MIT MANAGEMENT SLOAN SCHOOL

# BeaverGPT: A Generative AI Tool to Navigate Educational Content


Alexandre Jacquillat & Chara Podimata

Michel Leroy, Tommaso Serafin


03/13/2025

# Project overview

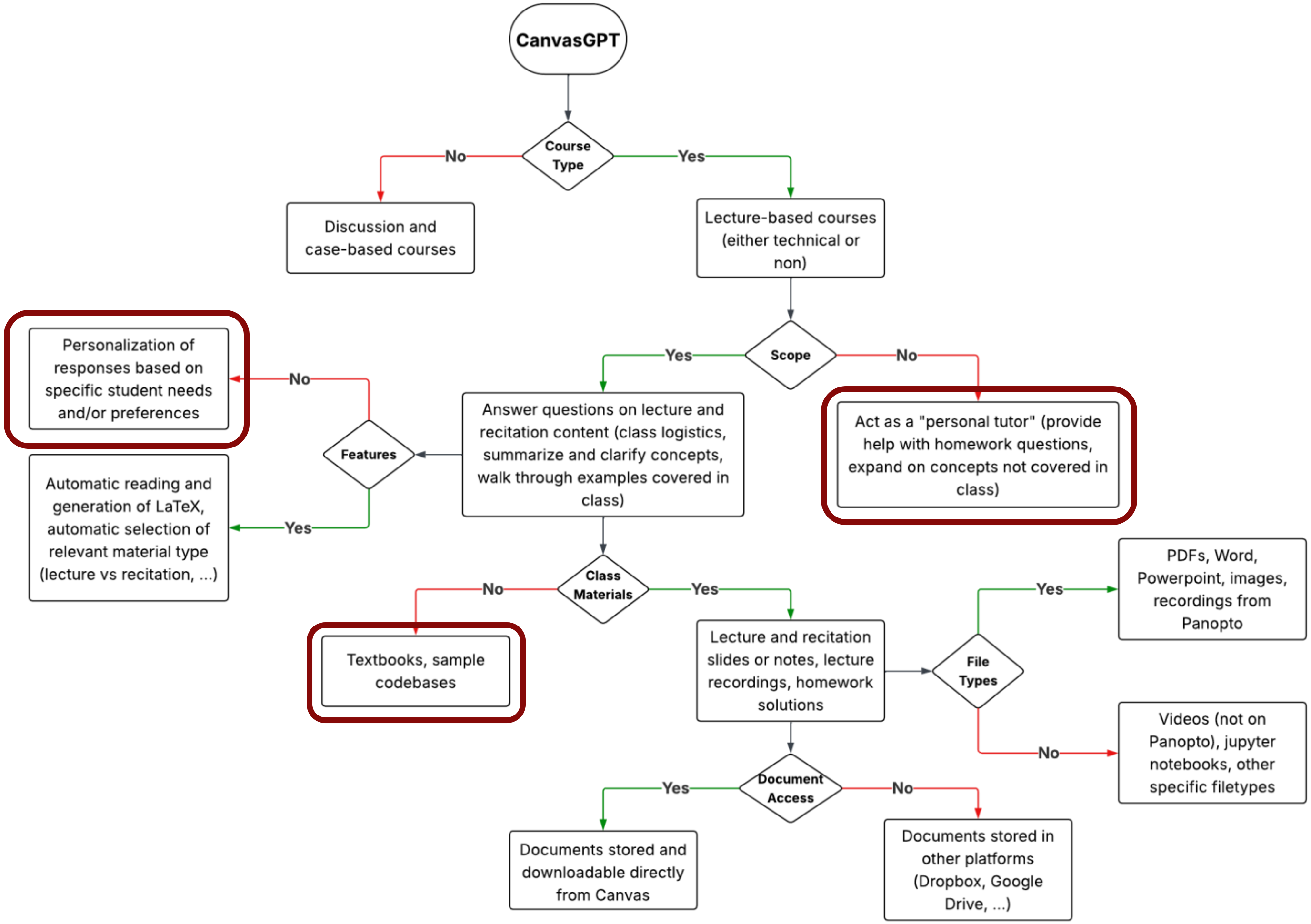
# Project Context

- **Growing Use of AI in Learning**: Students increasingly rely on large language models (LLMs) like ChatGPT to assist with coursework, answer questions, and clarify complex topics.
- **Challenges of Accuracy and Reliability**: While LLMs provide quick responses, they can generate incorrect or misleading information (hallucinations), requiring careful validation.
- **Need for Course-Specific Adaptation**: Generic LLMs may not fully understand course materials, technical language (e.g., LaTeX, programming), or specific instructor expectations.
- **Bridging Gaps in Learning**: LLMs can serve as tutors, helping students outside class hours by providing explanations, summarizing content, and reinforcing key concepts.
- **Ethical & Academic Integrity Issues**: The rise of AI tools raises concerns about plagiarism, unauthorized assistance, and maintaining fairness in academic assessments.
- **Need for Faculty Oversight**: Educators are exploring ways to integrate AI while ensuring students use it responsibly, emphasizing critical thinking and AI literacy.

# Project Scope

| CanvasGPT | TutorGPT |
|---|---|
| Course-specific Virtual TA | Virtual Tutor |
| Flexible and Course-agnostic | Topic and Material specific |
| Robust at Scale | Focus on Student Interactions |
| Integrated with Canvas | Standalone Platform |

# CanvasGPT Project Boundaries



# Knowledge Sources

## CanvasGPT

**Lecture & Recitation Slides**

**Lecture Recordings**

**Homeworks**

**Images, Diagrams & Plots**

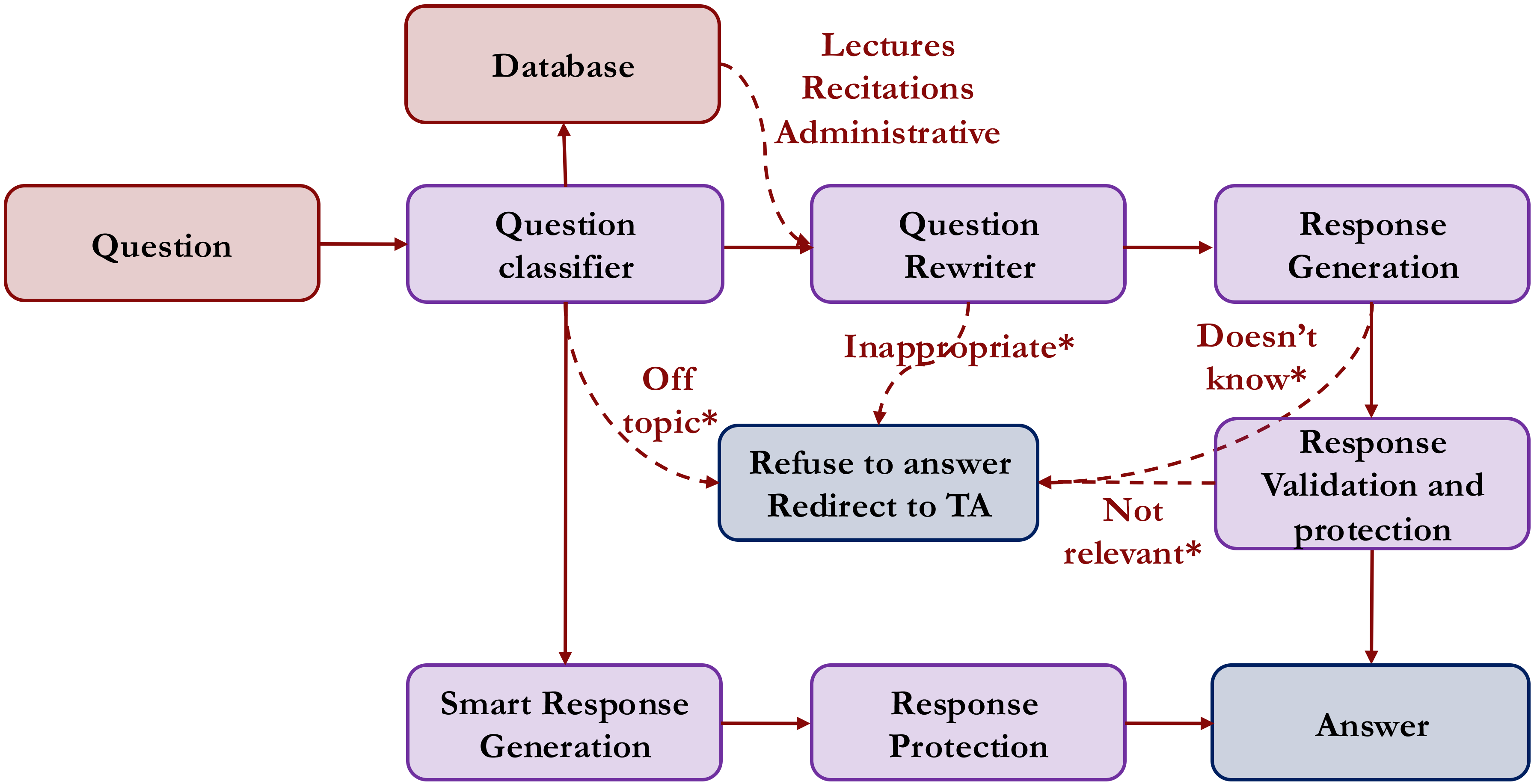
## TutorGPT

**Selected Textbook**

**Additional Selected Materials**

The **pre-processing**, **chucking** and **embedding** steps are similar and can be easily combined in a **single pipeline**

# Generation pipeline

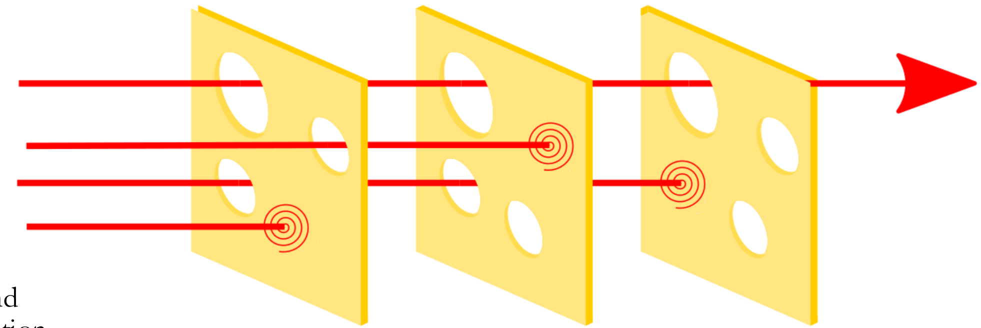





* common failure mode

# Robustness

## Swiss cheese model



Prompt Attack?
Off-Topic?

- LLM evaluation

From Homework?
Malicious?

- Cosine similarity and LLM evaluation

Addressed by known material?

- Cosine similarity evaluation

Aligns with our guidelines?

- LLM evaluation

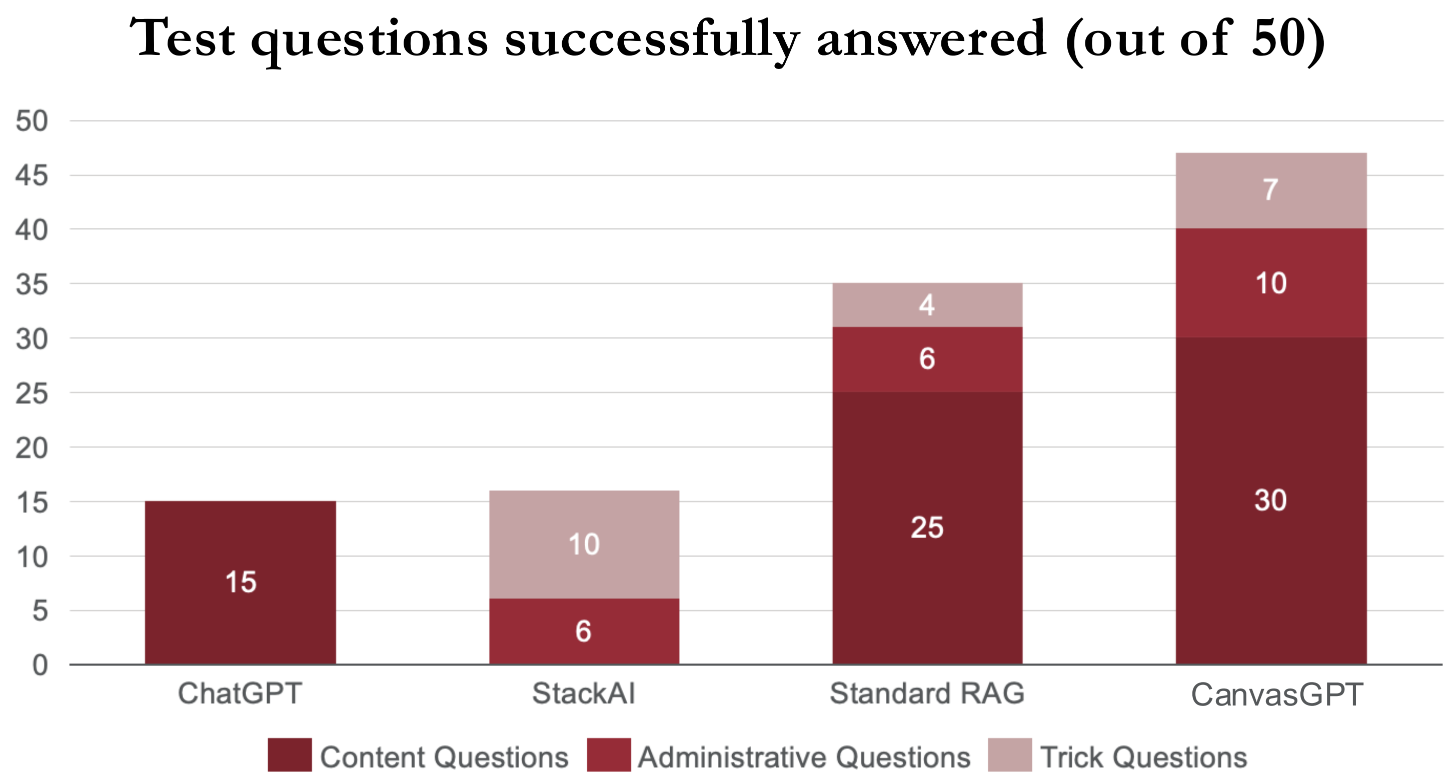
Answer relevant to the course and question?

- LLM evaluation

No harmful outputs?

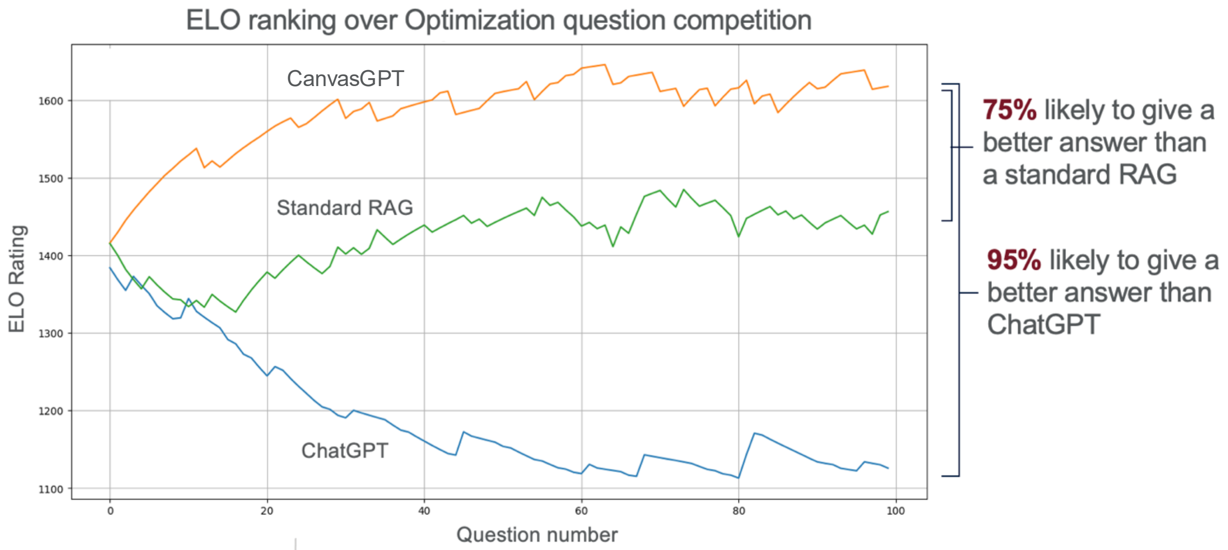
- Algorithmic evaluation

# Evaluation




- All models are evaluated on a test set of **real student questions**, as well as **edge cases** to ensure the desired behavior is achieved

# Finetuning




- Tuning pipeline based on **ELO scoring**
  - Rank models or iterations based on their relative performance in answering optimization-related questions, adjusting scores based on comparative outcomes.
  - Allows to assess robustness as well as accuracy, as higher-rated models need consistently strong answers across the entire question set to maintain their ranking

# Deployment plan: A/B Testing

## Agentic Student

Students can decide what class materials the tool can use to answer the question

**V.S.**

The type of class materials used to answer are selected by a custom-built, autonomous AI agent

## Agentic AI

## Extent of Personalization

Determine extent of student input in shaping the tool's behavior

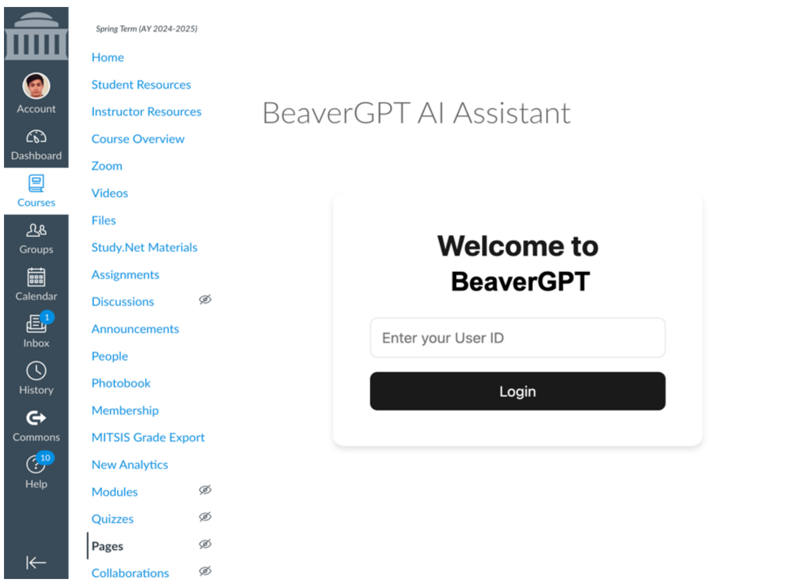
How to best address out-of-scope questions to avoid hallucinations while retaining student engagement

## Reliability Safeguards

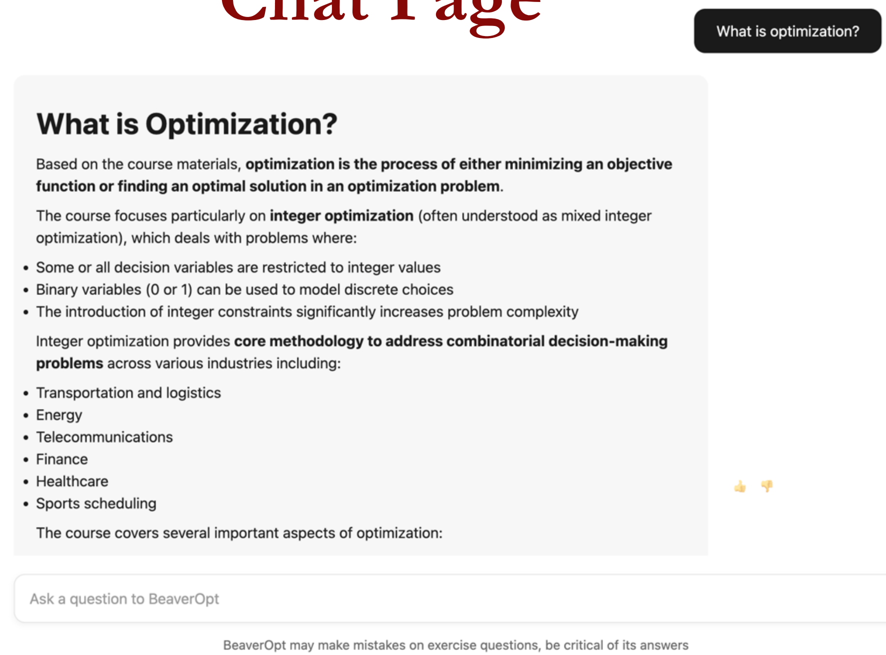
# Demo

# Custom-build UI

## Login Page




## Chat Page




- **Flexible**: The tool can be easily adjusted to specific needs, and different versions can be evaluated through A/B testing
- **Integrated**: The tool can be integrated within Canvas, requiring minimal upkeep

# Demo: Concept Questions

What is integer optimization?

## What is Integer Optimization?

Integer optimization is a mathematical optimization approach where **some or all decision variables are restricted to integer values**. It is often referred to as mixed-integer optimization when some variables are integers and others are continuous.

### Key characteristics:

- **Captures real-world discrete decisions** such as the number of aircraft (106 vs. 107) or warehouses (5 vs. 6)
- **Binary variables** (0 or 1) are particularly powerful for modeling discrete choices ("go/no-go" decisions)
- **Significantly enhances modeling power** compared to continuous optimization by enabling:
  - Logical relationships
  - If-else statements
  - Non-linearities
  - Other complex constraints

### Mathematical formulation:

For mixed-integer optimization (MIO):

$$\min c^T x$$

$$\text{s.t. } Ax = b$$

$$x \in \mathbb{Z}_+^p \times \mathbb{R}_+^{n-p}$$

Where some variables ($p$ of them) are restricted to integer values, while others remain continuous.

### Important considerations:

- Integer variables make optimization problems **much harder to solve** computationally
- The course focuses primarily on mixed-integer linear optimization
- Similar techniques apply to both pure integer and mixed-integer problems
- Mixed-integer non-linear optimization problems present even greater challenges

Integer optimization is both a science and an art, requiring careful formulation based on practical requirements and computational considerations.

Sources ordered by relevance:

- Lecture1 @ page 16 (Cosine similarity Score: 89.45/100)
- Lecture1 @ page 41 (Cosine similarity Score: 85.94/100)
- Lecture1 @ page 3 (Cosine similarity Score: 84.38/100)
- 15.083000 - E62-276 - Integer Optimization 02 03 2025 Sous-titres English (United States) from 09-00 to 09-55 (Cosine similarity Score: 79.69/100)

What is non-linear optimization?

## Non-linear Optimization

Non-linear optimization (NLO) refers to optimization problems that have either a non-linear objective function or non-linear constraints, in contrast to linear optimization.

### General Formulation

The standard form of a non-linear optimization problem is:

$$\min \quad f(x) \text{ s.t. } \quad g_i(x) \leq 0, \quad \forall i = 1, \ldots, m \; x \geq 0 \tag{1}$$

Where:

- $f : \mathbb{R}^n \to \mathbb{R}$ is a non-linear objective function
- $g_i : \mathbb{R}^n \to \mathbb{R}$ are non-linear constraint functions

### Important Sub-classes

Non-linear optimization encompasses several important sub-classes:

1. **Unconstrained optimization**: $\min f(x) : x \in \mathbb{R}^n$
2. **Convex optimization**: Problems where $f, g_1, \ldots, g_m$ are all convex functions
3. **Quadratic Programming (QP)**:

$$\min_x x^T Q x + c^T x, \quad \text{s.t.} \quad Ax = b, x \geq 0$$

- **Convex QP**: When $Q$ is positive semi-definite

4. **Quadratic Constrained Quadratic Programming (QCQP)**:

$$\min_x x^T Q_0 x + c_0^T x, \quad \text{s.t.} \quad x^T Q_i x + c_i^T x \leq r_i, Ax = b$$

5. **Second-order Cone Programming (SOCP)**:

$$\min_x c_0^T x, \quad \text{s.t.} \quad \|Q_i x + p_i^T\|_2 \leq s_i^T x + r_i, Ax = b$$

6. **Semi-definite Programming (SDP)**:

$$\min_X \langle C, X \rangle, \quad \text{s.t.} \quad \langle A_i, X \rangle = b_i, X \geq 0$$

### Applications

Non-linear optimization has numerous practical applications:

- **Machine learning**: Linear regression, logistic regression
- **Geometry**: Projection, distance, maximum inscribed ellipsoid
- **Finance**: Portfolio optimization
- **Operations research**: Power plant location, shortest path with coordination

### Key Takeaways

- Convex optimization problems can be solved efficiently because local optima are also global optima
- Some convex optimization problems are easier to solve than others
- Non-convex optimization problems are generally much harder to solve
- Non-linear optimization provides a powerful modeling framework for many real-world problems

Sources ordered by relevance:

# Demo: Use of audio

What makes the 2 facility location formulation computationally different?

## What makes the 2 facility location formulation computationally different?

Based on the lecture materials provided, there are two formulations for the facility location problem that are computationally different:

1. **Facility Location (FL)** formulation with constraints:
   - ∑(j=1 to n) xij = 1, ∀i
   - xij ≤ yj, ∀i, j
   - xij ∈{0, 1}, yj ∈{0, 1}
2. **Aggregate Facility Location (AFL)** formulation with constraints:
   - ∑(j=1 to n) xij = 1, ∀i
   - ∑(i=1 to m) xij ≤ myj, ∀j
   - xij ∈{0, 1}, yj ∈{0, 1}

The key computational differences between these formulations are:

1. **Tightness of the formulation**: The FL formulation is tighter than the AFL formulation. As stated in the lecture, "PFL ⊆ PAFL: (FL) is a stronger formulation than (AFL)." This means the continuous relaxation of FL is closer to the integer feasible region.
2. **Number of constraints**: The FL formulation has more constraints (m×n constraints for xij ≤ yj) compared to the AFL formulation (n constraints for ∑xij ≤ myj).
3. **Computational performance**: Despite having more constraints, the FL formulation performs better computationally for larger problem instances. The lecture shows a graph where "the computation time of less constrained model [AFL] actually goes up by a lot" while the tighter formulation (FL) "is actually quite stable" as problem size increases.
4. **Solution quality**: Both formulations are equivalent in terms of integer solutions (ZIO(AFL) = ZIO(FL)), but their continuous relaxations differ (ZLO(AFL) ≤ ZLO(FL)).

The lecture explicitly demonstrates this with computational results showing that for larger numbers of factories/customers, the FL formulation (with more constraints but tighter bounds) significantly outperforms the AFL formulation in terms of solution time.

As stated in the lecture: "Tighter formulations lead to smaller branching trees and more effective cuts; the impact on computational performance can be significant" and "The quality of an MIO formulation can be judged by the closeness of its feasible region to the one of its continuous relaxation."

Sources ordered by relevance:

- 15.083 - Recitation 2 - E51-376 Sous-titres English (United States) from 07-03 to 07-59 (Cosine similarity Score: 81.64/100)
- 15.083 - Recitation 2 - E51-376 Sous-titres English (United States) from 23-04 to 23-57 (Cosine similarity Score: 71.09/100)
- 15.C57: Optimization Methods (Cosine similarity Score: 69.14/100)
- 15.083000 - E62-276 - Integer Optimization 02 12 2025 Sous-titres English (United States) from 1-12-01 to 1-12-52 (Cosine similarity Score: 66.41/100)

The computational aspect was only mentioned in recitation, and the jupyter notebook support is not in our database

# Demo: Intelligence

**15.C57/15.C571/6.C57/6.C571/IDS.C57: Optimization** **Fall 2024**

**Assignment 1: Linear optimization**

*Assigned: September 09; Due: September 23.*

**True or False?** [20 pts]

Consider the standard form polyhedron $\mathcal{P} = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$. Assume that $\boldsymbol{A}$ has dimensions $m \times n$ and that its rows are linearly independent. Let $\boldsymbol{c} \in \mathbb{R}^n$. Consider the problem of minimizing $\boldsymbol{c}^\top \boldsymbol{x}$ over the polyhedron $\mathcal{P}$. For each of the following statements, state whether it is true (with a short proof) or whether it is false (with a counterexample).

(a) If $n = m + 1$, then $\mathcal{P}$ has at most two basic feasible solutions.

(b) The set of all optimal solutions is bounded.

(c) At every optimal solution, no more than $m$ variables can be positive.

(d) If there are more than one optimal solutions, there are uncountably many optimal solutions.

(e) If there are several optimal solutions, then at least two basic feasible solutions are optimal.

(f) Let $\boldsymbol{d} \in \mathbb{R}^n$. Consider the problem of minimizing $\max\{\boldsymbol{c}^\top \boldsymbol{x}, \boldsymbol{d}^\top \boldsymbol{x}\}$ over $\mathcal{P}$. If this problem has an optimal solution, it must have an optimal solution that is an extreme point of $\mathcal{P}$.

**TRUE**
**FALSE**
**FALSE**
**TRUE**
**FALSE**
**FALSE**

# Thank you!