CS743 EVOLUTIONARY COMPUTATION 3-0-1-4

Course Description

Evolutionary Computation (EC), inspired by Darwinian Evolution and a sub-field of Computational/Artificial Intelligence, is about the study of Evolutionary Algorithms (EAs) for search and optimization. EAs have been proven to be effective for solving ill-defined complex optimization problems encountered in Engineering, Science, Finance etc. This course both introduces to students the field of EC and exposes them to the nuances of applying and designing EAs for problems encountered in a wide range of application domains. This course is both appropriate for students who are interested in pursuing research in EC and for students who are intending to use EAs as a tool to solve problems of their interest. Pre-requisite for this course include linear algebra, probability and statistics.

Learning Objectives

- To understand Evolutionary Algorithms in the context of meta-heuristics and their important parametric components
- To learn to formulate a given problem as an optimization problem and apply EAs
- To understand and appreciate the state-of-the-art research in EC

Pedagogy

- flipped learning method for introductory topics
- practical hands-on implementation of algorithms with example problems and discussions on the implementation design
- research paper reading and discussion for advanced topics
- assignments and exams focusing on problem solving

Syllabus

Introduction to Evolutionary Computation – Evolutionary Algorithms (Genetic Algorithms, Genetic Programming, Differential Evolution, Evolution Strategies, Covariance Matrix Adaptation etc.) – Different Components of Evolutionary Algorithms.

Fitness Landscapes – Adaptive Parameter Control and Tuning – Constraint Handling – Niching and Fitness Sharing – Memetic Algorithms – Ensemble Evolutionary Algorithms - Hybridization with other techniques – Multi-Objective Optimization – Hyper-Heuristics – Special Forms of Evolution (Co-evolution and Speciation) – Experimental (statistical) Methods for the analysis of Evolutionary Algorithms – Theoretical Analysis of Evolutionary Algorithms – Interactive Evolutionary Algorithms – Experiment design and analysis involving Evolutionary Algorithms

Evolutionary Machine Learning – Surrogate Assisted Optimization – NeuroEvolution – Quality Diversity Algorithms – Open Ended Evolution. Applications of Evolutionary Algorithms.

References

1. A. E. Eiben and J. E. Smith, "An Introduction to Evolutionary Computing", Natural Computing Series, Springer, 2nd Edition, 2015.
2. Eyal Wirsansky, "Hands-On Genetic Algorithms with Python: Applying Genetic Algorithms to Solve Real-World Deep Learning and Artificial Intelligence Problems", Packt Publishing, 2020.
3. Iaroslav Omelianenko, "Hands-on Neuroevolution with Python: Build High-Performing Artificial Neural Network Architectures using Neuroevolution-based Algorithm", Packt Publishing, 2019.
4. Slim Bechikh, Rituparna Datta and Abhishek Gupta (Eds.), "Recent Advances in Evolutionary Multi-objective Optimization", Adaptation, Learning, and Optimization Book – 20, Springer, 2017.
5. Nelishia Pillay and Rong Qu, "Hyper-Heuristics: Theory and Applications", Springer, 2018.
6. Hitoshi Iba, "Evolutionary Approach to Machine Learning and Deep Neural Networks: Neuro-Evolution and Gene Regulatory Networks", Springer, 2018.

Course Outcomes

On successful completion of this course, the students will be able to
1. Formulate a given problem amenable for evolutionary optimization/search
2. Apply appropriate evolutionary algorithms for a given problem
3. Analyse the state-of-the-art evolutionary computation research literature
4. Design suitable evolutionary algorithms for a real world application

Evaluation Pattern

- Continuous Assessment – 40 marks (practical implementation – 15 marks, assignments – 10 marks, paper reading and presentation+viva – 15 marks)
- Internal examinations – 30 marks (15 marks for each internal examination)
- End Semester – 30 marks (problem solving – theory+implementation followed by viva)

Activities/Content with direct bearing on Employability/ Entrepreneurship/ Skill development

- Flipped learning method to improve self-learning skill
- Practical hands-on implementation of algorithms to learn effective design of solutions keeping in mind the compute requirements
- Research paper reading and discussion to inculcate independent thinking and critical analysis
- Solving synthetic and real-world problems to improve problem solving skills