



1. COURSE

CS365. Evolutionary Computing (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerequisites	:	CS262. Machine learning. (7 th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

La Computación Evolutiva comprende un conjunto de metodologías de búsqueda y optimización cuya base primordial es el Paradigma Neodarwiniano que agrupa la Herencia Genética (Mendel), el Seleccionismo (Weismann) y la Evolución de las Especies (Darwin) que, cuando llevadas a implementaciones computacionales, ofrecen una herramienta poderosa de optimización global para una determinada función objetivo. Son bastante robustos cuando se supone la existencia de muchos óptimos locales. De esta forma, estos algoritmos pueden aplicarse en diversos problemas de optimización.

5. GOALS

- Que el alumno sea capaz de entender y aplicar el Paradigma Neodarwiniano para solucionar problemas complejos de optimización.
- Entendimiento a detalle del principio, fundamentos teóricos, funcionamiento, implementación, interpretación de resultados y operación de los algoritmos de la Computación Evolutiva más populares y utilizados por la comunidad científica y profesional.
- Conocimiento del estado del arte en Computación Evolutiva
- Capacidad de tratar un problema real de optimización utilizando Computación Evolutiva

6. COMPETENCES

Nooutcomes

7. SPECIFIC COMPETENCES

Nospecificoutcomes

8. TOPICS

Unit 1: Introducción a la Optimización (4)	
Competences Expected: a,b	
Topics	Learning Outcomes
<ul style="list-style-type: none"> • Definiciones de Optimización: principio de estabilidad, optimización global. • Optimización Clásica: Definición del problema de optimización, concepto de convexidad, optimización numérica y combinatoria. • Técnicas de optimización clásica: optimización lineal, algoritmo simplex, optimización no lineal, algoritmos <i>steepest descent</i>, <i>conjugate gradient</i>, algoritmos de búsqueda, programación dinámica, • Heurísticas: definición, <i>Tabu search</i>, <i>Hill Climbing</i>, <i>Simulated Annealing</i>, <i>Evolutionary Algorithms</i> 	<ul style="list-style-type: none"> • Entender los principios básicos de la optimización • Entender e implementar algoritmos básicos de Optimización aplicados a problemas <i>benchmark</i>. • Entender la necesidad de uso de heurísticas
Readings : [Wei09], [RBK12]	

Unit 2: Computación Evolutiva: Conceptos básicos (8)	
Competences Expected: a,b,i	
Topics	Learning Outcomes
<ul style="list-style-type: none"> • Computación Evolutiva: definiciones • Ideas precursoras: El origen de las ideas, L'Eclerc, Lamarck, Darwin, Weismann, Mendel, Baldwin, Paradigma Neodarwiniano • Conceptos básicos de Computación Evolutiva: genes, cromosomas, individuos, población. • Paradigmas de la Computación Evolutiva: Programación Evolutiva, Estrategias Evolutivas, Algoritmos Genéticos, <i>Learning Classifier Systems</i>, Programación Genética. 	<ul style="list-style-type: none"> • Entender los principios básicos que rigen la computación evolutiva • Conocer el contexto en que surgió la computación evolutiva.
Readings : [RBK12], [Wei09], [Fog95], [koza98], [Mit04], [Mic96]	

Unit 3: Algoritmo Genético Canónico (8)	
Competences Expected: a,b,i	
Topics	Learning Outcomes
<ul style="list-style-type: none"> • Algoritmo Genético: definición, componentes. • Algoritmo Genético Canónico: procedimiento elemental, ciclo de un AG, representación (codificación binaria, real a binario, decodificación binario a real), inicialización de la población, evaluación y aptitud, selección (proporcional, torneo), operadores genéticos (cruces, mutaciones), el dilema <i>exploiting-exploring</i>, ajustes en la aptitud, ajustes en la selección. • Monitoreo de un AG: curvas <i>best-so-far</i>, <i>online</i>, <i>offline</i> • Convergencia • Teoría de <i>Schemata</i>: Máscaras, esquemas, definiciones y propiedades, <i>Schemata theorem</i>: impacto de la selección, cruce de 1 punto y mutación, teorema fundamental de los algoritmos genéticos, hipótesis de los bloques constructores. 	<ul style="list-style-type: none"> • Entender los algoritmos genéticos tradicionales. • Analizar y evaluar ventajas y desventajas del modelo genético tradicional. • Implementar un ejemplo de algoritmo genético tradicional y analizar su comportamiento.
Readings : [RBK12], [Hol75], [Gol89], [Mit04], [Mic96]	

Unit 4: Algoritmos Evolutivos en Optimización Numérica (8)	
Competences Expected: a,b,i	
Topics	Learning Outcomes
<ul style="list-style-type: none"> • Problemas con restricciones: definiciones, espacios válido e inválido. • Tratamiento de las restricciones: Penalización, reparación, uso de codificadores, operadores especializados. • Uso de codificación real: binario vs. real, algoritmo evolutivo con codificación real. • Modelo GENOCOP: tratamiento de restricciones lineales, inicialización, operadores, inicialización, modelo GENOCOP III para restricciones no lineales: reparación de individuos. 	<ul style="list-style-type: none"> • Comprensión de las formas de tratar problemas de optimización con restricciones. • Entender y analizar los algoritmos evolutivos con codificación real. • Evaluar la aplicación de computación evolutiva en problemas de optimización numérica
Readings : [RBK12], [Mic96], [Mic00], [SC00]	

Unit 5: Algoritmos Evolutivos en Optimización Combinatoria (8)	
Competences Expected: a,b,i	
Topics	Learning Outcomes
<ul style="list-style-type: none"> • Espacios discretos y finitos • Algoritmos Evolutivos discretos: definición, modelo discreto generalizado • Algoritmos Evolutivos de orden: representación de soluciones, operadores de orden: cruces, mutaciones • Aplicaciones: <i>Quadratic assignment Problem</i> – QAP, <i>Travelling Salesman Problem</i> – TSP • Problemas de Planificación: variables típicas, características, representación, codificadores, evaluación de una planificación. 	<ul style="list-style-type: none"> • Comprender e identificar el uso de Computación Evolutiva en problemas de optimización combinatoria • Evaluar la aplicación de computación evolutiva en problemas reales discretos
Readings : [RBK12], [Mit04], [Cru03]	

Unit 6: Paralelización y Multiobjetivos (8)	
Competences Expected: a,b,i,j	
Topics	Learning Outcomes
<ul style="list-style-type: none"> • PEA - Algoritmos Evolutivos en Paralelo: arquitecturas de paralelización, arquitecturas <i>master-slave</i>, <i>coarse-grained</i>, <i>fine-grained</i> e híbridas • Análisis de la ejecución de una implementación <i>master-slave</i>. • Optimización de Múltiples Objetivos: Definición formal, criterio de Pareto, Algoritmos Evolutivos Multi Objetivos (MOEA) sin uso de Pareto, MOEA con uso de Pareto: MOGA, NSGA, NPGA, NPGA2, PESA, SPEA, SPEA-II, Algoritmo Microgenético. • MOEA – Métricas de desempeño, investigación futura 	<ul style="list-style-type: none"> • Comprender y analizar la capacidad de paralelización de los modelos evolutivos • Analizar la aplicabilidad de Computación Evolutiva en problemas de múltiples objetivos • Implementación de modelos paralelos y multiobjetivo
Readings : [RBK12], [Can00], [Coe07]	

Unit 7: Algoritmos Genéticos Avanzados (16)	
Competences Expected: a,b,i,j	
Topics	Learning Outcomes
<ul style="list-style-type: none"> • HEA – Algoritmos Evolutivos Híbridos: Por qué hibridizar?, formas de hibridización, búsqueda local y aprendizaje. • GP – Programación Genética: definición, representación, ciclo de la GP. • CA – Algoritmos Culturales: Evolución Cultural, componentes, procedimiento, espacio de creencia, operadores culturales. • CoEv – Coevolución: características, modelo competitivo, modelo cooperativo. • DE – Evolución Diferencial: inicialización, operaciones, selección, DE vs. GA, variantes de DE, <i>Dynamic DE</i> • QIEA – Algoritmos Evolutivos con Inspiración Quántica: Computación cuántica, algoritmos con inspiración cuántica, QIEA-B, QIEA-R 	<ul style="list-style-type: none"> • Reconocer y analizar la necesidad de usar Algoritmos Evolutivos más avanzados • Implementación de modelos avanzados de computación evolutiva
Readings : [RBK12], [EIM+06], [Koz92], [Reynolds94], [SP95], [Cru07]	

9. WORKPLAN

9.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

9.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

9.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

10. EVALUATION SYSTEM

***** EVALUATION MISSING *****

11. BASIC BIBLIOGRAPHY

- [Can00] Erick Cantú-Paz. *Efficient and Accurate Parallel Genetic Algorithms*. Norwell, MA, USA: Kluwer Academic Publishers, 2000. ISBN: 0792372212.
- [Coe07] Carlos A. Coello Coello. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. 2nd Edition. Springer, Sept. 2007.
- [Cru03] André Vargas Abs da Cruz. “Otimização de planejamento com restrições de precedência usando algoritmos genéticos e co-evolução cooperativa”. MA thesis. Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, Feb. 2003. URL: http://www2.dbd.puc-rio.br/pergamum/biblioteca/php/mostrateses.php?open=1&arqtese=5000066121_03_Indice.html.
- [Cru07] André Abs da Cruz. “Algoritmos Evolutivos com Inspiração Quântica para Problemas com Representação Numérica”. (In Portuguese). PhD thesis. Rio de Janeiro, Brasil: Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, Mar. 2007.
- [EIM+06] Tarek A. El-Mihoub et al. “Hybrid Genetic Algorithms: A Review”. In: *Engineering Letters* 13.2 (Aug. 2006). ISSN: 1816-0948. URL: www.engineeringletters.com/issues_v13/issue.../EL_13_2_11.pdf.

- [Fog95] David B. Fogel. *Evolutionary Computation. Toward a New Philosophy of Machine Intelligence*. New York: The Institute of Electrical and Electronic Engineers, 1995.
- [Gol89] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, Massachusetts: Addison-Wesley Publishing Co., 1989.
- [Hol75] John Henry Holland. *Adaptation in Natural and Artificial Systems*. first. Ann Arbor, Michigan: University of Michigan Press, 1975.
- [Koz92] John R. Koza. *Genetic Programming. On the Programming of Computers by Means of Natural Selection*. Cambridge, Massachusetts: The MIT Press, 1992.
- [Mic00] Zbigniew Michalewicz. "Introduction to constraint-handling techniques, Decoders, Repair algorithms, Constraint-preserving operators". In: *Evolutionary Computation 2, Advanced Algorithms and Operators* (2000), pp. 38–40, 49–55, 56–61, 62–68.
- [Mic96] Zibgniew Michalewicz. *Genetic Algorithms+Data Structures = Evolution Programs*. Springer-Verlag, 1996.
- [Mit04] Melanie Mitchell. *An Introduction to Genetic Algorithms: Complex Adaptative Systems*. The MIT Press, 2004.
- [RBK12] Grzegorz Rozenberg, Thomas Bäck, and Joost N. Kok, eds. *Handbook of Natural Computing*. 1st. Springer Publishing Company, Incorporated, 2012. ISBN: 3540929096, 9783540929093.
- [SC00] Alice E. Smith and David W. Coit. "Penalty functions". In: *Evolutionary Computation 2, Advanced Algorithms and Operators* (2000), pp. 41–48.
- [SP95] Rainer Storn and Kenneth Price. *Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces*. Tech. rep. TR-95-012. Berkeley, California: International Computer Science Institute, Mar. 1995.
- [Wei09] Thomas Weise. *Global Optimization Algorithms - Theory and Application*. <http://www.it-weise.de>. 2009.