# Math 567: Integer and Combinatorial Optimization (Spring 2025)

| | |
|---|---|
| Credits | 3 |
| Time | Tue+Thu 12:05–1:20 PM |
| Location | TERR 24 (Pullman), TBD (Vancouver) |
| Instructor | Bala Krishnamoorthy |
| Check-in Hours | Tue 2–3 PM, Wed 10:30–11:30 AM (Zoom) |
| Email | kbala@wsu.edu |
| Web page | http://archive.math.wsu.edu/faculty/bkrishna/Math567.html |
| Text | **Class notes and handouts** |
| References | Dimitris Bertsimas and Robert Weismantel: Optimization over Integers<br>Dynamic Ideas, ISBN: 0-9759146-2-6. |
| | Laurence A. Wolsey: Integer Programming<br>John Wiley and Sons, ISBN: 0-471-28366-5. |

## Description of the Course

Solving optimization problems with variables restricted to take integer values, as opposed to real values, is called integer optimization. The subject, also commonly called integer programming (IP), uses concepts form various areas of mathematics and computer science including linear algebra, combinatorics, geometry of numbers, algorithms and data structures, and more recently, machine learning. IP techniques have been used to model and solve problems in electrical power systems, airline crew scheduling, economic lot-sizing, transportation and logistics, treatment of tumors using radiation, computational biology, and many other areas.

This graduate level course will provide a detailed treatment of the theory, solution methods, and applications of integer and combinatorial optimization. Topics covered could include IP formulations, binary expressions and conjunctive normal form (CNF), enumerative methods (branch-and-bound), theory of cutting planes, lattice-based approaches including basis reduction, basics of computational complexity, and machine learning approaches for branching, cutting planes, and other hard combinatorial optimization problems. We will also emphasize the use of state-of-the-art software packages to model and solve real-life problems. The packages AMPL and solver Gurobi will be introduced through homework problems and a project.

As a prerequisite, students should have taken an undergraduate level course in Linear Optimization (MATH 364, MATH 464, or equivalent), **or obtain the permission of the instructor** (exceptions for this requirement may be made on a case-by-case basis). Some familiarity with computer programming languages or packages such as Matlab or Python will be helpful.

## Organization and Grading

The course will have around six homework assignments. These assignments will include theoretical problems as well as ones that involve the use of software packages. Apart from the homework assignments, there will be two course projects that will involve the use of software packages AMPL (with Gurobi) and/or Matlab/Python. The total score for the course will calculated using the weights: homework: 60% and projects: 40%. There will be no exams. This total score for the course will be curved to assign grades.

**Software:** The modeling language AMPL (www.ampl.com) along with the solver Gurobi will be used in the course. Students will model several examples as well as real-life integer optimization problems using this software. Further analysis could be done using Matlab or Python. The course projects will involve creating a complete model for one or two real-life instances of integer optimization problems, interpreting the solutions obtained, or implementing heuristic algorithms for large scale combinatorial optimization problems.

## Academic Integrity

I encourage discussion of homework problems with others. But each student should submit their own (hand or type) written solutions and/or computer programs. You might search the internet for finding materials to enhance your understanding. If you use such material to assist in your homework submission, you **should** cite the relevant sources. Plagiarism or cheating will **not** be tolerated. In particular, do not copy blindly from internet sources! Such behavior is easy to detect, and will result in a zero grade for the item in question and possibly a failing grade for the entire course.

**AI Use Permitted with Acknowledgment:** You are welcome to try AI tools based on generative models such as CoPilot, Chat-GPT, Gemini, and others. Any such use in your homework assignments or projects should be properly acknowledged in your submission. One way to list such an acknowledgment is as follows:
*"This response is based on the solution provided by* CoPilot *when queried with* the exact text of the problem statement *on* Feb 12, 2025.*"* (change the entries in red appropriately).

At the same time, be warned that most of these tools are not (yet!) guaranteed to give you the correct response each time!

## University Syllabus

For information on the following WSU policies, please see the University Syllabus. Students are responsible for reading and understanding all university-wide policies and resources pertaining to all courses provided on this web page.

- Reasonable Accommodations
- Arrangements for Religious Reasons
- Emergencies on Campus (including active shooter and severe weather)
- Student Support Resources (including Student Care Network and Campus Resources and Support)

**Tentative List of Topics Covered**

1. Integer programming basics and formulations: 7 lectures
   - IP formulations, modeling with binary variables, facility location, traveling salesman and other network problems, disjunctive constraints
   - binary expressions - conjunctive normal form (CNF)
   - strength of formulations, aggregated and disaggregated formulations
2. Solvers and applications: 3 lectures
   - Introduction to commercial software packages (AMPL + Cplex/Gurobi)
   - $p$-median/$p$-center problems, fixed charge network flow, local area network (LAN) planning
3. Enumerative methods, branch-and-bound: 3 lectures
4. Integral polyhedra, matching, other topics: 6 lectures
   - totally unimodular matrices
   - network matrices, maximum-cardinality/maximum-weight matching, total dual integrality
5. Theory of valid inequalities: 4 lectures
   - Chvátal-Gomory (CG) cuts, mixed-integer rounding, disjunctive cuts
   - knapsack cover inequalities, Lovász-Schrijver procedure
6. Lattices and applications: 4 lectures
   - Hermite normal form (HNF) and Diophantine equations
   - basis reduction (BR), Lenstra's algorithm, BR-based reformulation techniques
   - shortest vector problem (SVP), closest vector problem (CVP)
7. Machine learning and IP: 3 lectures
   - Learning to branch, learning to select cuts
   - Learning hard optimization problems