



# UNIVERSITÀ DEGLI STUDI DI PALERMO

<b>DEPARTMENT</b>	Matematica e Informatica		
<b>ACADEMIC YEAR</b>	2015/2016		
<b>MASTER'S DEGREE (MSC)</b>	COMPUTER SCIENCE		
<b>SUBJECT</b>	HEURISTIC ALGORITHMS		
<b>TYPE OF EDUCATIONAL ACTIVITY</b>	B		
<b>AMBIT</b>	50341-Discipline Informatiche		
<b>CODE</b>	17989		
<b>SCIENTIFIC SECTOR(S)</b>	INF/01		
<b>HEAD PROFESSOR(S)</b>	RIZZO RICCARDO	Professore incaricato esterno	Univ. di PALERMO
<b>OTHER PROFESSOR(S)</b>			
<b>CREDITS</b>	6		
<b>INDIVIDUAL STUDY (Hrs)</b>	102		
<b>COURSE ACTIVITY (Hrs)</b>	48		
<b>PROPAEDEUTICAL SUBJECTS</b>			
<b>MUTUALIZATION</b>			
<b>YEAR</b>	2		
<b>TERM (SEMESTER)</b>	2° semester		
<b>ATTENDANCE</b>	Not mandatory		
<b>EVALUATION</b>	Out of 30		
<b>TEACHER OFFICE HOURS</b>	<b>RIZZO RICCARDO</b> Monday 09:30 13:30 Please send an email to <a href="mailto:riccardo.rizzo@cnr.it">riccardo.rizzo@cnr.it</a> and ask for an appointment that will be online or at my office: National Research Council of Italy, via Ugo La Malfa 153, Palermo --- Per favore mandare una email a <a href="mailto:riccardo.rizzo@cnr.it">riccardo.rizzo@cnr.it</a> per chiedere un appuntamento che puo' essere online o presso il mio ufficio al Consiglio Nazionale delle Ricerche, via Ugo La Malfa 153 Palermo		

<b>PREREQUISITES</b>	
<b>LEARNING OUTCOMES</b>	<p>Conoscenza e capacità di comprensione  Il corso fornisce le conoscenze teoriche e pratiche per ideare ed implementare algoritmi euristici di tipo evolutivo e algoritmi di ricerca locale (Tabu Search, Simulated Annealing, etc.) per la risoluzione di problemi di ottimizzazione ad elevata complessità computazionale.  Inoltre fornisce le conoscenze necessarie per l'uso avanzato di Python che sarà il linguaggio utilizzato per sviluppare tali algoritmi. Gli studenti acquisiranno la capacità di codificare opportunamente i dati di un problema in modo da poterlo risolvere all'interno del frame evolutivo e affineranno le proprie capacità di problem solving.</p> <p>Capacità di applicare conoscenza e comprensione  Il corso ha come obiettivo rendere lo studente capace di comprendere il funzionamento dei principali algoritmi euristici e metaeuristici, fornendo non solo le basi teoriche che giustificano la convergenza, ma anche gli strumenti pratici per la realizzazione di tali algoritmi mediante un linguaggio ormai divenuto uno standard di fatto.</p> <p>Autonomia di giudizio  Gli studenti sono guidati ad apprendere in maniera critica e responsabile tutto ciò che viene spiegato in aula e ad arricchire le proprie capacità di giudizio sia attraverso lo studio del materiale didattico indicato o fornito dal docente, sia attraverso la realizzazione di elaborati pratici consistenti nell'ideazione, implementazione e validazione di euristiche.</p> <p>Abilità comunicative  Attraverso l'interazione durante le lezioni e le attività di laboratorio previste, il corso tenderà a favorire lo sviluppo della capacità di comunicare in modo chiaro ed esauritivo le proprie ragioni ed argomentare le proprie conclusioni. Gli studenti dovranno altresì sviluppare la capacità di lavorare in gruppo, di confrontarsi sulle problematiche proposte rispettando i punti di vista diversi dal proprio e arricchendo le conoscenze acquisite durante il corso con la dialettica e il confronto tra pari.</p> <p>Capacità d'apprendimento  Attraverso approfondimenti e consultazione dei testi di riferimento, gli studenti saranno stimolati ad una conoscenza più approfondita e critica dei linguaggi di programmazione a loro già noti, tramite lo studio di come tali linguaggi possono essere compilati.</p>
<b>ASSESSMENT METHODS</b>	Discussione di un elaborato e prova orale.
<b>EDUCATIONAL OBJECTIVES</b>	<p>Il corso ha il duplice obiettivo di fornire agli studenti la conoscenza del linguaggio Python e di alcuni strumenti di calcolo non convenzionale per la risoluzione di problemi ad elevata complessità.</p> <p>Il linguaggio Python è uno strumento di programmazione molto usato per applicazioni scientifiche in moltissimi ambiti di ricerca per vari motivi: la grande disponibilità di librerie scientifiche, la facilità con cui si presta alla prototipazione rapida e la gratuità del linguaggio, delle librerie e dei principali strumenti di sviluppo. La conoscenza di base del linguaggio è oggetto del corso di "Information Retrieval e Semantic Web", quindi, dopo un breve riepilogo della sintassi e delle caratteristiche di base del linguaggio, verranno trattati alcuni argomenti specialistici collegati alle finalità del corso quali l'uso di librerie particolari, la visualizzazione dei dati, la programmazione ad oggetti. Verrà anche dato un cenno sulla parallelizzazione.  Il corso sarà focalizzato sulla versione Python 2.7.</p> <p>L'obiettivo primario del corso resta comunque quello di fornire gli strumenti sia teorici che pratici per affrontare la risoluzione di problemi di minimizzazione multi-obiettivo ad elevata complessità di calcolo, mediante paradigmi computazionali basati sull'evoluzione di sistemi complessi. In particolare verranno affrontate le tecniche di problem solving basate sugli algoritmi genetici, seriali e paralleli.  L'ottimizzazione di un sistema spesso si riduce alla risoluzione efficiente di un problema di minimizzazione di una o più funzioni, dette funzioni obiettivo. Tuttavia per molti problemi di ottimizzazione questa funzione target si basa su un elevato numero di parametri, spesso in competizione tra loro. Pertanto la ricerca della soluzione ottimale, che potrebbe anche non esistere, risulta comunque, estremamente dispendiosa in termini di potenza e tempi di calcolo. Gli algoritmi evolutivi sono euristiche di ottimizzazione stocastiche, basati sulla ricerca parallela nello spazio delle soluzioni di un dato problema, capaci di determinare soluzioni sub-ottimali a problemi di ottimizzazione complessi (anche multi-obiettivo) in tempi ragionevoli e con basso errore rispetto alle soluzioni ottime degli stessi.</p>

	Al termine del corso lo studente dovrebbe essere in grado di scrivere codice Python efficiente e, nello stesso tempo, possedere gli strumenti necessari per generare un codice prototipale per il Calcolo Scientifico basato sui paradigmi evolutivi.
<b>TEACHING METHODS</b>	Lezioni frontali e in laboratorio.
<b>SUGGESTED BIBLIOGRAPHY</b>	Oltre al materiale didattico fornito dal docente, si suggeriscono:  Swaroop C. H., A Byte of Python (online)  Guido van Rossum Fred L. Drake, Jr. The Python Tutorial. Release 2.7.8 (online)  Zbigniew Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer (1996) - ISBN: 978-3-540-60676-5  Daniel Ashlock - Evolutionary Computation for Modeling and Optimization, Springer (2006) - ISBN: 978-0-387-22196-0

### SYLLABUS

Hrs	Frontal teaching
1	Introduzione Presentazione del corso e del docente.
12	Python Comandi base; variabili e liste; tipi di dato; funzioni built-in; script e funzioni definite dall'utente; funzioni con input e output variabile; strutture di selezione; strutture iterative; lettura e scrittura di file; importazione di moduli esterni e scrittura di moduli; cenni sui moduli esterni disponibili; moduli Scipy e Numpy.
5	Spazi e Algoritmi di Ricerca, Tecniche di Ottimizzazione funzioni obiettivo; funzioni multi-obiettivo; minimizzazione delle funzioni obiettivo; spazio delle soluzioni; Pareto front; convergenza in probabilita ; Classificazione e breve panoramica sugli algoritmi di ricerca locale e sui metaeuristici: Algoritmi di ricerca locale, Ricerca Locale (Local Search o Hill Climbing); Esempi.
10	Fondamenti teorici degli algoritmi evolutivi e genetici Algoritmi stocastici; algoritmi evolutivi; algoritmi genetici; ricerca nello spazio delle soluzioni; codifica dei dati; cromosomi e popolazione; operatore di cross-over su: dati binari, dati reali, stringhe, permutazioni ed alberi; operatore di mutazione su: dati binari, dati reali, stringhe, permutazioni ed alberi; funzione di fitness; funzioni di selezione; modelli generazionali e modelli steady-state; struttura degli algoritmi genetici; complessita computazionale; probabilita di crossover; probabilita di mutazione; fitness landscape; teorema dello schema; algoritmi genetici paralleli; operatore di migrazione; Design e implementazione con Python di algoritmi genetici per risolvere problemi di ottimizzazione anche multi-obiettivo; risoluzione di esercizi.
10	Tabu search: definizioni generali e funzionamento; Implementazione di algoritmi di tabu search; implementazione dei processi di intensificazione e diversificazione; gestione automatica della dimensione della tabu list.
10	Simulated Annealing: definizioni generali e funzionamento, teoria del Simulated Annealing e suo comportamento asintotico; Implementazione di algoritmi di simulated annealing; gestione automatica della velocita di raffreddamento del processo.