SE / CS 3O03 / 4O03 (6O03) Operations Research – Fall 2015

Antoine Deza

1 What the students should know and be able to do

1. Students should know and understand
   (a) notions of hyperplane, vertex, and convexity in general dimension
   (b) methods for solving systems of linear equations (Gaussian elimination)
   (c) the formulation of (integer) linear optimization problem: standard form
   (d) simplex methods for linear optimization
   (e) dynamic programming
   (f) graph and network algorithms and their formulations as linear optimization
   (g) geometric and combinatorial aspects of linear optimization, main concepts behind central-path following method (Interior Point Method), challenges of non-linear, non convex optimization,

2. Students should be able to
   (a) show basic algebraic and geometric properties of linear optimization
   (b) formulate in standard form continuous and integer linear optimization problems
   (c) show correctness and convergence of the simplex algorithm for linear optimization
   (d) show correctness and complexity of dynamic programming for the knapsack problem
   (e) show correctness and complexity for minimum spanning tree, shortest path, and network flow algorithms
   (f) explain and implement the main algorithms presented
   (g) explain and identify which method/algorithm is suited for a given problem

2 Mapping to Attributes with their Indicators

A01 Knowledge	
Competence in specialized engineering knowledge	1a–1g
A03 Investigation	
Uses appropriate techniques to collect data	2a, 2b
Assess the accuracy and precision of results and recognize limitations of the approach	2c, 2d, 2e
A04 Design	
Recognizes and follows an engineering design process	2b, 2f, 2g
Recognizes and follows engineering design principles	2b, 2f, 2g
Obtains experience with open-ended problems	2g
Properly documents and communicates processes and outcomes	2b

Table 1: Rubric: (4) **Competence in Specialized Engineering Knowledge**
Student work used: assignments, midterm, and final exams.

	Expectations			
Topic	**Below**	**Marginal**	**Meets**	**Exceeds**
Simplex method for linear optimization 1a, 1b, 1c, 1d	cannot explain the main steps of the simplex method	can explain the simplex method with a limited understanding	can explain the simplex method with a solid understanding	can explain the simplex method with a full understanding
Dynamic programming and greedy algorithms 1e	cannot identify a dynamic programming instance	can identify a dynamic programming instance, but the formulation is not completely correct	can identify and correctly formulate a dynamic programming instance	can identify and correctly formulate an advanced dynamic programming instance
Graph and network algorithms 1f	cannot identify a graph and network instance	can identify a graph and network instance, but the formulation is not completely correct	can identify and correctly formulate a graph and network instance	can identify and correctly formulate an advanced graph and network instance
Geometric and combinatorial aspects of linear optimization and other optimization methods 1g	does not show good grasp of the geometric and combinatorial aspects of linear optimization	shows a limited understanding of the geometric and combinatorial aspects of linear optimization	shows a solid understanding of the geometric and combinatorial aspects of linear optimization	shows a full understanding of the geometric and combinatorial aspects of linear optimization

Table 2: Rubric: **What students should be able to do**
Student work used: assignments, midterm, and final exams.

	Expectations			
Topic	**Below**	**Marginal**	**Meets**	**Exceeds**
Solve a linear optimization instance 2a, 2b, 2c, 2f A1, A2: formulate and solve linear optimization instances	cannot properly formulate and solve a linear optimization instance	can properly formulate, but not properly solve a linear optimization instance	can properly formulate and solve a linear optimization instance	can properly formulate and solve a linear optimization instance, and add comments and interpretations
Solve a dynamic programming instance 2d, 2f, 2g A2: formulate and solve dynamic programming instances	cannot properly formulate and solve a dynamic programming instance	can properly formulate, but not properly solve a dynamic programming instance	can properly formulate and solve a dynamic programming instance	can properly formulate and solve a dynamic programming instance, and add comments and interpretations
Solve a graph or network instance 2e, 2f, 2g A3: formulate and solve a graph or network instance	cannot properly formulate and solve a graph or network instance	can properly formulate, but not properly solve a graph or network instance	can properly formulate and solve a graph or network instance	can properly formulate and solve a graph or network instance, and add comments and interpretations
Explain and show correctness of optimization methods and algorithms 2f, 2g A1, A2, A3: formulate and solve optimization instances	cannot properly explain and show correctness of an optimization instance	can properly explain, but not show correctness of an optimization instance	can properly explain and show correctness of an optimization instance	can properly explain and show correctness of an optimization instance, and add comments and interpretations