Advanced Algorithms

Floriano Zini

Free University of Bozen-Bolzano
Faculty of Computer Science

Academic Year 2013-2014

Lab 9 – Solution of assignments
Assignment 08

Exercise 7.18 page 226 DPV

Solution

A. Suppose we have a flow network G that has multiple sources $S_1, S_2, ..., S_n$ and multiple sinks $T_1, T_2, ..., T_m$. Each source $S_i$ has a set of outgoing edges $E_i$ ($i = 1, ..., n$), and each sink $T_j$ has a set of incoming edges $F_j$ ($j = 1, ..., m$). The max-flow problem on G can be reduced to the original max-flow problem by constructing a network $G'$ from G as follows:

- We introduce two additional vertices $S$ and $T$.
- We construct $n$ edges $e_1, e_2, ..., e_n$ each of them going from $S$ to $S_1, ..., S_n$.
- We construct $m$ edges $f_1, f_2, ..., f_m$ each of them going from $T_1, ..., T_m$ to $T$.
- For each $e_i$ from $S$ to $S_i$, $e_i$ has capacity equal to the sum of the capacities of all edges in $E_i$.
- For each $f_j$ from $T_j$ to $T$, $f_j$ has capacity equal to the sum of the capacities of all edges in $F_j$.
- $S$ is the single source of $G'$ and $T$ is the single sink of $G'$.
- The original $S_1, ..., S_n$ and $T_1, ..., T_m$ are treated as transshipment nodes.

Assignment 08

Exercise 7.18 page 226 DPV

Solution (cont.)

B. Suppose we have a flow network G with a single source $S$, a single sink $T$ and $n$ transshipment nodes $N_1, ..., N_n$. For each vertex but the source, G has also a maximum flow that can enter the vertex. Let $f_i$ be the maximum flow that can enter $N_i$, $i = 1, ..., n$, and let $f_t$ the maximum flow that can enter the sink node $T$. The max-flow problem on G can be reduced to the original max-flow problem by constructing a network $G'$ from G as follows:

- $G'$ has $S$ as source and $T$ as sink.
- For each $N_i$, $i = 1, ..., n$, $G'$ has two nodes $M_i$ and $M_i'$ such that for each incoming edge of $N_i$ there is an equivalent incoming edge of $M_i$, and for each outgoing edge of $N_i$ there is an equivalent outgoing edge of $M_i'$; in addition, there is an edge $e_i$ from $M_i$ to $M_i'$, whose capacity is $f_i$.
- For the sink $T$, we introduce in $G'$ a new sink node $U'$ and an additional transshipment node $U$ such that for each incoming edge of $T$ there is an equivalent incoming edge of $U$, and there is an edge $e_i$ from $U$ to $U'$, whose capacity is $f_i$. 
Assignment 08

Exercise 7.18 page 226 DPV

Solution (cont.)

C. Suppose we have a flow network G and each edge of G has not only a capacity, but also a lower bound on the flow it can carry. The max-flow problem on G can be modeled in LP similarly to reducing a normal max-flow problem. The only difference is that for every edge e in G, instead of the capacity and non negativity constraints \(0 \leq e \leq c_e\) (where \(c_e\) denotes the capacity of edge e), we have \(c_e_{\text{min}} \leq e \leq c_e_{\text{max}}\) (where \(c_e_{\text{min}}\) and \(c_e_{\text{max}}\) denote the lower and the upper bounds of the edge capacity, respectively).

The flow conservation constraints are the same and also the objective function to be maximized does not change.

Assignment 08

Exercise Solution

A. Some greedy strategies are:

1. Repeatedly pick the item with the maximum value
   Counter example:
   \(n = 3, m = 3\) kg
   \(i_1: v_1 = 7\) $; \(w_1 = 0.5\) kg
   \(i_2: v_2 = 9\) $; \(w_2 = 1\) kg
   \(i_3: v_3 = 14\) $; \(w_3 = 2.8\) kg
   Greedy solution: \((i_3, i_1)\)
   Optimal solution: \((i_1, i_2)\)

2. Repeatedly pick the item with the minimum weight
   Counter example:
   \(n = 3, m = 3\) kg
   \(i_1: v_1 = 15\) $; \(w_1 = 0.5\) kg
   \(i_2: v_2 = 6\) $; \(w_2 = 1\) kg
   \(i_3: v_3 = 14\) $; \(w_3 = 2\) kg
   Greedy solution: \((i_2, i_3)\)
   Optimal solution: \((i_1, i_3)\)

3. Repeatedly pick the item with the maximum weight
   Counter example:
   \(n = 3, m = 3\) kg
   \(i_1: v_1 = 15\) $; \(w_1 = 0.5\) kg
   \(i_2: v_2 = 6\) $; \(w_2 = 1\) kg
   \(i_3: v_3 = 14\) $; \(w_3 = 2\) kg
   Greedy solution: \((i_2, i_3)\)
   Optimal solution: \((i_1, i_3)\)
Assignment 08

Exercise
Solution (cont.)

B. Repeatedly pick the object with the maximum value per unit of weight

Counter example:

n (number of items) = 3, m (max weight in knapsack) = 3 kg
i₁: v₁ = 5 $; w₁ = 0.5 kg; v₁/w₁ = 10
i₂: v₂ = 6 $; w₂ = 1 kg; v₂/w₂ = 6
i₃: v₃ = 14 $; w₃ = 2 kg; v₃/w₃ = 7
Greedy solution: {i₃, i₁}
Optimal solution: {i₂, i₃}