Let have a graph G where:

- Vertices are the:
    - users which are not paired
    - and the paired users over D2D
      Note: The paired users are one vertex. You can think we had a graph G' where every user was a vertex and then we merged the D2D users vertices into one pair. Assuming they are distinct.
- Edges: lets start with no edges.

Note: Coloring the vertices is the same thing as assigning frequencies to the users and D2D pairs.

If (# of frequencies >= number of base station users + number of D2D pairs) {
You can just color all the base station users with one color and then color each D2D pair with the other colors => you would have no interference and no negative cost for not taking a D2D pair => it would be a minimum solution.
} else heuristically {
Lets start form a minimum above graph where you have not assigned the colors yet. Also the D2D pairs we choose to start with are the ones which have the biggest cost of not taking (heuristic choice). Now we want to add more D2D pairs. This pair should have on of the colors that already exits => when we connect it we will have and edge (clique) with the color we connect it to. Heuristically connect the new D2D pair to the color which gives the least amount of cost. The way you calculate this cost is:
if(you connect the pair to base station) {
cost = the cost of not taking the pair
} else {
cost = the interference joining the clique of the same color
}
Now we have to choose where do you want to connect the new D2D pair. We can find the min weight outgoing edge (as seen in picture) and compare it with the cost of not taking 2D2.
... You repeat this process until you are finished
}

Cost of not taking 2D2 pair

Base station users

New 2D2 pair

min weight

D2D pairs with the heights cost of not taking

Cost of not taking 2D2 pair

Base station users

another new 2D2 pair

D2D pairs with the heights cost of not taking