Reinforced Deep Learning for 5G Networks.

-Project Report by,
Sidhartha Kumar Paswan, 1901CS60
Shekhar Raj Suryavanshi, 1901CS54

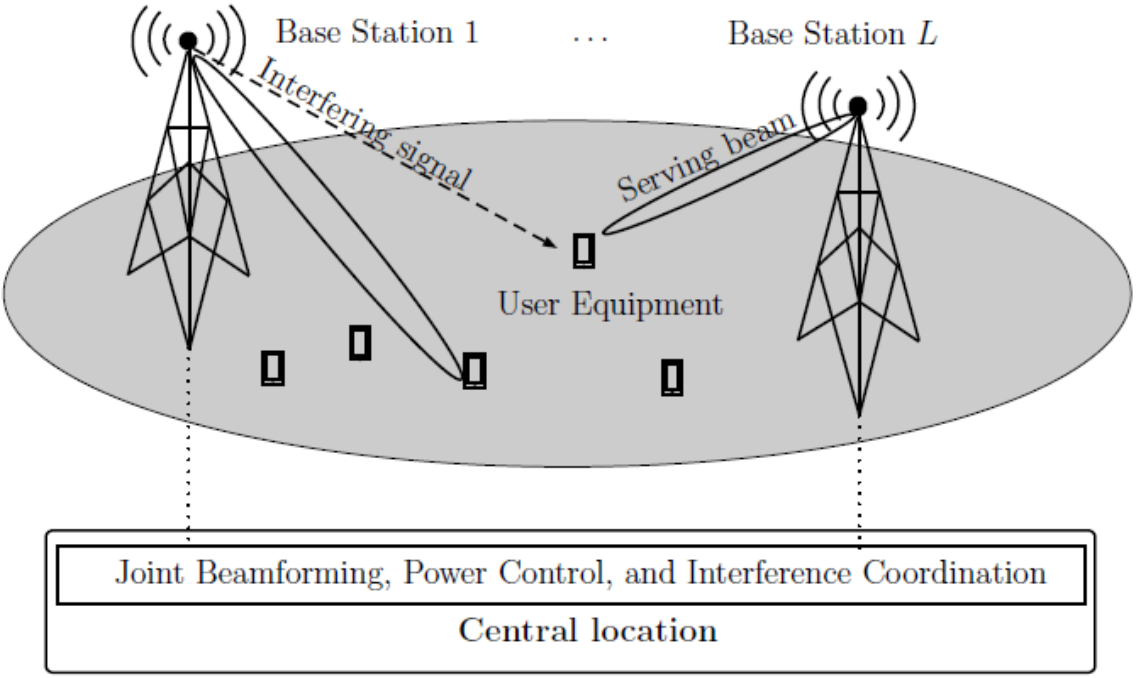
Introduction:

With the introduction of the fifth generation of wireless communication (5G), there is an obvious massive growth of traffic volume and so is the required increase in the data transfer rate and the challenge to decrease latency of our connection, nevertheless we expect better crystal clear voice quality as well as better reliability. To achieve all these, we propose an Algorithm based on Reinforced Deep Learning framework which will help us further to achieve maximum **Signal to Interference plus Noise Ratio (SINR)** which as the name suggest make sure we have a strong connection to serving Base Stations and minimal interference from other Base stations. **Power control** in voice bearers makes them more robust against wireless impairments, such as fading. It also enhances the usability of the network and increases the cellular capacity. For data bearers, **beamforming**, **power control**, and **interference coordination**, can improve the robustness of these data bearers, improve the data rates received by the end-users, and avoid retransmissions. We would use these above aforementioned terms to get the maximum achievable **SINR** value, which will ultimately make sure we are on par with the requirements we mentioned earlier.

Objective

We need to find an algorithm which can jointly solve the **beamforming, power control, and interference coordination actions.**

In contrast to other existing algorithms and methods we need a method where not only the **transmit power of the serving Base Station (BS) is controlled but also the transmit power of interfering BS is controlled**.



As shown in the above figure, we have a condition where a **UE**(User Equipment) is getting an interfering signal from another BS which is not currently serving the UE. This is the condition we need to consider in interference coordination for optimal or max SINR.

All these can be achieved using DRL.

Why DRL?

The reason why we choose deep reinforcement learning (DRL) is as follows:

- It will not require the knowledge of the channels in order to find the SINR-optimal beamforming vector. This is in contrast with the upper bound SINR performance, which finds the optimal beamforming vector by searching across all the beams in a codebook that maximizes the SINR (and this requires perfect knowledge of the channel). **Ultimately, saving us time and avoiding exhaustive search.**
- It **minimizes the involvement of the UE** in sending feedback to the BS by sending back **its received SINR along with its coordinates**, while the agent handles the power control and interference coordination commands to the involved BSs. The current industry standards require that the UE reports its

channel state information which is either a vector of length equal to the number of antenna elements or a matrix of dimension equal to the number of antenna elements in each direction. In our case, we achieve a reduction in the reporting overhead by using the UE coordinates instead. **Saves, UE battery and some CPU usage.**

- The current industry standards today only require the serving BS to send power control commands to the UE for the uplink direction. But we can have explicit **power control and interference coordination (PCIC**) commands sent by the UE to the serving and interfering BSs, using DRL's neural network.

Network Model

The major attributes of our network / environment will be:

- We consider **OFDM multi-access downlink cellular networks** of *N* number of BSs(*N*> 1).
- A BS transmits to one UE, in a downlink scenario.
- The BSs have **intersite distance of R.**
- The UE are randomly scattered in the service area of the BSs.
- The association between users and their serving BS is solely dependent on the distance in between them. **The closer BS is used as serving BS for a user equipment.**
- A user is served by **one BS** max at a time.
- The BS service area radius **r > R/2***(half the intersite distance),* to allow overlapping to get the interference condition.

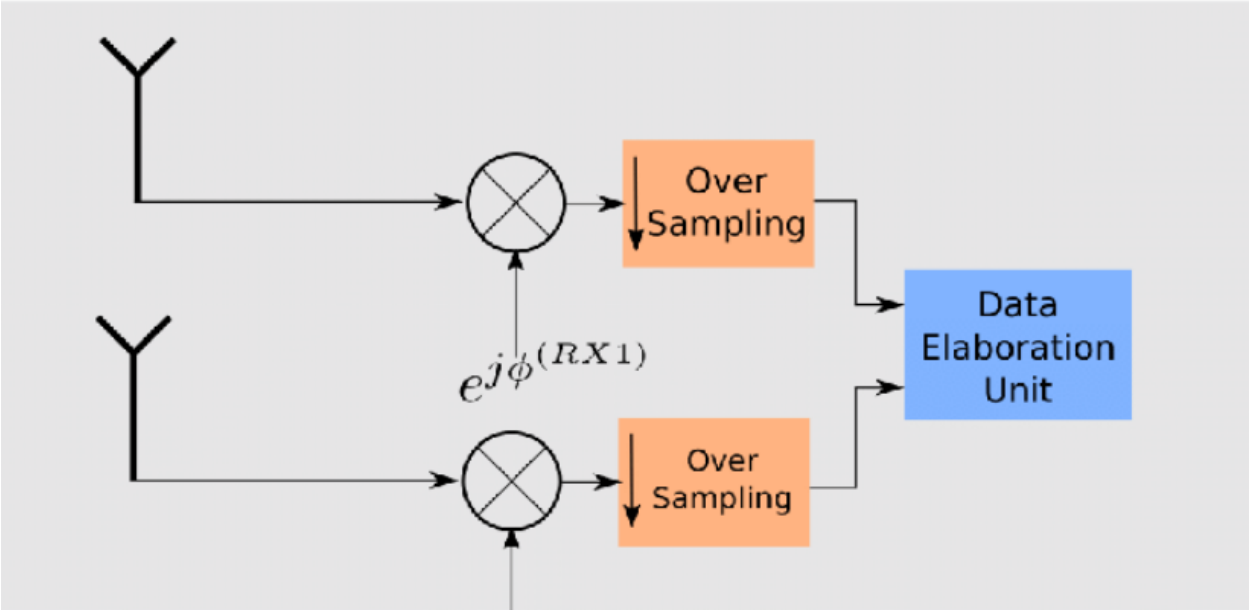
ULTIMATE CHALLENGE

Our ultimate challenge is to come up with an algorithm that can jointly optimize beamforming vectors , Power Control, Interference Coordination.

There are some existing algorithms that can fulfil our requirement but not completely.

EXISTING ALGORITHMS

1) FPA - FPA or Fixed Power Allocation is currently in use in many companies. The baseline of this algorithm is to provide and transmit the power signal at specific value. Transmitted signal is simply divided into all PRBs equally. Since, there is no implementation of Interference Coordination in this algorithm. So this is a major drawback of this algorithm.



2) **Tabular RL method -** We use a tabular setting of Q-learning (or “vanilla” Q-learning) to implement the algorithm for voice communication. During a tabular setting, the state-action value function Qπ(st, at) is represented by a table Q ∈ R|S|×|A|. there's no neural network involvement and therefore the Q-learning update is defined as:

$$Q_\pi(s_t, a_t) := (1-\alpha)Q_\pi(s_t, a_t) + \alpha\left(r_{s,s',a} + \gamma \max_{a'} Q_\pi(s', a')\right)$$

where Qπ(st, at) := [Q]st,at . Here, α > 0 is the learning rate of the Q-learning update and defines how aggressive the experience update is with reference to the prior experience. Computationally, the tabular setting suits problems with small state spaces, and maintaining a table Q is feasible.

Proposed Algorithm

We have proposed our algorithm that can jointly do these operations.

Working Of Algorithm

1. The algorithm will run for a large time interval T. At any time t, it will select an action based on policy. There are two types of policy in that algorithm. First one is **Exploration** and the second one is **Exploitation**.
2. Based on the two policies it will choose the action according to a suitable policy that will be a joint combination of Beamforming, Interference Coordination and Power Control.
3. After performing that action it will interact with the environment and based on that interaction,The Environment will give some feedback as reward.
4. Based on the reward it will assess the impact on effective SINR γℓeff[t].
5. Also based on that reward it trains the remaining DQN.
6. It repeats the above steps again and again.

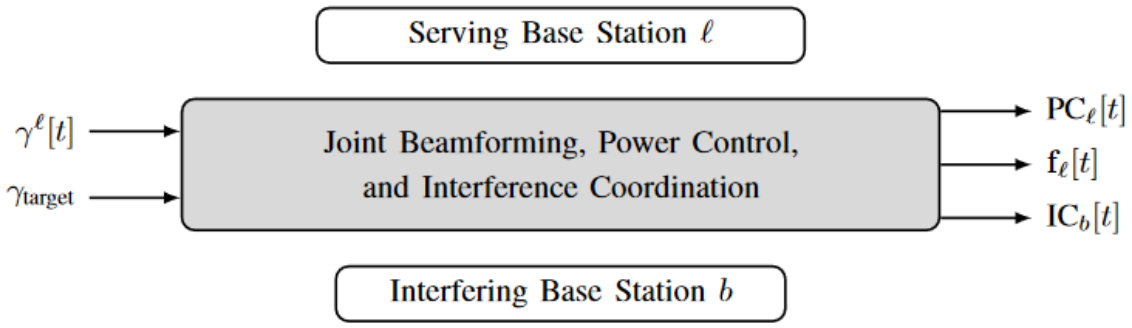


Fig. shows an overview working of proposed algorithm

Here is pseudo code for the algorithm

Input: The downlink received SINR measured by the UEs.

Output: Sequence of beamforming, power control, and interference coordination commands to solve (6).

Initialize time, states, actions, and replay buffer $\mathcal{D}$.
repeat
  repeat
    $t := t + 1$
    Observe current state s_t.
    $\epsilon := \max(\epsilon \cdot d, \epsilon_{\text{min}})$
    Sample $r \sim \text{Uniform}(0, 1)$
    if $r \leq \epsilon$ **then**
      Select an action $a_t \in \mathcal{A}$ at random.
    else
      Select an action $a_t = \arg\max_{a'} Q_\pi(s_t, a'; \boldsymbol{\theta}_t)$.
    end
    Compute $\gamma^{\ell}_{\text{eff}}[t]$ and $r_{s,s',a}[t; q]$ from (17).
    if $\gamma^{\ell}_{\text{eff}}[t] < \gamma_{\text{min}}$ **then**
      $r_{s,s',a}[t; q] := r_{\text{min}}$
      Abort episode.
    end
    Observe next state s'.
    Store experience $e[t] \triangleq (s_t, a_t, r_{s,s',a}, s')$ in $\mathcal{D}$.
    Minibatch sample from $\mathcal{D}$ for experience $e_j \triangleq (s_j, a_j, r_j, s_{j+1})$.
    Set $y_j := r_j + \gamma \max_{a'} Q_\pi(s_{j+1}, a'; \boldsymbol{\theta}_t)$
    Perform SGD on $(y_j - Q_\pi(s_j, a_j; \boldsymbol{\theta}_t))^2$ to find $\boldsymbol{\theta}^\star$
    Update $\boldsymbol{\theta}_t := \boldsymbol{\theta}^\star$ in the DQN and record loss L_t
    $s_t := s'$
  until $t \geq T$
until convergence or aborted
if $\gamma^{\ell}_{\text{eff}}[t] \geq \gamma_{\text{target}}$ **then** $r_{s,s',a}[t; q] := r_{s,s',a}[t; q] + r_{\text{max}}$

Reference-
https://www.researchgate.net/publication/334161397_Deep_Reinforcement_Learning_for_5G_Networks_Joint_Beamforming_Power_Control_and_Interference_Coordination

HOW TO MEASURE THE PERFORMANCE OF ALGORITHM?

Now, it is very important to measure the performance of algorithms. To measure the performance of any algorithm we generally looks for these following four terms-

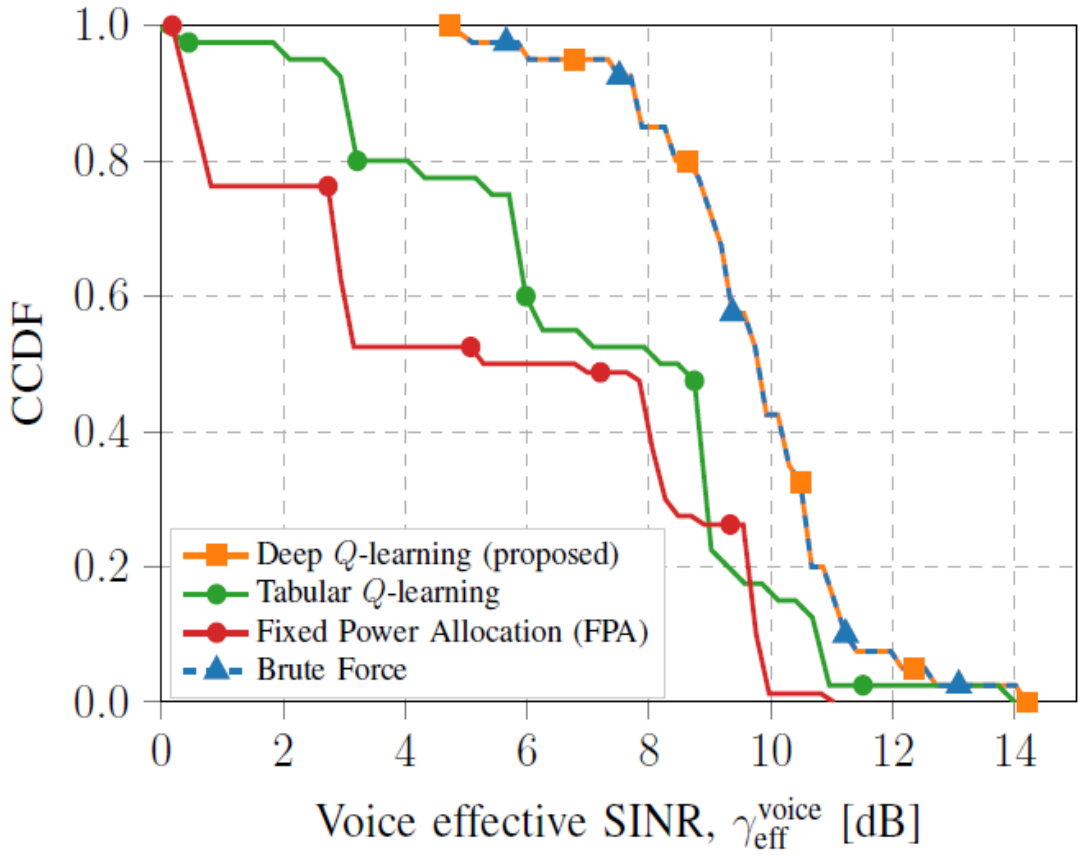
- **Convergence -** We define convergence ζ in terms of the episode at which the target SINR is fulfilled over the entire duration of T for all UEs within the network. We expect that because the number of antennas in the ULA M increases, the convergence time ζ also will increase. In voice, convergence as a function of M isn't applicable, since we only use single antennas. For several random seeds, we take the aggregated percentile convergence episode.
- **Run Time -** While calculating the boundary of the brute force algorithm run-time complexity is feasible, obtaining an identical expression for the proposed deep Q-learning algorithm could also be challenging due to lack of convergence and stability guarantees . Therefore, we obtain the run time from simulation per antenna size M.
- **Coverage -** We build a complement cumulative distribution function (CCDF) of γℓeff following and by running the simulation many times and changing the random seed, effectively changing the way the users are dropped in the network
- **Sum Rate Capacity -** Using the effective SINRs we calculate the average sum rate capacity by using below formula-

$$C = \frac{1}{T} \sum_{t=1}^{T} \sum_{j \in \{\ell, b\}} \log_2(1 + \gamma_{\text{eff}}^{j}[t])$$

Based on these factors and terms we can measures the perfomanced of any algorithms and two algorithms can be compared.

Results

Complementary Cumulative Distribution Function VS Voice Effective SINR

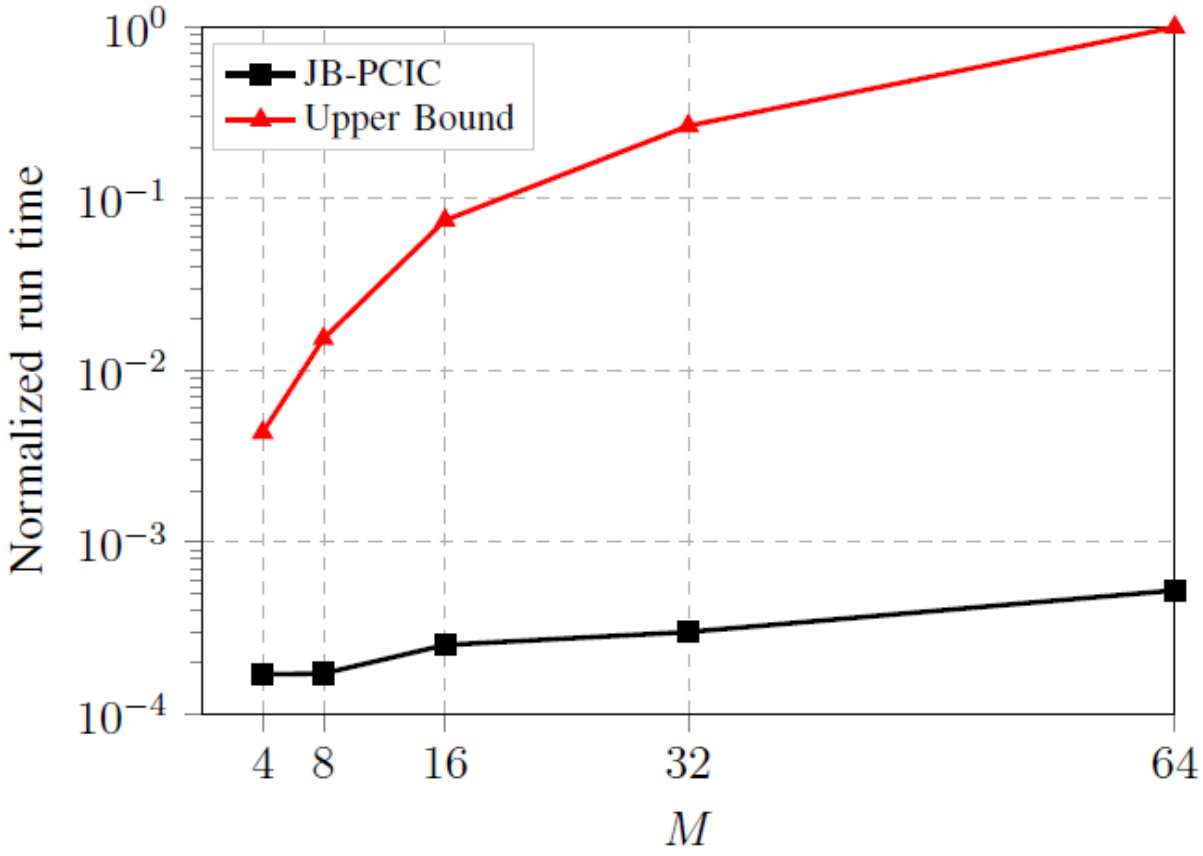


We can infer from the above graph that the

- FPA algorithm has the worst performance especially at low effective SINR regime since FPA has no power control or interference coordination.
- The tabular implementation of our proposed algorithm has better performance compared with the FPA, since power control and interference coordination are introduced to the base stations, though not as effectively, which explains why close to γeff = 9 dB tabular Q-learning PCIC underperforms FPA.
- Further, we observe that deep Q-learning outperforms since deep Q-learning has resulted in a higher reward compared to tabular Q-learning, because deep Q-learning has converged at a better solution unlike the tabular Q-learning the convergence of which may have been impeded by the choice of an initialization of the state-action value function.

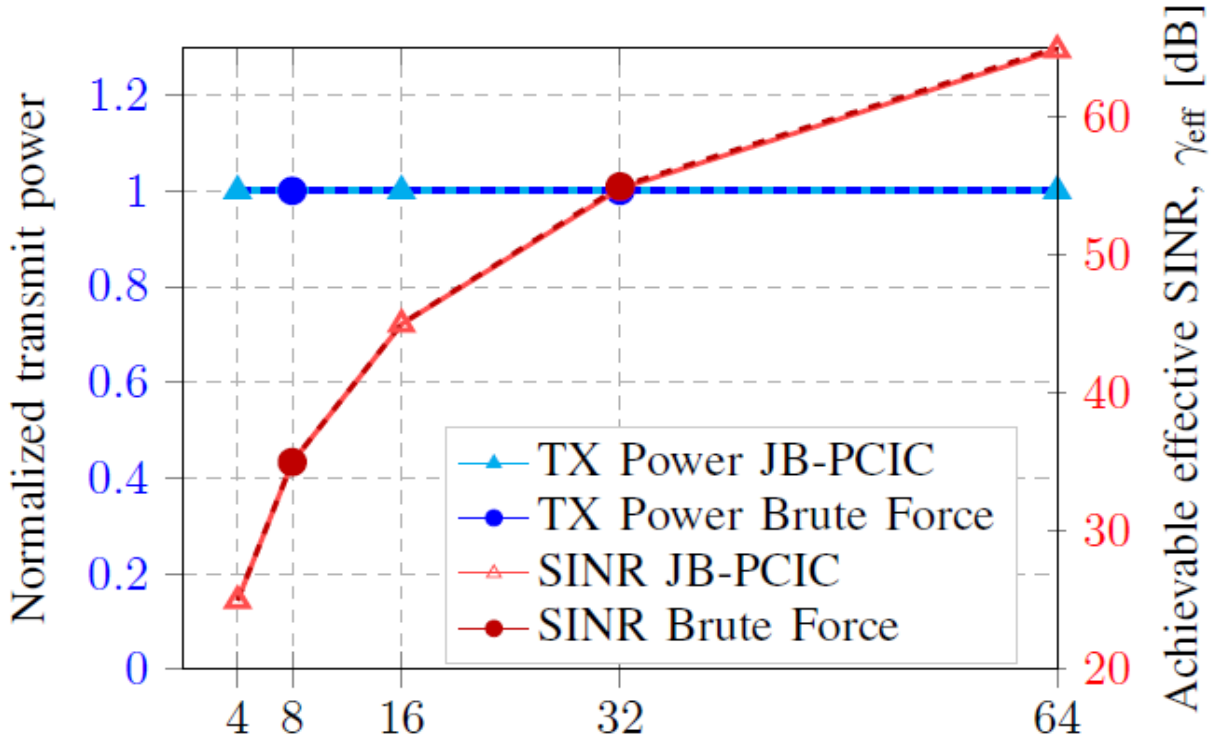
- However, as the effective SINR γeff approaches 13 dB, the users are close to the BS center and therefore all power control algorithms perform almost similarly thereafter.

Normalised Run time as a function of number of antennas



As we can see clearly, brute force algorithm has a significantly larger run time compared to the proposed algorithm due to the exponential nature of the run-time complexity. **Hence our proposed algorithm promises better latency on our connection.**

Achievable SINR and normalized transmit power for both the brute force and proposed JB-PCIC algorithms as a function of the number of antennas M.



- The achieved SINR is proportional to the ULA antenna size M
- The transmit power is almost equal to the maximum.
- the relative performance of JB-PCIC compared with the brute force performance, we observe that the performance gap of both the transmit power of the base stations and the SINR is almost diminished all across M.
- This is because of the DQN ability to estimate the function that leads to the upper limit of the performance. Further, we observe that the solution for the race condition is for both BSs to transmit at maximum power.

Conclusion

We sought to maximize the downlink SINR in a multi-access OFDM cellular network from a multi-antenna base station to single-antenna user equipment, and we developed a joint beamforming, power control, and interference coordination algorithm (JB-PCIC) using deep reinforcement learning. This algorithm resides at a central location and receives UE measurements over the backhaul. For voice bearers, the proposed algorithm outperformed both the tabular Q-learning algorithm and the industry standard fixed power allocation algorithm. Moreover, the overall amount of feedback from the UE is reduced because the UE sends its coordinates and would not need to send explicit commands for beamforming vector changes, power control, or interference coordination.