Course Description: Applications of integer and heuristic programming techniques for solving combinatorial optimization problems. Topics include computational complexity, relaxations, branch and bound, cutting planes, simulated annealing, tabu search, and genetic algorithms.

Prerequisites: IENG 350 or 553 and knowledge of a computer programming language.


Grading: Midterm Exam 25%
         Final Exam 25% (Mon., Dec. 14th, 3 – 5 p.m.)
         Projects 25%
         HW Assignments 25%

Tentative Grading Scale: 90 – 100 A
                        89 – 80 B
                        79 – 70 C
                        69 – 60 D
                        Below 60 F

Exams: All examinations will be closed book and closed notes. Also, calculators will not be permitted to complete the exams. All work must be shown in order to receive full credit, and instructions should be followed in order to avoid point deductions. There will be no makeup exams. However, if you miss an exam due to serious illness (documented) or serious family emergency (documented), then you will take a cumulative final exam. More specifically, if a student miss an exam, the student must email me explaining why they will miss (or have missed) the exam within 24 hours before (or after the exam). There is no makeup (cumulative final) exam without a proper and certified excuse. If you miss the final exam, you will be required to take a cumulative final exam.

Assignments: Assignments will be given periodically throughout the semester which may be collected and graded. You may be required to use Matlab or MPL/CPLEX to complete the assignments.

Academic Dishonesty: Acts of academic dishonesty such as cheating on any of the exams, assignments, or project will result in a failing grade (a grade of zero) on that exam, assignment, or project and will be reported to the student’s advisor, department, and college.
**Electronic Devices:** Cell phones or other electronic devices must be turned off and put away during class.

**Statement on Attendance:** Student attendance is mandatory. The basis for an excused absence will follow University and IMSE department policy. If an emergency arises that require an absence from a class, it is your responsibility to get the notes and all other information that was covered in class from a fellow student.

**Social Justice Statement:** West Virginia is committed to social justice. I concur with that commitment and expect to maintain a positive learning environment based upon open communication, mutual respect, and nondiscrimination. Our University does not discriminate on the basis of race, sex, age, disability, veteran status, religion, sexual orientation, color or national origin. Any suggestions as to how to further such a positive and open environment in this class will be appreciated and given serious consideration.

**Accommodations:** If you are a person with a disability and anticipate needing any type of accommodation in order to participate in this class, please advise me and make appropriate arrangements with Disability Services (304-293-6700).

**Tentative Course Topics** (may not be taught in this order)
1) Review of Linear Programming Models.
2) Transformation of Nonlinear Programming Models to Linear Programming Models.
4) Bounds and Relaxations (Linear and Lagrangean).
5) Integer Linear Programming Techniques (Exact Methods): Complete Enumeration, Partial Enumeration (Branch and Bound), and Cutting Planes.
6) Modeling and Solving Integer Programming Models using Microsoft Excel and MPL/CPLEX.
7) Computational Complexity.
8) Combinatorial Optimization Problems and their Formulations.
9) Heuristics (e.g., Lagrangean Heuristics, Exchange, Add/Drop).
10) Meta-heuristics (e.g., Simulated Annealing, Tabu Search, Genetic Algorithms).
11) Testing the Performance of a Heuristic.
12) Other Topics (e.g., Decomposition, Ant Systems, Other Meta-heuristics) if time permits.

**Student Learning Objectives:**
Upon completing the course, the student will be able to:
1) Formulate and solve simple integer programming/combinatorial optimization problems using exact integer programming techniques and practical software.
2) Understand how to use a modeling language and a commercial solver to solve and analyze integer linear programming problems.
3) Understand the computational complexity of solving large-scale combinatorial optimization problems.
4) Understand the strengths and weaknesses as well as how to apply meta-heuristics for solving large-scale combinatorial optimization problems.
5) Develop efficient heuristic (or approximation) techniques for solving large-scale combinatorial optimization problems encountered in the real world.

Prepared By: A. McKendall          Date: August 17, 2015