Coordinated Versus Decentralized Exploration In Multi-Agent Multi-Armed Bandits

Mithun Chakrabortya, Kai Yee Phoebe Chuaa, Sanmay Dasb, and Brendan Juba2,3,4

1Department of Computer Science and Engineering, Washington University in St. Louis
2School of Information and Computer Science, University of California, Irvine

Multi-Agent Multi-Armed Bandit Problem

Motivation: Development and solution of a model for ad hoc teamwork/6/5 where a population of robotic or software agents works cooperatively towards a common goal without pre-coordination and with expensive communication.

At each epoch \( t = 1, 2, \ldots, T \), arm \( a \) generates reward \( r_a \), \( \mathbb{E}[r_a] \) and \( \varphi \) (although expensive) in general, and VoI in particular. Both sets demonstrate improvement in performance by including and with expensive communication.

[ Haitam's notes on page 1: 
- Coordinated: 
  - Public agent maintains mobility distribution across arms \( P_{A}(a_1, a_2, \ldots, a_n) \) and assigns arm \( a \rightarrow \mu \), agents to arm \( a \). 
  - During \( t \), observes all rewards \( r_a \), and updates using MWU: 
    \[ P_{A}(a) = \frac{P_{A}(a) + \beta (r_a - \bar{r}_a)}{\beta + 1} \] 
    where \( \beta \in (0, 1) \), \( \bar{r}_a \rightarrow 0, Z_t = \sum P_{A}(a) \frac{r_a - \bar{r}_a}{\beta} \). 
- For performance metric, time-averaged per-agent cumulative (bandit) regret \( \mathcal{R} \) is given by:
  \[ \mathcal{R}(T) = \min_T \left[ \frac{1}{T} \sum r_{a} P_{A}(a) + \frac{1}{T} \sum r_{a} P_{A}(a) \right] \]
  
- This MWU algorithm is equivalent to decreasing softmax strategy on empirical means with temperature \( \tau = \frac{1}{\log T} \).

- Decentralized: 
  - Softmax with simple communication strategy

  - For each agent, each epoch \( t \) is either an action round or a broadcast round, \( t = 1 \) being an action round for every agent.

  - Action round:
    - Combine private table with public agent to compute private empirical means \( \mu_i = 1 / t \sum_j r_{ij} \), where \( r_{ij} \) is the reward obtained from arm \( a_i \) in epoch \( t \).
    - Apply decreasing Softmax with temperature \( T \) as in (1) to generate probability distribution over arms.
    - Draw an arm from above distribution; collect reward and add to team’s cumulative reward; update private table.
    - Decide whether next epoch is action round or broadcast round by Vol strategy shown in Flowchart 1 on the right.

  - For each agent, each epoch \( t \) is either an action round or a broadcast round, \( t = 1 \) being an action round for every agent.

  - Each agent \( j \) also has private table containing, for each arm \( a_i \):
    - Number of times \( t \) has pulled arm \( a_i \).
    - Sum of rewards gained \( b \) from above pulls.
    - Number of pulls among the above that \( j \) has broadcast:\( b_{ji} \).
    - Sum of rewards from arm \( j \) corresponding to above broadcast pulls:\( b_{ji} \).

  - For each agent, each epoch \( t \) is either an action round or a broadcast round, \( t = 1 \) being an action round for every agent.

  - Action round:
    - Combine private table with public agent to compute private empirical means \( \mu_i = 1 / t \sum_j r_{ij} \), where \( r_{ij} \) is the reward obtained from arm \( a_i \) in epoch \( t \).
    - Apply decreasing Softmax with temperature \( T \) as in (1) to generate probability distribution over arms.
    - Draw an arm from above distribution; collect reward and add to team’s cumulative reward; update private table.
    - Decide whether next epoch is action round or broadcast round by Vol strategy shown in Flowchart 1 on the right.

  - For each agent, each epoch \( t \) is either an action round or a broadcast round, \( t = 1 \) being an action round for every agent.

  - Each agent \( j \) also has private table containing, for each arm \( a_i \):
    - Number of times \( t \) has pulled arm \( a_i \).
    - Sum of rewards gained \( b \) from above pulls.
    - Number of pulls among the above that \( j \) has broadcast:\( b_{ji} \).
    - Sum of rewards from arm \( j \) corresponding to above broadcast pulls:\( b_{ji} \).

  - For each agent, each epoch \( t \) is either an action round or a broadcast round, \( t = 1 \) being an action round for every agent.

  - Action round:
    - Combine private table with public agent to compute private empirical means \( \mu_i = 1 / t \sum_j r_{ij} \), where \( r_{ij} \) is the reward obtained from arm \( a_i \) in epoch \( t \).
    - Apply decreasing Softmax with temperature \( T \) as in (1) to generate probability distribution over arms.
    - Draw an arm from above distribution; collect reward and add to team’s cumulative reward; update private table.
    - Decide whether next epoch is action round or broadcast round by Vol strategy shown in Flowchart 1 on the right.

  - For each agent, each epoch \( t \) is either an action round or a broadcast round, \( t = 1 \) being an action round for every agent.

  - Each agent \( j \) also has private table containing, for each arm \( a_i \):
    - Number of times \( t \) has pulled arm \( a_i \).
    - Sum of rewards gained \( b \) from above pulls.
    - Number of pulls among the above that \( j \) has broadcast:\( b_{ji} \).
    - Sum of rewards from arm \( j \) corresponding to above broadcast pulls:\( b_{ji} \).

  - For each agent, each epoch \( t \) is either an action round or a broadcast round, \( t = 1 \) being an action round for every agent.

  - Each agent \( j \) also has private table containing, for each arm \( a_i \):
    - Number of times \( t \) has pulled arm \( a_i \).
    - Sum of rewards gained \( b \) from above pulls.
    - Number of pulls among the above that \( j \) has broadcast:\( b_{ji} \).
    - Sum of rewards from arm \( j \) corresponding to above broadcast pulls:\( b_{ji} \).