# Learning Decision-Sufficient Datasets for Linear Optimization

Omar Bennouna, Amine Bennouna, Yuhan Ye,
Saurabh Amin, and Asu Ozdaglar

Department of Electrical Engineering and Computer Science, MIT
Department of Civil and Environmental Engineering, MIT
Kellogg School of Management, Northwestern University

STOC26 Workshop on ML for Algo Design
June, 2026

# Data-Driven Decisions

- In many real-world decision problems, optimization objectives (or cost) depend on parameters that are unknown and must be inferred from data.
  - Examples include routing problems with unknown travel times, weighted matching problems with unknown weights, hiring or secretary problems with unknown values.
- There is recent growing interest in finding cost prediction models that minimize downstream decision errors – **Smart predict, then optimize (SPO)** [Elmachtaub, Grigas 20]
- We focus on an alternative inquiry for linear optimization problems:

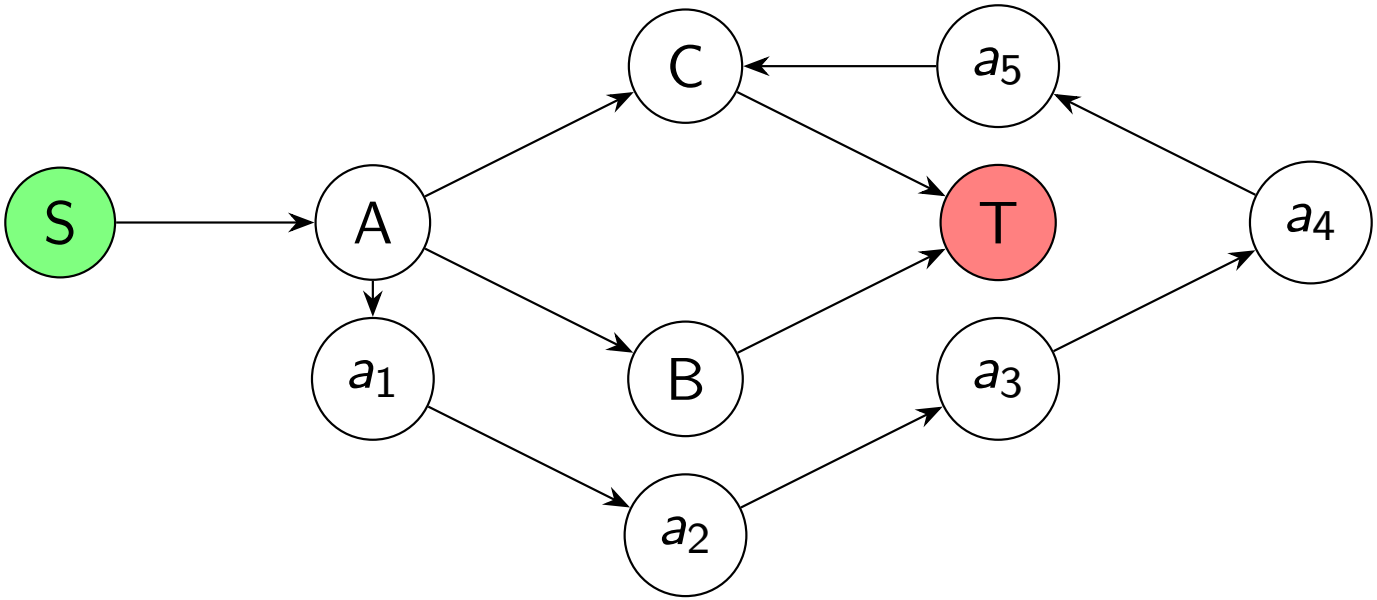
**Key Questions:**

- What data is sufficient to recover optimal solutions in linear programs with an unknown cost vector?
- How do we learn "compressed datasets" that perform well on future problem instances?

# Data Sufficiency in a Toy Example

- Consider a shortest path problem with source S and target T.
- Suppose the edge costs $c \in \mathbb{R}^{11}$ are unknown—Data corresponds to edge costs.

Which data informs the optimal decision here?

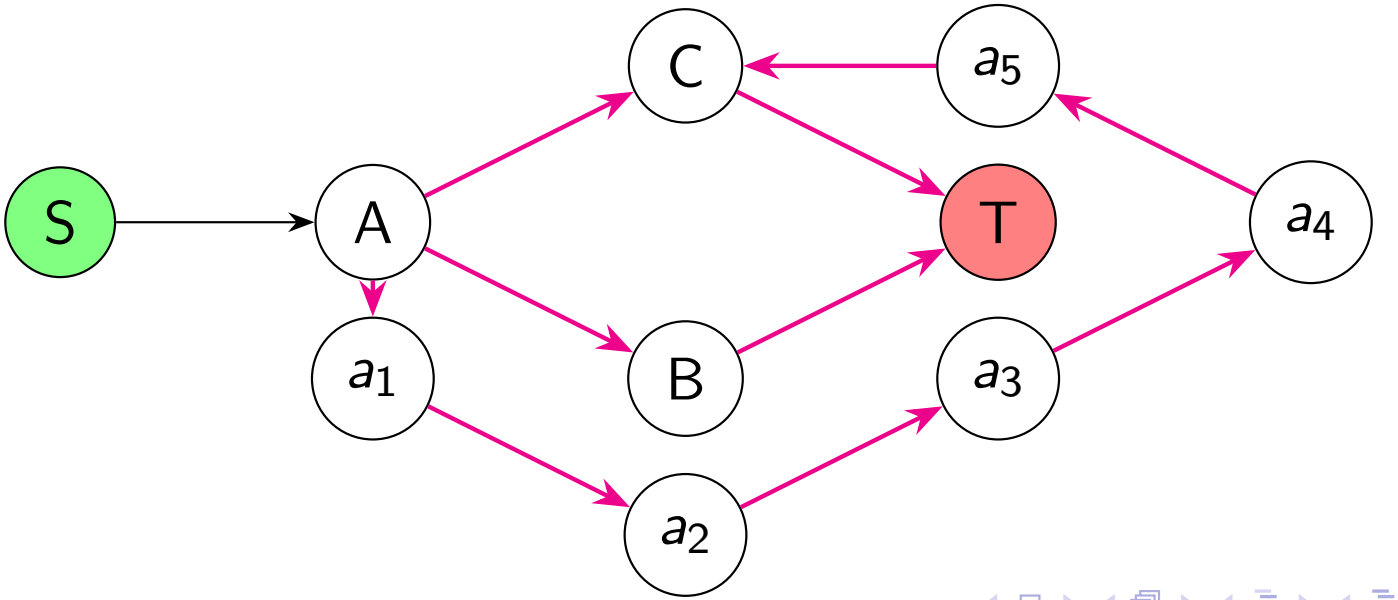


# Data Sufficiency in a Toy Example

- Consider a shortest path problem with source S and target T.
- Suppose the edge costs $c \in \mathbb{R}^{11}$ are unknown—Data corresponds to edge costs.

Which data informs the optimal decision here?

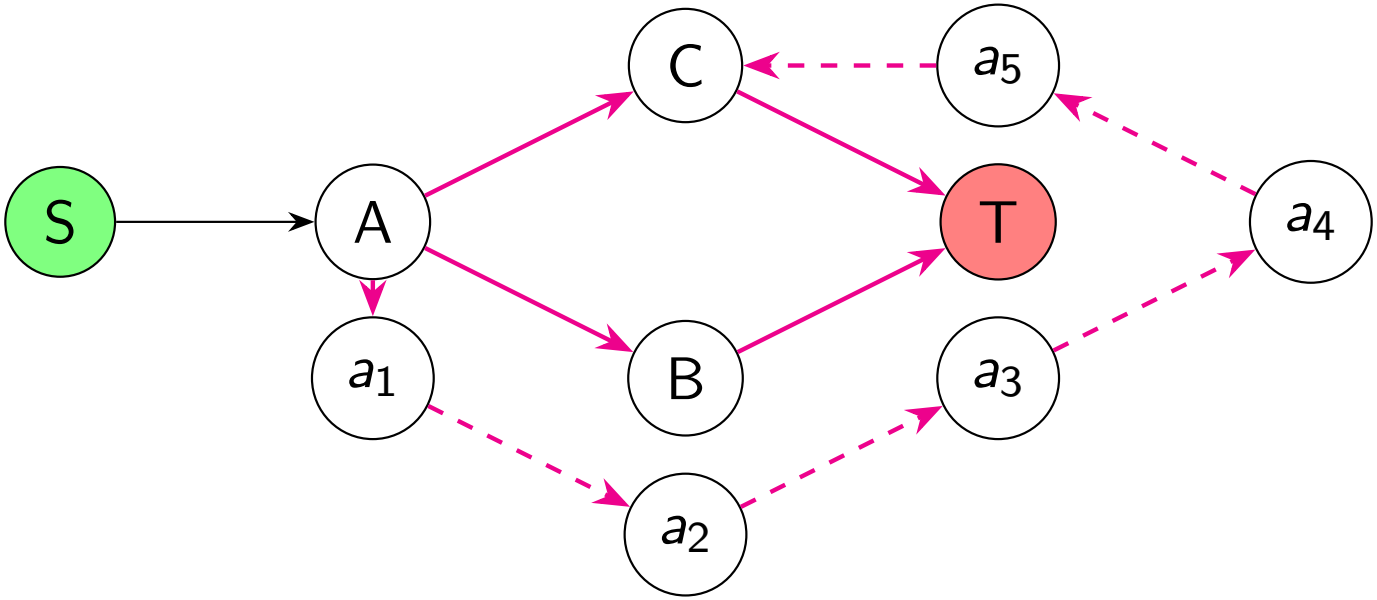
- With no prior information: You need all edge costs except $c_{SA}$ as $SA$ is always taken by any feasible decision.



# The Role of Prior Information

Which data informs the optimal decision here?

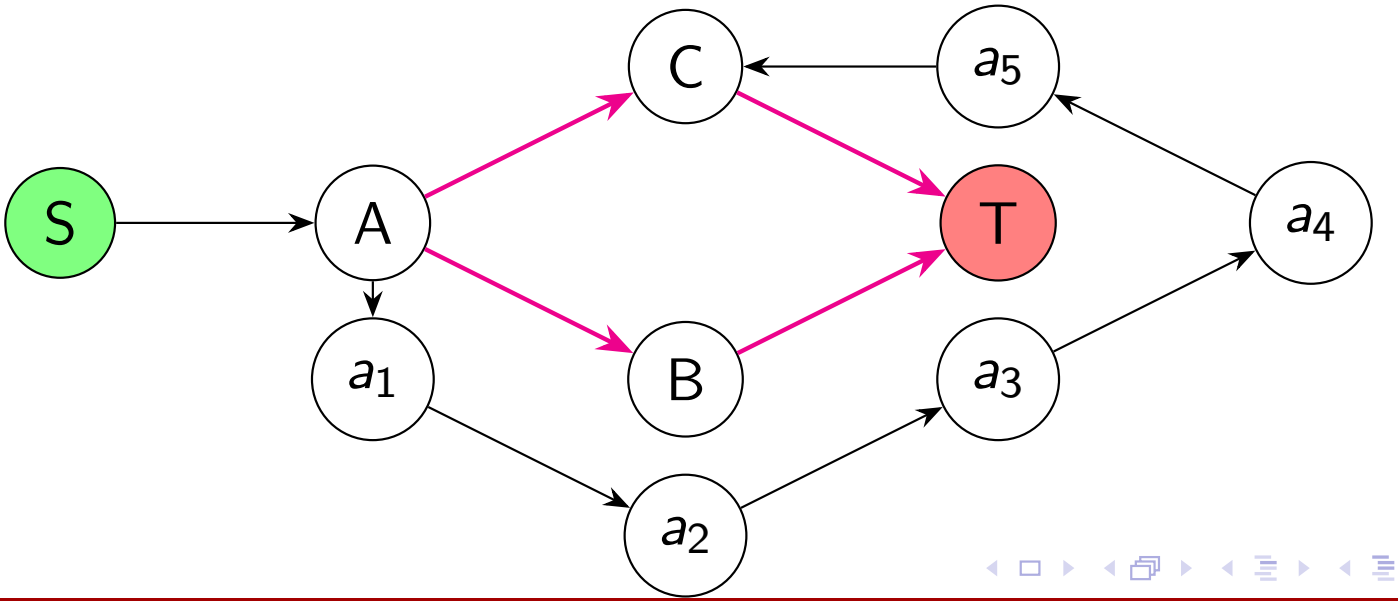
- With no prior information: You need all edge costs except $c_{SA}$ as $SA$ is always taken by any feasible decision.
- If this were a map on scale, we intuitively want to say that we don't need the "detour" $a_1, \ldots, a_5$ as it is likely always a suboptimal choice.



# The Role of Prior Information

Which data informs the optimal decision here?

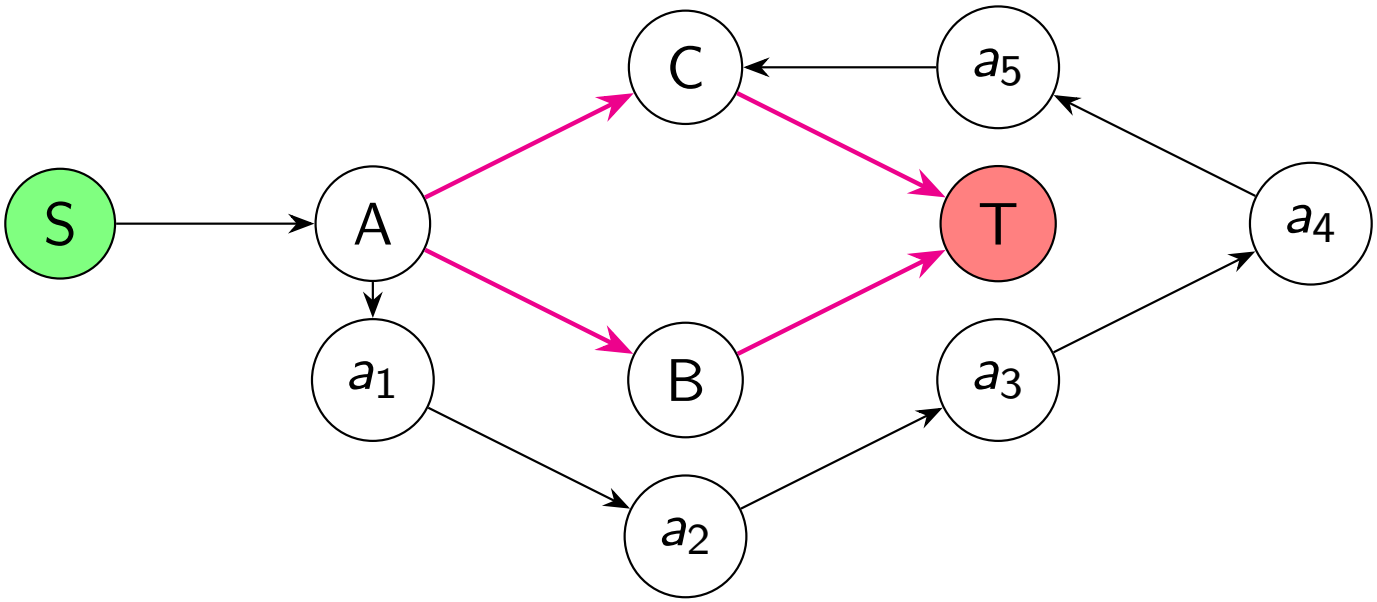
- With no prior information: You need all edge costs except $c_{SA}$ as $SA$ is always taken by any feasible decision.
- If this were a map on scale, we intuitively want to say that we don't need the "detour" $a_1, \ldots, a_5$ as it is likely always a suboptimal choice.
- With prior information $c \in \mathcal{C} := [1, 5]^{11}$: we only need $c_{AC}, c_{CT}$ and $c_{AB}, c_{BT}$. Prior information is important!



# The Role of Prior Information

Which data informs the optimal decision here?

- With prior information $c \in \mathcal{C} := [1, 5]^{11}$: we only need $c_{AC}, c_{CT}$ and $c_{AB}, c_{BT}$. Prior information is important!
- In fact, we only need the information $c_{AC} + c_{CT}$ and $c_{AB} + c_{BT}$.
- In practice, prior information comes from features such as edge costs being correlated with length, congestion, speed limit, historical travel time etc...



# Our Results

- We first present a geometric characterization of global sufficient decision datasets (SDD), which are sufficient to recover optimal decisions of a linear program for any cost in the uncertainty set.
- A global SDD captures all decision-relevant directions, whose size $d^*$ often much smaller than problem dimension.
- We show that computing the size of a minimal global SDD and its construction are NP-hard.

# Our Results

- We first present a geometric characterization of global sufficient decision datasets (SDD), which are sufficient to recover optimal decisions of a linear program for any cost in the uncertainty set.
- A global SDD captures all decision-relevant directions, whose size $d^*$ often much smaller than problem dimension.
- We show that computing the size of a minimal global SDD and its construction are NP-hard.

Going beyond worst-case analysis, we study learning decision-sufficient datasets using instances from an unknown distribution, with generalization guarantees.

- We introduce pointwise sufficiency, a relaxation that requires sufficiency only for an individual realized cost vector.
  - Under nondegeneracy, we provide a polynomial-time cutting-plane algorithm for constructing pointwise-sufficient decision datasets.
- We propose an algorithm that aggregates decision-relevant directions across samples, yielding a stable compression scheme of size at most $d^*$ with strong distribution-free learning guarantees.

# Background: Related Literature

- **Model compression for LP:** How to replace a large LP by a lower-dimensional or sketched formulation while preserving the relevant geometry for optimization.
  [Vu18], [Poirion23], [Sakaue–Oki24].
- **Data-driven algorithm design:** Selecting the best algorithm where samples from problem instances used to choose algorithm parameters with statistical performance guarantees.
  [Gupta–Roughgarden17/20], [Balcan et al. 17/18/21/24]
- **Sample compression scheme:** Choosing a subset of training examples with generalization guarantees (or low risk). When the learning certificate is stable, realizability yields fast distribution-free generalization bounds.
  [Valiant84], [Littlestone–Warmuth86], [Hanneke–Kontorovich21]
- **Active learning:** Instead of observing all information up front, seeks to select data points through sequential and adaptive exploration and rely on real-time feedback to guide data acquisition. [Dasgupta11], [Hanneke14], [Balcan et al. 06/09].

**Our Work.** In LPs, we learn a dataset (i.e., linear measurements of the unknown cost) that are sufficient for exact optimal-decision recovery.
[Bennouna et al. 25/26], [Ye et al. 26]

# Problem Formulation

- Decision-task: a linear program $\min_{x \in \mathcal{X}} c^\top x$, where $\mathcal{X} \subset \mathbb{R}^d$ is a bounded polyhedron.
- Prior information: The cost vector $c$ is unknown. Prior information is modeled through an uncertainty set $c \in \mathcal{C} \subset \mathbb{R}^d$. E.g.

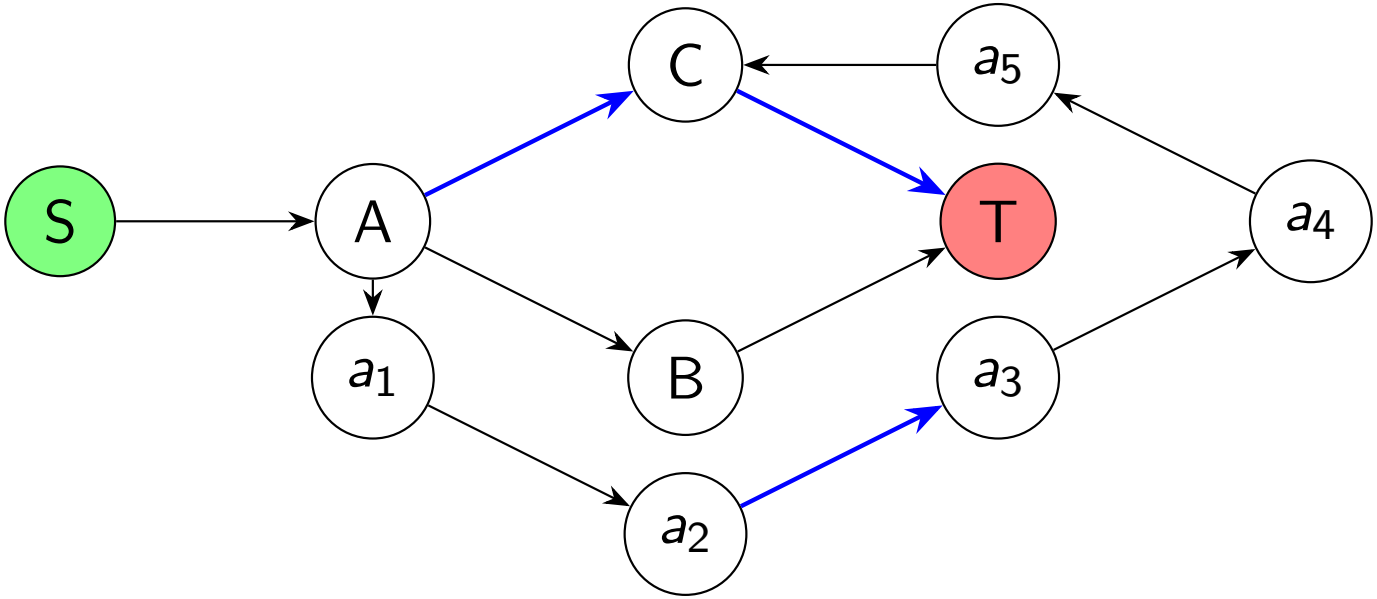
$$\mathcal{C} = \{c \in \mathbb{R}^d \ : \ \exists \alpha \in [\underline{\alpha}, \bar{\alpha}], \ \exists \epsilon \in [-\eta, \eta]^d, \ c = \alpha^\top \Phi + \epsilon\}$$

  for some features $\Phi \in \mathbb{R}^{k \times d}$.
- Dataset: Consists of a set of queries $\mathcal{D} = \{q_1, \ldots, q_N\} \subset \mathbb{R}^d$, which provides the observations $c^\top q_1, \ldots, c^\top q_N$.
  - Allows to measure linear combinations of unknown cost values.
  - Focus on noiseless case in this talk.

# Problem Formulation

- Decision-task: a linear program $\min_{x \in \mathcal{X}} c^\top x$, where $\mathcal{X}$ is a polyhedron.
- Prior information: We know that $c \in \mathcal{C}$.
- Dataset: $\mathcal{D} = \{q_1, \ldots, q_N\} \subset \mathbb{R}^d$ provides the observations $c^\top q_1, \ldots, c^\top q_N$.

*Example:* $\mathcal{D} = \{\delta_{AC} + \delta_{CT}, \delta_{a_2 a_3}\}$, where $\delta_e$ is the indicator vector at edge $e$, provides observations $\{c_{AC} + c_{CT}, c_{a_2 a_3}\}$.



# Problem Formulation: Sufficient Datasets

- Decision-task: a linear program $\min_{x \in \mathcal{X}} c^\top x$, where $\mathcal{X}$ is a polyhedron.
- Prior information: We know that $c \in \mathcal{C}$.
- Dataset: $\mathcal{D} = \{q_1, \ldots, q_N\}$ provides the observations $c^\top q_1, \ldots, c^\top q_N$.

*Given a decision task (defined by $\mathcal{X}$) and prior information (defined by $\mathcal{C}$), what dataset enables finding the task-optimal decision?*

# Problem Formulation: Sufficient Datasets

## Definition (Sufficient Dataset)

A set $\mathcal{D} := \{q_1, \ldots, q_N\}$ is sufficient for uncertainty set $\mathcal{C}$ and decision set $\mathcal{X}$ if there exists a mapping $\hat{x} : \mathbb{R}^N \longrightarrow \mathcal{X}$ such that

$$\forall c \in \mathcal{C}, \quad \hat{x}\left(c^\top q_1, \ldots, c^\top q_N\right) \in \arg\min_{x \in \mathcal{X}} \; c^\top x.$$

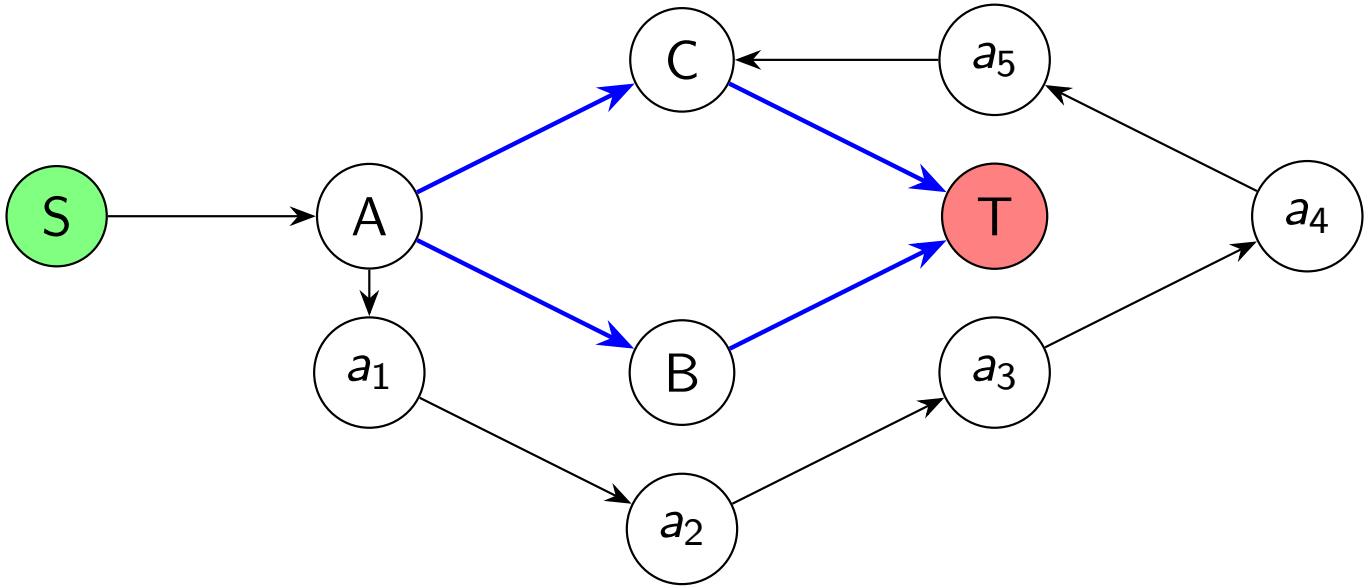
# Problem Formulation: Sufficient Datasets

## Definition (Sufficient Dataset)

A set $\mathcal{D} := \{q_1, \ldots, q_N\}$ is sufficient for uncertainty set $\mathcal{C}$ and decision set $\mathcal{X}$ if there exists a mapping $\hat{x} : \mathbb{R}^N \longrightarrow \mathcal{X}$ such that

$$\forall c \in \mathcal{C}, \quad \hat{x}\left(c^\top q_1, \ldots, c^\top q_N\right) \in \arg\min_{x \in \mathcal{X}} c^\top x.$$

$\mathcal{D} = \{\delta_{AC} + \delta_{CT}, \delta_{AB} + \delta_{BT}\}$ is sufficient for $\mathcal{C} = [1, 5]^{11}$ in the example below.



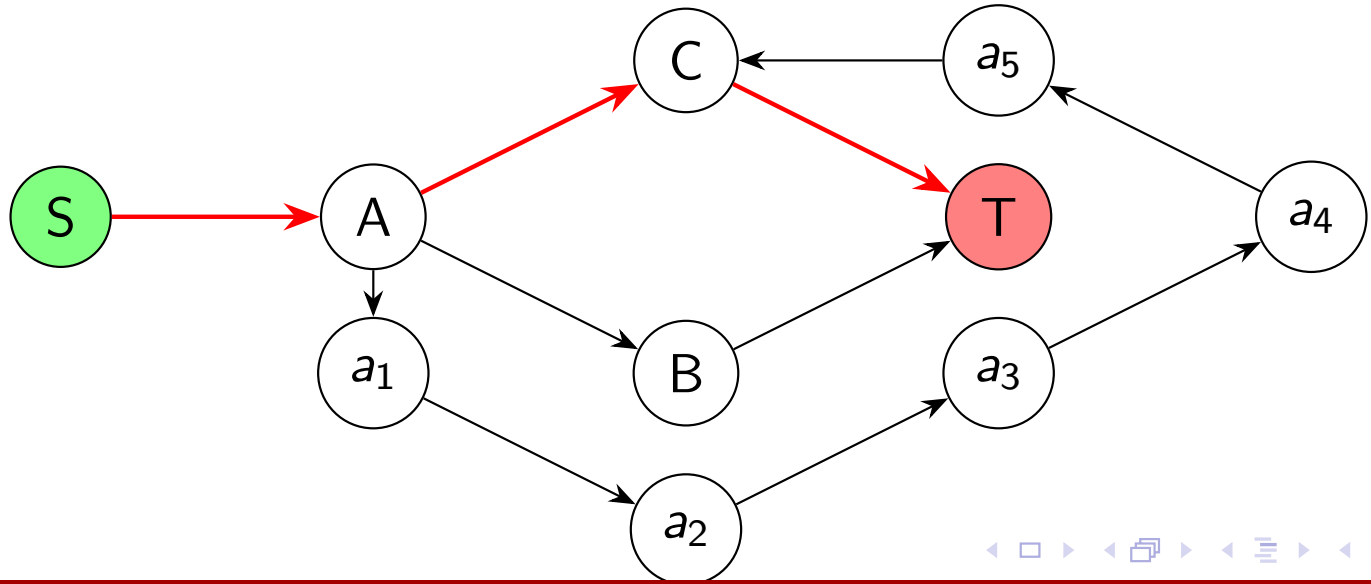
# Problem Formulation: Sufficient Datasets

## Definition (Sufficient Dataset)

A set $\mathcal{D} := \{q_1, \ldots, q_N\}$ is sufficient for uncertainty set $\mathcal{C}$ and decision set $\mathcal{X}$ if there exists a mapping $\hat{x} : \mathbb{R}^N \longrightarrow \mathcal{X}$ such that

$$\forall c \in \mathcal{C}, \quad \hat{x}\left(c^\top q_1, \ldots, c^\top q_N\right) \in \arg\min_{x \in \mathcal{X}} c^\top x.$$

$\mathcal{D} = \{\delta_{AC} + \delta_{CT}, \delta_{AB} + \delta_{BT}\}$ is sufficient for $\mathcal{C} = [1, 5]^{11}$ in the example below. In fact, $\hat{x}$ is: if $c_{AC} + c_{CT} < c_{AB} + c_{BT}$, then opt is



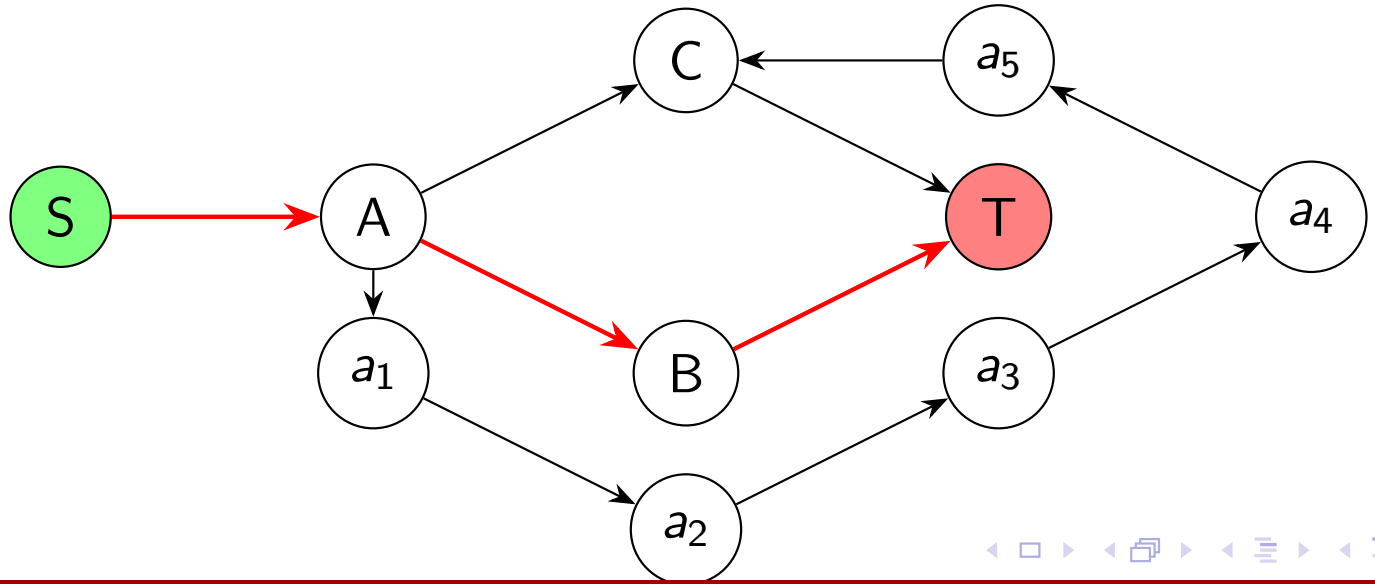
# Problem Formulation: Sufficient Datasets

## Definition (Sufficient Dataset)

A set $\mathcal{D} := \{q_1, \ldots, q_N\}$ is sufficient for uncertainty set $\mathcal{C}$ and decision set $\mathcal{X}$ if there exists a mapping $\hat{x} : \mathbb{R}^N \longrightarrow \mathcal{X}$ such that

$$\forall c \in \mathcal{C}, \quad \hat{x}\left(c^\top q_1, \ldots, c^\top q_N\right) \in \arg\min_{x \in \mathcal{X}} c^\top x.$$

$\mathcal{D} = \{\delta_{AC} + \delta_{CT}, \delta_{AB} + \delta_{BT}\}$ is sufficient for $\mathcal{C} = [1, 5]^{11}$ in the example below. In fact, $\hat{x}$ is: if $c_{AC} + c_{CT} > c_{AB} + c_{BT}$, then opt is



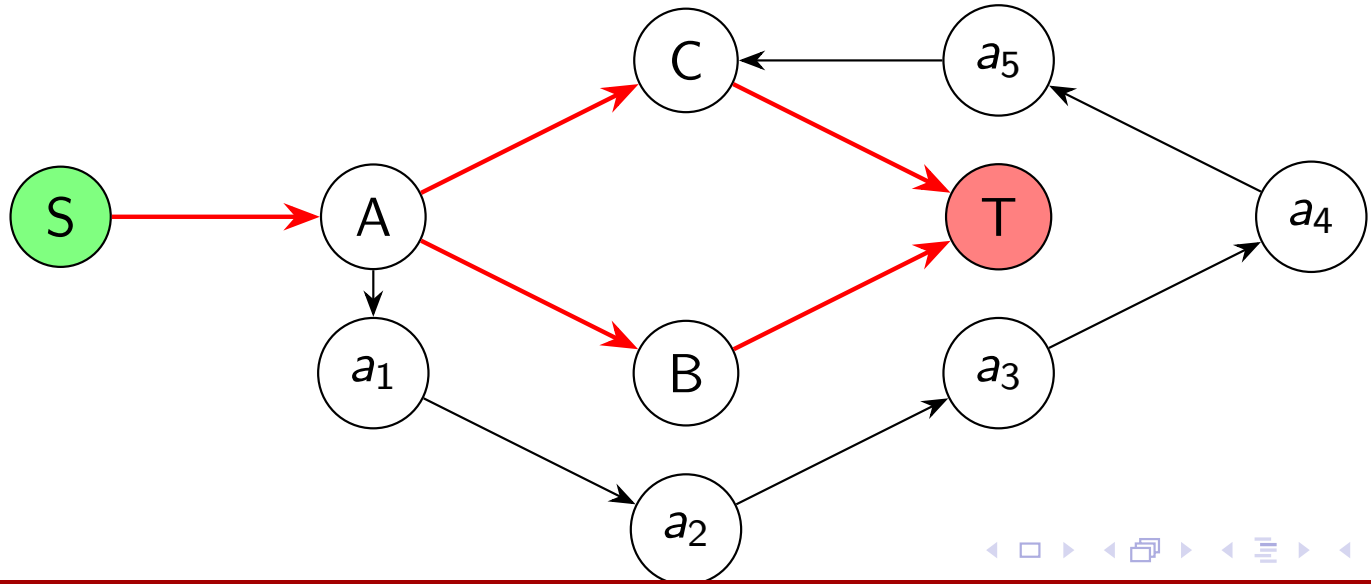
# Problem Formulation: Sufficient Datasets

## Definition (Sufficient Dataset)

A set $\mathcal{D} := \{q_1, \ldots, q_N\}$ is sufficient for uncertainty set $\mathcal{C}$ and decision set $\mathcal{X}$ if there exists a mapping $\hat{x} : \mathbb{R}^N \longrightarrow \mathcal{X}$ such that

$$\forall c \in \mathcal{C}, \quad \hat{x}\left(c^\top q_1, \ldots, c^\top q_N\right) \in \arg\min_{x \in \mathcal{X}} c^\top x.$$

$\mathcal{D} = \{\delta_{AC} + \delta_{CT}, \delta_{AB} + \delta_{BT}\}$ is sufficient for $\mathcal{C} = [1, 5]^{11}$ in the example below. In fact, $\hat{x}$ is: if $c_{AC} + c_{CT} = c_{AB} + c_{BT}$, then both are opt



# Characterizing Sufficient Datasets

# Useful Characterization for Sufficient Datasets

## Definition (Sufficient Dataset)

A set $\mathcal{D} := \{q_1, \ldots, q_N\}$ is sufficient for uncertainty set $\mathcal{C}$ and decision set $\mathcal{X}$ if there exists a mapping $\hat{x} : \mathbb{R}^N \longrightarrow \mathcal{X}$ such that

$$\forall c \in \mathcal{C}, \quad \hat{x}\left(c^\top q_1, \ldots, c^\top q_N\right) \in \arg\min_{x \in \mathcal{X}} c^\top x.$$

- Notice that observing $c^T q$ for all $q \in \mathcal{D}$ is equivalent to observing the projection of $c$ onto the span of $\mathcal{D}$.

## Proposition

*$\mathcal{D} := \{q_1, \ldots, q_N\}$ is sufficient for uncertainty set $\mathcal{C}$ and decision set $\mathcal{X}$ if and only if*

$$\forall c, c' \in \mathcal{C}, \quad c_{\text{span } \mathcal{D}} = c'_{\text{span } \mathcal{D}} \Longrightarrow \arg\min_{x \in \mathcal{X}} c^\top x = \arg\min_{x \in \mathcal{X}} c'^\top x.$$

# Warm-up: Characterization Under No Prior Information

Let us write $\mathcal{X} = \{x \in \Re^d \ : \ Ax = b, \ x \geq 0\}$.

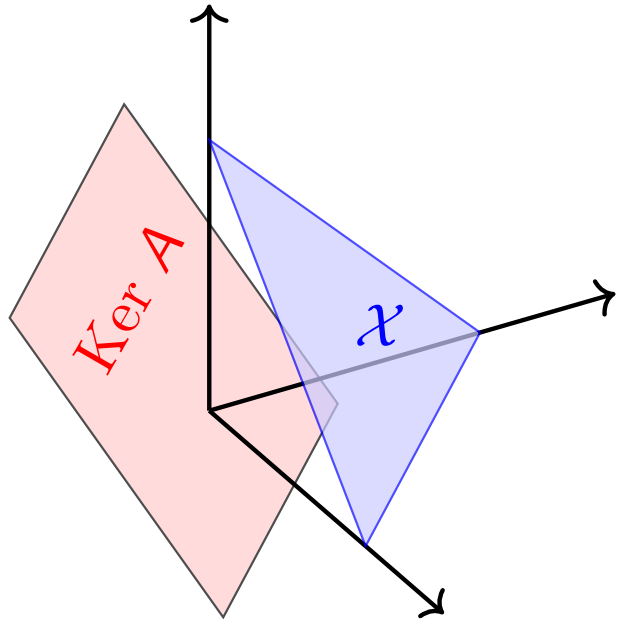
**Proposition**

*Suppose $\mathcal{C} = \mathbb{R}^d$ (no prior information). $\mathcal{D}$ is a sufficient dataset if and only if*

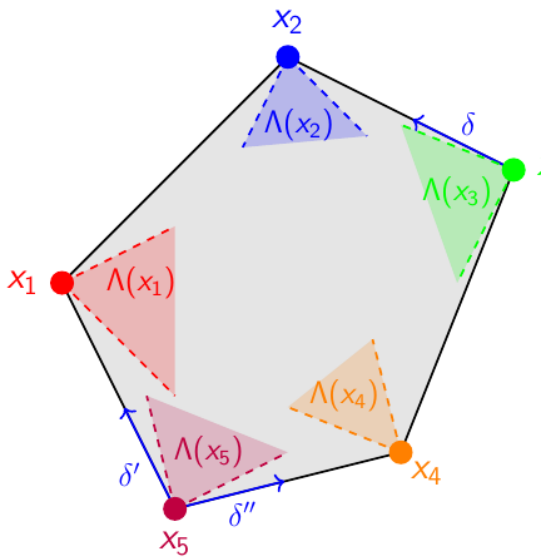
$$\text{Ker } A \subset \text{span } \mathcal{D}$$

*Intuition:*

- Can decompose the objective function as $c^\top_{\text{Ker } A} x + c^\top_{(\text{Ker } A)^\perp} x$.
- Any change from a feasible $x$ to feasible $x + \delta$ satisfies $\delta \in \text{Ker } A$.
- Hence, $c^\top_{(\text{Ker } A)^\perp} x$ is constant across all feasible solutions.

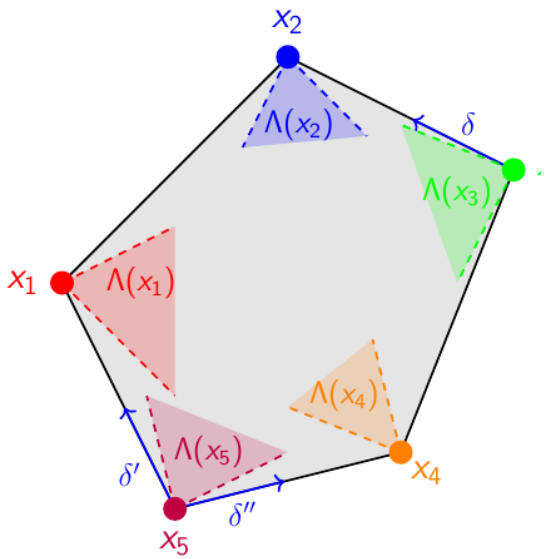


# Linear Programming Basics




**Extreme Points** ($\mathcal{X}^{\angle}$). $x \in \mathcal{X}$ with no $\lambda \in (0,1)$ and $y, z \in \mathcal{X}$ such that $x = \lambda y + (1-\lambda)z$.

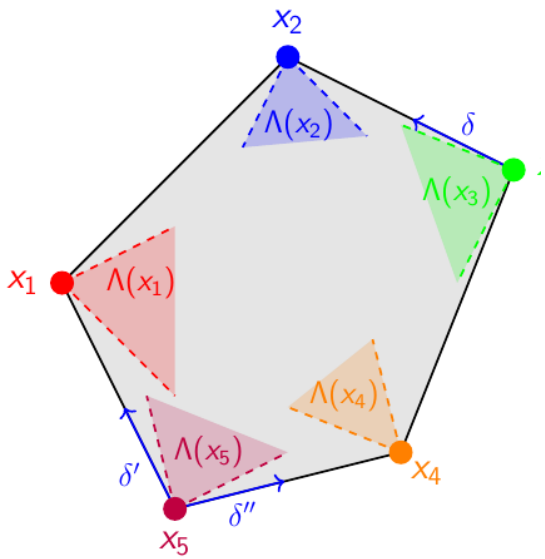
# Linear Programming Basics




**Extreme Points** $(\mathcal{X}^{\angle})$**.** $x \in \mathcal{X}$ with no $\lambda \in (0,1)$ and $y, z \in \mathcal{X}$ such that $x = \lambda y + (1-\lambda)z$.

**Extreme Directions** $(D(x^\star))$**.** For every $x^\star \in \mathcal{X}^{\angle}$, its extreme directions are feasible directions $(\delta \in \mathbb{R}^d \ : \ \exists \varepsilon > 0, \ x^\star + \varepsilon\delta \in \mathcal{X})$ that cannot be written as a convex combination of two non-proportional feasible directions.

# Linear Programming Basics

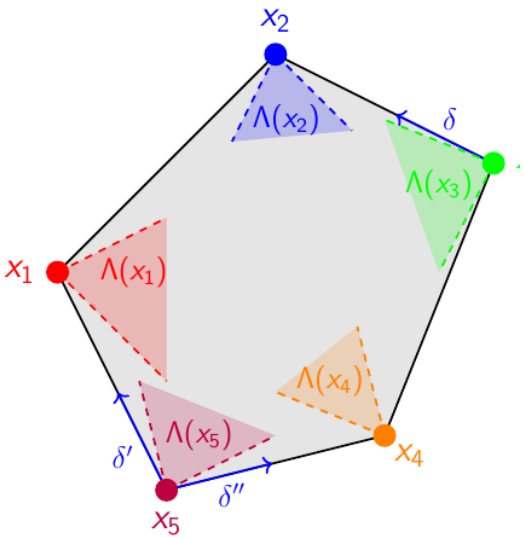


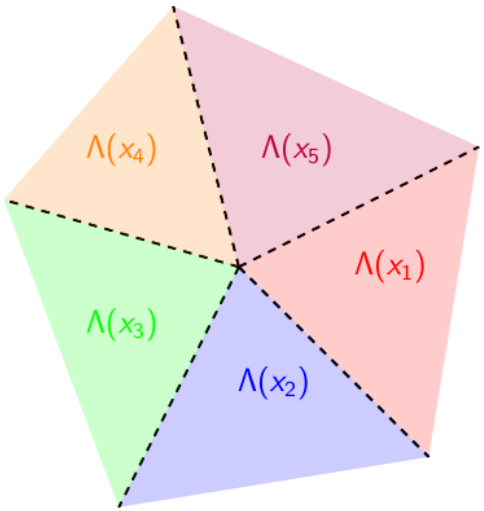

**Extreme Points** ($\mathcal{X}^{\angle}$). $x \in \mathcal{X}$ with no $\lambda \in (0,1)$ and $y, z \in \mathcal{X}$ such that $x = \lambda y + (1-\lambda)z$.

**Extreme Directions** ($D(x^\star)$). For every $x^\star \in \mathcal{X}^{\angle}$, its extreme directions are feasible directions ($\delta \in \mathbb{R}^d \ : \ \exists \varepsilon > 0, \ x^\star + \varepsilon\delta \in \mathcal{X}$) that cannot be written as a convex combination of two non-proportional feasible directions.

**Optimality Cones** ($\Lambda(x^\star)$). Every $x^\star \in \mathcal{X}^{\angle}$ is associated with a cone $\Lambda(x^\star) = \{c \in \mathbb{R}^d \ : \ x^\star \in \arg\min_{x \in \mathcal{X}} c^\top x\}$.

# Optimality Cones

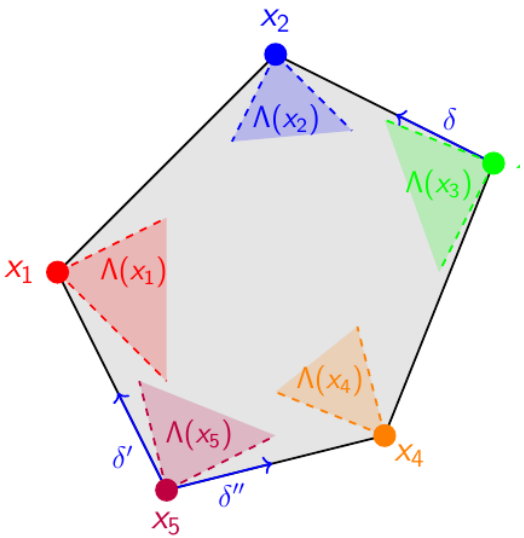


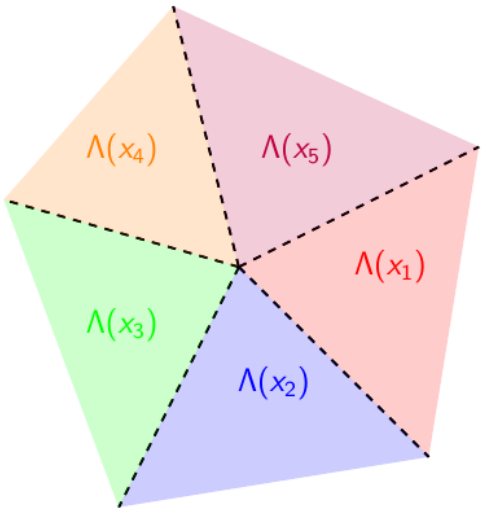



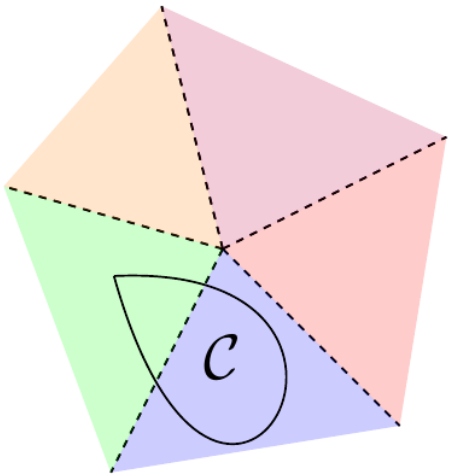

$x_i$ Extr Points
$\Lambda(x^\star)$ Opt Cones
$\delta, \delta', \delta''$ Extr Directions

- **Optimality Cones.** For $x^\star$ extr point, $\Lambda(x^\star) = \{c \ : \ x^\star \in \arg\min_{x \in \mathcal{X}} c^\top x\}$.
- The cones form a partition: $\bigcup_{x^\star \in \mathcal{X}^\angle} \Lambda(x^\star) = \mathbb{R}^d$.
- $c \in \Lambda(x^\star) \iff x^\star$ is optimal for cost $c$.
- Solving an LP with cost $c \iff$ finding which cone $\Lambda(x^\star)$ $c$ belongs to.

# A Geometric Perspective on Dataset Sufficiency

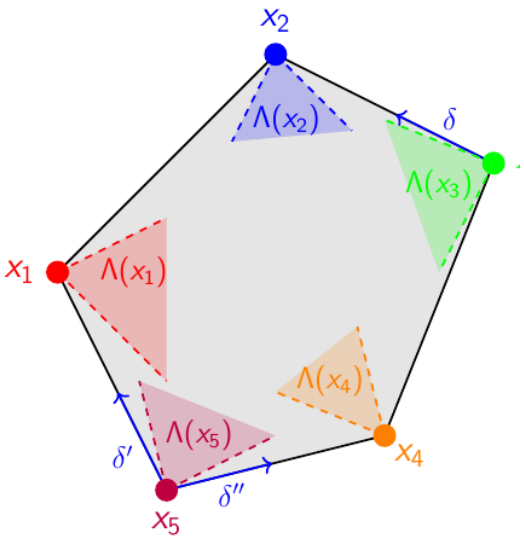


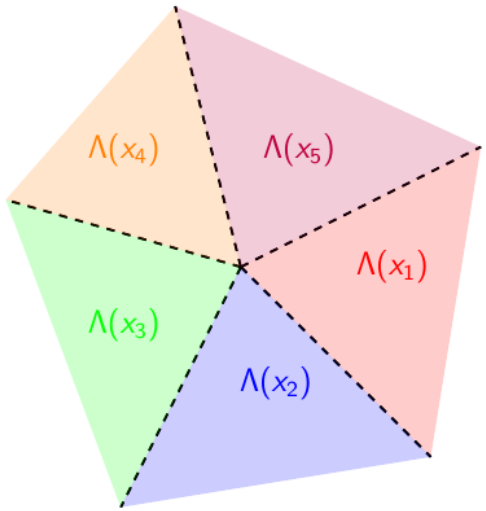



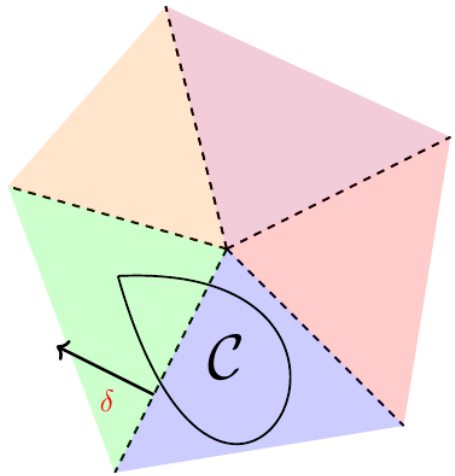




- *Back to our problem:* A sufficient dataset $\mathcal{D}$ needs therefore to *locate* $c$ with respect to the cones, using the projections $c^\top q, q \in \mathcal{D}$.
- Given an uncertainty set $\mathcal{C}$: A sufficient dataset needs to discriminate the remaining possible cones in $\mathcal{C}$—$\Lambda(x_2)$ and $\Lambda(x_3)$ in the example.

# A Geometric Perspective on Dataset Sufficiency

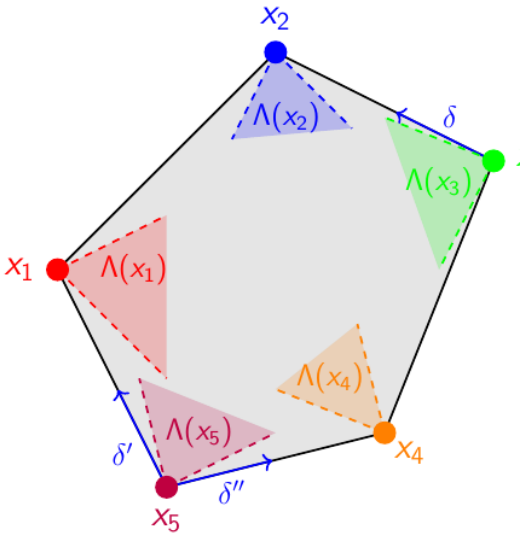


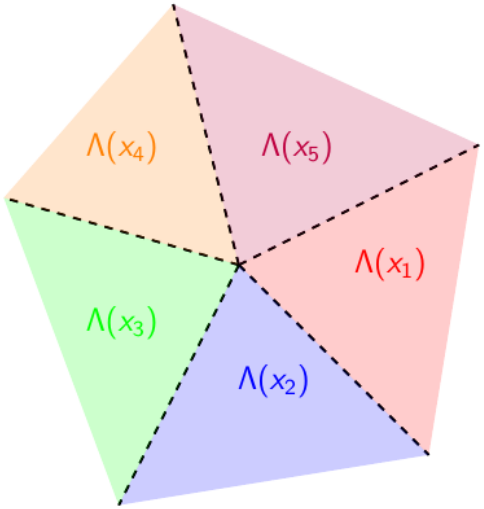



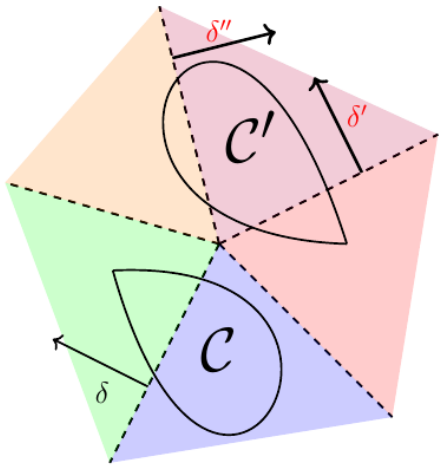




- *Back to our problem:* A sufficient dataset $\mathcal{D}$ needs therefore to *locate* $c$ with respect to the cones, using the projections $c^\top q, q \in \mathcal{D}$.
- Given an uncertainty set $\mathcal{C}$: A sufficient dataset needs to discriminate the remaining possible cones in $\mathcal{C}$—$\Lambda(x_2)$ and $\Lambda(x_3)$ in the example.
- In the example, it suffices to observe the projection $c^\top \delta$
- $\delta$ is the extr direction between $x_2$ and $x_3$.

# A Geometric Perspective on Dataset Sufficiency

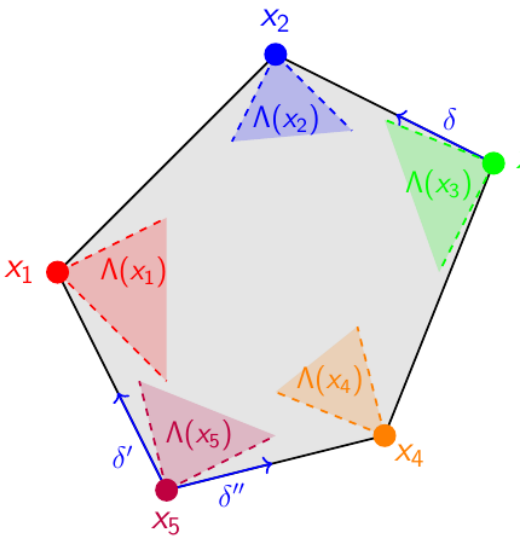


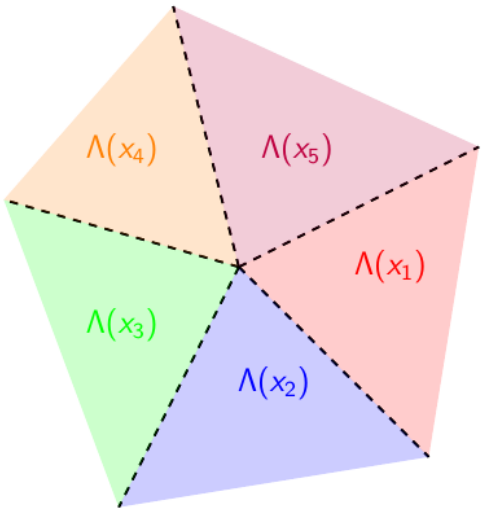



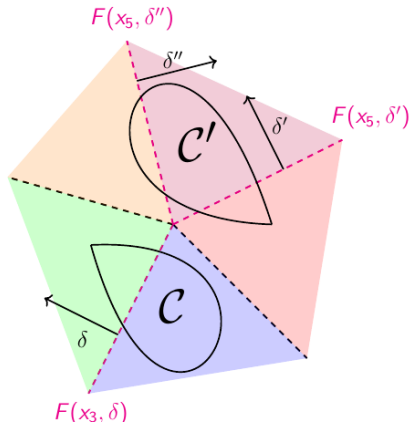




- For the set $\mathcal{C}$, observing the projection $c^\top \delta$ allows discriminating the cones $\Lambda(x_2)$ and $\Lambda(x_3)$.
- For the set $\mathcal{C}'$, observing $c^\top \delta'$ and $c^\top \delta''$ allow discriminating between $\Lambda(x_4)$, $\Lambda(x_5)$, $\Lambda(x_1)$.

# A Geometric Perspective on Dataset Sufficiency



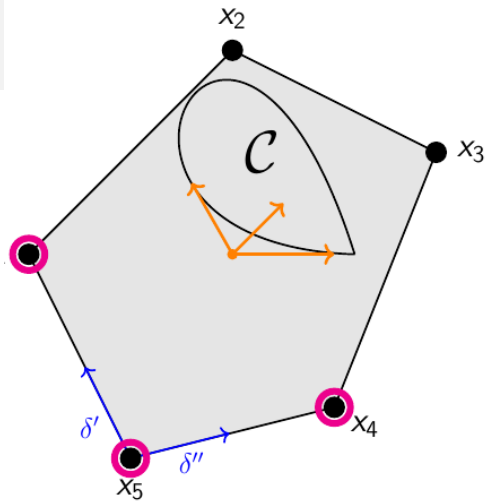




- More generally: each extr direction $\delta$ induces a face of a cone $F(x^\star, \delta)$.
- A sufficient dataset needs to span extr directions $\delta$ of faces $F(x^\star, \delta)$ intersecting the uncertainty set $\mathcal{C}$.

**Definition (Relevant Extreme Directions)**

$$\Delta(\mathcal{X}, \mathcal{C}) = \{\delta \in \mathbb{R}^d \ : \ \exists x^\star \in \mathcal{X}^\angle, \ \delta \in D(x^\star) \text{ and } F(x^\star, \delta) \cap \mathcal{C} \neq \varnothing\}.$$

# Exact Characterization




**Theorem**

*Assume $\mathcal{C}$ convex open.*
*$\mathcal{D}$ is sufficient* $\iff \Delta(\mathcal{X}, \mathcal{C}) \subset \operatorname{span} \mathcal{D}$.

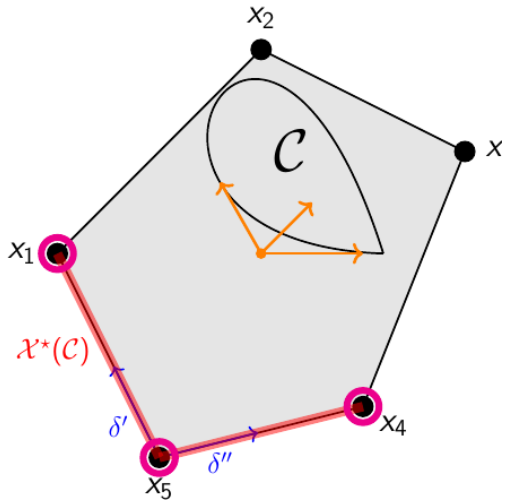
- Provides an exact geometric characterization.
- Downside: Hard to compute! It requires finding all possible optimal solutions that are "neighbors".

$$\Delta(\mathcal{X}, \mathcal{C}) = \{x - x' \ : \ \exists c \in \mathcal{C}, \ x, x' \in \operatorname*{argmin}_{x \in \mathcal{X}} c^\top x \ \cap \ \mathcal{X}^\angle\}$$

# A Tractable Characterization

**Theorem**

*Assume $\mathcal{C}$ convex open.*
*$\mathcal{D}$ is sufficient* $\iff \Delta(\mathcal{X},\mathcal{C}) \subset \operatorname{span} \mathcal{D}$.



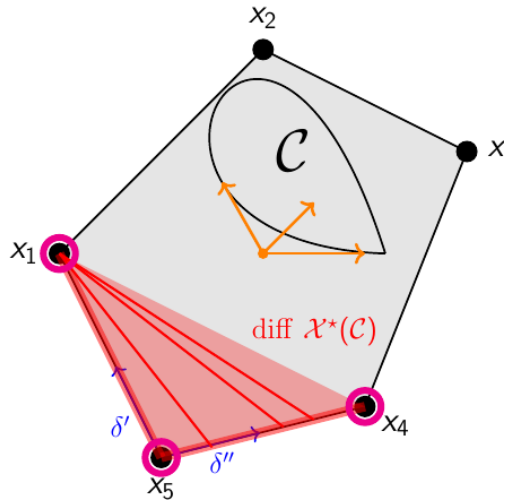

$$\Delta(\mathcal{X},\mathcal{C}) = \{x - x' \ : \ \exists c \in \mathcal{C},\ x, x' \in \operatorname*{argmin}_{x\in\mathcal{X}} c^\top x \cap \mathcal{X}^\angle\}$$

- If we relax the "neighboring condition", we get a set of *reachable solutions* $\mathcal{X}^\star(\mathcal{C}) := \bigcup_{c\in\mathcal{C}} \operatorname{argmin}_{x\in\mathcal{X}} c^\top x$.

# A Tractable Characterization

**Theorem**

*Assume $\mathcal{C}$ convex open.*
*$\mathcal{D}$ is sufficient* $\iff \Delta(\mathcal{X},\mathcal{C}) \subset \operatorname{span} \mathcal{D}$.



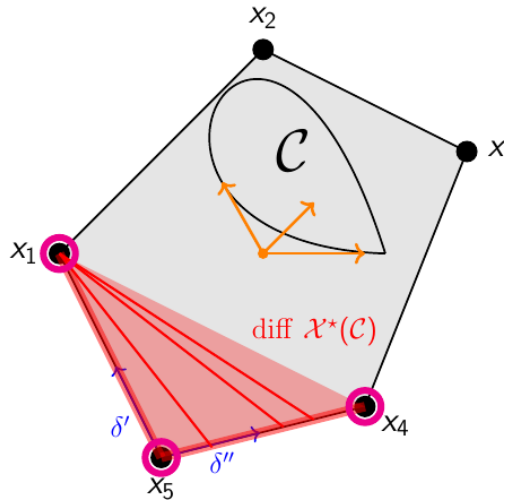

$$\Delta(\mathcal{X},\mathcal{C}) = \{x - x' \ : \ \exists c \in \mathcal{C},\ x, x' \in \operatorname*{argmin}_{x\in\mathcal{X}} c^\top x \cap \mathcal{X}^\angle\}$$

- If we relax the "neighboring condition", we get a set of *reachable solutions* $\mathcal{X}^\star(\mathcal{C}) := \bigcup_{c\in\mathcal{C}} \operatorname{argmin}_{x\in\mathcal{X}} c^\top x$.
- Its associated differences $\operatorname{dir} \mathcal{X}^\star(\mathcal{C}) = \{x - x' \ : \ x, x' \in \mathcal{X}^\star(\mathcal{C})\}$.

# A Tractable Characterization

**Theorem**

*Assume $\mathcal{C}$ convex open.*
*$\mathcal{D}$ is sufficient* $\iff \Delta(\mathcal{X},\mathcal{C}) \subset \operatorname{span} \mathcal{D}$.



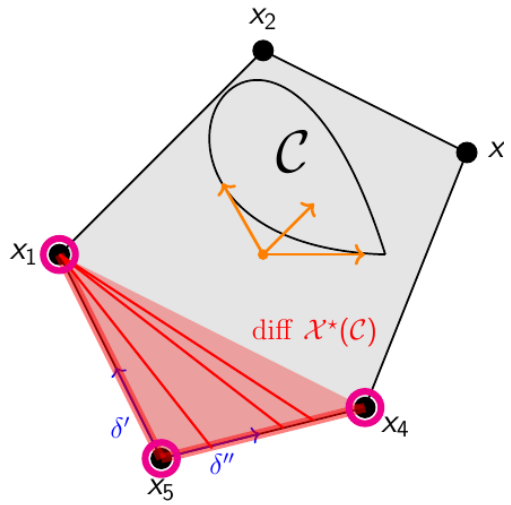

$$\Delta(\mathcal{X},\mathcal{C}) = \{x - x' \ : \ \exists c \in \mathcal{C},\ x, x' \in \operatorname*{argmin}_{x\in\mathcal{X}} c^\top x \cap \mathcal{X}^\angle\}$$

- If we relax the "neighboring condition", we get a set of *reachable solutions* $\mathcal{X}^\star(\mathcal{C}) := \bigcup_{c\in\mathcal{C}} \operatorname{argmin}_{x\in\mathcal{X}} c^\top x$.
- Its associated differences $\operatorname{dir} \mathcal{X}^\star(\mathcal{C}) = \{x - x' \ : \ x, x' \in \mathcal{X}^\star(\mathcal{C})\}$.

**Theorem**

*Assume $\mathcal{C}$ is convex. We have* $\operatorname{span} \Delta(\mathcal{X},\mathcal{C}) = \operatorname{span} \operatorname{dir} \mathcal{X}^\star(\mathcal{C})$.

# A Tractable Characterization




**Theorem**

*Assume $\mathcal{C}$ convex open.*
*$\mathcal{D}$ is sufficient* $\iff \Delta(\mathcal{X}, \mathcal{C}) \subset \operatorname{span} \mathcal{D}$.

**Theorem**

*Assume $\mathcal{C}$ is convex.*
*We have* $\operatorname{span} \Delta(\mathcal{X}, \mathcal{C}) = \operatorname{span} \operatorname{dir} \mathcal{X}^\star(\mathcal{C})$.

Combining both results gives:

**Theorem**

*Assume $\mathcal{C}$ convex open.*
*$\mathcal{D}$ is sufficient* $\iff \operatorname{span} \operatorname{dir} \mathcal{X}^\star(\mathcal{C}) \subset \operatorname{span} \mathcal{D}$.

*Given a task ($\mathcal{X}$) and prior information ($\mathcal{C}$),*
*how to construct the smallest sufficient dataset?*

# A Data Selection Algorithm: Smallest Sufficient Dataset

Recall $\mathcal{X}^\star(\mathcal{C}) := \bigcup_{c\in\mathcal{C}} \operatorname{argmin}_{x\in\mathcal{X}} c^\top x$ and the characterization:

$$\boxed{\mathcal{D} \text{ is sufficient} \iff \operatorname{span\ dir}\, \mathcal{X}^\star(\mathcal{C}) \subset \operatorname{span}\, \mathcal{D}}$$

We can write $\operatorname{span\ diff}\, \mathcal{X}^\star(\mathcal{C}) = \operatorname{span}\, \{x^\star - x_0 : x^\star \in \mathcal{X}^\star(\mathcal{C})\}$ for some fixed $x_0 \in \mathcal{X}^\star(\mathcal{C})$.

1. Start with $\mathcal{D} \leftarrow \varnothing$.
2. *Is there a cost scenario that changes my decision in a way NOT measured by my data?*
   Formally: Check if $\exists\, c \in \mathcal{C}$, $x^\star \in \operatorname{argmin}_{x\in\mathcal{X}} c^\top x$ with $x^\star - x_0 \notin \operatorname{span}\, \mathcal{D}$.
   - If YES, add a measurement of their difference: $\mathcal{D} \leftarrow \mathcal{D} \cup \{x^\star - x_0\}$.
   - If NO, the data $\mathcal{D}$ is sufficient.

# A Data Selection Algorithm: Detecting a Missing Direction

**Key algorithmic step.** Given the current dataset $\mathcal{D}$, check whether $\operatorname{span} \mathcal{D}$ misses a reachable decision direction:

$$\exists c \in \mathcal{C}, \quad x^\star \in \operatorname*{argmin}_{x \in \mathcal{X}} c^\top x \quad \text{with} \quad x^\star - x_0 \notin \operatorname{span} \mathcal{D}.$$

Let $P_{\mathcal{D}^\perp}$ denote projection onto $(\operatorname{span} \mathcal{D})^\perp$. Equivalently, test whether the norm problem has a positive optimum:

$$\sup_{c, x^\star} \left\| P_{\mathcal{D}^\perp}(x^\star - x_0) \right\| \quad \text{s.t. } c \in \mathcal{C},\ x^\star \in \operatorname*{argmin}_{x \in \mathcal{X}} c^\top x.$$

Maximizing a convex objective with bilinear and bilevel constraints.

- **Randomization trick.** Draw $\alpha \sim \mathcal{N}(0, I_d)$ once. For any fixed $v \neq 0$, $\Pr\{\alpha^\top v \neq 0\} = 1$.
- Thus, with probability one, it suffices to solve the two linearized search problems

  $$\sup_{c, x^\star} \alpha^\top P_{\mathcal{D}^\perp}(x^\star - x_0) \quad \text{and} \quad \inf_{c, x^\star} \alpha^\top P_{\mathcal{D}^\perp}(x^\star - x_0) \quad \text{s.t. } c \in \mathcal{C},\ x^\star \in \operatorname*{argmin}_{x \in \mathcal{X}} c^\top x.$$

- Both signs needed: $\alpha^\top v$ may be positive or negative. Since $x_0 \in \mathcal{X}^\star(\mathcal{C})$, value 0 is feasible; append the returned direction if the sup is $> 0$ or the inf is $< 0$.

# A Data Selection Algorithm: MIP Implementation

In either linearized search problem, the key constraint is $x^\star \in \operatorname{argmin}_{x \in \mathcal{X}} c^\top x$. For $\mathcal{X} = \{x \geq 0 : Ax = b\}$, LP optimality is equivalent to primal-dual feasibility plus complementary slackness:

$$x^\star \geq 0, \quad s \geq 0, \quad Ax^\star = b, \quad A^\top \lambda + s = c, \quad x_i^\star s_i = 0 \quad \forall i.$$

- Together with $c \in \mathcal{C}$, this gives a linear objective over $(c, x^\star, \lambda, s)$, plus complementarity.
- Complementarity is encoded with binary variables, e.g., $0 \leq x_i^\star \leq M_i z_i$ and $0 \leq s_i \leq \bar{M}_i(1 - z_i)$.
- When $\mathcal{C}$ is polyhedral and valid bounds are available, each search call is a mixed-integer program with $O(d + \#\text{constr}(\mathcal{X}))$ variables and $O(d + \#\text{constr}(\mathcal{X}) + \#\text{constr}(\mathcal{C}))$ constraints.

**Takeaway.** This gives an implementable MIP routine for the data-selection algorithm. It also motivates the hardness question: can we prove NP/coNP-hardness results for finding SDDs? We will address this in the second half of the talk.

# Applications

# Application: Shortest Path Problems

- The task is to navigate from the origin to destination in a neighborhood of Cambridge, MA.
- Assume we have an estimation $c_0 > 0$ of the edges travel time.

- Travel times (costs) $c$ are uncertain and vary in: $(1-\varepsilon)c_0 \leq c \leq (1+\varepsilon)c_0$, where $\varepsilon \in [0,1]$.
- Extreme directions $\delta$ in this problem are cycles.
- Applications: Railroad planning, navigation in altered/damaged networks...

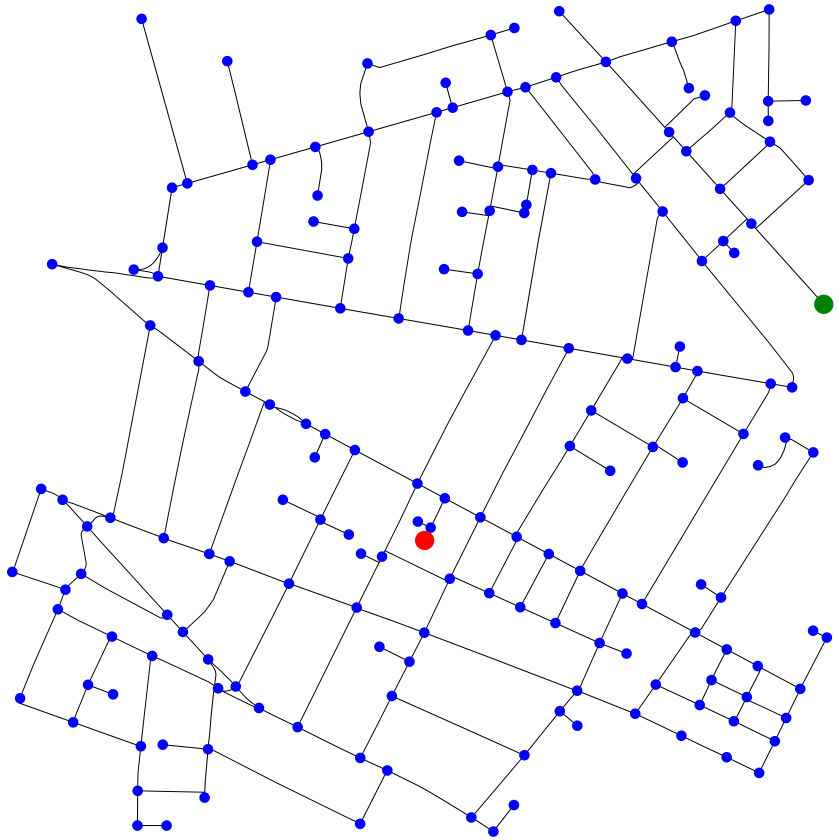


Figure: Neighborhood in Cambridge

# Application: Shortest Path Problems

- $c \in \mathcal{C}_\varepsilon := \{c \in \mathbb{R}^d,\ (1-\varepsilon)c_0 \leq c \leq (1+\varepsilon)c_0\}$.
- When $\varepsilon = 0$, $c = c_0$.

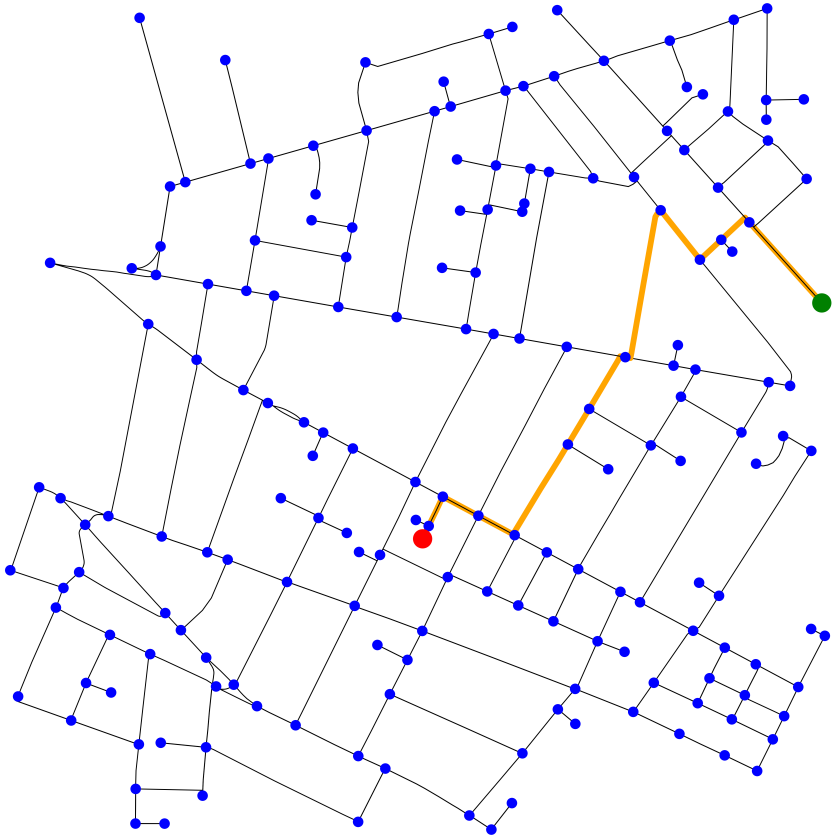


Figure: Shortest path for $c = c_0$.

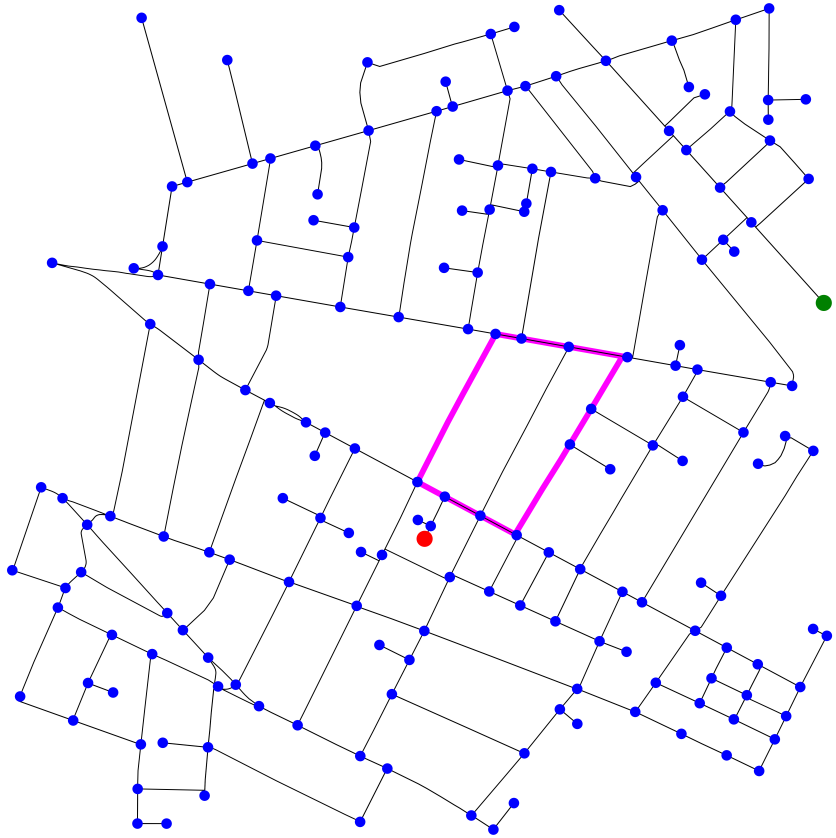


Figure: Edges to measure (in magenta) for $\varepsilon = 7\%$.

# Data Needed as a Function of Uncertainty

- $c \in \mathcal{C}_\varepsilon := \{c \in \mathbb{R}^d,\ (1-\varepsilon)c_0 \leq c \leq (1+\varepsilon)c_0\}$.
- More uncertainty ($\varepsilon \uparrow$) $\implies$ more data needed.

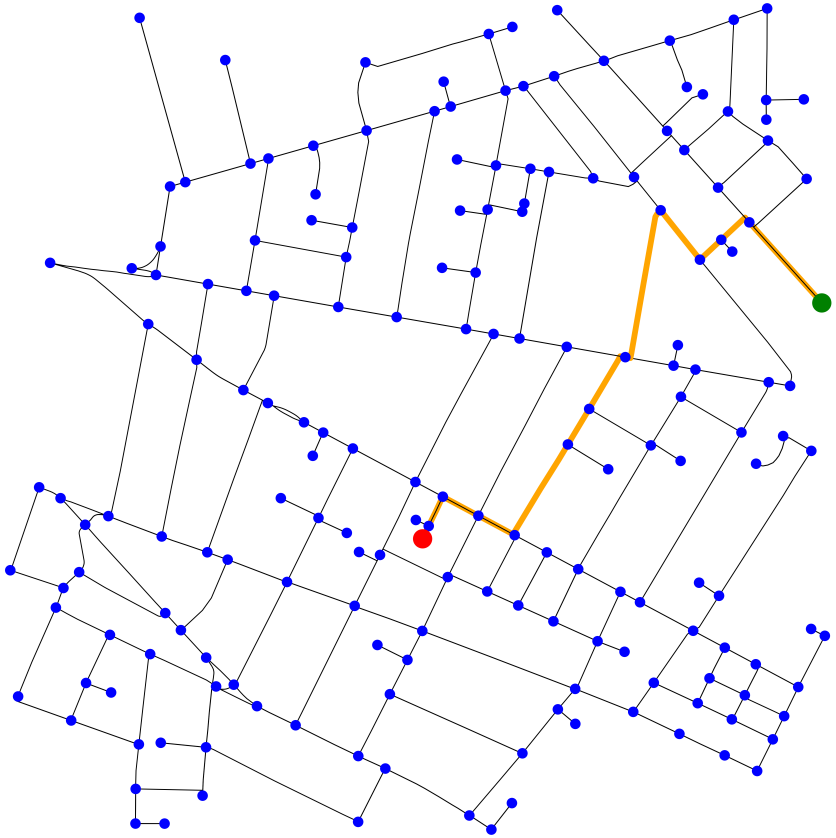


Figure: Shortest path for $c = c_0$.

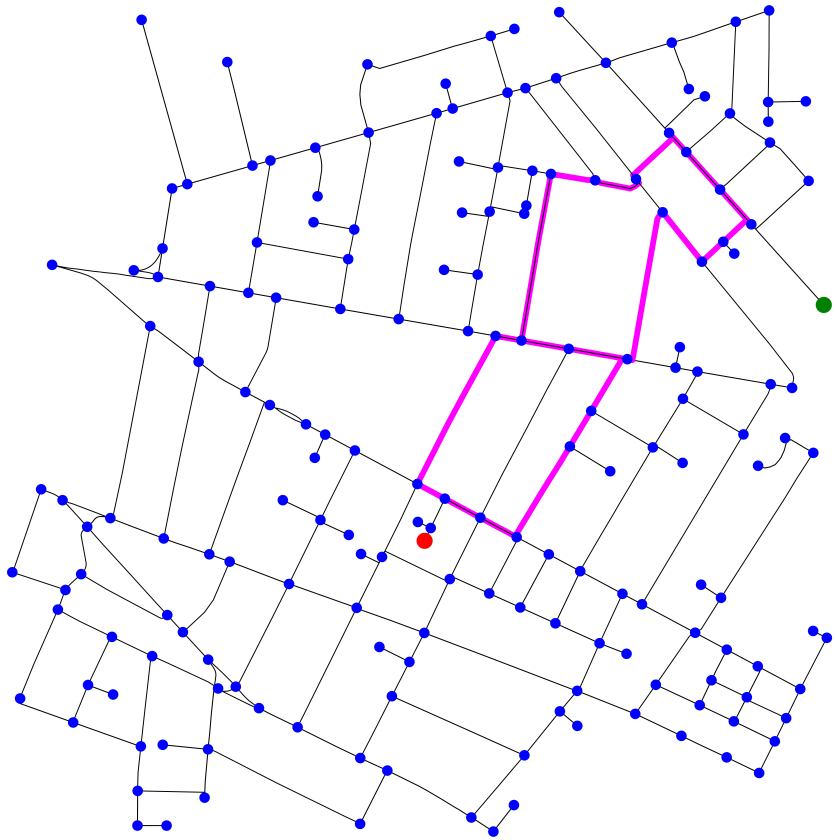


Figure: Edges to measure (in magenta) for $\varepsilon = 7.7\%$.

# Data Needed as a Function of Uncertainty

- $c \in \mathcal{C}_\varepsilon := \{c \in \mathbb{R}^d,\ (1-\varepsilon)c_0 \leq c \leq (1+\varepsilon)c_0\}$.
- More uncertainty ($\varepsilon \uparrow$) $\implies$ more data needed.

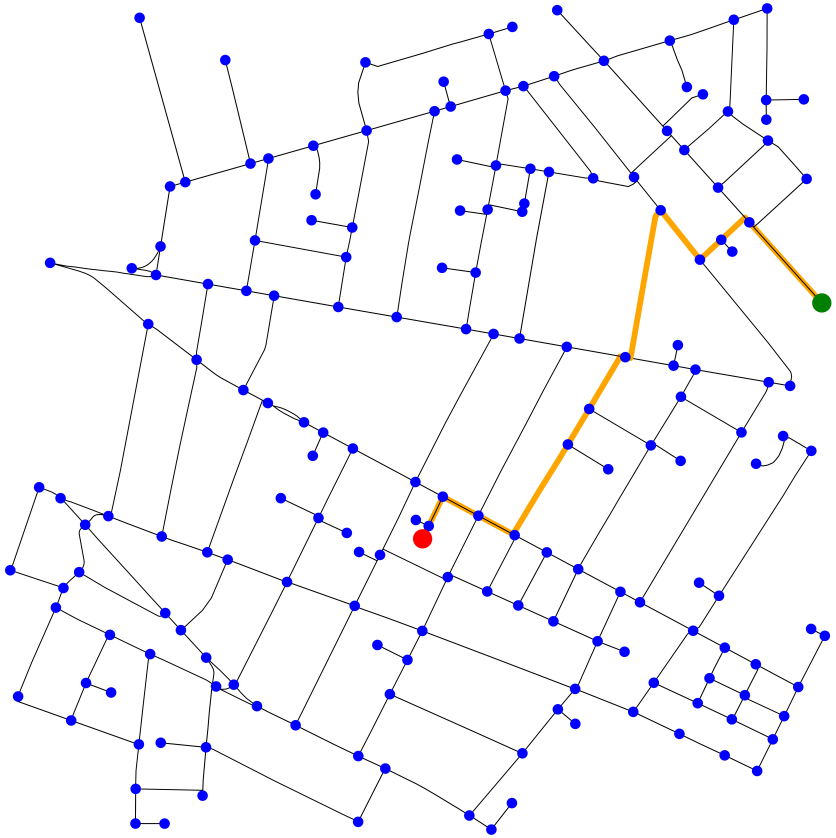


Figure: Shortest path for $c = c_0$.

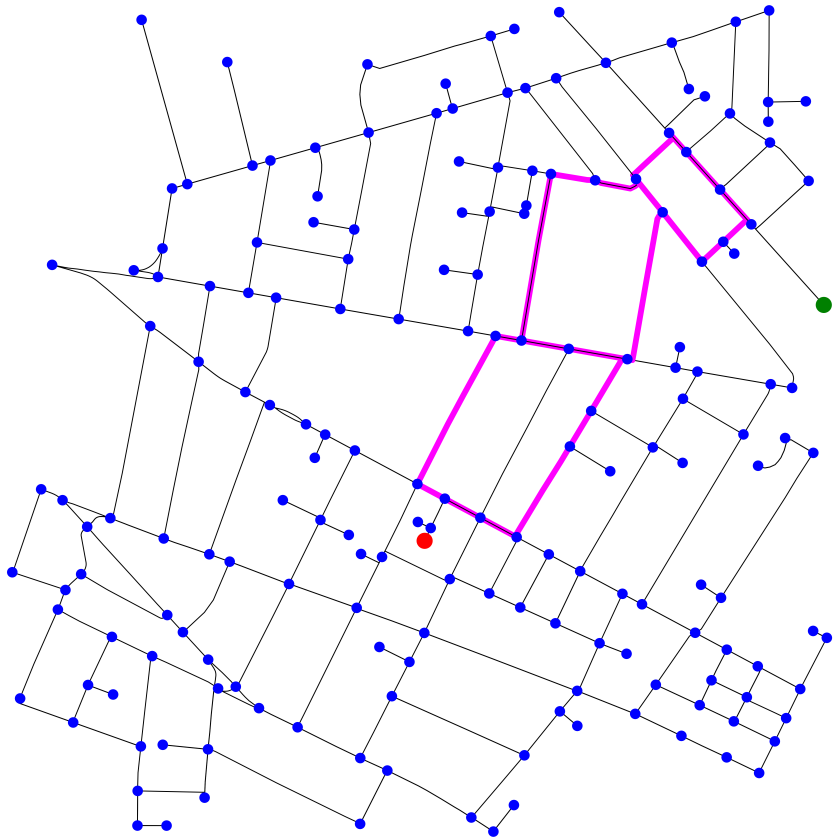


Figure: Edges to measure (in magenta) for $\varepsilon = 10\%$.

# Data Needed as a Function of Uncertainty

- More uncertainty ($\varepsilon \uparrow$) $\Longrightarrow$ more data needed.

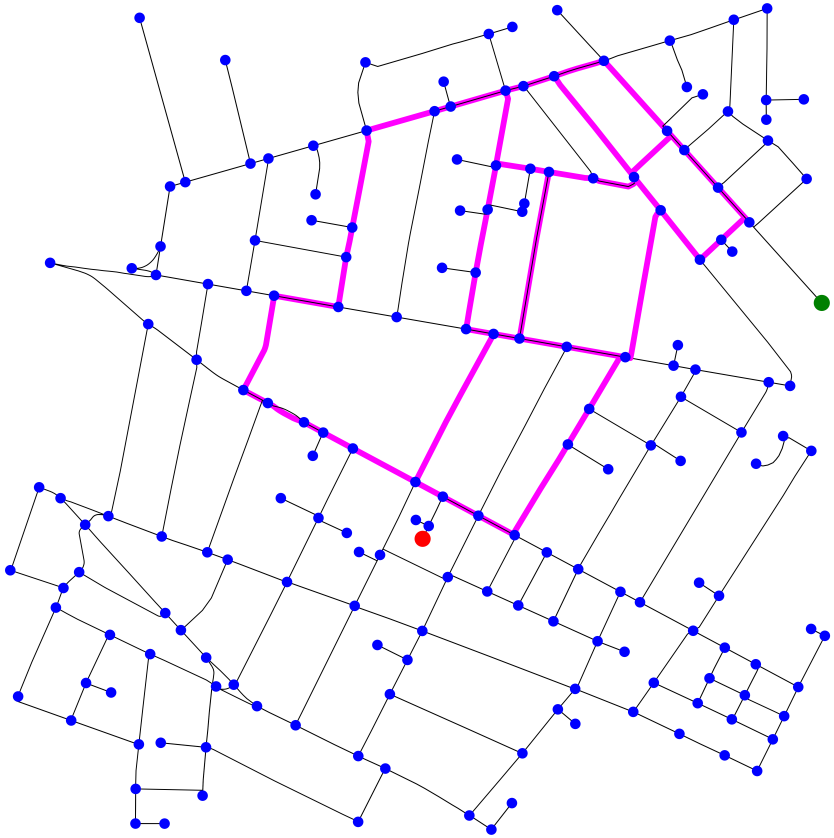


Figure: Edges to measure (in magenta) for $\varepsilon = 30\%$.

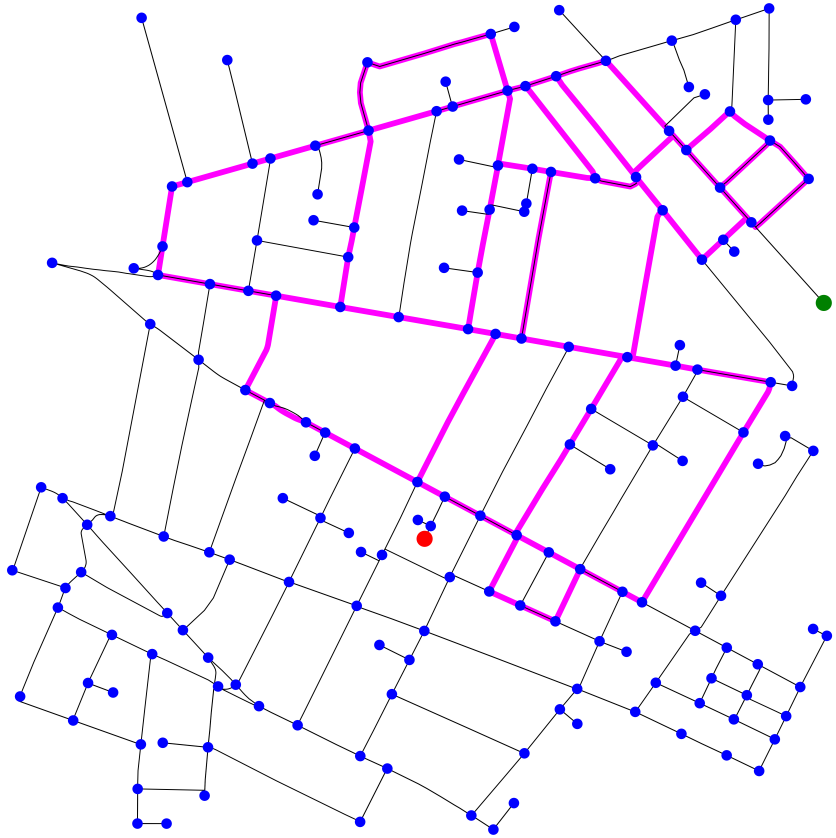


Figure: Edges to measure (in magenta) for $\varepsilon = 60\%$.

# Data Needed as a Function of Uncertainty

- When $\varepsilon = 99\%$ ($\sim \mathcal{C} = \mathbb{R}^d_{>0}$), every edge that can induce some optimal solution needs to be measured.

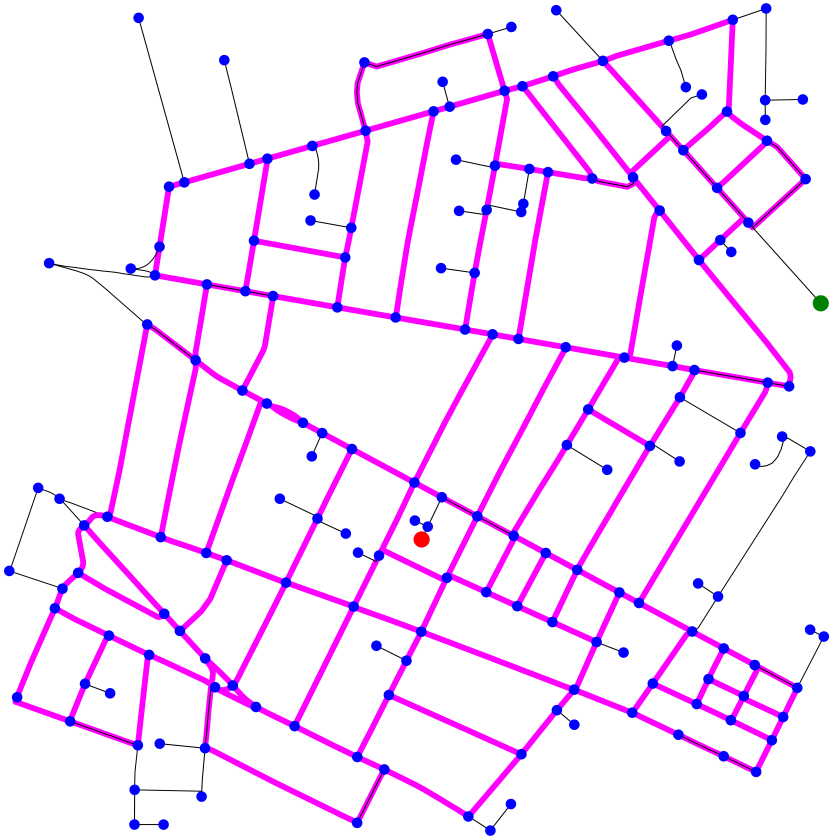


Figure: Edges to measure (in magenta) for $\varepsilon = 99\%$.

# Hardness of Constructing Minimum-Size SDDs

# Finding Minimum-Size Global SDDs Is Hard

- We focus on the polyhedral setting: $\mathcal{C}$ is a polyhedron.
- Recall the exact characterization for open convex $\mathcal{C}$:

$$\mathcal{D} \text{ is sufficient} \iff \operatorname{dir}(\mathcal{X}^\star(\mathcal{C})) \subseteq \operatorname{span} \mathcal{D}, \qquad d^\star = \dim\big(\operatorname{dir}(\mathcal{X}^\star(\mathcal{C}))\big).$$

- **Worst-case barrier**: Computing the dimension $d^\star$ is NP-hard [Ye et al. 26].

Proof Idea: Reduction from PISPP-W+ which is NP-complete [Ley–Merkert 25].

**PISPP-W+** = *partial inverse shortest path with weight increases*: given a DAG, lengths $d$, modification costs $\kappa$, budget $B$, and one arc $r$, decide whether

$$\exists w \geq 0, \quad \kappa^\top w \leq B, \quad \text{s.t. some shortest } s\text{–}t \text{ path under } d + w \text{ uses } r.$$

- **The PISPP-W+ answer becomes a dimension comparison.** Check if there is a cost modification for which a shortest path includes arc $r$, adding a new direction – compare $\dim \operatorname{dir}(\mathcal{X}^\star(\mathcal{C}))$ with $\dim \operatorname{dir}(\mathcal{X}_r^\star(\mathcal{C}))$.
- Deciding PISPP-W+ can be done by two queries to a "dimension oracle", showing computing $d^*$ is NP-hard.

# Beyond Worst-Case Analysis: Learning from Distributional Instances

# Beyond Worst-Case: Learning from Distributional Data

- The same LP structure is solved repeatedly, while costs vary across instances:

$$c \sim P_c, \qquad c_1, \ldots, c_n \overset{\text{i.i.d.}}{\sim} P_c, \qquad \operatorname{supp}(P_c) \subseteq \mathcal{C}.$$

- Goal: learn a small dataset $\widehat{\mathcal{D}}$ that is "sufficient" for a fresh draw with high probability, rather than for every worst-case $c \in \mathcal{C}$.
- **Traffic example.** The network is fixed and each road-segment travel time lies in a known range. Realized travel times change with time, weather, incidents, and congestion. Historical samples may reveal a few critical road-segment observations that determine the optimal decision for most future instances.
- This is a beyond-worst-case, distributional viewpoint: hard regions of $\mathcal{C}$ may be negligible under $P_c$. [Roughgarden 19/21; Balcan 21]

# Pointwise SDD vs. Global SDD

For a query dataset $\mathcal{D}$, define

$$\underbrace{s_{\mathcal{D}}(c) := (q^\top c)_{q\in\mathcal{D}}}_{\text{measurement observation}}, \qquad \underbrace{F_{\mathcal{D}}(c) := \{c' \in \mathcal{C} : \; s_{\mathcal{D}}(c') = s_{\mathcal{D}}(c)\}}_{\text{data-consistent fiber}}.$$

**Pointwise SDD at $c$.** $\mathcal{D}$ is pointwise sufficient at $c$ if

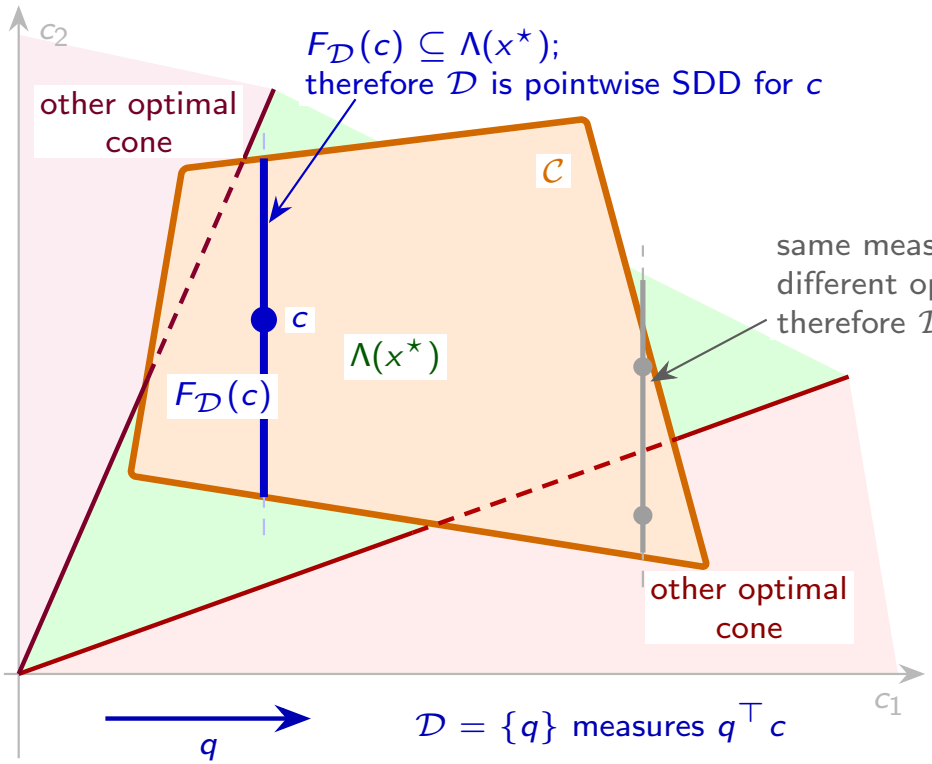
$$\exists x^\star \in \mathcal{X}^\star(c) \quad \text{s.t.} \quad F_{\mathcal{D}}(c) \subseteq \Lambda(x^\star).$$

Equivalently, $x^\star \in \mathcal{X}^\star(c')$ for every cost $c' \in F_{\mathcal{D}}(c)$.

- A **global SDD** must certify every measurement fiber inside $\mathcal{C}$: for each $c' \in \mathcal{C}$, there exists $x^\star(c') \in \mathcal{X}^\star(c')$ with $F_{\mathcal{D}}(c') \subseteq \Lambda(x^\star(c'))$.
- A **pointwise SDD at** $c$ only needs to certify the realized fiber $F_{\mathcal{D}}(c)$. For this realized $c$, the measurements $s_{\mathcal{D}}(c)$ are enough to find and recover one optimal solution $x^\star \in \mathcal{X}^\star(c)$.

# Pointwise SDD vs. Global SDD

$$\underbrace{s_{\mathcal{D}}(c) := (q^\top c)_{q\in\mathcal{D}}}_{\text{measurement observation}}, \qquad \underbrace{F_{\mathcal{D}}(c) := \{c' \in \mathcal{C} : \; s_{\mathcal{D}}(c') = s_{\mathcal{D}}(c)\}}_{\text{data-consistent fiber}}.$$




In the figure, $\mathcal{D} = \{q\}$ is **not** a global SDD for $\mathcal{C}$.

But $\mathcal{D} = \{q\}$ is **pointwise sufficient at** $c$, because

$$F_{\mathcal{D}}(c) \subseteq \Lambda(x^\star).$$

# Pointwise Verification: Hardness and a Tractable Regime

- Verifying pointwise sufficiency at $c$ is equivalent to the cone-containment test

$$\exists x^\star \in \mathcal{X}^\star(c) \quad \text{s.t.} \quad F_{\mathcal{D}}(c) \subseteq \Lambda(x^\star). \tag{CC}$$

- In general, this verification problem is coNP-hard. [Ye et al. 26]
- **Reason**: $H$-in-$V$ polytope containment reduces to the test (CC); this containment problem is coNP-hard.† [Freund–Orlin 85]
- Under LP nondegeneracy, an optimal basis $B$ gives the simple $H$-description

$$\Lambda(B) = \{c' : (c')^\top \delta(B, j) \geq 0 \ \ \forall j \in N\},$$

  where $\delta(B, j)$ is the edge direction obtained by increasing the nonbasic variable $j$ from basis $B$.
- With nondegeneracy, (CC) can be checked by polynomially many convex programs.

†An $H$-polytope is a bounded intersection of finitely many halfspaces, e.g., $P = \{z : Hz \leq h\}$. A $V$ polytope is a convex hull of finite set of points.

# Constructing Pointwise SDDs

**Algorithm 1: Cutting planes for pointwise SDDs.** **Input:** LP data, prior $\mathcal{C}$, realized cost $c$, initial dataset $\mathcal{D}_{\text{init}}$. **Initialize:** $k = 0$, $\mathcal{D}_0 \leftarrow \mathcal{D}_{\text{init}}$, observe $s_{\mathcal{D}_0}(c)$, and solve once at $c$ to obtain an optimal basis $B$ and decision $x(B)$.

1. **Form the current fiber.** $F_k := \{c' \in \mathcal{C} : \ q^\top c' = q^\top c \ \forall q \in \mathcal{D}_k\}$.
2. **Test containment in the fixed optimality cone.** Under nondegeneracy, the cone of the initialized basis is
$$\Lambda(B) = \{c' : (c')^\top \delta(B, j) \geq 0 \ \forall j \in N\}.$$
For each $j \in N$, solve the face-intersection subproblem
$$m_j := \min_{c' \in F_k} (c')^\top \delta(B, j). \qquad (\text{FI}_j)$$
3. **Stop if certified.** If $m_j \geq 0$ for all $j \in N$, then $F_k \subseteq \Lambda(B)$; return $\mathcal{D}_k$ and $x(B)$.
4. **Otherwise cut and repeat.** Choose a minimizer $c^{\text{out}} \in F_k$ from a violated test. Let $j^\star$ be the first facet of $\Lambda(B)$ crossed by $[c, c^{\text{out}}]$. Query $q_{k+1} := \delta(B, j^\star)$, observe $q_{k+1}^\top c$, update $\mathcal{D}_{k+1} := \mathcal{D}_k \cup \{q_{k+1}\}$, set $k \leftarrow k + 1$, and return to Step 1.

# Polynomial Implementation and Solution Recovery

**Polynomial implementation.**

- Under LP nondegeneracy, solve one initial LP over $\mathcal{X}$ at the realized cost $c$.
- If $\mathcal{C}$ is an $H$-polytope or ellipsoid, each subsequent iteration only performs $d - m$ face-intersection checks: these checks are LPs for $H$-polytopes and closed form for ellipsoids.

**Termination and complexity.**

- Each cut adds a linearly independent direction in $W_\star := \mathrm{dir}(\mathcal{X}^\star(\mathcal{C}))$.
- Since $\dim(W_\star) = d^\star$, Algorithm 1 adds at most $d^\star$ queries and uses at most $(d^\star + 1)(d - m)$ face-intersection checks.

# Learning SDDs over Samples

Define the 0–1 sufficiency loss and risk:

$$\ell(\mathcal{D}, c) := 1\{\mathcal{D} \text{ fails at } c\}, \qquad R(\mathcal{D}) := \Pr_{c \sim P_c}[\ell(\mathcal{D}, c) = 1].$$

**Algorithm 2: Learning sufficient decision datasets over samples.**
**Input:** prior $\mathcal{C}$, LP data, samples $c_1, \ldots, c_n$.
**Initialize:** $\mathcal{D}_0 = \emptyset$, hard set $T = \emptyset$.

**For** $i = 1, \ldots, n$**:**

1. Run Algorithm 1 on $c_i$, warm-started with $\mathcal{D}_{i-1}$.
2. Let the returned dataset be $\mathcal{D}_i$.
3. If $\mathcal{D}_i \neq \mathcal{D}_{i-1}$, mark $i$ hard: $T \leftarrow T \cup \{i\}$.

**Return:** $\mathcal{D}_n$ and the hard subsequence indexed by $T$.

- Samples with $\mathcal{D}_i = \mathcal{D}_{i-1}$ are already certified by the current dataset.
- **Hard samples are exactly the elements of** $T$: only hard samples add new decision-relevant directions.

# Stable Sample-Compression Scheme

Algorithm 2 induces a realizable stable sample-compression scheme.
[Hanneke–Kontorovich 21]

## Three key properties

- **Realizability:** $\ell(\mathcal{D}_n, c_i) = 0$ for every training sample $i = 1, \ldots, n$.
- **Stability:** $\mathcal{D}_n$ is fully determined by the hard subsequence $(c_i)_{i \in T}$; deleting non-hard samples changes nothing.
- **Small compression size:** $|T| \leq |\mathcal{D}_n| \leq d^\star$.

- $|T| \leq |\mathcal{D}_n|$ holds because each hard sample adds at least one query direction. $|\mathcal{D}_n| \leq d^\star$ holds because Algorithm 1 only adds independent directions in $W_\star$.
- This stable compression structure leads directly to the distribution-free learning certificate next.

# Distribution-Free Learning Certificate

**Theorem (Certificate via stable compression)**

*For any $\delta \in (0,1)$, with probability at least $1-\delta$ over $c_1, \ldots, c_n$, Algorithm 2 returns $(\mathcal{D}_n, T)$ satisfying*

$$R(\mathcal{D}_n) \le \frac{4}{n}\left(6|T| + \ln\frac{e}{\delta}\right) \le \frac{4}{n}\left(6d^\star + \ln\frac{e}{\delta}\right).$$

*Proof Intuition:*

- Without realizability, $\widetilde{O}(1/\sqrt{n})$ bound is the standard deviation scale.
- In a realizable stable compression scheme, the learner returns $h_S$ with zero training loss, and $h_S$ is determined by a small compression set $T(S) \subseteq S$.

$$|T(S)| \le k, \qquad \ell(h_S, z_i) = 0 \quad (i \le n).$$

- **Compression intuition.** Draw $S^+ = (z_1, \ldots, z_{n+1})$ and hold out a uniformly random index $J$. Run compression on $S^+$. If $J \notin T(S^+)$, stability says removing $z_J$ does not change the learned hypothesis, and the full run has zero loss on $z_J$. Hence

$$\Pr\{\text{test fails}\} \lesssim \frac{|T(S^+)|}{n+1} \le \frac{k}{n+1}.$$

For our SDD learner, $k = d^\star$, so $R(\mathcal{D}_n) = \widetilde{O}(d^\star/n)$.

[Hannaka, Kontorovich 21]

# Application: Contextual Linear Optimization

- Data are drawn as $(\xi, c) \sim P$, where $\xi \in \mathbb{R}^p$ is context and $c \in \mathcal{C} \subseteq \mathbb{R}^d$ is the downstream LP cost.
- A predictor $f(\xi)$ induces the plug-in decision

$$x^\star(f(\xi)) \in \arg\min_{x \in \mathcal{X}} f(\xi)^\top x.$$

- The **Smart Predict-then-Optimize (SPO)** loss measures the extra cost of using the predicted cost $\hat{c}$:

$$\ell_{\mathrm{SPO}}(\hat{c}, c) := c^\top x^\star(\hat{c}) - c^\top x^\star(c) \geq 0.$$

- SPO is decision-aware: prediction errors matter only through the downstream decision.
- If decisions only depend on $W_\star$, train a coordinate predictor in $\mathbb{R}^{d^\star}$ and lift it back to the original cost space. [Elmachtoub–Grigas 22; El Balghiti et al. 23]

# Improved SPO Generalization Bound

## Stage I: representation

Run Algorithm 2 over cost samples to learn a decision-relevant subspace

$$\widehat{W} = \operatorname{span} \widehat{\mathcal{D}} \approx W_\star.$$

Stable compression gives a pointwise-sufficiency failure certificate of order $O(d^\star/n)$.

## Stage II: prediction

Train an SPO predictor in the learned coordinates, then lift the prediction to $\mathbb{R}^d$ before solving the LP.

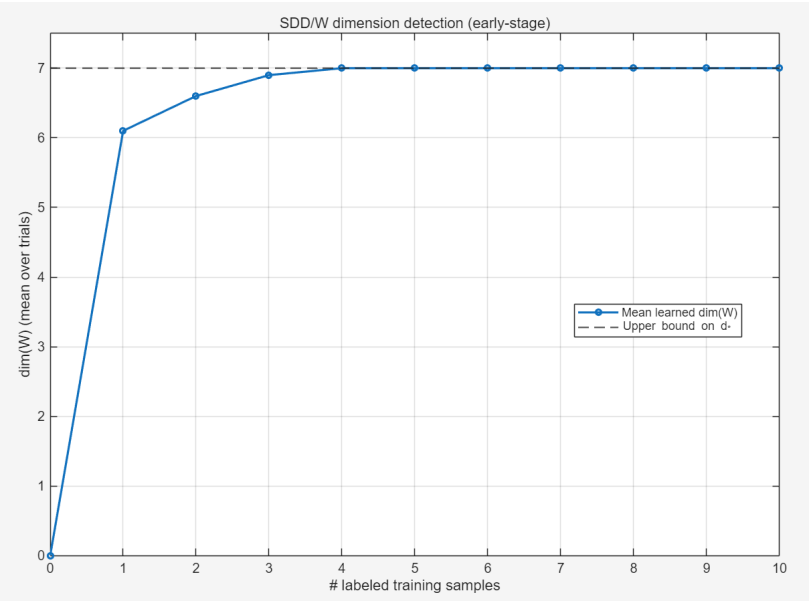
$$g(\xi) \in \mathbb{R}^{d^\star} \quad \mapsto \quad \hat{c}(\xi) \in \mathbb{R}^d.$$

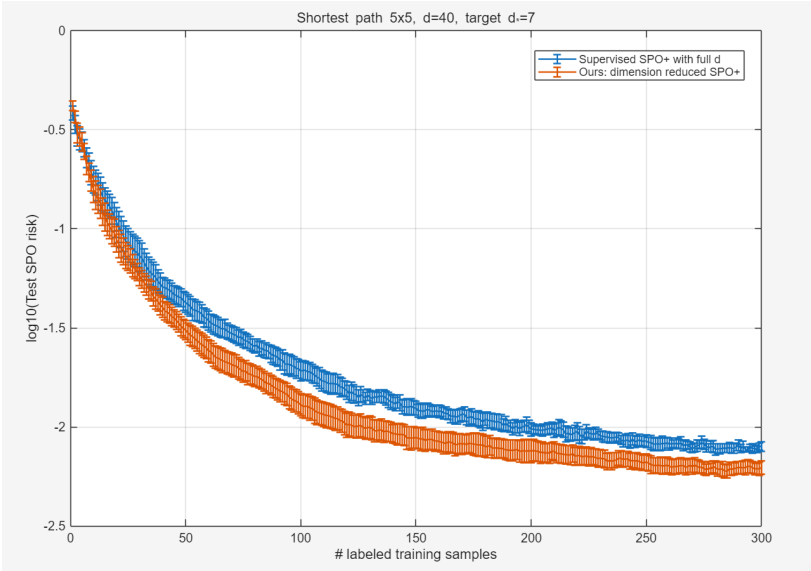
## Theorem (Compressed SPO generalization [El Balghiti et al. 23; Ye 26])

*For linear coordinate predictors, the SPO generalization term scales as*

$$\widetilde{O}\left(\sqrt{\frac{d^\star p}{n}}\right) \quad \textit{instead of} \quad \widetilde{O}\left(\sqrt{\frac{dp}{n}}\right).$$

# Numerical Illustration: Shortest Path




- 5-by-5 shortest-path grid: ambient cost dimension $d = 40$, context dimension $p = 5$, intrinsic dimension $d^\star = 7$.
- Stage I learns a low-dimensional decision-relevant subspace.
- Stage II trains SPO predictors in that subspace, improving out-of-sample SPO risk relative to full-dimensional training.

# Conclusions

- We presented a new framework to study sufficiency of datasets for recovering an optimal solution of linear programming problems.
- **Key idea:** *From "Learn Everything" to "Learn What Matters:"* — Geometric analysis reveals that prior information and task structure reduce data requirements while preserving optimality.

**Many future directions:**

- Quadratic, convex, mixed integer programs.
- Uncertainty in the constraints.
- Approximate optimality, noisy observations.
- Constraints on measurements.
- Adaptive decision information.

# Comparison to Data-Driven Algorithm Design

- **Data-driven algorithm design** is a line of work led by Nina Balcan and her group. It starts from a parameterized algorithm family $\{A_\theta\}$ and training instances drawn from an unknown application-specific distribution. [COLT 17; ICML 18; FOCS 18; STOC 21/JACM 24; NeurIPS 24]
- It chooses a parameter $\widehat{\theta}$ with strong average training performance, and proves generalization guarantees for the future performance of the learned algorithm $A_{\widehat{\theta}}$.
- **Data-driven projection for LPs** learns a low-dimensional projection and proves generalization bounds for the objective-value gap of the learned projection; the empirical projection-learning problem is generally nonconvex. [Sakaue–Oki 24]
- These works control *performance gaps* for the learned object ($A_{\widehat{\theta}}$ or a projection), typically at the uniform-convergence scale $\widetilde{O}(1/\sqrt{n})$.

**Our contrast.** We learn a measurement set $\mathcal{D}$, not a parameter setting or a projection matrix. The loss is exact decision-recovery failure: $\ell(\mathcal{D}, c) = 1\{\mathcal{D} \text{ fails at } c\}$.

Algorithm 2 constructs $\mathcal{D}$ in polynomial time under our tractability conditions, and realizable stable compression gives $R(\mathcal{D}_n) = \widetilde{O}(d^\star/n)$.