Power Allocation in Energy Sharing Environment

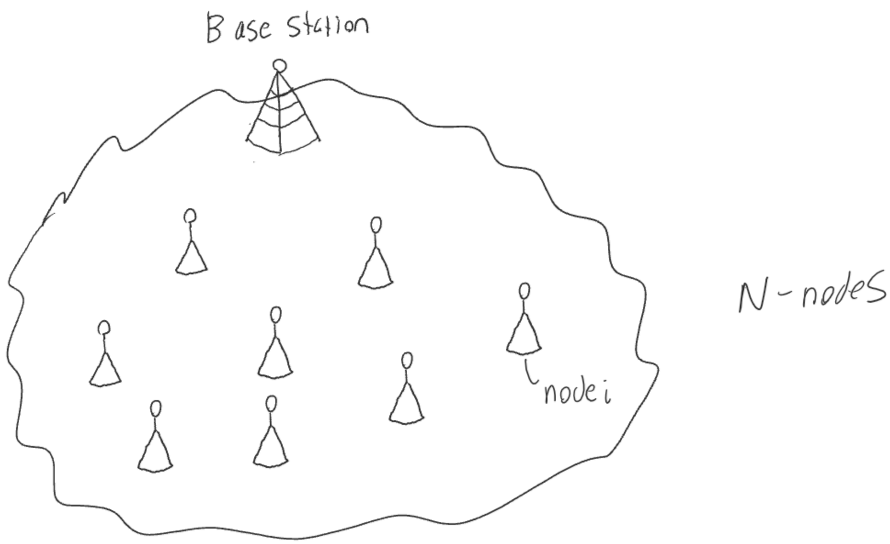
Chris Manna

EE550

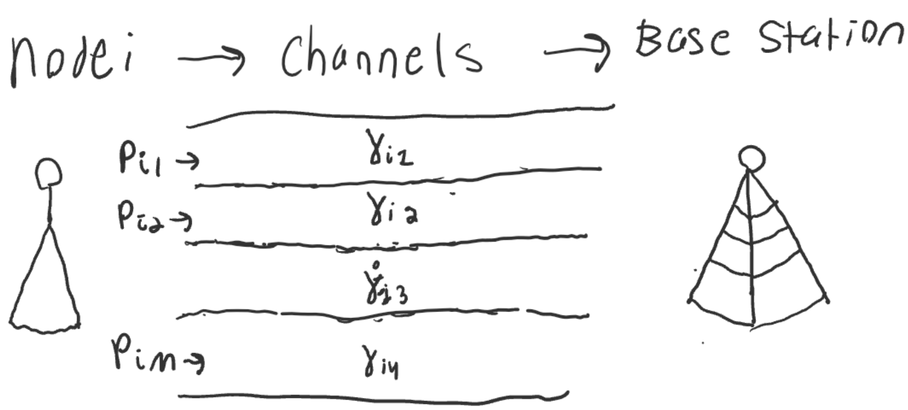
I. Motivation

My motivation for this project stemmed from my passion for sustainability along with my fascination with the waterfilling solution. With the rise of renewable energies as a primary power source, energy sharing technologies have gained prominence to address power limitations. My objective was to explore whether nodes trying to maximize throughput incorporating energy-sharing functionalities would produce a solution similar to the traditional waterfilling approach.

II. Problem Setup

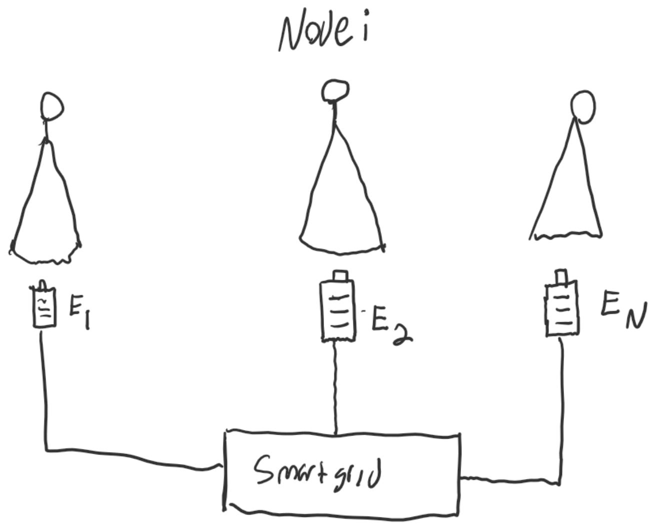


Let us have a sensor network as seen above. There are N nodes that communicate with a sink or base station. All nodes can send to the base station in a single hop and there is no interference in transmission between the nodes. The goal is to maximize the total throughput received by the base station.



Each node can communicate with the Base station through M channels. All channels for every node are independent resulting in N*M independent channels in the network. The throughput across one channel can thus be modeled as:

$$Throughput_{ij} = \log(1 + \gamma_{ij} p_{ij})$$



In addition, the Nodes have energy sharing capabilities. Let each Node i, have the power E_i available to transmit across its channels as well as transfer to other nodes. Let e_{ik} be the power that node i transmits to node k. Assume that energy transferred from one node to another can be used instantaneously for transmission.

During energy transfer, there is a certain degree of loss, denoted by the symbol α.

$$0 \leq \alpha \leq 1$$

Thus the power available at each node can be modeled as its current power, plus the energy it receives times the loss coefficient minus the energy it transfers.

$$\sum_{j=1}^{M} p_{ij} \leq E_i + \alpha \sum_{k=1}^{M} e_{ki} - \sum_{k=1}^{M} e_{ik}, \forall i \in \{1, 2, ... N\}$$

III. Problem Formulation

An optimization problem can then be created maximizing the total throughput across the network, subject to the power constraint for each node.

$$\max: \sum_{i=1}^{N} \sum_{j=1}^{M} \log(1 + \gamma_{ij} p_{ij})$$

s.t.

$$\text{C1: } \sum_{j=1}^{M} p_{ij} \leq E_i + \alpha \sum_{k=1}^{M} e_{ki} - \sum_{k=1}^{M} e_{ik}, \forall i \in \{1, 2, ... N\}$$

$$\text{C2: } p_{ij} \geq 0, e_{ik} \geq 0, e_{ii} = 0 \; \forall i \in \{1, 2, ... N\}, \forall k \in \{1, 2, ... N\}, \forall j \in \{1, 2, ... M\}$$

The problem can be transformed into a convex optimization problem by minimizing the negative sum of logarithms.

$$\min: -\sum_{i=1}^{N} \sum_{j=1}^{M} \log(1 + \gamma_{ij} p_{ij})$$

s.t.

$$\text{C1: } \sum_{j=1}^{M} p_{ij} \leq E_i + \alpha \sum_{k=1}^{M} e_{ki} - \sum_{k=1}^{M} e_{ik}, \forall i \in \{1, 2, ... N\}$$

$$\text{C2: } p_{ij} \geq 0, e_{ik} \geq 0, e_{ii} = 0, \forall i \in \{1, 2, ... N\}, \forall k \in \{1, 2, ... N\}, \forall j \in \{1, 2, ... M\}$$

IV. General Solving Utilizing Drift Plus Penalty

First to understand the network, a drift plus penalty algorithm was created from the optimization problem above. A virtual Queue, $Q_i(t)$, was created for each node's power constraint.

$$Q_i(t+1) = \max\left[Q_i(t) + \sum_{j=1}^{M} p_{ij}(t) - E_i - \alpha \sum_{k=1}^{M} e_{ki}(t) + \sum_{k=1}^{M} e_{ik}(t),\ 0\right],\ \forall i \in \{1, 2, \dots N\}$$

$$\text{Where } Q_i(0) = 0,\ \forall i \in \{1, 2, \dots N\}$$

The following convex optimization problem is then solved at each time slot. To create a compact set, constraints for maximum power per channel, P_{max} and maximum energy transferred from one node to another e_{max} were created.

$$\min:\ -V * \sum_{i=1}^{N}\sum_{j=1}^{M} \log\left(1 + \gamma_{ij} p_{ij}(t)\right) + \sum_{i=1}^{N} Q_i(t)\left(\sum_{j=1}^{M} p_{ij}(t) - \alpha \sum_{k=1}^{M} e_{ki}(t) + \sum_{k=1}^{M} e_{ik}(t)\right)$$

s.t.

$$p_{ij}(t) \in [0, P_{max}],\ \forall i \in \{1, 2, \dots N\},\ \forall j \in \{1, 2, \dots M\}$$

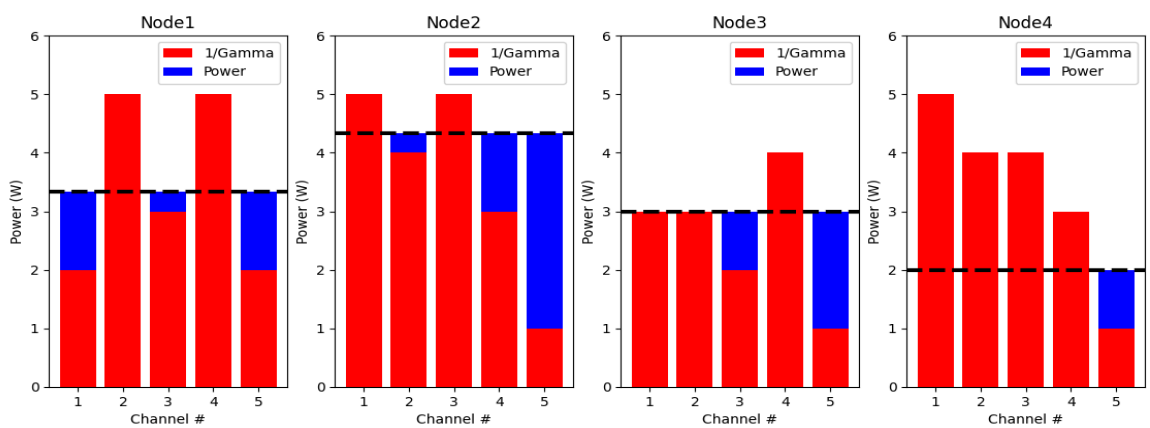
$$e_{ik}(t) \in [0, e_{max}],\ e_{ii}(t) = 0,\ \forall i \in \{1, 2, \dots N\}, \forall k \in \{1, 2, \dots N\},$$

The time average power was calculated for each channel per node.

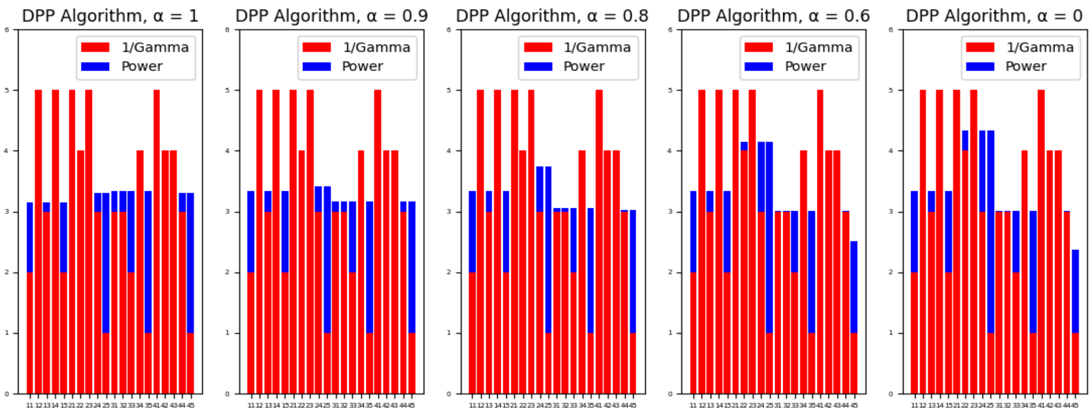
A simulation testing the DPP algorithm was then run with 4 nodes. Each node has 5 channels. The channel gain, gamma, for each channel was randomly chosen. Each node had the corresponding power:

$$E_1 = 3, E_2 = 5, E_3 = 3, E_4 = 1$$

Below shows the power allocation algorithm according to the waterfilling solution if no energy sharing between the nodes occurs.



The DPP algorithm was then run for various alpha values. The Power for all channels were plotted on one graph. Where the xlabel 12 , corresponds to node 1 channel 2.



When α = 0, the solution reverts to the independent isolated waterfilling solution for each node. When comparing the DPP algorithm with α = 0 to the independent isolated waterfilling solutions depicted in the previous figure, you'll notice that the water level for each node is identical. This aligns with intuition, as no energy is shared between nodes, making the isolated waterfilling solution optimal. As α decreases, power allocation increasingly resembles the isolated waterfilling solution. With higher loss, less energy is shared as transferring energy will no longer maximize total throughput. While the energy transferred varies for each value of α, after the energy sharing process, the optimal power allocation is determined by performing waterfilling for each node based on its resultant available power. Intriguingly, when α = 1, all channels almost “fill up” to a single waterfilling level closely resembling a unified waterfilling solution.

V. Case α = 1, Unified Water Filling

When α=1, power can be transferred to any node without any loss. Consequently, the problem can effectively be solved as a single node with N*M channels, where the total power available at the node equals the sum of the available power at all the nodes. This results in the unified waterfilling algorithm as shown below.

First we simplify the power constraint of the problem.

$$\text{C1: } \sum_{j=1}^{M} p_{ij} \leq E_i + \alpha \sum_{k=1}^{M} e_{ki} - \sum_{k=1}^{M} e_{ik}, \forall i \in \{1, 2, ... N\} \rightarrow \text{C1: } \sum_{i=1}^{N} \sum_{j=1}^{M} p_{ij} \leq \sum_{i=1}^{N} E_i$$

The problem can then be rewritten as:

$$\min: \ -\sum_{i=1}^{N} \sum_{j=1}^{M} \log(1 + \gamma_{ij} p_{ij})$$

$$\text{s.t.}$$

$$\text{C1: } \sum_{i=1}^{N} \sum_{j=1}^{M} p_{ij} \leq \sum_{i=1}^{N} E_i$$

$$\text{C2: } p_{ij} \geq 0 \ \forall i \in \{1, 2, ... N\}, \ \forall j \in \{1, 2, ... M\}$$

This can then be solved using a Lagrange multiplier:

$$\min: \; -\sum_{i=1}^{N}\sum_{j=1}^{M} \log(1+\gamma_{ij}p_{ij}) + u\sum_{i=1}^{N}\sum_{j=1}^{M} p_{ij}$$

This can then be separated and solved for each p_{ij}

$$\min: \; -\log(1+\gamma_{ij}p_{ij}) + up_{ij}, \; \forall\, i \in \{1, 2, \ldots N\}, \; \forall\, j \in \{1, 2, \ldots M\}$$

$$\frac{d}{dp_{ij}}(-\log(1+\gamma_{ij}p_{ij}) + up_{ij}) = 0$$

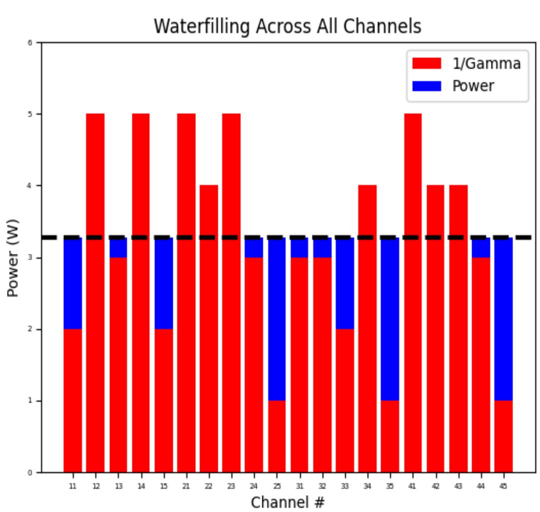
$$\frac{1}{1+\gamma_{ij}p_{ij}} + u = 0$$

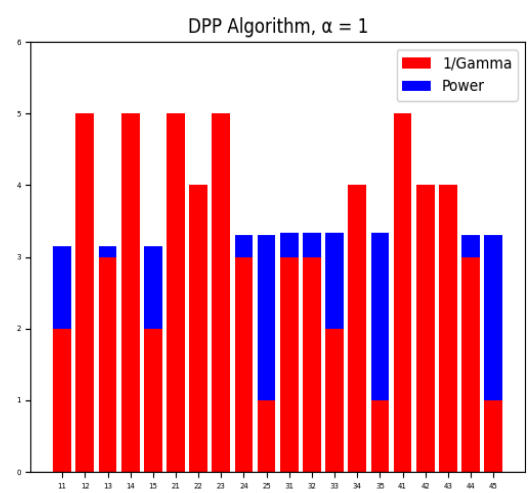
$$p_{ij} = \frac{1}{u} - \frac{1}{\gamma_{ij}}$$

$$p_{ij} = \max\left[\frac{1}{u} - \frac{1}{\gamma_{ij}}, 0\right]$$

Power is poured into each channel up to the threshold or "water level" $\frac{1}{u}$. This water level is adjusted to optimality where the total power allocated across all channels matches the total power available across all nodes.

Below are the results of both the unified waterfilling algorithm across all channels and the DPP algorithm when α=1. These results are for the scenario involving four nodes and five channels. While the DPP algorithm doesn't completely converge to the waterfilling solution due to queue oscillation, the available power is remarkably similar.





VI. General Case, 2 node scenario

a. Problem Formulation

To understand the case when α < 1, let's look at an example where there are just two nodes transferring from each other. First take the general optimization problem with Lagrange multipliers for each nodes power constraint

$$\min : \; -\sum_{i=1}^{N}\sum_{j=1}^{M}\log(1+\gamma_{ij}p_{ij}) + \sum_{i=1}^{N} u_i\left(\sum_{j=1}^{M} p_{ij} - \alpha\sum_{k=1}^{M} e_{ki} + \sum_{k=1}^{M} e_{ik}\right)$$

Then separate this into just two nodes i and k

$$\min : \; -\sum_{j=1}^{M}\log(1+\gamma_{ij}p_{ij}) - \sum_{j=1}^{M}\log(1+\gamma_{kj}p_{kj}) + u_i\left(\sum_{j=1}^{M} p_{ij} - \alpha e_{ki} + e_{ik}\right) + u_k\left(\sum_{j=1}^{M} p_{ij} - \alpha e_{ik} + e_{ki}\right)$$

Since the optimal power allocation is determined by performing waterfilling for each node based on its resultant available power, this part can be separated out from the energy sharing variables in the minimization equation. This leaves:

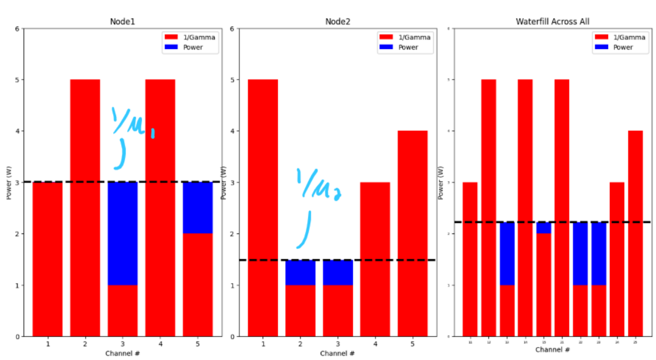
$$\min : u_i\left(-\alpha e_{ki} + e_{ik}\right) + u_k\left(-\alpha e_{ik} + e_{ki}\right)$$

$$\min : u_k e_{ki} - \alpha u_i e_{ki} + u_i e_{ik} - u_k \alpha e_{ik}$$

$$\min : e_{ki}\left(u_k - \alpha u_i\right) + e_{ik}\left(u_i - \alpha u_k\right)$$

Energy transfer should occur in only one direction. Since there is loss, transferring energy from node i to node k and vice versa would not make sense, as it would result in lost energy without any gain in throughput.

Below is a simulation involving two nodes and five channels. First, an isolated waterfilling is performed for each node independently. Then, the unified waterfilling algorithm is performed.



Let us denote the waterfilling level of the node i in an isolated (or non energy-sharing) environment as $\frac{1}{u_{i0}}$ and the waterfilling level of node i in an energy sharing environmental as $\frac{1}{u_i}$. The isolated waterfilling algorithm is calculated by:

$$p_{ij} = \max\left[\frac{1}{u_{io}} - \frac{1}{\gamma_{ij}}, 0\right] \; where \; \sum_{j=1}^{M} p_{ij} \le E_i \; \forall \, i \in \{1, 2\}$$

Energy is transferred to whichever node has a lower isolated water level.

$$\text{if } \frac{1}{u_{i0}} < \frac{1}{u_{k0}} \rightarrow e_{ki} \geq 0, \; e_{ik} = 0$$

Based on this, the optimal water levels when performing energy sharing for each node can be determined.

$$\min : e_{ki}\left(u_k - \alpha u_i\right)$$

$$\frac{d}{de_{ki}}\left(e_{ki}\left(u_k - \alpha u_i\right)\right) = 0$$

$$u_k - \alpha u_i = 0$$

$$u_k = \alpha u_i$$

$$\frac{1}{u_k} = \frac{1}{\alpha u_i}$$

In the case α = 1:

$$\frac{1}{u_k} = \frac{1}{u_i}$$

This supports our unified waterfilling solution, as in the absence of loss, the water level for both of the nodes should be equal.

In addition, it can be calculated on whether or not any waterfilling should occur.

$$\text{if } \frac{1}{u_{i0}} < \frac{1}{u_{k0}} \rightarrow e_{ki} \geq 0$$

$$\text{if } u_{i0} > u_{k0}, \rightarrow e_{ki} \geq 0$$

$$\text{if } \alpha u_{i0} \leq u_{k0} \rightarrow e_{ki} = 0$$

$$\text{if } \frac{1}{\alpha u_{i0}} \geq \frac{1}{u_{k0}} \rightarrow e_{ki} = 0$$

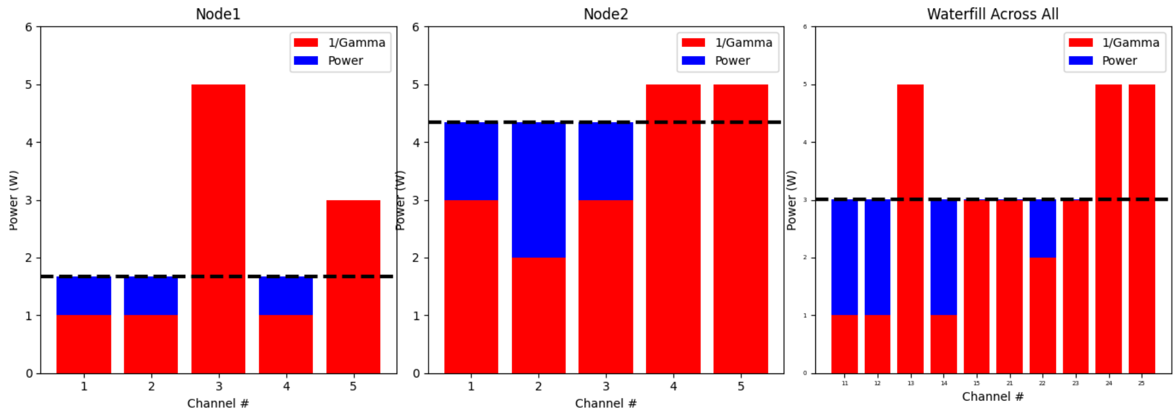
If the lower water level calculated from the isolated waterfilling algorithm divided by α is higher than the isolated water level of the other node, the loss in the system is too high, and no energy sharing should occur.

b. 2-node Algorithm

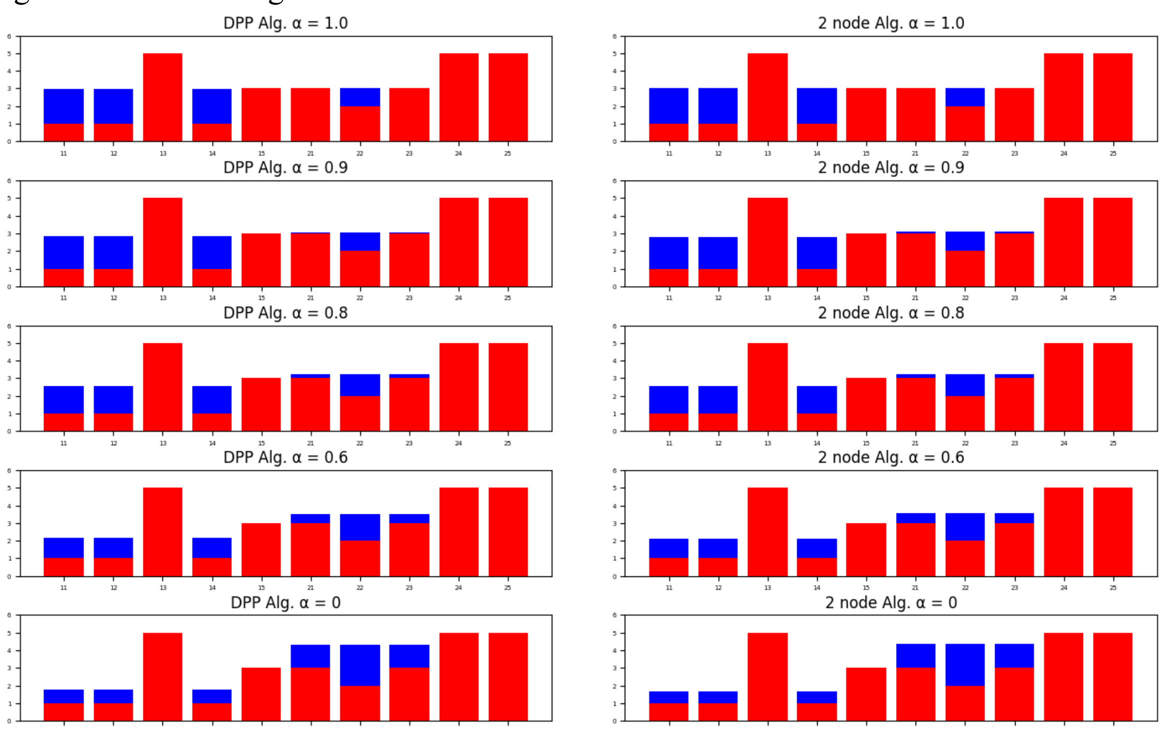
1. **Determine water filling solution with no energy sharing**
2. **Based on waterfilling values determine which node transfers energy to which**
    a. $\text{if } \frac{1}{u_{i0}} < \frac{1}{u_{k0}} \rightarrow e_{ki} \geq 0, \; e_{ik} = 0$
    b. $\text{if } \frac{1}{u_{k0}} < \frac{1}{u_{i0}} \rightarrow e_{ik} \geq 0, \; e_{ki} = 0$
3. **Determine if energy transfer should occur**
    a. $\text{if } \alpha u_{i0} > u_{k0} \rightarrow e_{ki} > 0$
4. **Create a step size. Start iteration**
    a. **transfer integer values of the step amount to and from the proper node accounting for alpha.**
    b. **Calculate the waterfilling solution of resultant energy**
    c. **Repeat until** $\alpha u_i \leq u_k$

c. Simulation

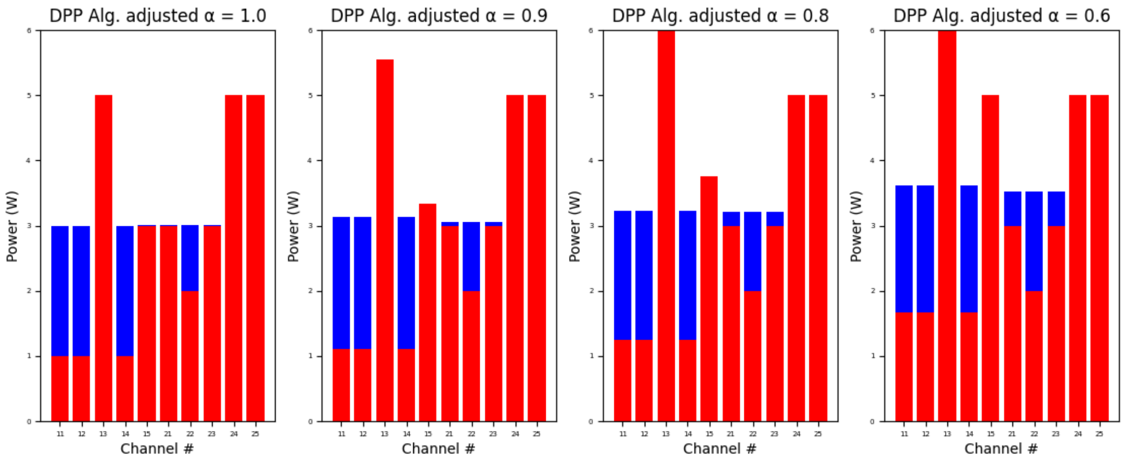
Below is another simulation involving two nodes and five channels. The first figure displays the nodes with =waterfilling algorithm with no energy sharing as well as the unified waterfilling algorithm(α= 1).



The following figure below compares the general DPP algorithm to the results of our 2-node algorithm for various alpha values. The nodes resultant power allocations are uncannily similar, demonstrating that this 2-node algorithm reflects the ideal solution.



This final figure below presents another way of visualizing this energy sharing waterfilling solution. Whichever node is receiving power is scaled by a factor of $\frac{1}{\alpha}$, as we know in the ideal solution if node i receives power from k, $\frac{1}{\alpha u_i} = \frac{1}{u_k}$. This yields a visual similar to the unified waterfilling solution when α = 1. This is executed on the DPP algorithm to again emphasize that the 2-node algorithm is the optimal solution. Note however that this will not work if no energy is shared.



VII. Future Work and Conclusion

In conclusion, the study reveals that even in the presence of energy loss, there exists an optimal deterministic solution to the 2-node energy sharing problem, where the waterfilling levels are proportional by a factor of alpha. Moreover, the application of drift plus penalty offers a viable approach to approximating multi-node solutions. It would be beneficial in the future to explore the extension of this 2-node intuition to scenarios with more nodes. Could a similar solution be derived or if it simply becomes too complex and the DPP algorithm is necessary.