Syllabus
Computer Science 601.436/636
Algorithmic Game Theory
Spring, 2020
(3 credits, EQ)

Description

This course provides an introduction to algorithmic game theory: the study of games from the perspective of algorithms and theoretical computer science. There will be a particular focus on games that arise naturally from economic interactions involving computer systems (such as economic interactions between large-scale networks, online advertising markets, etc.), but there will also be broad coverage of games and mechanisms of all sorts.

This course will have essentially three parts. We will begin by defining standard notions of equilibria (e.g. pure and mixed Nash, correlated and coarse-correlated, etc.) and studying algorithms for computing them (or at least approximations). This will include algorithms based on online learning and regret minimization, combinatorial algorithms, and algorithms based on convex optimization. We will also study the complexity-theoretic hardness of computing equilibria, focusing on Nash equilibria.

The second part of the class will be the study of the inefficiency of equilibria: how well each notion of equilibrium approximates important objectives such as the social welfare. This will include the study of the price of stability, the price of anarchy, price of total anarchy, etc., as well as connections between them. Important games we will study will include routing games, network formation games, potential games, and smooth games.

Finally, we will study algorithmic mechanism design: how to design algorithms that do not assume that the players will do what they are told, but rather take into account the incentives of the players. We will focus on incentive-compatible mechanisms, i.e., mechanisms for which the dominant strategy for each player is to tell the truth about their preferences. We will discuss combinatorial auctions, mechanisms without money, online mechanisms, and profit-maximizing mechanisms.

Prerequisites

Introduction to Algorithms / Algorithms I (601.433/633) or equivalent

Instructor

Professor Michael Dinitz, `mdinitz@cs.jhu.edu`, `http://www.cs.jhu.edu/~mdinitz/`
Office: Malone 217
Office hours: Tuesdays 9:30am–10:30am, and by appointment

Teaching Assistant

Yasamin Nazari, `ynazari@jhu.edu`, `http://www.cs.jhu.edu/~ynazari/`

Meetings

Tuesday, Thursday, 3–4:15 pm, Shaffer 301

Textbook
Nisan, Roughgarden, Tardos, and Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007. `http://www.cambridge.org/us/9780521872829`, username: agt1user, password: camb2agt.

Optional: Roughgarden, Tim. *Twenty Lectures on Algorithmic Game Theory*. Cambridge University Press, 2016. Available online from the JHU Library. `https://catalyst.library.jhu.edu/catalog/bib_7503277`

Online Resources
Course webpage: `http://www.cs.jhu.edu/~mdinitz/classes/AGT/Spring2020/`
Online discussion: `http://piazza.com/jhu/spring2020/601436636/home`
Homework submission and grading: Gradescope (`http://www.gradescope.com`). Use entry code 9BRPX3.

Course Objectives

(1) Students will learn the basic definitions of equilibria and game theory
(2) Students will learn basic algorithmic tools used to compute equilibria.
(3) Students will learn about the (in)efficiency of equilibria
(4) Students will learn the basics of algorithmic mechanism design, including incentive-compatibility and the limits that places on optimality.

Course Topics

- Basic definitions of equilibria
- Algorithms for computing equilibria
- Efficiency of equilibria
- Algorithmic mechanism design

Course Expectations & Grading
There will be homework assignments approximately every other week. There will also be a final project. Class participation is also required.

Homeworks: 60%
Final Project: 30%
Participation: 10%

This class will be graded on a curve, but not a strict one. That is, the correspondence between numeric and letter grades will be determined by the final distribution of numeric grades, but there is no specific letter grade distribution that will be targeted.
You are free to work on the homework in groups of up to 3, but you must write up your solutions entirely on your own. That is, collaboration is limited to discussing the problem, and does not include writing down the solution. Please list the members of your group on your submission.

Assignments & Readings
These will be posted on the course webpage.

Ethics
The strength of the university depends on academic and personal integrity. In this course, you must be honest and truthful, abiding by the *Computer Science Academic Integrity Policy*:

> Cheating is wrong. Cheating hurts our community by undermining academic integrity, creating mistrust, and fostering unfair competition. The university will punish cheaters with failure on an assignment, failure in a course, permanent transcript notation, suspension, and/or expulsion. Offenses may be reported to medical, law or other professional or graduate schools when a cheater applies.

Violations can include cheating on exams, plagiarism, reuse of assignments without permission, improper use of the Internet and electronic devices, unauthorized collaboration, alteration of graded assignments, forgery and falsification, lying, facilitating academic dishonesty, and unfair competition. Ignorance of these rules is not an excuse.

Academic honesty is required in all work you submit to be graded. Except where the instructor specifies group work, you must solve all homework and programming assignments without the help of others. For example, you must not look at anyone else.s solutions (including program code) to your homework problems. However, you may discuss assignment specifications (not solutions) with others to be sure you understand what is required by the assignment.

If your instructor permits using fragments of source code from outside sources, such as your textbook or on-line resources, you must properly cite the source. Not citing it constitutes plagiarism. Similarly, your group projects must list everyone who participated.

Falsifying program output or results is prohibited.

Your instructor is free to override parts of this policy for particular assignments. To protect yourself: (1) Ask the instructor if you are not sure what is permissible. (2) Seek help from the instructor, TA or CAs, as you are always encouraged to do, rather than from other students. (3) Cite any questionable sources of help you may have received.

On every exam, you will sign the following pledge: "I agree to complete this exam without unauthorized assistance from any person, materials or device. [Signed and dated]". Your course instructors will let you know where to find copies of old exams, if they are available.

You can find more information about university misconduct policies on the web at these sites:

- Undergraduates: `e-catalog.jhu.edu/undergrad-students/student-life-policies/`
- Graduate students: `e-catalog.jhu.edu/grad-students/graduate-specific-policies/`

Students with Disabilities

Any student with a disability who may need accommodations in this class must obtain an accommodation letter from Student Disability Services, 385 Garland, (410) 516–4720, `studentdisabilityservices@jhu.edu`.

ABET Outcomes

- An ability to apply knowledge of computing and mathematics appropriate to the discipline (a)
- An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution (b)
- An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs (c)
- An ability to function effectively on teams to accomplish a common goal (d)
- An ability to use current techniques, skills, and tools necessary for computing practice (i)
- An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices (j)