Department of Electrical and Computer Engineering
EECE 4542: Advanced Engineering Algorithms

Syllabus

Instructor: Waleed Meleis, Associate Professor
Contact: message me on Piazza
(if needed, my email address is w.meleis@northeastern.edu)

Office: 404 Dana
Mailbox: 420 Dana

Teaching Assistant: Alex Tazin, tazin.a@husky.neu.edu

CONTENT

This course covers classical and modern algorithms that efficiently solve computer engineering problems. These problems arise in a wide range of disciplines: computer aided design, parallel computing, computer architecture, and compiler design, as well as applied mathematics, computer science, and operations research. Examples include parallel scheduling, weighted completion time scheduling, shortest paths, traveling salesman, minimum spanning trees, optimal matching, graph partitioning, graph coloring, vertex covers, and knapsack.

These problems are usually NP-complete which makes it unlikely that optimal solutions can be found in a reasonable amount of time. However, solutions that are within 1% of optimality can be efficiently obtained by exploiting the underlying structure of the search space.

We will cover the fundamentals of algorithm analysis and complexity theory, and then survey a wide range of combinatorial optimization techniques, including exhaustive algorithms, greedy algorithms, integer and linear programming, branch and bound, simulated annealing, and genetic algorithms. We will consider the efficient generation of optimal solutions, the development and evaluation of heuristics, and the computation of tight upper and lower bounds. Students will get experience in implementing and evaluating these algorithms.

TOPICS COVERED

1. Analysis of algorithms
2. Complexity theory
3. Greedy algorithms
4. Linear programming
5. Integer programming/Branch and bound
6. Local search (tabu search, simulated annealing, genetic algorithms)
7. Network flow

PREREQUISITES

Students should have a good understanding of the material covered in Fundamentals of Engineering Algorithms (EECE 2560). Students should be familiar with a high-level programming language such as C, C++ or Java, and should have a good background in algorithm design.

STRUCTURE AND REQUIREMENTS

The course is built around a collection of programming projects. Projects will be completed in groups of two students. There will be one project submission for each group. However, each student must completely understand everything about the solutions they turn in.

Lectures will cover much, but not all, of the required material for the course. There will be reading assignments that accompany each lecture, and you are also responsible for this material.

There will be a weekly quiz. Each quiz will evaluate your understanding of the material covered on past projects, homework and lectures. There will be no midterm or final exam. Your lowest quiz score over all quizzes except the last will be dropped; you cannot drop your last quiz score.

There will be approximately 6 homework assignments. Homework will be graded "Complete" or "Incomplete". To receive a grade of "Complete", you must have solved, or attempted to solve, every question. Homework solutions will be posted.

You are required to attend all lectures, complete all assignments, and regularly do the reading.

Your final grade for the class will be based on the following weights: 35% projects, 5% homework, and 60% quizzes. A satisfactory score in each area is required. Projects and quizzes will be graded on an absolute scale from A to F, where A = 10, A- = 9, B+ = 8, etc.) These values will be used to compute your final weighted score. Your final weighted score will be converted into your final grade in the following way: 10.00-9.50 = A, 9.00-9.49 = A-, 8.00-8.99 = B+, 7.00-7.99 = B, 6.00-6.99 = B-, 5.00-5.99 = C+, 4.00-4.99 = C, 3.00-3.99 = C-, 2.00-2.99 = D, 1.00-1.99 = F.

POLICIES

Homework assignments and projects have fixed due dates. No late submissions will be accepted.

If you miss a quiz, you will not be allowed to retake it. All course requirements must be completed during the semester. No incompletes will be given.

Changes to grades will only be made in the first week after the graded work has been returned to you.

Class attendance is required. If you miss a lecture, you are responsible for all material that was covered, announcements that were made, and handouts that were distributed in class.

I encourage you to ask questions in class and participate in discussions. However, if I can hear you talking to your neighbor, then your voice is too loud.

You should check Piazza daily for announcements and other information.

Solutions you turn in must be your own. Copying someone else's work and presenting it as your own, or submitting the same solution as someone else, is not allowed. You must not look at another student's assignment, or allow another student to look at your assignment. Project partners should collaborate on their project.

All students must adhere to Northeastern University's Policy on Academic Integrity. If you violate this policy will receive a lower grade in the course, and may receive an F. You will also be referred to NU's Office of Student Conduct where penalties range from deferred suspension to expulsion from the university.

Exceptions to any course policy may be made if you have a personal emergency that prevents you from participating in the course. In this case you must make arrangements with me as soon as possible, preferably within 24 hours.

READING

Most readings are in the coursepack which is on sale in the NU bookstore.

1. E. Aarts and J. Lenstra, *Local Search in Combinatorial Optimization*, Wiley, 1997. 0471948225
2. R. Ahuja, T. Magnanti, and J. Orlinand, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, 1993. 013617549X
3. G. Brassard and P. Bratley, *Fundamentals of Algorithmics*, Prentice Hall, 1996. 0133350681
4. W. Cook, W. Cunningham, W. Pulleyblank, and A Schrijver, *Combinatorial Optimization*, Wiley Interscience, 1998. 047155894X
5. T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, McGraw Hill, 1991. 0070131430
6. R. Fourer, D. Gay, and B. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*, Boyd & Fraser, 1993. 0894262327
7. M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP Completeness*, Freeman, 1997. 0716710455
8. F. Hillier and G. Lieberman, *Introduction to Mathematical Programming*, McGraw-Hill, 1995. 0079118291
9. E. Lawler, J. Lenstra, A. Rinnooy Kan, and D. Shmoys, *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, Wiley, 1997. 0471904139
10. C. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Dover, 1998, 0486402584
11. H. Williams, *Model Building in Mathematical Programming*, Wiley, 1998. 0471941115

POLICY ON ACADEMIC INTEGRITY

Northeastern University is committed to the principles of intellectual honesty and integrity. All members of the Northeastern community are expected to maintain complete honesty in all academic work, presenting only that which is their own work in tests and all other assignments. If you have any questions regarding proper attribution of the work of others, please contact me prior to submitting the work for evaluation.

Academic integrity is important for two reasons. First, independent and original scholarship ensures that students derive the most from their educational experience and the pursuit of knowledge. Second, academic dishonesty violates the most fundamental values of an intellectual community and depreciates the achievements of the entire university community.

Accordingly, Northeastern University views academic dishonesty as one of the most serious offenses that a student can commit while in college. The following is a broad overview of what constitutes academic dishonesty, but is not meant to be an all-encompassing definition.

Cheating: Intentionally using or attempting to use unauthorized materials, information or study aids in any academic exercise.

Plagiarism: Intentionally or knowingly representing the words or ideas of another as ones own in any academic exercise without providing proper documentation of source by way of a footnote, endnote, or intertextual note.

Unauthorized Collaboration: This refers to instances when students, each claiming sole authorship, submit separate reports which are substantially similar to one another. While several students may have the same source material (as in case write-ups), the analysis, interpretation, and reporting of the data must be each individuals.

All members of the Northeastern University community, students, faculty, and staff share the responsibility to bring forward known acts of apparent academic dishonesty. Any member of the academic community who witnesses an act of academic dishonesty should report it to the appropriate faculty member or to the Director of Judicial Affairs. The charge will be investigated and if sufficient evidence is presented, the case will be referred to Northeastern University Student Judicial Hearing Board. If found responsible of an academic dishonesty violation, a minimum sanction of probation will follow. (adapted from NU's Academic Honesty and Integrity Policy)