Course Descriptions None 2022-2023

Course Title	Modelling and Computing
Course Code	EBC2180
ECTS Credits	6,5
Assessment	Whole/Half Grades

Period

Period	Start	End	Mon	Tue	Wed	Thu	Fri
4	6-2-2023	31-3-2023		X			X

Level	no level
Coordinator	Tom van der Zanden For more information:t.vanderzanden@maastrichtuniversity.nl
Language of instruction	English
Goals	* Understand combinatorial optimization problems. * Model combinatorial optimization problems. * Analyse time complexity of algorithms using big O-notation. * Suggest an algorithm solving a combinatorial optimization problem either to optimality or approximately. * Implement an algorithm in a programming language such as Java. * Classify the complexity of a combinatorial optimization problem. * Identify which instances sizes of a problem are (in-)tractable using an algorithm.
Description	PLEASE NOTE THAT THE INFORMATION ABOUT THE TEACHING AND ASSESSMENT METHOD(S) USED IN THIS COURSE IS WITH RESERVATION. A RE-EMERGENCE OF THE CORONAVIRUS AND NEW COUNTERMEASURES BY THE DUTCH GOVERNMENT MIGHT FORCE COORDINATORS TO CHANGE THE TEACHING AND ASSESSMENT METHODS USED. THE MOST UP-TO-DATE INFORMATION ABOUT THE TEACHING/ASSESSMENT METHOD(S) WILL BE AVAILABLE IN THE COURSE SYLLABUS. When operating a business, we often have to solve optimization problems to use available resources as efficiently as possible. Consider for instance a package delivery service: we must determine which packages to load into which vans, so that the total fuel costs of delivery are minimized. Often, there are many additional constraints that must be taken into account: for instance, the maximum loading capacity of a van and the maximum distance a driver is allowed to drive in one day. Clearly, with so many constraints, it becomes difficult to see if a solution exists at all, let alone to find one with minimum cost. This course introduces the students to the basics of modelling, algorithms and intractability to give them the tools to deal with these kinds of problems. The course will cover modelling business problems using (integer) linear programming, maximum flow and boolean logic and cover algorithmic techniques such as dynamic programming, greedy algorithms, the simplex method, local search and using solvers as black boxes. Simultaneously, the students are introduced to the intuition behind NP-hardness and reductions, and will be introduced to a set of canonical "hard" problems that can be used to assess the hardness of a business problem. By the end of the course, students will be able to take a business case and create several formulations of optimization problems related to such a case. For such a formulation, students will have an intuition of how hard it is to solve (any why), and be able to determine which algorithmic technique is suitable to tackle it. Formative assessment: Assignment 1,2 and 3 and feedback during tutorial meetings and mock exam Summative assessment: Assignment 4 and 5, and exam Instructional approach: Lectures, working sessions (exercices and flipped classroom) and programming assignments, including feedback
Literature	
Prerequisites	
Keywords	
Teaching methods	
Assessment methods	Written Exam
Evaluation in previous academic year	For the complete evaluation of this course please click http://iwio-sbe.maastrichtuniversity.nl/rapporten.asp?referrer=codeUM

This course belongs to the following programme / specialisation

Bachelor Business Analytics	Year 2 Compulsory Course(s)