University of Macau
Department of Computer and Information Science
SFTW221 –Operations Research II
Syllabus
1st Semester 2011/2012
Part A – Course Outline

Compulsory course in Computer Science

Catalog description:
(3-2) 4 credits. Network optimization, network flow, project management, dynamic programming, linear integer programming, classical theory of nonlinear programming.

Course Type:
Theoretical

Prerequisites:
- SFTW112

Textbook(s) and other required material:
- Hamdy A. Taha. (2007) **Operations Research: An Introduction.** 8th Ed., Pearson.

References:
- Hiller Lieberman. (2005) **Introduction to Operations Research.** 8th Ed., McGraw-Hill.

Major prerequisites by topic:
1. Linear algebra.
2. Linear programming.
3. Calculus.

Course objectives:
1. Be able to solve linear integer programming problems by B&B algorithm. [a, e, l]
2. Be able to find the minimal spanning tree, shortest-route, maximal flow, critical path of a network. [a, e]
3. Be able to solve some problems by dynamic programming. [a, e]
4. Have basic understanding of Karush-Kuhn-Tucker conditions for nonlinear optimization. [a]

Topics covered:
1. **Integer Linear Programming (ILP) (6 hours)**: Introduce ILP by some illustrative application problems such as capital budgeting, set-covering problem. Introduce the B&B algorithm for ILP.
2. **Introduction to network (1 hour)**: Give the definition of a network and talk about the definitions of flow, directed network, path, cycle, loop, connected network, tree, spanning tree.
3. **Minimal Spanning Tree (1 hours):** Introduce the algorithm for constructing the minimal spanning trees and give some application examples.
4. **Shortest-route (4 hours):** Introduce the importance of finding shortest-route by some application examples. Introduce the Dijkstra's algorithm and Folyd's algorithm for finding shortest-route of a network. Linear programming formulation of the shortest-route problem is also given.
5. **Maximal Flow (4.5 hours)**: Introduce the maximal flow problem by some application examples. Study the maximal flow algorithm and relate the maximal flow of a network with the minimal cut of the network. Linear programming formulation of the maximal flow problem is also given.
6. **Critical Path (4.5 hours)**: Study the network representation of a project and the critical path calculations for a project network. Construct the time schedule form the result of critical path calculation.

7. **Dynamic Programming (DP) (6 hours)**: Study the recursive nature of DP through a simple shortest-route problem. Introduce recursive functions and backward recursion for DP. Apply DP to solve some application problems such as knapsack model, work-force size model, equipment replacement model and investment model.
8. **Classical Theory of Nonlinear Optimization (7 hours)**: Study the necessary condition, sufficient condition, and, necessary and sufficient condition for a local minimum of an unconstrained problem. Study the KKT condition for a local minimum of a constrained problem. Study the convexity of a function and a set and relate them with optimality.
9. **Numerical Methods for Unconstrained Optimization (4 hours)**: Introduce the steepest descent method and the Newton's method. Study the convergence rate of the two methods.

Class/laboratory schedule:

Timetabled work in hours per week			No of teaching weeks	Total hours	Total credits	No/Duration of exam papers
Lecture	Tutorial	Practice				
3	2	Nil	14	70	4	1 / 3 hours

Student study effort required:

Class contact:	
Lecture	42 hours
Tutorial	28 hours
Other study effort	
Self-study	20 hours
Homework assignment	10 hours
Total student study effort	100 hours

Student assessment:

Final assessment will be determined on the basis of:

Homework	10%	Attendance	10%
Mid-term	30%	Final Exam	50%

Course assessment:

The assessment of course objectives will be determined on the basis of:

1. Homework and exams
2. Course evaluation

Course outline:

Weeks	Topic	Course work
1,2	**Integer Linear Programming (ILP)** Modeling examples; B&B method	Assignment#1
3	**Scope and Definition of Network Models** Nodes, arcs, definition of a network, flow of a network, directed network, path, cycle, loop, connected network, tree, spanning tree. **Minimal Spanning Tree** Minimal spanning tree algorithm	Assignment#2
3,4	**Shortest-Route** Examples of the shortest-route application, Dijkstra's algorithm, Floyd's algorithm, linear programming formulation	Assignment#2

5,6	**Maximal Flow** Enumeration of cuts, maximal flow algorithm, linear programming formulation	Assignment#3
6,7	**Critical Path Method** Network representation of a project, critical path computations, construction of the time schedule	Assignment#3
8	**Dynamic Programming (DP)** Shortest-route example, Recursive equation, forward and backward recursion	Assignment#4
9	**Midterm Exam**	
9,10	**Selected DP Applications** Knapsack model, work-force size model, equipment replacement model, investment model	Assignment#4
10-12	**Classical Theory of Nonlinear Optimization** Unconstrained problems, equality-constrained problems, inequality-constrained problems, Karush-Kuhn-Tucker (KKT) conditions	Assignment#5
13,14	**Numerical Methods for Unconstrained Optimization** Steepest descent method, Newton's method, convergence rate	
14	**Review**	

Contribution of course to meet the professional component:

This course prepares students to work professionally when dealing with engineering optimization problem in their career.

Relationship to CS program objectives and outcomes:

This course primarily contributes to Computer Science program outcomes that develop student abilities to:

(a) an ability to apply knowledge of mathematics, science, and engineering.

(e) an ability to analyze a problem, and identify, formulate and use the appropriate application requirements for obtaining its computing solution.

(l) an ability to use the computer/IT tools relevant to the discipline along with an understanding of their processes and limitations.

Relationship to CS program criteria:

Criterion	DS	PF	AL	AR	OS	NC	PL	HC	GV	IS	IM	SP	SE	CN
Scale: 1 (highest) to 4 (lowest)	2													1

Discrete Structures (DS), Programming Fundamentals (PF), Algorithms and Complexity (AL), Architecture and Organization (AR), Operating Systems (OS), Net-Centric Computing (NC), Programming Languages (PL), Human-Computer Interaction (HC), Graphics and Visual Computing (GV), Intelligent Systems (IS), Information Management (IM), Social and Professional Issues (SP), Software Engineering (SE), Computational Science (CN).

Course content distribution:

Percentage content for			
Mathematics	Science and engineering subjects	Complementary electives	**Total**
80%	20%	0%	100%

Coordinator:

Siu Long Lei, Assistant Professor of Mathematics.

Persons who prepared this description:

Siu Long Lei, Seak Weng Vong, July 25, 2011.

Part B General Course Information and Policies

1st Semester 2011

Instructor:	Dr. Siu Long Lei	Office: N327B
Office Hour:	Mon., Thur. 9:30AM – 11:30AM or by appointment	Phone: 8397-4378
Email:	sllei@umac.mo	

Time/Venue: Theory – Mon. 4:30PM-6:30PM & Wed. 8:30AM-9:30AM L109
Tutorial – Wed. 9:30AM-11:30AM U101 (Class A)
Fri. 5:30PM-7:30PM U102 (Class B)

Grading Distribution:

Percentage Grade	Final Grade	Percentage Grade	Final Grade
100 - 93	A	92 - 88	A–
87 - 83	B+	82 - 78	B
77 - 73	B–	72 - 68	C+
67 - 63	C	62 - 58	C–
57 - 53	D+	52 - 50	D
below 50	F		

Comment:
All students are expected to attend all lectures and examinations. It is your responsibility to read the relevant chapters in the text before and after class and to ask questions during class discussion. In order to be successful in this course, you should get as much practice as possible in solving problems outside the class hours. This must be done on a timely and regular basis, as a good understanding of the material covered in any particular section of this course depends heavily on an equally good understanding of the material covered in previous sections.

Homework Policy:
All homework must be an individual effort unless specifically noted. Your work must be neat, with answers clearly noted and supporting information provided. Late homework will not be accepted in general.

Note

- Check UMMoodle (https://ummoodle.umac.mo/) for announcement, homework and lectures. Report any mistake on your grades within one week after posting.
- No make-up exam is given except for CLEAR medical proof.
- Cheating is absolutely prohibited by the university.

Appendix:

Rubric for Program Outcomes:

Rubric for (a)	5 (Excellent)	3 (Average)	1 (Poor)
Understand the theoretic background	Students understand theoretic background and the limitations of the respective applications.	Students have some confusion on some background or do not understand theoretic background completely.	Students do not understand the background or do not study at all.
Use a correct model and formulation correctly	Students choose a model correctly and properly apply correct techniques.	Students choose a wrong model sometime, use a wrong formula, or a different technique.	Students use a wrong model and wrong formula, or do not know how to model.
Compute the problem correctly	Students use correct techniques, analyze the problems, and compute them correctly.	Students sometime solve problem mistakenly using wrong techniques.	Students do not know how to solve problems or use wrong techniques completely.

Rubric for (e)	5 (Excellent)	3 (Average)	1 (Poor)
Identify applications in engineering systems	Students understand problem and can identify fundamental formulation.	Students understand problem but cannot apply formulation, or cannot understand problem.	Students cannot identify correct terms for engineering applications.
Modeling, problem formulation and problem solving	Students choose and properly apply the correct techniques.	Students model correctly but cannot select proper technique or model incorrectly but solve correctly accordingly.	Students at loss as to how to solve a problem.

Rubric for (l)	5 (Excellent)	3 (Average)	1 (Poor)
Use modern computer/IT tools relevant to the discipline	Student uses computer/IT tools relevant to the engineering discipline, and understands their limitations.	Student uses computer /IT tools relevant to the engineering discipline.	Student does not use computer/IT tools relevantly, and does not understand their limitations.