Group Projects

The main objective of the group projects is to expose students to hands-on experiences of building intelligent agents (systems implementing Artificial Intelligence techniques for solving various problems) so that experimenting with such systems will gain students concrete insight and understanding about various AI ideas. Moreover, because students must form groups and each group works as a team, the project also trains the students’ capability for team work, communication and cooperation. The group project presentation session also provides the opportunity for training presentation and communication skills.

Required components of the group project:

It is required that each group project must involve an implementation component and an experimentation component, using any computer hardware platform and the software environment of choice by the group.

The implementation component:

The group must develop a computer program that implement the intelligent agent. The group does NOT have to write ALL programs from scratch in implementing an AI system. It is fine for a group to down-load open-source AI programs and to extend/customize/modify the program for various experiments in a particular application domain. For example, for a group project using Genetic Algorithm (GA) to discover gaming strategies of a particular (board or video) game, the group can incorporate any available open-source codes for GA in their system.

The experimentation component:

The experiments conducted in the project should be designed toward answering some key questions around a central theme. For example, consider a group project that implements the GRAPHPLAN algorithm together with an A*-like search method (for the solution extraction step) for planning robot navigation and problem-solving in an environment. Here the key questions to be answered could include: Which heuristic function for guiding the search is better in the sense that it helps generate plans efficiently - reducing the number of states explored? If a heuristic function is known to be not admissible, how was the quality of the plan generated by using such heuristic function in search? Is there a trade-off between admissibility of a heuristic function h and the efficiency in finding a plan with h? Experiments are needed to compare different heuristic functions in terms of efficiency of searching for a plan and the quality of the plans generated.

As another example, suppose a group project tries to compare two different approaches of using Neural Networks (NN) for hand-written character recognition: one approach uses the raw (gray-scale) image of the character as inputs to the NN, whereas the other approach uses 10 features (extracted from the raw image) as inputs to the NN. The key questions to be answered by the experiments include: Which approach is computationally more efficient? Which approach gives better classification accuracy? Among all the features considered in the second approach, which one (if any) is redundant? To answer such questions, clearly the group needs to design experiments that use varying sizes of training/testing data sets and record the computational time and classification accuracy of the two approaches.

Time Lines, Documentation of the group project and presentation:

The group should identify the topic of the project by Nov. 4, 2013 and submit an abstract of the proposed project by that date.

The group should submit the presentation slides by December 3, 2013.
The group presentation is to be conducted on Monday December 2, 2013

The group presentation should show the design, implementation of the AI system. In addition, the experiment design, experiment results and the interpretation of the results should be presented as well. Although a system demo is preferred as part of the group presentation, it is not mandatory.

Evaluation Criteria

The group project will be evaluated based on several factors:

(1) **The novelty of the project ideas and applications.**

The groups are encouraged to come up with original ideas and novel applications for the projects. A project with new ideas (algorithms, methods) on AI or new interesting applications of existing algorithms are scored higher than a project without much new idea/application.

(2) **The extensiveness of the study and experiments.**

A project that produces a more intelligent system by combining several AI techniques together, or a project that involves extensive experiments and thorough analysis of the experimental results are scored higher.

(3) **The quality of the group presentation.**

Some sample group projects:

(1) Design and simulation of a mobile robot that explores the environment and (possibly achieve given goals) through learning, search, and planning.

The project implements a software simulation of a mobile robot that can move around in a maze (an unknown environment) and learns to detect walls, avoid obstacles, find dead-ends, etc. Moreover the mobile robot could have other goals (such as finding gold, food, water, etc.) instead of trying to explore the environment.

One could implement various reinforcement learning algorithms or reflexive/goal-oriented/utility-based agents and test and compare the performance of these algorithms in several different two-dimensional grid worlds (with different obstacles and different assumptions on whether the sensor readings are noisy).

(2) Design and implementation of an intelligent game playing system which can play or gradually learns to play some games - board games (checkers, go, Othello, etc.) or interactive video games.

You may use the system to test the usefulness of Alpha-Beta pruning, the use of different evaluation functions for minimax search, the use of “pattern database”, and possibly a combination of techniques, etc.

Various learning algorithms (GA, reinforcement learning, polynomial function learning, decision trees, etc.) can be used as appropriate to learn the game-playing strategy.
You could also consider a game with chance (like Backgammon).

(3) Comparison of various search methods (and/or heuristic functions in $A^*$) for puzzle-solving problems.

(4) Design and implement a logical agent that maintains a knowledge base, accepts statements in propositional/first-order logic, and can answer queries posed to the knowledge base.

The inference of the agent could be using SAT solver or resolution.

(5) Design and implementation of an image classification system using Neural Networks. The project uses Neural Networks (backpropagation or some other type of NN) to build a classifier that can classify images into various categories.

You could also consider using neural networks for other kind of applications such as speech recognition - training a neural network to recognize spoken words.

(6) Design and implementation of a text mining system based on AI/ML techniques.

Various text mining topics can be explored. For example, one can build a text classification program (using Naive Bayes classifier) that can classify textual documents into different categories. Actually there is one such program available at CMU. See http://www.cs.cmu.edu/afs/cs.cmu.edu/user/mitchell/ftp/ml-examples.html. You can adapt, customize the code there and perform experiments on various issues of text mining.

One could also combine information retrieval techniques and ML methods (such as decision trees) for concept mining - extracting important key words (concepts) from texts, and for classifying texts into categories based on the extracted concepts.

(7) Design and implementation of an intelligent control system based on planning, search, and learning techniques.

The control system could be a complex system such as a mobile robot, a chemical plant control system, or a simple controller - for example, just a controller for an air-conditioner with the objective of maintaining certain temperature and humidity level and saving energy at the same time. One may need to identify the sensors (such as temperature and humidity readings) available to the controller, the actuators (actions the controller can perform), the performance objective and the experience/environment available to train the learning system.

(8) Design and implementation of GA-based systems for various optimization problems/applications.

For example, one could apply and compare several variants of GA for solving the traveling-salesman problem, or for the school class-time scheduling problems.

(9) Design and implementation of AI based systems to interesting application problems.