

Teaching Portfolio

Wang Liang

Department of Computer Science, UGA

Contents

- Teaching philosophy
- Example teaching syllabus
- Example grading rubric, with assignment
- Conclusion

Teaching philosophy

"If you would thoroughly know anything, teach it to others."

Tryon Edwards (1809-1894)

When I first read this quotation, I couldn't stop thinking about the essence of teaching. In this highly commercialized society, teaching that are of high quality gradually accumulate in the organizations which treat teaching as providing a service to the customers (students). Within the context of service providing, knowledge sharing becomes the basic of teaching. While on the other hand, it is stressed and required to take customers' needs into account and thus improve the quality and efficiency of knowledge sharing.

My lessons will be conducted in a delightful way. Students' concentration and their need to release the tension will be constantly watched and adjusted by amusing interruptions or pace changes to the knowledge conveying procedure. The purpose of stressing the delight in class is not just to make them laugh, but to improve the efficiency of learning.

In contrast to the flexibility in class conducting, the content of the class will be consistent. Students' specific needs or backgrounds will not affect the syllabus or the knowledge points in a single class, because changes at the time may threat the integrity of knowledge which should be well-designed before the class begins.

As for the content of the class, I will attempt to organize the knowledge in the application cases. Plus, some encouragements, motivations and philosophy of life will be incorporated, especially for relatively young students.

Example teaching syllabus

CSCI 9999 Computational Intelligence

Fall 2009

Syllabus

Instructor: Wang Liang
Office: 307A Boyd Graduate Studies Research Center
Office Tel: (706) 000-0911
Office hour: Tuesdays 11am-12pm or by appointment
Email: kavenvih@gmail.com

Textbook: *Computational Intelligence - Concepts to Implementations* by Eberhart & Shi

References:

1. *Introduction to Genetic Algorithms* by Melanie Mitchell
2. *Handbook of Genetic Algorithms* by Davis
3. *Machine Learning* by Tom Mitchell

Course Description: In this course we will study the techniques of computational intelligence, especially evolutionary computation and neural networks. We will begin with introductory discussions of how the techniques function in solving problems in the real world. Then we will move on to the hybrid of multiple techniques and how to choose the appropriate techniques for the problems that you want to solve. Topics that will be covered in this course are as follows:

1. Fundamentals of evolutionary computation techniques.
2. Design and analysis of Genetic Algorithms.
3. Design and analysis of Particle Swarm Optimization.
4. Fundamentals of neural networks.
5. Advanced neural network architectures.
6. Applications of evolutionary computation techniques.

Some programming is required in this course. The programming projects will be done in whatever programming language you like.

Grading

Assignments	55% (homework, reports, projects, and presentations)
Midterm Exam	20% (around Oct. 02, 2009)
Final Exam	25% (around Dec. 11, 2009)

Disability

If there is any student who feels s/he may need an accommodation based on the impact of a disability, please contact me privately to discuss your specific needs.

Policies

Each student is expected to do his/her own work. Any evidence of academic dishonesty will not be tolerated, and will be subject to disciplinary action. Students are expected to familiarize

themselves with the academic honesty policy of the Department of Computer Science (attached) and the University of Georgia as stated in the following document:
http://www.uga.edu/honesty/ahpd/culture_honesty.htm

Computer Science
Departmental Policy Statement
Academic Honesty

The Computer Science Department recognizes honesty and integrity as necessary to the academic function of the University. Therefore all students are reminded that the CS faculty requires compliance with the conduct regulations found in the University of Georgia Student Handbook. Academic honesty means that any work you submit is your own work.

Common forms of academic dishonesty against which students should guard are:

1. Copying from another student's test paper or laboratory report, or allowing another student to copy from you;
2. Fabricating data (computer, statistical) for an assignment;
3. Helping another student to write a laboratory report or computer software code that the student will present as his own work, or accepting such help and presenting the work as your own;
4. Turning in material from a public source such as a book or the Internet as your own work.

Three steps to help prevent academic dishonesty are:

1. Familiarize yourself with the regulations.
2. If you have any doubt about what constitutes academic dishonesty, ask your instructor or a staff member at the Office of Judicial Programs.
3. Refuse to assist students who want to cheat.

All faculty, staff and students are encouraged to report all suspected cases of academic dishonesty. All cases of suspected academic dishonesty (cheating) will be referred to the Office of Judicial Programs. Penalties imposed by the Office of Judicial Programs may include a failing grade in the course and a notation on the student's transcript. Repeated violations are punishable by expulsion from the University. For further information please refer to the UGA Code of Conduct, available at the URL below.

<http://www.uga.edu/deanofstudents/judicial/downloads/conduct0304.doc>

Example grading rubric, with assignment

Assignment One Grading Rubric

1. *Explain why the order in which items are listed in the representation is unimportant for the naïve approach to the knapsack problem, but makes a big difference if we use the decoder approach.*

- a) In naïve representation, every bit in the genotype is translated into the phenotype, so the order is unimportant. – 2 Points
- b) Basic illustration of decoding scheme – 6 Points
- c) In representation with decoding, for the same genotype, different item order may be decoded into different phenotypes. So the order is important – 2 Points

If some statement is ambiguous but related to a key point from the above, half score will be given.

2. *Find a problem where EAs would certainly perform very poorly compared to alternative approaches. Explain why you expect this to be the case.*

This is an open-ended question. Many answers deserve full score.

In some cases, 1 point is taken if someone doesn't offer an alternate approach in solving the specific problems in which GA performs poorly.

3.

[20 points] The $subset_{21}$ problem is stated as follows. Given a set of N positive integers $X = \{x_1, x_2, \dots, x_n\}$. Find a subset P of the set X such that the sum of the elements of P is equal to 21. For example, if $N=5$ and the set $X = \{12, 17, 3, 24, 6\}$, the set $P = \{12, 3, 6\}$ is a valid solution for the $subset_{21}$ problem in this example.

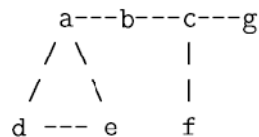
Formulate the $subset_{21}$ problem as a Genetic Algorithm optimization. You may use binary representation, OR any representation that you think is more appropriate. you should specify:

- A fitness function. Give 3 examples of individuals and their fitness values if you are solving the above example (i.e. $X = \{12, 17, 3, 24, 6\}$).
 - A set of mutation and/or crossover and/or repair operators. Intelligent operators that are suitable for this particular domain will earn more credit.
 - A termination criterion for the Genetic Algorithm optimization which insures that you terminate with a valid solution for the $subset_{21}$ problem if a solution exists.
- a) Fitness function: many answers are appropriate. (10 points). But if someone chooses the fitness function without taking the absolute value, such as (21-SumOfSubset), 4 points will be taken because GA can not work in that way. And if the illustrative examples are not given, 1 point off.
 - b) Mutation or Crossover operations: any operations suitable for integer representation can earn full credit. (5 points)
 - c) Termination Criteria: should be coherent with the fitness function and guarantee a valid solution when terminated. (5 points).

4.

[20 points] The **graph k-coloring** problem is stated as follows: Given an undirected graph $G = (V, E)$ with N vertices and M edges and an integer k . Assign to each vertex v in V a color $c(v)$ such that $1 \leq c(v) \leq k$ and $c(u) \neq c(v)$ for every edge (u, v) in E . In other words you want to color each vertex with one of the k colors you have and no two adjacent vertices can have the same color.

For example, the following graph can be 3-colored using the following color assignments: $a=1, b=2, c=1, d=2, e=3, f=2, g=3$



Formulate the **graph k-coloring** problem as an evolutionary optimization. You may use a vector of integer representation, OR any representation that you think is more appropriate. you should specify:

- A representation.
 - A fitness function. Give 3 examples of individuals and their fitness values if you are solving the above example.
 - A set of mutation and/or crossover and/or repair operators. Intelligent operators that are suitable for this particular domain will earn more credit.
 - A termination criterion for the Genetic Algorithm optimization which insures that you terminate with a valid solution for the **graph k-coloring** problem if a solution exists.
- a) Representation: integer or permutation representation (5 points)
 - b) Fitness function: number of conflicts or any other suitable fitness functions. (10 points). If the fitness function is not to be minimized or maximized, 4 points off. And if the illustrative examples are not given, 1 point off.
 - c) Mutation or Crossover operations: any operations appropriate for the selected representation will be acceptable. If the operations are not for the selected representation, 2 points off. (4 points)
 - d) Termination Criteria: should be coherent with the fitness function and guarantee a valid solution when terminated. (1 points).

Conclusion

From the very beginning of my first lecture, I have defined my teaching as a service provided to the students/customers. It is always an art to provide good service to the customers, since students typically have conflicting demands – they want knowledge as well as free time. So my approach is to make the teaching and learning as delightful as possible, and try to motivate the students instead of put pressure on them.