Air Force Institute of Technology

Graduate School of Engineering & Management

Department of Electrical and Computer Engineering

**CSCE686 Advanced Algorithm Design**

*“A Course in Deterministic & Stochastic Heuristic/Metaheuristic Optimization”*

Spring Quarter, 2019, 4 credits

**Course Description:** This course provides a theoretical and practical foundation for continuing the understanding and analysis associated with the design, complexity and correctness of algorithms and associated heuristics and metaheuristics for “complex” problems. Emphasis is on the *informal* specification and integration of general problem domain models with general algorithmic designs within a software engineering design perspective. Classification of various deterministic algorithm domains (solution strategies) including branch and bound, backtracking, divide and conquer, and Tabu search in concert with global and local *heurist*i*c* search concepts are applied to NP-Complete and PSpace problems. Stochastic search algorithms techniques are also analyzed via metaheuristrics. Such approaches include nature-inspired or bio-inspired evolutionary algorithms (genetic algorithms), ant colony agents, and simulated annealing. *Symbolic* properties of various approaches are addressed using graph theory, predicate calculus and computational models through the use of abstract data types (ADTs). Approximation algorithms are addressed regarding polynominal solution existence. Algorithmic problem solutions are also associated with proper data structure selection, implementation and visualization considerations. Selected *AF application optimization problems* associated with discrete patterns in artificial intelligence, knowledge based systems, software engineering, data base management, imaging, signal processing, VLSI, simulation, task scheduling, network optimization, planning and computer architecture are related through similarity of algorithm domain structures.

**Prerequisite:** CSCE486/586, Information Structures or equivalent (data structures - sets, sequences, arrays, graphs, lists, queues, stacks, trees; abstract data types (ADTs)); (algorithms - sorting; search:: depth-first, breath-first, best-first, dynamic programming) (analysis - counting and time/space complexity; ), NP-Completeness, and NP-C problems; (introductory graph theory (digraphs, paths, cycles, connectivity, ...) - shortest path problems; network flow model optimization; (math - recursive and iterative processes; probability and statistics; (SE - object-oriented software design, , C++, JAVA, Python, or other HOL programming capability, ...

**Texts:**

* El-Ghazali Talbi, **Metaheuristics, from Design to Implementation**, Wiley, ‘09
* Williamson and Shmoys, **The Design of Approximation Algorithms**, Cambridge, ‘11

(available on line)

* Course lecture notes

**Secondary Texts**: (integrated partial coverage of some material)

* Aarts & Lenstra, *Local Search in Combinational Optimization*, Princeton University

Press, 2003 (Hardcover by Wiley, 1997)

* Christofides, ***Graph Theory, An Algorithmic Approach***, Academic Press, 1975.
* de Castro, Leandro Nunes**,** *Fundamentals of Natural Computing*, Chapman & Hall, ‘06
* Dreo, Petrowski, Siarry, Taillard, *Metaheuristics for Hard Optimization*, Springer, ‘06
* Floreaqno and Mattiussi, *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies*, MIT press, 2008
* Kleinberg and Tardos, *Algorithm Design*, Addison Wesley, 2005 (CSCE586 text)
* Michaelewicz and Fogel, *How to Solve it: Modern Heuristics*, 2ed,Springer, 04’
* Pearl, ***Intelligent Search Strategies for Computer Problem Solving***, AW, 1984.
* Vazirani, *Approximation algorithms*, Springer, 2003
* Yang, X., *Nature-Inspired Optimization Algorithms*, Elsevier, 2014
* Xin- She Yang , *Engineering Optimization: An Introduction with Metaheueristic Applications,* Wiley, 2010

**Educational Goals:** Specific course educational goals are reveled as topics are studied:

* *understanding* of NP-complete (NPC) and PSpace problem domains, concepts, classifications and their design integration (LP, ILP, NP-Strong, NP-weak,…)
* *ability* to describe and use the variety of deterministic algorithmic search techniques (DFS, DFS/BT, BFS, Z\*, A\*, Divide and Conquer, Dynamic Programming, Tabu search, min/Max, symmetry reduction-jump point, …)
* *ability* to describe and use variety of nature-inspired/bio-inspired stochastic algorithmic search techniques (SA, EAs, GAs, PSO, DE, EDA, Ants, Bees, AIS, Swarm Intelligence)
* *ability* to use heuristic/metaheuristic global and local search (neighborhood) methods
* *ability* to select and employ appropriate approximation algorithms for LP and ILP models with and without constraints possibly resulting in a PTAS or a FPTAS model.
* *ability* to formalize the NPC domain representation and algorithm domain representation
* *understanding and ability* to use a variety of general search algorithm design-templates with application to constrained NPC combinatoric problem domains for problem solving
* *understandin*g of various taxonomies for search algorithm domains
* *ability* to refine (transformation, instantiate) general search algorithms to solve specific problem types integrating proper data structures along with appropriate metaheuristics and heuristics
* *understanding* of the design of distributed/parallel search using ”agent” structures
* *continued ability* to show that a problem type is NPC or of greater complexity

**Topical Outline:** *(advanced heuristic and metaheuristic algorithm development)*

* Heuristics/Metaheuristics Concepts (models, optimization techniques, representations,

constraints, parameter tuning, performance analysis, …)

* Solvable and Unsolvable Problems and Intractable Problems
* Problem Domain Model and Algorithm Domain Concepts and classifications
* Search Heuristics and their Formal properties (Matroids, POMDP, ...)

*•* Problem Descriptions (formal, informal, graphical models, linear vs. non-linear, ...)

*•*  NP-complete Problem Modeling and Algorithm integration (satisfiability, knapsack, TSP, set cover, coloring, network flow, scheduling, cliques, independent sets, ...) – PSPACE, …

* Approximation Techniques (NPC problems, relaxation of constraints/dim., local search, …)

*•* Algorithmic Search Techniques Integration for Problem Solving, Applications and their

Classification including heuristics/metaheuristics and ”agent” structures; examples included:

* Linear Problem Domain Models (Linear/Integer Programming; dual, simplex, …)
* Greedy Algorithms (depth first search; hill climbing, simulated annealing, Tabu search)
* Global Search vs. Local Search (memetic algorithm combinations)
* Rounding data (deterministic, random) – Cuts and Metrics
* Dynamic Programming Algorithm (scheduling, imaging, best-first comparison,...)
* Divide and Conquer - Branch and Bound (cryptography, matrix multiply)
* Graph Search (depth-first, breath-first, best-first, general best-first: A\*, AO\*, Z\*, ID,...)
* Stochastic/Heuristic Algorithms (simulated annealing, Tabu search, min/Max, ..)
* Bio-inspired metaheuristics (genetic algorithms, ant colony, particle swarm, bee,…)
* Solving Dynamic NP-C problems; changing model parameters
* Constraint Satisfaction and Handling (standard, hybrid techniques, data mining, …)
* Algorithm Complexity Modeling (NP-C, PSpace, EXPTIME,, mappings, solvability, …)
* Algorithm Educational and Visualization Environments
* Performance evaluation – design of computational experiments, statistical analysis, …

**Evaluation:** Homework; Evaluations of NPC software; Final exam; an individual project that emphasizes the software development, use and analysis of specific NPC or greater complexity problems.

**Project:**

* Development, AF use, and analysis of at least 2 *combined* NP complete problems (or greater) as defined by the instructor. Use/development of software and selection based on mutual agreement of optimization domain. Algorithm development should focus both on a deterministic search algorithm, a stochastic search algorithm, and local search. Emphasis on integrated algorithm high level/low level design and implementation, design of experiments and statistical analysis of results. A formal report is required per a given conference/journal paper outline (ACM or IEEE). Project can be an aspect of MS thesis research (if needed, a list will be provided). More details in third week of quarter.

**Grading:** Grades will be based upon homework assignments, evaluations, project, and the final exam as follows:

* Homework - 25%
* Software Evaluations – 10%
* Project – 35% (project determined by instructor and student)
* Final - 30%,
* Quick Quizzes - ??

**Workshop:** Sometimes a weekly workshop/seminar will be scheduled in order to discuss homework and evaluations. \*class, groups, individual

**Other References: (incomplete)** *“Different ways of presenting lecture material”*

*•* Aho, Hopcraft & Ullman, **The Design and Analysis of Computer Algorithms**, AW, 1974

*•* Aho, Hopcraft & Ullman, **Data Structures and Algorithms**, AW, 1987

* Banachowski, Kreczmar and Rytter, **Analysis of Algorithms and Data Structures**, AW,1991
* Bertsekas, D. P., **Network Optimization: Continuous and Discrete Models (Optimization, Computation, and Control),** MIT, 1998
* Bonabeau, Dorigo, Theraulaz, **Swarm Intelligence,** Oxford Press, 1999

*•* Booch & Bryan **Software Engineering with Ada**, Benjamin Cummings, 1994

*•* Brassard & Bratley, **Algorithmics, Theory and Practice**, Prentice Hall, 1988. (out of print)

*•* Brassard & Bratley, **Fundamentals of Algorithmics**, Prentice Hall, 1996

* Coello, Lamont and Van veldhuizen, ***Evolutionary Algorithms for*** ***Solving Multiobjective***

***Problems***, 2002, Kluwer/Springer, 2nd edition 2007

* Coello and Lamont, **Applications of Multi-Objective Evolutionary Algorithms,** World

Scientific, 2004

*•* Cook, Cunningham, Polleyblank, Schrijver, **Combinatorial Optimization**, Wiley, 1992

* Cormen, Leiserson & Rivest, **Introduction to Algorithms**, McGraw-Hill, 1991
* Corne, Dorigo &Glover, **New Ideas in Optimization**, McGraw-Hill, 1999
* Garey, M. R. and Johnson, D. S., **Computers and Intractability - A Guide to the theory   
   of** **NP-Completeness**, Freeman, 1979.
* Goldberg, **Genetic Algorithms in Search, Optimization and Learning**, AW, 1989

*•* Harel, **Algorithmics, The spirit of computing**, Addison-Wesley, 1987

*•* Horowitz & Sahni, **Fundamentals of Computer Algorithms**, Computer Science, 1982

*•* Jennings and Wooldridge, eds., **Agent Technology**, Springer, 1998

* Korsh, **Data Structures, Algorithms, and Program Style**, PWS Computer Science, ‘86
* LaValle, Steven, **Planning Algorithms**, Cambridge University Press, 2006

*•* Manber, **Introduction to Algorithms, A Creative Approach**, Addison Wesley, 1989

*•* Michalewicz, **Genetic Algorithms + Data Structures = Evolution Programs**, Springer 1999

* Neapolitan & Naimipour, *Foundations of Algorithms*, Jones & Bartlett, 1997

*•* Reingold, Nievergelt & Deo, **Combinatorial Algorithms, theory/practice**, Prentice-Hall 1977

*•* Rawlins, **Compared to What? an introduction to the analysis of algorithms**, Computer

Science Press, 1992

* Skiena, **The Algorithm Design Manual**, Springer, 1997 (see author’s web site)

*•* Williamson, **Combinatorics for Computer Science**, Computer Science Press, 1985

*•* Wilson & Watkins, **Graphs - An Introductory Approach**, Wiley, 1990

*•* Weiss, **Data Structures and Algorithm Analysis**, Benjamin/Cummings, 1992

**Estimated ABET Category Content:** Engineering Science(40%), Math(20%),

Engineering Design(40%)

**CSCE686 Course Director:** Professor Gary B. Lamont, room 311B, bldg 640, AFIT/ENG

**Suggested Future or Concurrent Classes:**

*•* CSCE631 Machines, Logic, and Languages (a capstone course for CS/CE students)

*•* CSCE656/790 Parallel and Distributed Algorithms

*•* CSCE786 Reactive Concurrent Systems (temporal logic)

*•* CSCE886 Evolutionary Computation/Multi-Objective Optimization (Pareto front, …)

**Associated Special Studies Available:** (considerable experience in various algorithmic domains)

-- CSCE699 ”Special Study in Specific Problem Solution Optimization” Solving   
 complex very high dimensional constrained problems that are extensions of

Classical NPC problems (single-objective and multi-objective)

-- Population based stochastic optimization search models (detailed analysis)

* + - Cognitive Systems
    - Ant colony optimization, wasps, bees, …
    - Genetic algorithm and parameter optimizations
    - Artificial Immune Systems (Intrusion and anomaly detection)
    - Particle swarm- differential algorithm optimizations
    - POMDP, I-POMDP, DEC-POMDP understanding and application
    - Multi-Objective evolutionary algorithm (MOEAs) design
    - Game theory and optimization playing

-- Parallel and Distributed optimization algorithms for NPC problems (extending   
 CSCE656, CSCE790) **“Efficiency vs. Effectiveness**

**Academic Integrity:** (Reference: Student Handbook, ENOI 36 – 107, *Academic Integrity*)

**NO LAP-TOP COMPUTERS or TABLETS or CELL PHONES to be used during CLASS!!**

GBL rev t 3/2019