



UNIVERSITÀ DEGLI STUDI DI PALERMO

DIPARTIMENTO	Matematica e Informatica		
ANNO ACCADEMICO OFFERTA	2020/2021		
ANNO ACCADEMICO EROGAZIONE	2021/2022		
CORSO DILAUREA MAGISTRALE	INFORMATICA		
INSEGNAMENTO	OTTIMIZZAZIONE EURISTICA		
TIPO DI ATTIVITA'	B		
AMBITO	50341-Discipline Informatiche		
CODICE INSEGNAMENTO	21194		
SETTORI SCIENTIFICO-DISCIPLINARI	INF/01		
DOCENTE RESPONSABILE	RIZZO RICCARDO	Professore incaricato esterno	Univ. di PALERMO
ALTRI DOCENTI			
CFU	6		
NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE	102		
NUMERO DI ORE RISERVATE ALLA DIDATTICA ASSISTITA	48		
PROPEDEUTICITA'			
MUTUAZIONI			
ANNO DI CORSO	2		
PERIODO DELLE LEZIONI	1° semestre		
MODALITA' DI FREQUENZA	Facoltativa		
TIPO DI VALUTAZIONE	Voto in trentesimi		
ORARIO DI RICEVIMENTO DEGLI STUDENTI	RIZZO RICCARDO Lunedì 09:30 13:30 Please send an email to riccardo.rizzo@cnr.it 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 riccardo.rizzo@cnr.it per chiedere un appuntamento che puo' essere online o presso il mio ufficio al Consiglio Nazionale delle Ricerche, via Ugo La Malfa 153 Palermo		

PREREQUISITI	Capacita' di programmazione, di scrivere i passi di un algoritmo a partire da un insieme di equazioni.
RISULTATI DI APPRENDIMENTO ATTESI	Conoscenza e capacita' 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 complessita' computazionale. Inoltre fornisce le conoscenze necessarie per l'uso avanzato di Python che sara' il linguaggio utilizzato per sviluppare tali algoritmi. Gli studenti acquisiranno la capacita' di codificare opportunamente i dati di un problema in modo da poterlo risolvere all'interno di del frame evolutivo e affineranno le proprie capacita' di problem solving. Capacita' di applicare conoscenza e comprensione. ----- Il corso ha come obiettivo rendere lo studente capace di comprendere il funzionamento dei principali algoritmi euristici e metaeuristici, fornendogli 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. Autonomia di giudizio ----- Gli studenti sono guidati ad apprendere in maniera critica e responsabile tutto cio' che viene spiegato in aula e ad arricchire le proprie capacita' di giudizio sia attraverso lo studio del materiale didattico indicato o fornito dal docente, sia attraverso al realizzazione di elaborati pratici consistenti nell'ideazione, implementazione e validazione di euristiche. Abilita' comunicative ----- Attraverso l'interazione durante le lezioni e le attivita' di laboratorio previste, il corso tendera' a favorire lo sviluppo della capacita' di comunicare in modo chiaro ed esaustivo le proprie ragioni ed argomentare le proprie conclusioni. Gli studenti dovranno altresì sviluppare la capacita' 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. Capacita' d'apprendimento ----- Acquisizione di un metodo di studio adeguato, supportato dalla consultazione dei testi e dalla risoluzione di esercizi e quesiti proposti periodicamente durante il corso.
VALUTAZIONE DELL'APPRENDIMENTO	La valutazione sara' divisa in due momenti: una prova scritta ed una prova orale. La prova scritta mira a stabilire se il discente e' capace di applicare le conoscenze apprese alla soluzione di un problema, la prova orale mira a stabilire il grado di approfondimento raggiunto e la capacita' di argomentare la esposizione. Valutazione della prova scritta ----- La prova scritta valuta la capacita' di applicare le conoscenze apprese nella soluzione di problemi pratici e di casi concreti. Il discente dovra dimostrare di avere capito il problema da risolvere, di sapere pianificare una possibile soluzione, sotto forma di algoritmo, e di sapere come implementare tale algoritmo. Essendo la materia di carattere fortemente applicativo, sara' oggetto di valutazione anche la realizzabilita' della soluzione proposta. Valutazione della prova orale ----- L'esaminando dovra' rispondere ad almeno tre domande poste oralmente su tutte le parti del programma, con riferimento ai testi consigliati. La verifica finale valuta se lo studente ha acquisito conoscenza e comprensione degli argomenti e se e' capace di applicare le conoscenze in ordine alla soluzione i casi concreti. Inoltre dovra' essere in grado di esporre ed argomentare in modo da trasferire tali conoscenze efficacemente all'esaminatore. Al di sotto di tale soglia (incapacita' di esporre gli argomenti, incapacita' di applicare le conoscenze nella risoluzione di anche semplici casi concreti) l'esame risulta insufficiente. Quanto piu' l'esaminando riesce ad interagire con l'esaminatore e tanto piu' le sue conoscenze vanno in dettaglio, tanto piu' la valutazione sara' positiva. Esito del voto

	30-30 e lode: Valutazione eccellente/ottimo. Ottima conoscenza degli argomenti, ottima capacita' analitica anche in nuovi contesti; ottima proprieta' di linguaggio e di apprendimento, lo studente ha applicato le nozioni impartite per la soluzione del problema assegnato in maniera originale. 27-29: Valutazione molto buono. Buona padronanza degli argomenti, piena proprieta' di linguaggio, prova scritta che mostra piena capacita' di applicazione delle nozioni studiate. 24-26: Valutazione buono. Conoscenza di base dei principali argomenti, discreta proprieta' di linguaggio, mostra limitate capacita' di programmazione e applicazione delle nozioni acquisite, prova scritta con qualche errore facilmente riconosciuto e corretto. 21-23: Soddisfacente. Parziale padronanza degli argomenti del corso, soddisfacente proprieta' linguaggio, prova scritta non funzionante o con palesi errori che pero' sono riconosciuti e corretti. 18-20: Sufficiente. Minima conoscenza degli argomenti del corso e del linguaggio tecnico, prova scritta non funzionante con errori che sono riconosciuti con palese sforzo e non in autonomia.
OBIETTIVI FORMATIVI	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 complessita'. Il linguaggio Python e' uno strumento di programmazione molto usato per applicazioni scientifiche in moltissimi ambiti di ricerca per vari motivi: la grande disponibilita' di librerie scientifiche, la facilita' con cui si presta alla prototipazione rapida e la gratuita' del linguaggio, delle librerie e dei principali strumenti di sviluppo. La conoscenza di base del linguaggio sara' oggetto della prima parte del corso, quindi, dopo una ntroduzione alla sintassi ed alle caratteristiche di base del linguaggio, verranno trattati alcuni argomneti specialistici collegati alle finalita' del corso quali l'uso di librerie particolari, la visualizzazione dei dati, la programmazione ad oggetti. Verra' anche dato un cenno sulla parallelizzazione. 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 complessita' 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 piu' 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. 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.
ORGANIZZAZIONE DELLA DIDATTICA	Lezioni frontali in aula attrezzata gli studenti avranno a disposizione dei computer per seguire le lezioni in modo interattivo.
TESTI CONSIGLIATI	Materiale di riferimento Dispense fornite al corso e rese disponibili on line Testi consigliati ----- Linguaggio Python Swaroop C. H., A Byte of Python (online) Guido van Rossum Fred L. Drake, Jr. The Python Tutorial. (online) S.J. Russel P. Norvig Intelligenza Artificiale - Un approccio moderno, Pearson

PROGRAMMA

ORE	Lezioni
1	Introduzione Presentazione del corso e del docente.
6	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;
6	Spazi e Algoritmi di Ricerca, Tecniche di Ottimizzazione funzioni obiettivo; funzioni multi-obiettivo; minimizzazione delle funzioni obiettivo; spazio delle soluzioni; Pareto front; convergenza in probabilit�; Classificazione e breve panoramica sugli algoritmi di ricerca locale e sui metaeuristici: Algoritmi di ricerca locale, Ricerca Locale (Local Search o Hill Climbing); Esempi.
6	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 crossover 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;
6	Modelli generazionali e modelli steady-state; struttura degli algoritmi genetici; complessita' computazionale; probabilit� di crossover; probabilit� 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.
5	Tabu search: definizioni generali e funzionamento; Relazione e confronto con algoritmi di ricerca locale; Implementazione di algoritmi di tabu search;
5	Implementazione dei processi di intensificazione e diversificazione; gestione automatica della dimensione della tabu list
6	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
2	Analisi degli metodi euristici basati sulla popolazione. Algoritmi (λ , μ)
5	Particle swarm optimization: definizione dell'algoritmo, topologia dello sciame, modifiche all'algoritmo originale. Discussione della implementazione. Cenni sulla Differential evolution