De Montfort University

Module template proforma

Basic module information

Module Title:	Computational Intelligence Optimisation				
Module Code:	IMAT5232	Credit Value:	15.00	DMU Credit Level:	5
Owning Board:	IMAT				
Faculty:	Faculty of Technology				
Academic Period:	2018				
Module Leader:	Fabio Caraffini				
Module Pre-requisites:					
Maximum student numbers on module (if applicable):					
Semester module can run:	Semester 2				
Version Number:	19				

Module description (including outline content)
Computational Intelligence Optimisation (CIO) is a subject that integrates artificial intelligence into algorithms for solving optimisation problems that could not be solved by exact methods. Thus, CIO is the subject that
defines and designs meta-heuristics to address domains where pure mathematical approaches are not applicable.
The terms metaheuristics, refers to all those general purpose algorithms whose working mechanism do not require any particular assumption to operate. In (real-valued) otimisation, metaheuristic algorithms find their place in the vast majority of real-world applications. Indeed, they are extremely versatile and can tackle optimisation problems in engineering, economics and all applied sciences. The CIO subject contains algorithmic structure based on metaphors such as evolution and collective intelligence. This module will provide students with an appreciation of both theoretical and implementation issues of such algorithms. Selected algorithms (negotiated between the lecturer and each student) will be studied in practical work.

Outline content:
Generalities: Definition and postulate of optimisation problems, fitness landscape and problem features, No Free Lunch Theorem
Classical derivative free methods: General concepts of Rosenbrock, Hooke-Jeeves, Nelder-Mead Algorithms, Simulated Annealing, Multi-start search
Popular population-based algorithms: Genetic Algorithms, Evolution Strategy, other examples of modern evolutionary approaches, Particle Swarm Optimisation, Differential Evolution, other examples of modern swarm intelligence algorithms, perturbation mechanisms of an algorithmic scheme
An overview on modern approaches: Adaptive Systems, Hyper-heuristics, Memetic Computing
An overview on special problems: Multi-objective, noisy, dynamic, computationally expensive, and large scale problems
Application examples: control theory, robotics, embedded systems, training of neural networks.

Learning outcomes

1 demonstrate a comprehensive understanding of Computational Intelligence Optimisation;
2 be able to critically apply and implement the taught CIO algorithms to given test problems

Assessment

Type of assessment	**Duration or volume**	**Assessment weighting %**	**Final assessment Y/N**	**Minimum threshold mark %** (if not 40% for	**Essential component Y/N**	**Anonymously marked**

				UG, 50% for PGT)	(approval needed for UG essential components)	**Y/N**
Task 1: Maths LO1	4 exercises	20	N	50	N	Y
Task 2: Single-solution LO1, LO2	Software + report (5 pages)	30	N	50	N	Y
Task 3: Metaheuristic LO1, LO2	Software + report/paper (10 pages)	40	N	50	N	N
Discussion board LO2	At least two comments	10	N	50	N	N

Assessment notes
Coursework consists of three tasks. Tasks are submitted at different times (i.e. there is a first interim submission, a second interim submission and a final submission), but the possibility of submitting, or resubmitting, the three tasks all together via the Final Submission link is also given to the students. For details, refer to the Coursework Specification Template. A brief description of each task is reported below.

TASK 1
The student will select and solve 4 exercise out of a given list of suggested exercises. Steps have to be shown and reported in a pdf file to be submitted via Turnitin. This task is worth 20% of the final mark.

TASK 2
The student will have to write the code implementing one single-solution optimiser, chosen from those presented in the lectures, and test it over 4 popular test-bed Numerical results will be displayed and commented in a short report. This report has to be submitted via Turnitin and it is worth 30% of the final mark.
N.B. Code can be either attached in the report, or simply sent via email or Turnitin.

TASK 3
After the diffusion of NFLT, the need of choosing and designing optimisation algorithms, tailored to given problems, arose. In this light, the student will have to design a novel optimisation algorithm capable of outperforming at least two comparison algorithms (see details in the Coursework Specification from) over the same problems. The comparison will be carried out by optimising the functions of a provided benchmark suite for optimisation (unless the student wanted to implement its own original real-world application). In addition to the code, the student will hand in a report where he will analyse the problem/problems, justify the algorithmic choices, describe the newly designed algorithm as well as the numerical results. The report will be written in the fashion of a (mini)Paper in the filed, and submitted via Turnitin. The report will be worth 40% of the final mark.
N.B. Code can be attached in the report, or simply sent via email or Turnitin.

Discussion board contributions account for 10% of the module marks.

Reassessment
Reassessment will be by failed component

Expected methods of delivery
This module will be delivered to distance learning and on-site students using e-learning facilities for all students.

The teaching strategy is to present students with appropriate learning materials suitable for directed study by distance learning. Students will also be given exercises to support the fundamental topics in the syllabus and where appropriate the discussion board will be used. Both practical and theoretical aspects of the subject area will be addressed.

For on-site students timetabled lecture/lab/tutorial sessions will be used as well as making use of the e-learning facilities.
DL students will study purely using e-learning mechanisms.

Teaching and learning hours

This is a 15 credits module associated to a volume of study of 150 hours. The total amount of hours is obtained by undertaking the following learning activities:

-LECTURES (20 hours);
-LABS (10 hours);
-CONSOLIDATION (40 hours);
-COLLABORATIVE ACTIVITIES (10 hours);
-READING (40 hours);
-REVISION (30 hours).