Date:2022-01-08

Continue with the interim report and finish the One-to-one stable matching algorithm section:

4.2.8 One-to-one stable matching algorithm

Based on the result of Algorithm 4, each SWIPT-Supported D2D link j and CUE k has a preference list respectively. A perfect stable matching will be called if all the SWIPT-Supported D2D links can find their own partners. However, sometimes it is very hard to achieve that under some circumstances like bad channel condition or sometimes the distance of communication is too long. Also, as mentioned in the pre-matching algorithm, each SWIPT-Supported D2D link j will be pre-matched with different number of CUEs because of EH sensitivity which means some SWIPT-Supported D2D links select limited number of CUEs, under that circumstance, some SWIPT-Supported D2D links have limited number of CUEs in its partner selection. Also, considering each CUE k has its own preference list Ω_k^C as well, so they may match D2D link based on its preference list. All the situations above will lead to some D2D links failing find a CUE to match. Considering user fairness, a one-to-one stable matching algorithm is proposed to solve this problem.

In [20], a group proposing to other groups will be supported by stable matching. So, in algorithm 5, for each D2D link j, it tries to send a proposal to a CUE k first based on its preference list Ω_i^D. Then this algorithm will check if the CUE k receives another proposal from other D2D link α. If the current CUE k receives only one proposal from SWIPT-Supported D2D link j, then CUE k will be thought of as the D2D link i's best partner and it will be placed into the partner group ϕ. If the CUE k receives more than one proposal from other D2D α, then this algorithm will check if one of them only has one preferred CUE in its partner selection, if there is a D2D link j having only one preferred CUE k in its preference list, then CUE k will accept it as its best partners and reject the others. If all the D2D links sending proposal to CUE k have more than one preferred CUE in their preference lists, then CUE k will choose one of them based on its preference list Ω_k^C and the others will be rejected.

In each loop of a SWIPT-Supported D2D link j, if it is matched with a CUE k, then it will be removed from the group $EnaD$, and if it is rejected by a CUE k, it will be added to $EnaD$. And no matter whether the SWIPT-Supported D2D link is matched with its most preferred CUE k in its preference list, it will delete its current CUE k from its preference list at end of each loop.

After all the SWIPT-Supported D2D links have been matched with one CUE, all the unmatched CUEs from preference list will be gathered into ϕ^R which will be reused by non-EH D2D links from pre-matching algorithm.

The pseudo code for this algorithm based on my understadning is given as following:

Algorithm 5: One-to-one Stable Matching Algorithm

Input : $\Omega_i^D, \Omega_k^C, EnaD, C, PS$

Output : ϕ, ϕ^R

Step 1 : **while** $EnaD \neq \phi$ **do**

Step 2 : **for** $i \in EnaD$ **do**

Step 3 : Let i propose its most preferred CUE Ω_i^D which should be the first CUE in Ω_i^D

Step 4 : **for** $k \in PS$

Step 6 : **if** k is the most preferred CUE for i **then**

Step 7 : **if** k receives only one proposal from D2D link i then

Step 8 : i will be matched with k,$\phi = (i,k)$

Step 9 : The matched i will be removed form $EnaD$

break

Step 10 : **elseif** k receives not only one proposal from different D2D links **then**

Step 11 : **if** one of the D2D links α has only one preferred CUE in its preference list **then**

Step 12 : α will be matched with k, $\phi = (i,k)$, it will be removed from $EnaD$

Step 13 : **elseif** all the D2D links have more than one preferred CUE in their preference lists **then**

Step 14 : The CUE k will be matched with its most preferred D2D link i' from its preference list Ω_k^C, and the matched D2D link will be removed form $EnaD$

Step 15 : **end if**

Step 15 : **end if**

Step 16 : **elseif** k **is the most preferred CUE for** i **then**

Step 17 : **continue**

Step 18 : **end if**

Step 19 : **end for**

Step 20 : i deletes its most preferred CUE from Ω_i^D

Step 21 : **end for**

Step 22 : **end while**

Step 23 : Gather all the unmatched CUEs in ϕ_R