Exercise Sheet #10

Due: 14.07.2020, 15:15

Fred Hamprecht, Alberto Bailoni
cvflecture.s20@gmail.com

General Regulations

- No individual submission! Hand in your solutions in groups of two or three people (preferred).
- All outputs, in particular images, should be visible in the notebook. We should not have to run your notebook or uncomment lines / change some code to see your results!
- When asked, remember to comment / interpret your results.
- Submit all your files in a single `.zip` archive (not `.rar` or `.tar`) and upload it using the Physics platform https://uebungen.physik.uni-heidelberg.de/h/1176. **Only one person in the group should upload the zip archive**: in the upload interface you will see the option to indicate your exercise-partners.

1 Minimum cost flow and tracking by assignment

In this exercise, our goal is to track particles in Brownian motion[1] appearing in the toy video shown in Fig. 1, which has a resolution of 7×7 pixels and three time frames. An object detector is applied to the video and returns a total of six detections $\{v_1, \ldots, v_6\}$: our task is to assign these detections to a number of particles N_{tot} not known a priori, by keeping in mind that not all the detections have to be assigned to a particle. We also make the following assumptions:

- Each detection v_i has an associated negative cost c_i shown in Fig. 1: the higher the confidence of the detection i, the lower the cost c_i. In other words, if a detection has high confidence, we get an higher reward by assigning that detection to an actual particle.
- In order to assign two detections v_i (at time t) and v_j (at time $t+1$) to the same particle, we need to pay the following cost:
$$c_{ij} = d^2(v_i, v_j) \tag{1}$$
where $d(v_i, v_j)$ is the euclidean distance between v_i and v_j in the given 7×7 grid.

1.1 Tracking without particle merging or division (12pt+2pt bonus)

(a) Directed graph (2pt) Similarly to what was done during the lecture, formulate the tracking assignment problem as a minimum cost flow problem. First, draw the associated directed graph G and for each of its directed edges define a capacity μ_e and a cost c_e. Assume that particles cannot divide/merge and that we only allow particles to appear at frame $t = 1$ and disappear after frame $t = 3$.

(b) Optimization problem (1pt) Given a flow x_e associated to each edge of the graph, define the minimum cost flow optimization problem as it was done during the lecture.

(c) Maximum flow (2pt) Given the current assumptions (no particle division, merging, appearance and disappearance), what is the maximum number of particles N_{tot} that can be assigned? How does this number change if we allow the appearance of new particles at frames $t > 1$? What if we also allow the disappearance of particles after frames $t = 1$ and $t = 2$?

(d) Shortest-path (1pt) Given your answer to the previous task, can you say in which special case the minimum cost flow problem reduces to a shortest path problem? Explain your answer.

[1] Brownian motion is the random motion of particles suspended in a fluid (a liquid or a gas) resulting from their collision with the fast-moving molecules in the fluid.

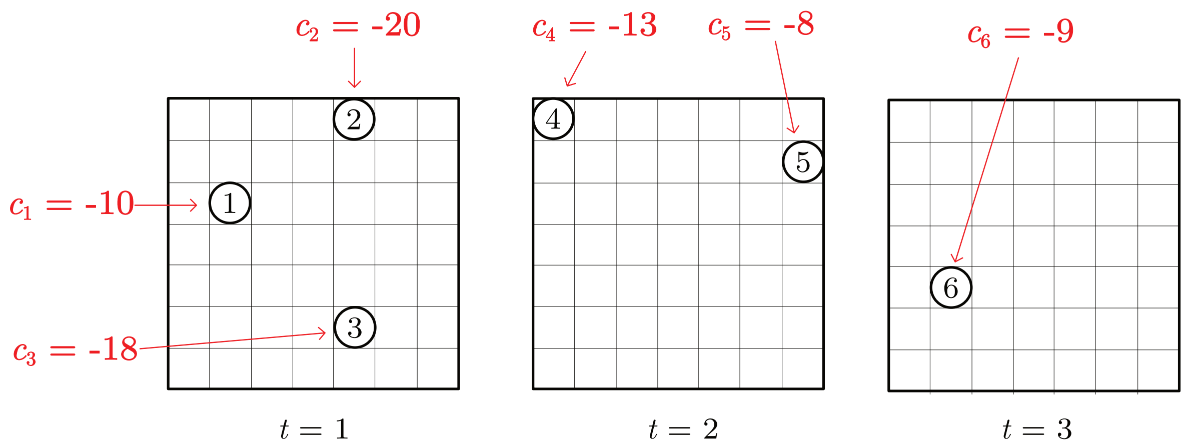

Figure 1: Detections of particles on a small video with three frames and a resolution 7×7.

(e) Zero particles case (1pt) Can it happen that in the solution of the minimum cost flow $N_{\text{tot}} = 0$? Explain your answer.

(f) Linear Program (2pt) We now want to solve the optimization problem numerically by using the functions provided in `scipy`. If you have not already done it, rewrite the optimization problem as a linear program of the form (for the moment assume that particles cannot appear/disappear):

$$\min_x \; c^T x \tag{2}$$
$$\text{s.t.} \quad A_{\text{eq}}\, x = b_{\text{eq}} \tag{3}$$
$$A_{\text{leq}}\, x \leq b_{\text{leq}} \tag{4}$$
$$x_{\text{low}} \leq x \leq x_{\text{high}}. \tag{5}$$

Note that the problem can be written as a linear program (without integer constraints) only because, under the current assumptions (no particle merging/division), the associated constraint matrix is totally unimodular[2] *Hints: The matrix of constraints should look very similar to the incidence matrix associated to the directed graph G, feel free to compute it numerically if you prefer. Note that you may not need to define both A_{eq} and A_{leq}.*

(g) Solve the LP (3pt) Now solve the LP numerically by following the instructions given in the jupyter notebook `ex10.ipynb`. What is the resulting number of particles N_{tot}? Comment your results.

(h) Particle appearance and disappearance (2pt bonus) Now assume that a particle can appear at any frame with a cost $c_{\text{app}} = 6$ and disappear with a cost $c_{\text{dis}} = 10$. How does the solution change? In this new problem, can we say something about the maximum distance that a particle can travel between one frame and the next one? Explain your answer.

1.2 Bonus: Tracking with particle merging and/or division (10pt bonus)

In this second bonus part of the exercise, we will see how the tracking problem formulation changes when we allow particles to merge and divide.

(a) Incorrect models (2pt) Consider the directed graphs shown in Fig. 2, in which we would like to model the division of a particle. Assume that all edges have capacity equal to one (both the orange ones and

[2] This follows from the fact that any incidence matrix of a directed graph is totally unimodular. For more interesting properties about total unimodularity, see the associated wikipedia page or the slides at the following link: https://wwwhome.ewi.utwente.nl/~uetzm/do/DO_Lecture6.pdf

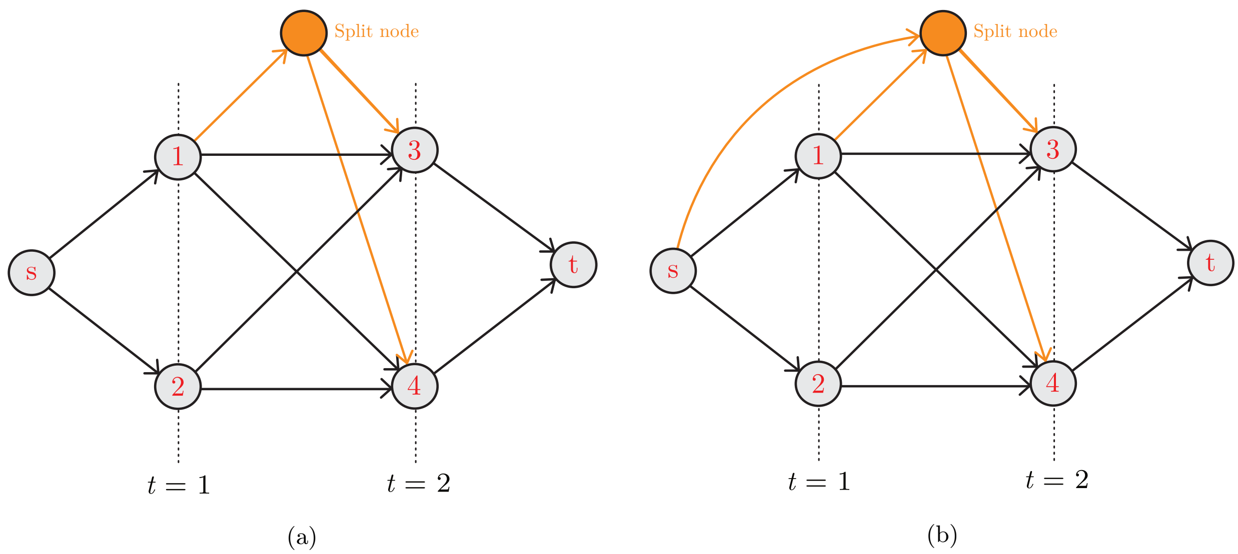

Figure 2: Two possible ways to model the division of one particle. In the example, detection 1 divides into detections 3 and 4. All edges have capacities equal to one.

the black ones) and that the *split-node* vertex is not different from the others. Explain in words why solving a standard minimum cost flow (like the one you used in the previous exercise) on the two directed graphs shown in Fig. 2a and 2b does not correctly model the particle division. *Hint: It may be useful to read section 3.3 of paper [1]. You can find the* `.pdf` *file of the article in the assignment material.*

(b) Flow coupling constraints (1pt) A flow coupling constraint makes sure that two different edges in the graph always have the same flow. Where would you introduce an additional flow coupling constraint in the model of Fig. 2b to make sure that the particle division is correctly modeled?

(c) Particle merging (1pt) How would model a merge between two particles? Draw a simple example describing your idea.

(d) Minimal counterexample (3pt) We now have models to correctly describe particle divisions and merging. However, introducing these additional flow coupling constraints comes at a cost: it is indeed possible to show that this coupled version of the minimum cost flow problem[3] is NP-hard [2]. Now construct a minimal min-cost flow problem with flow coupling constraints and show that it cannot be represented by a LP with a totally unimodular constraint matrix. What can you conclude from this result? *Hint: in order to prove that the constraint matrix is not totally unimodular, you should find a small squared sub-matrix that has determinant different from* $0, 1, -1$.

(e) Non-Integral solutions (1pt) For your minimal counterexample, find at least one cost vector that does and one that does not result in an integral solution.

(f) Discuss the method proposed in [1] (2pt) In Section 3.4 of [1], the authors reformulate this problem in terms of a directed hypergraph[4] and then solve the associated LP. In view of your previous findings, discuss the method proposed by the authors.

[3] In the literature, a minimum cost flow problem with flow coupling constraints is also known as *minimum integer equal flow problem* or *minimum integral flow with homologous arcs problem.*

[4] A hypergraph is a generalization of a graph in which an edge can join any number of vertices.

References

[1] D. Padfield, J. Rittscher, and B. Roysam. Coupled minimum-cost flow cell tracking for high-throughput quantitative analysis. *Medical image analysis*, 15(4):650–668, 2011.

[2] S. Sahni. Computationally related problems. *SIAM Journal on computing*, 3(4):262–279, 1974.