Instructor: Ichiro Suzuki, EMS 1219, 229-3718, suzuki@uwm.edu

Office Hours: MW 12:30 pm – 1:30 pm (till 5/10/17), or by appointment.

Prerequisite: Simultaneous enrollment in CompSci 535 (Algorithm Design and Analysis). Good programming skills. Graduate Standing for CS790.

Textbooks:


- **Recommended references**: We need to discuss asymptotic analysis of algorithms and NP-completeness. For these either of the following books (both are our CS535 textbook in recent semesters), as well as most other data structures/algorithms books you may have, should be a good reference.
  
  

- To study various heuristics, we plan to use handouts.

Course Homepage: Login to UWM D2L.

Background: Every initial configuration of Rubik’s cube can be solved in no more than 20 steps, but as yet no one knows how to compute such a short solution efficiently. Computing an optimal solution in a game having a large configuration space can indeed be extremely challenging. Similarly, many practically important optimization and search problems in various subareas of computer science and engineering are computationally challenging (e.g., transportation scheduling, motion planning of high degree-of-freedom robots, and more). Nonetheless, practically useful programs for these problems that produce “good” outputs have been written quite successfully, using various heuristic algorithms and methods.

Objectives:

- To study various heuristic algorithms and methods for coping with such challenging problems, including approximation algorithms, hill climbing, potential method, branch and bound, A*, and various randomized algorithms such as genetic algorithms and simulated annealing.

- To test some of these ideas through programming projects.

- To learn about recent applications of heuristic algorithms through further reading.

- To briefly discuss NP-completeness/hardness that characterizes a class of problems believed by many to be computationally challenging. (Currently, no other undergraduate course in the Computer Science curriculum covers NP-completeness.)
Outline: Tentatively, the lecture will cover the following material. Some topics may be skipped or assigned for self-study, and additional material may be introduced. (1 lecture), (2 lectures), etc., indicate an approximate number of lectures on each subject.

- Brief review of CS535 material: growth of functions, asymptotic notation and algorithm analysis. [GT] Ch. 1, [CLRS] Sec. 1.1–2.3, 3.1–3.2. (4 lectures)
- Introduction to NP-completeness. This is a new topic for undergraduate students (but not for graduate students). [GT] Sec. 13.3–13.3, [CLRS] Sec. 34.1–34.5. (4 lectures)
- Introduction to computational experiments. McGeoch [M] Ch. 1-3. (6 lectures)
- Approximation algorithms. [GT] Sec. 13.4, [CLRS] Sec. 35.1–35.3. (2 lectures)
- Branch and bound, backtracking, A*. [GT] Sec. 13.5, handouts or other resources. (4 lectures)
- Hill climbing, potential method. Handouts or other resources. (2 lectures)
- Genetic algorithms, simulated annealing. Handouts or other resources. (3 lectures)
- Other randomized algorithms. Handouts or other resources. (4 lectures)
- Additional topics in [M]. (As needed or time permitting)

Grading (Tentative):

- Homework 20%
- Midterm Exam (Wed 3/8/17) 20%
- Final Exam (Wed 5/17/17 3:00 pm – 5:00 pm) 20%; Final Exam may be a written test, a project presentation, or a combination of both.
- Projects 40%
- Some projects may require further reading, research and/or a presentation. Undergraduate students and graduate students may have different assignments.
- Approximate grading scale (subject to adjustment): A or A- for a total score of 85/100 or higher, B+, B or B- for 75 or higher, C+, C or C- for 65 or higher, D+, D or D- for 55 or higher, and F for below 55.

Study Guidelines and Requirements: TBA

Policies: TBA