Lecture 24 Bilevel Optimization

Can Li

ChE 597: Computational Optimization
Purdue University

Motivation

Bilevel optimization addresses problems structured in a hierarchical manner, where a top-level (”leader”) optimization problem incorporates another optimization problem at a lower level (”follower”) as a constraint. The leader aims to optimize their objective, taking into account the optimal responses of the follower to the leader’s decisions.

Stackelberg game (leader-follower game)

Bilevel optimization dates back to the seminal publications on leader-follower games by von Stackelberg (1934).

- The leader moves first, setting a strategy that the followers respond to.
- Followers react optimally to the leader's decision.

Mathematical Formulation

- Leader's problem: $\max_{x_l} f_l(x_l, x_f^*(x_l))$
- Follower's problem: $\max_{x_f} f_f(x_l, x_f)$
- Bilevel formulation:

$$\max_{x_l} f_l(x_l, x_f) \quad \text{s.t. } x_f \in \arg\max_{x_f} f_f(x_l, x_f)$$

- Where x_l and x_f are the decisions of the leader and follower, respectively. $x_f^*(x_l)$ is the optimal response of the follower given the leader's decision is x_l. f_l and f_f are the "pay-off" functions for the leader and the follower.

Stackelberg Game - Extended Market Entry Example

1. **Market Demand and Price**
   - Market price P inversely relates to total quantity $Q = q_l + q_f$: $P(Q) = a - bQ$, where $a, b > 0$.
2. **Profit Functions**
   - Leader's profit: $\pi_l = \Big(a - b(q_l + q_f)\Big)q_l - c_l q_l$
   - Follower's profit: $\pi_f = \Big(a - b(q_l + q_f)\Big)q_f - c_f q_f$
   - c_l and c_f are marginal costs for the leader and follower, respectively.
3. **Decision Sequence**
   - Leader first decides q_l anticipating the follower's response.
   - Follower observes q_l and decides q_f.

Deriving Optimal Quantities

1. **Follower's Best Response Function**
   - Follower's profit: $\pi_f = \Big(a - b(q_l + q_f)\Big)q_f - c_f q_f$
   - Differentiate π_f w.r.t. q_f and set to 0: $\frac{\partial \pi_f}{\partial q_f} = a - b(q_l + 2q_f) - c_f = 0$
   - Solve for $q_f^*(q_l)$: $q_f^* = \frac{a - c_f - bq_l}{2b}$
2. **Leader's Optimization Problem**
   - Substitute $q_f^*(q_l)$ to π_l. $\pi_l = \Big(a - b(q_l + q_f^*(q_l))\Big)q_l - c_l q_l$
   - Differentiate π_l w.r.t. q_l, $\frac{\partial \pi_l}{\partial q_l} = 0$ and solve for q_l^*.
3. **Substituting q_l^* to find q_f^***
   - With q_l^* known, substitute back to find q_f^* in closed form.

Toll setting

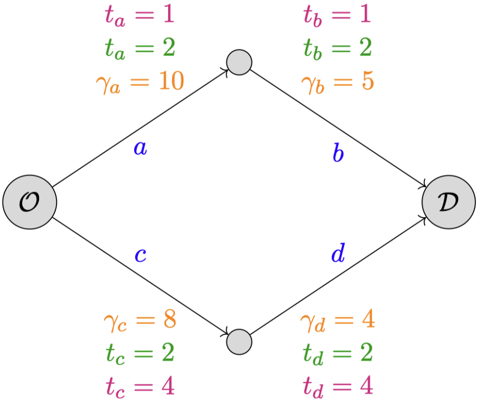

Figure: γ: base travel cost (travel time, gas cost, etc), t: toll. ref: Schmidt and Beck, 2023

Toll setting

Imagine a transportation network, via which a set of drivers want to reach their destination, starting from their origin. Usually, the objective of these travelers is to travel from their origin to their destination at minimum costs. In this situation, costs can, e.g., be travel time, toll costs, or a combination of both.

On the other hand, there usually is a toll setting agency, which decides on the tolls imposed on certain parts of the highway system. This toll setting agency wants to maximize the revenues based on the tolls and the travelers, afterward, minimize their traveling costs. The toll setting agency is the leader and the travelers are the followers in this setting. Again, the leader anticipates the optimal reaction of the followers, whereas the followers' decisions obviously depend on the decision of the leader. Whereas we only had one follower in the pricing example, we now have multiple followers. The former is called a single-leader single-follower problem, whereas the latter is called a single-leader multi-follower problem.

Sets and Parameters:

- γ_{ij}: Base travel cost (excluding tolls) between node i and node j. This can take into account travel time, fuel cost, etc.
- D: The demand or the number of travelers seeking to travel from the origin to the destination.
- E: The set of edges (road segments) on which tolls can be imposed.
- V: The set of nodes (intersections, origins, destinations) in the network.
- P: The set of all paths from the origin to the destination.

Variables:

- x_p: The flow of travelers on path p, representing the number of travelers choosing each path.
- t_{ij}: The toll set on edge i, j, which is the decision variable for the toll setting agency.

Leader's Problem (Upper Level)

Objective Function

Maximize toll revenue by optimizing the tolls t_{ij} on each edge:

$$\max \quad \sum_{(i,j)\in E} t_{ij} \sum_{p\in P:(i,j)\in p} x_p$$

Constraints

- Non-negative tolls: Ensuring that tolls cannot be negative.

$$t_{ij} \geq 0 \quad \forall (i,j) \in E$$

- Additional constraints may include limits on maximum tolls or regulations specific to certain roads or areas.
- The leader's problem focuses on setting tolls t_{ij} for each edge in the network to maximize total revenue, considering how toll changes affect traveler path choices x_p.

Follower's Problem (Lower Level)

Objective Function

Minimize travel cost for the given OD pair by selecting optimal paths, considering both travel time and tolls:

$$\min \quad \sum_{p \in P} x_p \cdot \left(\sum_{(i,j) \in p} (\gamma_{ij} + t_{ij}) \right)$$

Constraints

- Demand satisfaction: Ensuring that the total number of travelers D is evenly distributed across the selected paths.

$$\sum_{p \in P} x_p = D$$

- Non-negativity of path flows:

$$x_p \geq 0 \quad \forall p \in P$$

Bilevel Optimization Problem Formulation

$$\max \quad \sum_{(i,j)\in E} t_{ij} \cdot \sum_{p\in P:(i,j)\in p} x_p$$

$$\text{s.t.} \;\; t_{ij} \geq 0 \quad \forall (i,j) \in E$$

$$x = \arg\,\min_x \left\{ \sum_{p\in P} x_p \cdot \left(\sum_{(i,j)\in p} (\gamma_{ij} + t_{ij}) \right) \right\}$$

$$\text{s.t.} \;\; \sum_{p\in P} x_p = D$$

$$x_p \geq 0 \quad \forall p \in P$$

The upper level problem seeks to maximize toll revenues considering the response of the travelers (lower level), who minimize their total travel cost, which includes tolls.

Bilevel Optimization Problem

upper-level problem

$$\begin{aligned} \min_{x \in X, y} \quad & F(x,y) \\ \text{s.t.} \quad & G(x,y) \geq 0, \\ & y \in S(x) \end{aligned}$$

lower-level problem

$$\begin{aligned} \min_{y \in Y} \quad & f(x,y) \\ \text{s.t.} \quad & g(x,y) \geq 0. \end{aligned}$$

- Variables $x \in \mathbb{R}^{n_x}$ are upper-level variables (leader's decisions) and $y \in \mathbb{R}^{n_y}$ are lower-level variables (follower's decisions).
- The objective functions $F, f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}$, and constraint functions $G : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}^m$ and $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}^\ell$.
- The sets X and Y may denote integrality constraints, e.g., $X = \mathbb{Z}^{n_x}$ for integer programs.
- The lower level problem is parameterized by the upper level's decision x.
- The set $S(x)$ is also called rational reaction set of the follower. $S(x) = \{y \mid y = \arg\min_y f(x,y), \text{ s.t. } g(x,y) \geq 0\}$

Optimistic v.s. Pessimistic

When the rational reaction set $S(x)$ is not a singleton, which decision will the follower take?

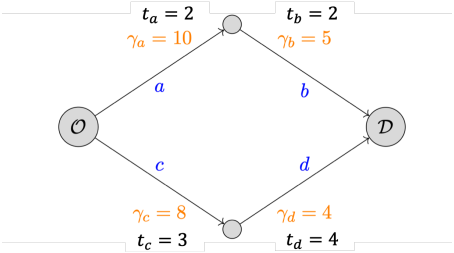

Figure: γ: base travel cost (travel time, gas cost, etc), t: toll. Adapted from Schmidt and Beck, 2023

Both the *ab* route and the *cd* route are optimal for the follower. If the follower takes the *cd* route, the leader will make more money (optimistic).

- Optimistic: When multiple responses are optimal, the follower takes the decision that work best for the leader.
- Pessimistic: When multiple responses are optimal, the follower takes the decision that work worst for the leader.

Optimistic formulation

upper-level problem

$$\begin{aligned} \min_{x \in X, y} \quad & F(x, y) \\ \text{s.t.} \quad & G(x, y) \geq 0, \\ & y \in S(x) \\ & S(x) = \{y \mid y = \arg\min_y f(x, y), \ \text{s.t.} \ g(x, y) \geq 0\} \end{aligned}$$

lower-level problem

$$\begin{aligned} \min_{y \in Y} \quad & f(x, y) \\ \text{s.t.} \quad & g(x, y) \geq 0. \end{aligned}$$

- Among all the y that minimizes the follower's objective, the one that minimizes the leader's objective is chosen.
- Note that in the leader's problem, we are minimizing over y as well.
- If $S(x)$ is a singleton for all feasible x, then there is no difference between optimistic and pessimistic.
- This is the most commonly used framework.

Pessimistic formulation

$$\min_{x \in X} \max_{y \in S(x)} F(x, y)$$

$$\text{s.t. } G(x, y) \geq 0,$$

- Choose the y that is the worst response for the leader.
- More challenging to solve and also less used.

Single Level Reformulation using the Optimal Value Function

We focus on the optimistic formulation.

$$\begin{array}{rl} \min_{x\in X, y} & F(x,y) \\ \text{s.t.} & G(x,y) \geq 0, \\ & y \in S(x), \end{array}$$

Define the value function of the follower's problem,

$$\varphi(x) := \min_{y\in Y}\{f(x,y) : g(x,y) \geq 0\}$$

We can reformulate the bilevel problem as a single level problem using the value function

$$\begin{array}{rl} \min_{x\in X, y\in Y} & F(x,y) \\ \text{s.t.} & G(x,y) \geq 0, g(x,y) \geq 0, \\ & f(x,y) \leq \varphi(x). \end{array}$$

This looks like a usual single-level problem. However, the problem is that the optimal-value function $\varphi : \mathbb{R}^{n_x} \to \mathbb{R}$ is difficult to obtain.

Reformulations based on Optimality conditions

- Replace $y \in S(x)$ by the optimality conditions (needs to be necessary and sufficient) of the lower level problem.

Recall the optimality conditions we've covered before.

- For LP (strong duality)
  1. primal feasibility
  2. dual feasibility
  3. primal and dual objectives are equal ($c^T x = p^T b$)

  or alternatively (KKT conditions),
  1. primal feasibility
  2. dual feasibility (stationarity)
  3. complementary slackness
- For convex NLP (KKT conditions)
  1. primal feasibility
  2. dual feasibility
  3. complementary slackness
  4. stationarity

KKT Reformulation for LP-LP Bilevel Problems

$$\begin{array}{ll} \min_{x,y} & c_x^\top x + c_y^\top y \\ \text{s.t.} & Ax + By \geq a, \\ & y \in \arg\min_{\bar{y}} \left\{ d^\top \bar{y} : Cx + D\bar{y} \geq b \right\} \end{array}$$

with $c_x \in \mathbb{R}^{n_x}, c_y, d \in \mathbb{R}^{n_y}, A \in \mathbb{R}^{m \times n_x}, B \in \mathbb{R}^{m \times n_y}$, and $a \in \mathbb{R}^m$ as well as $C \in \mathbb{R}^{\ell \times n_x}, D \in \mathbb{R}^{\ell \times n_y}$, and $b \in \mathbb{R}^\ell$.

The lower level problem can be replaced by its KKT condition:

- dual feasibility

$$D^\top \lambda = d, \quad \lambda \geq 0,$$

- primal feasibility

$$Cx + Dy \geq b,$$

- complementarity slackness

$$\lambda_i \left(C_i \cdot x + D_i \cdot y - b_i \right) = 0 \text{ for all } i = 1, \ldots, \ell.$$

where $C_i.$ denotes the i th row of C.

Convex lower-level problem

upper-level problem

$$\begin{aligned} \min_{x\in X,y} \quad & F(x,y) \\ \text{s.t.} \quad & G(x,y) \geq 0, \\ & y \in S(x) \\ & S(x) = \{y \mid y = \arg\min_y f(x,y), \ \text{s.t. } g(x,y) \geq 0\} \end{aligned}$$

lower-level problem

$$\begin{aligned} \min_{y\in Y} \quad & f(x,y) \\ \text{s.t.} \quad & g(x,y) \geq 0. \end{aligned}$$

- $y \mapsto f(x,y)$ is a convex function and that $y \mapsto g_i(x,y)$, $i = 1, \ldots, \ell$, are concave functions for all $x \in X$, i.e., for all feasible leader's decisions.
- Y is a convex set that, e.g., contains simple bound constraints on the variables of the follower.
- This means that the lower-level problem is indeed an x-parametric convex problem.
- Assume Slater's constraint qualification for the lower level.

KKT Reformulation for convex lower-level problem

upper-level problem

$$\begin{array}{rl} \min_{x \in X, y} & F(x,y) \\ \text{s.t.} & G(x,y) \geq 0, \\ & y \in S(x) \\ & S(x) = \{y \mid y = \arg\min_y f(x,y), \ \text{s.t.} \ g(x,y) \geq 0\} \end{array}$$

lower-level problem

$$\begin{array}{rl} \min_{y \in Y} & f(x,y) \\ \text{s.t.} & g(x,y) \geq 0. \end{array}$$

KKT reformulation

$$\begin{array}{rl} \min_{x,y,\lambda} & F(x,y) \\ \text{s.t.} & x \in X, \\ & \nabla_y \mathcal{L}(x,y,\lambda) = 0, \\ & g(x,y) \geq 0, \\ & \lambda \geq 0, \\ & \lambda^\top g(x,y) = 0. \end{array}$$

where $\nabla_y \mathcal{L}(x,y,\lambda) = \nabla_y f(x,y) - \sum_{i=1}^{\ell} \lambda_i \nabla_y g_i(x,y)$

Limitations of the KKT reformulation

- The main computational challenge of the KKT reformulation is the complementarity constraints (nonconvex).
- In the literature, this problem is called a mathematical program with complementarity constraints (MPCC) or Mathematical Programs with Equilibrium Constraints (MPEC). An active research area in nonsmooth optimization.
- A naive approach is to use a global solver for solving the MPCC but it won't scale.
- Another naive approach is to use an MILP formulation with big-M for each complementarity constraint,

$$g_i(x)\lambda_i = 0$$

can be reformulated as

$$\begin{gathered} 0 \leq g_i(x) \leq Mz_i \\ 0 \leq \lambda_i \leq M'(1 - z_i) \\ z_i \in \{0, 1\} \end{gathered}$$

MILP formulation of the KKT formulation

LP-LP

$$\begin{array}{ll} \min_{x,y} & c_x^\top x + c_y^\top y \\ \text{s.t.} & Ax + By \geq a, \\ & y \in \arg\min_{\bar{y}} \left\{ d^\top \bar{y} : Cx + D\bar{y} \geq b \right\} \end{array}$$

MILP formulation:

$$\begin{aligned} \min_{x,y,\lambda,z} \; & c_x^\top x + c_y^\top y \\ \text{s.t.} \; & Ax + By \geq a, Cx + Dy \geq b \\ & D^\top \lambda = d, \lambda \geq 0 \\ & \lambda_i \leq M z_i \quad \text{for all } i = 1, \ldots, \ell \\ & C_i \cdot x + D_i \cdot y - b_i \leq M\left(1 - z_i\right) \quad \text{for all } i = 1, \ldots, \ell \\ & z_i \in \{0, 1\} \quad \text{for all } i = 1, \ldots, \ell \end{aligned}$$

Strong duality-based reformulation for LP-LP

LP-LP

$$\begin{array}{ll} \min_{x,y} & c_x^\top x + c_y^\top y \\ \text{s.t.} & Ax + By \geq a, \\ & y \in \arg\min_{\bar{y}} \left\{ d^\top \bar{y} : Cx + D\bar{y} \geq b \right\} \end{array}$$

Strong duality formulation:

$$\begin{aligned} \min_{x,y,\lambda} & c_x^\top x + c_y^\top y \\ \text{s.t. } & Ax + By \geq a, Cx + Dy \geq b, \\ & D^\top \lambda = d, \lambda \geq 0 \\ & d^\top y \leq (b - Cx)^\top \lambda. \end{aligned}$$

nonconvex with bilinear terms $x^\top C\lambda$.

Lower level with integer variables

- We cannot use optimality conditions.
- Still an active research area. Typically, the algorithms involve branch-and-bound and/or cutting planes.

Robust Optimization

- Can be seen as a special case of bilevel optimization.
- Robust Optimization is an approach for handling uncertainty.
- Unlike stochastic programming, robust optimization does not require explicit probability distributions.
- Focuses on worst-case scenarios to ensure decisions are effective under all possible realizations of the uncertainty.
- Common in many fields including operations research, finance, and engineering.

Robust Optimization Problem Formulation

Problem Statement:

$$\begin{aligned} &\min_{x} \max_{c \in U_c} c^\top x \\ &\text{s.t. } a_i^\top x \leq b_i, \quad \forall a_i \in U_{a_i}, \forall b_i \in U_{b_i}, \quad i = 1, \ldots, m. \end{aligned}$$

Key Elements:

- Decision variable x is chosen to minimize the worst-case cost.
- The cost coefficients c, constraints coefficients a_i, and bounds b_i are uncertain but confined within sets U_c, U_{a_i}, and U_{b_i}. Typically, these sets are polyhedron or ellipsoids.

Equivalent Reformulation

The robust problem can be reformulated using an auxiliary variable α to linearize the objective:

$$\begin{aligned} &\min_{x,\alpha} \alpha \\ &\text{s.t. } c^\top x \leq \alpha, \quad \forall c \in U_c \\ &a_i^\top x \leq b_i, \quad \forall a_i \in U_{a_i}, \forall b_i \in U_{b_i}, \quad i = 1, \ldots, m. \end{aligned}$$

- α serves as the upper bound on the worst-case value of $c^\top x$.
- This transformation simplifies handling the max-min objective.
- Ensures the solution is feasible under all scenarios specified by the uncertainty sets.
- This formulation is called a **Semi-infinite program**. It has a finite number of variables and an infinite number of constraints when the uncertainty sets are polyhedron/ellipsoids.

Cutting Plane Algorithm

Master problem:

$$
\begin{aligned}
&\min_{x,\alpha} \alpha \\
&\text{s.t. } c^\top x \le \alpha, \quad \forall c \in U_c^k \\
&a_i^\top x \le b_i, \quad \forall a_i \in U_{a_i}^k, \forall b_i \in U_{b_i}^k, \quad i = 1, \ldots, m.
\end{aligned}
$$

- Start with a finite number of constraints at a given iteration k. Obtain the optimal solution of x^k, α^k
- Find the "most violated" constraints by solving

$$\max_c c^\top x^k - \alpha^k \quad \text{s.t.} c \in U_c$$

$$\max_{a,b} a_i^\top x^k - b_i^k \quad \text{s.t.} a_i \in U_{a_i}, b_i \in U_{b_i}$$

- If none of the constraints are violated, we terminate. Otherwise, add all the violated constraints to the master problem and repeat.

References

- Grossmann, I. E. (2021). Advanced optimization for process systems engineering. Cambridge University Press.
- Beck, Y., & Schmidt, M. (2021). A gentle and incomplete introduction to bilevel optimization.