Ingegneria Informatica Magistrale (2023)

72935 - Operations Research M

PDF, ADOC.

Learning outcomes

Mathematical models, linear programming, simplex algorithm, duality. Integer linear programming, cutting planes, branch-and-bound. Complexity. Dynamic programming. Discrete simulation.

Course contents

Course contents

Prerequisites:

It is required that the student understands spoken Italian in order to follow the lectures.

MODULE 1 (Mathematical Optimization): Silvano Martello

1. The scientific method: systems, models, methodologies
2. Mathematical Programming

2.1 Optimization problems

2.2 Convex sets and functions

2.3 Convex programming

1. Linear programming

3.1 General, canonical and standard forms

3.2 Bases e basic solutions

3.3 Convex polytopes

1. Simplex algorithm

4.1 Moving among basic solutions

4.2 Tableau and pivoting

4.3 Optimality criterion

4.4 Simplex algorithm

4.5 Two-phase method

4.6 Geometrical aspects

1. Duality

5.1 Dual of a linear programming problem

5.2 Duality properties

5.3 Farkas' lemma

5.4 Complementary slackness

5.5 Dual simplex algorithm

5.6 Sensitivity analysis

1. Integer linear programming

6.1 Unimodularity

6.2 Cutting-plane algorithms and Gomory cuts

6.3 Branch-and-bound algorithm

6.4 Exploration strategies

6.5 0-1 knapsack problem

6.6 Branch-and-cut algorithms

6.7 Software and freeware

1. Complexity theory

7.1 Recognition version

7.2 P and NP classes

7.3 NP-complete problems

7.4 Dynamic programming

7.5 Strong NP-completeness

MODULE 2 (Discrete Simulation): Andrea Lodi

Introduction to discrete simulation

Static and dynamic description of a system

Temporary and permanent entities

Events and event notices.

Exercises on modeling realistic systems.

72937 - Computer Architecture M

PDF, ADOC.

Learning outcomes

Focus is on quantitative aspects of computer architecture.

Expected outcome is understanding of:

instruction level parallelism and latency tolerance

memory hierarchy

Uniform Memory Access (UMA) architectures

protection and task management

architecture impact on power and performance at processor and system level

Within the course the student will practice abstraction as the most effective method to handle the complexity of modern computer architectures. The final test verifies the ability to apply methods and concepts offered by the course.

Course contents

Processor Architecture:

Instruction set architecture for multitasking protected systems (Intel IA32 architecture)

Instruction level parallelism; dependencies; hazards; hazards avoidance

superscalar architectures; superpipelining and underpipelining; non blocking architectures; out of order execution (Tomasulo approach); speculative execution;

Introduction to simultaneous multithreading, logical processors, multicore architectures and virtualisation

Memory Hierarchy

Goals, reference model and performance parameters

memory hierarchy in the Intel IA32 architecture: segmentation, virtual memory, main memory and caches

address mapping, write policies, replacement policies; implementation techniques and performance analysis

the MESI protocol for memory coherence in shared memory multiprocessor architecture (UMA architectures)

System Architecture

Interleaved memory access in multibyte data bus systems

Uniform Memory Access (UMA) architectures, and introduction to NUMA multicore based architectures

Bus hierarchy and bus protocols

Multimaster systems with DMA based I/O: hardware/software power/performance perspective

72940 - Computational Models And Languages M

PDF, ADOC.

Learning outcomes

At the end of this course unit, conceived according to a multi-paradigm and multi-language constructive approach, the student has a deep knowledge on the fundamental concepts of programming languages and related computational models, knows the foundations of computability, knows and is able to apply the basic interpreter and compiler techniques, and possesses the basic concepts of functional programming and is able to apply such concepts in typical practical situations.

More precisely, the student will know the main formal methods for language definition in terms of syntax and semantics, for both programming languages and specification languages, and will be able to apply the main techniques for language evaluation and recognition for interpreters and compilers, including the use of the most common tools. Students will also be able to define reasonably simple languages understanding their properties, implementing the corresponding interpreters, and evaluating the pros and cons of different choices/computational paradigms in application design.

Course contents

The course aims to provide a rational view over the fundamental concepts of programming languages, relating them to the different computational models and to the problem of language translation and recognition: solid foundations are coupled to a strong experimental approach.

Contents:

Computability and Turing machine (5hrs)

Formal description of programming languages: grammars and Chomsky classification. Relationship between grammars and language interpreters/translators: lexical analysis, top-down and bottom-up techniques for the syntactical analysis of regular languages and context-free languages. Overview on methods for the formal description of the semantic aspects of a language. (18hrs)

Structure and organisation of interpreters/compilers, and their run-time support: examples in Java. Parser generator tools. (14hrs)

Introduction to Model-Driven approaches: the Xtext case. (3 hrs)

Iterative vs. recursive computational models, tail recursion optimisation, basic concepts of functional programming, closures, models for function evaluation, intro to the basics of Lambda calculus (9hrs)

Javascript as an example of (dynamic) functional language with a prototype-based object model (6hrs)

Scala and Kotlin as notable examples of blended programming languages on the Java platform. (9hrs)

72942 - Information Security M

PDF, ADOC.

Learning outcomes

Knowledge and engineering skills related to the design, development and deployment of algorithms and protocols for securing systems and networks.

Course contents

The aim of the course is to provide an in-depth study of the models, systems and mechanisms for securing processing systems with both a theoretical and a practical focus.

Suggested background: to gain more from the course it is important to have clear the concepts and tools provided by the computer networks, operating systems and computer security laboratory courses.

The course contents are divided into five macro-areas:

1. Modern cryptography applied

Insights and pitfalls in using PRNG, stream ciphers and block ciphers, cryptographically secure hash functions, asymmetric ciphers

Examples of attacks based on the incorrect use of ciphers and correct methods of use

Examples of cryptographic applications in some scenarios (wireless networks, cloud, IoT, ..)

Symmetric and asymmetric cryptographic key management models and systems (Key distribution

center, PKI, PGP)

1. Authentication Models and Systems

Recalls on authentication systems and principles of designing secure authentication protocols

Single Sign-on authentication models with related examples of protocols / systems (Kerberos, ...)

Federated authentication models with relative examples of protocols / systems (Oauth, OpenID, SAML, ..)

1. Access control models and systems

Identity-based models with related examples of protocols / systems

Role-based models with related examples of protocols / systems

Attribute- and context-based models with related examples of protocols / systems

1. Automated security management models and systems: paradigm based on policy-based management
2. Blockchain technologies

Principles of operation

Hints of operation of the Bitcoin and Ethereum platforms

72943 - Digital Systems M

PDF, ADOC.

Learning outcomes

The corse provides an introduction to modern software methodologies for the design of embedded systems. Application contexts and projects are mostly concerned with image processing and deep-learning networks applied to computer vision problems.

Course contents

Design methodologies for embedded systems with particular emphasis on computer vision applications and deep-learning.

72945 - Real-Time Systems M

PDF, ADOC.

Learning outcomes

The course aims to provide a rational view of the main issues involved in the design of real-time systems, emphasizing peculiar principles, implementation strategies, supporting environments and performance evaluation methods.

Course contents

An introduction to real-time systems: peculiar features and typical application domains. Hard vs. soft real-time tasks. Temporal parameters and reference models for periodic, sporadic and aperiodic tasks. Real-time scheduling approaches: the clock-driven approach and the priority-driven approach. Algorithms for scheduling hard real-time periodic/sporadic tasks: RM, DM, EDF and LSTF. Server-based strategies for optimizing the response time of soft real-time aperiodic tasks: polling server, deferrable server, priority exchange server, sporadic server, constant utilization server, total bandwidth server. Shared resources access protocols: priority inheritance protocol, priority-ceiling protocol, immediate priority-ceiling protocol, stack resource policy. Pre-run-time schedulability analysis of systems that schedule tasks using static or dynamic priorities: processor-utilization analysis, response-time analysis, processor-demand analysis. Exemplification of theoretical and methodological issues with reference to design patterns typical of the industrial automation application domain.

72938 - Foundations Of Artificial Intelligence T

PDF, ADOC.

Learning outcomes

Introduction to main principles and methods in Artificial

Intelligence. Artificial Intelligence based methodologies and

techniques for solving problems with particular emphasis on

knowledge-based systems and computational logic techniques. Desing

and implementation of some practical systems based on procedural

and declarative programming languages.

Course contents

The course introduces the student to the principles and methods

used to solve Artificial Intelligence methods, with a particular

attention to knowledge-based systems and computational logic

approaches. In particular, the Prolog programming language is introduced as a tool for implementing Artificial Intelligence systems. Moreover, seminars on specific Artificial Intelligence topics are planned. This course is preparatory for the course of Intelligent Systems.

Prerequisites: attending the course requires a medium knowledge of a high level programming language, in order to successfully understand case studies and applications presented during the lessons. Regarding the course contents, no prerequisites are required: the student will be gradually introduced to the fundamental notions of the Artificial Intelligence, and no assumption about previous knowledge is made.

Contents:

Introduction to Artificial Intelligence: brief history of AI, main application fields, introduction to knowledge-based systems and architectural organization.

Problem solving in AI: representation through the notion of state, forward e backward reasoning, solving as a search and search strategies (informed and non). Games, constraint satisfaction problems, and planning problems.

Knowledge Representation: First Order Predicate Logic, Production Rules Systems, Knowledge-based systems, Some hints about formal ontologies.

Languages for Artficial Intelligence. Prolog: from logic to logic programming, Prolog programs as solvers, desing and development of simple Prolog programs, few notes about meta-predicates and meta-interpreters.

84531 - Infrastructures For Cloud Computing and Big Data M

PDF, ADOC.

Learning outcomes

The class tends to enhance the capacity of orientating in the process of defining the strategies for a distributed system and applying them in different related applications.

The students face the principles and the main problems of distributed large systems and are exposed to some standard and widely solutions, by following the class and via individual work. At the end, students are expected to be able to know the properties of most diffused middleware and the evolutions one can expect from that technology, by mastering the properties for designing a real application: most well spread strategies are presented and discussed.

Course contents

The course covers several topics central in modern global data and processing infrastructures, such as Data Centers, MultiCloud Systems, Federation of resources, etc. typically supporting Industry 5.0 and Smart city applications, via the following basic concepts:

Advanced models for large distributed & cloud systems, from C/S to message exchange. Particular importance is given to the distinction and the implications of the two models,

Replication, group and many-to-many communication, and systems for QoS

Middleware for development and management of large distributed & cloud systems

Infrastructures for global data storage and processing

Specific modern systems for big data processing and scalable global supports

All above issues are basic topics on which students must be aware of and very competent of, obtained by deep reflections and personal considerations.

The class explores the following topics:

Advanced models for large distributed & cloud systems

Class Starting: general information and presentation of the Class (use cases)

Goals, Basics, and Models: classifications, C/S vs. Message exchange, service and cloud models, parallelization models

Middleware & Cloud Models: definitions, categories, basic organization, and patterns for large distributed and cloud systems, Cloud internals design.

Replication, group and many-to-many communication, and systems for QoS

Different consistency degrees and impact on service properties (BASE and CAP)

Replication: models, strategies and protocols

Communication and groups: models, protocols and algorithms

Systems and protocols for QoS

Multicast and MOM middleware

Middleware for large distributed & cloud systems

CORBA: middleware and operating environment

MOM: examples of very thin environments

OpenStack: an example of a widely-diffused cloud IaaS

Novel infrastructures for global data storage and processing

Global data storage: solutions for non-traditional NoSQL data memorization (Cassandra and MongoDB)

Global data processing: batching and streaming based big data processing (Map-Reduce, Spark, and Storm and S4)

Main properties for effective design projects

72939 - Software Systems Engineering M

PDF, ADOC.

Learning outcomes

Detailed knowledge about languages, models and technologies for the analysis, the development, the documentation, the deployment and the maintenance of (distributed) software systems.

Course contents

At the end of the course, the student:

is able to set-up cooperative software production processes, based on agile (SCRUM) development that exploit also executable models exprressed using custom meta-models;

is able to design and develop software systems with related testing plans, in an incremental and evolutive way, by starting from the problem and from the application domain rather than from the implementation technology, also by using executable models of requirement and problem analysis;

is able to critically evaluate the continuos evolution of software technologies, both as regars the computational aspects and the software development process, by operationally acquiring knowledge on languages, methodologies and tools such as Kotlin, Gradle, SCRUM, SpringBoot,

DevOps, Docker, etc.

is able to understand the role of the different styles of software architectures (layers, client-server, pipeline, microkernel, service-based, event-driven, space-based, microservices) and how to select the most appropriate architectural style for each different sub-system;

is able to face the analysis, the design and development of distributed, heterogeneous proactive-reactive applications (together with related development platforms and run-time supports) with particular reference to computational models based on message-passing and event-driven paradigms;

is able to realize message-based interactions among distributed software components by using high-level logical models and implementations based on different protocols (TCP, UDP, HTTP, CoAP, MQTT);

is able to understand how it is possible to design and build software development environments able to automatic code generation (Sofware Factories in ecosystems like Eclipse and IntelliJ) based on Model Driven Software Development (MDSD) and on Domain Specific Languages (DSL);

is able to develop application able to combine high-level aspects (with particular reference to AI) with low-level aspects related to Internet of Things (IOT) devices, in the context of both virtual and real environments, built by using low-cost computers like RaspberryPi and Arduino;

is able to apply the concepts, the devices and the tools developed in the course for the design and development of a final application that exploits one or more IOT devices - in particular Differental Drive Robots (DDR) with sensors - that can operate in relatively autonomous way in different virtual or real enviroments, without modifying the software that does express the business logic of the problem.

72947 - Operating Systems M

PDF, ADOC.

Learning outcomes

Knowledges of main design aspects concerning the organization of concurrent systems. Models for sincronization and communication between processes/threads. Methods for analysis and synthesis of concurrent systems.

Course contents

1.System protection and security

models, policies and mechanisms

multilevel security

Reference Monitor and trusted systems

2.Virtualization

hardware virtualization: goals and solutions

virtual machine monitor implementation

Case study analysis: xen hypervisor

Virtualization and cloud computing

3.Concurrent programming

preliminaries

non sequential processes.

forms of interaction between concurrent processes

architectures and languages for concurrent programming

4.Shared memory model

mutual exclusion

semaphores

monitors

conditions

Use of concurrent languages in the shared memory model. The pthread library for concurrent programming.

5.Message passing model

preliminaries

channels and communication primitives

send and receive

guarded commands

Rendez-vous and RPC

Use of concurrent languages in the message passing model: go, ada.

6.Shared memory kernel

Implementation of thread management/synchronization in a mono-processor kernel

Implementation of thread management/synchronization in a multi-processor kernel: SMP, loosely-coupled kernels.

1. Distributed Systems

distributed programming: centralized and decentralized algorithms.Scalability, fault tolerance.

algorithms for the synchronization of distributed processes: logical clocks, distributed mutual exclusion, election algorithms.

1. Parallel Programming

Parallel architectures, HPC systems.

Architetture per il calcolo parallelo. Sistemi HPC.

Parallel programming models: shared memory and distributed memory.

Parallel software development: MPI and OpenMP libraries.

Outlines on CUDA programming.

72957 - Design Activity joined to Computer Architecture M

PDF, ADOC.

Learning outcomes

Apply the knowledge acquired in the Computer Architecture M course to independently carry out an in-depth activity on a topic agreed with the lecturer in charge of the course.

Course contents

Projects of power management of systems and processors for high-performance computing systems.

Projects of RISC-V based computing architectures and low level software layers (Linux, RTOS, NUTTX, FreeRTOS, ROS, Compilers) for Cyber Physical System (CPS) applications.

Projects of RISC-V based computing architectures and low level software layers to support zero-trust computing.

Neuromorphic / braininspired computing architecture projects and their use in Cyber Physical System (CPS) applications.

72975 - Software Systems Engineering Project Work M

PDF, ADOC.

Learning outcomes

In this module we apply the abilities achieved in the course 34791 - INGEGNERIA DEI SISTEMI SOFTWARE LM for the devolpment of some specific argument or project

Course contents

Development of a software project

72980 - Project Work For Computational Models And Languages M

PDF, ADOC.

Learning outcomes

The aim of this activity is to apply the knowledge, metholodologies and tools learned in the main module to a practical project, either proposed by the student and validated by the teacher, or proposed by the teacher ad accepted by the student.

Course contents

The project can deal with any aspect in the Languages and Computational Models area, including the development of multi-paradigm applications or the contribution to research activity in the area. The activity must state explicitly the initial requirements, provide an in-depth analysys of the problem, discuss the project and the architecture of the proposed solution (including the evaluation of possible alternatives and the motivation(s) for choosing the specific one), up to the implementation choices, testing methodologies and techniques. A final comprehensive demo is expected and adequately evaluated.

72998 - Project Work For Information Security M

PDF, ADOC.

Learning outcomes

Apply the knowledge acquired in the Information Security M course to independently carry out an in-depth activity on a topic agreed with the lecturer in charge of the course.

Course contents

The contents of the project activity concern topics related to the field of IT security including security of mobile systems, IoT systems, blockchain and cloud / edge

73000 - Project Work For Digital Systems M

PDF, ADOC.

Learning outcomes

Project development with platform and software tools introduced in Digital System M

Course contents

Development of a project with an embedded system

97261 - Project Activities of Technologies and Systems for Database Management and Big Data M

PDF, ADOC.

Learning outcomes

Autonomous exploitation of knowledge obtained in the course "Technology and systems for the management of Databases and Big Data M" in a technological application on a theme in agreement with the teacher.

Course contents

The project can deal with any practical application about the contents presented in the course "Technology and Systems for the Management of Databases and Big Data M. In general, the project can be also about any topic that can be referred to the field of data management. In general, the subject of the activity must have a practical flavor, i.e. the activity must focus on the practical application of principles, methods and techniques acquired within the course. The subject of the project activity must be discussed and agreed upon with the teacher before the starting of the activity itself.

72968 - Project Work For Foundations Of Artificial Intelligence M

PDF, ADOC.

Learning outcomes

The aim of this activity is to apply the knowledge and tools

acquired in the associated course of "Fondamenti di Intelligenza Artificiale M"/"Fundamentals of Artificial Intelligence M" to a practical project. The topic of the project is proposed by the student and must be agreed with the teacher.

Course contents

The project can deal with any practical application about the contents presented in the course "Fondamenti di Intelligenza Artificiale M"/"Fundamentals of Artificial Intelligence M". In general, the project can be also about any topic that can be referred to the AI. In general, the subject of the activity must have a practical flavour, i.e. the activity must focus on the practical application of principles, methods and techniques acquired within the course. The subject of the project activity must be discussed and agreed upon with the teacher before the starting of the activity itself.

72994 - Project Work For Operations Research

PDF, ADOC.

Learning outcomes

Application of the methodologies acquired in the course "Operations Research M" to the development of an autonomous activity on a theme agreed with the teacher.

Course contents

Implementation and experimental testing of algorithms for combinatorial optimization problems and for real-life applications.

73004 - Real Time Systems Design Activities M

PDF, ADOC.

Learning outcomes

Apply the knowledge acquired in the Real-Time Systems M course for the autonomous development of an in-depth activity on a topic agreed with the lecturer.

Course contents

Typical, although not exclusive, topics related to this activity are:

1) development and implementation of algorithms for scheduling hard/soft real-time tasks in a single- or multi-processor platform;

2) design and implementation of the infrastructure necessary to support the application of sophisticated access protocols to shared resources;

3) development of suitable tools for a priori verification of the schedulability of real-time applications comprising an arbitrary number of tasks and shared resources.

The experimental evaluation of the performance actually achievable with the developed prototype solution represents an essential aspect of the design activity.

84533 - Project Work on Infrastructures for Cloud Computing and Big Data M

PDF, ADOC.

Learning outcomes

The project is assigned on an individual base, with a negotiation with the student to identify an area at the state of the art within the field of large middleware systems. At the end of the project, the students have acquired a deep knowledge of the area and also capacity in understanding the global solution strategies. They have to apply the notions and skills acquired in the related class toward a design of a typical (limited) system with some predefined requirements and specific significant technology. The entire execution life cycle is stressed and some performance tests must be pursued.

Course contents

The choice of the topics to organize the project around is part of the final evaluation, the same as the technical repost given in at the end and the presentation of it (slides for project presentation).

The areas to be chosen can be (other areas can be negotiated):

Middleware and service management

Cloud Computing

Scalable global Computing

Federated Data Centers

Multicloud

Serverless comnputing and FaaS

IoT Cloud

Sustainable Infrastructures for Smart Environments

Smart city Infrastructures

Industry 5.0

Some more detailed examples:

Observability of Serverless computing

Serverless computing represents a novel Cloud Paradigm fostering a loss of control of computational resources for customers that can then focus on developing business logic. In this context, observability frameworks seek evolution in order to provide needed insights to customers to develop, deploy and optimize their services while leaving orchestration and management duties to cloud providers.

Event Mesh

The ever-increasing reliance on IT services of many businesses and realities shaping our society is rapidly increasing the number of IT services deployed opening many challenges for their orchestration and integration. Event mesh architectures aim to relieve this complexity by providing built-in security, observability, and management mechanisms of services deployed and asynchronous, uncoupled, and reliable communication among them.

Data Management and Data Mesh

The increasing pervasivity of devices and services continuously producing data, such as IoT and mobile devices, is opening many new possibilities and enabling a rapid evolution of our society. However, the intrinsic distribution of these data rises major challenges in data management including data ownership management, data availability, and data resiliency. In this context, the Data Mesh approach proposes both technical and organizational solutions supporting data management in highly distributed and heterogeneous scenarios.

29206 - Innovation and Project Management M

PDF, ADOC.

Learning outcomes

Students will learn the key concepts and techniques concerning the strategic management of innovation and the effective management of new product development processess. Students will also learn the key methods and techniques concerning project management

Course contents

PREREQUISITES:

The understandings of the main concepts and techniques of general management represents an important prerequisite for the course (in particular for what concerns the techniques for the financial valuation of investments). The course will be held in Italian, however foreign students (i.e. Erasmus students) could complete the project work and give the final exam in English

MAIN TOPICS:

Strategic management of innovation

Innovation development projects: characteristics and specificities

Forms and models of innovation

Technological discontinuities and sector dynamics: analysis of technological trends

The diffusion curves of innovations in the market

Intellectual property management in innovation projects

The development of the innovative project plan

The project plan

Industry analysis

Understanding of customer needs and Design Thinking

Approaches for data searches

Business model

The management of innovative projects

Project management: definitions and key principles

The life cycle of projects

The organizational structures for project management

The role of the project manager and the project team

The start of the project

The stakeholders of the project

Project planning: purpose management and WBS

Project planning: time management

Plan, assign and control project resources

Project risk management

Project monitoring and control

Agile Project Management and SCRUM

72971 - Innovation And Project Management Project Work M

PDF, ADOC.

Learning outcomes

Apply the knowledge acquired in the course of "Gestione dell'innovazione e dei progetti LM" to carry out autonomously an in-depth activity on a topic agreed with the teacher in charge of the course.

Course contents

1.- New product development projects: key concepts.

1. – Idea generation: internal and external sources
2. – Idea selection and evaluation: key techniques.
3. – New product development: key decisions and best practices.
4. – Testing, validation and commercial launch of new products
5. – Project management: Concepts and techniques.

Readings

78098 - Project Work On Network Optimization M

PDF, ADOC.

Learning outcomes

Application of the methodologies acquired in the course "Network Optimization M" to the development of an autonomous activity on a theme agreed with the teacher.

Course contents

Implementation and experimental testing of algorithms for network optimization problems.

73002 - Distributed System Design Activities M

PDF, ADOC.

Learning outcomes

To apply the skills, competences,and abilities deriving

from the course "Distributed Systems M" in order to autonomously

complete a design/implementation project activity related to a

specifictopic previously agreed with the teacher and then

investigated in-depth, from both the analysis and project-oriented

points of view.

During and after course it will be possible to decide the

project topic by contacting the teacher via email:

paolo.bellavista@unibo.it

The results of the project (code + documentation) should be

provided to the teacher at least 5 days before the oral

examination.

Course contents

See the syllabus and topic coverage of the course

"Distributed Systems M", and also you can refer to the web pages of the previous academic year:

http://lia.disi.unibo.it/Courses/sd2223-info/

91948 - Project Work on Algorithms for Combinatorial Optimization Problems M

PDF, ADOC.

Learning outcomes

At the end of the project work, the students are able to apply the techniques acquired in the Algorithms for combinatorial optimization problems m course to implement effective algorithms for determining the optimal solution of a Combinatorial Optimization problem, and to analyze the corresponding computational performance.

Course contents

Design and implementation of an exact algorithm for a combinatorial optimization problem.

78917 - Project Work On Computer Vision and Image Processing M

PDF, ADOC.

Learning outcomes

At the end of the project work the students are able to apply the principles, methods and skills acquired in the Computer Vision and Image Processing course to design a software system aimed at addressing a specific use case dealing with a typical application scenario.

Course contents

See "73302- Computer Vision and Image Processing M"

94464 - Project Work on Diagnosis and Control M

PDF, ADOC.

Learning outcomes

At the end of the course, students :

are aware of state-of-the-art embedded systems used in several industry segments

know the main architectures, design methodologies and tools for embedded systems

are able to design hardware and software embedded systems in some representative case studies

Course contents

See Diagnosis and Control M

78919 - Project Work On Intelligent Systems M

PDF, ADOC.

Learning outcomes

At the end of the project work the students are able to apply the notions and skills acquired in the intelligent system course to design tools for solving real life applications.

Course contents

To be discussed with the professor

78913 - Project Work On Mobile Systems M

PDF, ADOC.

Learning outcomes

The project work is assigned on an individual base, with a negotiation with the student to identify a field at the state of the art within the large area of efficient (e.g., battery-efficient) mobile middleware and mobile applications. At the end of the project, the students have acquired a deep knowledge of the selected field and also effective capabilities of fully understanding the related global solution strategies. They have to apply the notions and skills acquired in the related lectures toward the design, prototyping, testing, and performance evaluation of a typically limited solution exemplifying the selected field and exploiting specific significant enabling technologies.

Course contents

See the syllabus and topic coverage of the course "Mobile Systems M" on the Web site at Virtuale @UNIBO, which will be populated during the lecturing period.

As a reference to the previous academic years, please consider also the material about the course structure and its possible project activities at the dedicated Web site for the 2022 edition:

http://lia.disi.unibo.it/Courses/sm2122-info/

78909 - Project Work On Multimedia Data Management M

PDF, ADOC.

Learning outcomes

The project activity aims to apply the notions and skills acquired during the course by developing a project. The project consists in the resolution of a problem concerning the management of multimedia data. The topic of the project can be proposed either by the teacher and the student.

Course contents

A scientific paper is usually considered as the base for the development of a new multimedia data management algorithm and/or infrastructure and/or service.

73302 - Computer Vision And Image Processing M

PDF, ADOC.

Learning outcomes

At the end of the course the students know the basic principles of computer vision and image processing algorithms. Thereby, they are able to understand and apply a variety of algorithms and operators aimed at either extracting relevant semantic information from digital images or improving image quality. They also understand the diverse challenges and design choices characterizing the main applications and acquire familiarity with software tools widely adopted in these scenarios.

Course contents

Introduction – Basic definitions related to image processing and computer vision. An overview across major application domains.

Image Formation and Acquisition – Geometry of image formation. Pinhole camera and perspective projection. Geometry of stereopsis. Using lenses. Field of view and depth of field. Projective coordinates and perspective projection matrix. Camera calibration: intrinsic and extrinsic parameters, lens distortion. Camera calibration based on planar targets and homography estimation (Zhang's algorithm). Image rectification and stereo calibration. Basic notions on image sensing, sampling and quantization.

Image Filtering –Convolution and correlation. Mean and Gaussian filtering. Median Filtering. Bilateral filtering. Non-local means.

Image Segmentation and Blob Analysis – Gray-level Histogram. Binarization by global thresholding.

Automatic threshold estimation. Spatially adaptive binarization. Colour-based segmentation. Binary Morphology Operators. Connected components labeling and blob analysis.

Local Features – Edge features and image gradient, Smooth derivatives (Sobel), Canny edge detector. Keypoint detectors and descriptors. Harris Corners. Scale invariant features. SIFT features. Efficient feature matching by kd-trees.

Instance Detection – Pattern matching by SSD, SAD, NCC and ZNCC. Shape-based mathing. Hough Transform for analystic shapes, Generalized Hough Transform. Object detection by local invariant features: Hough-based voting, least-squares similarity estimation.

Deep Learning for Computer Vision – Review of machine learning basics. Image Classification. Linear Classifiers and Fully Connected Neural Networks. Convolutional Neural Networks (CNN). Successful CNN architectures for image classification: AlexNet, VGG, Inception, ResNet. Transfer Learning. CNN for Object Detection: R-CNN, Feature Pyramid Network (FPN), Faster R-CNN, YOLO.

Readings/Bib

35166 - Diagnosis And Control M

PDF, ADOC.

Learning outcomes

The course aims to give a systematic overview of the main available methodologies and of the technical norms that should be used to rationally overcome problems due to faults and malfunctioning affecting modern automatic systems. Fault diagnosis and fault tolerant control methodologies as well as the functional safety tools, norms and standards that regulate safety-critical systems design are topics of the course. At the end of the course students are able to design algorithms for fault detection, to design fault tolerant schemes, and have an overview of safety norms in industrial settings.

Course contents

Introduction:

- Basic concepts;
- Nomenclature.

Reliability and Availability:

- Main definitions and concepts;
- Basics of non-state space methods;
- Basics of state space methods;

Safety:

- Safety critical systems;

- The IEC61508 standard;
- Safety life cycle;
- Fault Analysis techniques (HAZOP, FMEA, FTA)
- Layer of Protection Analysis (LOPA);
- SIL levels;

Redundancy for Fault Tolerance:

- Static and Dynamic redundancy;
- Architectures and performance;
- Analytic redundancy.

Basics of some application domains:

- Automatic machines;
- Automotive.

Model Based Fault Diagnosis;

- Basics on Fault Detection and Isolation (FDI) and links with

previous Sections.

- Signal-based methods

(useful for Model-based ones as well; SHT: LRT, GLRT, SPRT

\Chi^2)

- Parity equations (I/O and SS models, Deterministic/Stochastic)
- Unknown Input Observers (UIO).

REMARK: Contents are under review

Rea

92994 - Optimal Control M

PDF, ADOC.

Learning outcomes

The course deals with optimization-based control of dynamical systems, namely it provides theoretical and numerical methods to compute control system trajectories that minimize a performance index, and focuses on their application to trajectory optimization and maneuvering of Autonomous Systems. At the end of the course students will know how to (i) set-up an optimal control problem and characterize optimality conditions, (ii) develop numerical optimization

methods to compute optimal, feasible trajectories, and (iii) design optimization-based receding-horizon control schemes for nonlinear systems. To bridge the gap between theory and application, students will apply the proposed techniques to trajectory optimization and maneuvering of Autonomous Systems in a number of application domains including Autonomous Vehicles, Robotic Systems (e.g., Aerial Robots) and other Mechatronic Systems.

Course contents

Introduction to optimal control

Motivating application domains and tasks for the optimal control of dynamical systems: maneuvering and trajectory optimization of Autonomous Vehicles, Robotic Systems (e.g., Autonomous Mobile Robots) and other Mechatronic Systems. Optimal control problem formulation. Examples of optimal control problems in the presented application domains.

Nonlinear optimization

Unconstrained optimization and optimization over convex sets: problem formulation and optimality conditions. Special problem classes: convex optimization and quadratic optimization. Constrained optimization: Lagrangian function, Lagrange multipliers, Karush-Kuhn-Tucker (KKT) optimality conditions. Optimization algorithms: descent (line-search) methods, gradient and Newton methods, projected gradient methods, barrier function methods, Sequential Quadratic Programming (SQP). Software tools and coding on case studies for nonlinear optimization.

Optimality conditions for optimal control

Nonlinear programming reformulation of optimal control and KKT optimality conditions. Unconstrained optimal control: reformulation via shooting, derivation of the reduced-cost gradient, Hamiltonian definition, necessary conditions for optimal control. Pontryagin Maximum Principle.

Linear Quadratic (LQ) optimal control

Finite horizon: problem formulation, necessary and sufficient conditions for optimality via Riccati equation, feedback structure of the optimal control. Infinite-horizon optimal control. Trajectory tracking via optimal control: Linear Quadratic Regulator (LQR). Continuous-time version of the LQ optimal control. Case studies in autonomous vehicles and robotics. Software tools and coding on case studies.

Dynamic programming

Principle of optimality, value function and Bellman’s equation. Discrete-time Minimum Principle for optimal control. LQ optimal control via dynamic programming. Dynamic programming and Reinforcement Learning.

Numerical methods for optimal control

Gradient and Newton methods for optimal control. Barrier function method for constrained optimal control. SQP for optimal control. Reinforcement Learning: introduction and basic schemes. Software tools and coding on case studies for optimal control of autonomous vehicles and robots.

Optimization-based control techniques

Optimal control for trajectory generation and optimization. Receding horizon hierarchical control schemes. Model Predictive Control: introduction, nominal schemes on linear and nonlinear systems, extensions and applications. Software tools and coding on case studies from autonomous vehicles and robots.

73025 - Distributed Systems M

PDF, ADOC.

Learning outcomes

This graduate course aims at providing students with deep

advanced know-how about the methodologies, models, tools, and

mechanisms for the design, implementation, and runtime

evaluation/validation of enterprise applications deployed over

wide-scale distributed systems.

Previous course requirements: no one (but some contents from the

BSc courses of Computer Networks T and Web Technologies T will be

useful in some classes)

The learning outcomes of the course will include:

architecture modeling principles for

distributed enterprise applications:

requirements and design principles

Design, development, and implementation of distributed

applications based on Application Servers

(e.g., JBoss) and components (e.g.,

Enterprise Java Beans)

management of complex and articulated

container-based distributed systems, also

via lightweight models and

technologies (e.g., via Spring) and via

advanced persistency solutions (e.g., via the JPA and Hibernate technologies)

design, development, and implementation of distributed support systems for runtime monitoring and control (properties such as scalability, fault-tolerance, reliability, …; e.g., via the JMX technology)

design, development, and implementation of highly scalable distributed applications for high-end clusters, by specifically considering the clustering support available in JBoss (full configuration); some introductory notions of online stream processing over big data

Course contents

The course will aim at deeply and thoroughly facing the following topics:

- methodologies and architectural models for the design, implementation, and deployment of enterprise-level distributed applications
- component-based model evolution and component integration into distributed architectures (typically 3-tier and Web-integrated)
- Application Servers (e.g., JBoss) and middleware/frameworks for the runtime support of enterprise-level distributed applications

-

differentiated naming services and their integration,

especially in enterprise-level deployment environments; examples

related to Java Naming and Directory Interface

(JNDI)

- from the

starting Enterprise Java Beans model (EJB1.0) to the current

widespread adoption of EJB 3.0 (motivations and evolution

guidelines)

o Persistency

o Interactions with data

o Session-oriented and

message-oriented components

o Interceptors

o Transactions

o Examples and exercises

(integrated with the JBoss application server)

-

messaging systems, e.g., Java Messaging System (JMS)

and rapid overview of Enterprise Service Bus and Java Business

Integration

- towards

effective and efficient enterprise models with lightweight

containers: the Spring example

o Spring and inversion of

control

o Spring and aspect-oriented

programming

o transaction management

-

persistency: evolution of persistency support models

in the development of enterprise applications. The examples of

JPA and Hibernate.

o transparent

persistency

o mapping and query

support

o metadata support

o performance

-

monitoring, control, and runtime management of

application servers and of distributed support frameworks in

general: the JMX example

o efficiency/effectiveness

and performance evaluation

o scalability

o fault-tolerance

o reliability

-

clustering and proprietary mechanisms/solutions for

clustering in JBoss

- directions of evolution towards asynchronous models and technologies for high scalability of enterprise servers

o Framework node.js and asynch event management

o Function as a Service (FaaS) and supporting frameworks

- introductory notions about

high-performance online stream processing of big data

over clustered distributed systems

- several case

studies (also provided via seminars via external companies,

added to the regular classes schedule)

The course will be associated with a set of practical lab

exercises, in which the students will be solicited to perform

guided exercise activities but in an autonomous way and in their

free time. These activities will be necessary to complete the study

of the course and to reach the desired skills; texts and solutions

of these ees will be made available at the official course Web

site.

The set of proposed lab exercises will include:

- 1 exercise about EJB in the WildFly application

server

- 1 exercise about messaging and Java Business Integration (JBI)

- 1 exercise about Spring

- 1 exercise about JMX

- 1 exercise about clustering in WildFly

Readings/Bibliography

All the teachning material (presentation slides, discussed

exercises with solutions, exercise proposals, project examples and

proposals) used during the classes will be avai

92990 - System Theory and Advanced Control M

PDF, ADOC.

Learning outcomes

The course will provide students with the fundamental tools for the analysis and control of multivariable dynamic systems and their structural properties. Basic tools of system theory will be introduced, such as possible representation of dynamic linear systems, structural properties (stability, observability, controllability), special normal forms, Kalman decomposition, and others. The course will also address some aspects of modern multivariable control schemes starting from optimal control (in the deterministic setting), adaptive and robust control, also presenting basic aspects of nonlinear control systems.

At the end of the course students master all the basic principles of system theory by studying in a systematic way properties of multivariable dynamic systems, and have a good knowledge of modern control tools for multivariable systems.

Course contents

- Dynamic systems representation
- Stability of linear systems
- Reachability and controllability
- Observability and reconstructability
- Robust and Adaptive Control

92858 - Autonomous and Adaptive Systems M

PDF, ADOC.

Learning outcomes

At the end of this course student will have a solid

understanding of the state of the art and the key conceptual and practical aspects of the design, implementation and evaluation of intelligent machines and autonomous systems that learn by interacting with their environment.

The course also takes into consideration ethical, societal and philosophical aspects related to these technologies.

Course contents

Introduction to the design of adaptive and autonomous systems: intelligent agents and intelligent

machines, automatic vs autonomous decision-making.

Introduction to Reinforcement Learning (RL): multi-armed bandits, Montecarlo methods, tabular methods, approximation function methods, policy-based methods.

Applications of RL to games, classic control theory problems and robotics.

RL and neuroscience.

Introduction to algorithmic game theory for multi-agent learning systems: cooperation and coordination, social dilemmas, Multi-Agent Reinforcement Learning.

Bio-inspired adaptive systems.

Intelligent machines that create: Generative Learning, AI creativity, AI and the Arts.

The "brave new world": transformers, agents based on foundational models, training using Reinforcement Learning from human-feedback (RLHF).

Open problems and the future: safety, value alignment, super-intelligence, controllability, self-awareness.

Ethical implications of AI and autonomous systems.

The course will include labs in which we will discuss implementation oriented aspects of the techniques and methodologies presented during the course.

Rea

92995 - Distributed Autonomous Systems M

PDF, ADOC.

Learning outcomes

The course focuses on a novel class of autonomous control and processing systems known as cyber-physical networks or distributed autonomous systems. These multi-agent systems consist of many cyber-physical agents that aim at performing global tasks via local communication and computation in a cooperative way. At the end of the course students will know how to design selected distributed control, learning and optimization methods to solve complex tasks in cooperative network systems. To bridge the gap between theory and application, laboratory activities will allow students to apply the proposed techniques to a number of application domains, including, e.g., planning and control of cooperative autonomous robots, machine learning and data analytics, decision making and optimal control of cooperative autonomous systems.

Course contents

Introduction to Distributed Autonomous Systems

New paradigms and applications domains of autonomous systems: decision systems for data

analytics (e.g., recommender systems and localization), sensor networks, cooperative robotics, cooperative mobility, smart energy systems. Introduction to distributed systems: centralized versus distributed computing, key properties and main goals for distributed systems.

Modeling of distributed systems

Models of distributed systems. Graph theory as a tool to model communication among network agents. Preliminaries on graph theory and examples. Distributed algorithms and distributed control laws. Introduction to Python programming with a focus on distributed computing and cooperative robotics via Robotic Operating System 2.

Basic distributed algorithms

Averaging protocols and linear consensus algorithms for multi-agent systems. Complex tasks (e.g., autonomous formation control, containment in leader-follower networks) based on linear consensus algorithms. Practical Python implementation of averaging and distributed control laws on case study examples borrowed from opinion dynamics and sensor networks.

Introduction to distributed optimization

Optimization basics. Main problem set-ups and examples from estimation, machine learning, decision making, and control in cyber-physical networks.

Distributed decision making via consensus optimization

Consensus optimization algorithms based on averaging. Distributed gradient and gradient tracking methods. Practical Python implementation on case studies borrowed from distributed federated machine learning, e.g., logistic regression, support vector machine, training of neural networks for classification.

Cooperative robotics via distributed optimization

Constraint-coupled optimization algorithms based on decomposition schemes. Aggregative optimization. Practical Python implementation on case study examples borrowed from task allocation, formation control and surveillance in cooperative robotics. ROS 2 toolboxes for cooperative robotics.

78778 - Intelligent Systems M

PDF, ADOC.

Learning outcomes

At the end of the course the students are able to use the main AI techniques to develop tools for solving real life applications.

The students are able to understand and apply a wide range of techniques such as constraint programming, symbolic and sub-symbolic machine learning techniques, planning and swarm intelligence.

Course contents

Module 1:

PLANNING

Non-linear planning

Conditional planning

Graph-based planning

Planning for robotics

OPTIMIZATION

Constraint Programming and Global constraints

Search strategies

Applications

Module 2:

SWARM INTELLIGENCE

Ant colony

Bee Colony

Particle Swarm Optimization

MACHINE LEARNING (symbolic and sub-symbolic approaches)

Decision trees - random forests

Neural networks

Bayesian approaches

Inductive logic programming

R

78779 - Mobile Systems M

PDF, ADOC.

Learning outcomes

At the end of the course the students are able to effectively develop and dynamically manage the runtime provisioning of mobile services. This requires acquiring expertise and theoretical skills, as

well as design/implementation abilities, related to models and solutions for mobile systems, for mobile services provided on top of them, and for support systems (middleware) needed for their effective runtime execution.

Course contents

The course intends to provide students with methodology, modeling, design, and implementation skills/expertise related to the development, deployment, and runtime evaluation of mobile systems, of mobile services, and of middleware supports for the effective management of those services at provisioning time.

Constraints on previous knowledge: nothing (but the contents of the courses "Computer Networks T", "Web Technologies T" and, only very partially, "Infrastructures for Cloud Computing and Big Data M" could be useful and valuable in a few parts of the course)

Course contents

- Introduction to wireless communication systems: propagation models; fading models; rapid introductory overview of main types of wireless communication technologies

o IEEE 802.11: general features; from CSMA/CD over Ethernet to CSMA/CA; hidden and exposed terminal issues; MACA; configurations for infrastructured and ad-hoc modes;

o IEEE 802.16;

o IEEE 802.20;

o IEEE 802.11 for municipal meshes;

o Wi-Fi Direct.

o Cellular networks and handoff: GSM architecture; handoff with common MSC; handoff with differentiated MSCs; handoff classification

o Bluetooth: protocol stack; different possible topologies; discovery service; scatternet and multi-hop communications; software stack for Bluetooth Java programming

o Rapid overview of ZigBee: features, architecture, possible topologies

- Mobile Ad Hoc Network (MANET): definition and application domains; routing in MANET, reactive/proactive/geographic/hybrid routing; Dynamic Source Routing (DSR); Ad-hoc On Demand Distance Vector (AODV); Greedy Perimeter Stateless Routing (GPSR); rapid overview on clustering solutions; LEACH, HEED, REDMAN
- Mobility and handoff management: introduction and definitions
- Location update and location search. Mobile IP: features, architecture, protocol, issues. Hierarchical Mobile IPv6 (HMIPv6). Split connection and I-TCP
- Positioning systems: motivations; taxonomy (physical/symbolic, centralized/distributed, absolute/relative, accuracy, precision, cost, and limitations). Base techniques: lateration, time difference of arrival, angulation, scene analysis, proximity

o Positioning systems for ad-hoc networks

o Positioning systems with additional hardware. GPS: features, limitations, differential GPS. Active Badge, AHLoS

o Positioning systems with no additional hardware: based on Bluetooth, PlaceLab, RADAR, Ekahau, Sensor Fusion, Universal Location Framework, JSR-179

- Development platforms for mobile systems (smart phone): overview, general concepts, definitions; concise comparison of J2ME, .NET CF, FlashLite, Android.
- Android: introduction; architecture; OS kernel layer; power management and wakelock; native libraries; Dalvik VM; application framework; core applications; application model; activity lifecycle; intents and intent filters; threading model; examples and exercises. Description of the proposed exercise on Android
- Rapid overview on iOS: architecture; multi-tasking model; simple programming examples; native/Web applications for iOS (introductory overview of HTML5); unlocking and jailbreaking
- Mobile middleware: definitions, motivations, advantages. Relevance of determining principles and patterns. Internet principles: end-to-end and robustness principles. Web principles. SOA principles. Mobile computing principles. Cross-layering principle. Architectural patterns, general and specific for mobile computing.
- Internet of Things (IoT) platforms: definitions and reference architectues, primary state-of-the-art platforms in the market, mobile and industrial IoT, research directions under current investigation, use cases of industrial applications
- Discovery services: definition, taxonomy, auto-configuration, discovery, access to resources/services

o Jini, with description of proposed exercise

o Service location Protocol (SLP)

o UPnP: architecture, description files, bridging possibilities, examples and exercises. Description of proposed exercise on UPnP

- Session management in converged 4G networks: Session Initiation Protocol (SIP), main features, scalability limitations. Session management in IP Multimedia Sub-system (IMS): architecture, functional entities, protocol examples, limitations in terms of monitoring, load-balancing, and scalability
- Messaging support in mobile systems: motivations; principles and architecture; protocols for message exchange; locator

o JMS: features, architecture, messaging models, reliability and quality, examples

o CORBA Messaging (AMI and TII)

o Rapid overview on XMPP, Web Services, and RESTful Web Services

- Event management and pub/sub model: possible topologies for event routers. Propagation of interests and subscriptions. Routing decisions

o Examples of distributed event systems: OMG DDS (partitions and quality levels); Java model for distributed events; General Event Notification Architecture (GENA); RSS/Atom; SIP Event Framework; Web Services Eventing & Notification

- Data synchronization in mobile systems: pessimistic/optimistic approaches; versioning; detection; reconciliation

o SyncML: representation protocol; synchronization protocol; examples

- Edge/Fog Computing: efficient integration of sensor-edge-cloud; distributed control and intelligence; offloading; proactive approaches and mobility prediction

o ETSI Mobile Edge Computing: emerging standard specifications; deployment approaches in 5G; orchestration and ETSI MANO

- Selection of primary application domains (4 examples, from vehicular ad hoc applications to federated social spontaneous networks),also to suggest/solicit some possible project activities to associate with the course
- Several case studies (also presented within company seminars, in addition to the regular lectures hours)

The course will be associated with a set of practical lab exercises, where the students will have the opportunity of completing their preparation through guided and simplified projects, to be solved autonomously. These lab activities will be necessary for the full achievement of the desired abilities and course objectives; texts and solutions of proposed exercises will be available at the Course Web site. The proposed set of exercises is described in the Teaching Methods section.

78096 - Multimedia Data Management M

PDF, ADOC.

Learning outcomes

The course aims to provide the knowledge and skills necessary for the effective and efficient management of multimedia (MM) data, with particular attention to the problems of MM data representation, MM data retrieval models, and interaction paradigms between the user and the MM system (both for purposes of data presentation and exploration). We first consider architectures of traditional ("standalone") MM systems; then, we concentrate on more complex MM services, by primarily focusing on search engines, social networks and recommendation systems.

Course contents

Basics on Multimedia Data Management

Multimedia data and content representations

MM data and applications

MM data coding

MM data content representation

How to find MM data of interest

Description models for complex MM objects

Similarity measures for MM data content

MM Data Base Management Systems

Efficient algorithms for MM data retrieval

MM query formulation paradigms

Sequential retrieval of MM data

Index-based retrieval of MM data

Automatic techniques for MM data semantic annotations

Browsing MM data collections

MM data presentation

User interfaces

Visualization paradigms

Dimensionality reduction techniques

Result accuracy, use cases and real applications

Quality of the results and relevance feedback techniques

Use cases and demos of some applications

Multimedia Data on the Web

Web search engines

Graph-based data: semantic Web and social networks

Web recommender systems

N.B. For students of the second cycle degree programmes (LM) in Artificial intelligence and Computer Science, the "Efficient algorithms for MM data retrieval" part of the program is not required.

91717 - Algorithms for Combinatorial Optimization Problems M

PDF, ADOC.

Learning outcomes

At the end of the course, the students are able to analyze the computational complexity and to define the Mixed Integer Linear Programming (MILP) models of the Combinatorial Optimization problems. The students are able to obtain tight relaxations of the MILP models, and to design exact enumerative algorithms (based on the branch-and-bound, branch-and-cut and branch-and-price approaches) for determining the optimal solution of the Combinatorial Optimization problems. The students are also able to solve the MILP models by using the software package CPLEX.

Course contents

The main goals of the Course are the definition of the mathematical models and the design of the most effective exact algorithms for the solution of NP-Hard Combinatorial Optimization problems. The experimental evaluation of the proposed models and algorithms will be also analyzed.

The program of the Course consists of the following topics:

1. Classification of the optimization problems.
2. Definition of the Mathematical Models, and analysis of the computational complexity of important combinatorial optimization problems.
3. Exact algorithms for the solution of NP_Hard problems. "Branch-and-Bound" algorithms: decision trees, relaxation techniques. Exact algorithms for the effective solution of the Knapsack Problems, the Asymmetric Travelling Salesman Problem, and the Set Covering Problem.
4. "Branch-and-Cut" algorithms: addition of valid constraints for the strengthening of the relaxed problems, separation procedures for the solution of models with a large number of constraints. Exact algorithms for the effective solution of the Asymmetric Travelling Salesman Problem.
5. "Branch-and-Price" algorithms: column generation procedures for the solution of models with a large number of decisional variables. Exact algorithms for the effective solution of the Bin Packing Problem and of the Vertex Coloring Problem.
6. Experimental evaluation of the computational performance of the proposed models and algorithms.

Basic knowledge of Computer Science and Operations Research courses are required.

97430 - Cybersecurity M

PDF, ADOC.

Learning outcomes

In a digital world, every activity is vulnerable to cyber attacks. At the end of the course the students are able to know and evaluate the most dangerous cyber threats to the society and to specific organizations and industries. Moreover, they are expected to be able to design and build secure systems by adopting modern defensive strategies and technologies that are discussed in class and experimented in lab.

Course contents

Scenarios and strategies

A top-down approach to cybersecurity

Information security

Cyber threats to industry

Analysis of cyber threats

Qualitative and quantitative evaluation

Management of cyber risks: methodologies and technologies

Assessment and validation

Machine Learning for cybersecurity

Disruptive solutions

Competences and jobs of cybersecurity

73030 - Network Optimization M

PDF, ADOC.

Learning outcomes

The course introduces the student to graph and network problems and to the main algorithmic techniqes in this area. At the end of the course the student has the ability to model industrial problems that can be described through graphs and networks, and has competence on the main methods for their solution.

Course contents

Prerequisites:

It is required that the student has followed the course Operations Research M or an equivalent course on Operations Research.

Contents:

1. Fundamentals from Operations Research M
2. Graphs and networks theory

2.1 Introduction

2.2 Terminology

1. Basic problems on graphs

3.1 Shortest spanning tree

3.2 Shortest path problems

3.3 CPM method for project management0

1. Flows in networks

4.1 Maximum flow problems

4.2 Minimum cost flow problems

4.3 Mathematical models for graph and network problems

4.4 Unimodularity conditions

4.5 Matching and assignment problems

1. Optimal circuits

5.1 Hamiltonian circuits

5.2 Traveling salesman problem

5.3 Vehicle routing problems

1. Relaxations

6.1 Surrogate relaxation

6.2 Lagrangian relaxation and Lagrangian decomposition

6.3 Subgradient optimization

6.4 Reduction techniques

1. Approximation and heuristic algorithms

7.1 Greedy algorithms

7.2 Worst-case performance

7.3 Local search

7.4 Metaheuristic algorithms

75493 - Protocols And Architectures For Space Networks M

PDF, ADOC.

Learning outcomes

At the end of the course the students know the foundamentals of architecture and protocols for sapce networking. In particular, they gain a deep knowledge of DTN architecture and Bundle Protocol, as defined by RFC4838 and RFC 5050, when applied to space communications (GEO & LEO satellites, Interplanetary Internet). They are also able to set up testbeds and cary out experiments with the two most important DTN BP implementations, DTN2 and ION.

Course contents

Challenged networks

Definition of “challenged networks”. Challenges: long delays, significant losses, intermittent connectivity, network partitioning. Examples: Interplanetary Internet, satelllite networks (GEO and LEO), emergency networks, underwater networks, communications in areas not covered by ordinary TLC networks (extreme environemnts), tactical networks, etc.

The DTN architecture

Description of the DTN architecture (RFC 4838).

The Bundle Protocol

Description of the Bundle Protocol (RFC 5050 and 9171). Lab activities: use of Unibo-BP, DTN2 and ION (NASA JPL) bundle protocol implementations.

Application of the DTN architecture to the satellite networks

Characteristics of satellite networks based on Geostationary (GEO) and Low Earth Orbit (LEO) constallations. Lab activities: example of application of the DTN architecture to GEO and LEO systems.

Application of the DTN architecture to Interplanetary Internet

Characteristics of Interplanetary networks. The LTP protocol. Lab activities: examples of application of the DTN architecture to IPN systems.

Re

72953 - Principles Of Computer Graphics M

PDF, ADOC.

Learning outcomes

Knowledge of basic and advanced techniques for geometry processing and computer graphics, with particular reference to modeling and realistic rendering of 3D scenes on the computer.

Course contents

Raster-scan systems, I/O devices, graphics libraries, event-driven programming. 3D mesh and graphics representation. 2D/3D geometric transformations, viewing transformations, perspective and parallel projections, window-viewport transformations. Graphics pipeline. Real-time rendering, algorithms with hidden parts removal (hidden lines and hidden surfaces), illumination models and shading algorithms (Z-buffer), texture mapping. 3D polygonal modeling. The course includes a practical activity in which the HTML5 programming language + JavaScript + WebGL graphics library and GLSL programming will be used (module 1 (6 CFU)).

In module 2 (2 CFU) we will see the Three.js cross-browser JavaScript library, an API used to create interactive and animated 3D Computer Graphics applications on the Web using WebGL. Geometric modeling of NURBS curves and surfaces with examples in Three.js.

35253 - Hardware – Software Design Methods M

PDF, ADOC.

69494 - MULTIMEDIA SERVICES AND APPLICATIONS M

PDF, ADOC.

Learning outcomes

At the end of the course the student will have knowledge of efficient multimedia signal encoding and delivery; standards for voice, audio, image, video compression; software development kits for multimedia platforms; multimedia rendering; standards for multimedia signal delivery over the Internet, DVB, 3GPP, IEEE 802.x protocols.

Course contents

Notes on Information Theory and Source Coding: Definition of Information, Entropy, Lossless and Lossy Coding.

Image Compression: Image representations, JPEG Standard.

Video Compression: Motion Compensation Techniques, H.261, H.263, MPEG-1, MPEG-2, H.264 (aka AVC), H.265 (aka HEVC).

Audio Compression: Basic of Digital Audio, Waveform Coding, Perceptive Coding (Vocoders, MPEG Audio (MP3)).

Multimedia Protocols: Multimedia Broadcasting (MPEG Transport Stream, MPEG Program Stream), Media Transport (RTP/RTCP), Session Description Protocol (SDP), QoS management in IP Networks (DiffServ, IntServ).

Multimedia Streaming Services: QoS for streaming services; Technologies (Caching, Content Distribution Networks, Multicast, Adaptive HTTP Streaming) and Architectures (Netflix, IPTV).

Multimedia Interactive Services: QoS for voice services;VoIP (SIP, Mobile VoIP, IP Multimedia Subsystem, VoLTE)

97431 - Scalable and Reliable Services M

PDF, ADOC.

Learning outcomes

Digital services have to meet various requirements in terms of performance, scalability and reliability. We analyze the whole lifecycle of high quality services, from design to deployment, testing and operation. In each phase, we aim to provide the students with the ability to gather and analyze data, to identify sources of outages and critical dependencies, and to avoid bottlenecks and single points of failure. Finally, we expect students to be prepared to evaluate and discuss alternatives and justify investment in support of business services.

Course contents

Introduction

Cloud services

Cloud provider architectures

High-quality services: performance, scalability, reliability (and security)

Lifecycle of high-quality services

Part 1: Architectures for scalable and reliable services

Sources of outages and critical dependencies

Bottlenecks and single points of failure

Design: maturity assessment, architectural principles for reliability and scalability, AKF scale cube

model

Migration to cloud

Critical data management: Strategy, Governance, Management

Implementation alternatives on cloud platforms and multi-clouds

Part 2: Service deployment

Secure and robust software production

Secure and robust DevOps

Labs on cloud platforms

Part 3: Tests

Benchmarking

Performance testing

Stress testing

Part 4: Scalable and reliable organizations

Roles and responsibilities

Leaders and managers

Putting all together

Capacity planning

Understanding and managing complexity

Managing changes

Part 5: Issues, incident and crises

Performance monitoring

Warning signs in operation

Incidents management

Crisis management