Seventh Annual
Cyber-Physical Systems Principal Investigators' Meeting
Arlington, VA | October 31 – November 1, 2016

Design of Network Dynamics for Strategic Team-Competition

PIs: Carolyn Beck, Angelia Nedić, Alex Olshevsky

CNS 15-44953

Abstract:

We consider the competitive-strategic domain for two opposing teams of agents, modeled in the framework of a repeated game over network. Motivating applications include those in which a large collection of autonomous agents must play a game against an adversary or team of adversaries.

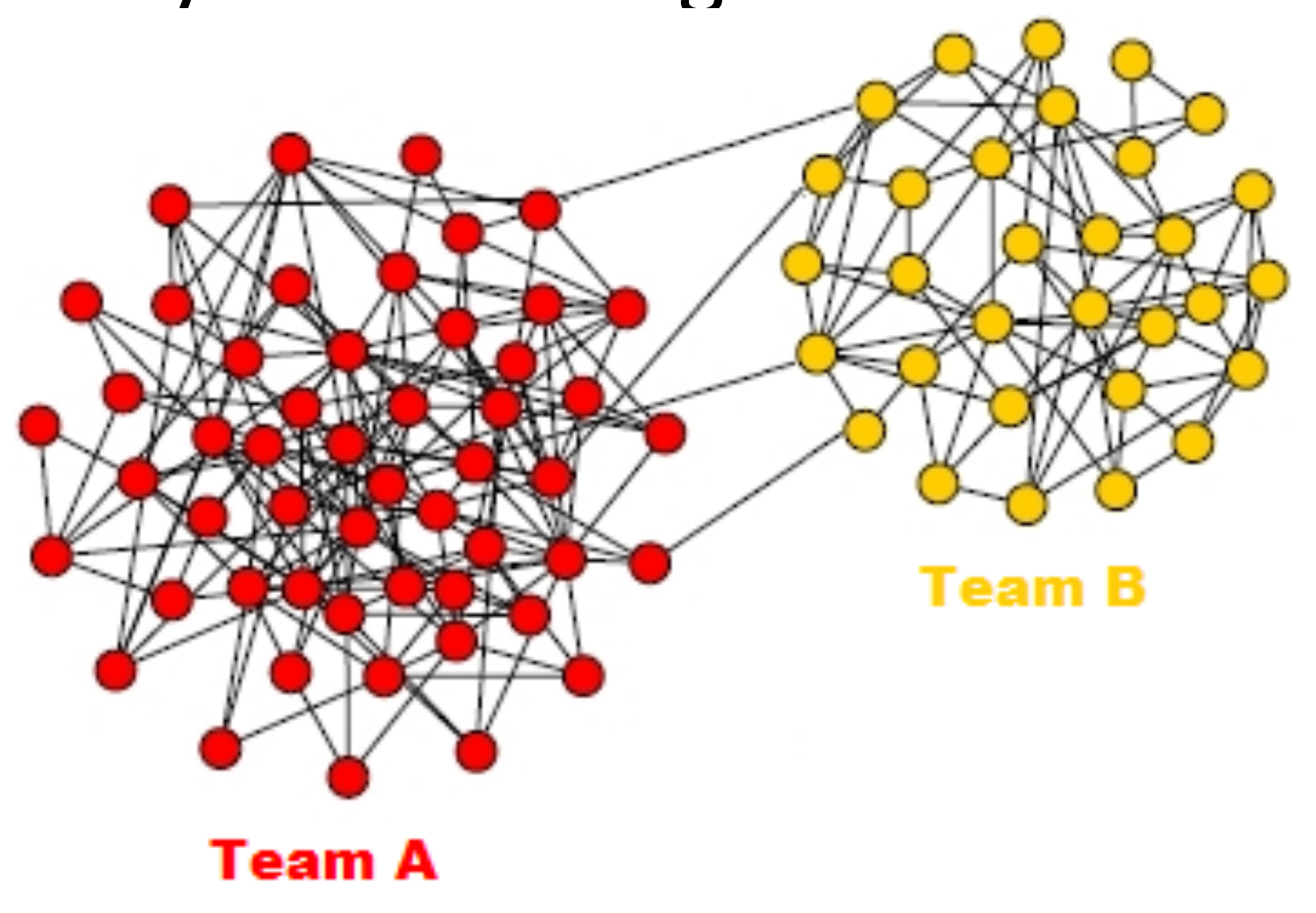
One example is the problem of controlling large arrays of wind farm turbines: each rotating blade creates a downstream wake and every turbine must determine the optimal rotation speed in the face of complex aerodynamic interactions. Faster rotation by one turbine may lead to lower wind speed at downstream turbines; further complications result from unpredictable changes in wind speed and direction. We would like to extract maximum power from the wind turbines; this comprises a minimum regret game.

Goals:

- Dynamic games among networked players
- Decentralized dynamic clustering
- Dynamic information aggregation in networks

Framework:

- Two opposing teams of networked agents interconnected over a network
- Collaboration within teams; competition between teams
- Subset of agents in Team A have partial information about states of subset of agents in Team B
- Actions of opposing team partially observed by subset of agents



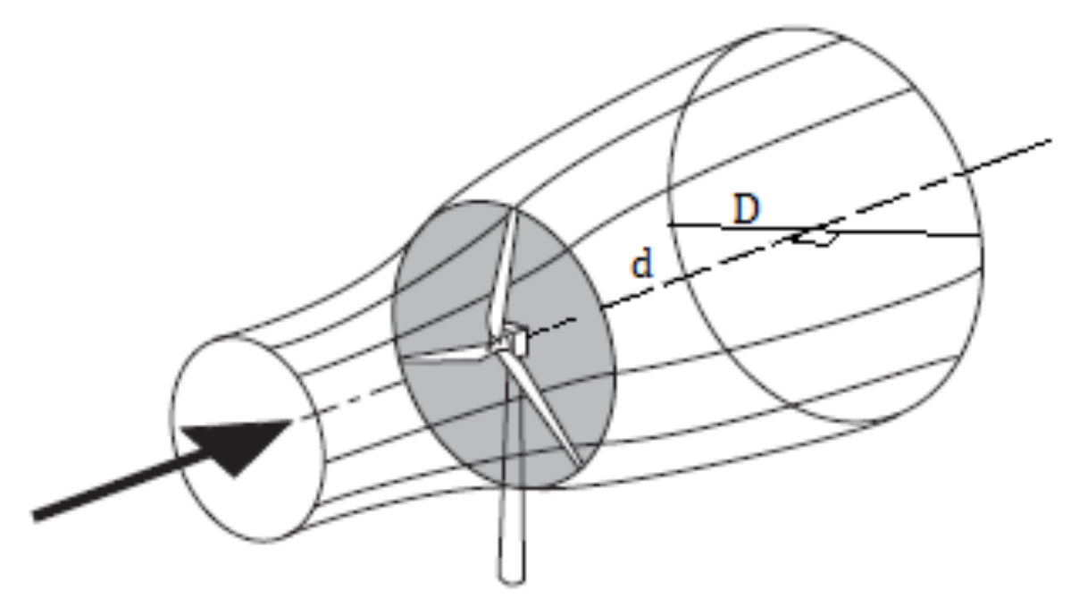
Wind Farm Model

- Given uniform wind with magnitude U_∞, the Park wake model states that each turbine affects wind velocity downstream by a diminishing multiplicative factor

$$V_i = U_\infty(1 - \Delta V_i)$$

with

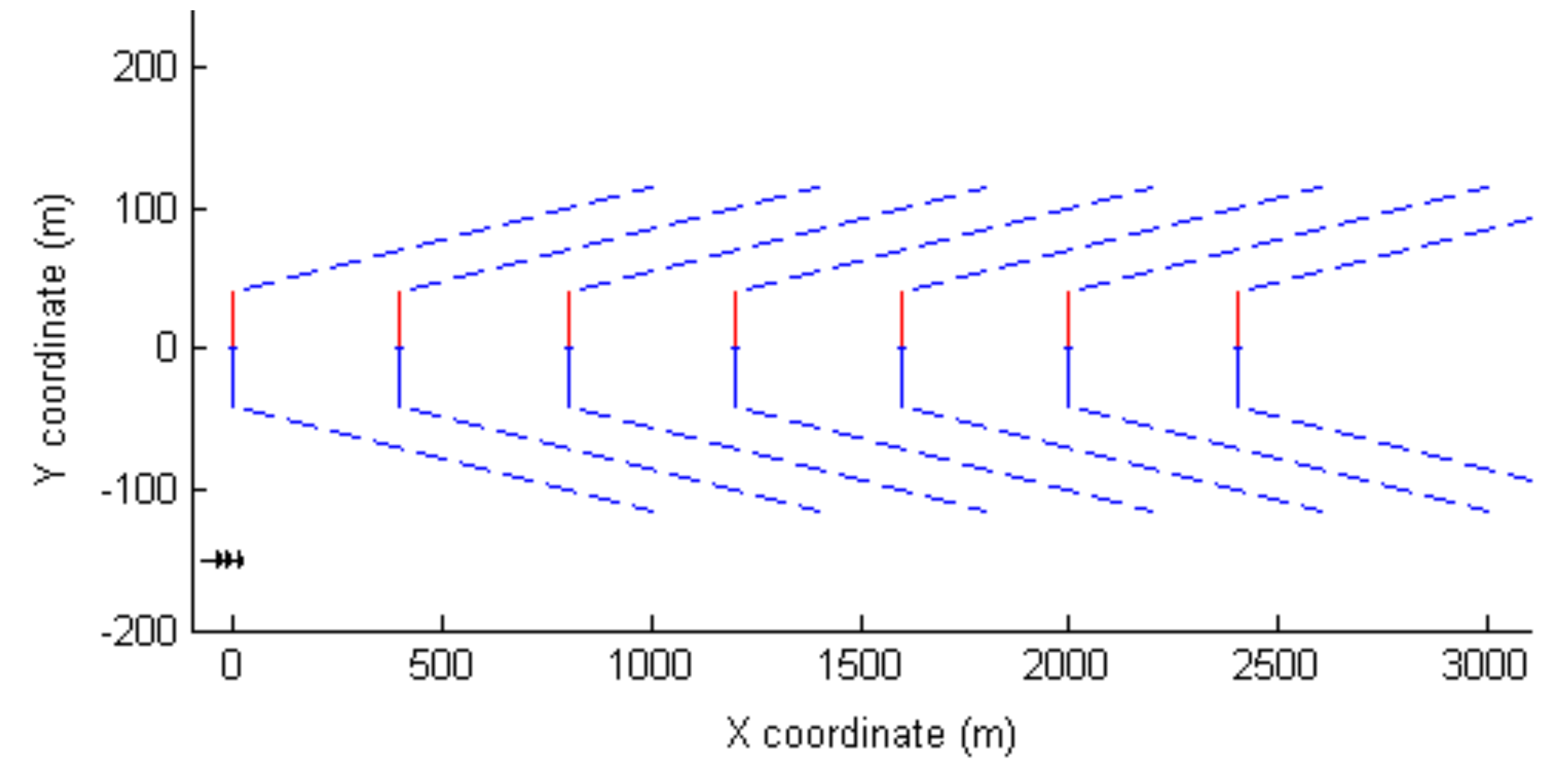
$$\Delta V_i = 2u_i\left(\frac{D}{D+2K\Delta d}\right)^2$$



- At a given location in the intersection of wakes 1…k, the aggregate wind velocity is determined by

$$V_i = U_\infty\left(1 - 2\sqrt{\sum_{i=1}^{k}(p_i \Delta V_i)^2}\right)$$

Preliminary Results: Wind Farm



- Given an array of turbines as shown above, the optimal solution, found by exhaustive search is shown below

$$u_{opt} =$$

$$\begin{pmatrix} 0.2260 & 0.1910 & 0.1940 & 0.1960 & 0.1990 & 0.2110 & 0.3333 \end{pmatrix}$$

- Our DP based, non-decreasing algorithm converges much more quickly to

$$u_{array} =$$

$$\begin{pmatrix} 0.2265 & 0.1856 & 0.1874 & 0.1925 & 0.1950 & 0.2056 & 0.3333 \end{pmatrix}$$

NSF grant CNS 15-44953
beck3@illinois.edu, Angelia.Nedich@asu.edu
aolshev2@illinois.edu