



Programma del Corso integrato "Ricerca operativa e Ottimizzazione"

Il corso e' composto da 2 moduli: 1) Ottimizzazione Combinatoria, 2) Ricerca Operativa

Programma del Modulo "Ottimizzazione Combinatoria"

- Codice: F0141
- Tipo di corso: Obbligatorio (Laurea in Informatica percorso Generale)
- Livello del corso: Lauree di Primo Livello
- Semestre: 2

Numero di crediti ECTS: (Laurea in Informatica) 6 (carico 150 ore)

Docenti: Claudio Arbib (Claudio.Arbib@univaq.it)

1	Obiettivi del corso	<p>Apprendere tecniche algoritmiche per alcuni problemi di Ottimizzazione Combinatoria. Saper formulare e risolvere problemi di Ottimizzazione Combinatoria come problemi di Programmazione Lineare Intera. Conoscere la complessità dei problemi studiati.</p>
2	Contenuti del corso e risultati formativi (descrittori di Dublino)	<p>Gli argomenti trattati nel corso comprendono:</p> <ul style="list-style-type: none"> • Prerequisiti: grafi, definizioni e proprietà fondamentali. Elementi di teoria della complessità computazionale. Esempi. Problemi di Ottimizzazione Combinatoria. Definizioni fondamentali. Esempi: trasversale (vertex-cover), insieme stabile, insieme dominante, edge-cover, matching (perfetto), knapsack, colorazione di un grafo (numero e indice cromatico), cammino minimo, sottografo ricoprente etc. Formulazioni come programmazione lineare 0-1. • Matrici unimodulari e totalmente unimodulari. Condizioni necessarie e condizioni sufficienti. Teorema di Hoffmann-Kruskal. Involucro convesso e formulazione ideale di un problema di PL intera. Esempi: cammino minimo, flusso a costo minimo, matching bipartito, grafi intervallo. • Cammini su grafi diretti aciclici (DAG) e programmazione dinamica. Esempi di applicazione: cammini ottimi su DAG, knapsack 0-1. [Proiezione di poliedri, sistemi di disequazioni lineari, Teorema di Fourier-Veronese, metodo di Fourier-Motzkin. Richiami di teoria della dualità nella programmazione lineare. Metodo primale duale. Esempio di applicazione: Metodo di Dijkstra]. • Matroidi. Algoritmo Greedy. Definizione e caratterizzazione di un matroide (Teorema di Rado). Richiami di algebra lineare: dipendenza e indipendenza lineare, basi, unicità della rappresentazione, Teorema di Steinitz o dello scambio. Esempi: matroide banale, matroide grafico, matroide partizione, matroide vettoriale, [matroide matching]. Rappresentazione di matroidi. [Un applicazione industriale]. [Funzione rango, funzioni submodulari e supermodulari. Rango di un matroide. Funzione di supporto di un insieme convesso. Poliedro submodulare. Estensione di Lóvasz. Poliedro di un matroide, intersezione di matroidi, intersezione di due poliedri submodulari]. • Algoritmi ad approssimazione garantita. Definizioni. Schemi polinomiali di approssimazione (PTAS, EPTAS, FPTAS). Esempi: TSP: double tree and Christofides; Knapsack 0-1. • Matching (perfetto/non perfetto, pesato/non pesato, bipartito/non bipartito). Disequaglianze duali deboli: trasversale, numero di stabilità, edge-cover e matching. Il caso bipartito: Teoremi di Gallai e König. [Matrici bistocastiche e matching bipartito perfetto. Matching generale: poliedro del matching, Teorema di Edmonds]. • Rilassamento lineare di un problema di programmazione lineare (mista) intera. Enumerazione implicita: metodo di branch-and-bound. Bound lineari, esempio: Knapsack intero, Knapsack 0-1. Dicotomia su una variabile frazionaria. Bound combinatorici, esempio: TSP. • [Separazione di un poliedro intero. Cenni sul Metodo dell'Ellissoide. Separazione e ottimizzazione: Generazione di disequaglianze valide. Soluzione di PL con un numero esponenziale di disequazioni, esempi: Taglio minimo, TSP, Knapsack 0-1, Insieme Stabile].

		<p>Alla fine del corso, lo studente dovrebbe:</p> <ul style="list-style-type: none"> • Saper valutare la complessità dei problemi di ottimizzazione combinatoria, saperli formulare come programmazione lineare 0-1, conoscere le fondamentali relazioni primale-duale. Disporre delle nozioni principali sulla teoria dei Matroidi, sulla programmazione ricorsiva, sulle matrici unimodulari, sulla complessità computazionale di algoritmi e problemi. Conoscere gli algoritmi standard per il matching bipartito e il percorso minimo. Conoscere euristiche per il TSP e lo Knapsack. Conoscere algoritmi generali di enumerazione implicita (branch-and-bound). • Comprendere la differenza fra un problema "facile" e uno "difficile". Valutare la complessità di un algoritmo e, per casi semplici, di un problema. Saper riconoscere se una matrice è o no totalmente unimodulare. Risolvere con algoritmi standard problemi di albero ricoprente, matching bipartito, knapsack, percorso ottimo, TSP. Formulare problemi di ottimizzazione combinatoria o problemi di ottimizzazione binaria derivati da applicazioni in termini di programmazione lineare 0-1. • Distinguere quando un problema ha forma matroidale e riconoscerne il vantaggio. Sapere qual è il vantaggio offerto da una matrice totalmente unimodulare. Sapere quando è opportuno risolvere un problema per programmazione dinamica. Conoscere i limiti delle euristiche. • Dimostrare rigorosamente un semplice teorema. Dimostrare o confutare (con un controesempio) una semplice congettura. Comprendere il ruolo dell'ottimizzazione combinatoria nelle applicazioni. • Rivolgersi allo studio di funzioni submodulari, combinatorica poliedrale, metodi avanzati per la programmazione lineare intera.
3	Prerequisiti	Conoscenza degli algoritmi di base, nozioni di algebra lineare, programmazione lineare
4	Modalità e lingua di insegnamento	<p>Lezioni, esercitazioni</p> <p>Lingua: Italiano</p> <p>Testi/Bibliografia</p> <ul style="list-style-type: none"> • W.J. Cook, W. H. Cunningham, W. R. Pulleyblank, A. Schrijver, Combinatorial Optimization. Wiley. 1997. • C.H. Papadimitriou, K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity. Dover Books on Computer Science. 1998. http://www.amazon.com/Combinatorial-Optimization-Algorithms-Complexity-Computer/dp/0486402584 • A. Sassano,, Modelli e Algoritmi della Ricerca Operativa. Franco Angeli Editore.
5	Metodi di accertamento	Prova scritta e orale
Programma del Modulo "Ricerca Operativa"		
<ul style="list-style-type: none"> • Codice: F0140 • Tipo di corso: Obbligatorio (Laurea in Informatica percorso Generale) • Livello del corso: Lauree di Primo Livello • Semestre: 2 		
Numero di crediti ECTS: (Laurea in Informatica) 6 (carico 150 ore)		
Docenti: Stefano Smriglio (Stefano.Smriglio@univaq.it)		
1	Obiettivi del corso	Mettere lo studente in grado di formulare e risolvere problemi di ottimizzazione di base, in particolare problemi di programmazione lineare.
2	Contenuti del corso e risultati formativi (descrittori di Dublino)	<p>Gli argomenti trattati nel corso comprendono:</p> <ul style="list-style-type: none"> • Problemi di Ottimizzazione: variabili di decisione, vincoli e obiettivi; tecniche di formulazione e classificazione dei modelli • Problemi di Ottimizzazione convessa; punti di ottimo locale e globale • Geometria della Programmazione Lineare • Il metodo del simplesso • Teoria delle dualità in Programmazione Lineare e sue applicazioni • Interpretazione duale del metodo del simplesso e metodo del simplesso duale <p>Alla fine del corso, lo studente dovrebbe:</p> <ul style="list-style-type: none"> • Acquisire la conoscenza dei problemi di Ottimizzazione e delle tecniche di

		<p>modellazione matematica dei problemi decisionali. Acquisire la conoscenza di alcuni algoritmi di soluzione di problemi di Programmazione Lineare.</p> <ul style="list-style-type: none"> • Acquisire la capacità di riconoscere i problemi di Ottimizzazione e sviluppare modelli matematici di problemi decisionali. Acquisire la capacità di calcolare soluzioni di problemi di Programmazione Lineare • Acquisire autonomia nelle scelte modellistiche ed algoritmiche relative a problemi decisionali complessi • Essere in grado di sostenere una conversazione e di leggere testi su argomenti riguardanti la modellazione di problemi decisionali e la Programmazione Lineare • Acquisire attitudini all'aggiornamento flessibile di conoscenze e competenze nel campo dell'Ottimizzazione e dei problemi di Ottimizzazione che sorgono in varie aree, quali la matematica, l'informatica e l'ingegneria gestionale
3	Prerequisiti	Spazi vettoriali, prodotto scalare, prodotto tra matrici, matrici inverse
4	Modalità e lingua di insegnamento	<p>Lezioni frontali Lingua: Italiano Testi/Bibliografia</p> <ul style="list-style-type: none"> • Dimitris Bertsimas and John N. Tsitsiklis, <i>Introduction to Linear Optimization</i>. Athena Scientific. 1997. • Matteo Fischetti, <i>Lezioni di Ricerca Operativa</i>. Progetto Libreria Padova. 1995. • Antonio Sassano, <i>Modelli e Algoritmi della Ricerca Operativa</i>. Franco Angeli. 1992.
5	Metodi di accertamento	<p>1. una prova scritta consistente in un numero variabile di esercizi (formulazione di problemi, esercizi su proprietà algebriche o geometriche dei problemi, risoluzione mediante algoritmi noti); 2. un colloquio orale su argomenti teorici, per sostenere il quale lo studente deve aver conseguito un giudizio sufficiente alla prova scritta Avvertenza importante: una prova scritta sufficiente dà diritto a sostenere la prova orale nel medesimo appello, ma non in appelli successivi. A tale regola non sono ammesse deroghe di alcun tipo. Nel corso dell'anno è prevista una prova intermedia, il cui esito, se sufficiente, può essere utilizzato dallo studente a titolo di esonero dalla prova scritta, limitatamente alla relativa parte di programma. Nella carriera di uno studente la prova intermedia può essere sostenuta una sola volta.</p>