



MONASH University

FIT4010
Advanced topics in algorithms and discrete structures

Unit Guide

Semester 2, 2009

The information contained in this unit guide is correct at time of publication. The University has the right to change any of the elements contained in this document at any time.

Last updated : 13 Jul 2009

Table of Contents

<u>FIT4010 Advanced topics in algorithms and discrete structures - Semester 2, 2009</u>	1
<u>Chief Examiner:</u>	1
<u>Lecturer(s) / Leader(s):</u>	1
<u>Clayton</u>	1
<u>Introduction</u>	2
<u>Unit synopsis</u>	2
<u>Learning outcomes</u>	2
<u>Contact hours</u>	2
<u>Workload</u>	2
<u>Unit relationships</u>	3
<u>Prerequisites</u>	3
<u>Relationships</u>	3
<u>Teaching and learning method</u>	4
<u>Timetable information</u>	4
<u>Tutorial allocation</u>	4
<u>Unit Schedule</u>	4
<u>Unit Resources</u>	5
<u>Prescribed text(s) and readings</u>	5
<u>Recommended text(s) and readings</u>	5
<u>Required software and/or hardware</u>	5
<u>Equipment and consumables required or provided</u>	5
<u>Study resources</u>	5
<u>Assessment</u>	6
<u>Overview</u>	6
<u>Faculty assessment policy</u>	6
<u>Assignment tasks</u>	6
<u>Due dates and extensions</u>	7
<u>Late assignment</u>	7
<u>Return dates</u>	7
<u>Appendix</u>	8

FIT4010 Advanced topics in algorithms and discrete structures - Semester 2, 2009

Chief Examiner:

Associate Professor Bernd Meyer

Associate Professor
Phone: +61 3 990 52240
Fax: +61 3 990 31077

Lecturer(s) / Leader(s):

Clayton

Associate Professor Bernd Meyer

Associate Professor
Phone: +61 3 990 52240
Fax: +61 3 990 31077

Professor Kimbal Marriott

Professor
Phone: +61 3 990 55525
Fax: +61 3 990 32745

Introduction

Welcome to FIT4010 Advanced Topics in Algorithms and Discrete Structures. This year we will be looking at algorithms to solve continuous and discrete optimisation problems, in particular constrained optimisation problems. These are very important in many scientific areas and have many industrial applications, such as timetabling, resource allocation, visualization, airline scheduling or fleet-coordination. Unfortunately, almost all of these problems are computationally very difficult and, therefore, specialized techniques are required to handle them. Hands-on experience will be provided through practical assignments.

Unit synopsis

Algorithms are the most fundamental area for all aspects of computer science and software engineering. Discrete structures, such as those treated in graph theory, set theory, combinatorics and symbolic logic form the mathematical underpinning of the study of algorithms. As well-designed algorithms and data structures are essential for the good performance of an information system, an in-depth understanding of the theoretical properties of algorithms is essential for any computer scientist. As importantly, the theoretical investigation of algorithms leads to a deeper understanding of problem structures and classes of problems and the knowledge of a large variety of algorithm types enables the designer to approach a new problem from different angles. Topics for this unit include: Computability and Complexity Automata Theory Advanced Analysis and Design of Algorithms Parallel and Distributed Algorithms Numerical Algorithms Cryptographic algorithms Spatial/geometric algorithms

Learning outcomes

At the completion of this unit students will:

1. have an improved understanding of the issues involved in designing algorithms in the chosen specialisation area(s) and in analysing their performance;
2. understand the mathematical formalisms that are relevant for these algorithms;
3. have learned to recognise tasks that can be solved with these algorithms;
4. be able to judge the limitations of these methods;
5. be able to choose and apply algorithms and data structures in the chosen specialisation area(s);
6. be able to evaluate the performance of algorithms using formal approaches;
7. will be able to design modified algorithms in the chosen area to suit particular problem structures.

Contact hours

Lectures and laboratory classes: 3hrs/week

Workload

- two hour lecture and
- one hour tutorial (or laboratory) (requiring advance preparation)
- a minimum of 2-3 hours of personal study per one hour of contact time in order to satisfy the reading and assignment expectations.
- You will need to allocate up to 5 hours per week in some weeks, for use of a computer.

Unit relationships

Prerequisites

Completion of the Bachelor of Computer Science or equivalent to the entry requirements for the Honours program. Students must also have enrolment approval from the Honours Coordinator.

Relationships

The unit is an elective in the Bachelor of Computer Science Honours programme.

Teaching and learning method

The main teaching forum will be the lectures and tutorials. Students are also expected to make use of the on-line discussion forums for any questions regarding the unit material or organisation.

Timetable information

For information on timetabling for on-campus classes please refer to MUTTS, <http://mutts.monash.edu.au/MUTTS/>

Tutorial allocation

On-campus students should register for tutorials/laboratories using the Allocate+ system:

<http://allocate.cc.monash.edu.au/>

Unit Schedule

Week	Topic	Key dates
1	Syllabus overview; what is constrained optimisation and what is it good for; complexity of constrained optimization; NP-completeness; Cook's Theorem	
2	Linear Programming and the Simplex Method; Duality	
3	Lagrange Multiplier Methods and KKT Conditions	
4	Quadratic Programming, Interior Point Methods	
5	Integer Programming and Mixed Integer Programming; Branch & Bound Methods;	
6	Cutting Planes; Network Flow	Assignment 1 due August 28
7	Modelling with LP/IP/MIP	
8	Local Search, Simulated Annealing; Heuristics	
9	Tabu Search & Evolutionary Methods	
10	Other Meta-Heuristic Methods	
Mid semester break		
11	Constraint Propagation and Constraint Solving	
12	Modelling with Constraint Languages	Assignment 2 due October 9
13	Other Modelling Languages; Research Frontiers in Optimization and Constraint Handling	Assignment 3 due October 16

Unit Resources

Prescribed text(s) and readings

There are no prescribed books.

Recommended text(s) and readings

There are several recommended books for this subject:

- Introduction to Mathematical Programming. Wayne L. Winston. Duxbury Press, 1995.
- Introduction to Operations Research. F.S. Hillier and G.J. Lieberman. McGraw-Hill, 8th Ed, 2005.
- Constraint Programming-An Introduction. Kim Marriott and Peter Stuckey. MIT Press, 1998.

In addition to this, selected research papers will be referenced throughout the course.
The lecture material will be loosely based on this material and will be available on the Web.

Required software and/or hardware

Students will need access to the latest version of Java and are encouraged to make use of GLPK, which can be obtained under GNU licence (free) from <http://www.gnu.org/software/glpk/>

Other software that will be used is the MiniZinc language, software will be distributed in the initial tutorial and can be obtained from <http://www.g12.cs.mu.oz.au/minizinc/>

Equipment and consumables required or provided

Students will need access to:

- a personal computer running Linux, Mac OS X, or Cygwin under Windows
- the internet via dial-up connection or preferably by broadband
- a printer for assignments

Study resources

Study resources we will provide for your study are:

- Weekly detailed lecture notes;
- Weekly tutorial or laboratory tasks;
- Assignment specification;
- Discussion groups;
- This Unit Guide outlining the administrative information for the unit;
- The unit web site on Moodle, where resources outlined above will be made available.

Assessment

Overview

Assignment and Examination, relative weight depending on topic composition. When no exam is given students will be expected to demonstrate their knowledge by solving practical problems and maybe required to give an oral report. This variability is designed to give flexibility to the lecturer to decided the most appropriate form of examination for a given choice of topics.

Faculty assessment policy

To pass a unit which includes an examination as part of the assessment a student must obtain:

- 40% or more in the unit's examination, and
- 40% or more in the unit's total non-examination assessment, and
- an overall unit mark of 50% or more.

If a student does not achieve 40% or more in the unit examination or the unit non-examination total assessment, and the total mark for the unit is greater than 44% then a mark of no greater than 44-N will be recorded for the unit.

This unit is assessed with three non-exam assessments. To pass the unit you must:

- achieve no less than 40% of the marks in each assignment
- achieve no less than 50% of the possible marks overall

Assignment tasks

Assignment coversheets

Assignment coversheets are available via "Student Forms" on the Faculty website:

<http://www.infotech.monash.edu.au/resources/student/forms/>

You MUST submit a completed coversheet with all assignments, ensuring that the plagiarism declaration section is signed.

Assignment submission and return procedures, and assessment criteria will be specified with each assignment.

• Assignment task 1

Title:

Assignment 1: Continuous Constraint Optimization - Linear and Quadratic Programming

Description:

Weighting:

30%

Due date:

August 28

- **Assignment task 2**

Title:

Assignment 2: Combinatorial Constraint Optimization - Integer Programming

Description:

Weighting:

40 %

Due date:

October 9

- **Assignment task 3**

Title:

Assignment 3: Modelling with Constraints

Description:

Weighting:

30%

Due date:

October 16

Due dates and extensions

Please make every effort to submit work by the due dates. It is your responsibility to structure your study program around assignment deadlines, family, work and other commitments. Factors such as normal work pressures, vacations, etc. are not regarded as appropriate reasons for granting extensions. Students are advised to NOT assume that granting of an extension is a matter of course.

Students requesting an extension for any assessment during semester (eg. Assignments, tests or presentations) are required to submit a Special Consideration application form (in-semester exam/assessment task), along with original copies of supporting documentation, directly to their lecturer within two working days before the assessment submission deadline. Lecturers will provide specific outcomes directly to students via email within 2 working days. The lecturer reserves the right to refuse late applications.

A copy of the email or other written communication of an extension must be attached to the assignment submission.

Refer to the Faculty Special consideration webpage or further details and to access application forms:
<http://www.infotech.monash.edu.au/resources/student/equity/special-consideration.html>

Late assignment

Assignments received after the due date will be subject to a penalty of 10% per day, including weekends. Assignments received later than one week (seven days) after the due date will not normally be accepted.

Return dates

Students can expect assignments to be returned within two weeks of the submission date or after receipt, whichever is later.

Appendix

Please visit the following URL: <http://www.infotech.monash.edu.au/units/appendix.html> for further information about:

- Continuous improvement
- Unit evaluations
- Communication, participation and feedback
- Library access
- Monash University Studies Online (MUSO)
- Plagiarism, cheating and collusion
- Register of counselling about plagiarism
- Non-discriminatory language
- Students with disability
- End of semester special consideration / deferred exams