COURSE DESCRIPTION

Dept., Number	CS480	Course Title	Artificial Intelligence: Planning and Control
Semester hours	3	Course Coordinator	Dr. Shlomo Argamon, Associate Professor

Current Catalog Description

Introduction to computational methods of intelligent control of autonomous agents, and the use of programming paradigms that support development of flexible and reactive systems. These include heuristic search, knowledge representation, constraint satisfaction, probabilistic reasoning, decision-theoretic control, and sensor interpretation. Particular focus will be places on real-world application of the material. (3-0-3). Prerequisite: CS 331 or CS 401 or CS 403. Corequisite: MATH 474 or equivalent. (3-0-3) (T)

Textbook

Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach,* Prentice Hall Publishers, 2nd Edition, 2003, ISBN-0131038052

References

LISP References - textbook WWW page http://www.cs.berkeley.edu/~russell/aima.html

Course Outcomes

Students should be able to:

- Describe the Turing test.
- Explain the concepts of optimal reasoning, human-like reasoning, optimal behavior, human-like behavior.
- Develop "PAGE" descriptions of an agents and determine which agent type is applicable to a problem.
- Solve problems in a functional programming language (LISP)
- Formulate an efficient problem space for a problem expressed in English by expressing that problem space in terms of states, operators, an initial state, and a description of a goal state.
- Describe the problem of combinatorial explosion and its consequences.
- Select an appropriate brute-force search algorithm for a problem, implement it, and characterize its time and space complexities.
- Select an appropriate heuristic search algorithm for a problem and implement it by designing the necessary heuristic evaluation function.
- Describe under what conditions heuristic algorithms guarantee optimal solution.
- Implement minimax search with alpha-beta pruning for some two-player game.
- Formulate a problem specified in English as a constraint-satisfaction problem and

implement it using a chronological backtracking algorithm.

- Explain the operation of the resolution technique for theorem proving.
- Apply Bayes theorem to determine conditional probabilities.
- Explain the distinction between monotonic and non-monotonic inference.
- Explain the differences among the three main styles of learning: supervised, reinforcement, and unsupervised.
- Implement simple algorithms for supervised learning, reinforcement learning, and unsupervised learning.
- Determine which of the three learning styles is appropriate to a particular problem domain.
- Compare and contrast each of the following techniques, providing examples of when each strategy is superior: decision trees, neural networks, and belief networks. Explain the nearest neighbor algorithm and its place within learning theory.

Relationship between Course Outcomes and Program Outcomes

The following Program Outcomes are supported by the above Course Outcomes:

- a. An ability to apply knowledge of computing and mathematics appropriate to the discipline
- i. An ability to use current techniques, skills, and tools necessary for computing practices.
- j. An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices

Prerequisites by Topic

- Programming including recursion
- Discrete mathematics and data structures
- Basic analysis of algorithms

Major Topics Covered in the Course

1. Introduction, History of AI, Intelligent agents	3 hours
2. Functional Programming (LISP)	7.5 hours
3. Uninformed search, Informed search, Constraint satisfaction, Game-playing	12 hours
4. Logical agents, Propositional logic, First-order logic, Inference in first-order logic	4.5 hours
5. Uncertainty, Probability, Belief networks, Belief network inference, Optimal decisions under uncertainty, Optimal sequential decisions	10.5 hours
6. Learning, Neural networks, Bayesian learning	6 hours
Midterm Exam	1.5 hours
Final Exam	-
	45 hours

Assessment Plan for the Course

End of every semester Course Objective Assessments by CS department. End of semester Course Evaluations by IIT. Reviewed every Spring semester by CS Undergraduate Studies Committee for possible updates in the following Fall. Once every 4-5 years a detailed review of all materials for the course is made by the CS Undergraduate Studies Committee.

How Data in the Course is Used to Assess Program Outcomes (unless adequately covered already in the assessment discussion under Criterion 4)

See the assessment discussion under Criterion 4

For a computer science program

Estimate Curriculum Category Content (Semester hours)

Area	Core	Advanced	Area	Core	Advanced
Algorithms		3	Software design		
Data structures			Concepts of programming languages		