1st Annual CDIO Conference
Queen's University
Kingston, Ontario, Canada
June 7 to 8, 2005

ACTIVE LEARNING GAMES

Olivier L. de Weck[1], Il Yong Kim[2] and Rania Hassan[3]

Department of Aeronautics and Astronautics
Engineering Systems Division
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

ABSTRACT

This paper discusses active learning games as a potentially important pedagogical technique in support of formal classroom education. A brief review of the active learning literature is given, followed by a list of known active learning games relevant to the CDIO engineering education context.

As a specific example of an active learning game we present the learning objectives, rules, and implementation of a "Genetic Algorithm Game" that is used to introduce this class of evolutionary optimization algorithms to graduate students. Genetic algorithms do not require mathematically advanced formulations. Nevertheless, many students are experiencing conceptual difficulties in understanding the abstract nature of genetic operators and how the algorithm is able to successfully search complex design spaces for good solutions. We have found that playing the "Genetic Algorithm Game" during class is an effective tool that helps students experience and reinforce the inner workings of genetic algorithms. This activity enhances conceptual learning and initial student feedback has been very positive.

We believe that this example of an "in-class game" provides a template for similar active learning activities in other domains. Clearly, playing such games requires thought, preparation and time (typically between 10-30 minutes per game), as well as careful synchronization with the main lecture content. Faculty-student interactions are enhanced by assigning secondary materials for self-study and using the gained lecture time for active learning games that reinforce the primary materials and key concepts.

INTRODUCTION

We define active learning games as *"A pedagogical technique that uses playful in-class activities designed to actively engage students with key concepts, the faculty and each other".*

We will first review literature in the area of active learning and summarize some of its principles. Next, we present an overview of active learning games – those that are meant to be

[1] Assistant Professor, Department of Aeronautics and Astronautics, Engineering Systems Division, Room 33-410, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA, Telephone: (617) 253-0255, email: deweck@mit.edu – corresponding author, web: http://deweck.mit.edu
[2] Assistant Professor, Mechanical and Materials Engineering, Queen's University, Kingston, Ontario, K7L 3N6, Canada, email: iykim@me.queensu.ca , web: http://me.queensu.ca/people/iykim/index.asp
[3] Postdoctoral Associate, MIT, Room 33-412, email: rhassan@mit.edu

conducted during classroom lectures – as a specific pedagogical technique under the umbrella of active learning. Why do we believe that active learning games are effective? We report empirical evidence of the success of this method at MIT, where a number of different active learning techniques have been used (Fig.1). A more rigorous assessment of the effectiveness of active learning techniques in general and active learning games in particular would require a scientific experiment with control and test groups. Such an experiment is proposed in this paper.

Figure 1: Active learning in the windtunnel [11]

Active Learning Activity	Ref.	Time [min]
"Muddy" Cards	[9]	2-3
Concept Questions	[10]	5 ea.
PRS System Use	[11]	2
Hands-on Experiments	[12]	20-40
Recitation Q&A session	[11]	20-40
Active Learning Games	here	10-30

Table 1: Active learning techniques

Active learning is also described in CDIO's Standard 8:

CDIO™ Standard 8 — Active Learning
Teaching and learning based on active experiential learning methods

Description: Active learning methods engage students directly in thinking and problem solving activities. There is less emphasis on passive transmission of information, and more on engaging students in manipulating, applying, analyzing, and evaluating ideas. Active learning in lecture-based courses can include such methods as partner and small-group discussions, demonstrations, debates, concept questions, and feedback from students about what they are learning. Active learning is considered experiential when students take on roles that simulate professional engineering practice, for example, design-build projects, simulations, and case studies.

Rationale: Students remember less than a fourth of what they hear and only about half of what they see and hear. By engaging students in thinking about concepts, particularly new ideas, and requiring some kind of overt response, students not only learn more, they recognize for themselves what and how they learn. This process of metacognition helps to increase students' motivation to achieve program learning outcomes and form habits of lifelong learning. With active learning methods, instructors can help students make connections among key concepts and facilitate the application of this knowledge to new settings.

Evidence:
- successful implementation of active learning methods, documented, for example, by observation or self-report
- a majority of instructors using active learning methods
- high levels of student achievement of all CDIO learning outcomes
- high levels of student satisfaction with learning methods

Active learning games fall into this category, but have not yet been directly discussed as an active learning tool in the literature. This paper presents an example of such a game for teaching genetic algorithms to students in the area of engineering design and optimization. The paper discusses the learning objectives, rules, implementation and assessment of a "Genetic Algorithm Game" that is used to introduce this important class of evolutionary optimization algorithms. We first review the literature on active learning and give an overview of known active learning games. Next, a brief primer on genetic algorithms is given and the GA game instructions are presented as well as the supporting materials to execute this activity in the context of a graduate or undergraduate class.

Emergence of Active Learning

The engineering educational community has gradually embraced the possibilities of increasing student engagement and learning through methods that can together be described as "Active Learning". Though the term eludes precise definition, common themes that run amongst the many attempts to define it include the following: teaching that "puts the responsibility of organizing what is to be learned in the hands of the learners themselves" [1] and that challenges students to "engage in such higher-order thinking tasks as analysis, synthesis, and evaluation" [2] instead of passively absorbing material. In other words, active learning in the classroom is implemented through "instructional activities involving students in doing things and thinking about what they are doing" [2].

Engineering programs that have implemented active learning curricula agree that there are advantages to activating students instead of relying on the traditional lecture-only approach. Parcover and McCuen in their discussion of a constructivist learning method for teaching engineering design argue that learning "is not simply a matter of transferring information from lecturer to listener…conceptual development and comprehension requires the opportunity to question, explain, and test beliefs" [3]. Van Dijk and colleagues conducted an experimental study on the effectiveness of interactive lectures, using a control group [4]. They concluded that interactive lectures increase student motivation, but do not automatically result in more effective learning. It appears that interactive lectures should be combined with other activities inside and outside the classroom to amplify the learning impact. This point is emphasized by Bales' assertion that long-term retention of lecture material for the average student is only 5%. This rate, he argues, can be increased up to 75% for lectures incorporating practical exercises [5]. The literature is embryonic when it comes to specific examples of how active learning should be derived from higher-level curriculum goals and how it may be implemented in practice. MIT's active learning approach is embedded in a larger initiative for curriculum reform called CDIO (conceive-design-implement-operate). This reform, initiated by Crawley [6], changes the context of aerospace education from engineering science to lifecycle engineering. The contribution of this article is to demonstrate and discuss a promising implementation of active learning, with the example of the "Genetic Algorithm Game".

Overview of Active Learning Games

A number of activities have been proposed and are being used by various institutions and instructors that can be qualified as "active learning games". A flow diagram for a typical active learning game, embedded in a classroom lecture, is shown in Figure 2. Typically, such games are not played at the beginning of a lecture, but after the initial material and concepts have been presented. An active learning game will only be effective once the instructor has presented the key ideas and concepts. As such, the active learning game serves to reinforce the material, not to introduce it in the first place. The first step is to explain the rules of the game and what its objective is in reinforcing conceptual learning. Next, the logistics of the game are organized, which typically involves subdividing the students in the class into teams or groups and distributing any materials that might be required to play the game. Third, the game is actively played with the instructor serving as both a facilitator and referee. Classroom games typically involve several rounds or iterations, or require that one rotate through all teams at least once. Oftentimes, active learning games require students to stand up and report their results to the class or give a short oral narrative of their experience. We have found it to be effective that the instructor capture the key results and insights from playing the game on the blackboard, whiteboard or computer screen. The game concludes with a period of reflection and discussion and a summary of the game results, which allows transitioning the class back into lecture mode. Typical active learning games last for 10-30 minutes and usually a 90-minute lecture will allow for only one such activity. Placing such an activity in the middle of a lecture also helps in maintaining alertness and lengthens student's attention span. We have found it difficult to play active learning games in classes with more than 40 students or in classes with both local and distance education students present at the same time.

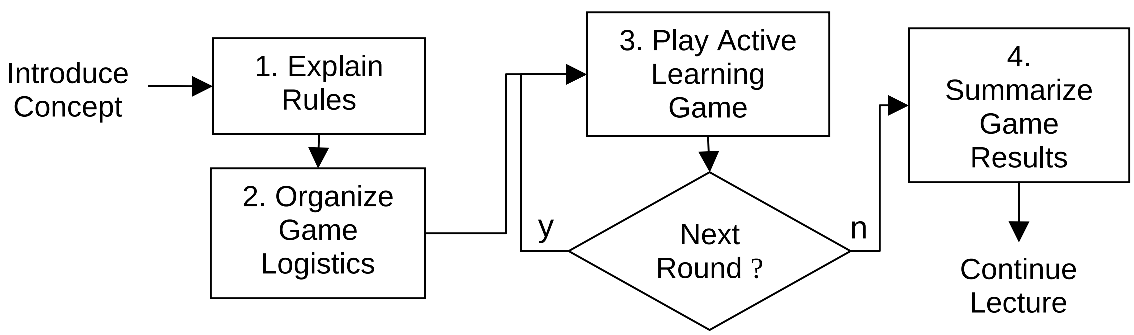

Figure 2: Typical flow of active learning games

Table 2 contains a list of active learning games known to the authors. For each game we provide the name under which it is known, the primary learning objective of the game, a brief summary of the activities that are conducted, the estimated duration as well as the name and institution of the instructor who invented or has practiced this game recently.

Game	Learning Objective	Activities	Time [min]	Instructor
"Genetic Algorithm Game"	Be able to explain how genetic operators work and how they interact during search.	Described in this paper.	10-30	Olivier de Weck (MIT)
"Men from Mars"	Understand the difference between form and function	Students form groups and receive a set of unusual objects, where the form is easily described, but the function is not obvious (e.g. bathtub stopper….). They discuss the object in groups and report out their guess of what the function is.	10-15	Edward Crawley (MIT)

"Jigsaw Method"	Consider a complex business or engineering problem from a variety of perspectives	Form teams and assign them a particular function inside the company (marketing, engineering, manufacturing, finance,) after some amount of work/discussion re-shuffle members into new teams and repeat.	15-30	Tim Simpson (Penn State) [13]
"Beer Game"	To clarify the advantages of taking an integrated approach to managing the supply chain; it particularly demonstrates the value of sharing information across the various supply chain components	Considers a simplified beer supply chain, consisting of a single retailer, a single wholesaler, which supplies the retailer, a single distributor, which supplies the wholesaler, and a single factory with unlimited raw materials, which makes (brews) the beer and supplies the distributor. Each component in the supply chain has unlimited storage capacity, and there is a fixed supply lead-time and order delay time between each component.	>20[4]	David-Simchi-Levi (MIT) [12]
Multiobjective City Planning Game	Visualize how emphasizing different objectives will lead to different solutions in system design	A hypothetical 5x5 block city is to be designed by defining zones (residential, commercial, and recreational) and roadways. Students form stakeholder groups and create configurations that emphasize competing objectives (minimize commuting time, maximize tax revenue…) and then compare answers and rationale.	45-60	Olivier de Weck (MIT)

Table 2: List of active learning games known to the authors

The following sections explain the details of the Genetic Algorithm Game as a particular example of an active learning (in-class) game.

GENETIC ALGORITHM GAME

GA Primer

Genetic algorithms (GAs) are a search method that is inspired by genetics and natural selection. While applicable to many optimization problems, we teach this algorithm in the context of a class on multidisciplinary design optimization [14]. The design variables (phenotype) are encoded as a string of numbers (typically in binary code), i.e. the genotype is also called a chromosome. Candidate designs are expressed by a set of chromosomes, each representing a member of a population of designs. Starting from a population of randomly generated chromosomes, they evolve into better designs over generations, going through the operations of crossover, mutation, and selection.

[4] The beer game can be played in-class with multiple players, or it can be done as an off-line activity.

GAs were first developed by Holland in1975 [7], refined by Goldberg in 1989 [8], who also outlined fundamental theory and applications. The premises of GA are as follows:

- Natural Selection is a very successful organizing principle for optimizing individuals and populations of individuals in nature.
- If we can mimic natural selection, then we will be able to optimize more successfully.
- A possible design of a system – as represented by its design vector **x** - can be considered as an individual who is fighting for survival within a larger population.
- Only the fittest survive – Fitness is assessed via an objective function

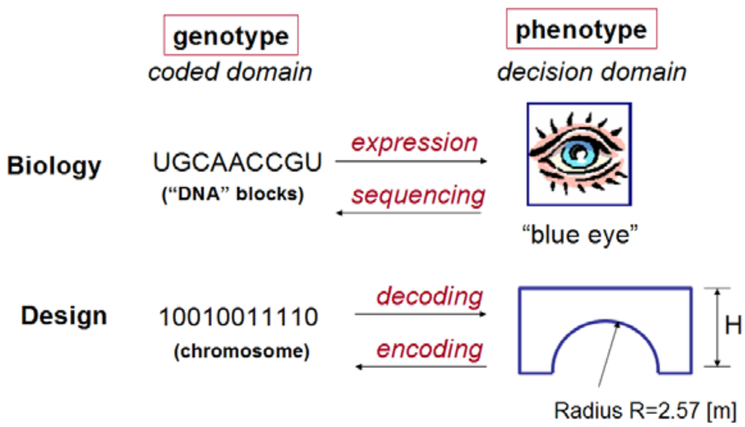

Fig. 3: Genotype and phenotype in GA.

In GAs, a design is expressed by a chromosome in the genotype space, which is obtained by encoding design information from the phenotype space. The encoding scheme is provided by a user, and at this stage, the chromosome length is determined depending on the desired resolution of a design variable. Figure 3 compares a chromosome in the genotype space and a design in the phenotype space for biology and engineering design. Each design has a performance that reflects its fitness within a population.

A typical flow chart for a GA is shown in Fig. 4. An initial population of a prescribed size is randomly generated in the first step. Among the design candidates in the initial population, those designs whose fitness is relatively high are selected as parents for next step. The selection can be done based on the fitness ranking among designs. Other methods, such as the roulette wheel selection and the tournament selection method, are also used. It is important to do selection such that diversity in the population is preserved.

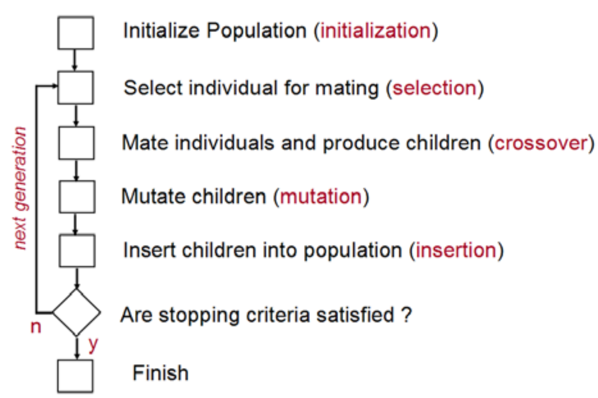

Fig. 4: Flowchart of a GA showing genetic operators

The selected chromosomes are then mated, producing offspring. This mating, called crossover in GAs, is achieved by exchanging some part (s) of a chromosome with its partner. The crossover for mating may be made at one point or at multiple points. Figure 5 shows a single point crossover. The position of the crossover point is selected randomly.

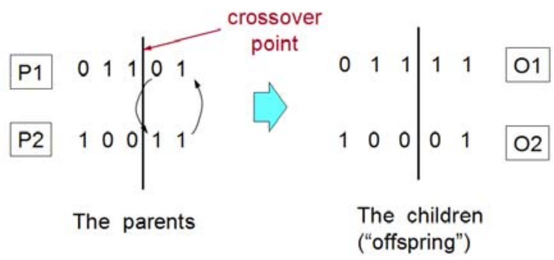

Fig. 5: Single point crossover: parents P1 and P2 on the left, offspring O1 and O2 on the right.

Mutation modifies the children chromosomes to restore diversity. Some bits in chromosomes are flipped in this step. Too little mutation leads to an impoverished genetic pool with increasing number of generations. In this case, it may converge too early, obtaining only local optima. On the other hand, too much mutation decreases the convergence rate and undermines the fitness-based selection bias.

By inserting new children into the population, one generation is completed. It has been shown that repeating this procedure can lead to improved mean population fitness and finding the chromosome (or sets of chromosomes) that will maximize the objective function. If the algorithm satisfies the termination criterion, it will finish; otherwise it goes back to the selection operator and repeat the process for more generations. GAs, like other heuristics search methods, do not have clear, obvious termination criteria, but there are some options for termination criteria:

- A prescribed number of generations has been completed - typically O(100)
- The standard deviation in performance of individuals in the population falls below a threshold $\sigma_J<\varepsilon$
- Stagnation: no or small marginal improvement from one generation to the next
- A particular point in the search space is encountered

The GA searches a population of points in parallel, not only a single point, and probabilistic transition rules, not deterministic ones, are used. Derivative information is not required.

GA Game Procedures

A game was created to enhance conceptual learning about GA's in an undergraduate or graduate engineering classroom setting. Typically 20-40 students will play this game at the same time. Each student represents one member in a hypothetical population that is to be optimized. The genetic algorithm game is managed by a game master (instructor) and a facilitator (teaching assistant). Each student is given one initial population paper chromosome (Fig. 6) and a set of blank paper chromosomes (Fig. 7) for use in future generations. Pairs of dice and pencils are also given. The fitness function is the sum of the digits of each chromosome (F), which will be a number between 0 and 12, e.g. fitness F = 7 for the first chromosome in Fig. 6.

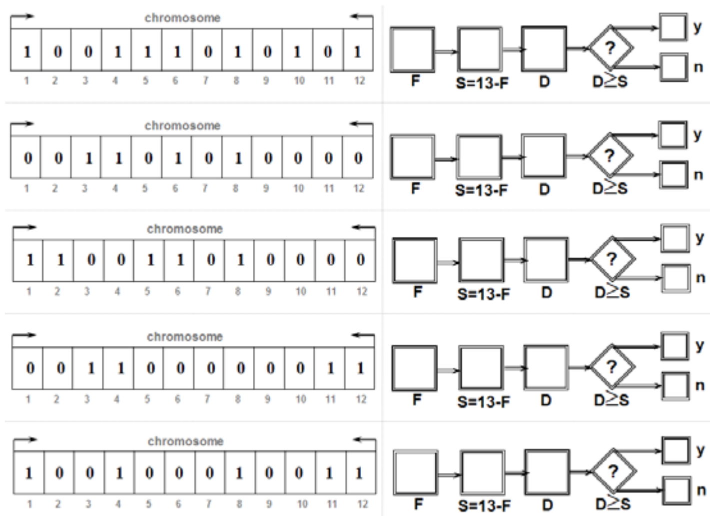

Fig. 6: Samples of initial population chromosomes

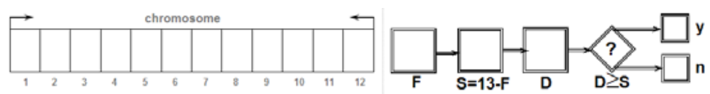

Fig. 7: Blank paper chromosome

Students then go through the steps of the genetic algorithm flowchart (Fig.4) by rolling dice, splicing chromosomes, exchanging parts of chromosomes with other students in the class and recombining and transcribing new chromosomes. After each generation each student in the class calls out their current fitness level (a number between 0 and 12) and the instructor keeps track of minimum, mean and maximum fitness levels across the class on the blackboard. Detailed GA game instructions are given in Appendix A for instructors wishing to replicate this activity at their own institution. Figure 8 (left) shows the fitness distribution of the initial population (40 individuals).

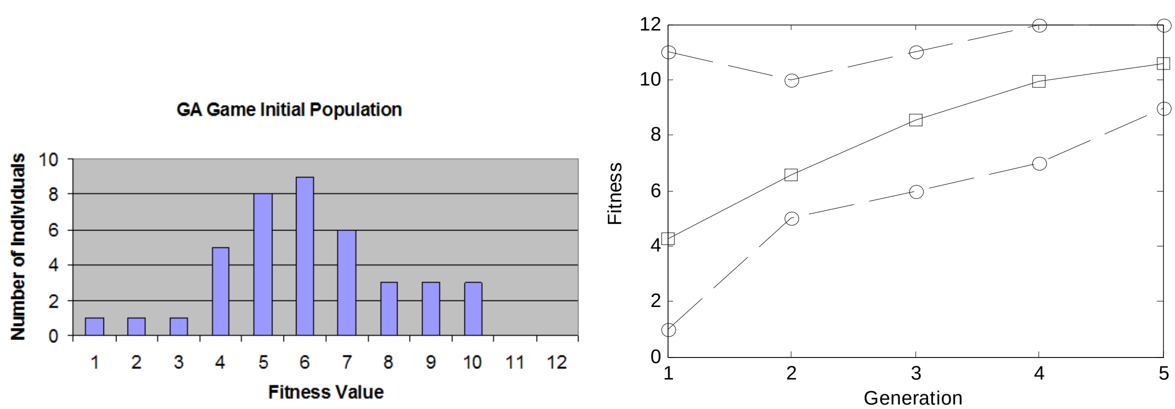

Fig. 8: (left) Fitness distribution of initial population (40 individuals, normal distribution, μ=6, σ=1.5), (right) Simulated progression of GA game (40 students) with mean shown as the solid line and the maximum fitness and minimum fitness shown as the dashed lines, respectively.

Generally, it is observed that the average fitness of the class population tends to increase, see Fig. 8 (right). While it is desirable to play 5 rounds of the game as shown in the figure, it is generally sufficient to play 3 rounds to see the trend of increasing fitness. Five rounds are necessary to see the decreasing rate of improvement from one generation to the next. Each rounds takes about 5 minutes to complete. We have been playing this game – among others - for the last 4 years as part of our MDO class at MIT [14] and students confirm that this leaves a memorable impression on them. The active learning games have also been explicitly mentioned in the end-of-semester surveys. Nevertheless, we acknowledge the necessity to measure the effectiveness of this active learning approach in a scientific experiment. This could be accomplished, for example, by comparing a student group who receives a lecture on genetic algorithms and plays the GA game with a control group who only follows a traditional lecture. Describing this experiment is the subject of the next section. We have obtained approval for this experiment and results will be contained in a future paper.

Pedagogical Experiment Setup

It was decided to design a controlled experiment to assess the effectiveness of the GA game activity relative to a lecture-only format. The test protocol is summarized as follows:

<u>Purpose of the Study</u>

Research in Engineering Education has recently centered around methods that collectively are termed "Active Learning". However, there is debate as to what types of active learning padagogies and activities yield the most benefit, particularly in imparting conceptual understanding of complex topics. Active learning games explore and exercise key concepts during class and are intended to reinforce the theory, while providing a welcome change of pace during lecture. This study will attempt to quantitatively measure the incremental learning benefit of such games, relative to a lecture-only format. The study uses "Genetic Algorithms (GA)", a relatively recent class of computational optimization algorithms, as the topic of the lecture. The hope is that the insights regarding the effectiveness of "active learning games" will eventually be generalizable to other domains. Despite their name, "genetic algorithms" do not involve any type of medical/biological procedures or manipulation of genetic material of the student test subjects.

<u>Study Protocol</u>

<u>A. Experimental procedures:</u>

The study subjects are divided into two groups: a control group and a test group. Both groups are homogenized in terms of gender, age, undergraduate or graduate study year and GPA. Both groups will listen to a one-hour lecture on Genetic Algorithms. This lecture is identical for both groups. In addition to the lecture, the test group will play the GA game during lecture, whereby the faculty and staff will act as facilitators. At the end of the lecture both groups will take a multiple-choice quiz (see Appendix B) to measure their conceptual understanding of "genetic algorithms". The quiz results will be processed statistically and results will be compiled. We will be looking for statistically significant differences in the test scores and attempt to quantify whether or not the GA game has had a beneficial impact on conceptual learning.

<u>B. Type and number of subjects involved:</u>

Each of the groups will contain roughly 30 test subjects (60 test subjects total for the study). The test subjects will be undergraduate students (sophomores, juniors, seniors) and graduate students at MIT. The number of 30 per group is driven primarily by the need to have a statistically representative sample size.

The actual time requirement for the test subjects will be approximately 90 minutes/person for subjects in the control group and 120 minutes/person for subjects in the test group. Before the start of the lecture, each student in each of the two groups will be explained the purpose of the study and the research procedure. They will then each sign a consent form if they still agree to participate.

C. Procedures to ensure confidentiality:
Each student will fill in the quiz after lecture (15 minutes) and mark his or her name on the quiz response form. It is necessary for us to correlate the study subject names with the test scores to check for bias and confounding factors. In the statistical evaluation of the effectiveness of the GA game and subsequent publication all test data will be averaged. Students will not be allowed to see each other's test scores. The test subject's performance on this quiz will have no impact on their regular academic performance or GPA.

We will report results of this experiment in future work.

CONCLUSIONS

This paper discusses active learning games as a potentially important pedagogical technique in support of formal classroom education. A brief review of the active learning literature was given, followed by a list of known active learning games relevant to the CDIO engineering education context. As a specific example of an active learning game we presented the learning objectives, rules, and implementation of a "Genetic Algorithm Game" that is used to introduce this class of evolutionary optimization algorithms to graduate students.

Active learning games, played during lectures, have two primary benefits. First, they allow the students to reinforce their understanding of one or two key concepts by participating in an active learning activity, usually involving the faculty and their peers. Second, such games represent a welcome break in the class from the passive learning mode and help to lengthen attention span and engagement. Initial feedback to these games has been positive as revealed by end-of-semester surveys. Nevertheless, it must be pointed out that active learning games require careful planning and execution and they take substantial time away from traditional lecture time, typically on the order of 10-30 minutes per game. This requires that secondary materials be displaced from the lecture and assigned as reading materials for self-study. We have found that active learning games work well for classes with up to 40 individuals, but are not recommended for larger classes or for classes with a mix of local and distance-learning students where the latter group participates via videoconferencing.

Future work in active learning game research will focus on these three questions:

1. Measure the effectiveness of active learning games in a series of controlled scientific experiments by comparing homogenized test and control groups.

2. Attempt to explain the dynamics of active and passive learning modes that underpin the effectiveness of active learning games, e.g. using system dynamics models where the dynamic relationship between active learning, passive learning, the amount of lecture content and student's attention spans are related.

3. Compile a more comprehensive list of known and emerging active learning games (extension of Table 2) and compare their characteristics and domains of applicability.

REFERENCES

[1] Dodge B., "Active Learning on the Web", San Diego State University, URL: <http://edweb.sdsu.edu/people/bdodge/active/ActiveLearningk-12.html>, accessed August 7, 2003.

[2] Bonwell, Charles C. and Eison, James A., "Active Learning: creating excitement in the classroom," ASHE-ERIC Higher Education Report No. 1, *The George Washington University, School of Education and Human Development,* Washington, D.C.

[3] Parcover, J.A., and McCuen R., "Discovery Approach to Teaching Engineering Design," *Journal of Professional Issues in Engineering Education and Practice,* Vol. 121, No. 4, 1995, pp. 236-241.

[4] Van Dijk, L.A., Van Den Berg G.C., and Van Keulen H., "Interactive lectures in engineering education", *European Journal of Engineering Education*, Vol. 26, No. 1, 2001, pp. 15-28

[5] Bales, E., "Corporate universities versus traditional universities: friends or foes," Key Note Address, *Conference on Innovative Practices in Business Education*, Orlando, FL, 4-7 December, 1996.

[6] Crawley, E. F., Greitzer, E.M., Widnall, S.E., Hall, S.R., et. al., "Reform of the Aeronautics and Astronautics Curriculum at MIT", *ASEE Journal of Engineering Education*, Vol. 83, No. 1, pages 47-56, January 1994.

[7] Holland J., "Adaptation in natural and artificial systems", University of Michigan Press, 1975

[8] Goldberg, D.E.," Genetic Algorithms in Search, Optimization and Machine Learning", Addison Wesley, 1989

[9] Muddiest Point in the Lecture ("Muddy Cards"), URL: http://www.cdio.org/tools/ikit/ikit_tandl/reports/mudcards.pdf

[10] D. Darmofal, D. Soderholm and D. Brodeur, "Using Concept Maps and Concept Questions to Enhance Conceptual Understanding", *ASEE/IEEE Frontiers in Education Conference*, Boston, MA, USA, 06-09 November 2002.

[11] de Weck O. L., Young P.W. and Adams D., "The Three Principles of Powered Flight: An Active Learning Approach", Paper ASEE-2003-522, *2003 ASEE Annual Conference & Exposition*, Nashville, Tennessee, 22-25 June, 2003

[12] Web-based Beer Game: http://beergame.mit.edu/

[13] Jigsaw Method website: http://www.jigsaw.org/steps.htm

[14] Multidisciplinary System Design Optimization (16.888/ESD.77), OpenCourseWare, http://ocw.mit.edu

Appendix A

Genetic Algorithm - Game Instructions

Purpose: Illustrate the flow and operators of genetic algorithms with a hands-on game. The objective of the "optimization" is to maximize the number of "1""'s on each individual chromosome. The maximum fitness that can be achieved is 12. Typically, the class size has to be between 12 and 40 and 3-4 generations will be played. Duration: 20 min.

Materials: Initial population chromosomes, blank chromosomes, pencils, dice, blackboard and chalk, rotating pointer, calculator

Personnel:
- Game Master (1)
- Facilitators (1-2)
- Participants (12-40)

Steps:

1. START: Count the number of participants: N = population size
2. Hand out initial population chromosomes to the class – each person gets one
3. Hand out pairs of dice and pencils – one set per row
4. Hand out a number of blank chromosomes ca. 4-5 per person
5. Generation Count = GEN=1;
6. Fitness evaluation: Every individual adds up number of "1"'s in his/her chromosome; fills in the sum in the field marked "F".
7. One-by-one individuals call out the "F" score aloud. Facilitator records scores on blackboard or laptop and computes average score for the entire population. The facilitator also records the highest scoring individual. The facilitator announces the current average and maximum fitness to the class. Decide whether to continue the game. If yes goto step 8, else STOP.
8. Individuals calculate "S". Roll two dice and fill in result in field "D" (2-12). If D>=S, check "y", else check "n".
9. Selection: Everyone who has "n" is not selected for mating. These individuals have to hold up their chromosomes and they are collected by the facilitator and removed from the game. The number of non-selected individuals is NS < N.
10. The NS highest scoring individuals in the population make a one-to-one copy of their chromosome onto an empty chromosome.
11. Insertion: The NS duplicated chromosomes are collected by the facilitator and handed back out to the individuals who where not selected. Now everyone has a chromosome again.
12. The Game Master rolls 3 dice to determine the random crossover location: Xbit=ceil(sum of three dice score)/2; Should be a number between 2 and 9. Announce Xbit aloud.
13. Crossover: Individuals randomly select a new mating partner in their geographic neighborhood. Once a pair has found each other, they separate their chromosome by splitting it down the middle after Xbit and exchange the front and back substrings, respectively.
14. Copying: Individuals copy their new chromosome composed of the front and back substrings onto a blank chromosome.
15. Mutation: Game Master rotates the pointer. The individual who the pointer is aimed at has to flip the bit corresponding to his/her month of birth. GEN=GEN+1; Goto step 6.

GA Quiz **Appendix B**

Student Name _________________________ Time: 15 min

Group ___ Control Group (had lecture only)
___ Test Group (had lecture and played game)

A Quiz on Genetic Algorithms

Multiple Choice Questions: circle the correct answer

1) Does the basic idea of the GA as an optimization-based search method rely on:
   a. Competitive behavior of individuals
   b. Cooperative behavior of individuals
   c. Both

Answer: a

2) What is the main advantage (s) of using the GA over other search methods?

   a. It moves from a set of solutions to another set in one iteration rather than moving from a single point to another
   b. It can handle discrete design variables
   c. It can handle nonlinear objective and constraint functions
   d. All the above
   e. a and b
   f. b and c
   g. a and c

Answer: f

3) What is the first operator that takes place in a GA iteration:
   a. Crossover
   b. Selection
   c. Mutation
   d. It depends how the algorithm is coded
   e. Either a. or c. occur first

Answer: b

4) Which of the following is represented by the mutation operator in the GA:
   a. Survival of the fittest
   b. Reproduction or mating
   c. Promotion of diversity

Answer: c

5) A mutation rate of 0.001 has the following meaning:
   a. Every 1000^{th} individual of a population will be mutated on average
   b. The likelihood that any given bit will be flipped randomly is 10^{-3}

c. Every 1000th generation will be subject to mutation
d. Both a. and b. are correct
e. Both a. and c. are correct
f. Both b. and c. are correct

Answer: b

6) Which of the following is represented by the selection operator in the GA:
    a. Survival of the fittest
    b. Reproduction or mating
    c. Promotion of diversity

Answer: a

7) What is the name of the selection scheme that operates on pairs of two individual chromosomes at a time:
    a. Roulette wheel selection
    b. Twin fitness ranking
    c. Tournament selection
    d. Binary elitism
    e. Random selection

Answer: c

8) Which of the following is represented by the crossover operator in the GA:
    a. Survival of the fittest
    b. Reproduction or mating
    c. Promotion of diversity
    d. Switching from one population to another

Answer: b

9) Parent 1's chromosome is: [01001110], Parent 2's chromosome is: [10110011]. Which of the following chromosomes are valid children from parent 1 and 2, assuming single point crossover:
    a. [11001110]
    b. [01000010]
    c. [10111110]
    d. both a. and b.
    e. both b. and c.
    f. both a. and c.

Answer: f

10) When does a genetic algorithm typically stop:
    a. when the maximum number of generations has been reached
    b. when the global optimum is found
    c. when the average population fitness does not increase more than some predefined amount from one generation to the next
    d. when all chromosomes have converged and are identical
    e. a. or b.

f. b. and d.
g. a. or c.
h. all answers are correct

Answer: g

Text Questions:

11) Can optimality be proved when solving an optimization problem using the GA? Why or Why not?

Answer: No, optimality cannot be proved because Kuhn-Tucker Conditions cannot be shown., i.e. gradients are not used in the search process.

12) What enables the GA to handle discrete design variables?

Answer: Binary variable encoding to 0’s and 1’s.

13) How does the GA compare to other search methods in term of computational cost?

Answer: the GA is expensive because it is a population-based search method, i.e. moves from a number of points to a number of points in one iteration.

Open Ended Question:

14) What kind of application (s) you think you can apply the GA to optimize or design?

Answer: Non-linear, non-convex problems with discrete variables such as in structural topology optimization, aircraft and automotive design, design of biomedical devices, optimal packaging problems with non-symmetric boundary conditions, facility location problems with non-convex objective functions