Evolutionary Bilevel Optimization: Applications and Methods

Ankur Sinha
Production and Quantitative Methods
Indian Institute of Management
Ahmedabad, India
asinha@iimahd.ernet.in

Kalyanmoy Deb
Electrical and Computer Engineering
Michigan State University
East Lansing, MI 48864, USA
kdeb@egr.msu.edu

Outline
- Bilevel Optimization: An Introduction
- Genesis
- Efficient Solution Methodologies
- Test Problem Construction
- Results
- Multi-objective Bilevel Optimization
- Applications

What is Bilevel Optimization?
- Two levels of optimization tasks
  - Upper level: \((x_u, x_l)\)
  - Lower level: \(x_l\), \(x_u\) is fixed
- An upper level feasible solution must be an optimal lower level solution: \((x_u, x_l^*(x_u))\)

Min is default, can be max in any of the levels

Min\((x_u, x_l)\)\(F(x_u, x_l)\),
st \(x_l \in \arg\min_{x_l}(x_u)\)
\(f(x_u, x_l)\) \(\geq \{g(x_u, x_l) \geq 0, h(x_u, x_l) = 0\}
\(G(x_u, x_l) \geq 0, H(x_u, x_l) = 0\)
\(x_u_{\min} \leq x_u \leq x_u_{\max}\), \(x_l_{\min} \leq x_l \leq x_l_{\max}\)

An Illustration
- Lower level solution \(x_l\) can be a singleton or multi-valued
- Upper level solution corresponds to the best combination of lower level optimum and upper level values
- Uncertainty about which point the lower level decision maker chooses makes the bilevel formulation ill-defined

Single LL solution
Multiple LL solutions
Multi-level Optimization

- Multi-level (L levels) optimization
- Two or more levels of optimization
- Nested structure

\[
\min F(x_1, x_2, \ldots, x_L)
\]

s.t

 Origin of Bilevel Programming

An Extension of Mathematical Programming

- All optimization problems are special cases of bilevel programming
  - Bracken and McGill (1973)

Stackelberg Games

- Bilevel programs commonly appear in game theory when there is a leader and follower
  - Stackelberg (1952)

Stackelberg Games

Two Player Game

- Leader makes the first move
- Follower reacts rationally and then makes its move
- Leader has the first mover’s advantage
- Problem is non-symmetric

Example

- Hierarchical decision making
- Consider a leader dictating selling price and supply
- Customers respond rationally to the leader’s decision
- Leader needs to anticipate this response to maximize profit

History

- Time
- Number of Articles

No of Articles vs Time

- 2010 to 2015
More than 10% share

Topic Models

Fig. 1: Topic: Methods.

Fig. 3: Topic: Classical game theory.

Fig. 2: Topic: Optimality conditions.

Fig. 4: Topic: Network design.

Around 5-10% share

Topic Models

Fig. 5: Topic: Supply chain applications.

Fig. 7: Topic: Electricity transmission applications.

Fig. 6: Topic: Optimal design applications.

Fig. 8: Topic: Telecommunication applications.

Around 5% share

Topic Models

Fig. 9: Topic: Business applications.

Fig. 11: Topic: Hierarchical decision making applications.

Fig. 10: Topic: Computer architecture and circuit design.

Fig. 12: Topic: Environment applications.

Less than 5% share

Topic Models

Fig. 13: Topic: Facility location applications.

Fig. 15: Topic: Machine learning applications.

Fig. 14: Topic: Vehicle Routing applications.

Fig. 16: Topic: Defense applications.
A Mathematical Bilevel Optimization Problem

- \( x_0 = x, \ x = y \)
- For a \( x \), maximize \( y \)
- Bold line is solution set for LL
- \( \text{Min } 3y + x \) for bold line is solution (6,2)

Single and Bi-objective Treatments

- Does not help solve Bilevel problem by
  - Ignoring of lower level or
  - Treating as a bi-objective optimization
- Need to look at differently

Multi-level Optimization: A Generic Optimization Problem

- Multi-level (L levels) optimization
  - Two (L=2) or more levels of optimization
  - Ideally, nested optimization
- Usual single, multi- and many-objective optimization problems
  - Special cases (L=1) of L-level optimization
  - Number of objectives can be more than one at each level
- Bilevel: A more generic optimization concept than single-level optimization
Similarities with Constrained Single-Objective Optimization with Equality Constraints

- A single-objective optimization problem:
  \[
  \begin{align*}
  \text{Minimize} & \quad f(x) \\
  \text{Subject to} & \quad h_k(x) = 0, \quad \forall k \\
  & \quad g_j(x) \geq 0, \quad \forall j
  \end{align*}
  \]

- Equality constraint: 
  \[ x_l = \Psi(x \setminus x_l) \]

- Usually, a root-finding problem
- A solution \( x \) is feasible, only if it satisfies all constraints

In EBO, LL problem is an optimization problem
- A solution \( (x_u, x_l) \) is not feasible, unless \( x_l \) is a solution to the LL optimization problem

Some Applications

Bilevel Problems in Practice

- Often appears from functional feasibility
  - Stability, equilibrium, solution to a set of PDEs
  - Ideally, lower level task must implement above
  - Dual problem solving in theoretical optimization
- Lower level is bypassed by approximation or by using direct simplified solution principles
  - Due to lack of suitable BO techniques
- Stackelberg games: Leader-follower
  - Leader must be restricted to follower’s decisions
  - Follower must respect leader’s decisions

Structural Optimization

- **Upper level**: Topology
- **Lower level**: Sizes and coordinates

Does it make sense to know cross-section sizes before settling down on topology?
Toll Setting Problem

- Authority’s (Upper level) problem:
  - Authority responsible for highway system wants to maximize its revenues earned from toll.
  - The authority has to solve the highway users optimization problem for all possible tolls.

- Highway users’ (Lower level) problem:
  - For any toll chosen by the authority, highway users try to minimize their own travel costs.
  - A high toll will deter users to take the highway, lowering the revenues.

Stackelberg Competition

- Competition between a leader and a follower firm (Duopoly).
- Leader solves the following optimization problem to maximize its profit:

\[
\begin{align*}
\max_{\phi} \quad & \Pi_{L} = P(\phi, q_{L}) q_{L} - C(q_{L}) \\
\text{s.t.} \quad & q_{L} \in \arg\max \{\Pi_{U} = P(\phi, q_{U}) q_{U} - C(q_{U})\}, \\
& \phi + q_{L} \geq Q, \\
& q_{L}, q_{U}, Q \geq 0.
\end{align*}
\]

Where \( Q \) is the quantity demanded, \( P(\phi, q_{L}) \) is the price of the goods sold, and \( C(\cdot) \) is the cost of production of the respective firm. The variables in this model are the production levels of each firm \( q_{L}, q_{U} \) and demand \( Q \).

Seller-Buyer Strategies

- An owner (UL) of a company dictates the selling price and supply. She/he wants to maximize profit.
- Buyers (LL) look at the product quality, pricing and various other options available to maximize their utility.
- Mixed integer programs on similar lines have been formulated by Heliporn et al. (2010).

Taxation Strategy

- Recently, there had been a controversy in Finland for gold mining in the Kuusamo region in Finland.
- The region is a famous tourist resort endowed with immense natural beauty.
- For any taxation strategy by the government (UL), the mining company (LL) optimizes its own profits.

Does it make sense for the miners to venture into it before knowing the governmental tax policies?
Defense Applications

Hub-and-Spoke Networks

Interdiction Problem
Attacker-Defender
Two levels
Attacker
Maximize operating costs post attack
Defender
Minimize operating cost

Protection Problem
Defender-Attacker
Two levels
Defender
Minimize the maximum damage post fortification
Attacker
Maximize damage

Robust Design
Defender-Attacker-Defender

Three levels
Defender
Take interdiction problem into account during design phase
Attacker
Maximize operating costs post attack
Defender
Minimize operating cost

Agri-business Management

Upper Level
Regulator
Objective 1: Minimize Pollution (Fertilizer)
Objective 2: Maximize Revenues

Lower Level
Multiple Farms
Objective: Maximize Individual Profit

Parameter Tuning

Upper Level: Find optimal parameters that maximize algorithm performance over a number of initial conditions

Lower Level: Run the optimization algorithm to find optimized solution

Researchers commonly rely on grid search or random search

Inverse Optimal Control

- While performing actions humans optimize certain unknown cost function
- It might be interesting to have an idea of the cost function that might help in designing efficient humanoids
- Given the data corresponding to the motion identifying the reward or cost function becomes an inverse problem

Parameter Tuning

Upper Level Optimization (Algorithm parameters, p)

Different Initial Conditions

Lower Level Optimization (Problem variables, s)

Optimization Problem (G)

Inverse Optimal Control

- While performing actions humans optimize certain unknown cost function
- It might be interesting to have an idea of the cost function that might help in designing efficient humanoids
- Given the data corresponding to the motion identifying the reward or cost function becomes an inverse problem
Min-Max Problems

- Typical min-max problem
  \[ \min_x \max_y f(x, y), \]
  \[ \text{Subject to } (x, y) \in (X, Y). \]
- Can be solved as a Bilevel problem
  \[ \min_{x, y} f(x, y), \]
  \[ \text{Subject to } y = \arg\max_x \{ f(x, y), (x, y) \in (X, Y) \}. \]
- Co-evolutionary problems are ideal candidates for Bilevel optimization

Solution Methodologies

- Single-level reduction using KKT
- Descent methods
  - Savard and Gauvin (1994), Vicente et al. (1994)
- Penalty function methods
- Trust region methods
  - Colson et al. (2005)
- Using lower level optimal value function
  - Mitsos (2010)

Special Cases

- Linear bilevel problems
  - Reducible to a mixed integer linear program
- Bilevel problems with combinatorial variables at upper level and linear program at lower level
  - Reducible to a mixed integer linear program
- Bilevel problems with combinatorial variables at both levels
  - Very hard to solve
- Bilevel problems with similar objectives at both levels
  - Reduces to minmax or minmin (min) problems
  - Ideas of duality can be utilized

Properties of Bilevel Problems

- Bilevel problems are typically non-convex, disconnected and strongly NP-hard
- Solving an optimization problem produces one or more feasible solutions
- Multiple global solutions at lower level can induce additional challenges
- Two levels can be cooperating or conflicting
Why Use Evolutionary Algorithms?

First, no implementable mathematical optimality conditions exist (Dempe, Dutta, Mordukhovich, 2007)

- LL problem is replaced with KKT conditions and constraint qualification (CQ) conditions of LL
- UL problem requires KKT of LL-KKT conditions, but handling LL-CQ conditions in UL-KKT becomes difficult
- Involves second-order differentials

Moreover, classical numerical optimization methods require various simplifying assumptions like continuity, differentiability and convexity

- Most real-world applications do not follow these assumptions

EA’s flexible operators, direct use of objectives, and population approach should help solve BO problems better

Niche of Evolutionary Methods (cont.)

- Usually, LL solutions are multi-modal
- Usually, BO problems are multi-objective BO
- Importantly, classical or theoretical optimization literature do not provide us with good methods and applicable results
- Other complexities (robustness, parallel implementation, fixed budget) can be handled efficiently

EAs for Bilevel Optimization

- Most of the EAs for bilevel optimization have been nested in nature
  - Using one algorithm for upper level and solving the lower level optimization problem for every upper level point
  - Not very interesting!
  - Expensive even for small instances!
  - Non-scalable!

Bilevel Optimization using EAs

- EA at upper level and exact method at lower level
  - Mathieu et al. (1994): LP for lower level and GA for upper level
  - Yin (2000): Frank-Wolfe Algorithm for lower level and EA for upper level

- EA at both upper and lower level
  - Li et al. (2006): Particle Swarm Optimization at both levels
  - Angelo et al. (2013): Differential Evolution at both levels
  - Sinha et al. (2014): Genetic Algorithm at both levels

- EA used after single-level reduction
  - EA researchers have also tried replacing the lower level problems using KKT (Hejazi et al. (2002), Wang et al. (2008), Li et al. (2007))
**Bilevel Optimization using EAs**

Approximating lower level level rational response

Approximating lower level optimal value function
- Sinha, Malo, Deb. (2016): Iteratively approximates lower level optimal function value with upper level decision vector (Discussed later)

Trust region method
- Sinha, Soun and Deb (2017)
  (To be presented on June 8, 14:30-16:30, Room 8)

---

**Can EAs be really useful for bilevel optimization?**

• Nested approaches are certainly not the way forward

---

**Can EAs be really useful for bilevel optimization?**

• It is noteworthy that at each iteration an EA has a population of points
  • Can these population of points be put to use to approximate certain mappings in bilevel?
  • Exploiting the structure and properties of the problem is essential!
**Approach 1**  
(Lower Level Reaction Set Mapping)

\[
\Psi(x_u) = \arg\min_{x_l} \{ f(x_u, x_l) : g_j(x_u, x_l) \leq 0, j = 1, \ldots, J \}
\]

\[
\min_{x_u, x_l} F(x_u, x_l)
\]

s.t.

\[
x_l \in \Psi(x_u)
\]

\[
G_k(x_u, x_l) \leq 0, k = 1, \ldots, K
\]

Step 0: Solve the lower level problem completely for the initial population  
Step 1: Use the population members to approximate the \(\Psi\)-mapping locally  
Step 2: Solve the reduced single level problem for a few iterations  
Step 3: Update the local \(\Psi\)-mappings and continue  
Step 4: If termination criteria not met, go to Step 2
Approximate $\Psi$-mapping

Using approximate $\Psi$-mapping

Approximation Choice

- Tried different strategies for localized approximation, like,
  - Linear Approximation
  - Piecewise linear approximation
  - Quadratic approximation
- Results were favorable and similar with piecewise-linear as well as quadratic approximation
- Decided to use quadratic approximation because of its simplicity
- More complex techniques like neural networks are an obvious extension but require large number of points

Set-valued $\Psi$ becomes problematic

Dual challenge: 1. Finding the set and 2. Approximating the set
Efficient Evolutionary Bilevel Optimization Algorithm (BLEAQ)

- Nested algorithm is expensive
- Train a meta-model for $x_u^*(x_u)$
- Quadratic approximation of the inducible region
  - BLEAQ constructs $\Psi$ (Sinha, Malo and Deb, 2013)
- Use meta-model until possible, else solve LL optimization problem

Meta-model of $\Psi(x_u)$

BLEAQ is being modified for handling multiple LL solutions

Approach 2
(Optimal Value Function Mapping)

$$\varphi(x_u) = \min_{x_u, x_1} \{ f(x_u, x_1) : x_1 \in \Omega(x_u) \}$$

$$\min_{x_u, x_1} F(x_u, x_1)$$

s.t.

$$f(x_u, x_1) \leq \varphi(x_u)$$

$$g_j(x_u, x_1) \leq 0, j = 1, \ldots, J$$

$$G_k(x_u, x_1) \leq 0, k = 1, \ldots, K$$

Step 0: Solve the lower level problem completely for the initial population
Step 1: Use the population members to approximate the $\varphi$-mapping locally
Step 2: Solve the reduced single level problem for a few iterations
Step 3: Update the local $\varphi$-mappings and continue
Step 4: If termination criteria not met, go to Step 2
**Issues**

\[ \varphi(x_u) = \min_{x_1} \{ f(x_u, x_1) : x_1 \in \Omega(x_u) \} \]

\[
\begin{align*}
\min_{x_u, x_1} & \quad F(x_u, x_1) \\
\text{s.t.} & \quad f(x_u, x_1) \leq \varphi(x_u) \\
& \quad g_j(x_u, x_1) \leq 0, j = 1, \ldots, J \\
& \quad G_k(x_u, x_1) \leq 0, k = 1, \ldots, K
\end{align*}
\]

- The approximate \( \varphi \)-mapping makes the region highly constrained
- With errors in estimation of \( \varphi \)-mapping the reduced problem might become infeasible

**Comparison of Two Methods**

**Approach 1**

\[
\begin{align*}
\min_{x_u, x_1} & \quad F(x_u, x_1) \\
\text{s.t.} & \quad x_1 \in \Psi(x_u) \\
& \quad G_k(x_u, x_1) \leq 0, k = 1, \ldots, K
\end{align*}
\]

\[
\begin{align*}
\min & \quad F(x_u, \Psi(x_u)) \\
\text{s.t.} & \quad G_k(x_u, \Psi(x_u)) \leq 0, k = 1, \ldots, K
\end{align*}
\]

To be solved only with respect to \( x_u \)

**Approach 2**

\[
\begin{align*}
\min_{x_u, x_1} & \quad F(x_u, x_1) \\
\text{s.t.} & \quad f(x_u, x_1) \leq \varphi(x_u) + \varepsilon_i \\
& \quad g_j(x_u, x_1) \leq 0, j = 1, \ldots, J \\
& \quad G_k(x_u, x_1) \leq 0, k = 1, \ldots, K
\end{align*}
\]

Approaches to 0 with increase in iterations \( i \)

- The approximate \( \varphi \)-mapping makes the region highly constrained
- With errors in estimation of \( \varphi \)-mapping the reduced problem might become infeasible
- Therefore, we relax the \( \varphi \)-constraint using \( \varepsilon_i \)

\[ \varphi \] being a function is easier to approximate than \( \Psi \)
Test Problems

• Given that a convergence proof is difficult, we can only use test problems to justify that the ideas work!
• First, we begin with some simple test problems.
Test Problems

<table>
<thead>
<tr>
<th>Problem</th>
<th>Formulation</th>
<th>Best Known Sol.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP1</td>
<td>$\min_{x \in \mathbb{R}^2} F(x) = \left( \cos(5\pi x_1) - 1 \right)^2 + \left( \cos(5\pi x_2) - 1 \right)^2$</td>
<td>$x^* = (0,0)$, $f^* = 0$</td>
</tr>
<tr>
<td>TP2</td>
<td>$\min_{x \in \mathbb{R}^2} F(x) = \left( x_1^2 + x_2^2 \right)^2 + \left( x_1^2 - x_2 \right)^2$</td>
<td>$x^* = (0.0, 0.0)$, $f^* = 0$</td>
</tr>
</tbody>
</table>

Results on TPs

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varphi$-Appx</td>
<td>$\psi$-Appx</td>
</tr>
<tr>
<td>TP1</td>
<td>134</td>
</tr>
<tr>
<td>TP2</td>
<td>148</td>
</tr>
<tr>
<td>TP3</td>
<td>299</td>
</tr>
<tr>
<td>TP4</td>
<td>3773</td>
</tr>
<tr>
<td>TP5</td>
<td>2941</td>
</tr>
<tr>
<td>TP6</td>
<td>1689</td>
</tr>
<tr>
<td>TP7</td>
<td>2126</td>
</tr>
<tr>
<td>TP8</td>
<td>2699</td>
</tr>
</tbody>
</table>

Comparison with other approaches

<table>
<thead>
<tr>
<th>Mean Func. Eval. (UL+LL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varphi$-appx</td>
</tr>
<tr>
<td>TP1</td>
</tr>
<tr>
<td>TP2</td>
</tr>
<tr>
<td>TP3</td>
</tr>
<tr>
<td>TP4</td>
</tr>
<tr>
<td>TP5</td>
</tr>
<tr>
<td>TP6</td>
</tr>
<tr>
<td>TP7</td>
</tr>
<tr>
<td>TP8</td>
</tr>
</tbody>
</table>

Most of the evaluations were spent in initialization.
Let us modify the test problems!

\[ f_{\text{new}}(x, y) = \begin{cases} F(x, y) + y_p^2 + y_q^2 & \text{if } y_p, y_q \in [-1, 1] \\ f(x, y) + (y_p - y_q)^2 & \text{otherwise} \end{cases} \]

Modification leads to multiple lower level optimal solutions for each upper level decision vector.

### Results (Modified Test Problems)

<table>
<thead>
<tr>
<th>Upper Level Function Evaluations</th>
<th>Lower Level Function Evaluations</th>
<th>Both Methods Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\phi)-Apex.</td>
<td>(\psi)-Apex.</td>
<td>(\phi)-Apex.</td>
</tr>
<tr>
<td>Min</td>
<td>Med</td>
<td>Max</td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>m-TP1</td>
<td>130</td>
<td>172</td>
</tr>
<tr>
<td>m-TP2</td>
<td>116</td>
<td>217</td>
</tr>
<tr>
<td>m-TP3</td>
<td>129</td>
<td>233</td>
</tr>
<tr>
<td>m-TP4</td>
<td>198</td>
<td>564</td>
</tr>
<tr>
<td>m-TP5</td>
<td>160</td>
<td>218</td>
</tr>
<tr>
<td>m-TP6</td>
<td>167</td>
<td>174</td>
</tr>
<tr>
<td>m-TP7</td>
<td>114</td>
<td>214</td>
</tr>
<tr>
<td>m-TP8</td>
<td>150</td>
<td>466</td>
</tr>
</tbody>
</table>

### Bilevel Test Problem Construction

- Test problems with controllable difficulties are often required to evaluate evolutionary algorithms.
- Controllable and segregated difficulties help to identify what aspects the algorithm is unable to handle.
- Scenarios with conflicting or cooperative relationships help to identify the algorithm's performance under varying conditions.

### Requirements

- Controllable difficulty in convergence at upper and lower levels.
- Controllable difficulty caused by interaction of two levels.
- Multiple global solutions at the lower level for any given set of upper level variables.
- Clear identification of relationships between lower level optimal solutions and upper level variables.
- Scalability to any number of decision variables at upper and lower levels.
- Constraints (preferably scalable) at upper and lower levels.
- Possibility to have conflict or cooperation at the two levels.
- The optimal solution of the bilevel optimization is known.
Test Problem Framework

The objectives and variables on both levels are decomposed as follows:

\[ F(x_u, x_l) = F_1(x_{u1}) + F_2(x_{l1}) + F_3(x_{u2}, x_{l2}) \]

\[ f(x_u, x_l) = f_1(x_{u1}, x_{u2}) + f_2(x_{l1}) + f_3(x_{u2}, x_{l2}) \]

where \( x_u = (x_{u1}, x_{u2}) \) and \( x_l = (x_{l1}, x_{l2}) \)

(Sinha, Malo and Deb, 2014)

Roles of Variables

Panel A: Decomposition of decision variables

<table>
<thead>
<tr>
<th>Vector</th>
<th>Purpose</th>
<th>Vector</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_{u1} )</td>
<td>Compulsory in upper level</td>
<td>( x_{l1} )</td>
<td>Compulsory in lower level</td>
</tr>
<tr>
<td>( x_{u2} )</td>
<td>Interaction with upper-level</td>
<td>( x_{l2} )</td>
<td>Interaction with upper-level</td>
</tr>
</tbody>
</table>

Panel B: Decomposition of objective functions

<table>
<thead>
<tr>
<th>Component</th>
<th>Purpose</th>
<th>Component</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1(x_{u1}) )</td>
<td>Difficulty in convergence</td>
<td>( f_1(x_{l1}) )</td>
<td>Difficulty in convergence</td>
</tr>
<tr>
<td>( f_2(x_{u1}) )</td>
<td>Conflict/co-operation</td>
<td>( f_2(x_{l1}) )</td>
<td>Difficulty in interaction</td>
</tr>
<tr>
<td>( f_3(x_{u2}, x_{l2}) )</td>
<td>Difficulty in interaction</td>
<td>( f_3(x_{u2}, x_{l2}) )</td>
<td>Difficulty in interaction</td>
</tr>
</tbody>
</table>

Controlling Difficulty for Convergence

- Convergence difficulties can be induced via following routes:
- Dedicated components: \( F_1 \) (upper) and \( f_2 \) (lower)
- Example:

\[ F(x_u, x_l) = F_1(x_{u1}) + F_2(x_{l1}) + F_3(x_{u2}, x_{l2}) \]

Quadratic

\[ f(x_u, x_l) = f_1(x_{u1}, x_{u2}) + f_2(x_{l1}) + f_3(x_{u2}, x_{l2}) \]

Multi-modal

Controlling Difficulty in Interactions

- Interaction between variables \( x_{u2} \) and \( x_{l2} \) could be chosen as follows:
  - Dedicated components: \( F_3 \) and \( f_3 \)
  - Example:

\[ F(x_u, x_l) = F_1(x_{u1}) + F_2(x_{l1}) + F_3(x_{u2}, x_{l2}) \]

\[ \sum_{i=1}^{r} z_{i1}^2 + \sum_{i=1}^{r} (z_{i2}^2 - \tan z_{i1}^2)^2 \]

\[ f(x_u, x_l) = f_1(x_{u1}, x_{u2}) + f_2(x_{l1}) + f_3(x_{u2}, x_{l2}) \]

\[ \sum_{i=1}^{r} (z_{i2}^2 - \tan z_{i1}^2)^2 \]
### Difficulty due to Conflict/Co-operation

- Dedicated components: $F_2$ and $f_2$ or $F_3$ and $f_3$ may be used to induce conflict or cooperation
- Examples:
  - Cooperative interaction: Improvement in lower-level improves upper-level (e.g. $F_2 = f_2$)
  - Conflicting: interaction = improvement in lower-level worsens upper-level (e.g. $F_2 = -f_3$)
  - Mixed interaction is also possible

### Difficulty due to Constraints

Constraints are included at both levels with one or more of the following properties:
- Constraints exist, but are not active at the optimum
- A subset of constraints, or all the constraints are active at the optimum
- Upper level constraints are functions of only upper level variables, and lower level constraints are functions of only lower level variables
- Upper level constraints are functions of upper as well as lower level variables, and lower level constraints are also functions of upper as well as lower level variables
- Lower level constraints lead to multiple global solutions at the lower level
- Constraints are scalable at both levels

### Controlled Multimodality

- Obtain multiple lower-level optima for every upper level solution:
  - Component used: $f_2$
  - Example: Multimodality at lower-level
  
  \[
  f_1(x_{u1}, x_{u2}) = (x_{u1}^1)^2 + (x_{u1}^2)^2 + (x_{u2}^2)^2 + (x_{u2}^2)^2
  \]

  \[
  f_2(x_{u1}) = (x_{u1}^1 - x_{u1}^2)^2
  \]

  Where, $x_{u1}^1 = x_{u1}^2$

  \[
  f_3(x_{u2}, x_{u2}) = (x_{u2}^1 - x_{u2}^2)^2 + (x_{u2}^1 - x_{u2}^2)^2
  \]

  \[
  x_{u1}^1 = x_{u1}^2
  \]

  \[
  x_{u1}^1 = x_{u1}^2 = 0
  \]

### Problem 1

#### Interaction: Cooperative

- Lower level: Convex (w.r.t. lower level variables)
- Upper level: Convex (induced space)

#### Upper and Lower Function Contours
The values of the variables at the optima are efficiently handled lower level multi-modality without geometrically more difficult than the previous test problem. The Rosenbrock's (banana) function such that the global optimum, which is at (x₁, x₂) = (0, 0), is significantly far from the local optima.

\[ f(x) = \sum_{i=1}^{p} (x_i - 1)^2 - \prod_{i=1}^{p} \cos(2\pi x_i) \]

Upper and Lower Function Contours

### Problem 2

Interaction: Conflicting

Lower level: Convex (w.r.t. lower-level variables)

Upper level: Convex (induced space)

### Problem 3

Interaction: Cooperative

Lower level: Multimodality using Rastrigin's function

Upper level: Convex (induced space)

### Problem 4

Interaction: Conflicting

Lower level: Multimodality using Rastrigin's function

Upper level: Convex (induced space)

### Results Using BLEAQ

- Following are the results for 10 variable instances of the test problems (Sinha et al., 2014) using BLEAQ
- Comparison performed against nested evolutionary approach

**Number of Runs:** 21

**Savings: Ratio of FE required by nested approach against BLEAQ**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LL</td>
<td>UL</td>
<td>LL</td>
</tr>
<tr>
<td>SMD1</td>
<td>99315</td>
<td>610</td>
<td>110716 (14.71)</td>
</tr>
<tr>
<td>SMD2</td>
<td>90732</td>
<td>376</td>
<td>91023 (16.49)</td>
</tr>
<tr>
<td>SMD3</td>
<td>110701</td>
<td>620</td>
<td>125546 (11.25)</td>
</tr>
<tr>
<td>SMD4</td>
<td>61326</td>
<td>410</td>
<td>81434 (13.59)</td>
</tr>
<tr>
<td>SMD5</td>
<td>102868</td>
<td>330</td>
<td>129371 (15.41)</td>
</tr>
<tr>
<td>SMD6</td>
<td>95887</td>
<td>734</td>
<td>118456 (14.12)</td>
</tr>
</tbody>
</table>

For other problems as well, the improvement is more than an order of magnitude.
Overall Results on Eight Test Problems

- Median results for eight bilevel test problems
- Comparison against the evolutionary algorithm of Wang et al. (2005)
  - BLEAQ is an order of magnitude better

<table>
<thead>
<tr>
<th></th>
<th>BLEAQ</th>
<th>WJL</th>
<th>Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP1</td>
<td>15.432</td>
<td>85,499</td>
<td>5.54</td>
</tr>
<tr>
<td>TP2</td>
<td>15.632</td>
<td>256,227</td>
<td>16.39</td>
</tr>
<tr>
<td>TP3</td>
<td>4844</td>
<td>92,526</td>
<td>19.10</td>
</tr>
<tr>
<td>TP4</td>
<td>16,422</td>
<td>291,817</td>
<td>17.77</td>
</tr>
<tr>
<td>TP5</td>
<td>15,524</td>
<td>77,302</td>
<td>4.98</td>
</tr>
<tr>
<td>TP6</td>
<td>17,421</td>
<td>163,701</td>
<td>9.40</td>
</tr>
<tr>
<td>TP7</td>
<td>257,243</td>
<td>1,074,742</td>
<td>4.18</td>
</tr>
<tr>
<td>TP8</td>
<td>12,553</td>
<td>213,522</td>
<td>17.04</td>
</tr>
</tbody>
</table>
Advanced EBO Ideas (cont.)

- Highly constrained EBO
- Mixed-integer EBO
- EBO with a fixed budget at LL and UL
- EBO versus EO for $F=f$
- Error propagation from lower level to upper level
  - Theoretical convergence studies
- Evolutionary Multi-Level Optimization (EMLO)

Multi-objective Extension

- Bilevel problems may involve optimization of multiple objectives at one or both levels
- Dempe et al. (2006) developed KKT conditions
- Little work has been done in the direction of multi-objective bilevel algorithms (Eichfelder (2007), Deb and Sinha (2010))
- A general multi-objective bilevel problem may be formulated as follows:

$$\min_{x \in \mathcal{X}} F(x_0, x_1) = (f_1(x_0, x_1), \ldots, f_k(x_0, x_1))$$
subject to
$$x_1 \in \arg\min_{x_1} \{f(x_0, x_1), \ldots, f_k(x_0, x_1)\}$$
$$g_i(x_0, x_1) \geq 0, \forall i \in I$$
$$h_j(x_0, x_1) = 0, \forall j \in J.$$
Preference Structure Known

- Two levels of decision making
- Multiple objectives involved at both the levels

Leader Objectives: Max Objective 1
Max Objective 2

Followers Objectives: Max objective 1
Max objective 2

Lower level problem becomes single objective

Uncertainty from Unknown Preference Structure

- Two levels of decision making
- Multiple objectives involved at both the levels

Leader Objectives: Max Objective 1
Max Objective 2

Followers Objectives: Max objective 1
Max objective 2

There is uncertainty around the frontier

Challenges

- Such problems can be very difficult to handle
- Optimistic formulation makes little sense in these problems
- Considering a known preference structure (and accounting for uncertainties) might be a realistic and viable approach

Test problem 1 and results

\[ f(x) = \begin{bmatrix} x_1^2 + x_2^2 \end{bmatrix} \]

Subject to:
\[ \begin{cases} \begin{array}{l} \text{Minimize} \quad f(x) = \begin{bmatrix} x_1^2 + x_2^2 \end{bmatrix} \quad \text{subject to} \quad \begin{array}{l} G_1(x) = 0, \quad x \in [0, 5] \times [0, 5], \quad x_1, x_2 \leq 5, \end{array} \\
\end{array} \end{cases} \]

Lower level Pareto front depends on x
- Upper level Pareto-optimal front lies on constraint \( G_1 \)
  - Maximum two solutions from each x
  - Not all x in upper Pareto-optimal front
- Solutions possible even below the upper level Pareto-optimal front, but they are infeasible
A Business DM Problem
CEO: Leader and Dept Head: follower
- Weighted sum solution (Zhang et al, 2007) is an extreme solution

Mine Taxation Strategy Problem from Finland
Kuusamo has natural beauty and a famous tourist resort
- Contains large amounts of gold deposits
Dragon Mining is interested in mining in the region
Pros:
- Generate a large number of jobs
- Monetary gains
Cons:
- Run-off water from mining will pollute Kitkajoki river
- Ore contains Uranium, mining may blemish reputation
- Open pit mines located next to Ruka slopes will be a turn-off for skiing and hiking enthusiasts
- Permanent damage to the nature
**BLEAQ Results**

Preferred strategy: ~75% profit to the government, ~25% to company

**Some Problems**

Scalable in terms of variables and objectives, controllable difficulties

**EMBO Test Problem**

Construction Principle

- Difficulties identified
- Bottom-up approach
- Five-step procedure
- Conflict between lower and upper levels

**EMBO with Decision-Making**

- Preference in LL Pareto front may not lead to UL Pareto solutions
- Converse is not true
- Raises interesting hierarchy among UL and LL decision makers

Raises interesting scenarios, which we are currently pursuing!
Importance of LL Decision-Maker

• LL Decision-maker can make a decision on her own
• M-BLEAQ (Sinha, Malo, Deb, ECJ 2014)

Variation in Expectation

• Optimistic PO front: No power on LL DM
• Pessimistic PO front: Complete power on LL DM
• Leader optimizes worst outcome from LL
• The difference between optimistic and pessimistic fronts provide DM ideas at UL

Bilevel Optimization with Uncertainties

- Uncertainty is, in most cases, inevitable in practical applications.
- Sources of uncertainties:
  - Imperfect implementation, changing environment, etc.
- In the context of bilevel optimization problems
  - Uncertainty in design variables and parameters.
  - Uncertainty in objective and constraint function computations (Noise)
  - Uncertainty in decision making information.
  - Uncertainty in control of decision-making preferences between two levels.
- Uncertainties in the context of bilevel optimization have NOT been formally studied.
- Clear mathematical definitions and formulations of robust/reliable bilevel solutions do NOT exist.

Robust Bilevel Optimization

- Both upper and lower-level variables are uncertain within their neighborhoods: Type-I Robustness:
  \[
  \begin{align*}
  \min_{(x,y)} & \quad f(x,y), \\
  \text{s.t. } & \quad y \in \arg\min_{(y)} \{ f(x,y) | g_j(x + \Delta x, y + \Delta y) \leq 0, \\
  & \qquad \forall \Delta x \in B_{\Delta x}, \Delta y \in B_{\Delta y}, j = 1, \ldots, J_k \}, \\
  & \qquad g_j(x + \Delta x, y + \Delta y) \leq 0, \forall \Delta x \in B_{\Delta x}, \Delta y \in B_{\Delta y}, \\
  & \qquad \quad j = 1, \ldots, J_l. 
  \end{align*}
  \]
  \[
  f^\text{eff}(x,y) = \frac{1}{|B_{\Delta x} \times B_{\Delta y}|} \int_{B_{\Delta x} \times B_{\Delta y}} f(x,y) dx dy, \\
  f^\text{opt}(x,y) = \frac{1}{|B_{\Delta x} \times B_{\Delta y}|} \int_{B_{\Delta x} \times B_{\Delta y}} f^\text{opt}(x,y) dx dy. 
  \]

Note that even if \( \Delta y = 0 \), LL is uncertain due to \( \Delta x \) perturbation, stays as parameter uncertainties at LL.
Bilevel Optimization with Uncertainties: Big Picture

Robust Bi-Level Optimization

Global & Sensitive

Local & Robust

Robustness-based (Cont.)

Type-II Robustness

<table>
<thead>
<tr>
<th>2-Variable</th>
<th>LUV</th>
<th>LLV</th>
<th>LUS</th>
<th>LIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid</td>
<td>0.545</td>
<td>1.169</td>
<td>0.60</td>
<td>1.18</td>
</tr>
<tr>
<td>SF</td>
<td>0.047</td>
<td>1.1906</td>
<td>0.5992</td>
<td>1.1819</td>
</tr>
<tr>
<td>Worst</td>
<td>0.0439</td>
<td>1.1826</td>
<td>0.5442</td>
<td>1.2116</td>
</tr>
</tbody>
</table>
Constrained Bilevel Optimization

- Constraints exist in almost every practical engineering design problem, and play a critical role in deciding the optimal solution.
- The deterministic optimum usually lies on a constraint surface or at the intersection of constraint surfaces.
- Fail to remain feasible in many occasions.
- Many studies aim to handle this issue in single level optimization, none yet in the domain of bilevel optimization.

Reliability in Bilevel Problem

- Reliable bilevel solution definition:
  \[
  \min_{x, y} F(x, y), \quad \text{subject to } y \in \arg\min_y \left\{ f(x, y) \left| P(\bigwedge_{i=1}^{I} g_i(x, y) \leq 0) \geq R \right. \right\},
  \]
  \[
  \left. P(\bigwedge_{i=1}^{I} g_i(x, y) \leq 0) \right. \geq R.
  \]
- \( P(f) \) signify the joint probability of the solution \((x, y)\) being feasible for all constraints.
- The effect of uncertainties in lower, upper or both levels are different because of the unequal importance of each level.
- Test problems proposed for the purpose of concept demonstration, NOT for algorithm performance assessment.
A Scenario

- Constraint functions at both levels:
  
  \[
  \begin{align*}
  &\text{Maximize:} & F_{ij}(x,y) = x_j, \\
  &\text{subject to} & y \in \arg\max(y) & \quad \frac{f(x,y)}{a_i} \geq 1, 2, 3, \\
  & & G_{i}(x,y) \geq 0, & i = 1, 2, 3, \\
  & & -4 \leq y_i \leq 10, -100 \leq y_j \leq 200.
  \end{align*}
  \]

- Feasible Region

Tri-Level Optimization

- Three levels of optimization problems interlinked by two consecutive levels
  - Min \( F(x,y,z) \)
  - Min \( F(y,z), \) given \( x \)
  - Min \( F(z), \) given \( x \) and \( y \)

- Constraints are expected at every level
- To make an application realistic, we need to replace lowest level heuristic/rule based

A Case Study: Supply Chain Management

Yearly

- Strategic Level Planning

\[
\text{Min. } F(x,y) = \sum_{i} f_{i}(x,y).
\]

- Operational Level

\[
\begin{align*}
 &\text{Min. } f_{ij}(x,y) = \sum_{i} f_{i}(x,y), \\
 &\text{subject to } y_j \leq C_j, j = 1, \ldots, \text{ (routes)}, \\
 &x_j = E_j, j = 1, \ldots, \text{ (routes)}, \\
 &f(x,y) = \max_{i} \left\{ D_i(x,y) \right\}, \\
 &y = a, x, y_j \in \text{ (limits)}.
\end{align*}
\]

Weekly

- Operational and Planning

Advantage of Using Bilevel Optimization Over Single-Level Optimization:

<table>
<thead>
<tr>
<th>Model</th>
<th>Single-level</th>
<th>Bilevel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transportation Cost</td>
<td>$16.2M</td>
<td>$15.5M (4.3%)</td>
</tr>
</tbody>
</table>

State-Level and Zip-Level Results

44,000 ZIP Codes

- UL: Plant to destination layout
- LL: At each plant, carrier companies are pre-determined
- Allocation of goods to carrier is LL task

State- and Zip-Level Results

44,000 ZIP Codes

- BLEAQ can be extended
- Currently pursuing
Uncertainty in Demand

- Seasonal demand of goods assumed with uncertainty
- Robust bi-level optimization performed
- Robust solution is able to handle uncertainties better than the deterministic solution

Bi-Objective at Operational Level

- Operational level at each plant considers two objectives:
  - Transport cost
  - Service quality obtained carrier companies
  \[ J(x, y) = \sum_{j=1}^{I} y_j \cdot C_j \]
- Produces a PO front at each plant
- Strategic level chooses the best overall Transport Cost

Tri-Level Consideration

Min. \( f(x, y) = \sum_i f_i(x, y) \)  
where \( y \) solve
Min. \( J(x, y) = \sum_i J_i(x, y) \)  
where \( y \) solve
Min. \( f_i(x, y) = \sum_j f_i(x, y) \)  
(Operational level)

\[
\begin{align*}
J(x, y) &= \min \sum_i D_i(x) \cdot C_i(y) \\
&= \min \sum_i D_i(x) \cdot C_i(y) \\
&= \sum_i D_i(x) \cdot C_i(y) \\
&= \sum_{j=1}^{I} y_j \cdot C_j \\
\end{align*}
\]

At one of the plants

Advantage of Using Tri-level Optimization:

<table>
<thead>
<tr>
<th></th>
<th>Single-level</th>
<th>Bi-level</th>
<th>Tri-level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transportation Cost</td>
<td>$16.2M</td>
<td>$15.5M</td>
<td>$15.5M</td>
</tr>
</tbody>
</table>

Conclusions

- Bilevel problems are plenty in practice, but are avoided due to lack of efficient methods
- Bilevel optimization received lukewarm interest by EA researchers so far
- Population approach of EA makes tremendous potential
- Nested nature of the problem makes the task computationally expensive
- Meta-modeling based EBO and its extensions show promise
- Extension to Tri-Level optimization is needed
- Application to industry would be beneficial
Some References


Other Main Collaborators

Dr. Pekka Malo
Department of Information and Service Economy
Aalto University School of Business, Finland
Email: pekka.malo@aalto.fi

Zhichao Lu, PhD Student
Electrical and Computer Engineering
Michigan State University
East Lansing, MI 48864, USA
kdeb@egr.msu.edu

Source: www.bilevel.org