2021-11-27

Finish the interpretation of Algorithm for preference list

Preference lists for Constraint Stable Matching Algorithm

In this section, a perference will be constructed based on the result from the Outer loop Algorithm. From the outer loop Algorithm , we should get three different array for storing the data: **Transmission power, power splitting ratio and Energy efficiency.** And through the Algorithm 2, for each segment of links i in SWIPT-Enabled D2D systems we should get certain optimal values of three parameters mentioned above at a certain segment of piecewise EH model j.

And for this section, we should create two preference lists for **SWIPT-Enabled sets** and **CUE** sets

1. For **SWIPT-Enabled links** , we defintely want them to match thoses links in **CUE set** which can maximize its Energy efficiency for each segment since want the sum of each segment's Energy efficiency to be **maximum.**

1.To achieve that, we first need to sort the link in CUE set,remember from algorithm 2, we already get a optimal EE array for each segment i where each segment's Energy efficiecny is already reaches the maximum point, now what we have to do is find the corresponding segment k in SiD which can help the D2D link to get the corresponding maximum value

2.So, we first need to traverse the Energy Efficiecny array, sort them in the descend order, and for each segment, we have a corresponding i segment i which is easy to find in the D2D link set, so what we need to do is to find corresponding k.

3. To solve that, we need to first traverse the CUE set then test each segment k to check if the current segment k of CUE is apporiate for help the segment i of SWIPT-Enabled D2D link to achieve the maximum value of EE of its segment.And all of this can be solved using the equation below:

We just need to use the function for each loop and store them into a temporary value

$$\mathrm{EE}_i^D=\frac{T_i^D}{\mathrm{EC}_i^D}=\frac{\log_2\left(1+\frac{P_i^D h_i^D}{(P_k^C h_{k,i}+N_0)+\frac{N_1}{(1-\lambda_i^e)}}\right)}{P_i^D+2P_{\mathrm{cir}}-\mathrm{EH}_i^D}.$$

2. And for the CUE, the partner selection can also be done by using the same methodology: at the CUE link set, for each interation, we need to scan the D2D link, and sort the interference power in ascending order and place the corresponding D2D link, then the preference list for CUE can be established.

2. And for the CUE, the partner selection can also be done by using the same methodology: at the CUE link set, for each interation, we need to scan the D2D link, and sort the interference power in ascending order and place the corresponding D2D link, then the preference list for CUE can be established.

$$\begin{aligned}\mathrm{SINR}_k^C&=\frac{P_k^C h_k^C}{P_i^D h_i^B+N_0+N_1}\\&=\frac{P_k^C d_k^{C-\alpha} h_k^{\prime C}}{P_i^D d_i^B h_i^{B'}+N_0+N_1}\\\mathrm{SINR}_i^D&=\frac{P_i^D h_i^D}{P_k^C h_{k,i}+N_0+N_1}\\&=\frac{P_i^D d_i^{D-\alpha} h_i^{\prime D}}{P_k^C d_{k,i}{}^{-\alpha} h_{k,i}^{\prime}+N_0+N_1}\end{aligned}$$

where the interference powe can be calculated by P_iD*d_iB*h_iB_rayleigh. Where the h_iB_rayleigh is the rayleigh channel coefficient and diB is the distance from D2D link to basestation. **Since the method for calculating rayleigh channel coefficient haven't been figured out, so hB_rayleigh will be just used for it temporarily.**

1. First traverse the CUE set, then for eack link in the CUE set k, scan each link i in SWIPT-Enabled group.
2. For each link i, there will be intereference between i and basestation, create an array for storing the interference power on single D2D link
3. For each loop of D2D link, sort sort each the intereference power in ascending oder
4. Then repeat the step 1 and step 2, placing the corresponding D2D link in the partner selection set using hashmap.
5. In this case, for each link k from CUE set, the interference power will not get affected by link k.