Version: Feb. 9_2005

56:238 Evolutionary Computation

http://css.engineering.uiowa.edu/~ie238/

Spring 2005

Objectives: Genetic algorithms, genetic programming, evolutionary strategies, evolutionary algorithms, evolutionary optimization, evolutionary neural networks, learning classifier systems, and development of evolutionary systems for applications in industry and medicine. Genetic programming formalisms and applications are emphasized. Case studies are integral part of the course.

Textbook: A.E. Eiben and J.E. Smith, *Introduction to Evolutionary Computing*, Springer, Heilderberg, Germany, 2003.

References:
T. Back, D.B. Fogel, and Z. Michalewicz (Eds), *Evolutionary Computation: Basic Algorithms and Operators*, Vol. 1 and Vol. 2, Institute of Physics Publishing, Philadelphia, PA, 2000.
Y.C. Jin (Ed.), *Knowledge Incorporation in Evolutionary Computation*, Springer, New York, 2005.
J. R. Koza *et al*., *Genetic Programming IV*, Kluwer, Norwell, MA, 2003.
R. Riolo and B. Worzel (Eds), *Genetic Programming Theory and Practice*, Norwell, MA, 2003.
R. Sarker and M. Mohammadian, and X. Yao (Eds), *Evolutionary Optimization*, Kluwer, Norwell, MA, 2002.
C.A. Coello Coello, D. A. Van Veldhuitzen, and G. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer, New York, 2002.
M.V. Butz, *Anticipatory Learning Classifier Systems*, Kluwer, Norwell, MA, 2002.
W. Banzhaf, P. Nordin, R.E. Keller, and Frank D. Framcone, *Genetic Programming*, Morgan Kaufmann, San Francisco, CA, 1998.

Websites:

http://www.red3d.com/cwr/evolve.html	Evolutionary Computation and its application to art and design
http://www.cs.umd.edu/users/seanl/gp/	Evolutionary Computation
http://www.genetic-programming.com/	Genetic Programming
http://alphard.ethz.ch/gerber/approx/default.html	Genetic Programming Applet

Journals:
Genetic Programming and Evolvable Machines (On line Journal available at UI)
http://www.kluweronline.com/issn/1389-2576/contents
IEEE Transactions on Evolutionary Computation (available at IEEE Explore http://ieeexplore.ieee.org/Xplore/)
Evolutionary Computation
Search also Elsevier Science Direct http://www.sciencedirect.com/science

Instructor:
Andrew Kusiak
2139 Seamans Center
andrew-kusiak@uiowa.edu
http://www.icaen.uiowa.edu/~ankusiak
Tel: 319 - 335 5934
Fax: 319 - 335 5669

Instructor Office Hours:

10:45 – noon, TTh and 5:00 - 6:00 pm, Th
2139 Seamans Center

Class Time: 6:00 - 8:30, Th

Classroom: 3026 Seamans Center

Course Contents:

1. Introduction
2. Machine learning and evolutionary computation
3. Genetic programming and biology
4. Genetic programming formalisms
5. Fundamentals of genetic programming
6. Statistics in genetic programming
7. Applications of genetic programming
8. Genetic programming software
9. Evolutionary optimization
10. Evolutionary neural networks
11. Case studies

Final Exam (Date and time will be provided at http://www.registrar.uiowa.edu/exams/)

Course grading scheme

Homeworks	**20%**
Project	**40%**
Midterm Exam	**20%**
Final Exam	**20%**

Up to five homeworks will be assigned in this class. Homework will usually involve reading recent journal papers on an evolutionary computation topic and reporting the key ideas learned form the paper in class.

Project grading scheme

10% Project proposal
20% Project presentation
60% Project content
10% Attendance of all project presentations

Exams: Midterm and final will be given.

Dates to Remember:

Classroom presentations of project proposals:

- February 24

Classroom presentations of projects:

- April 26

Project reports due:

- April 26

Midterm:

- Week of March 7

All homeworks, exams, and project materials are to be emailed to *andrew-kusiak@uiowa.edu*. Follow the flowing format Your_Name_ 56_238_Homework_1 in naming a compressed folder containing your files, e.g., Smith_56_238_ Homework_1.

SEMESTER PROJECT

THE SEMEXTER PROJECT MAY TAKE ONE THE FOLLOWING THREE FORMS:

A. **Application Project** (Teams of two students are allowed)
You need to describe the problem considered for your project and propose a solution approach. Ideally, the project should be based on an existing application. The solution approach could be based on an existing freeware that could be found on the web.
Hint: To identify software (freeware) tool to be used for solving the selected problem (application) you may follow the following steps:

- Search the www.
- Identify an evolutionary computation tool (e.g., genetic programming algorithm, evolutionary optimizer) that is suitable for the selected problem.
- Get familiar with the tool.
- Prepare user's manual.
- Prepare Power Point presentation.
- Demonstrate the application of the tool for your problem in class.
- Prepare project report according to the format presented in this syllabus.

B. **Research Paper** (Teaming is not encouraged)
You may choose a specific problem, develop a model, and solve with an evolutionary computation approach. This type of project should survey the existing literature, identify and summarize a research problem, present existing methods for solving the problem, formulate a new solution approach, and report computational results.

C. **Software Development** Project (Teams of two students are allowed)
The student(s) will be responsible for the development of software for some of the algorithms discussed in class. The code should be written in a modern language, e.g., C, C++, Visual Basic, ASP and a user-friendly interface should be developed. Web implementation of the software is encouraged.

PROJECT REPORT FORMAT

The project report should be prepared on a word processor and should contain figures and tables that are necessary to make the report complete. Be concise in your writing and consult technical writing references as needed.

The term project report should be prepared in the following format:

A. **Application Project**

1. Introduction
2. Problem definition
3. Project goals
4. Model formulation
5. Solution approach
6. Computational study
7. Conclusions

B. **Research Paper**

1. Abstract (approximately 100 words)
2. Statement of the problem
3. Literature review
4. Existing models and solution approaches

5. Proposed model and/or solution approach
6. Examples
7. Conclusions

C. **Software Development Project**

1. Introduction
2. Algorithm description
3. User's manual
4. Example problems (2)
5. Computer code description

The developed software should run on the College of Engineering network.

REFERENCES ON TECHNICAL WRITING

[1] Hacker, D., *A Writer's Reference*, Bedford/St. Martin's, 1999.
[2] Markel, M., *Technical Communication*, Bedford/St. Martin's, 2001.

Semester Project Guidelines

Time estimate

It is expected that each student should spend not less than 60 hours on the project.

Project presentation

Each project proposal and project results are to be presented in class.

Project report

Each team (student) should submit the following items (one per project):

- Hard copy of the project report
- Project report file and your class presentation Power Point slides. For software development projects, submit also the source code, executable, and specify the computer hardware and software needed to run your program. Compress the folder with all files with Winzip and email the zipped folder to andrew-kusiak@uiowa.edu. Label each zipped folder with your name and course number, e.g., Smith_56_238_Project, and reference your email accordingly, e.g., RE: Smith_56_238_Project.