Towards Understanding Bilevel Multi-objective Optimization with Deterministic Lower Level Decisions

Ankur Sinha Ankur.Sinha@aalto.fi
Department of Information and Service Economy, Aalto University School of Business
PO Box 21210, FIN-00076 Aalto, Helsinki, Finland
Pekka Malo Pekka.Malo@aalto.fi
Department of Information and Service Economy, Aalto University School of Business
PO Box 21210, FIN-00076 Aalto, Helsinki, Finland
Kalyanmoy Deb Kdeb@egr.msu.edu
Department of Electrical and Computer Engineering, Michigan State University
East Lansing MI 48824, USA

COIN Report Number 2014019

Abstract

Bilevel decision making and optimization problems are commonly framed as leader-follower problems, where the leader desires to optimize her own decision taking the decisions of the follower into account. These problems are known as Stackelberg problems in the domain of game theory, and as bilevel problems in the domain of mathematical programming. In a number of practical scenarios, both the leaders and the followers might be faced with multiple criteria bringing bilevel multi-criteria decision making aspects into the problem. In such cases, the Pareto-optimal frontier of the leader is influenced by the decision structure of the follower facing multiple objectives. In this paper, we analyze this effect by modeling the lower level decision maker using value functions. We study the problem using test cases and propose a technique that can be used to solve such problems.

Keywords

Stackelberg game, Bilevel optimization, Multi-objective optimization, Evolutionary algorithms, Quadratic approximations.

1 Introduction

Bilevel optimization problems have been widely studied by both researchers as well as practitioners. The work has been driven by a number of applications that are bilevel in nature; for instance in transportation (network design, optimal pricing) (1; 2), economics (Stackelberg games, principal-agent problem, policy decisions) (3; 4; 5; 6), management (network facility location, coordination of multi-divisional firms) (7; 8), engineering (optimal design, optimal chemical equilibria) (9; 10). The recent methodological and practical developments on bilevel optimization have been mostly directed towards problems with single objective at both levels. Apart from a few studies in classical optimization (11; 12) and evolutionary optimization (13; 14; 15), little work has been done in the domain of multi-objective bilevel optimization. Most of these studies have not considered decision making intricacies that can arise from hierarchical decision interactions in the presence of multiple objectives.

While solving a bilevel optimization problem with multiple objectives at both levels, many a times the assumption is that the follower has little decision making power. This means the follower allows the leader to utilize any solution from her (follower's) frontier. This is an optimistic assumption and is often not realistic. A leader may anticipate the decisions of a follower and optimize her decisions accordingly, but it is unrealistic to assume that the leader can choose the solutions best suited to her from the follower's frontier. In this paper, we study cases where the lower level decision maker has sufficient power to make a deterministic decision from her own frontier. We analyze what kind of an impact deterministic lower level decisions have on the upper level frontier. We also highlight issues that need further attention.

To begin with, we provide a review of some of the recent work on multi-objective bilevel optimization. This is followed by the description of multi-objective bilevel optimization with and without decision making at the lower level. The lower level decision making aspects and its impact on the upper level Pareto-frontier are analyzed using two test problems. Thereafter, we propose an evolutionary algorithm to solve such multi-objective bilevel optimization problems where the lower level decisions are determined by a value function. The performance of the evolutionary algorithm is evaluated on test problems and comparisons have been drawn with an earlier approach (15).

2 Past Studies on Multi-objective Bilevel Optimization

There exists a significant amount of work on single objective bilevel problems, but little has been done on multi-objective bilevel problems primarily because of the computational and decision making complexities that such problems offer. In this section, we highlight the few studies available on multi-objective bilevel optimization. Studies by (11; 12) utilize classical techniques to handle simple multi-objective 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 are solved to Pareto-optimality. In the study by (14), the authors use ϵ-constraint method at both levels of multi-objective 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.

One of the first studies, utilizing an evolutionary approach for bilevel multi-objective algorithms was by (16). The study involved multiple objectives at the upper lever, 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 (13) used a particle swarm optimization 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 multi-objective PSO approach. Recently, a hybrid bilevel evolutionary multi-objective optimization algorithm approach coupled with local search was proposed in (15) (For earlier versions, refer (17; 18; 19; 20)). In the paper, the authors handled non-linear as well as discrete bilevel problems with relatively larger number of variables. The study also provided a suite of test problems for bilevel multi-objective optimization. Other recent work related to bilevel multi-objective optimization can be found in (21; 22; 23; 24; 25). There has been some work done on decision making aspects primarily at the upper level. For example, in (26) the authors propose interaction with the upper level decision maker during optimization to find the most preferred point instead of the entire Pareto-frontier. Since multi-objective bilevel optimization is computationally expensive, such an approach was justified as it led to enormous savings in computational expense. However, decision making at the lower level was ignored in this study.

3 Bilevel Multi-objective Optimization and Decision Making

In this section, we provide different formulations for a bilevel multi-objective optimization problem that contains two levels of optimization. The upper level optimization problem is the leader's problem (upper level decision maker) and the lower level optimization problem is the follower's problem (lower level decision maker). First, we consider a formulation, where there is no decision making involved at the lower level and all lower level Pareto-optimal solutions are considered at the upper level. Second, we consider a formulation, where the decision maker acts at the lower level and chooses a solution to her liking. This becomes the only possible feasible solution at the upper level.

3.1 Bilevel Multi-objective Optimization

Bilevel multi-objective optimization problem contains two levels of multi-objective optimization tasks. There are two types of variables in these problems; 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 multi-objective problem is solved with respect to the lower level variables, while the upper level variables act as parameters to the optimization problem. The optimistic formulation of such problems requires that the Pareto-optimal solutions of the lower level optimization problem may be considered as possible feasible solutions for the upper level optimization problem. Below, we provide two equivalent definitions of a bilevel multi-objective optimization problem.

Definition 1. *For the upper-level objective function $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ and lower-level objective function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^q$*

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

The above definition can be stated in terms of set-valued mappings as follows:

Definition 2. *Let $\Psi : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ be a set-valued mapping,*

$$\Psi(x_u) = \textit{argmin}_{x_l} \{ f(x_u, x_l) = (f_1(x_u, x_l), \ldots, f_2(x_u, x_l)) : \quad g_j(x_u, x_l) \le 0, j = 1, \ldots, J \},$$

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 multi-objective optimization problem can be expressed as a constrained multi-objective optimization problem as follows:

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

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

3.2 Decision Making at Lower Level

According to the formulation of a multi-objective bilevel problem in the previous sub-section, the follower provides all Pareto-optimal points to the leader, who chooses the most suitable point in accordance with the upper level objectives. However, this is rarely the case, as in reality it might often happen that the follower is interested in optimizing her own objectives and making her own decision for a given upper level vector. If the leader wants to solve such a problem where the follower has sufficient decision making power, then she needs to have a complete knowledge of the follower's decision structure. The decision structure of the follower may be represented in the form of a value function. If the value function of the lower level decision maker is known, then such an optimization problem can be formulated as follows:

Definition 3. *For the upper-level objective function* $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ *and lower-level objective function* $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^q$

$$\begin{aligned}
\underset{x_u \in X_U, x_l \in X_L}{\text{minimize}} \quad & F(x_u, x_l) = (F_1(x_u, x_l), \ldots, F_p(x_u, x_l)) \\
\text{subject to} \quad & x_l \in \textit{argmin}_{x_l}\{V(f_1(x_u, x_l), \ldots, f_q(x_u, x_l); \omega) : \;\; g_j(x_u, x_l) \leq 0, j = 1, \ldots, J\} \\
& G_k(x_u, x_l) \leq 0, k = 1, \ldots, K,
\end{aligned}$$

where V denotes the follower's value function, and ω is the parameter vector of the assumed value function form. For instance, if V is linear, such that $V(f_1(x_u, x_l), \ldots, f_q(x_u, x_l); \omega) = \sum_{i=1}^{q} \omega_i f_i(x_u, x_l)$, then $\omega_i \; \forall \; i \in \{1, \ldots, q\}$ represent the value function parameters.

If it is assumed that the lower level decision maker always returns a single point for a given x_u, then the definition gets modified as follows:

Definition 4. *For the upper-level objective function* $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ *and lower-level objective function* $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^q$

$$\begin{aligned}
\underset{x_u \in X_U, x_l \in X_L}{\text{minimize}} \quad & F(x_u, x_l) = (F_1(x_u, x_l), \ldots, F_p(x_u, x_l)) \\
\text{subject to} \quad & x_l = \textit{argmin}_{x_l}\{V(f_1(x_u, x_l), \ldots, f_q(x_u, x_l)) : \;\; g_j(x_u, x_l) \leq 0, j = 1, \ldots, J\} \\
& G_k(x_u, x_l) \leq 0, k = 1, \ldots, K
\end{aligned}$$

In this paper, we aim to solve the problem formulated above. We assume that the leader has a complete knowledge of the follower's value function. Based on this information, we solve the bilevel problem to identify the upper level Pareto-frontier. Once the upper level Pareto-frontier is available to the leader, it becomes a multi-criteria decision making problem for the leader that we do not consider in this paper.

4 A Graphical Representation for Bilevel Multi-objective Optimization with Lower Level Decisions

Bilevel optimization problems are known to be computationally demanding. However, in case of multiple objectives at both levels of a bilevel optimization problem, an additional difficulty enters because the decision making aspects need to be considered. Even though the upper level decision maker is aware of the objectives of the lower level decision maker, she has little idea about the decisions the lower level decision maker might make from a multitude of lower level Pareto-optimal solutions. In order to handle the problem, the upper level decision maker needs to identify the preference structure of the lower level decision maker through studies or surveys.

Figure 1 shows the scenario, where the shaded region ($\Psi(x_u)$) represents the follower's Pareto-optimal solution for any given leader's decision (x_u). These are the rational actions, which the follower may make for a given leader's action. If the leader is aware of the follower's objectives, she will be able to identify the shaded region completely by solving the multi-objective optimization problem for the follower for all x_u. However, information about the follower's preferences on the lower level Pareto-optimal solutions is required by the leader to make an appropriate decision. If the preferences of the follower are perfectly known, then the lower level decision for any x_u is given by $\sigma(x_u)$, shown in the figure. In such a case, it is possible for the leader to solve the hierarchical optimization task completely, only when $\sigma(x_u)$ is available.

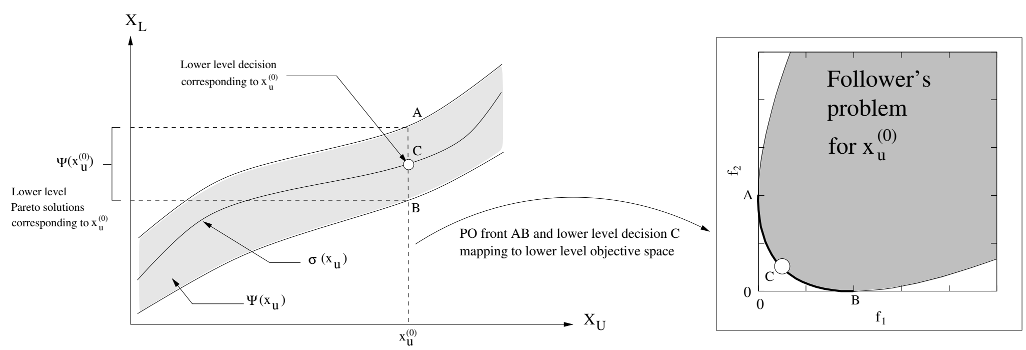

Fig. 1. Lower level Pareto-optimal solutions ($\psi(x_u)$) and corresponding decisions $\sigma(x_u)$.

Next, we look at the multi-objective bilevel problem in the objective spaces of the leader and the follower. In the case when the leader solves the bilevel problem taking into account the actions of the follower, each point on the leader's Pareto-frontier corresponds to one of the points on the follower's Pareto-frontier. This has been shown in Figure 2, where points A_u, B_u and C_u are realized when the follower's choices are A_l, B_l and C_l. Points A_l, B_l and C_l lie on the lower level Pareto-optimal front corresponding to $x_u^{(1)}$, $x_u^{(2)}$ and $x_u^{(3)}$ respectively. If the follower decides to use a different preference structure, the Pareto-frontier for the leader may change. It may happen that the Pareto-frontier at the upper level improves, deteriorates or does not change. It may also happen that with change in the follower's preferences, upper level points $x_u^{(1)}$, $x_u^{(2)}$ and $x_u^{(3)}$ are no longer a part of the upper level frontier. Therefore, the Pareto-optimal solutions at the upper level are entirely dependent on the decision structure of the follower. In the next section, using examples we compare the leader's frontier corresponding to a follower with sufficient decision power and a follower with no decision power.

5 Examples

In this section, we consider two examples from the literature (11; 15) and solve the problem analytically to show how the frontier at the upper level changes when the lower level decision maker exercises her decisions. A comparison has been made against the scenario when no lower level decision making is performed.

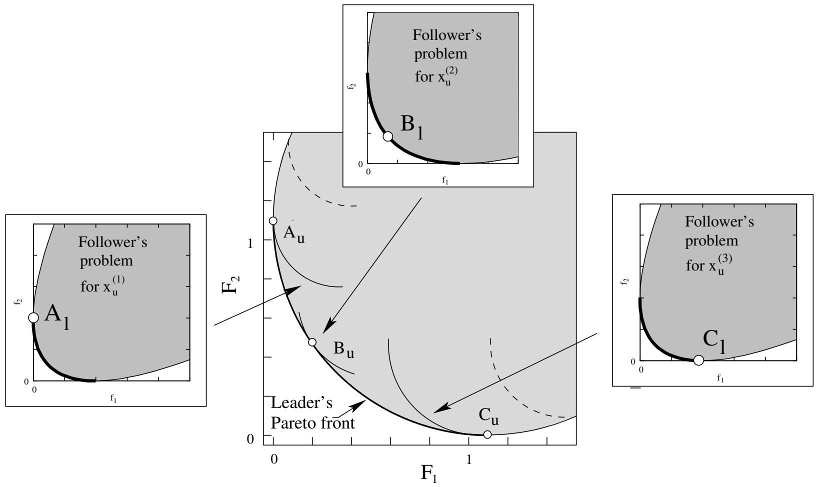

Fig. 2. The small figures show the follower's problem for different x_u. When the follower's preference structure is known, the leader optimizes the bilevel problem such that the follower's decisions corresponds to the leader's Pareto-frontier. A_l, B_l and C_l represent the follower's decisions for $x_u^{(1)}$, $x_u^{(2)}$ 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.

Example 1

Consider the following bilevel multi-objective optimization problem (11) with two objectives at each level. It contains three variables with y_1, y_2 belonging to x_l and x belonging to x_u.

$$\begin{aligned} &\text{minimize} \;\; F(x, y_1, y_2) = \begin{Bmatrix} y_1 - x \\ y_2 \end{Bmatrix}, \\ &\text{subject to} \;(y_1, y_2) \in \text{argmin}_{(y_1,y_2)} \left\{ f(x, y_1, y_2) = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \middle| g_1(x, y_1, y_2) = x^2 - y_1^2 - y_2^2 \geq 0 \right\}, \\ &\qquad G_1(y_1, y_2) = 1 + y_1 + y_2 \geq 0, \\ &\qquad -1 \leq y_1, y_2 \leq 1, \quad 0 \leq x \leq 1. \end{aligned} \tag{1}$$

For any fixed value of x, the feasible region of the lower-level problem is the area inside a circle with center at origin ($y_1 = y_2 = 0$) and radius equal to x. The Pareto-optimal set for the lower-level optimization task for a fixed x is the south-west quarter of the circle:

$$\{(y_1, y_2) \in \mathbf{R}^2 \quad | \quad y_1^2 + y_2^2 = x^2, y_1 \leq 0, y_2 \leq 0\}.$$

.

Let us first consider the upper level frontier with no lower level decision making. This represents the best possible frontier at the upper level as all the lower level Pareto-optimal members are available to the upper level decision maker. It is an ideal scenario for the upper level decision maker, where she freely chooses a suitable point from the lower level frontier. For the above example, such an upper level frontier is shown in Figure 3. The upper level Pareto-optimal set for this scenario can be generated as follows:

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

From the Figure 3 it is clear that at most two members from the lower level frontiers corresponding to $x \in [\sqrt{0.5}, 1]$ participate in the upper level front. Note that for $x = 0.9$ points B and C are Pareto-optimal at the upper level and point A is infeasible because of the upper level constraint.

Next, let us consider the problem in the context of this paper, where the lower level decision maker has sufficient power to choose a point from her Pareto-optimal front. If one assumes a particular value function, say $V(f_1, f_2) = 5x^2 f_1 + f_2$, then the upper level frontier is given as shown in Figure 4. It is noteworthy that the assumed value function also contains x. This kind of dependency may not always exist, but has been considered here to show that the lower level value function may take any possible form. The theoretical upper level Pareto-optimal frontier corresponding to the deterministic lower level value function is theoretically given as:

$$V(f_1, f_2) = 5x^2 f_1 + f_2, \text{then } y_1 = -\sqrt{\frac{25x^6}{(1+25x^4)}}, y_2 = -\sqrt{x^2 - y_1^2}, x \in [0.447, 0.797]$$

The leader's Pareto-optimal frontier corresponding to deterministic decisions of the follower is much worse as compared to the Pareto-optimal frontier corresponding to no decisions by the follower. From Figure 4 the leader can easily evaluate how worse she gets from the best possible frontier when she chooses a point on the Pareto-optimal frontier with deterministic lower level decisions.

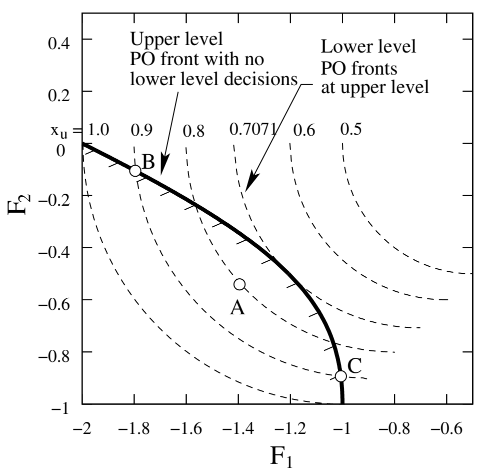

Fig. 3. Example 1: Upper level Pareto-optimal front (with no lower level decision making) and few representative lower level Pareto-optimal fronts in upper level objective space.

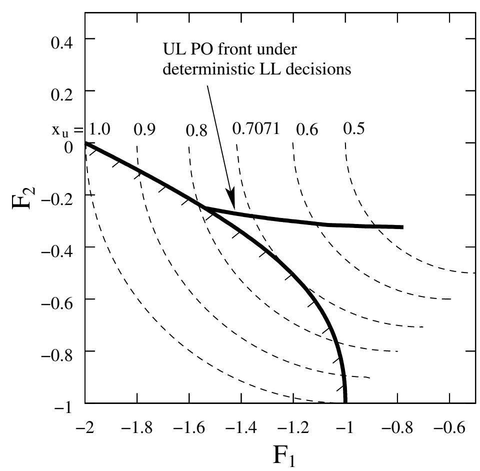

Fig. 4. Example 1: Upper level (UL) Pareto-optimal front when lower level (LL) decisions are given by $V(f_1, f_2) = 5x^2 f_1 + f_2$.

Example 2

Let us consider another simple multi-objective bilevel optimization problem that is discussed in (19; 15). The problem is scalable in terms of lower level variables, and contains a single upper level variable. For K variables at the lower level, $x_l = (y_1, \ldots, y_K)$ and $x_u = (x)$; the problem is defined as follows:

$$
\begin{aligned}
&\text{Minimize} \quad F(x_u, x_l) = \begin{pmatrix} (y_1-1)^2 + \sum_{i=2}^{K} y_i^2 + x^2 \\ (y_1-1)^2 + \sum_{i=2}^{K} y_i^2 + (x-1)^2 \end{pmatrix}, \\
&\text{subject to} \\
&(y_1, y_2, \ldots, y_K) \in \text{argmin}_{(y_1, y_2, \ldots, y_K)} \left\{ f(x_u, x_l) = \begin{pmatrix} y_1^2 + \sum_{i=2}^{K} y_i^2 \\ (y_1 - x)^2 + \sum_{i=2}^{K} y_i^2 \end{pmatrix} \right\}, \\
&-1 \le (x, y_1, y_2, \ldots, y_K) \le 2.
\end{aligned} \tag{3}
$$

For any x, the Pareto-optimal solutions of the lower level optimization problem are given as follows: $\{x_l \in \mathbf{R}^K \big| y_1 \in [0, x], y_i = 0, \text{ for } i = 2, \ldots, K\}$. In this paper, we choose $K = 14$, such that the problem contains 15 variables. The best possible frontier at the upper level may be obtained when there is no decision maker at the lower level, and all the lower level Pareto-optimal members are available at the upper level. In this example, such a frontier corresponds to the following conditions: $\{(x_u, x_l) \in \mathbf{R}^{K+1} \big| y_1 = x, y_i = 0, \text{ for } i = 2, \ldots, K, x \in [0.5, 1.0]\}$.

Now let us consider that there exists a decision maker at the lower level, whose value function is $V(f_1, f_2) = 2f_1 + f_2$. The upper level frontier for this case is shown in Figure 6. The theoretical upper level Pareto-optimal frontier for the deterministic lower level value function is given as:

$$
V(f_1, f_2) = 2f_1 + f_2, \text{then } y_1 = \frac{x}{3}, y_i = 0 \; \forall \; i = 2, \ldots, K, x \in [0.300, 1.201]
$$

We once again observe that the realized Pareto-optimal frontier for the leader is much worse when the follower freely exercises her decisions.

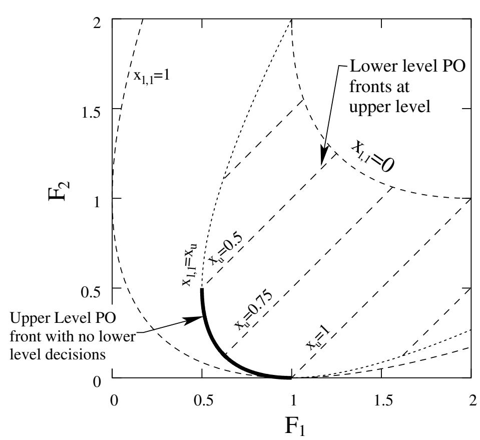

Fig. 5. Example 2: Upper level Pareto-optimal front (with no lower level decision making) and few representative lower level Pareto-optimal fronts in upper level objective space.

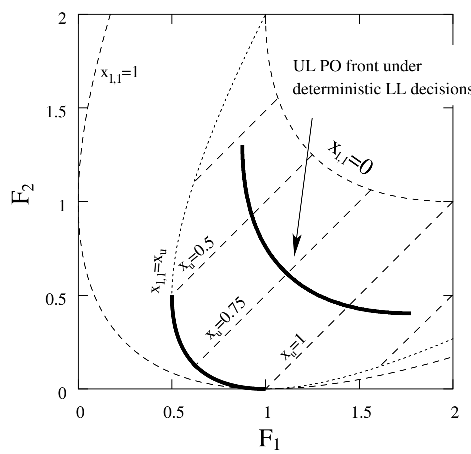

Fig. 6. Example 2: Upper level (UL) Pareto-optimal front when lower level (LL) decisions are given by $V(f_1, f_2) = 2f_1 + f_2$.

6 Algorithm Description

In this section, we introduce an evolutionary algorithm for solving bilevel problems where the upper level has multiple objectives and the lower level decisions are modeled using a value function. This means that the algorithm solves a problem with multiple objectives at upper level and single objective at the lower level. The approach is an extension of a recently proposed algorithm for single objective bilevel optimization (27; 28), and is referred as multi-objective bilevel evolutionary algorithm based on quadratic approximations (m-BLEAQ). The proposed approach is based on estimation of unknown lower level decisions using quadratic approximations, when lower level decisions corresponding to a few upper level vectors are known. The approximation helps in reducing the number of lower level optimization calls that leads to computational savings. The working of the algorithm has been shown through a flowchart in Figure 7.

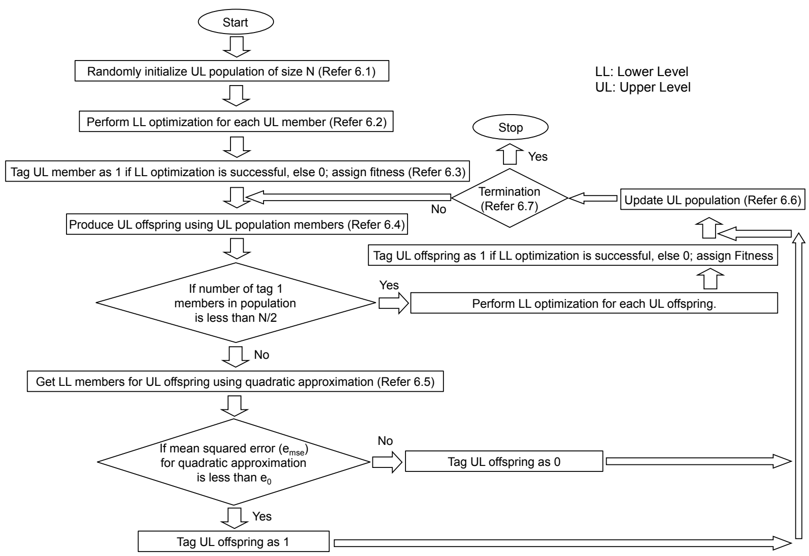

Fig. 7. Flowchart for m-BLEAQ.

6.1 Population Structure

The population structure at the upper level is shown in Figure 8. The first column represents the upper level population members and the second column represents the corresponding lower level population members that have been computed through lower level optimization or quadratic approximation. Based on the quality of the lower level members the upper level members are tagged as 0 or 1. For tag 1 upper level members the corresponding lower level members are expected to be close to lower level optimum.

6.2 Lower Level Optimization

A steady state evolutionary algorithm[1] for global optimization is used at the lower level to find the optimum. The fitness assignment at the lower level is performed based on lower level function value and constraints. The upper level vector for which lower level optimization is being performed is kept fixed during the optimization run.

---

[1] A classical optimization strategy can be used to replace the lower level evolutionary algorithm, if the lower level optimization problem adheres to the requirements of the classical approach.

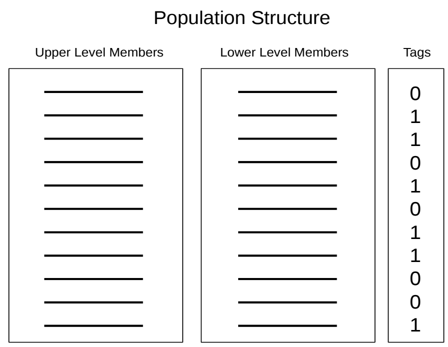

Fig. 8. Population structure at upper level of m-BLEAQ.

Step 1 Randomly initialize a lower level population of size N. Assign fitness to the members based on lower level objective functions and constraints.
Step 2 Randomly choose 6 members from the population, and perform a tournament selection. This gives 3 parents for crossover.
Step. 3 Create 2 offsprings from the parents using genetic operators on the lower level variables only.
Step. 4 Randomly choose 2 members from the population, and pool them with 2 offsprings. The 2 best members from the pool replace the chosen members from the population.
Step. 5 Perform a termination check. Proceed to next generation (Step 2), if the termination criteria is not satisfied, otherwise proceed to the next step.
Step. 6 The best obtained lower level member is paired with the corresponding upper level member in the upper level population.

6.3 Fitness Evaluation

Fitness assignment for feasible upper level member is performed based on their non-domination rank and crowding distance (29). For a given upper level member x, if the non-domination rank is given as $N_R(x)$ and crowding distance within its frontier is given as $C_D(x)$, then the fitness for the member is calculated as follows:

$$F_u(x) = \frac{1}{N_R(x) + e^{-C_D(x)}}, \tag{4}$$

Fitness for an infeasible upper level member is computed by subtracting the sum of upper level constraint violations from the fitness value of the worst feasible member.

The fitness during lower level optimization is given by lower level function values for the feasible members. For the infeasible lower level members, we subtract the sum of lower level constraint violations from the fitness value of the worst feasible member at that level.

6.4 Genetic Operations

A parent centric crossover and a polynomial mutation is performed to generate new parents. The crossover operator is similar to the PCX operator proposed in (30) with slight modifications. The operator requires 3 parents to create an offspring that are selected using tournament selection. The crossover operation is performed as show below:

$$c = x^{(p)} + \omega_\xi d + \omega_\eta \frac{p^{(2)} - p^{(1)}}{2} \tag{5}$$

The terms used in the above equation are defined as follows:

- $x^{(p)}$ is the *index* parent
- $d = x^{(p)} - g$, where g is the mean of μ parents
- $p^{(1)}$ and $p^{(2)}$ are the other two parents
- $\omega_\xi = 0.1$ and $\omega_\eta = \frac{dim(x^{(p)})}{||x^{(p)}-g||_1}$ are the two parameters.

Upper level crossovers and mutations are performed on upper level variables, while lower level crossovers and mutations are performed on lower level variables.

6.5 Quadratic Approximations

At any generation of the m-BLEAQ algorithm, we attempt to maintain at least $\frac{N}{2}$ tag 1 members. These are the upper level members for which the lower level optimal solutions are accurately known. We utilize these members to compute the lower level optimal solutions of the new upper level members. Based on the quality of the quadratic approximation, the estimated lower level optimum might be accurate or inaccurate. Figure 9 shows a scenario where there are three members for which lower level decisions are known. We utilize these members to construct a quadratic approximation that provides an estimate for the unknown lower level decision.

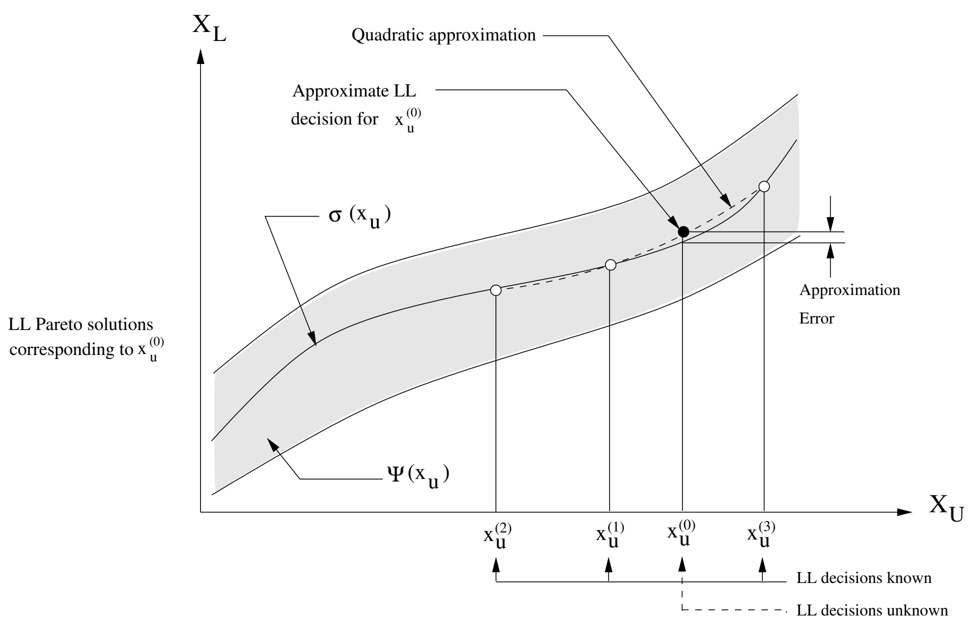

Fig. 9. Approximating decisions for an unknown upper level vector when decisions corresponding to few upper level vectors are known.

Figure 9 explains the approximation in the presence of a single lower and upper level variable. When multiple lower and upper level variables are present, we utilize all the upper level variables to construct the quadratic approximation for each lower level variable. Therefore, the number of quadratic approximations are as many as the number of lower level variables, and each lower level variable is a function of all the upper level variables. We choose the closest upper level members for quadratic approximation around the point for which we intend to estimate the lower level decision. Such an approximation is expected to provide a reliable local estimate. We propose to utilize at least $\frac{1}{2}[(dim(x_u) + 1)(dim(x_u) + 2)] + dim(x_u)$ upper level points for constructing the approximation.

6.6 Update at Upper Level

The upper level population is updated by choosing 2 random members from the population. The members are pooled with 2 offsprings generated through genetic operations, and the best members from the pool are chosen to replace the selected population members.

6.7 Termination

At the upper level we terminate the algorithm based on maximum upper level function evaluations ($T_{\max}$). We use an improvement based termination at the lower level such that if the improvement in the lower level function value is less than $1e - 5$ for 100 consecutive generations then we terminate the optimization.

6.8 Archiving

We store all the tag 1 upper level members produced by the algorithm in an archive. The final upper level Pareto-optimal solutions are presented to the user by providing the best frontier in the archive set.

6.9 Parameters

For all the computations in this paper we fix the algorithm parameters as $N = 50$. Crossover probability is fixed at 0.9 and the mutation probability is fixed at 0.1.

7 Results

We evaluate the m-BLEAQ algorithm on the examples that we discussed in an earlier section. Since we know the Pareto-optimal frontier for both problems, it is easy test the performance using Inverted Generalization Distance (IGD) (31) metric. We compare our results against the H-BLEMO approach proposed in (15).

In order to compute the IGD value, we generate 500 evenly distributed points on the upper level Pareto-optimal front of the two problems. An average distance in some sense (31) is computed between these evenly distributed points and the points on the frontier achieved by the algorithm. The smaller the IGD value the better is the performance of the approach. The IGD metric is able to provide a measure for both convergence and diversity. While presenting the results for m-BLEAQ and H-BLEMO we fix the maximum number of upper level function evaluations ($T_{\max}$) and then determine the IGD value achieved by both methods. H-BLEMO has been slightly modified at the lower level to incorporate a similar lower level termination criteria as in m-BLEAQ. The results are presented in Tables 1, 2, 3 and 4. Figures 10 and 11 show the Pareto-optimal fronts achieved by m-BLEAQ from one of the sample runs for the two test problems.

Table 1. Minimum, median and maximum IGD values obtained from 21 runs of m-BLEAQ and H-BLEMO when $T_{\max} = 5000$.

Prob.	No of Vars.	IGD (m-BLEAQ)			IGD (H-BLEMO)		
		Min	Med	Max	Min	Med	Max
Ex1	3	0.0021	0.0026	0.0033	0.0425	0.0409	0.0980
Ex2	15	0.0017	0.0027	0.0036	0.0398	0.0683	0.0532

Table 2. Minimum, median and maximum lower level function evaluations (LLFE) from 21 runs of m-BLEAQ and H-BLEMO when $T_{\max} = 5000$.

Prob.	LLFE (m-BLEAQ)			Savings: $\frac{\text{H-BLEMO (Med)}}{\text{m-BLEAQ (Med)}}$
	Min	Med	Max	LLFE
Ex1	56043	73689	81201	5.12
Ex2	33054	47679	66533	5.38

Table 3. Minimum, median and maximum IGD values obtained from 21 runs of m-BLEAQ and H-BLEMO when $T_{\max} = 10000$.

Prob.	No of Vars.	IGD (m-BLEAQ)			IGD (H-BLEMO)		
		Min	Med	Max	Min	Med	Max
Ex1	3	0.0008	0.0011	0.0013	0.0206	0.0361	0.0500
Ex2	15	0.0009	0.0012	0.0013	0.0178	0.0243	0.0305

Table 4. Minimum, median and maximum lower level function evaluations (LLFE) from 21 runs of m-BLEAQ and H-BLEMO when $T_{\max} = 10000$.

Prob.	LLFE (m-BLEAQ)			Savings: $\frac{\text{H-BLEMO (Med)}}{\text{m-BLEAQ (Med)}}$
	Min	Med	Max	LLFE
Ex1	77735	94104	99187	8.57
Ex2	45377	65674	91898	7.63

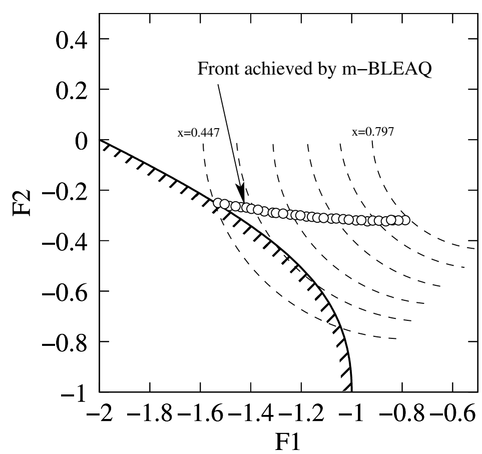

Fig. 10. Example 1: Pareto-optimal front obtained using m-BLEAQ from one of the runs when $T_{\max} = 5000$.

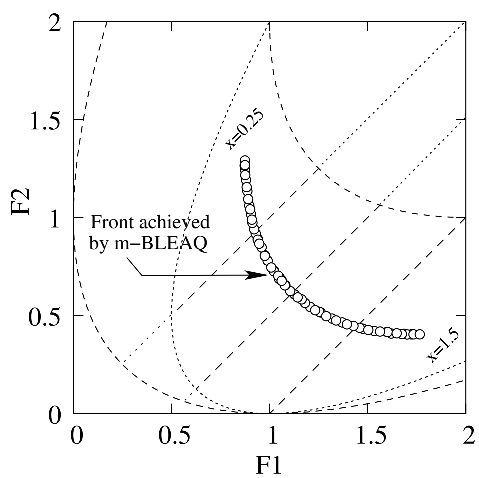

Fig. 11. Example 2: Pareto-optimal front obtained using m-BLEAQ from one of the runs when $T_{\max} = 5000$.

It is noteworthy that the H-BLEMO is capable of handling multiple objectives at both levels, but in the current formulation the lower level is represented by a value function, which means that H-BLEMO is handling a single objective problem at the lower level and a multi-objective problem at the upper level. On the other hand m-BLEAQ cannot directly handle multiple objectives at both levels. However, with multiple objectives at upper level and single objective at lower level m-BLEAQ is able to achieve much lower IGD values as compared to H-BLEMO for the same number of upper level function evaluations and much fewer lower level function evaluations.

8 Conclusions and Future Work

In this paper, we have analyzed bilevel optimization problems with multiple objectives at upper and lower level. However, in order to account for deterministic decisions of the follower, multiple objectives at the lower level have been replaced by a value function. We have considered a realistic scenario in this paper where the follower has some decision power based on which she chooses a solution from her Pareto-optimal frontier. Through examples we have shown that decisions of the follower may have significant impact on the leader's frontier when compared against the case of not accounting the follower's decisions. Recent studies on bilevel multi-objective problems have ignored such decision interactions in multi-objective bilevel optimization.

We have extended a recently proposed algorithm for single objective bilevel optimization (BLEAQ) to handle multi-objective bilevel problems with deterministic lower level decision. The extended algorithm (m-BLEAQ) is found to be computationally efficient when compared against an earlier proposed strategy (H-BLEMO). As a future research, we intend to study how the frontier at the upper level changes when the decision structure (value function) of the follower varies. It is not always possible to deterministically ascertain the value function of a decision maker. Therefore, future efforts will be directed towards handling problems with lower level decision uncertainty. It will also be interesting to consider cooperation between leader and follower, where the follower agrees to return a part of her frontier as possible lower level decisions to the leader. Such kind of cooperation by the follower may lead to an improved Pareto-optimal frontier at the upper level. Such a study will also allow to evaluate the extent of compromises and gains that can be made by the leader and the follower through mutual cooperation.

Bibliography

[1] A. Migdalas, Bilevel programming in traffic planning: Models, methods and challenge, Journal of Global Optimization 7 (4) (1995) 381–405. doi:10.1007/BF01099649.
URL http://dx.doi.org/10.1007/BF01099649

[2] L. Brotcorne, M. Labbe, P. Marcotte, G. Savard, A bilevel model for toll optimization on a multicommodity transportation network, Transportation Science 35 (4) (2001) 345–358.

[3] D. Fudenberg, J. Tirole, Game theory, MIT Press, 1993.

[4] F. J. Wang, J. Periaux, Multi-point optimization using gas and Nash/Stackelberg games for high lift multi-airfoil design in aerodynamics, in: Proceedings of the 2001 Congress on Evolutionary Computation (CEC-2001), 2001, pp. 552–559.

[5] A. Sinha, P. Malo, A. Frantsev, K. Deb, Finding optimal strategies in a multi-period multi-leader-follower stackelberg game using an evolutionary algorithm, Computers & Operations Research 41 (2014) 374–385.

[6] A. Sinha, P. Malo, A. Frantsev, K. Deb, 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-2013), IEEE Press, 2013.

[7] H. Sun, Z. Gao, J. Wu, A bi-level programming model and solution algorithm for the location of logistics distribution centers, Applied Mathematical Modelling 32 (4) (2008) 610 – 616. doi:http://dx.doi.org/10.1016/j.apm.2007.02.007.
URL http://www.sciencedirect.com/science/article/pii/S0307904X07000340

[8] J. F. Bard, Coordination of multi-divisional firm through two levels of management, Omega 11 (5) (1983) 457–465.

[9] C. Kirjner-Neto, E. Polak, A. Kiureghian, An outer approximations approach to reliability-based optimal design of structures, Journal of Optimization Theory and Applications 98 (1) (1998) 1–16. doi:10.1023/A:1022647728419.

[10] W. Smith, R. Missen, Chemical Reaction Equilibrium Analysis: Theory and Algorithms, John Wiley & Sons, New York, 1982.

[11] G. Eichfelder, Solving nonlinear multiobjective bilevel optimization problems with coupled upper level constraints, Tech. Rep. Preprint No. 320, Preprint-Series of the Institute of Applied Mathematics, Univ. Erlangen-NÃijrnberg, Germany (2007).

[12] G. Eichfelder, Multiobjective bilevel optimization, Math. Program. 123 (2) (2010) 419–449. doi:10.1007/s10107-008-0259-0.
URL http://dx.doi.org/10.1007/s10107-008-0259-0

[13] W. Halter, S. Mostaghim, Bilevel optimization of multi-component chemical systems using particle swarm optimization, in: Proceedings of World Congress on Computational Intelligence (WCCI-2006), 2006, pp. 1240–1247.

[14] X. Shi., H. S. Xia, Model and interactive algorithm of bi-level multi-objective decision-making with multiple interconnected decision makers, Journal of Multi-Criteria Decision Analysis 10 (1) (2001) 27–34.

[15] K. Deb, A. Sinha, An efficient and accurate solution methodology for bilevel multi-objective programming problems using a hybrid evolutionary-local-search algorithm, Evolutionary Computation Journal 18 (3) (2010) 403–449.

[16] Y. Yin, Genetic algorithm based approach for bilevel programming models, Journal of Transportation Engineering 126 (2) (2000) 115–120.

[17] K. Deb, A. Sinha, Constructing test problems for bilevel evolutionary multi-objective optimization, in: 2009 IEEE Congress on Evolutionary Computation (CEC-2009), IEEE Press, 2009, pp. 1153–1160.

[18] A. Sinha, K. Deb, Towards understanding evolutionary bilevel multi-objective optimization algorithm, in: IFAC Workshop on Control Applications of Optimization (IFAC-2009), Vol. 7, Elsevier, 2009.

[19] K. Deb, A. Sinha, Solving bilevel multi-objective optimization problems using evolutionary algorithms, in: Evolutionary Multi-Criterion Optimization (EMO-2009), Berlin, Germany: Springer-Verlag, 2009, pp. 110–124.

[20] K. Deb, A. Sinha, An evolutionary approach for bilevel multi-objective problems, in: Cutting-Edge Research Topics on Multiple Criteria Decision Making, Communications in Computer and Information Science, Vol. 35, Berlin, Germany: Springer, 2009, pp. 17–24.

[21] S. Pramanik, P. P. Dey, Bi-level multi-objective programming problem with fuzzy parameters, International Journal of Computer Applications 30 (10) (2011) 13–20, published by Foundation of Computer Science, New York, USA.

[22] M. Linnala, E. Madetoja, H. Ruotsalainen, J. HÃď mÃď lÃď inen, Bi-level optimization for a dynamic multiobjective problem, Engineering Optimization 44 (2) (2012) 195–207.

[23] C. O. Pieume, L. P. Fotso, P. Siarry, Solving bilevel programming problems with multicriteria optimization techniques, OPSEARCH 46 (2) (2009) 169–183.

[24] S. Ruuska, K. Miettinen, Constructing evolutionary algorithms for bilevel multiobjective optimization, in: Evolutionary Computation (CEC), 2012 IEEE Congress on, 2012, pp. 1–7.

[25] T. Zhang, T. Hu, Y. Zheng, X. Guo, An improved particle swarm optimization for solving bilevel multiobjective programming problem, Journal of Applied Mathematics.

[26] A. Sinha, Bilevel multi-objective optimization problem solving using progressively interactive evolutionary algorithm, in: Proceedings of the Sixth International Conference on Evolutionary Multi-Criterion Optimization (EMO-2011), Berlin, Germany: Springer-Verlag, 2011, pp. 269–284.

[27] A. Sinha, P. Malo, K. Deb, Efficient evolutionary algorithm for single-objective bilevel optimization, CoRR abs/1303.3901.

[28] A. Sinha, P. Malo, K. Deb, An improved bilevel evolutionary algorithm based on quadratic approximations, in: 2014 IEEE Congress on Evolutionary Computation (CEC-2014), IEEE Press, 2014, pp. 1870–1877.

[29] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197.

[30] A. Sinha, A. Srinivasan, K. Deb, A population-based, parent centric procedure for constrained real-parameter optimization, in: 2006 IEEE Congress on Evolutionary Computation (CEC-2006), IEEE Press, 2006, pp. 239–245.

[31] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, V. G. da Fonseca, Performance Assessment of Multiobjective Optimizers: An Analysis and Review, IEEE Transactions on Evolutionary Computation 7 (2) (2003) 117–132.