Block 4: From Data to Decisions - Optimization Modeling

Python Module for Incoming ISE & OR PhD Students

Will Kirschenman

August 7, 2025 | 1:00 PM - 1:50 PM

North Carolina State University

NC STATE UNIVERSITY

Welcome Back to Block 4!

- **Goal:** Learn to make optimal decisions with Python optimization
- **Duration:** 50 minutes of hands-on optimization modeling
- **Format:** Presentation + interactive Pyomo exercises

What We'll Cover

Optimization fundamentals · Pyomo modeling · Linear Programming · Real-world applications · Solver comparison

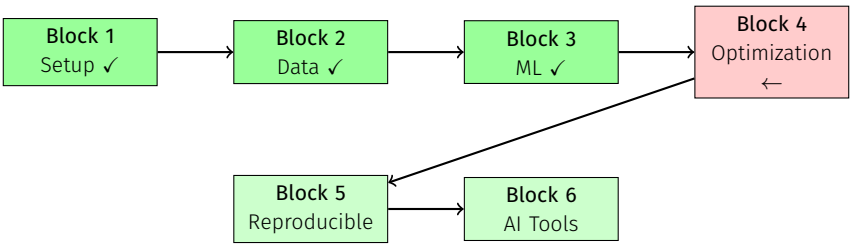
Session Learning Objectives

By the end of Block 4, you will:

1. Understand **optimization fundamentals** and Linear Programming
2. Become familiar with **Pyomo modeling** for real-world decision problems
3. Build and solve **Mixed-Integer Linear Programs** (MILPs)
4. Apply optimization to **course scheduling** and resource allocation
5. Compare **solver performance** and understand trade-offs
6. Appreciate the power of **moving from prediction to prescription**

Our Mission
Transform from asking ”what might happen?” to deciding ”what should happen!”

Recap: Where We Are



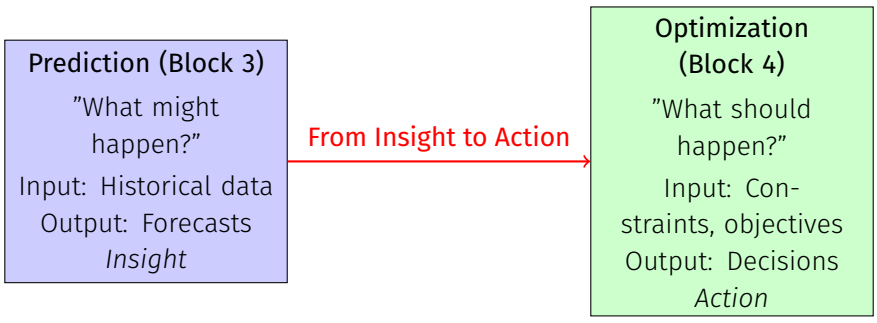
The Journey So Far

Block 3: ”Based on your data, you’ll likely publish 3.2 papers”

Block 4: ”To maximize research output, schedule your time THIS way”

Optimization Fundamentals

What is Optimization?

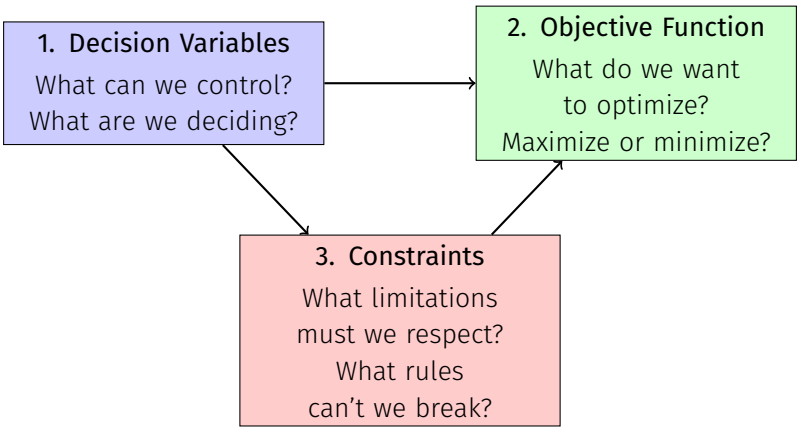


PhD Context

Optimization helps you make the **best** decisions given your constraints and goals!

The Three Essential Components

Every optimization problem has exactly three parts:



Remember

If any component is missing, you don't have an optimization problem!

PhD Student Time Allocation Example

Scenario: You have 10 hours to allocate between studying and sleeping. How should you maximize satisfaction?

Mathematical Formulation
Decision Variables:

- x_1 = hours studying
- x_2 = hours sleeping

Objective Function:

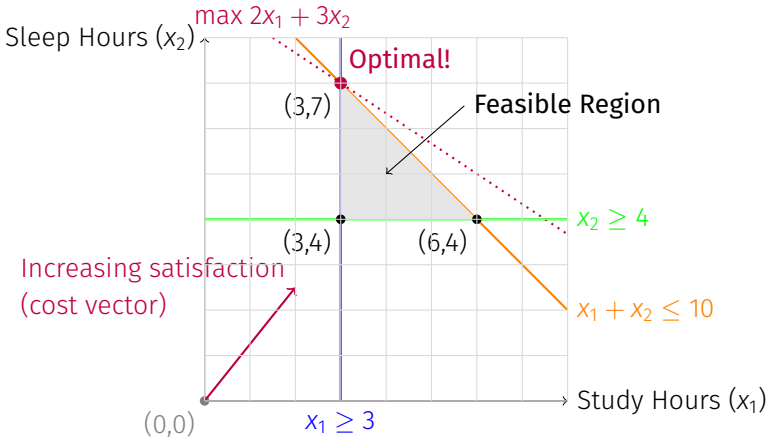
- Maximize $2x_1 + 3x_2$

Constraints

- $x_1 + x_2 \leq 10$ (time limit)
- $x_1 \geq 3$ (min study)
- $x_2 \geq 4$ (min sleep)
- $x_1, x_2 \geq 0$ (non-negative)

The Answer
Optimal solution: Study 3 hours, sleep 7 hours. Total satisfaction = 27 points!

Linear Programming Visualization

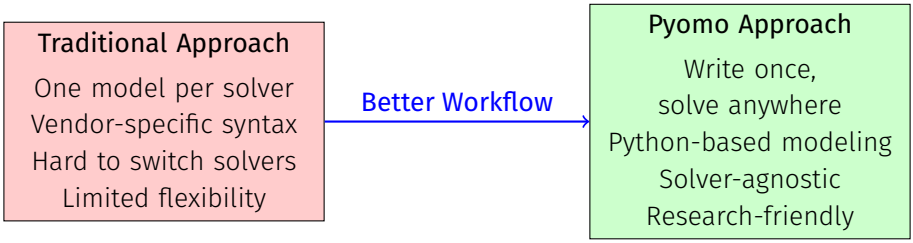


Key Insight

The optimal solution is always at a corner point of the feasible region!

Introduction to Pyomo

Why Pyomo for Optimization?



Supported Solvers

- **Open-source**: CBC, GLPK, HiGHS
- **Commercial**: Gurobi, CPLEX
- **Specialized**: BARON, Couenne, Hexaly

Perfect for Coursework or Research

Develop with free solvers, scale with commercial ones. Academic licenses make it affordable!

Pyomo Modeling Workflow

Four simple steps to optimization success:

```
model = pyo.ConcreteModel()

# Step 2: Define decision variables
model.x1 = pyo.Var(within=pyo.NonNegativeReals)
model.x2 = pyo.Var(within=pyo.NonNegativeReals)

# Step 3: Define objective and constraints
model.objective = pyo.Objective(expr=2*model.x1 + 3*model.x2, sense=pyo.maximize)
model.time_limit = pyo.Constraint(expr=model.x1 + model.x2 <= 10)
model.min_study = pyo.Constraint(expr=model.x1 >= 3)

# Step 4: Solve the model
solver = pyo.SolverFactory('cbc')
result = solver.solve(model)
print(f"Study␣{pyo.value(model.x1)},␣Sleep␣{pyo.value(model.x2)}")
```

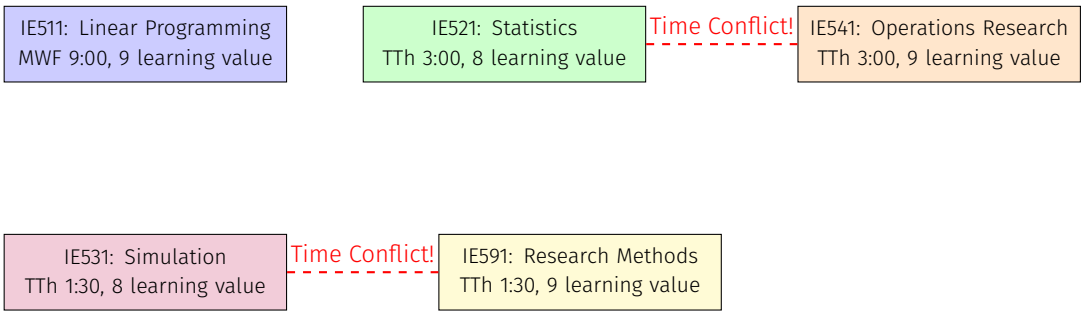
That's It!

Four steps to solve any linear optimization problem. The syntax stays the same for complex models!

Real-World Application

NC State Course Scheduling Challenge

The ultimate PhD student optimization problem:



Constraints

9-12 credits · No time conflicts · Area balance · Prerequisites · Maximize learning value

Course Scheduling ILP Model

This is an Integer Linear Program (ILP):

```
# Decision variables: Binary (0/1) for each course
model.take_course = pyo.Var(COURSES, within=pyo.Binary)

# Objective: Maximize total learning value
model.objective = pyo.Objective(
    expr=sum(learning_value[c] * model.take_course[c] for c in COURSES),
    sense=pyo.maximize
)

# Constraints
# 1. Credit requirements (9-12 credits)
model.min_credits = pyo.Constraint(
    expr=sum(credits[c] * model.take_course[c] for c in COURSES) >= 9
)

# 2. Time conflicts (max 1 course per time slot)
for slot in TIME_SLOTS:
    model.add_component(f'conflict_{slot}', pyo.Constraint(
        expr=sum(model.take_course[c] for c in courses_in_slot[slot]) <= 1
    ))
```

Optimal Course Schedule Results

The solver finds the best schedule:

Course	Name	Credits	Time	Learning
IE511	Linear Programming	3	MWF 9:00	8
IE521	Statistics for Engineers	3	TTh 3:00	7
IE531	Simulation	3	TTh 1:30	7
IE591	Research Methods	3	MWF 1:30	6

Solution Summary

- **Total Credits**: 12
- **Total Learning**: 28 points
- **Areas Covered**: 3 different
- **Time Conflicts**: None!

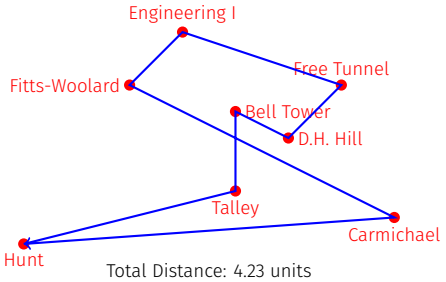
Why This Schedule?

The model chose high-value courses while respecting all constraints. No conflicts, balanced areas!

Solver Performance

The Traveling Salesman Problem (TSP)

Challenge: Visit all NC State landmarks in the shortest route



The Challenge

TSP is NP-hard: With 8 locations, an asymmetric TSP has $(8-1)! = 5,040$ possible tours!

This is where solver performance matters.

Solver Performance Comparison

Different solvers, different performance:

Solver	Type	Solve Time	Status
HiGHS	Open-source	0.0123s	Optimal
CBC	Open-source	0.0456s	Optimal
GLPK	Open-source	0.0234s	Optimal
Gurobi	Commercial	0.0034s	Optimal
CPLEX	Commercial	0.0028s	Optimal

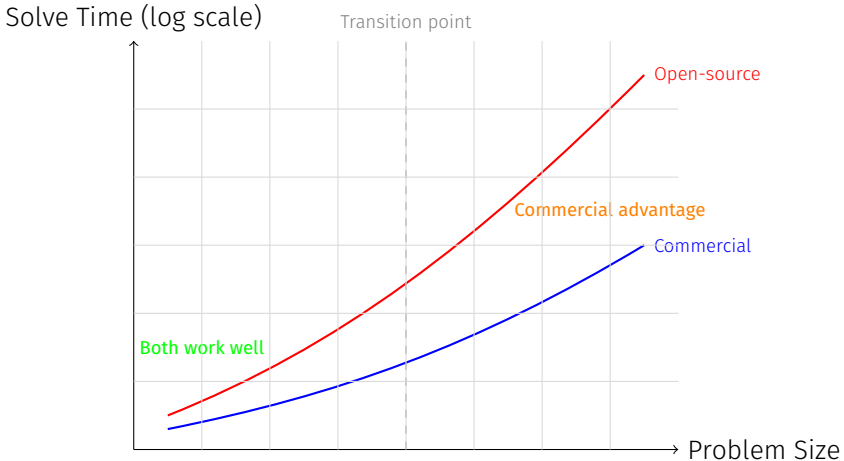
Open-Source Solvers

- **Free** for all use
- Good for **development**
- Slower on large problems
- Great for **learning**

Commercial Solvers

- **Academic licenses** available
- Much faster on large problems
- Better for **production**
- Industry standard

When Solver Choice Matters



Strategy for PhD Students

Develop and prototype with open-source, scale to commercial for large problems or tight deadlines!

Wrap-up & Preview

What We Accomplished in Block 4 ✓

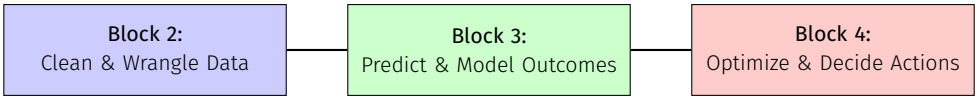
- ✓**Mastered optimization fundamentals** - variables, objectives, constraints
- ✓**Learned Pyomo modeling** - solver-agnostic Python optimization
- ✓**Built real decision models** - coffee budgets to course scheduling
- ✓**Solved Integer Programs** - complex scheduling with binary decisions
- ✓**Compared solver performance** - open-source vs commercial trade-offs
- ✓**Moved from prediction to prescription** - from insight to action!

The Optimization Mindset

You now think in terms of decisions, not just predictions. That's the OR/ISE advantage!

From Data Science to Decision Science

Your complete analytics toolkit:



The Complete Flow

Clean data → Predict outcomes → Optimize decisions. You're now a complete data scientist!

Applications in Your PhD Research

Optimization will appear throughout your studies:

Coursework Applications

- **Experimental design**: Optimize sample sizes
- **Resource allocation**: Budget research time
- **Parameter tuning**: Optimize model hyperparameters
- **Research planning**: Schedule your PhD timeline

Research Domains

- **Manufacturing**: Production scheduling
- **Healthcare**: Staff scheduling, treatment plans
- **Transportation**: Route optimization, logistics
- **Finance**: Portfolio optimization, risk management

Remember

Every time you need to make the ”best” decision under constraints, that’s an optimization problem!

Preview: Block 5 (2:00 PM - 2:50 PM)

From Individual Work to Collaborative Research

What's Coming

- **Version control** with Git and GitHub
- **Reproducible research** workflows
- **Collaboration** best practices
- **Open science** principles
- **Documentation** strategies

Why This Matters

- Share your optimization models
- Collaborate with advisors and peers
- Reproduce your research results
- Prepare for industry/academia
- Build your professional portfolio

10-minute break, then we master research workflows!

Key Takeaways from Optimization

Technical Skills

- **Pyomo modeling** for any optimization problem
- **Problem formulation** - variables, objectives, constraints
- **Solver selection** and performance understanding
- **MILP modeling** for complex decisions
- **Solution interpretation** and validation

Research Mindset

- Think in terms of decisions, not just analysis
- Understand trade-offs and constraints
- Move from ”what happened” to ”what should happen”
- Appreciate the value of optimal solutions
- Consider computational complexity

Remember: The best optimization model is one that helps you make better decisions

From data to insights to actions - that’s the power of OR/ISE!

Next Steps in Your Optimization Journey

To continue growing your optimization skills:

1. **Practice More**: Try the interactive exercises with different scenarios
2. **Get Academic Licenses**: Apply for Gurobi/CPLEX student licenses
3. **Explore Advanced Topics**: Stochastic, robust, nonlinear optimization
4. **Join the Community**: INFORMS, optimization conferences, research groups
5. **Apply to Research**: Use optimization in your PhD research area

Resources

Pyomo docs • Gurobi Academic Program • OR-Tools • INFORMS • NC State ISE optimization faculty

Questions?

See you in 10 minutes for Block 5!