Distributed Beamforming Problem

Problem: beampattern matching

- to match the desired beam and null values at multiple receiver locations
- need not consider separate problems for beam and null
- receiver locations can change over time
- channel can change over time

Approach: Optimization

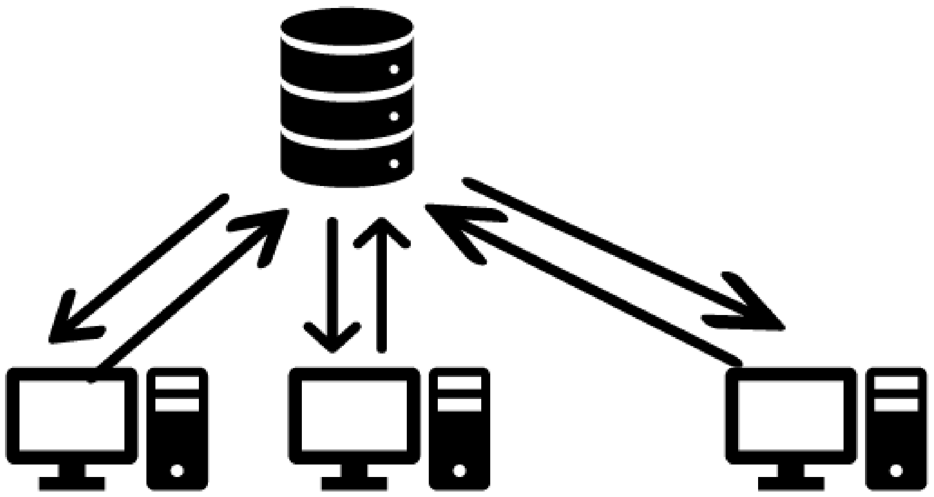
- coefficient perturbation method
  - optimization variables: beamforming phase and amplitude of each agent
  - agents' locations are fixed
  - minimize the sum-of-squares error between the desired beampattern and formed beampattern at the receiver locations
  - non-convex problem
- node perturbation method
  - node placement: select the best position of each agent
  - node selection: select a subset of nodes based on their positions and beamforming coefficients
  - additional variables: position of agents, active agents

Performance Measures

- open-loop or closed-loop
  - do the agents need feedback from the beam receivers and null receivers?
- guarantee convergence to a local or global optimum solution
  - does the algorithm converge for arbitrary initialization?
- rate of convergence
  - how many iterations are required to get a close enough solution?
- what is considered as close enough?
  - measure the gap in dB
- robustness
  - channel parameters may not be known precisely
  - channel noise, measurement noise

Proposed Setting

- server-agent based setting
  - agents can communicate with a common server
  - no peer-to-peer communication



- what is a server?
  - can exchange information with the agents
  - can process information (addition, multiplication, etc. basic operations)
  - need not know the beamforming coefficients, agents' and receivers' locations, desired beam and null values

Proposed Algorithm: IPG

- works in server-agent settings
- open-loop or closed-loop
  - does not require feedback from the receivers
- assumes channel information
  - multipath fading, path loss
- guarantee convergence to a local or global optimum solution
  - needs to be analyzed
- rate of convergence
  - faster than conventional and adaptive gradient-descent methods
  - simulated for uniform and non-uniform weights on the samples
- only simulated with fixed agent locations on 1-D and 2-D
- robustness has not been considered
- issue in simulation for moving agents

Questions

- existence of a solution
  - given the number of agents, number of receivers, receiver locations, beam and null values, can the desired beam be formed?
- resolve issue for the case of moving agents
- integrate with node selection method
  - alternate minimization approach
  - select active agents: as it is
  - iteratively update position of each agent : use IPG
  - check if open-loop or not
- simulate the working algorithm in phase-array toolbox
- acceleration without a server and assuming peer-to-peer *connected* network of agents