Advanced Algorithms

Floriano Zini

Free University of Bozen-Bolzano
Faculty of Computer Science

Academic Year 2013-2014

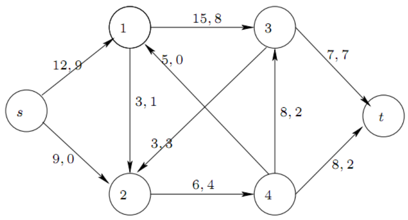
Lab 9 – Exercises on network optimization and others

Exercise 1

Network Flow

In the network reported in the following figure the pair of numbers on each edge indicate the capacity and the current flow on that edge, respectively

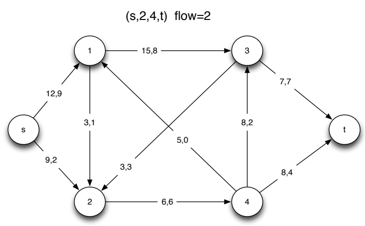
- Find the maximum flow from *s* to *t* using the Edmond-Karp algorithm



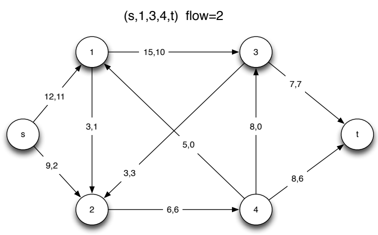
Exercise 1

Solution

- Iteration 1



- Iteration 2



Maximum flow = 11+2 = 7 +6 = 13

Exercise 2

The knapsack problem is a problem in combinatorial optimization that can be summarized as follows.

- A thief breaks into a house carrying a knapsack
- The capacity of the knapsack is limited to *m* Kgs, since the thief can carry at most *m* Kgs
- In the house there are *n* objects among which the thief can choose
- Each object *i* has some weight w_i and some value v_i

The thief has to choose which objects to steal in order to maximize his revenue but not exceeding the weight he can carry

This is called 0-1 knapsack problem because each object can either be stolen (1) or not stolen (0)

A. Provide an LP formulation of the 0-1 Knapsack problem

B. How a brute-force approach will work? Which is the complexity of this brute force algorithm?

Exercise 2

Solution

A. This is an optimization problem for solving which we can use *n* binary variables $x_1, \ldots, x_n$, one for each object in the house

- A linear programming formulation for the 0-1 knapsack is the following:
  - maximize $\sum^{n}_{i=1} v_i * x_i$
    Subject to
    - $\sum^{n}_{i=1} w_i * x_i \leq m$
    - $x_i \in \{0,1\}, 1 \leq i \leq n$

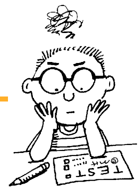
Exercise 2

Solution

B. Brute-force approach

- Since there are *n* objects, there are 2^n possible combinations of these objects
- For each combination we have to compute the value and the weight with cost *O(n)*, in order to find the one with the maximum value and total weight less than or equals to *m*
- Thus, the total cost is $O(n*2^n)$

Assignment 08



Exercise 7.18 page 226 DPV

There are many common variations of the maximum flow problem. Here are three of them.

A. There are many sources and many sinks, and we wish to maximize the total flow from all sources to all sinks

B. Each vertex also has a capacity on the maximum flow that can enter it

C. Each edge has not only a capacity, but also a lower bound on the flow it can carry

Each of these variations can be solved efficiently. Show this by reducing (A) and (B) to the original max-flow problem, and reducing (C) and to linear programming

Assignment 08

Exercise

- Consider the 0-1 knapsack problem of Exercise 2

A. Propose some greedy strategy to solve the problem and find a counterexample for each of them

B. Consider the greedy strategy of selecting at each step the object with the maximum value per unit of weight and provide a counterexample showing that this strategy does not return the optimal solution