as a constrained optimization problem as follows:

$$\begin{aligned} &\min_{x_u \in X_U, x_l \in X_L} \quad F(x_u, x_l) \\ &\text{subject to} \\ &\qquad x_l \in \Psi(x_u) \\ &\qquad G_k(x_u, x_l) \leq 0, k = 1, \ldots, K \end{aligned}$$

Note that if the Ψ-mapping can somehow be determined, the problem reduces to a single level constrained optimization task. However, that is rarely the case. Evolutionary computation studies that rely on iteratively mapping this set to avoid frequent lower level optimization are [48, 49]. The idea behind the algorithm has been shown through Figs. 3 and 4. To begin with, the lower level problem is completely solved for a few upper level decision vectors. For example, in Fig. 3 the lower level decisions corresponding to upper level decisions a, b, c, d, e and f are determined by solving a lower level problem completely. The lower level decisions for these members correspond to the actual Ψ-mapping (unknown). These member are then used to find an approximate Ψ-mapping locally as shown in Fig. 4. For every new upper level

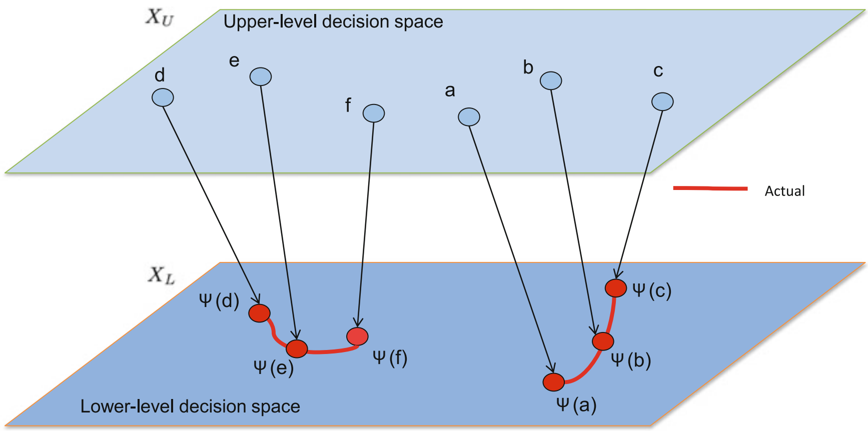

Fig. 3 Solving the lower level optimization problem completely for random upper level members like a, b, c, d, e and f provides the corresponding lower level optimal solutions represented by $\Psi(a)$, $\Psi(b)$, $\Psi(c)$, $\Psi(d)$, $\Psi(e)$ and $\Psi(f)$. The Ψ-mapping is assumed to be single valued

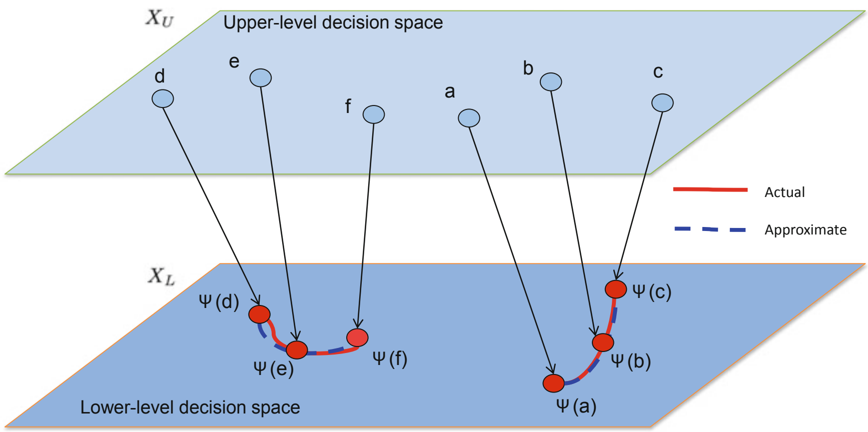

Fig. 4 An approximate mapping for the lower level reaction set estimated using the actual values $\Psi(a)$, $\Psi(b)$, $\Psi(c)$, $\Psi(d)$, $\Psi(e)$ and $\Psi(f)$. Local approximations are preferable over a global approximation of the Ψ-mapping

member, the local approximation is used to identify the lower level decision instead of solving the lower level optimization problem. The idea is used iteratively until convergence. The idea works well when the Ψ-mapping is single valued.

3.4 Lower Level Optimal Value Function

Another equivalent definition of the problem in Definition 1 can be given in terms of the lower level optimal value function that is defined below [56]:

Definition 4 Let $\varphi : X_U \to R$ be the lower level optimal value function mapping,

$$\varphi(x_u) = \min_{x_l \in X_L} \{f(x_u, x_l) : g_j(x_u, x_l) \leq 0, j = 1, \ldots, J\},$$

which represents the minimum lower level function value corresponding to any upper level decision vector. Then the bilevel optimization problem can be expressed as follows:

$$\begin{aligned} &\min_{x_u \in X_U, x_l \in X_L} \quad F(x_u, x_l) \\ &\text{subject to} \\ &\qquad f(x_u, x_l) \leq \varphi(x_u) \\ &\qquad g_j(x_u, x_l) \leq 0, j = 1, \ldots, J \\ &\qquad G_k(x_u, x_l) \leq 0, k = 1, \ldots, K. \end{aligned}$$

The φ-mapping can be approximated iteratively during the generations of the evolutionary algorithm, and a reduced problem described in Definition 4 can be frequently solved to converge towards the bilevel optimum. An evolutionary algorithm relying on this idea can be found in [50]. Approximating the optimal value function mapping offers an advantage over approximating reaction set mapping, as the optimal value function mapping is not set valued. Moreover, it returns a scalar for any given upper level decision vector. Figure 5 shows an example where the lower level problem has multiple optimal solutions for some upper level decisions and single optimal solutions for others. In all situations, the φ-mapping remains single valued scalar. Though there are advantages associated with estimating the φ-mapping, it is also interesting to note in Definition 4 that the reduced single level problem has to be

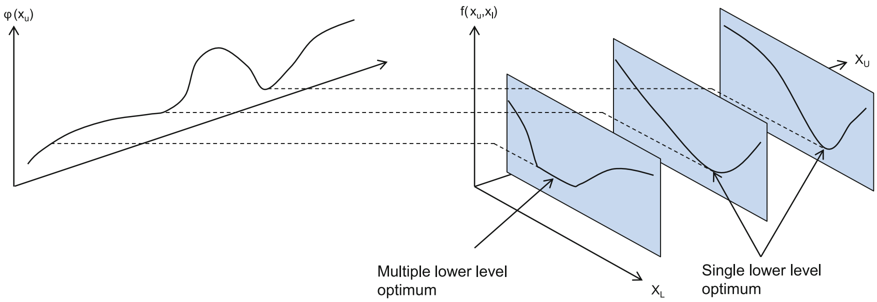

Fig. 5 An example showing φ-mapping and how it depends on the lower level optimization problem

solved with respect to both upper and lower level variables, while in Definition 7, the lower level variables are directly available from the Ψ-mapping. Therefore, there exists a trade-off.

4 Comparison of Metamodeling Based Evolutionary Approaches for Bilevel Optimization

In this section, we provide the steps of two different evolutionary bilevel algorithms, where one utilizes iterative approximation of the Ψ-mapping, while the other utilizes iterative approximation of the φ-mapping in the intermediate steps. The steps of the algorithms are provided through a flowchart in Fig. 6. For brevity, we do not discuss the steps of the evolutionary algorithm, as any scheme can be utilized in the provided framework to handle bilevel optimization problems. For further information about the implementation of the approaches the readers are referred to [50].

The intermediate steps of the above algorithms utilizes quadratic approximation for approximating the Ψ and the φ mappings. Both the ideas were tested on a set of 8 test problems given in Tables 1 and 2. To assess the savings achieved by the two approximation approaches, we compare them against a nested approach where the approximation idea is not incorporated, but the same evolutionary algorithm

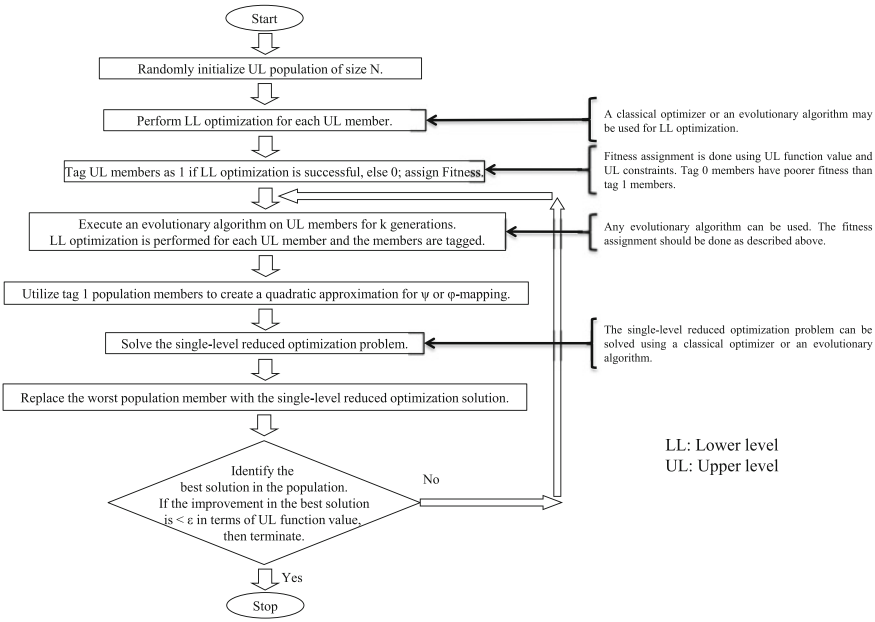

Fig. 6 Flowchart for incorporating approximated φ-mapping in an evolutionary algorithm

Table 1 Standard test problems TP1–TP5

Problem	Formulation	Best Known Sol.
TP1 $n=2,$ $m=2$	$\underset{(x,y)}{\text{Minimize}}\ F(x,y)=(x_1-30)^2+(x_2-20)^2-20y_1+20y_2,$ s.t. $y\in \underset{(y)}{\text{argmin}}\left\{\begin{array}{l} f(x,y)=(x_1-y_1)^2+(x_2-y_2)^2 \\ 0\le y_i\le 10,\quad i=1,2\end{array}\right\},$ $x_1+2x_2\ge 30, x_1+x_2\le 25, x_2\le 15$	$F=225.0$ $f=100.0$
TP2 $n=2,$ $m=2$	$\underset{(x,y)}{\text{Minimize}}\ F(x,y)=2x_1+2x_2-3y_1-3y_2-60,$ s.t. $y\in \underset{(y)}{\text{argmin}}\left\{\begin{array}{l} f(x,y)=(y_1-x_1+20)^2+(y_2-x_2+20)^2 \\ x_1-2y_1\ge 10, x_2-2y_2\ge 10 \\ -10\ge y_i\ge 20,\quad i=1,2\end{array}\right\},$ $x_1+x_2+y_1-2y_2\le 40,$ $0\le x_i\le 50,\quad i=1,2.$	$F=0.0$ $f=100.0$
TP3 $n=2,$ $m=2$	$\underset{(x,y)}{\text{Minimize}}\ F(x,y)=-(x_1)^2-3(x_2)^2-4y_1+(y_2)^2,$ s.t. $y\in \underset{(y)}{\text{argmin}}\left\{\begin{array}{l} f(x,y)=2(x_1)^2+(y_1)^2-5y_2 \\ (x_1)^2-2x_1+(x_2)^2-2y_1+y_2\ge -3 \\ x_2+3y_1-4y_2\ge 4 \\ 0\le y_i,\quad i=1,2\end{array}\right\},$ $(x_1)^2+2x_2\le 4,$ $0\le x_i,\quad i=1,2$	$F=-18.6787$ $f=-1.0156$
TP4 $n=2,$ $m=3$	$\underset{(x,y)}{\text{Minimize}}\ F(x,y)=-8x_1-4x_2+4y_1-40y_2-4y_3,$ s.t. $y\in \underset{(y)}{\text{argmin}}\left\{\begin{array}{l} f(x,y)=x_1+2x_2+y_1+y_2+2y_3 \\ y_2+y_3-y_1\le 1 \\ 2x_1-y_1+2y_2-0.5y_3\le 1 \\ 2x_2+2y_1-y_2-0.5y_3\le 1 \\ 0\le y_i,\quad i=1,2,3\end{array}\right\},$ $0\le x_i,\quad i=1,2$	$F=-29.2$ $f=3.2$

(Note that $x=x_u$ and $y=x_l$)

described in Fig. 6 is used at the upper level and a lower level optimization problem is solved for every upper level member. Hereafter, we refer this benchmark as a no-approximation approach. Whenever lower level optimization is required, we rely on sequential quadratic programming to solve the problem for all cases. Table 3 provides the median function evaluations (31 runs) at the upper and lower level required by each of the three cases, i.e., φ-approximation, Ψ-approximation and no-approximation. Detailed results from multiple runs are presented through Figs. 7 and 8. Interestingly, both the approximation ideas perform significantly well on all

Table 2 Standard test problems TP6–TP8

Problem	Formulation	Best Known Sol.
TP5 $n = 2, m = 2$	$\underset{(x,y)}{\text{Minimize}}\ F(x, y) = rt(x)x - 3y_1 - 4y_2 + 0.5t(y)y,$ s.t. $y \in \underset{(y)}{\text{argmin}} \left\{ \begin{array}{l} f(x, y) = 0.5t(y)hy - t(b(x))y \\ -0.333y_1 + y_2 - 2 \leq 0 \\ y_1 - 0.333y_2 - 2 \leq 0 \\ 0 \leq y_i, \quad i = 1, 2 \end{array} \right\},$ where $h = \begin{pmatrix} 1 & 3 \\ 3 & 10 \end{pmatrix}, b(x) = \begin{pmatrix} -1 & 2 \\ 3 & -3 \end{pmatrix} x, r = 0.1$ $t(\cdot)$ denotes transpose of a vector	$F = -3.6$ $f = -2.0$
TP6 $n = 1, m = 2$	$\underset{(x,y)}{\text{Minimize}}\ F(x, y) = (x_1 - 1)^2 + 2y_1 - 2x_1,$ s.t. $y \in \underset{(y)}{\text{argmin}} \left\{ \begin{array}{l} f(x, y) = (2y_1 - 4)^2 + \\ (2y_2 - 1)^2 + x_1 y_1 \\ 4x_1 + 5y_1 + 4y_2 \leq 12 \\ 4y_2 - 4x_1 - 5y_1 \leq -4 \\ 4x_1 - 4y_1 + 5y_2 \leq 4 \\ 4y_1 - 4x_1 + 5y_2 \leq 4 \\ 0 \leq y_i, \quad i = 1, 2 \end{array} \right\},$ $0 \leq x_1$	$F = -1.2091$ $f = 7.6145$
TP7 $n = 2, m = 2$	$\underset{(x,y)}{\text{Minimize}}\ F(x, y) = -\frac{(x_1+y_1)(x_2+y_2)}{1+x_1y_1+x_2y_2},$ s.t. $y \in \underset{(y)}{\text{argmin}} \left\{ \begin{array}{l} f(x, y) = \frac{(x_1+y_1)(x_2+y_2)}{1+x_1y_1+x_2y_2} \\ 0 \leq y_i \leq x_i, \quad i = 1, 2 \end{array} \right\},$ $(x_1)^2 + (x_2)^2 \leq 100$ $x_1 - x_2 \leq 0$ $0 \leq x_i, \quad i = 1, 2$	$F = -1.96$ $f = 1.96$
TP8 $n = 2, m = 2$	$\underset{(x,y)}{\text{Minimize}}\ F(x, y) = \lvert 2x_1 + 2x_2 - 3y_1 - 3y_2 - 60 \rvert,$ s.t. $y \in \underset{(y)}{\text{argmin}} \left\{ \begin{array}{l} f(x, y) = (y_1 - x_1 + 20)^2 + \\ (y_2 - x_2 + 20)^2 \\ 2y_1 - x_1 + 10 \leq 0 \\ 2y_2 - x_2 + 10 \leq 0 \\ -10 \leq y_i \leq 20, \quad i = 1, 2 \end{array} \right\},$ $x_1 + x_2 + y_1 - 2y_2 \leq 40$ $0 \leq x_i \leq 50, \quad i = 1, 2$	$F = 0.0$ $f = 100.0$

(Note that $x = x_u$ and $y = x_l$)

Table 3 Median function evaluations for the upper level (UL) and the lower level (LL) from 31 runs of different algorithms

	UL func. evals.			LL func. evals.			Savings	
	φ-appx Med	Ψ-appx Med	No-appx Med	φ-appx Med	Ψ-appx Med	No-appx Med	φ	Ψ
TP1	134	150	–	1438	2061	–	Large	Large
TP2	148	193	436	1498	2852	5686	73 %	50 %
TP3	187	137	633	2478	1422	6867	64 %	79 %
TP4	299	426	1755	3288	6256	19764	83 %	69 %
TP5	175	270	576	2591	2880	6558	61 %	56 %
TP6	110	94	144	1489	1155	1984	25 %	41 %
TP7	166	133	193	2171	1481	2870	24 %	47 %
TP8	212	343	403	2366	5035	7996	69 %	36 %

The savings represent the proportion of total function evaluations (LL+UL) saved because of using the approximation when compared with no-approximation approach

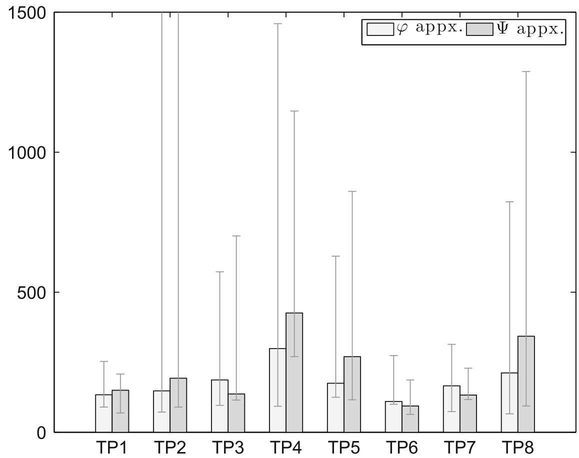

Fig. 7 Box plot (31 runs/samples) for the upper level function evaluations required for test problems 1–8

the problems as compared to the no-approximation approach. The savings column in the table shows the proportion of function evaluations savings that can be directly attributed to φ and Ψ approximations. Slight difference in performance between the two approximation strategies can be attributed to the quality of approximations achieved for specific test problems. To provide the readers an idea about the extent of savings in function evaluations obtained from using metamodeling based strategies, we also provide comparisons with earlier evolutionary approaches [43, 44] in Table 4. These approaches are based on single-level reduction using lower level KKT conditions. A significantly poor performance of these methods emphasizes the fact that even when it is possible to write the KKT constraints for the lower level problem, a single level reduction might not necessarily make the problem easy to solve.

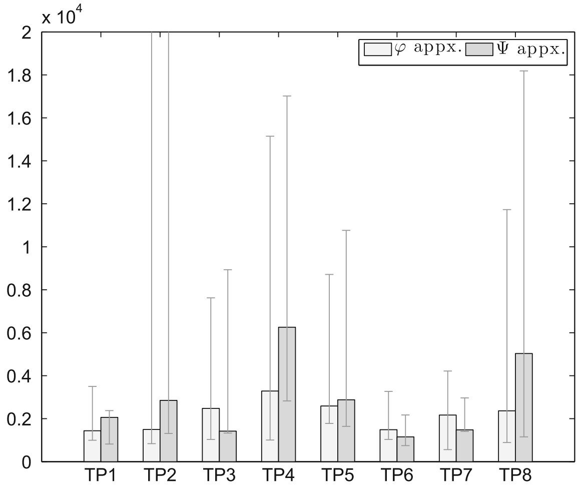

Fig. 8 Box plot (31 runs/samples) for the lower level function evaluations required for test problems 1–8

Table 4 Mean of the sum of upper level (UL) and lower level (LL) function evaluations for different approaches

	Mean func. evals. (UL+LL)				
	φ-appx.	Ψ-appx.	No-appx.	WJL [43]	WLD [44]
TP1	1595	2381	35896	85499	86067
TP2	1716	3284	5832	256227	171346
TP3	2902	1489	7469	92526	95851
TP4	3773	6806	21745	291817	211937
TP5	2941	3451	7559	77302	69471
TP6	1689	1162	1485	163701	65942
TP7	2126	1597	2389	1074742	944105
TP8	2699	4892	5215	213522	182121

It is noteworthy that the Ψ-mapping in a bilevel optimization problem could be a set-valued mapping as shown in Fig. 9, i.e. for some or all upper level decision vectors in the search space, the lower level optimization problem may have multiple optimal solutions. Such a situation offers dual challenges; first, finding the Ψ-set is difficult; second, approximating the set is also difficult. In such cases approximating the Ψ-mapping will not help. To test this hypothesis, we modified all the 8 test problems by adding two additional lower level variables (y_p and y_q) that makes the Ψ-mapping in all the test problems as set-valued for the entire domain of Ψ. The modification

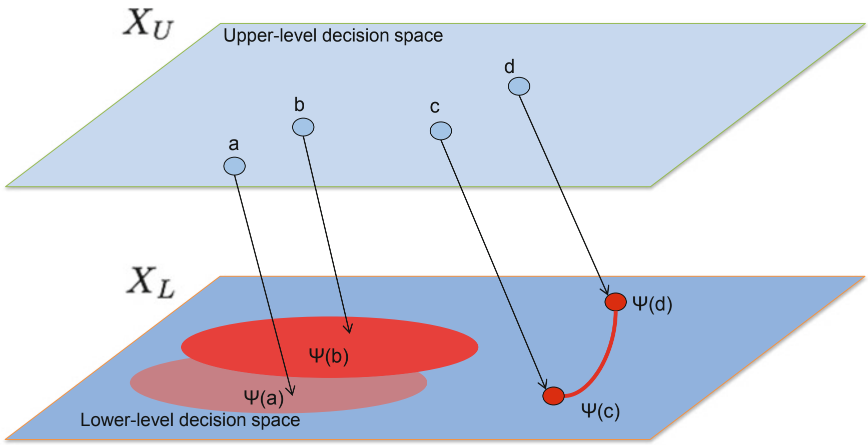

Fig. 9 A scenario where the Ψ-mapping is set-valued in some regions and single-valued in other regions

does not change the original bilevel solution. This was achieved by modifying the upper and lower level functions for all the test problems as follows:

$$F^{new}(x, y) = F(x, y) + y_p^2 + y_q^2$$
$$f^{new}(x, y) = f(x, y) + (y_p - y_q)^2$$
$$y_p, y_q \in [-1, 1]$$

Note that the above modification necessarily makes the lower level problem have multiple optimal solutions corresponding to all x, as the added term gets minimized at $y_p = y_q$ which has infinitely many solutions. Out of the infinitely many lower level optimal solutions, the upper level prefers $y_p = y_q = 0$. With this simple modification, we execute our algorithm with φ-approximation and Ψ-approximation on all test problems, the results for which are presented through Tables 5 and 6. For all the problems, the Ψ-approximation idea fails. The φ-approximation idea continues to work effectively as before. The slight increase in function evaluations for the Ψ-approximation approach comes from the fact that there are additional variables in the problem.

To conclude, the Ψ-mapping offers the advantage that if it can be approximated accurately, it readily gives the optimal lower level variables. However, in cases when this mapping is set-valued, approximating Ψ can be very difficult. On the other hand, the φ-mapping is always single-valued, approximating which is much easier, and is therefore more preferred over the Ψ-mapping. The results shown in this section clearly demonstrate that even a simple modification that leads to multiple lower level optimal solutions, makes the Ψ-approximation strategy fail because of poor quality of approximation. To our best knowledge, most of the studies utilizing metamodeling techniques to solve bilevel optimization problems have mostly relied on approximat-

Table 5 Minimum, median and maximum function evaluations at the upper level (UL) from 31 runs of the φ-approximation algorithm on the modified test problems (m-TP)

	φ-appx.			Ψ-appx.	No-appx.
	Min	Med	Max	Min/Med/Max	Min/Med/Max
m-TP1	130	172	338	–	–
m-TP2	116	217	–	–	–
m-TP3	129	233	787	–	–
m-TP4	198	564	2831	–	–
m-TP5	160	218	953	–	–
m-TP6	167	174	529	–	–
m-TP7	114	214	473	–	–
m-TP8	150	466	2459	–	–

The other two approaches fail on all the test problems

Table 6 Minimum, median and maximum function evaluations at the lower level (LL) from 31 runs of the φ-approximation algorithm on the modified test problems (m-TP)

	φ-appx.			Ψ-appx.	No-appx.
	Min	Med	Max	Min/Med/Max	Min/Med/Max
m-TP1	2096	2680	8629	–	–
m-TP2	2574	4360	–	–	–
m-TP3	1394	3280	13031	–	–
m-TP4	1978	5792	28687	–	–
m-TP5	3206	4360	17407	–	–
m-TP6	2617	3520	8698	–	–
m-TP7	1514	5590	11811	–	–
m-TP8	2521	6240	35993	–	–

The other two approaches fail on all the test problems

ing the Ψ-mapping. Given the ease and reliability offered by the φ-approximation over Ψ-approximation, we believe that future research on metamodeling-based techniques should closely look at the benefits of the φ-approximation.

5 Multiobjective Bilevel Optimization

A substantial body of research exists on single-objective bilevel optimization, but relatively few papers have considered bilevel problems with multiple objectives on both levels. Even less research has been done to understand the impacts of decision-interaction and uncertainty that arise in multiobjective bilevel problems. One of the reasons for little research in the area is that the problem becomes both mathematically and computationally intractable even with simplifying assumptions like continuity, differentiability, convexity etc. However, given that multiobjective bilevel problems exist in practice, researchers have tried to explore ideas to handle these problems.

Some of the studies on multiobjective bilevel optimization that exist are mostly directed towards development of techniques for solving optimistic formulation of the problem, where the decision-makers are assumed to co-operate and the leader can freely choose any Pareto-optimal lower-level solution. Studies by Eichfelder [57, 58] utilize classical techniques to solve simple multiobjective bilevel problems. The lower level problems are handled using a numerical optimization technique, and the upper level problem is handled using an adaptive exhaustive search method. This makes the solution procedure computationally demanding and non-scalable to large-scale problems. The method is close to a nested strategy, where each of the lower level optimization problems is solved to Pareto-optimality. Shi and Xia [59] use the ϵ-constraint method at both levels of a multiobjective bilevel problem to convert the problem into an ϵ-constraint bilevel problem. The ϵ-parameter is elicited from the decision maker, and the problem is solved by replacing the lower level constrained optimization problem with its KKT conditions. The problem is solved for different ϵ-parameters, until a satisfactory solution is found.

With the surge in computation power, a number of nested evolutionary algorithms have also been proposed, which solve the lower level problem completely for every upper level vector to arrive at the problem optima. One of the first studies, utilizing an evolutionary approach for bilevel multiobjective algorithms was in [35]. The study involved multiple objectives at the upper level, and a single objective at the lower level. The study suggested a nested genetic algorithm, and applied it on a transportation planning and management problem. Later [60] used a particle swarm optimization (PSO)-based nested strategy to solve a multi-component chemical system. The lower level problem in their application problem was linear for which they used a specialized linear multiobjective PSO approach. Recently, a hybrid bilevel evolutionary multiobjective optimization algorithm approach coupled with local search was proposed in [61]. In the paper, the authors handled nonlinear as well as discrete bilevel problems with a relatively large number of variables. The study also provided a suite of test problems for bilevel multiobjective optimization. An extension to this study [62] attempted to solve bilevel multiobjective optimization with fewer function evaluations by interacting with the leader. The idea in this study was to interact with the upper level decision maker only to model her preferences and find the most preferred Pareto-optimal point instead of the entire frontier. The study borrowed ideas from the area of preference-based evolutionary algorithms.

Until recently, the focus has been primarily on algorithms for handling deterministic problems. Less emphasis has been paid to the decision-making intricacies that arise in practical multiobjective bilevel problems. The first concern is the reliance on the assumption that transfers decision-making power to the leader by allowing her to freely choose any Pareto-optimal solution from the lower-level optimal frontier. In practical problems, the preferences of the lower-level decision maker may not be aligned with the leader. Although a leader can anticipate the follower's actions and optimize her strategy accordingly, it is unrealistic to assume that she can decide which trade-off the follower should choose. To solve hierarchical problems with conflicting decision-makers, a few studies have proposed a line of interactive fuzzy programming models [63, 64]. The methods have been successfully used to handle decentralized bilevel problems that have more than one lower level decision maker [65]. However, the assumption of mutual co-operation and repeated interactions between decision-makers is not necessarily feasible; e.g., in homeland security applications and competitive business decisions. The second concern is the decision-uncertainty. The strategy chosen by the follower may well deviate from what is expected by the leader, which thus gives rise to uncertainty about the realized outcome. It is worthwhile to note that the notion of decision-uncertainty that emanates from not knowing the follower's preferences exactly is different from the uncertainty that follows from non-preference related factors such as stochastic model parameters.

6 Multiobjective Bilevel Optimization and Decision Making

In this section, we provide three different formulations for a multiobjective bilevel optimization problem. First, we consider the standard formulation, where there is no decision making involved at the lower level and all the lower level Pareto-optimal solutions are considered at the upper level (see Fig. 10). Second, we consider a formulation, where the decision maker acts at the lower level and chooses a solution to her liking. The preference structure of the follower is known to the leader and can

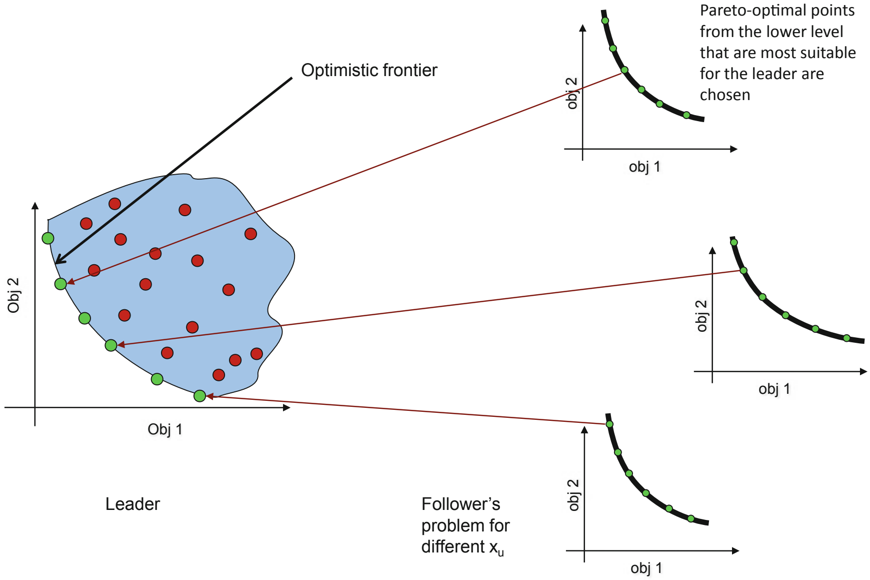

Fig. 10 Optimistic bilevel multiobjective optimization

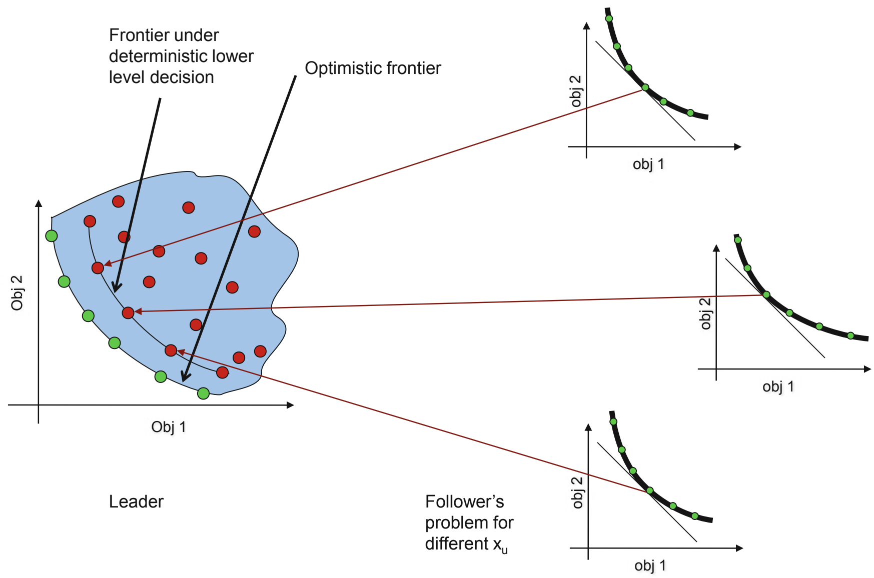

Fig. 11 Bilevel multiobjective optimization with deterministic lower level decisions

be modeled as a value function (see Fig. 11). Finally, we discuss a problem, where the lower level decision maker's preferences are not known with certainty and the upper level decision maker needs to take this decision-uncertainty into account when choosing her optimal strategy (see Fig. 12).

6.1 *Multiobjective Bilevel Optimization: The Optimistic Formulation*

Bilevel multiobjective optimization is a nested optimization problem involving two levels of multiobjective optimization tasks. The structure of a bilevel multiobjective problem demands that only the Pareto-optimal solutions to the lower level optimization problem may be considered as feasible solutions for the upper level optimization problem. There are two classes of variables in a bilevel optimization problem; namely, the upper level variables $x_u \in X_U \subset \mathbb{R}^n$, and the lower level variables $x_l \in X_L \subset \mathbb{R}^m$. The lower level multiobjective problem is solved with respect to the lower level variables, x_l, and the upper level variables, x_u act as parameters to the optimization problem. Each x_u corresponds to a different lower level optimization problem, leading to a different Pareto-optimal front. The upper level problem is optimized with respect to both classes of variables, $x = (x_u, x_l)$.

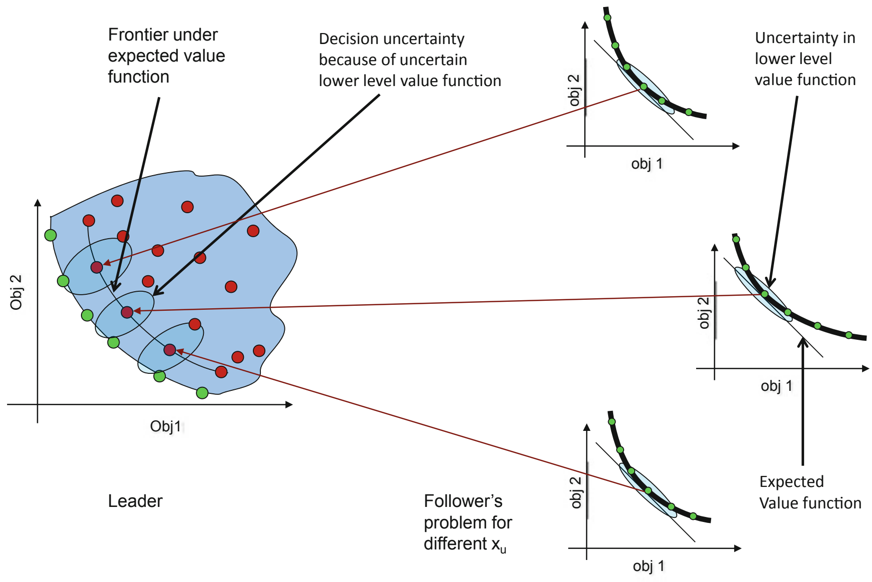

Fig. 12 Bilevel multiobjective optimization with uncertainty in lower level decisions

Definition 5 For the upper-level objective function $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ and lower-level objective function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^q$, the bilevel problem is given by

$$\begin{aligned}
&\min_{x_u \in X_U, x_l \in X_L} \; F(x_u, x_l) = (F_1(x_u, x_l), \ldots, F_p(x_u, x_l)) \\
&\text{subject to} \\
&\quad x_l \in \underset{x_l}{\operatorname{argmin}}\{f(x_u, x_l) = (f_1(x_u, x_l), \ldots, f_q(x_u, x_l)) : \\
&\qquad\qquad\qquad g_j(x_u, x_l) \le 0, \; j = 1, \ldots, J\} \\
&\quad G_k(x_u, x_l) \le 0, \; k = 1, \ldots, K
\end{aligned}$$

where $G_k : X_U \times X_L \to \mathbb{R}$, $k = 1, \ldots, K$ denote the upper level constraints, and $g_j : X_U \times X_L \to \mathbb{R}$ represent the lower level constraints, respectively. Equality constraints may also exist that have been avoided for brevity.

An equivalent formulation of the above problem can be stated in terms of set-valued mappings as follows:

Definition 6 Let $\Psi : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ be a set-valued mapping,

$$\begin{aligned}
\Psi(x_u) = \underset{x_l}{\operatorname{argmin}}\{f(x_u, x_l) = (f_1(x_u, x_l), \ldots, f_2(x_u, x_l)) : \\
g_j(x_u, x_l) \le 0, \; j = 1, \ldots, J\},
\end{aligned}$$

which represents the constraint defined by the lower-level optimization problem, i.e. $\Psi(x_u) \subset X_L$ for every $x_u \in X_U$. Then the bilevel multiobjective optimization problem can be expressed as a constrained multiobjective optimization problem:

$$\begin{aligned}\min_{x_u \in X_U, x_l \in X_L} \quad & F(x_u, x_l) = (F_1(x_u, x_l), \ldots, F_p(x_u, x_l)) \\ \text{subject to} \quad & x_l \in \Psi(x_u) \\ & G_k(x_u, x_l) \le 0, k = 1, \ldots, K\end{aligned}$$

where Ψ can be interpreted as a parameterized range-constraint for the lower-level decision vector x_l.

In the above two formulations, the lower level decision maker is assumed to cooperate with the upper level decision maker, such that she provides all Pareto-optimal points to the upper level decision maker who then chooses the best point according to the upper level objectives. The assumption effectively reduces the influence of the follower and transfers the decision-making power to the leader. Alternatively, one can say that the lower-level decision maker is assumed to be indifferent to all lower-level Pareto-optimal solutions. Though this formulation has been studied in the past, it is a highly unrealistic formulation where decision making aspects at the lower level are not taken into account.

Next, we demonstrate the optimistic formulation through a simple multiobjective bilevel optimization problem taken from [58].

Example 1 The problem has a single upper level and two lower level variables; such that $x_u = (x)$ and $x_l = (y_1, y_2)^T$. The formulation of the problem is given below:

$$\begin{aligned}&\text{Minimize } F(x, y_1, y_2) = \begin{pmatrix} y_1 - x \\ y_2 \end{pmatrix}, \\ &\text{subject to } (y_1, y_2) \in \underset{(y_1, y_2)}{\operatorname{argmin}} \left\{ f(x, y_1, y_2) = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \middle| g_1(x) = x^2 - y_1^2 - y_2^2 \ge 0 \right\}, \\ &\qquad G_1(x) = 1 + y_1 + y_2 \ge 0, \\ &\qquad -1 \le y_1, y_2 \le 1, \quad 0 \le x \le 1.\end{aligned} \tag{1}$$

The Pareto-optimal set for the lower level optimization task for a given x is the bottom-left quarter of the circle: $\{(y_1, y_2) \in \mathbb{R}^2 \mid y_1^2 + y_2^2 = x^2, y_1 \le 0, y_2 \le 0\}$. Lower level frontiers corresponding to different x are shown in Fig. 13. As observed from the figure, the linear constraint at the upper level does not allow the entire quarter circle to be feasible for some x. Therefore, at most two points from the quarter circle belong to the upper level Pareto-optimal set of the bilevel problem that is shown in Fig. 14. The lower level frontiers for different x are also plotted in the upper level objective space. Figures 13 and 14 also show three points A, B and C for $x = 0.9$, where points A and B participate in the upper level frontier while point C is rendered infeasible because of the upper level constraint. The analytical Pareto-optimal set for this problem is given as:

$$\left\{(x, y_1, y_2) \in \mathbb{R}^3 \mid x \in \left[\frac{1}{\sqrt{2}}, 1\right], y_1 = -1 - y_2, y_2 = -\frac{1}{2} \pm \frac{1}{4}\sqrt{8x^2 - 4}\right\}. \tag{2}$$

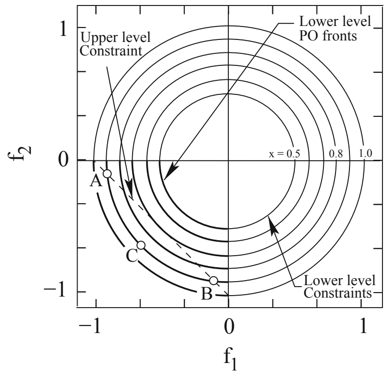

Fig. 13 Lower level Pareto-optimal fronts for different x_u in lower level objective space

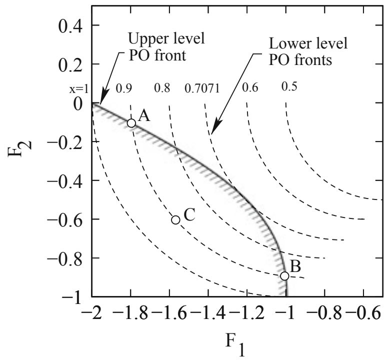

Fig. 14 Upper level Pareto-optimal front and lower level fronts in upper level objective space

This problem demonstrates that leader takes all the lower level Pareto-optimal solutions and then based on her constraints and non-domination criterion decides the solutions to be kept. Once this multiobjective bilevel problem is given, the upper level Pareto-frontier can be identified without considering any decision making aspects.

Some of the studies that attempted to handle the optimistic formulation are [57, 58] in the area of mathematical optimization and [61, 66] in the area of evolutionary computation. In [61], authors utilize a hierarchical evolutionary multiobjective optimization approach to solve a number of difficult multiobjective bilevel problems. Though the approach retains a nested structure, a number of intelligent schemes were employed that led to savings, when compared to a brute force nested algorithm. Some of the ideas utilized include; adjusting the number of subpopulations and their sizes for lower level search adaptively, solving the lower level problem with an evolutionary algorithm for a few generations and then employing local search on members that are likely to participate in upper level non-dominated frontier, utilizing a hypervolume-based termination criterion at both levels, and using archive that keeps those solutions that are feasible (with respect to constraints and lower level problem) and non-dominated at the upper level.

Table 7 Ratio of median function evaluations required by Algorithm-1 [61] against Algorithm-3 [66] and Algorithm-2 (purely nested) against Algorithm-3 [66]

Pr. No.	$\frac{Algorithm-1}{Algorithm-3}$		$\frac{Algorithm-2}{Algorithm-3}$	
	Total LL FE	Total UL FE	Total LL FE	Total UL FE
DS1	1.54	1.23	17.51	13.58
DS2	1.33	1.11	17.07	11.33
DS3	1.43	1.19	18.03	11.21
DS4	1.28	1.25	16.06	13.59
DS5	1.32	1.21	19.89	12.27

Table 8 Function evaluations (FE) required by Algorithm-3 for the upper level (UL) and lower level (LL)

Pr. No. (var.)	Best		Median		Worst	
	Total LL FE	Total UL FE	Total LL FE	Total UL FE	Total LL FE	Total UL FE
DS1 (20)	1946496	72334	2215966	74502	2430513	86697
DS2 (20)	3728378	93015	3728256	110006	4584177	126416
DS3 (20)	2540181	90754	3295798	100015	3733238	104025
DS4 (10)	904806	33804	1118631	42986	1339842	50686
DS5 (10)	1187359	38477	1356863	47071	1684170	59325

Best, median and worst values have been computed from 21 runs of the algorithm on each test problem. The lower level function evaluations include the evaluations of local search as well

Recently, along the lines of Ψ-mapping approximation, a multi-fiber approach has been proposed in [66]. In this approach the authors attempt to approximate the Ψ-mapping using multiple discrete fibers. The Ψ-mapping is more likely to be a (moving) set in the context of multiobjective bilevel optimization; therefore, ideas that can approximate sets have to be employed. This is one of the approaches the tries to exploit the structure and properties of the problem to solve it. The scheme can not be termed nested, but still requires solving some instances of the lower level problem to construct an approximation of the Ψ-mapping. In Table 7 we provide the results for three algorithms; algorithm 1 [61], algorithm 2 (purely nested) and algorithm 3 [66]; on a set of 5 test problems [61, 67]. The numbers in the table represent the ratio of function evaluations required by algorithm 1 and algorithm 2 with respect to algorithm 3. The function evaluations for algorithm 3 can be found in Table 8.

Before concluding the discussion on the optimistic formulations and solution procedures for multiobjective bilevel optimization, we would like to highlight that it is possible to write this formulation with multiple objectives at upper level and single objective at lower level. However, this comes at the cost of increased variables at the upper level. The following formulation has been known in mathematical optimization, but one of the first studies in the context of evolutionary optimization can be found in [68].

Definition 7 For a scalarizing function $S : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ with weight vector $w \in W \subset \mathbb{R}^p$

$$
\begin{aligned}
\min_{x_u \in X_U, x_l \in X_L, w \in W} \quad & F(x_u, x_l) \\
\text{subject to} \quad & x_l \in \underset{x_l \in X_L}{\operatorname{argmin}} \{ S(f(x_u, x_l), w) \\
& \qquad\qquad\qquad \text{subject to} \quad g_j(x_u, x_l) \leq 0, \; j = 1, \ldots, J \} \\
& G_k(x_u, x_l) \leq 0, k = 1, \ldots, K,
\end{aligned}
$$

where w acts as an upper level vector along with x_u.

It is important in the above formulation that the scalarizing function is able to span the entire lower level Pareto-optimal set through different values of w. The idea behind the formulation is that by changing w, one can select different Pareto-optimal solutions from the lower level corresponding to each upper level decision vector.

6.2 *Multiobjective Bilevel Optimization with Deterministic Decisions at Lower Level*

Considering the decision-making situations that arise in practice, a departure from the assumption of an indifferent lower level decision maker is necessary. Rather than providing all Pareto-optimal points to the leader, the follower is likely to act according to her own interests and choose the most preferred lower level solution herself. As a result, the allowance of lower level decision making has a substantial impact on the formulation of multiobjective bilevel optimization problems. First, the lower level problem can no longer be viewed as a range-constraint that depends only on lower-level objectives. Instead it is better interpreted as a selection function that maps a given upper level decision to a corresponding Pareto-optimal lower level solution that is most preferred by the follower. Second, in order to solve the bilevel problem, the upper level decision maker now needs to model the follower's behavior by anticipating her preferences towards different objectives. Naturally, these changes lead to a number of intricacies that were not encountered in the previous formulations. This formulation assumes perfect information to the leader about the follower's preference structure. Using the preference structure information it is possible to reduce the lower level problem into a single objective optimization problem [69].

Definition 8 Let $\xi \in \Xi$ denote a vector of parameters describing the follower's preferences. If the upper level decision maker has complete knowledge of the follower's preferences, the follower's actions can then be modeled via selection mapping

$$\sigma : X_U \times \Xi \to X_L, \quad \sigma(x_u, \xi) \in \Psi(x_u), \tag{3}$$

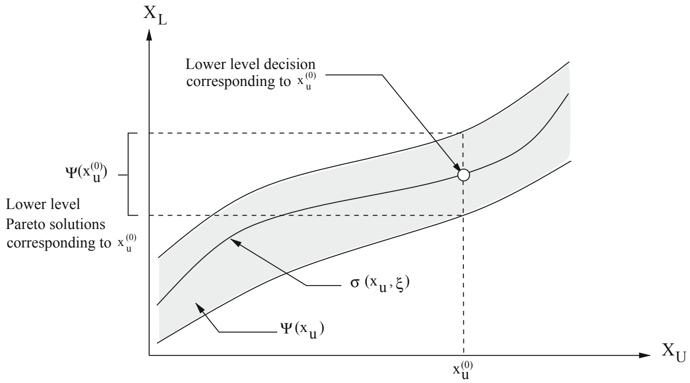

Fig. 15 Decision making under certainty

where Ψ is the set-valued mapping given by Definition 2. The resulting bilevel problem can be rewritten as follows:

$$\begin{aligned} \min_{x_u \in X_U} \quad & F(x_u, x_l) = (F_1(x_u, x_l), \ldots, F_p(x_u, x_l)) \\ \text{subject to} \quad & x_l = \sigma(x_u, \xi) \in \Psi(x_u) \\ & G_k(x_u, x_l) \leq 0, k = 1, \ldots, K \end{aligned} \tag{4}$$

To illustrate the definition, consider Fig. 15, where the shaded region

$$\operatorname{gph} \Psi = \{(x_u, x_l) \ : \ x_l \in \Psi(x_u)\} \tag{5}$$

represents the follower's Pareto-optimal solutions $\Psi(x_u)$ for any given leader's decision x_u. These are the rational reactions, which the follower may choose depending on her preferences. If the leader is aware of the follower's objectives, she will be able to identify the shaded region completely by solving the follower's multiobjective optimization problem for all x_u. However, if the follower is able to act according to her own preferences, she will choose only one preferred solution $\sigma(x_u, \xi)$ for every upper level decision x_u. When the preferences of the follower are perfectly known, the leader can identify $\sigma(\cdot, \xi)$ that characterizes follower's rational reactions for different x_u, and solve the hierarchical optimization task completely.

6.3 Multiobjective Bilevel Optimization with Lower Level Decision Uncertainty

The assumption that the follower's preferences are perfectly known to the leader itself might be an inaccurate description of real life scenarios. Most practitioners

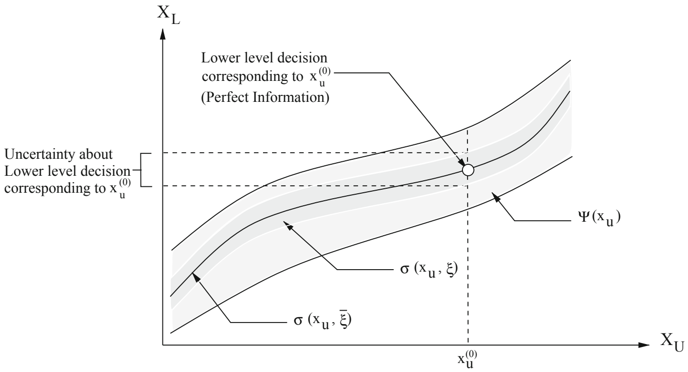

Fig. 16 Decision making under uncertainty

would find it hard to accept this even when constructing approximations. A natural path towards a more realistic framework would be to relax the axiom of perfect information by assuming that the leader is only partially aware of the follower's preferences. This lack of information leads to the notion of lower level decision uncertainty that is experienced by the leader while solving the bilevel optimization task [70].

For illustration, consider Fig. 16, where the expected behavior of the follower is shown as the graph of the selection mapping $\sigma(\cdot, \bar{\xi})$, where $\bar{\xi}$ represents the expected preference known to the leader. The narrow dark shaded band shows the region of uncertainty in which the follower makes her decisions. For different preferences ξ, $\sigma(\cdot, \xi)$ represents the corresponding decisions of the follower. If the leader is aware of the follower's objectives, the uncertainty region identified by a random ξ is always bounded by gph Ψ because $\sigma(x_u, \xi) \in \Psi(x_u)$ for all $x_u \in X_U$ and $\xi \in \Xi$. However, it is noteworthy that this band is not directly available to the leader but needs to be modeled. In a situation, where the leader cannot elicit follower's preferences by interacting with the follower, a feasible strategy is to utilize the prior information she has about the follower and incorporate it in a tractable stochastic model that characterizes the follower's behavior.

To accommodate the decision uncertainty, we assume that the follower's preferences are described by a random variable $\xi \sim \mathcal{D}_\xi$, which takes values in a set Ξ of $\mathbb{R}^q$. The probability distribution $\mathcal{D}_\xi$ reflects the leader's uncertainty and prior information about follower's expected behavior. In this framework, the assumption of preference uncertainty is equivalent to saying that the lower level decision is a random variable with a distribution that is parametrized by a given upper level decision x_u, i.e. $x_l \sim \mathcal{D}_\sigma(x_u)$. This means that the lower level decision uncertainty experienced by the leader will vary point-wise depending on the follower's objectives and the leader's own decision.

For demonstration of the uncertainty aspects in the objective spaces of the leader and the follower, consider Figs. 17 and 18 that show two different scenarios. In the first scenario, we assume a deterministic situation where the follower's prefer-

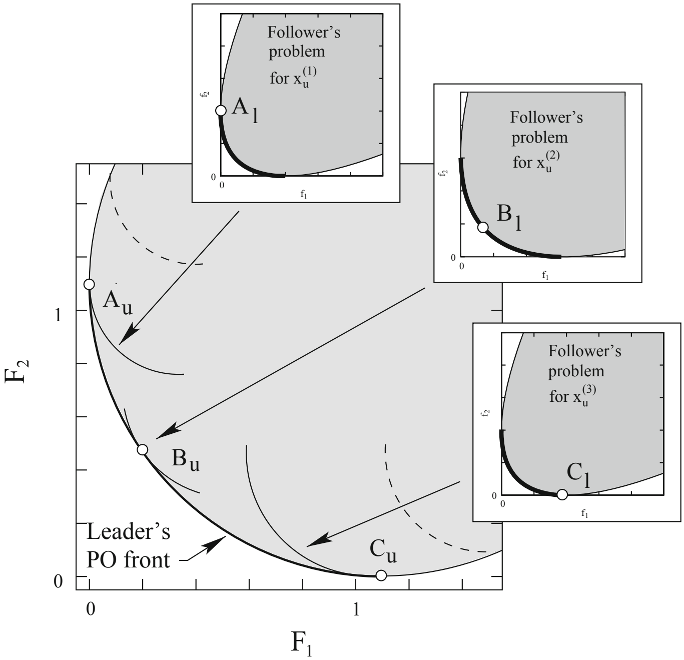

Fig. 17 Insets: follower's problem for different x_u. A_l, B_l and C_l represent the follower's decisions for $x_u^{(1)}$, $x_u^{(2)}$ and $x_u^{(3)}$ respectively. A_u, B_u and C_u are the corresponding points for the leader in the leader's objective space

ences and actions are known with certainty. Both leader and follower are assumed to have two objectives, i.e., $p = q = 2$. In this case, the leader solves the bilevel problem in Definition 8 under perfect information. Therefore, each point on the leader's Pareto-frontier corresponds to one of the points on the follower's Pareto-frontier. If $\bar{\xi}$ is the given vector of follower's preferences, then for any leader's choice $x_u^{(i)}$ the corresponding lower level decision is given by $x_l^{(i)} = \sigma(x_u^{(i)}, \bar{\xi})$. This is shown in Fig. 17, where the upper level points $A_u = F(x_u^{(1)}, \sigma(x_u^{(1)}, \bar{\xi}))$, $B_u = F(x_u^{(2)}, \sigma(x_u^{(2)}, \bar{\xi}))$, and $C_u = F(x_u^{(3)}, \sigma(x_u^{(3)}, \bar{\xi}))$ are paired with the points $A_l = f(x_u^{(1)}, \sigma(x_u^{(1)}, \bar{\xi}))$, $B_l = f(x_u^{(2)}, \sigma(x_u^{(2)}, \bar{\xi}))$ and $C_l = f(x_u^{(3)}, \sigma(x_u^{(3)}, \bar{\xi}))$ that lie on the follower's Pareto-front for $x_u^{(1)}$, $x_u^{(2)}$, and $x_u^{(3)}$, respectively.

The situation can be contrasted from another scenario shown in Fig. 18, where the follower's preferences are uncertain. The leader is still assumed to be fully aware of the form of σ, but she no longer knows the true value of ξ. By assuming a prior information $\xi \sim \mathcal{D}_\xi$, the leader can attempt to solve the bilevel problem based on the expected preferences of the follower, i.e.

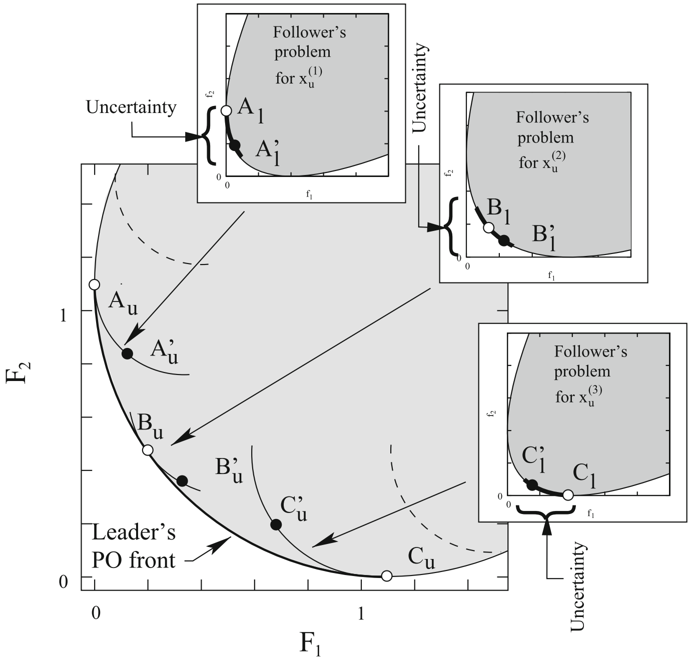

Fig. 18 Insets: follower's problem for different x_u. A_l, B_l and C_l are the expected decisions of the follower. $A_l^{'}$, $B_l^{'}$ and $C_l^{'}$ are the actual decisions that the follower takes. The corresponding points for the leader are shown in the leader's objective space

$$\begin{aligned} \min_{x_u \in X_U} \quad & F(x_u, \bar{x}_l) \\ \text{subject to} \quad & \bar{x}_l = \sigma(x_u, E[\xi]) \in \Psi(x_u), \quad \xi \sim \mathcal{D}_\xi \\ & G_k(x_u, \bar{x}_l) \leq 0, k = 1, \ldots, K. \end{aligned} \tag{6}$$

For convenience of the example, we assume that the expected actions are the same as the actions in Fig. 17, i.e., $\sigma(x_u, E[\xi])] = \sigma(x_u, \bar{\xi})$ for all x_u. As a result, the leader obtains a *Pareto-frontier corresponding to the follower's expected value function* (POF-EVF). However, when she begins to implement the given strategies, the follower's realized actions may deviate from the expected strategies obtained by solving (6). Since ξ is uncertain from the leader's perspective, the follower's true preferences ξ can differ from $\bar{\xi}$ that was expected based on prior information. As shown in the figure, for any strategy $x_u^{(1)}$, $x_u^{(2)}$ or $x_u^{(3)}$ chosen by the leader, the follower may prefer to choose $A_l^{'}$, $B_l^{'}$ or $C_l^{'}$ instead of A_l, B_l or C_l expected by the leader. It is found that because of the follower's deviation from the expected actions, the leader no longer operates on the POF-EVF. In the objective space, the uncertainty experienced by the leader is reflected in the probability and size of deviations away from the POF-EVF. The follower, on the other hand, does not experience similar uncertainty, because she can always observe the action taken by the leader before making her own decision.

Depending on the problem, uncertainty of the lower level decision maker's preferences may lead to significant losses at the upper level. Therefore, the leader would like to solve the bilevel problem taking the uncertainties into account. While making a decision, the leader might prefer those regions on its frontier, which are less sensitive to lower level uncertainties and at the same time offer an acceptable trade-off between the objectives. For instance, in the context of the above example, we observe that the expected variation in the objective space is considerably less at the region corresponding to $x_u^{(2)}$ than at $x_u^{(1)}$ or $x_u^{(3)}$. If the leader chooses this point, she knows that the realized upper level objective values are only little affected by the actions of the lower level decision maker. From the perspective of practical decision making, it is valuable for the leader to be aware of the level of uncertainty associated with different strategies.

7 Future Research Directions

In this chapter, we have tried to provide an introduction to the work done in the area of bilevel optimization using evolutionary algorithms. The main topics covered include;

1. Single objective bilevel optimization and promising ideas that might be useful in solving complex bilevel problems.
2. Multiobjective bilevel optimization methods and decision making intricacies.

While the above two topics themselves offer significant opportunity of future research, there also exist other areas within bilevel optimization that are less explored and offer potential for future research. For instance, there can be other forms of uncertainties in bilevel optimization, like, variable and parameter uncertainties. Some preliminary work on these topics can be found in [71, 72]. With an increase in computational power, there is an enormous scope of development of distributed computing methods that can solve bilevel problems with large number of variables or objectives in a short time. However, at this point it is worth mentioning that in the last decade a number of evolutionary algorithms have been developed that are computationally very expensive and purely nested. Future research ideas on evolutionary computation should rely also on exploiting the structure and properties of bilevel problems, which will ensure better scalability of the procedures. To conclude, almost every other discipline faces application problems that are bilevel in nature. This offers application oriented research opportunities both from modeling and solution perspectives.

References

1. Von Stackelberg, H.: The Theory of the Market Economy. Oxford University Press, New York (1952)
2. Bracken, J., McGill, J.T.: Mathematical programs with optimization problems in the constraints. Op. Res. **21**(1), 37–44 (1973)
3. Wen, U.-P., Hsu, S.-T.: Linear bi-level programming problems–a review. J. Op. Res. Soc. 125–133 (1991)
4. Ben-Ayed, O.: Bilevel linear programming. Comput. Op. Res. **20**(5), 485–501 (1993)
5. Bard, J.F., Moore, J.T.: A branch and bound algorithm for the bilevel programming problem. SIAM J. Sci. Stat. Comput. **11**(2), 281–292 (1990)
6. Edmunds, T.A., Bard, J.F.: Algorithms for nonlinear bilevel mathematical programs. IEEE Trans. Syst. Man Cybern. **21**(1), 83–89 (1991)
7. Al-Khayyal, F.A., Horst, R., Pardalos, P.M.: Global optimization of concave functions subject to quadratic constraints: an application in nonlinear bilevel programming. Ann. Op. Res. **34**(1), 125–147 (1992)
8. Liu, G., Han, J., Wang, S.: A trust region algorithm for bilevel programing problems. Chin. Sci. Bull. **43**(10), 820–824 (1998)
9. 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)
10. Vicente, L., Savard, G., Júdice, J.: Descent approaches for quadratic bilevel programming. J. Optim. Theory Appl. **81**(2), 379–399 (1994)
11. Brown, G., Carlyle, M., Diehl, D., Kline, J., Wood, K.: A two-sided optimization for theater ballistic missile defense. Op. Res. **53**(5), 745–763 (2005)
12. Wein, L.M.: Or forum-homeland security: From mathematical models to policy implementation: the 2008 philip mccord morse lecture. Op. Res. **57**(4), 801–811 (2009)
13. An, B., Ordóñez, F., Tambe, M., Shieh, E., Yang, R., Baldwin, C., DiRenzo III, J., Moretti, K., Maule, B., Meyer, G.: A deployed quantal response-based patrol planning system for the us coast guard. Interfaces **43**(5), 400–420 (2013)
14. Labbé, M., Marcotte, P., Savard, G.: A bilevel model of taxation and its application to optimal highway pricing. Manag. Sci. **44**(12), 1608–1622 (1998). part-1
15. Sinha, A., Malo, P., Frantsev, A., Deb, K.: Multi-objective stackelberg game between a regulating authority and a mining company: a case study in environmental economics. In: 2013 IEEE Congress on Evolutionary Computation (CEC), pp. 478–485. IEEE (2013)
16. Sinha, A., Malo, P., Deb, K.: Transportation policy formulation as a multi-objective bilevel optimization problem. In: 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 1651–1658. IEEE (2015)
17. Nicholls, M.G.: Aluminum production modelingâĂŤa nonlinear bilevel programming approach. Op. Res. **43**(2), 208–218 (1995)
18. Hu, X., Ralph, D.: Using epecs to model bilevel games in restructured electricity markets with locational prices. Op. Res. **55**(5), 809–827 (2007)
19. Williams, N., Kannan, P., Azarm, S.: Retail channel structure impact on strategic engineering product design. Manag. Sci. **57**(5), 897–914 (2011)
20. Migdalas, A.: Bilevel programming in traffic planning: models, methods and challenge. J. Glob. Optim. **7**(4), 381–405 (1995)
21. Constantin, I., Florian, M.: Optimizing frequencies in a transit network: a nonlinear bi-level programming approach. Int. Trans. Op. Res. **2**(2), 149–164 (1995)
22. Brotcorne, L., Labbé, M., Marcotte, P., Savard, G.: A bilevel model for toll optimization on a multicommodity transportation network. Transp. Sci. **35**(4), 345–358 (2001)
23. Sun, H., Gao, Z., Wu, J.: A bi-level programming model and solution algorithm for the location of logistics distribution centers. Appl. Math. Model. **32**(4), 610–616 (2008)
24. Bard, J.F.: Coordination of a multidivisional organization through two levels of management. Omega **11**(5), 457–468 (1983)

25. Jin, Q., Feng, S.: Bi-level simulated annealing algorithm for facility location. Syst. Eng. **2**, 007 (2007)
26. Uno, T., Katagiri, H., Kato, K.: An evolutionary multi-agent based search method for stackelberg solutions of bilevel facility location problems. Int. J. Innov. Comput. Inf. Control **4**(5), 1033–1042 (2008)
27. Smith, W.R., Missen, R.W.: Chemical reaction equilibrium analysis: theory and algorithms, Wiley, xvi+ 364, 23 x 15 cm, illustrated (1982)
28. Clark, P.A., Westerberg, A.W.: Bilevel programming for steady-state chemical process designâĂŤi. fundamentals and algorithms. Comput. Chem. Eng. **14**(1), 87–97 (1990)
29. Bendsoe, M.P.: Optimization of Structural Topology, Shape, and Material, vol. 2. Springer, Berlin (1995)
30. Snorre, C., Michael, P., Wynter, L.: Stochastic bilevel programming in structural optimization (1997)
31. Mombaur, K., Truong, A., Laumond, J.-P.: From human to humanoid locomotionâĂŤan inverse optimal control approach. Auton. Robots **28**(3), 369–383 (2010)
32. Albrecht, S., Ramirez-Amaro, K., Ruiz-Ugalde, F., Weikersdorfer, D., Leibold, M., Ulbrich, M., Beetz, M.: Imitating human reaching motions using physically inspired optimization principles. In: 2011 11th IEEE-RAS International Conference on Humanoid Robots (Humanoids), pp. 602–607. IEEE (2011)
33. Bäck, T.: Evolutionary algorithms in theory and practice (1996)
34. Mathieu, R., Pittard, L., Anandalingam, G.: Genetic algorithm based approach to bi-level linear programming, Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle **28**(1), 1–21 (1994)
35. Yin, Y.: Genetic-algorithms-based approach for bilevel programming models. J. Transp. Eng. **126**(2), 115–120 (2000)
36. Li, X., Tian, P., Min, X.: A hierarchical particle swarm optimization for solving bilevel programming problems. Artif. Intell. Soft Comput.–ICAISC 2006, pp. 1169–1178 (2006)
37. Li, H., Wang, Y.: A hybrid genetic algorithm for solving nonlinear bilevel programming problems based on the simplex method. In: ICNC 2007 Third International Conference on Natural Computation, vol. 4, pp. 91–95. IEEE (2007)
38. Zhu, X, Yu Q., Wang, X.: A hybrid differential evolution algorithm for solving nonlinear bilevel programming with linear constraints. In: ICCI 2006 5th IEEE International Conference on Cognitive Informatics, vol. 1, pp. 126–131. IEEE (2006)
39. Sinha, A., Malo, P., Frantsev, A., Deb, K.: Finding optimal strategies in a multi-period multi-leader-follower stackelberg game using an evolutionary algorithm. Comput. Op. Res. **41**, 374–385 (2014)
40. Angelo, J.S., Krempser, E., Barbosa, H.J.: Differential evolution for bilevel programming. In: 2013 IEEE Congress on Evolutionary Computation (CEC), pp. 470–477. IEEE (2013)
41. Angelo, J.S., Barbosa, H.J.: A study on the use of heuristics to solve a bilevel programming problem. Int. Trans. Op. Res. **22**(5), 861–882 (2015)
42. Hejazi, S.R., Memariani, A., Jahanshahloo, G., Sepehri, M.M.: Linear bilevel programming solution by genetic algorithm. Comput. Op. Res. **29**(13), 1913–1925 (2002)
43. Wang, Y., Jiao, Y.-C., Li, H.: An evolutionary algorithm for solving nonlinear bilevel programming based on a new constraint-handling scheme. IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev. **35**(2), 221–232 (2005)
44. Wang, Y., Li, H., Dang, C.: A new evolutionary algorithm for a class of nonlinear bilevel programming problems and its global convergence. INFORMS J. Comput. **23**(4), 618–629 (2011)
45. Jiang, Y., Li, X., Huang, C., Wu, X.: Application of particle swarm optimization based on chks smoothing function for solving nonlinear bilevel programming problem. Appl. Math. Comput. **219**(9), 4332–4339 (2013)
46. Li, H.: A genetic algorithm using a finite search space for solving nonlinear/linear fractional bilevel programming problems. Ann. Op. Res. **235**(1), 543–558 (2015)

47. Wan, Z., Wang, G., Sun, B.: A hybrid intelligent algorithm by combining particle swarm optimization with chaos searching technique for solving nonlinear bilevel programming problems. Swarm Evolut. Comput. **8**, 26–32 (2013)
48. Sinha, A., Malo, P., Deb, K.: Efficient evolutionary algorithm for single-objective bilevel optimization. arXivpreprintXiv arXiv:1303.3901 (2013)
49. Sinha, A., Malo, P., Deb, K.: An improved bilevel evolutionary algorithm based on quadratic approximations. In: 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 1870–1877. IEEE (2014)
50. Angelo, J.S., Krempser, E., Barbosa, H.J.: Solving optimistic bilevel programs by iteratively approximating lower level optimal value function. In: 2013 IEEE Congress on Evolutionary Computation (CEC), pp. 470–477. IEEE (2013)
51. Bialas, W.F., Karwan, M.H.: Two-level linear programming. Manag. Sci. **30**(8), 1004–1020 (1984)
52. Chen, Y., Florian, M.: On the geometric structure of linear bilevel programs: a dual approach. Centre de Recherche sur les Transports **867** (1992)
53. Tuy, H., Migdalas, A., Värbrand, P.: A global optimization approach for the linear two-level program. J. Glob. Optim. **3**(1), 1–23 (1993)
54. Bard, J.F., Falk, J.E.: An explicit solution to the multi-level programming problem. Comput. Op. Res. **9**(1), 77–100 (1982)
55. Fortuny-Amat, J., McCarl, B.: A representation and economic interpretation of a two-level programming problem. J. Op. Rese. Soc. 783–792 (1981)
56. Ye, J.J., Zhu, D.: New necessary optimality conditions for bilevel programs by combining the mpec and value function approaches. SIAM J. Optim. **20**(4), 1885–1905 (2010)
57. Eichfelder, G.: Solving nonlinear multiobjective bilevel optimization problems with coupled upper level constraints. Inst. für Angewandte Mathematik (2007)
58. Eichfelder, G.: Multiobjective bilevel optimization. Math. Program. **123**(2), 419–449 (2010)
59. Shi, X., Xia, H.S.: Model and interactive algorithm of bi-level multi-objective decision-making with multiple interconnected decision makers. J. Multi-Criteria Decis. Anal. **10**(1), 27–34 (2001)
60. Halter, W., Mostaghim, S.: Bilevel optimization of multi-component chemical systems using particle swarm optimization. In: CEC 2006 IEEE Congress on Evolutionary Computation, pp. 1240–1247. IEEE (2006)
61. Deb, K., Sinha, A.: An efficient and accurate solution methodology for bilevel multi-objective programming problems using a hybrid evolutionary-local-search algorithm. Evol. Comput. **18**(3), 403–449 (2010)
62. Sinha, A.: Bilevel multi-objective optimization problem solving using progressively interactive emo. In: Evolutionary Multi-Criterion Optimization, pp. 269–284. Springer (2011)
63. Lai, Y.-J.: Hierarchical optimization: a satisfactory solution. Fuzzy Sets Syst. **77**(3), 321–335 (1996)
64. Sakawa, M., Nishizaki, I.: Interactive fuzzy programming for decentralized two-level linear programming problems. Fuzzy Sets Syst. **125**(3), 301–315 (2002)
65. Chen, L.-H., Chen, H.-H.: Considering decision decentralizations to solve bi-level multi-objective decision-making problems: a fuzzy approach. Appl. Math. Model. **37**(10), 6884–6898 (2013)
66. Sinha, A., Malo, P., Deb, K.: Approximated set-valued mapping approach for handling multi-objective bilevel problems. In: Working paper. IEEE (2016)
67. Deb, K., Sinha, A.: Constructing test problems for bilevel evolutionary multi-objective optimization. In: 2009. CEC'09 IEEE Congress on Evolutionary Computation, pp. 1153–1160. IEEE (2009)
68. Gupta, A., Ong, Y.-S.: An evolutionary algorithm with adaptive scalarization for multiobjective bilevel programs. In: 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 1636–1642. IEEE (2015)
69. Sinha, A., Malo, P., Deb, K.: Towards understanding bilevel multi-objective optimization with deterministic lower level decisions. In: Evolutionary Multi-Criterion Optimization. pp. 426–443. Springer (2015)

70. Sinha, A., Malo, P., Deb, K., Korhonen, P., Wallenius, J.: Solving bilevel multi-criterion optimization problems with lower level decision uncertainty (2015)
71. Lu, Z., Deb, K., Sinha, A.: Handling decision variable uncertainty in bilevel optimization problems. In: 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 1683–1690. IEEE (2015)
72. Deb, K., Lu, Z., Sinha. A.: Finding reliable solutions in bilevel optimization problems under uncertainties. In: 18th Annual Conference on Genetic and Evolutionary Computation, 2016. GECCO 2016. IEEE (2016)