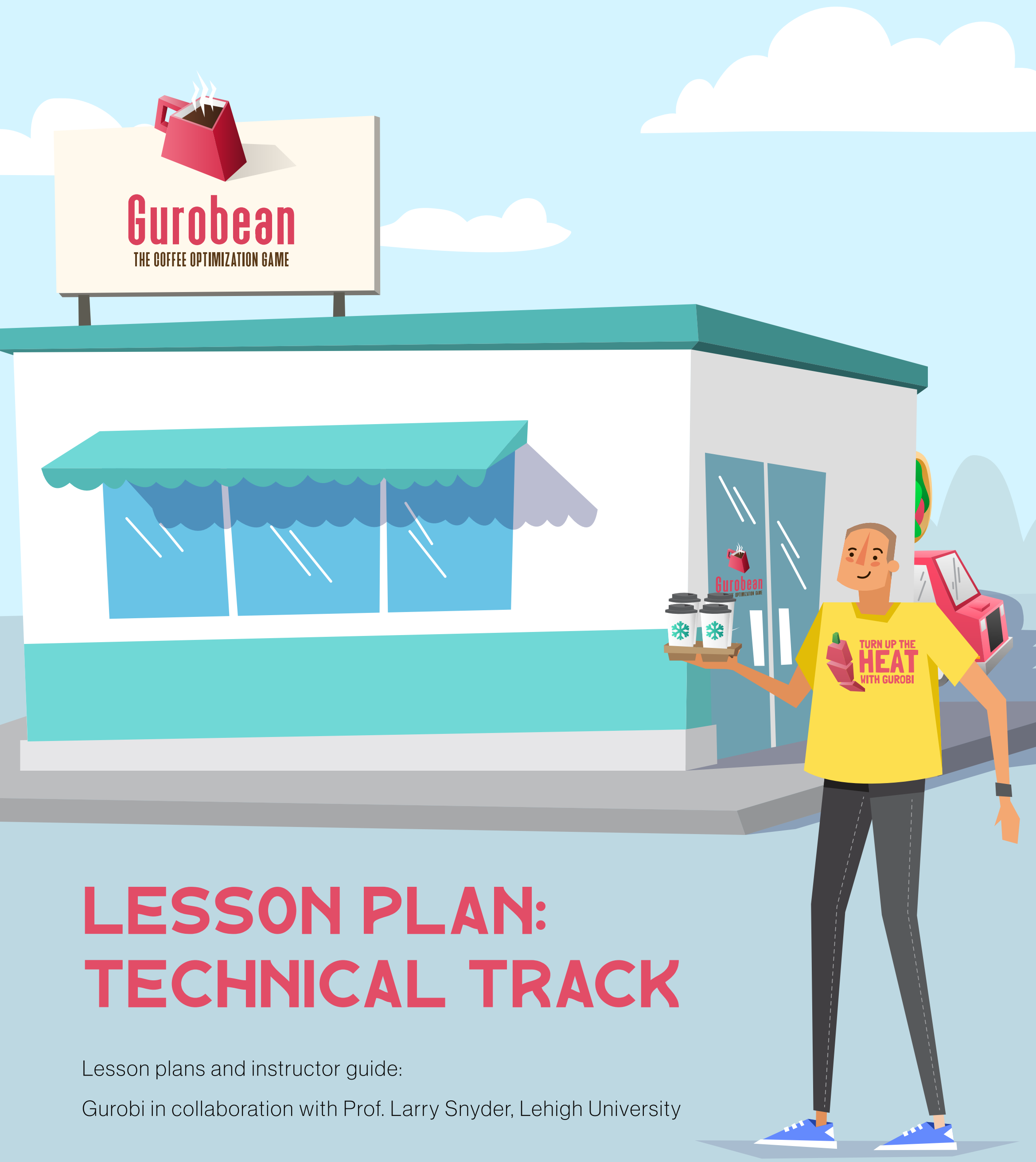
Gurobi®



LESSON PLAN: TECHNICAL TRACK

Lesson plans and instructor guide:

Gurobi in collaboration with Prof. Larry Snyder, Lehigh University

INTRODUCTION (FOR INSTRUCTORS)

The Gurobean Optimization Game introduces students to the challenges of decision-making in stochastic, dynamic, and nonlinear systems. Unlike static optimization problems, Gurobean evaluates decisions using simulation, making performance an emergent property rather than an immediate calculation.

This version of the teaching guide is intended for technically inclined students, although no prior optimization knowledge is required. This includes those studying operations research, computer science, engineering, or advanced analytics, and the guide is well suited as a conceptual lead-in to formal coursework in optimization, simulation, supply chain, or queueing systems.

Intended Audience

This guide is designed for learners who:

- Are comfortable with mathematical modeling or algorithms,
- Have seen or will soon see optimization models formally
- May not yet have intuition for why simulation and expected-value thinking are necessary.

This version works especially well before introducing:

- Integer or nonlinear programming,
- Simulation modeling,
- Queueing theory.

Learning Objectives (Technical Emphasis)

After the lesson, students should be able to articulate that:

- Optimization models define decisions, objectives, and constraints, but evaluation may require simulation.
- Expected-value optimal solutions are not guaranteed to dominate in every realization.
- Nonlinearity and congestion introduce feedback effects that invalidate simple reasoning.
- Simulation output must be interpreted statistically, not deterministically.
- Feasibility defines the solution space; performance defines solution quality.

SAMPLE LESSON STRUCTURE (45–75 MINUTES)

Before class

Play through several rounds yourself. Read the Game Guide first, then watch how the performance metrics stabilize as each simulation runs.

Find a "you beat Gurobi" moment. Identify at least one round where the optimizer's solution is outperformed in a single run; you'll use it to make the expected-value point.

(Optional) Have students play Round 1 beforehand. Learning the controls ahead of time saves class time and lets you start with the concepts.

In class

Introduce the game. Frame it as a decision-making system, not just a game: decision variables (the sliders), constraints (ingredients, capacity), stochastic customer arrivals, and performance evaluated by simulation.

Play the early rounds. Have students set the sliders, predict what will happen, then run the simulation and compare their predictions to the observed behavior.

Discussion. What did you expect, and what actually happened? Why did the results change between runs even though the decisions didn't?

Play the later rounds. Introduce pricing, queueing, and congestion. Watch profit stop behaving smoothly.

Discussion. Why did profit stop improving, or become non-monotonic? Which simplifying assumptions broke? If you were designing an algorithm to solve this, what would it do?

Choose one: continue into the rounds that emphasize uncertainty, or step away for an "optimization & simulation 101" primer (expected value, why enumeration is impractical, what a solver does, transient vs. steady-state simulation).

Discussion: optimization vs. simulation. At the comparison screen, emphasize that Gurobi optimizes expected value, a single simulation generates one realization, and variance explains the discrepancies. This is a natural bridge to stochastic optimization, robustness, and Monte Carlo evaluation.

Discussion: how the objective's shape changes. Put two Profit Monitor curves side by side (for example, profit against markup and profit against cups brewed). For each, have students mark where the maximum sits, how high it is, and how sharp it is.

Then change a fixed quantity such as the beans available, or the price, and ask them to predict how the peak moves before testing. Which combination yields the highest peak, and where on the curve does it occur? Tie a flat peak to robustness and a sharp peak to sensitivity, and note that a binding resource constraint can place the optimum at the limit rather than at the curve's natural peak.

After Class

Championship Mode (homework or in class). Spin up a private Class Championship on a shared, fixed scenario, or send students to the weekly Global Championship to compete on a worldwide leaderboard. See Part 3 for both options.

Coffee Break

Go get some coffee! ☕

CORE TEACHING MOMENTS (TECHNICAL)

Simulation Warm-Up Bias

Early results are unstable because the system starts empty. Discuss:

- Transient vs. steady-state behavior,
- Burn-in periods.

Nonlinearity

Highlight:

- Non-convex response surfaces,
- Sensitivity to small changes,
- How the objective's shape shifts with the scenario: the peak moves, changes height, and changes sharpness as the available ingredients, the price, and the round vary.

Expected Value vs. Realization

Explicitly discuss:

- Why "winning one run" is meaningless,
- How expected value governs optimality.

INSTRUCTOR INTERVENTION POINTS

After the first simulation run: ask students why results changed even though decisions didn't.

When profit fluctuates: emphasize that variability is expected, not an error.

When students beat Gurobi: explain expected value vs. single realizations explicitly.

When all solutions seem "reasonable": reinforce that nonlinearity creates many local optima.

OPTIONAL EXTENSIONS

- Formalize a single round as a mathematical optimization model.
- Discuss why closed-form objectives disappear in later rounds.
- Compare brute-force enumeration vs. solver approaches.

WHAT STUDENTS SHOULD TAKE AWAY

Gurobean shows that models do not eliminate uncertainty; they help us choose rationally before uncertainty is revealed. Simulation is not a replacement for optimization, but a necessary partner in complex systems.