Sum Rate Maximization for RIS Aided I2V Communication

Ayush Madhan Sohini (2019156)
Tathagat Pal (2019211)

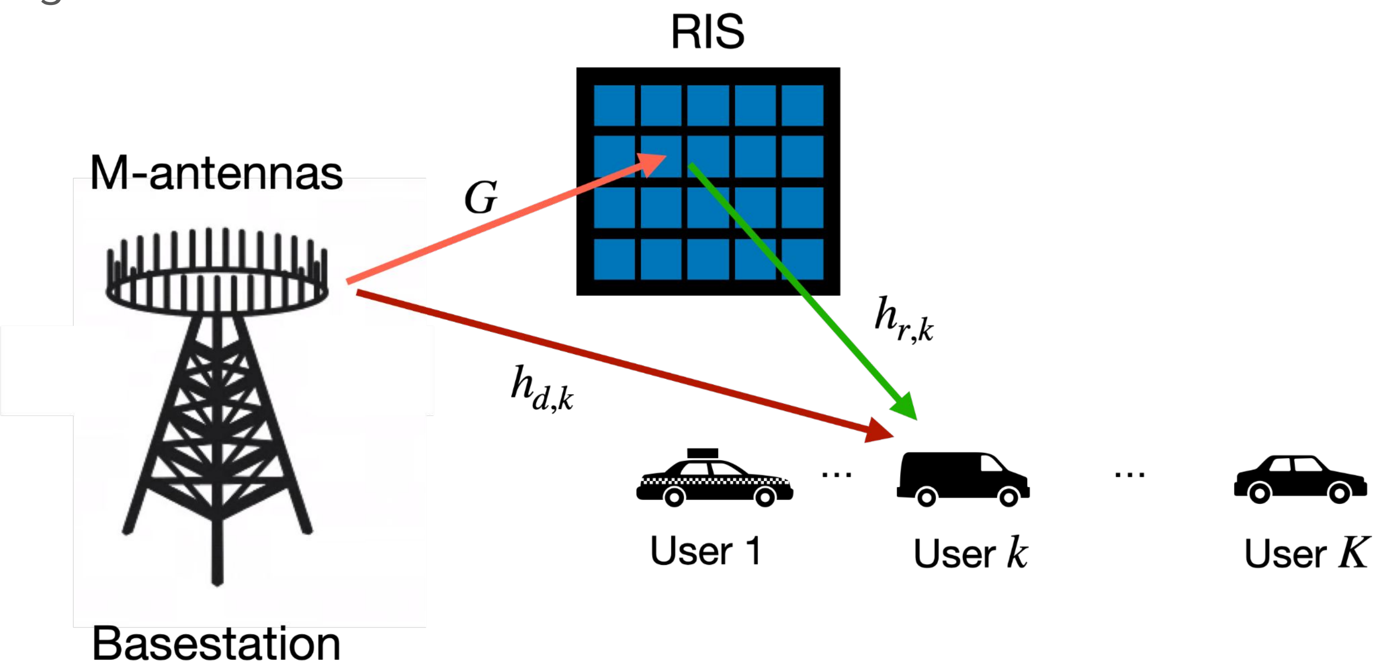
INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY **DELHI**

Introduction

- As we are approaching technology beyond 5G, users are demanding higher throughput, lower latency etc.
- B5G and 6G has caught attention of IoV (Internet of Vehicles), which includes I2V, V2V, V2X communication.
- I2V suffers from attenuation and path losses. We aim to improve the quality of service (estimating throughput here), from the base station (infrastructure) to the vehicles.
- Reconfigurable intelligent surface (RIS) is a programmable structure that can be used to control the propagation of electromagnetic waves by changing the electric and magnetic properties of the surface.

System Model

- 1 basestation (infrastructure) equipped with M antennas.
- Reconfigurable intelligent surfaces with N reflection elements.
- K single-antenna vehicles.



Evaluation Metrics: SINR and Sum Rate

SINR at the k^{th} vehicle,

$$\gamma_k = \frac{|(h_{d,k}^H + \theta^H H_{r,k})w_k|^2}{\sum_{i=1,i\neq k}^{K} |(h_{d,k}^H + \theta^H H_{r,k})w_i|^2 + \sigma_o^2},$$

where $h_{d,k}^H$ is the LoS channel gain between BS and k^{th} vehicle,

θ is the phase-shift matrix,

$$H_{r,k} = diag(h_{r,k}^H)G \in C^{N\times M},$$

w_k is the beamforming vector at k^{th} vehicle, $\sum_{k=1}^{K} \|w_k\|^2 \leq P_T$

σ_o^2 is the channel noise.

Sum rate is given as $log(1 + \gamma_k)$

Main Challenges

- The Sum Rate of a system, when we consider a multi user interference channel, is a **Non Convex** function.
- The challenge arises when we want to optimize this function as non convex functions have multiple local optimal values.
- Our work in this project is to analyse a method proposed by Guo et al., where this complex function is broken down into several sub optimal near convex parts, and they're collectively optimized.
- We will further analyze how we can use a deep learning approach to solve this problem. For this, we'll implement a custom DNN (unsupervised learning), with the loss function set to either the SINR or the sum-rate, and give near optimal beamforming vectors and phase shifts.

Optimization Techniques to be Explored

- Fractional Programming (to break the fractional problem inside a logarithmic function into simple convex parts)
    - Lagrangian Dual Transform
    - Quadratic Transform
- Unsupervised Learning.
- Study the tradeoffs between the analytical model and unsupervised learning.

Timeline

- Literature Review and understanding the existing work to be finished by midsem.
- One week after midsem: Implement an analytical model to maximize the sum rate/SINR.
- Implementation of the DNN after midsem.
- Hyperparameter Tuning alongside DNN modelling.
- Report writing before endsem.

Objective Function

SINR at the k^{th} vehicle,

$$\gamma_k = \frac{|(h_{d,k}^H + \theta^H H_{r,k})w_k|^2}{\sum_{i=1,i\neq k}^{K} |(h_{d,k}^H + \theta^H H_{r,k})w_i|^2 + \sigma_o^2},$$

where $h_{d,k}^H$ is the LoS channel gain between BS and k^{th} vehicle,

θ is the phase-shift matrix,

$$H_{r,k} = diag(h_{r,k}^H)G \in C^{N\times M},$$

w_k is the beamforming vector at k^{th} vehicle, $\sum^{K} \|w_k\|^2 \leq P_T$

$$\boxed{maximize \quad f(\boldsymbol{\theta}, \boldsymbol{w}) = \sum_{i=1}^{k} \log_2(1 + \gamma_k)}$$

Closed-Form Fractional Programming Approach

$$\textit{maximize} \quad f(\boldsymbol{\theta}, \mathbf{w}) = \sum_{i=1}^{k} \log_2(1 + \gamma_k)$$ is the equivalent sum-of-logarithms-of-ratio problem:

$$\max_x \sum_{k=1}^{K} \log\left(1 + \frac{|A_k(\mathrm{x})|}{B_k(\mathrm{x}) - |A_k(\mathrm{x})|^2}\right)$$

- This is not a convex function, so it is hard to find a global optimal solution.
- We use Lagrangian Dual Transform and Quadratic Transformation to obtain the following optimization problem:

$$\max_{\mathbf{x}, \boldsymbol{\alpha}} \sum_{k=1}^{K} \left(\log(1+\alpha_k) - \alpha_k + (1+\alpha_k)\frac{|A_k(\mathbf{x})|^2}{B_k(\mathbf{x})}\right)$$

$$\textbf{s.t. } \alpha_k \geq 0, \quad \forall k = 1, \cdots, K,$$

- Using Quadratic Transform

$$\max_x \sum_{k=1}^{K} \left(\frac{|A_k(\mathrm{x})|^2}{B_k(\mathrm{x})}\right)$$ is equivalent to $$\max_{\boldsymbol{x}, \beta} \sum_{k=1}^{K} \left(2Re\{\beta_k^* A_k(\mathrm{x}) - |\beta_k|^2 B_k(\mathrm{x})\}\right)$$

Auxiliary Variables Introduced

$$\alpha_k = \frac{\bar{\zeta}_k^2 + \bar{\zeta}_k\sqrt{\bar{\zeta}_k^2 + 4}}{2},$$

$$\beta_k = \frac{\sqrt{\omega_k(1+\bar{\alpha}_k)}(\mathbf{h}_{\mathrm{d},k}^{\mathrm{H}} + \bar{\boldsymbol{\theta}}^{\mathrm{H}}\mathbf{H}_{\mathrm{r},k})\bar{\mathbf{w}}_k}{\sum_{i=1}^{K}\left|(\mathbf{h}_{\mathrm{d},k}^{\mathrm{H}} + \bar{\boldsymbol{\theta}}^{\mathrm{H}}\mathbf{H}_{\mathrm{r},k})\bar{\mathbf{w}}_i\right|^2 + \sigma_0^2},$$

where $\bar{\zeta}_k = \frac{1}{\sqrt{\omega_k}}\mathrm{Re}\left\{\bar{\beta}_k^* \, \bar{\mathbf{h}}_k^{\mathrm{H}}\bar{\mathbf{w}}_k\right\}$ and $\bar{\mathbf{h}}_k$ the combined channel:

$$\bar{\mathbf{h}}_k = \mathbf{h}_{\mathrm{d},k} + \mathbf{H}_{\mathrm{r},k}^{\mathrm{H}}\bar{\boldsymbol{\theta}}.$$

Prox Linear Update for Beamforming Vector

$$\boldsymbol{W} = \arg\min_{\boldsymbol{W}} \sum_{k=1}^{K} \left(\mathrm{Re}\left\{ \mathbf{g}_k^{\mathrm{H}} (\mathbf{w}_k - \hat{\mathbf{w}}_k) \right\} + \frac{L}{2} \|\mathbf{w}_k - \hat{\mathbf{w}}_k\|^2 \right)$$
$$\textbf{s.t.} \ \sum \|\mathbf{w}_k\|^2 \leq P_{\mathrm{T}},$$

where $L > 0$, the gradient is denoted by

$$\mathbf{g}_k = -\left. \frac{\partial f_{\mathrm{A2}}}{\partial \mathbf{w}_k} \right|_{\mathbf{w}_k = \hat{\mathbf{w}}_k}$$
$$= -2\sqrt{\omega_k(1+\bar{\alpha}_k)}\bar{\beta}_k \bar{\mathbf{h}}_k + 2\sum_{i=1}^{K} |\bar{\beta}_i|^2 \bar{\mathbf{h}}_i \bar{\mathbf{h}}_i^{\mathrm{H}} \hat{\mathbf{w}}_k,$$

$$L = 2\| \sum_{i=1}^{k} |\beta_i|^2 \bar{\boldsymbol{h}}_i \bar{\boldsymbol{h}}_i^H \|_F$$

Successive Convex Optimization for θ

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \; f(\boldsymbol{\theta}) \triangleq \boldsymbol{\theta}^{\mathrm{H}}\boldsymbol{U}\boldsymbol{\theta} - 2\mathrm{Re}\left\{\boldsymbol{\theta}^{\mathrm{H}}\boldsymbol{\nu}\right\}$$
$$\textbf{s.t. } |\theta_n| = 1, \quad \forall n = 1, \cdots, N.$$

where $\boldsymbol{U}$ and $\boldsymbol{\nu}$ are

$$\boldsymbol{U} = \sum_{k=1}^{K} \left|\bar{\beta}_k\right|^2 \sum_{i=1}^{K} \bar{\mathbf{a}}_{i,k}\bar{\mathbf{a}}_{i,k}^{\mathrm{H}},$$
$$\boldsymbol{\nu} = \sum_{k=1}^{K} \left(\sqrt{\omega_k(1+\bar{\alpha}_k)}\bar{\beta}_k^* \, \bar{\mathbf{a}}_{k,k} - |\beta_k|^2 \sum_{i=1}^{K} \bar{b}_{i,k}^* \bar{\mathbf{a}}_{i,k} \right),$$

with $\bar{\mathbf{a}}_{i,k} = \mathbf{H}_{\mathrm{r},k}\bar{\mathbf{w}}_i$, and $\bar{b}_{i,k} = \mathbf{h}_{\mathrm{d},k}^{\mathrm{H}}\bar{\mathbf{w}}_i$. We further replace θ_n by φ_n, where $\theta_n = e^{\jmath\varphi_n}$ and $\varphi_n \in \mathbb{R}$, and then the update rule is recast to

$$\boldsymbol{\varphi} = \arg\min_{\boldsymbol{\varphi}\in\mathbb{R}^N} \; f(\boldsymbol{\varphi}) \triangleq (e^{\jmath\boldsymbol{\varphi}})^{\mathrm{H}}\boldsymbol{U}e^{\jmath\boldsymbol{\varphi}} - 2\mathrm{Re}\left\{\boldsymbol{\nu}^{\mathrm{H}}e^{\jmath\boldsymbol{\varphi}}\right\}$$

where $\boldsymbol{\varphi} = [\varphi_1, \cdots, \varphi_N]^{\mathrm{T}}$.

Then update rule is recast to:

$$arg\,min_{\psi\in\mathbb{R}^N} f^*(\psi) \triangleq (e^{j\psi})^H U e^{j\psi} - 2Re\{v^H e^{j\psi}\}.$$

$$\psi = \psi - \frac{\nabla f^*(\psi)}{\delta}.$$

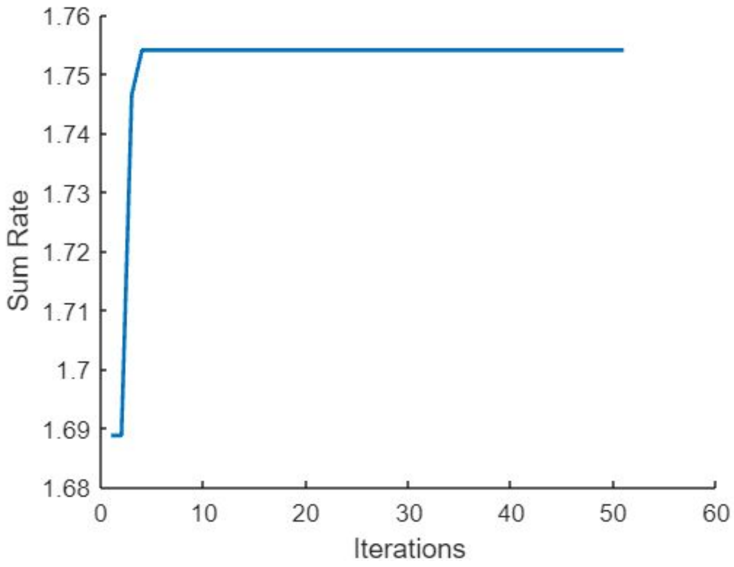
$$\nabla f^*(\psi) = 2Re\{-je^{-j\psi} \circ (Ue^{j\psi} - v)\}.$$

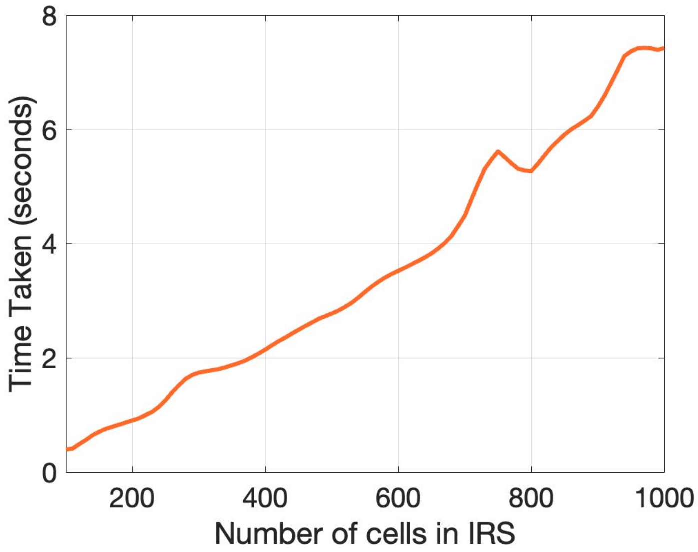
Using Armijo rule, δ to determine step size:

$$\nabla f^*(\bar{\psi}) - \nabla f^*(\psi) \geq \zeta_\delta \|\nabla f^*(\bar{\psi})\|^2$$

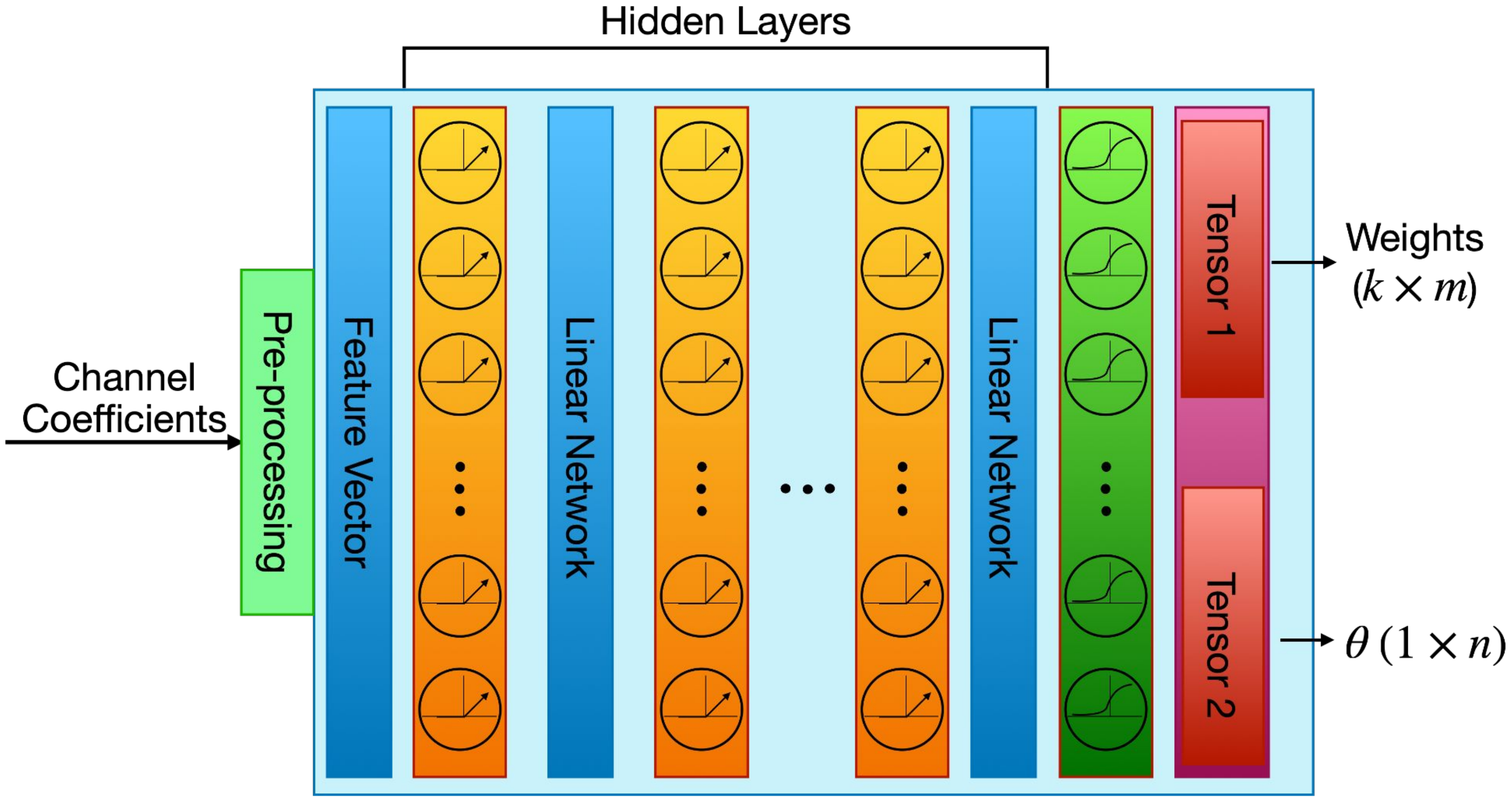
Results

- We averaged our results for 2000 Monte Carlo simulations wherein each simulation generated a random channel and randomly distributed vehicles around the basestation with Tx=30, N=100.
- We also check how the computational time of the algorithm behaves as the number of cells in the RIS changes.





DNN Approach



DNN-Unsupervised Learning

Feature Design:

- Preprocess all the channel coefficients and design a suitable feature vector *f*.
- We exploit an inherent product structure of channel coefficients corresponding to each reflecting element and transmit antenna.
- Given Loss is a concave function, we obtain a suboptimal solution.
- We concatenate vector h_{dk} ★ h_{rk} into one vector with real and imaginary components of complex numbers

Loss function:

- We exploit the closed-form equation for throughput i.e., to define a loss function *L*.
- The loss function is given as:

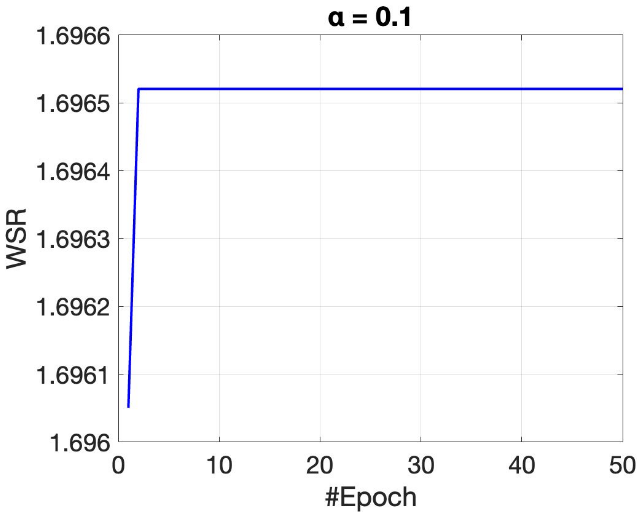
$$L = -\frac{1}{k^2}\sum_{k=1}^{K} w_k \mathrm{log}(1+\gamma_k)$$

DNN

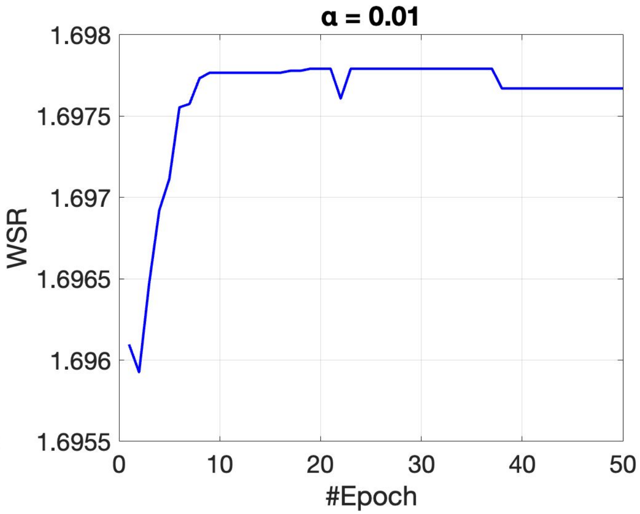
Neural network architecture and training:

- [Input] ⇒[1028, 640, 320] ⇒ [Weights and θ]
- Optimizer: Adam
- Learning rate (best performance): 0.01
- Each hidden layer consists of ReLU
- Output layer consists of a sigmoid activation function. Sigmoid activation function inherently satisfies the unit magnitude constraints of phase shifts.

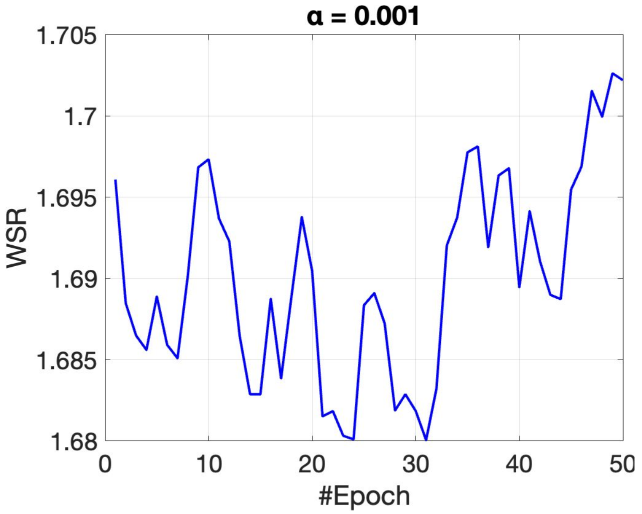
Results (2000 MC Simulations)



α = 0.1

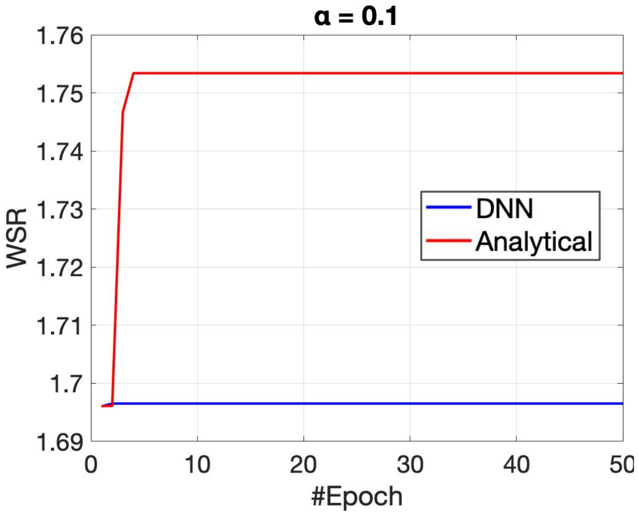


α = 0.01

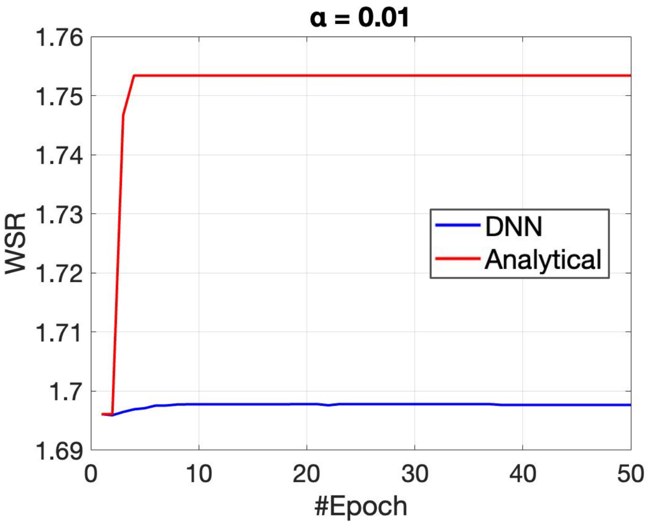


α = 0.001

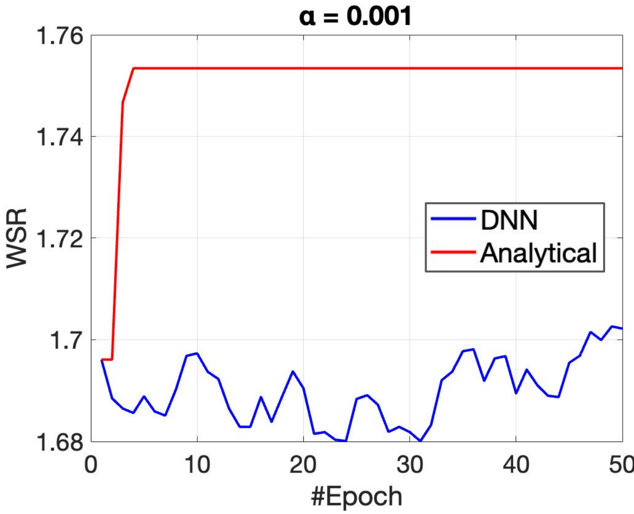
Results: Suboptimality of the DNN Model



α = 0.1

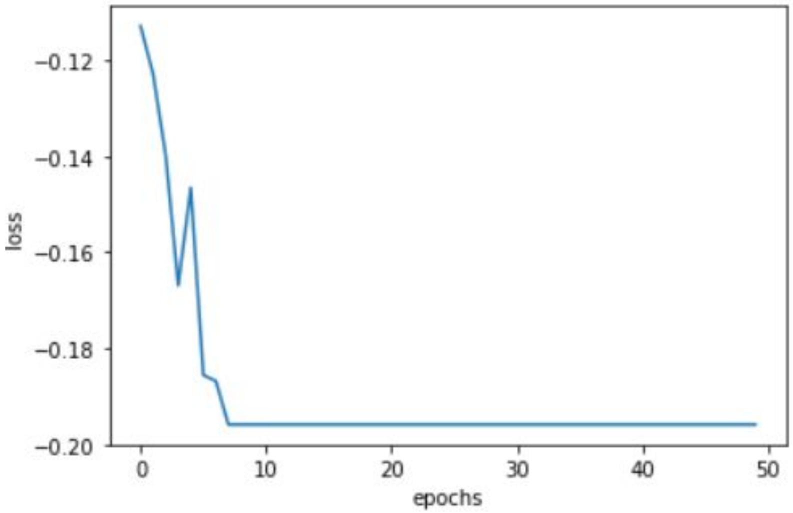


α = 0.01



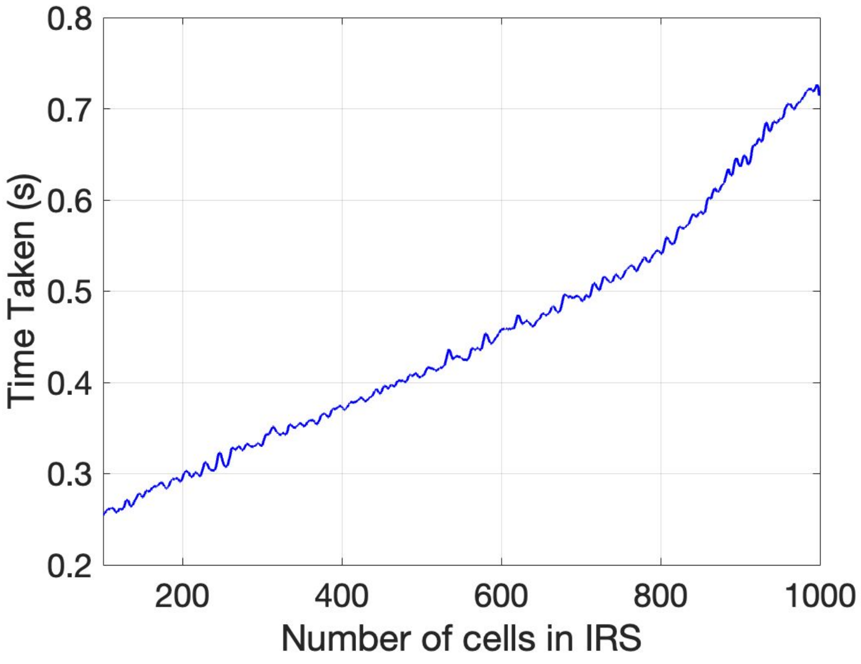
α = 0.001

DNN: Loss Analysis

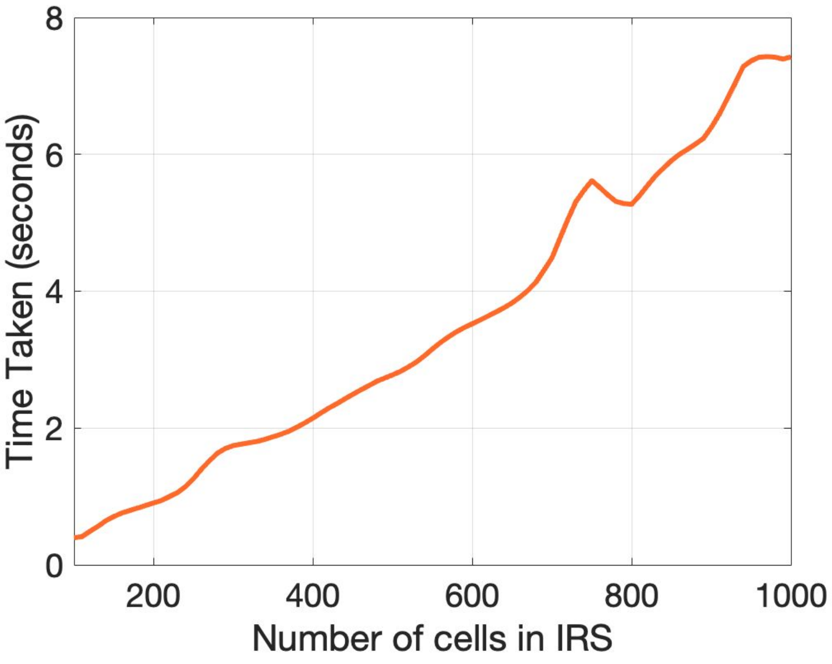


Time Complexity Analysis

DNN



Analytical



Analysis

- Since the objective is concave minimization the optimizer tends to find the nearest local optimal solution in every iteration.
- Gain in throughput minimal as compared to analytical solution.
- However time taken to find the optimum solution using DNN is marginally lesser as compared to analytical solution.
- More model tuning is required.

Timeline

- **Literature Review and understanding the existing work to be finished by midsem (✓)**
- **One week after midsem: Implement an analytical model to maximize the sum rate/SINR (✓)**
- **Implementation of the DNN after midsem (✓)**
- **Hyperparameter Tuning alongside DNN modelling (✓)**
- **Report writing before endsem (✓)**