Engineering Faculty Document No. 10-19
04/23/2018

TO: The Faculty of the College of Engineering

FROM: The Faculty of the School of Industrial Engineering

RE: New Graduate Course – IE 54100: Nature-Inspired Computation

IE 54100 NATURE-INSPIRED COMPUTATION
SEM 1 or 2, Lecture 3, Cr. 3
Graduate Standing OR [(CS 15900 OR CS 15800 OR CS 18000) AND IE 33500 AND IE 33000]

COURSE DESCRIPTION: This course is about algorithms that are inspired by naturally occurring phenomena and applying them to optimization, design and learning problems. The focus is on the process of abstracting algorithms from the observed phenomenon, their outcome analysis and comparison as well as their "science". This will be done primarily through the lens of evolutionary computation, swarm intelligence (ant colony and particle-based methods) and neural networks.

REASON: The course is designed to meet the needs of students within the School of Industrial Engineering (IE), and potentially other computationally oriented majors studying learning, optimization and the optimal design of complex systems. The course fits within the operations research area of the School of IE, and is complementary to traditional training in optimization (e.g., IE 53500 – Linear Programming, IE 53800 – Nonlinear Optimization Algorithms and Models, etc.). There was previously a course on Heuristic Optimization (IE 63100), which was expired after associated faculty left the university. However, there has been a persistent need to cover heuristic methods (especially those inspired by nature) of optimization. This course fills that gap.

4/23/18

Abhijit Deshmukh
Professor and Head
School of Industrial Engineering

Detailed Graduate Course Proposal for Academic Review

Note: The detailed course proposal is intended for academic review by the appropriate area committee of the Graduate Council. It supplements the Form 40G that is intended for administrative review of the Graduate School and Registrar.

To: Purdue University Graduate Council

From: Faculty Member: Mario Ventresca   Department: IE
Campus: West Lafayette

Date: February 26, 2018

Subject: Proposal for New Graduate Course

Contact for information if questions arise:

Name: Patrick Brunese
Phone: 49611
Email: pbrunese@purdue.edu
Address: GRIS 165

Course Number: IE 54100
Course Title: Nature-Inspired Computation
Short Title: Nature-Inspired Computation

Course Description:

This course is about algorithms that are inspired by naturally occurring phenomena and applying them to optimization, design and learning problems. The focus is on the process of abstracting algorithms from the observed phenomenon, their outcome analysis and comparison as well as their “science”. This will be done primarily through the lens of evolutionary computation, swarm intelligence (ant colony and particle-based methods) and neural networks.

A. Justification for the Course

Justification of the need for the course

- Design problems in modern complex systems involve consideration of large numbers of variables, constraints, and complex mathematical structures in both static and dynamic environments. Traditional mathematical approaches (i.e., mathematical optimization) are beneficial, but often limited computationally or the problem being solved is ill-defined (i.e., an appropriate objective function may be too difficult to devise or evaluate). Design methods arising from the study of natural systems provide computationally beneficial approaches to coping with the complexity of the design environment. This course provides an overview of numerous nature-inspired optimization methods, as well as knowledge related to the practical analysis of these algorithms.
- The course is designed to meet the needs of students within the School of Industrial Engineering (IE), and potentially other computationally oriented majors studying learning, optimization and the optimal design of complex systems. The course fits within the operations research area of the School of IE, and is complementary to traditional training in optimization (e.g., IE 53500 – Linear Programming, IE 53800 – Nonlinear Optimization Algorithms and Models, etc.). There was previously a course on Heuristic Optimization (IE 63100), which was expired after associated faculty left the university. However, there has been a persistent need to cover heuristic methods (especially those inspired by nature) of optimization. This course fills that gap.
- The course, if created, could be used to meet degree requirements for IE undergraduate, master, and Ph.D. level students. It could also be potentially be added as a course within the "Computational Optimization" core area of the Computational Science & Engineering interdisciplinary graduate program.

Justification that course will be taught at a graduate level

- One facet of the course involves students preparing and presenting a lecture based on primary literature. Often this material has not yet made it to textbooks. The strategy is to utilize textbooks and instructor lecture notes to form a foundation for the student being able to comprehend more advanced primary literature and complete a course project.
- Assessments are problem-solving oriented with some degree of open-endedness that allows students some freedom in how they decide to approach their solutions. This demands students to internalize what has been taught in lecture and utilize their problem solving skills to modify the material to suit the nuances of the assessments/homeworks. The course project is proposed by each student and must be accepted by the instructor as being sufficiently novel to warrant further study within the constraints of the semester and material covered in lecture. Emphasis on performing appropriate analyses is given throughout the lectures, and is expected in the project. Repeating existing studies is not sufficient.

- Lecture topics are current in that there are constantly new problems being solved using nature-inspired heuristics from various problem domains. This is true for large IT companies such as Google who have been using evolutionary algorithms to automate the design of deep neural networks in it's OpenAI framework (openai.com). Moreover, the broad research community should also be interested as these methods have been used to provide a better understanding of natural phenomena such as the evolutionary origin of communication (Grouchy et al. On The Evolutionary Origin of Symbolic Communication, Nature Scientific Reports 6:34651, 2016).

Justification of the demand for the course

- Anticipated enrollment
    - Undergraduate 1-5
    - Graduate 10-20

Justification for online delivery

The course will not be delivered online.

B. Learning Outcomes and Methods of Assessment

Learning Outcomes	Assessment Methods
Describe the natural phenomena that motivate the discussed algorithms.	• Student lecture requires specifying the basis/motivation of the approach being taught
Understand the strengths, weaknesses and appropriateness of nature-inspired algorithms.	• Assignments, project and student lecture all have questions specifically addressing this issue, e.g., by comparing to other techniques
Apply nature-inspired algorithms to optimization, design and learning problems.	• Assignment and project require students to solve such problems using the techniques taught in class
Understand fundamental concepts of NP-hardness and computational complexity.	• Assignment (usually first) concerns these issues: students are required to determine computational complexity of algorithms and perform reductions to show NP-completeness
Prove algorithm convergence rates using probabilistic arguments.	• Assignment questions require students to perform the calculations/proofs for specific algorithms and problems
Perform appropriate analyses on and between the outputs of stochastic algorithms.	• Assignment asks students to compare different approaches using statistically appropriate methods. Although a set of methods is discussed in the class, the specific ones needed to address the questions are usually not specified in the assignment and are left to the student to decide.
Analyze search space structure using statistical and information theoretic measures, and explain its impact on algorithm behavior and output.	• Assignment questions asking students to perform this analysis for a designated problem and to extrapolate beyond the numerical results to interpret their meaning wrt expected algorithm behavior

- Homework Assignments: Homeworks are due every 3-4 weeks, and explore the main topics in class over that duration. These are a mixture of computational/programming, algorithm design and mathematics that also emphasize problem solving, critical thinking and analysis skills. They are required to write the assignment in Latex in a format used by IEEE or ACM conferences.
- Course Project: The project will be completed in two phases. Phase 1 consists of a proposal to the instructor regarding a problem and proposed methods. Phase 2 consists of a 8-12 page (not including references) double column IEEE formatted report (written in Latex), in addition to all source codes, which are to be properly documented and include instructions for reproducing experimental results. Projects should be completed individually or by groups of no more than 3 students (expectations scale with group size).
- Student-Guided Lecture: Each student will be responsible for preparing and conducting one 20-30 minute lecture, allowing for an additional 5-7 minutes question/answer period.

A copy of the lecture notes must be prepared in PDF and given to the class before the lecture is conducted. The topic will be assigned by the instructor. Grading and feedback will be 50% based on class input, and 50% from the instructor.

Final Grading Criteria

Describing the criteria that will be used to assess students and how the final grade will be determined. Add and delete rows as needed.

Assessment Methods (should match method types in the previous table)	Weight Toward Final Course Grade
Project	45%
Homework	40%
Student-Guided Lecture	15%

Methods of Instruction

Class Hrs/Week	Method of Instruction	Contribution to Outcomes
3	Lecture	The lecture sessions introduce the theoretical content and models the algorithm analysis process. Using the approaches presented in class gives the students a framework on which to build their own skills through the course assignments.

C. Prerequisite(s)

- Students should be comfortable writing computer programs in a modern language (e.g., C/C++, Java, Matlab, Python, etc.)
    - CS 15900 – Programming Applications for Engineers or CS 15800 – C Programming or CS 18000 – Problem Solving and Object Oriented Programming
- Students should be familiar with optimization methods and algorithms
    - IE 33500 – Operations Research – Optimization
- Students should be able to perform basic statistical analyses, and be able to create plots using a programming language such as SAS/R/Minitab
    - IE 33000 – Probability and Statistics In Engineering II

D. Course Instructor(s)

Name	Rank	School, dept., or center	Graduate Faculty or expected date
Mario Ventresca	Assistant Professor	IE	Yes

Mario Ventresca's research falls in the general area of computational science and engineering, which focuses on the design, analysis and application of computational and mathematical approaches for solving challenging real-world problems, as well as for understanding real and artificial complex systems. He has interest in different application domains, such as:

- **Approximation algorithms**, which aim to efficiently solve difficult optimization problems, while also providing guarantees on solution quality and run-time performance.
- **Automated design and inference**, whereby robust algorithms are developed for the purpose of automatically designing complex systems, or to ascertain the underlying principles or rules of a given system.
- **Biomedical science and operations research**, which seeks to deepen our understanding of biological processes with the intention of enhancing diagnosis and treatment capabilities.
- **Complexity engineering**, where concepts from complexity science such as emergence and self-organization are applied to the design of complex systems.
- **Discrete optimization**, which is concerned with problems where we must select the best solution from a finite number of feasible solutions; encompassing areas such as linear programming, graph theory, scheduling and routing.
- **Machine learning**, which leverages computational power to learn from large and/or complex data sets in order to develop intelligent systems.
- **Nature-inspired computing**, which seeks to develop algorithms based on concepts such as evolution and swarming intelligence to solve complex real-world problems.
- **Network science**, which focuses on understanding, controlling and predicting the structure and function of interconnected systems as well as processes acting upon them

E. Course Schedule or Outline

Option 1: Schedule Format

Week	Topic(s)
1	• Turing Machines • Turing Machines and Computational Complexity • Computational Complexity
2	• Reductions • Cook-Levin Theorem
3	• Reductions • No Free Lunch Theorem • Empirical Algorithms
4	• Characteristics of Natural Systems/Algorithms • Evolution
5	• Evolutionary Algorithms
6	• Fitness Landscape Analysis
7	• Evolutionary Algorithm Theory
8	• Genetic Programming • Swarm Intelligence
9	• Ant Colony Optimization • Particle Swarm Optimization
10	• Brains and Learning • Perceptron • Multi-Layer Neural Networks
11	• Self-Organizing Maps • Dynamic Environments
12	• Estimation of Distribution Algorithms • Black-Box Complexity
13	• Multiobjective Evolutionary Algorithms • Cellular Automata/Artificial Life
14	• Artificial Immune Systems • Co-Evolutionary Algorithms

Week	Topic(s)
15	• Evolutionary Robotics
16	Not applicable

F. Reading List (including course text)

Primary Reading List

Aside from course notes, students are requested to read the following:

- Optimization in dynamic environments: a survey on problems, methods and measures, Cruz, González, Pelta. *Soft Computing* 15:1427-1448, 2011.
- A survey of repair methods used as constraint handling techniques in evolutionary algorithms, Salcedo-Sanz, *Computer Science Review*, 3(3):175-192, 2009.
- Algorithm runtime prediction: Methods & evaluation, Hutter, Xu,Hoos, Leyton-Brown, *Artificial Intelligence*, 206:79-111, 2014.
- A Tutorial on Evolutionary Multiobjective Optimization, Zitzler, Laumanns, Bleuler, *in Metaheuristics for Multiobjective Optimization*, pp:3-37, 2004.]
- Simple Explanation of the No Free Lunch Theorem and its Implications, Ho, Pepyne, *Journal of Optimization Theory and Applications*, 115(3):549-570, 2002.

Secondary Reading List

- Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications, L.N. de Castro (2006), CRC Press.
- Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies, D. Floreano and C. Mattiussi (2008), MIT Press.
- Evolutionary Optimization Algorithms, D. Simon (2013), John Wiley & Sons.

G. Library Resources

Name of journal, proceedings, book, video, or other acquisition	**Already in Libraries?**
Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies, D. Floreano and C. Mattiussi (2008), MIT Press.	Yes, e-Book
Evolutionary Optimization Algorithms, D. Simon (2013), John Wiley & Sons.	Yes, e-Book
Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications, L.N. de Castro (2006), CRC Press.	No

H. Course Syllabus (now required)

See attached.

IE 590: Nature-Inspired Computing

Syllabus for Fall 2017

Course Overview

This course is about algorithms that are inspired by naturally occurring phenomena and applying them to optimization, design and learning problems. The focus is on the process of abstracting algorithms from the observed phenomenon, their outcome analysis and comparison as well as their "science". This will be done primarily through the lens of evolutionary computation and swarm intelligence (ant colony and particle-based methods). Time permitting, neural networks will also be briefly introduced.

Prerequisites: Students must be very comfortable programming in at least one modern programming language such as C/C++/Java/Matlab/Python, and have familiarity with basic optimization methods (e.g., linear programming). Ability to perform basic statistical analyses/hypothesis tests and create plots using a programming language such as SAS/R/Minitab will be required.
** Microsoft Excel, Word and similar software tools will NOT be permitted.*

Instructor Mario Ventresca
E-mail: `mventresca@purdue.edu`
Office hours: Tuesdays 11am-12pm (appointment recommended)
Office location: Grissom 292

TA Viplove Arora (`arora34@purdue.edu`)

Lectures MWF 10:30-11:20pm, GRIS 126

Textbook None required. Suggested (one of):

- *Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications*, L. N. de Castro (2006), CRC Press.
- *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies*, D. Floreano and C. Mattiussi (2008), MIT Press.
- *Evolutionary Optimization Algorithms*, D. Simon (2013), Wiley.

Website `https://mycourses.purdue.edu/`

*** Contacting instructor via email:** To prevent delay or loss by spam blockers, use the subject line (without quotes and appropriate course component): "IE590: lecture/project/homework/other"

Learning Outcomes

Upon completing this course students will be able to:

1. Describe the natural phenomena that motivate the discussed algorithms.
2. Understand the strengths, weaknesses and appropriateness of nature-inspired algorithms.
3. Apply nature-inspired algorithms to optimization, design and learning problems.
4. Understand fundamental concepts of NP-hardness and computational complexity.
5. Prove algorithm convergence rates using probabilistic arguments.
6. Perform appropriate analyses on and between the outputs of stochastic algorithms.
7. Analyze search space structure using statistical and information theoretic measures, and explain its impact on algorithm behavior and output.

Grading

This course has four grading components, as shown below. Due dates TBD. Assignments will require programming.

Learning Item	**Number**	**Item Value**	**Percentage of Final Grade**
Assignment	4	10%	40%
Project	1	45%	45%
Lecture	1	15%	15%

Your letter grade will be determined as follows, where X is your final percent grade:

Letter Grade	**Percentage**	**Description**
A, A+	$90 \leq X$	Excellent. Comprehensive knowledge and understanding of the subject matter.
B, B+	$80 \leq X < 90$	Very Good. Strong knowledge and understanding of the subject matter.
C-, C, C+	$70 \leq X < 80$	Good. Reasonable knowledge and understanding of the subject matter.
D-, D, D+	$60 \leq X < 70$	Marginal. Minimum knowledge and understanding of the subject matter.
F	$X < 60$	Failing. Unacceptable knowledge and understanding of the subject matter.

* For a grade above 80%, your letter grade will be indicated by a '+' if your numerical grade is in the upper half of the interval (i.e., for $X \in [85, 89)$ then you will receive a 'B+'). Where a '-' grade is possible the range will be divided into thirds, i.e., $[70, 74), [74, 77), [77, 80)$. <u>No grade curving will be conducted</u>.

Class Policies

Attendance: The university class attendance policy is available at

`http://www.purdue.edu/studentregulations/regulations_procedures/classes.html`

- Lecture attendance is expected and is your responsibility. Many topics are elaborated and discussed in lecture, and the information of these discussions is not posted online or easily attainable from other students.
- Be on time for lectures and labs. Late attendance (more than 10 minutes) will be considered as absence. If you are late for class please enter and seat yourself in a fashion that does not disturb the class.
- Attendance to student lectures will be considered when determining your final grade.

Homework: will NOT be accepted past the due date.

- MUST be done individually, and include ALL references (not including course notes or the textbook), including web pages.
- You are encouraged to discuss with other students, but handing in highly similar work is plagiarism (see below) and will be punished accordingly. If you discussed with any students, even if not in the course, you MUST list them in your assignment.
- ALL assignments will be examined by SafeAssign, a plagiarism detection service available in Blackboard. Do **NOT** rewrite the assignment question.
- Latex MUST be used to format your assignment. ALL assignments must be in a SINGLE pdf file, ONLY the most recently submitted assignment will be graded.
- **MUST** (otherwise a grade of zero) include as the first page, a cover sheet indicating (in order) the course title, assignment number, date, your name, PUID, and a brief statement indicating:

  "I am aware of, and understand, the Purdue Academic Misconduct Policies and attest that this submitted work is solely my own. I accept any repercussions if found in violation."

Project: The project will be completed in two phases. The first phase will be to ensure the project is appropriate, but students should be in communication with the instructor well before the deadline. The final project submission will require an 8-12 page (not including references) double column IEEE formatted report (written in Latex), in addition to all source codes, which are to be properly documented and include instructions for reproducing experimental results. Projects should be completed individually or by groups of no more than 3 students (expectations scale with group size).

Lecture: Each student will be responsible for preparing and conducting one 20min lecture, allowing for an additional 5-7 minutes question/answer period. A copy of the lecture notes must be prepared in pdf and given to the class before the lecture is conducted (otherwise a grade of zero). The topic will be assigned by the instructor. Grading and feedback will be 50% based on class input, and 50% from the instructor.

Re-grading: Request for regrading an examination or assignment will be considered only with a clearly written explanation, submitted in class only to the instructor and only AFTER at least 24 hours self-evaluation and within SEVEN days from the time the graded work is returned to the class. The group project is not subject to regrading. A regrade request form is available on Blackboard, and must be submitted with a hard-copy of your assignment/exam.

Class conduct: In order to ensure the best learning environment for all, please:

- be courteous to your fellow students and instructor.
- turn off or put your cellular phones on silent mode. If you must answer the phone, please quietly leave the classroom.
- do not use laptops or tablets during class if you intend to play movies, video games, etc. This disrupts nearby students and negatively affects their learning experience and thus is not permitted. Failure to adhere to this request may result in loss of the privilege.
- raise your hand if you have any questions or comments during the lecture and wait until the instructor calls on you.
- do not chatter with friends or purposely disturb other students.
- leave and re-enter the classroom quietly if you must use the restroom. You do not need to ask permission to leave unless during an examination.

Misconduct: Any type of misconduct as defined in Student Conduct (Part 5) of the University Regulations will not be tolerated: `http://www.purdue.edu/studentregulations/`. The instructors will follow the regulations strictly. These are a few examples:

- substituting on a course or exam for another student
- using someone else to write a paper or assignment and submitting it as one's own work, with or without monetary exchange
- giving or receiving answers by use of signals during an exam
- copying with or without the other person's knowledge during an exam
- doing class assignments for someone else
- turning in a paper that has been purchased from a commercial research firm or obtained from the internet
- padding items of a bibliography
- collaborating with other students on assignments when it is not allowed
- obtaining a test from the exam site, completing and submitting it later
- altering answers on a scored test and submitting it for a regrade
- stealing class assignments from other students and submitting them as one's own
- fabricating data
- destroying or stealing the work of other students
- using the exact language of someone else without the use of quotation marks and without giving proper credit to the author

- plagiarizing published material, class assignments, or lab reports
  - presenting the sequence of ideas or arranging someone else's material (even if expressed in one's own words) without giving appropriate acknowledgment
  - submitting work (in whole or part) written by someone else but representing it as your own

Emergency Procedures

In the event of a major campus emergency, course requirements, deadlines and grading percentages are subject to changes that may be necessitated by a revised semester calendar or other circumstances. Changes about the course will be announced through the Blackboard and/or class mailing list.

In case of emergency:

- To report an emergency, call 911.
- To obtain updates regarding an ongoing emergency, and to sign up for Purdue Alert text messages, view `http://www.purdue.edu/emergency/`
- There are nearly 300 Emergency Telephones outdoors across campus and in parking garages that connect directly to the Purdue Police Department (PUPD). If you feel threatened or need help, push the button and you will be connected immediately.
- If we hear a **fire alarm**, we will immediately suspend class, **evacuate the building**, and proceed outdoors, and away from the building. **Do not use the elevator.**
- If we are notified of a Shelter in Place requirement for a tornado warning, we will suspend class and shelter in the lowest level of this building away from windows and doors.
- If we are notified of a Shelter in Place requirement for a hazardous materials release, or a civil disturbance, including a shooting or other use of weapons, we will suspend class and shelter in our classroom, shutting any open doors or windows, locking or securing the door, and turning off the lights.
- Your course syllabus includes additional preparedness information as it might impact this class, including classroom suspension for severe weather or other emergencies. Please review the syllabus and the Emergency Preparedness website for additional information.
  `http://www.purdue.edu/ehps/emergency_preparedness/index.html`

Tentative Schedule (will be revised based on enrollment)

Week	Date	Topic	Remark
1	Aug 21	Course Introduction	
	Aug 23	Turing machines	
	Aug 25	Turing machines & computational complexity	
2	Aug 28	Computational complexity	
	Aug 30	Reductions	
	Sept 1	Cook-Levin Theorem	
3	Sept 4	No classes (Labor Day)	
	Sept 6	Reductions	
	Sept 8	No Free Lunch Theorem	
4	Sept 11	Empirical algorithms	
	Sept 13	Characteristics of Natural Systems/Algorithms	
	Sept 15	Evolution	
5	Sept 18	Evolution	
	Sept 20	Evolutionary Algorithms	
	Sept 22	Evolutionary Algorithms	
6	Sept 25	Evolutionary Algorithms	
	Sept 27	Fitness Landscape Analysis	
	Sept 29	Fitness Landscape Analysis	
7	Oct 2	Fitness Landscape Analysis	
	Oct 4	EA Theory	
	Oct 6	EA Theory	
8	Oct 9	No lecture (Fall break)	
	Oct 11	EA Theory	
	Oct 13	Genetic Programming	
9	Oct 16	Swarm Intelligence	
	Oct 18	Ant Colony Optimization	
	Oct 20	Ant Colony Optimization	
10	Oct 23	Particle Swarm Optimization	
	Oct 25	Brains and Learning	
	Oct 27	Perceptron	
11	Oct 30	Multi-Layer Neural Networks	
	Nov 1	Self Organizing Maps	
	Nov 3	TBD	
12	Nov 6	Dynamic Environments	
	Nov 8	Estimation of Distribution Algorithms	
	Nov 10	TBD	
13	Nov 13	Black-box complexity	
	Nov 15	Multiobjective EAs	
	Nov 17	Cellular Automata/artificial life	
14	Nov 20	Artificial Immune Systems	
	Nov 22	No lecture (Thanksgiving)	
	Nov 24	No lecture (Thanksgiving)	
15	Nov 27	Co-Evolutionary Algorithms	
	Nov 29	Evolutionary Robotics	
	Dec 1	TBD	
16	Dec 4		
	Dec 6		
	Dec 8		
	Dec 10		Project Due (11:59pm)