COMP 7/8282 Evolutionary Computation Spring 2020

Instructor: Dr. Dipankar Dasgupta

Class Schedule: Monday –FIT 324 Time: 4:20PM - 7:20PM (Monday)

Contact Information:

Office: 333 Dunn Hall	**Department Office:** 209 Dunn Hall
Phone: 678-4147	**Department Phone:** 678-5465
E-mail: dasgupta@memphis.edu	

Office Hours:

Monday	Tuesday	Wednesday	Thursday	Friday
2PM – 3PM		2PM – 3PM		
*By appointment and participation in **eCoursewere** discussion board* ***(https://elearn.memphis.edu)***				

COMP 7/8282: Course Description
Computational implementation of biological analogies, such as genetic algorithms, genetic programming, evolutionary strategies, engineering; representation, fitness functions, fitness landscapes, automatically defined functions; applications to optimization, machine learning, software development.

PREREQUISITE: COMP 6601 or permission of instructor. Knowledge of Data Structures and Programming languages will be required.

Course Work & Assignments

Evolution by natural selection is one of the most compelling themes of modern science, and it has revolutionized the way we think about biological systems. In this course, we will study a form of evolution that can take place in a computer (i.e. evolvable systems). Evolutionary Computation (EC) techniques are general-purpose search and optimization procedures which are appealing because of their simplicity, extensibility and ease of interfacing. For last 30 years, ECs have been applied in growing numbers in every field of science and engineering including recent applications in Big Data Analysis, Deep Learning and other Machine Learning parameter optimization.

In this course, there will be a final project which involves implementing EC algorithms on a problem of your own choice. The course is intended to serve the needs of students, engineers, designers, developers, researchers in any scientific discipline who are interested in applications of evolutionary techniques.

I'll introduce evolutionary algorithms and related strategies in first few weeks of the course. After that there will be a large number of assigned readings. Students are expected to participate in class discussions and should plan on reading and reviewing carefully (2 to 3 times) scientific EC papers along with the selected readings during the semester.
In addition, there will be a substantial work needed to be done on the term project. It is anticipated that the final project will involve implementing an evolutionary computation technique and testing it

out on a problem. In addition to the program, you will have to submit a report describing the program, the problem on which you tested it, and the results you achieved. Your project grade will depend on the quality of the written report and demo as well as the quality of the project. Each group (of 2 students) needs to submit the **project proposal** on or before **February 24th, 2020 (Monday)**. The proposal should be (at least) one page long explaining the problem that you would like to solve using EC techniques (should have three paragraphs – objective, problem statement, model or software to be used, and at least 5 references).

Suggested Textbooks:

- ***Introduction to Evolutionary Computing*, A. E Eiben and J. E. Smith, Second Printing, Springer, 2007.**
- ***Evolutionary Optimization Algorithms* 1st Edition by Dan Simon, Apr 29, 2013**
- ***Evolutionary Algorithms in Engineering Applications,* Editors: Dipankar Dasgupta and Zbigniew Michalewicz, Springer-Verlag, 1997.**

Reference Books:

- ***Genetic Algorithms in Search, Optimization, and Machine Learning* by David Goldberg, Addision Wesley, 1989.**
- ***Handbook of Evolutionary Computation* , Editors: Back, Fogal, Michalewicz, Oxford University Press, 1997.**
- ***Nature-Inspired Optimization Algorithms* – by Xin-She Yang (Author), June 30, 2016**
- ***The Design of Innovation,* David Goldberg, Kluwer Academic, 2002.**
- ***Genetic and Evolutionary Computing (GECCO): Proceedings of the Eleventh International Conference on Genetic and Evolutionary Computing.***

Evaluation:

Students are expected to participate in class discussions. Participation in class will be viewed as a continuous two-sided feedback process, which (a) allow students to assess themselves on their progress in learning the material/understanding the security issues; and (b) allows the instructor to assess how well he is fostering the communication process with and among students. Good evaluations will thus reflect not only your grasp of the material, but also how well you take advantage of class time and how well you end up communicating with other people (class participants) about the course material. The evaluation process will include paper presentations, assignments, tests, quizzes, and a term project.

Grading

Your final grade for the course will be based on the grades in the following course-related activities (given in percentages):

Class performance	10%
Paper Presentations/Discussant/Reviews	20%
Tests/quizzes/Exam	30%
Assignments/Lab Exercises	10%
Term project + proposal	30%

Grading Scale:

A+	95.1-100	**B+**	85.1-88	**C+**	76.1-79	**D+**	60.1-66
A	90.1 -95	**B**	82.1-85	**C**	70.1-76	**D**	50 - 60
A-	88.1 -90	**B-**	79.1-82	**C-**	66.1-70	**F**	< 50

Course Policies:

Students are expected to attend all scheduled classes and submit assignments on time. If you miss a class, it is your responsibility to obtain the notes for the missed class from another student, check course website and catch up on the course content. There will be no make up test for this course.

Plagiarism/Cheating Policy:

Plagiarism or cheating behavior in any form is unethical and detrimental to proper education and ***will not be tolerated.*** All work submitted by a student (projects, programming assignments, lab assignments, quizzes, tests, etc.) is expected to be a student's own work. The plagiarism is incurred when any part of anybody else's work is passed as your own (no proper credit is listed to the sources in your own work) so the reader is led to believe it is therefore your own effort. Students are allowed and encouraged to discuss with each other and look up resources in the literature (including the internet) on their assignments, but ***appropriate references must be included for the materials consulted****,* and appropriate citations made when the material is taken verbatim. **Ethical behavior** is an important part of this course.

If plagiarism or cheating occurs, the student will receive a failing grade on the assignment and (at the instructor's discretion) a failing grade in the course. The course instructor may also decide to forward the incident to the University Judicial Affairs Office for further disciplinary action. For further information on U of M code of student conduct and academic discipline procedures, please refer to: **http://www.memphis.edu/studentconduct/pdfs/csrr.pdf** and the student handbook and planner.

"Your written work may be submitted to Turnitin.com, or a similar electronic detection method, for an evaluation of the originality of your ideas and proper use and attribution of sources. As part of this process, you may be required to submit electronic as well as hard copies of your written work, or be given other instructions to follow. By taking this course, you agree that all assignments may undergo this review process and that the assignment may be included as a source document in Turnitin.com's restricted access database solely for the purpose of detecting plagiarism in such documents. Any assignment not submitted according to the procedures given by the instructor may be penalized or may not be accepted at all." (Office of Legal Counsel, October 17, 2005).

<u>Tentative schedule</u> *(topics to be covered and other course-related activities during the semester)*:

WEEKS	LECTURE TOPICS	
WEEK 1	Introduction and Course Aims and Agenda.	
	Overview of EC – History, Variants, Operators, Application Domains.	
WEEK 2	GA Theories – Schema Analysis, Representation, Search Space and Problem Space. GA Operators and Selection Schemes.	
WEEK 3	Structured GA – Model and applications. Fitness Landscape, Correlation, Deceptive Functions, NFL Theorem.	
WEEK 4	Multimodal Functions – Niching Techniques	(P1, P2)
	Parallel GAs (PGA) – Fine & Coarse Grain Models, Island Model, Migration Strategies.	(P3, P4)
WEEK 5	Genetic Programming – Representation, Data Structure, ADF, etc.	
	Constrained GP, Linear GP, Advanced GP.	(P5, P6)
	Term Project Proposal submission (TBD)	
WEEK 6	Combination of GA and NN (COGANN) – Neurogenetic Learning and Design: Direct and Indirect Encoding,	(P7, P8)
	Mid-Term (week before the Spring Break)	
WEEK 7	Dominance, Messy GA. Grammar Rules, Cellular Encoding.	(P9, P10)
WEEK 8	Hybrid Approaches: EVO-Fuzzy	(P11, P12)
WEEK 9	**Project Progress Presentation/Report (TBD)**	
WEEK 10	Example Applications (in selected domain)	(Demo)
WEEK 11	Example Applications (in selected domain)	(Demo)
WEEK 12	Classifier Systems – Credit Assignment (Bucket Brigade), Zeroth-level Classifier (ZLC).	
WEEK 13	Advanced topics – Variable length GAs, Linkage Learning, Evolvable Hardware. **Project Demo and <u>Report Submission (May 4th).</u>**	

NOTE 1: Each project should be demonstrated to the instructor on **the last week of classes**. It will not be possible to get any extensions for the project work, so please plan on finishing on or before the submission date. *Each group (with maximum of two) members & both members need to actively participate in Project Demo*.

NOTE 2: **Two** *paper presentations have been scheduled on each day of presentation, and each presenter will get 15 minutes (plus 5 minutes for question and answer). Px stands for paper number (number* ***x****) to be presented as per the list.*

NOTE 3: We will be using eCourseware for lecture notes, grades and some submissions. If I need to communicate with the class as a group, I'll be using elearn. You may need to check your email regularly.