Applied Combinatorial Optimization

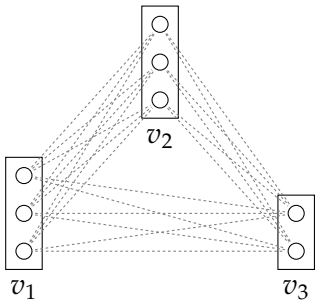
Exercise sheet 1

Dr. Bogdan Savchynskyy <bogdan.savchynskyy@iwr.uni-heidelberg.de>
Sebastian Stricker <sebastian.stricker@iwr.uni-heidelberg.de>

———— ⋆ ————

Each exercise is worth 1 point.

Exercise 1.1 To get started, implement the computation of the total cost for an arbitrary labeling of an arbitrary labeling problem. Evaluate the objective for the problem instances described below.



$\theta_1 = [0.3, 0.9, 0.4]$,
$\theta_2 = [0.8, 0.1, 0.3]$,
$\theta_3 = [0.2, 0.5]$

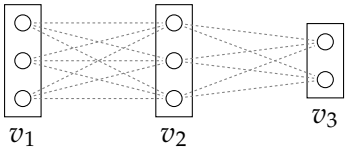
$$\forall \{u, v\} \in \mathcal{E} : \theta_{uv} = \begin{cases} 0 & \text{if } x_u = x_v, \\ \lambda & \text{otherwise.} \end{cases}$$

	$\lambda = 0$	$\lambda = 0.2$	$\lambda = 0.5$
$y = [0, 0, 0]$			
$y = [0, 2, 1]$			

We provide example code that you can use on our website (`example_1_2.py`). Test your implementation with the provided bigger model (`test_1_2.py`).

Exercise 1.2 Implement a function `bruteforce_min(nodes, edges)` that computes the optimal objective value $\min_{y \in \mathcal{Y}_\mathcal{V}} E(\theta, y)$ using a brute-force technique. Test your implementation with the problem instances from the previous exercise. Check your implementation on the test-cases provided. Derive the time complexity of the brute-force algorithm.

Exercise 1.3 For this exercise assume that the labeling problem is chain-structured. As an example you can remove the edge $e = (v_1, v_3)$ from the graph provided in Exercise 1.1 and obtain the following chain-structured problem instance:



$\theta_1 = [0.3, 0.9, 0.4]$,
$\theta_2 = [0.8, 0.1, 0.3]$,
$\theta_3 = [0.2, 0.5]$

$$\forall \{u, v\} \in \mathcal{E} : \theta_{uv} = \begin{cases} 0 & \text{if } x_u = x_v, \\ \lambda & \text{otherwise.} \end{cases}$$

Implement the dynamic programming algorithm discussed in the lecture to compute all Bellman functions. Additionally, implement the backtracking pass to infer the optimal label assignment. What is the time complexity of this algorithm?

Exercise 1.4 Seam carving[1] is a technique for content-aware image rescaling. Implement this method by modeling the problem as a labeling problem.

$$\theta_i(j) = \|\Delta_{\text{hor}}(i,j)\| + \|\Delta_{\text{ver}}(i,j)\| \qquad \theta_{i,i+1}(j,j') = (j-j')^2$$
$$\Delta_{\text{hor}}(i,j) = p(i,j-1) - p(i,j+1) \qquad \Delta_{\text{ver}}(i,j) = p(i-1,j) - p(i+1,j)$$

You can solve the resulting chain-structured labeling problem to remove a single pixel from each row efficiently by using your dynamic programming approach implemented in Exercise 1.3. Resize the provided image `tower.jpg` with this technique by iteratively removing pixels until the image width equals 100 pixels.

[1] Avidan, Shai, and Ariel Shamir. "Seam carving for content-aware image resizing." ACM Transactions on graphics (TOG). Vol. 26. No. 3. ACM, 2007. https://dl.acm.org/citation.cfm?id=1276390
Short overview: https://en.wikipedia.org/wiki/Seam_carving