Foundations of Algorithms
Homework #7, Spring 2023

All members of the collaboration group are expected to participate fully in solving collaborative problems. Note, however, that each student is required to write up their solutions individually. Common solution descriptions from a collaboration group will not be accepted. Furthermore, to receive credit for a collaboration problem, each student in the collaboration group must actively and substantially contribute to the collaboration. This implies that no single student should post a complete solution to any problem at the beginning of the collaboration process.

Problems for Grading

1. [100 points] (Integer Programming) Multi-object tracking within optical video is a common problem in machine learning. Tracking is extremely difficult in general as the number, sizes, and dynamics of objects can be large. Further, objects can by occluded by other objects and scenery, causing variable time gaps between one detection and the next.

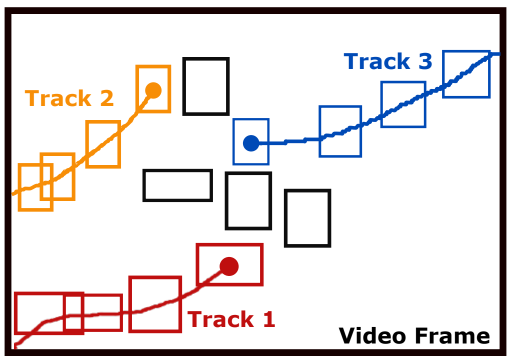

Figure 1: A tracking system that has three active tracks and has received four detections (black boxes) from the *Detector*. Note that the time intervals between detections assigned to tracks varies, as does the shape of their bounding boxes.

Tracking systems require several key components:

- **Detector**– A model that signals when an object of interest is apparently present, and returns a bounding box in pixel coordinates.
- **Tracker**– Software module that stores information regarding active tracks, and assigns object detections to tracks using a similarity score matrix produced by the *Evaluator*. Optimally, the *Tracker* assigns detections to tracks in a manner that maximizes the sum of the similarity scores from each track/detection pair.
- **Evaluator**– Given N active tracks and M detections, produces a score matrix $\boldsymbol{S} \in \mathbb{R}^{M \times N}$ such that element $s_{ij} \in \boldsymbol{S}$ indicates the similarity of detection i to track j under some similarity measure. Note that this definition leaves the actual definition of "track" fairly abstract. In the simplest case, for example, the *Evaluator* may define s_{ij} as the Euclidean distance between the center of the final detection within track j and the new detection i. More sophisticated models may use multiple past detections within a track, or model the dynamics of an objects (e.g., with a Kalman filter).

In this problem, assume that you'd like to produce a novel tracking system. The *Detector* and *Evaluator* have already been created, and you must now describe the *0-1 integer linear program* that the *Tracker* will solve to assign detections to tracks. **Integer linear programs are constraint problems where the variables are restricted to be integers, and are typically NP-hard. In particular, 0-1 integer linear programs restrict the variables further by requiring them to be Boolean (i.e., zero or one).**

Assume that you are provided with the score matrix S, detections can only be matched to a single track, and that tracks cannot be assigned more than once.

(a) [30 points] What do the variables in this problem represent? How many are there?

(b) [30 points] Define the objective for this 0-1 integer linear program.

(c) [40 points] Define the entire 0-1 integer linear program, including constraints, in standard form. How many constraints are there in the program, total?