

Theme B: Optimization Techniques for Food Systems

- Develop mathematical programming models to:
 - optimize supply chain architecture;
 - determine location of tracking devices and instrumentation; and
 - determine optimal responses to contamination events.
- **Major challenge:** the mathematical programming models might only be able to represent a “partial, distorted view” of the system.
 - for instance, constraints/variables might be missing; and
 - the objective function might be difficult or impossible to formulate.
- Our approach, **from a CS perspective**, will focus on understanding whether approximately optimal solutions can still be reached.
 - Theme C explores the design and analysis of mechanisms to incentivize the actors of the supply chain to reveal additional information about the system.

Problem setup:

- **Input:** network models (graphs).
 - *Nodes* correspond to producers, distributors, and retailers.
 - *Edges* correspond to costs, capacities, and flows.
- **Output:** supply chain designs.
 - Must support accurate traceback.
 - Must include sensor localization to optimize data collection.
 - Must be able to identify contamination and its spread.

Formulation

- Existing mathematical programming formulations of such supply chain problems result in *optimization problems* that:
 - **Take as input** the supply chain network structure; its capacity and other costs; a range of contamination scenarios; and network constraints.
 - **Return as output** the location and nature of tracking devices and the traceback flow under each contamination scenario.
- The above modelling process results in massive optimization problems, often convex.
- The “wastage” problem also has mathematical programming formulations, modelling total waste, cost of waste, and cost of supply chain.

Technical Challenge: Partial System View

- The *constraint set* is partially specified:
 - various actors might not reveal their constraints at all, or
 - they might reveal “sanitized” constraints, to preserve their privacy.
- The *objective function* might be hard to formulate explicitly:
 - costs in the supply chain might interact in complicated ways; and
 - privacy considerations could result in an incomplete objective function.

Technical Approach

- The **Computer Science** perspective: sensitivity theory for optimization.
 - Understand how noise/missing/partial constraints and/or an approximate objective function affect the quality of the solution.
- **Our approach:** look at the “dual” perspective, e.g., analyze sampling approaches to solve optimization problems.
 - **Starting point:** assume that the underlying optimization problem is simple, e.g., a Linear Program (LP: linear objective function and linear constraints).
 - We will seek to understand the effect of sub-sampling or adding noise to the variable and/or the constraint space to the set of feasible solutions.
 - We will seek to characterize settings where approximately optimal solutions can be identified.
 - Moving from LPs to convex programming and addressing similar questions is a major open problem.
- Our work will go well beyond classical sensitivity theory, which typically deals with small noise in the input.

Technical Approach

To make this more concrete, consider the following simple LP (packing) setting:

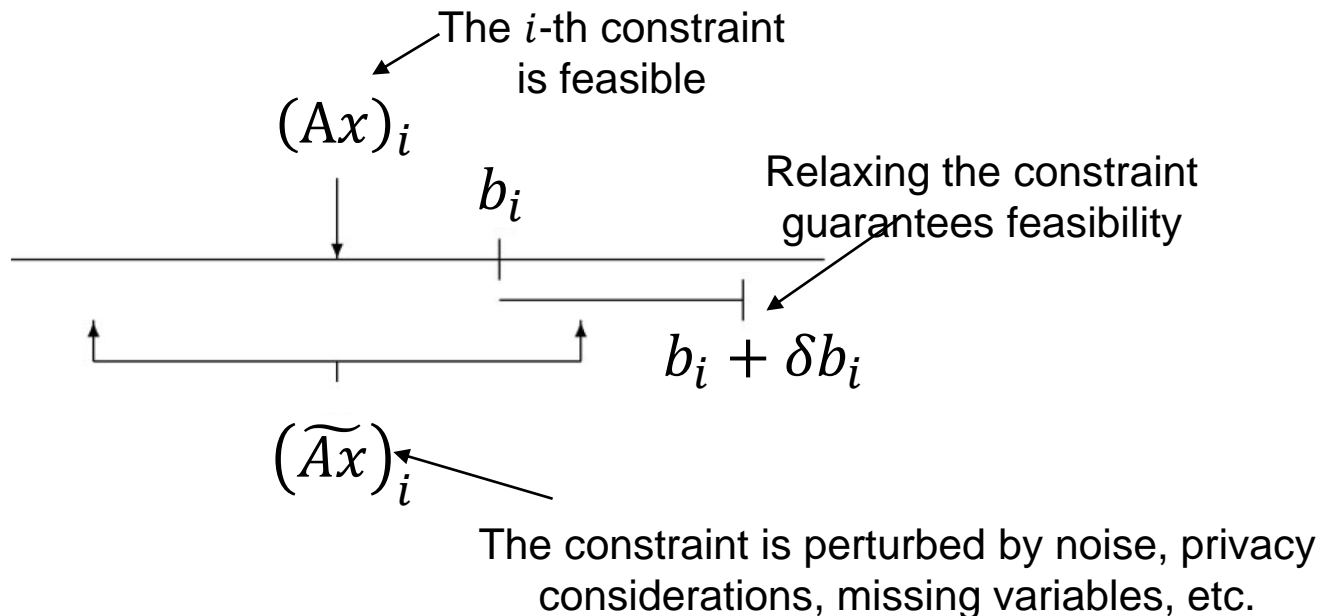
$$Ax \leq b, 0 \leq x \leq 1$$

Here A is an $m \times n$ matrix and x (and b) are vectors in R^n (and R^m).

- Assume that the above LP is feasible, e.g., has a solution vector x that satisfies all the constraints.
- Assume that some constraints are perturbed, e.g., for some or all $i = 1 \dots m$, $(Ax)_i$ is replaced by $(\widetilde{A}x)_i$.
 - This could be the result of missing variables, noise, privacy preservation, etc.
- Can we guarantee that the perturbed system still has a feasible solution? We would like the answer to depend on the structure of A ; we are willing to relax b_i to $b_i + \delta b_i$.
 - We would like to minimize δb_i depending on the structure of A .

Even this very simple setting is non-trivial to formally investigate (some prior work exists).

Technical Approach



Thoroughly understanding such questions for convex programming is an important objective of the proposed research.

Technical Approach

- **Complementary aspect:** Each actor solves an optimization problem on his constraints and variables and optimization function, **but** the actor is unwilling to reveal his local optimization problem (at least to its entirety) to the other actors in order to solve the global optimization problem.
- **Simplified model:** Let the underlying optimization problem G be broken up into smaller problems G_1, G_2, \dots, G_p (one problem for each of the p actors).
- **The Computer Science perspective:** assuming that each actor can solve its respective optimization problem G_1, G_2, \dots, G_p , what is the minimal amount of information that each actor should exchange with other actors in order to (at least approximately) solve the overall problem?
 - That could include partially revealing the local optimization problem G_i , a sketch of G_i , the local optimal solution, etc.
- Connections with distributed LP solvers; little is known for convex opt

Novelty of Proposed Approach

- Sensitivity analysis of LPs: fundamental research area.
- Many important questions are still open:
 - Impact of missing variables/constraints on feasibility.
 - Impact of missing variables/constraints on optimal solution.
 - Early stopping for iterative algorithms in the presence of noise: overfitting to noisy inputs wastes computational time and leads to poor generalization performance.
- **Our approach:** we will use Randomized Linear Algebra to understand the behavior of the optimization problems in the presence of noise.