2024 / 25

School of Science and Computing
+353 (0)51 302037
Eleanor.Reade@setu.ie
www.wit.ie/schools/science_computing

Ollscoil Teicneolaíochta an Oirdheiscirt
South East Technological University

Module Descriptor

Graph Theory and Optimisation (Computing and Mathematics)

Graph Theory and Optimisation (A13984)

Short Title:	Graph Theory and Optimisation		
Department:	Computing and Mathematics		
Credits:	10	**Level:**	Postgraduate

Description of Module / Aims

The module introduces the students to the fundamental concepts and techniques in graph theory and network based combinatorial optimisation, focusing on the relationships between algorithms and associated data structures. The module also includes discussions of applications to problems from computing, engineering, and operations research thus illustrating the broad applicability of the theory.

Programmes

		stage/semester/status
COMP-0302	MSc in Computer Science (Enterprise Software Systems) (WD_KCESS_R)	1 / 0 / E
COMP-0302	MSc in Computing (Information Systems Processes) (WD_KISYP_R)	1 / 0 / E

Indicative Content

- Algorithmic Complexity: order notation; analysis of algorithms; greedy heuristics and approximate algorithms; classification of decision problems; complexity classes of P, NP and Co-NP. NP-completeness; NP-hardness
- Optimal Routes in Graphs: DFS, BFS and hybrid traversal; applications of shortest routes in graphs; methods of Dijkstra and of Floyd; applications of longest routes in directed acyclic graphs
- Minimum Spanning Trees: applications of trees and of rooted trees; properties of rooted trees and of m-ary trees (with binary trees as a special case); metermination of minimum spanning trees by Kruskal's, Prim's and Boruvka's methods; analysis of MST algorithms
- Planarity: Euler's formula; applications of graph planarity with respect to optimal facility lay-outs; Kuratowski's theorem; crossing numbers of graphs embedded in the plane
- Centres and Medians: vertex eccentricity; radius and diameter of a graph; graph centre and a graph median; applications to optimal facility placement in communication networks
- Vertex and Edge Colourings: vertex and edge chromatic numbers; heuristic methods for graph colourings; applications of vertex & edge colourings with respect to scheduling
- Eulerian Graphs: the Knigsberg bridge problem; Chinese postman problem and variants; DeBruijn sequences and gray codes; application to model based testing
- Hamiltonian Graphs: the travelling salesman problem (TSP); heuristics for the travelling salesman problem (such as the method of nearest neighbours and Christofides' method of cheapest insertion); post-optimisation methods for the travelling salesman problem (such as 2-opt)
- Network Flow: The matching problem; matching and augmenting paths; the network flow problem; assignment problem, weighted edge cover problem; multi-comodity flow problem
- Social Networks: power-law networks; the small-world phenomenon; navigation in social networks; sampling of network properties; generation of networks; the Erdos Renyi model
- Emerging Topics: dynamic graph algorithms; clustering and topology trees; randomised algorithms; graph drawing algorithm; on-line algorithms: paging scheduling and load balancing

Learning Outcomes

On successful completion of this module, a student will be able to:

1. Interpret complexity classes (in particular P vs NP), completeness and hardness, and the relationships between classes by reduction.
2. Interpret the theory underlying exact and heuristic approaches to combinatorial optimisation problems.
3. Prove and interpret standard results in graph theory.
4. Develop, implement, and critically evaluate the correctness and performance of standard graph and combinatorial optimisation algorithms.
5. Select and implement appropriate formulations and algorithms from graph theory and combinatorial optimisation to analyse problems from computing, engineering, and operations research.

Learning and Teaching Methods

- Delivery of the module will be through lectures and computer based practicals.
- The lectures will develop theory, lead students through worked examples and introduce the context for the module material.
- The practical sessions will be used to discuss applications of the theory and will involve applying a number of problems and algorithms discussed during the course, as well as a practical implementation of the relevant data-structures. This way the students get practical, hands-on experience with techniques from graph theory and combinatorial optimisation.
- Given the level of mathematical notation and concepts used in the module, it is advisable that a student has basic programming skills (graduate of a BSc or BEng involving significant programming experience) and at least a moderate mathematical ability (at least one year of undergraduate mathematics), in particular, basic linear algebra (including eigenvalues and eigenvectors) and an understanding of simple combinatorial counting processes involving permutations and combinations.

Learning Modes

Learning Type	F/T Hours	P/T Hours
Lecture	24	24
Practical	24	24
Independent Learning	222	222

Assessment Methods

	Weighting		Outcomes Assessed
Final Written Examination	50%		1,2,3
Continuous Assessment	50%		
Assignment		20%	4
Assignment		30%	5

Assessment Criteria

<40%: Inability to demonstrate knowledge or understanding of the concepts outlined in the syllabus content, inability to apply concepts to selected problems.

40%–59%: Able to demonstrate a basic understanding of the fundamental concepts outlined in syllabus content. Able to find the solution set to basic problems, and in addition, able to supply a reasonable interpretation of generated results.

60%–69%: All the above, in addition be able to independently determine appropriate mathematical techniques to analyse applied problems and to express their work with rigour and precision.

70%–100%: All the above to an excellent level. Demonstrates an ability to put a solution into a context and assess whether such solutions are meaningful.

Essential Material(s)

- Wilson, R. *Introduction to Graph Theory.* London: Pearson, 2012.

Supplementary Material(s)

- "Algorithmic Graph Theory." http://www.personal.kent.edu/ rmuhamma/GraphTheory/graphTheory.htm
- "The Math Forum (Drexel)." http://mathforum.org/library/topics/graph_theory/
- Ammann, P. and J. Offutt. *Introduction to Software Testing.* UK: Cambridge, 2008.
- Atallah, M. *Algorithms and Theory of Computation Handbook.* NY: CRC Pres, 1999.
- McHugh, J. *Algorithmic Graph Theory.* NY: Prentice Hall, 1989.
- Sedgewick, R. *Algorithms in Java: Graph Algorithms.* NY: Addison-Wesley, 2003.

Requested Resources

- Computer Lab: BYOD Lab