Dynamic matching

Sujet de stage M2

Context

We consider a dynamic matching model with random arrivals – a dynamic version of the bipartite matching model. As in the static setting, it is based on a bipartite graph. In the discrete-time dynamic model there are arrivals of units of ‘supply’ and ‘demand’ that can wait in queues located at the nodes in the network. A control policy determines which items are matched at each time.

The theory of matching has a long history in economics, mathematics, and graph theory [7], with applications found in many other areas such as chemistry and information theory. Most of the work is in a static setting. The dynamic model has received recent attention in [5, 3, 6]. The most compelling application is organ donation: United Network for Organ Sharing (UNOS) offers kidney paired donation (KPD). This is a transplant option for candidates who have a living donor who is medically able, but cannot donate a kidney to their intended candidate because they are incompatible (i.e., poorly matched) [1]. In this and many other applications, data arrives sequentially and randomly, so that matching decisions must be made in real-time, taking into account the uncertainty of future requirements for supply or demand. The choice of matching decisions can be cast as an optimal control problem for a dynamic matching model.

Objectives

The objective of this internship is to study the performance (the number of unmatched customers/servers in the system) under various matching policies proposed in the literature [5, 3, 2]: MS (Match the Shortest), FIFO (match the oldest), and priorities. There are two possible directions for the internship:

• searching for heuristics to minimize the total number of unmatched customers/servers in the system using reinforcement learning [8] and some recent results on the properties of the optimal policy [4];

• extending the model to include impatience (unmatched items may leave the system after some delay).

Prerequisites: Random structures and algorithms, queueing networks.
Programming skills: Python or Matlab.

Contact: Ana Busic (ana.busic@inria.fr)
Inria Paris and
Département d’Informatique de l’École normale supérieure (DI ENS), PSL University
References


