CMPUT 497 / Winter 2017 / Ioanis Nikolaidis

Motivating Example – Two Bouncing Nodes

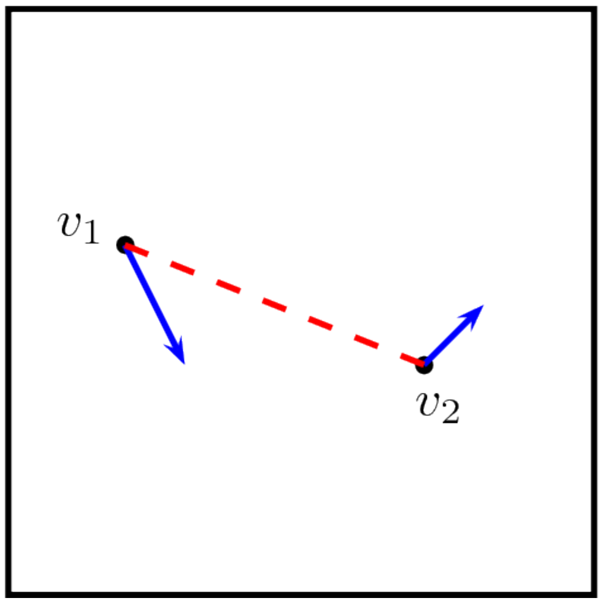

Figure 1: Two nodes bouncing at the boundary.

We wish to simulate the behavior of two nodes that move independently of each other. Their world is a square. When a node reaches a boundary, it bounces back into the interior of the square. The nodes are assumed to never collide. The movement of a node between any two successive bounces is characterized by constant speed and direction (represented by the velocity vectors v_1 and v_2 respectively). The velocity changes only when the node bounces at the boundary. Let us assume that the velocity, when the node bounces at the boundary is chosen from a uniform random distribution. For reasons of simplicity let the x-axis speed (seen as displacement per unit of time) is chosen separately from the y-axis speed (seen again as displacement per unit of time) and each of the two is chosen from a uniform random distribution in the range of s_{min}, s_{max}. Additionally the angle of reflection/bounce is chosen randomly uninformly each time a node bounces at a boundary (unlike real physical systems where angle of incidence is equal to the angle of reflection). So in essence, the uniformly random angle is chosen between -90 and 90 relative to the normal, pointing to the "interior" of the square.

Let's now define the Euclidean distance of the nodes (seen in red dashed line in the figure). Assume an additional parameter r indicating when the two nodes can be assumed to be in communication range with each other. That is, if their distance is less or equal to r they can communicate. You can assume that the side of the square is equal to a unit of distance, hence r can be expressed relative to this unit. The same, i.e., relative to the side of the square, you can express the displacement per unit of time. In short, your model parameters are: s_{min}, s_{max}, r.

Exercise

Write a simulation of the two bouncing nodes. The simulation will take as input the three parameters, and two additional parameters: one which will be the value to use to seed the random number generator(s) it uses, and one which is the duration of the simulation.

The output of the simulation are the statistics of the node "encounters". An encounter is the interval from the moment the two nodes get sufficiently close to each other (distance r between them), continue moving and continue being within r or less to each other, and then, eventually (end of encounter) they become more than r apart. Clearly, a large number of encounters can occur during the simulation.

Specifically, the output of a single simulation run is the frequency histogram of these encounter times, as well as the average encounter time.

A number of questions of how to correctly and efficiently code the solution will be asked during our class meeting. Cheers!