Date:2022-01-02

Continue with my interim report:

Formulate the EE maximization by transforming the original equation

0/381/480 Interim Report

$$EE_i^D = \frac{T_i^D}{EC_i^D} = \frac{log2\left(1 + \frac{(1-\theta_i)P_i^D h_i^D}{(1-\theta_i)(P_k^c h_{k,i} + N_0) + N_1}\right)}{P_i^D + 2P_{circuit} - EH_i^D} \qquad (6)$$

to a Lagrange optimization problem:

4.2.4 Formulation of each D2D link's EE maximization problem

According to equation (6), since the value of power splitting ratio θ_i^D and transmission power P_i^D of each SWIPT-Supported D2D link i are unknown, so they can be thought of as two variables of the EE_i^D. So, the problem of maximizing EE of each SWIPT-Supported D2D link i can be transformed to finding each D2D link i's optimal value of power splitting ratio and transmission power with constraints. It can be formulated as:

$$\boldsymbol{C1}: \max_{\theta_i, P_i^D, i \in EnaD} EE_i^D$$

$$\text{s.t.} \quad \begin{aligned} &N1: 0 < P_i^D < P_{max} \\ &N2: 0 \le \theta_i \le 1 \\ &N3: T_i^D \ge T_{min}^D \\ &N4: T_k^C \ge T_{min}^C \\ &N5: P_i^R \ge P_{thresold}^1 \\ &N6: P_{thresold}^j \le P_i^R \le P_{thresold}^{j+1} \end{aligned} \qquad (10)$$

Where N1-N6 are constraints of this formulated problem. N1 shows that the transmission cannot exceed the maximum transmission power. N2 establishes the range of the power splitting ratio of each SWIPT-Supported D2D link i. N3 and N4 set the minimum Throughput which is required for each SWIPT-Supported D2D link i and CUE k respectively. N5 shows the minimum received power that needs to be achieved for activating EH model. N6 shows that the receiver of D2D link i will work at the jth segment of the piecewise linear EH model.

However, (6) is a fractional function which is hard to solve directly. According to [18] a maximization problem like $\max\{N(x)/D(x)\}$ can be transformed to another non-fractional maximization problem $\max\{N(x)-qD(x)\}$ where q is the solution of the former maximization problem. So $\boldsymbol{C1}$ can be similarly transformed to a simpler non-fractional function $\boldsymbol{C2}$:

$$\boldsymbol{C2}: \max_{\theta_i, P_i^D, i\in EnaD} T_i^D - G_i^D EC_i^D \qquad (11)$$

$$\text{s.t.} \quad N1-N6$$

According to [22], to achieve G_i^D, there is one theorem that can be used:

Theorem 1: The optimal G_i^{D*} can be achieved only when $T_i^{D*} - G_i^{D*}EC_i^{D*} = 0$, where G_i^{D*} is the solution of $\boldsymbol{C1}$, and T_i^{D*}, EC_i^{D*} are the optimal values of Throughput, total energy consumption of each SWIPT-Supported D2D link i.

To solve $\boldsymbol{C2}$, it can be thought of as an optimization problem with multiple inequality constraints. So the corresponding Lagrange function can be given:

$$\begin{aligned} &L(P_i^D, \theta_i, \alpha, \beta, \gamma, \delta, \sigma) \\ &= T_i^D - G_i^D EC_i^D - \alpha(P_i^D - P_{max}) - \beta(\theta_i - 1) + \gamma(T_i^D - T_{min}^D) \quad (12) \\ &+ \delta\left(T_i^C - T_{min}^C\right) + \sigma(P_i^R - P_{thresold}^1) \end{aligned}$$

Where $\alpha, \beta, \gamma, \delta, \sigma$ are the Lagrange multipliers. The original optimization can be solved by employing the Lagrange dual optimization function. For (11), there is a proposition:

$$\boldsymbol{Proposition\ 1}: Let\ T_i^D - G_i^D EC_i^D = f(x), then\ f(x) = \min\ L(P_i^D, \theta_i, \alpha, \beta, \gamma, \delta, \sigma)$$

So, the original optimization problem can be expressed as:

$$\max f(x) = \max\min\ f(x) = \min\ L(P_i^D, \theta_i, \alpha, \beta, \gamma, \delta, \sigma)$$

Then according to duality of Lagrange function, the corresponding Lagrange dual problem can be expressed as:

$$\boldsymbol{C3}\ \min_{\alpha,\beta,\gamma,\delta,\sigma\geq 0}\ \max_{(P_i^D, \theta_i)}\ L(P_i^D, \theta_i, \alpha, \beta, \gamma, \delta, \sigma)\ i \in EnaD$$

To solve $\boldsymbol{C3}$, Here is a proposition which need to be used first:

$\boldsymbol{Proposition\ 2}$: $\boldsymbol{C3}$ is convex with respect to θ_i and with respect to P_i^D when $\theta\_(i)$ and P_i^D are fixed respectively.

Then based on the result of proposition 2, by using block coordinate descent (BCD) method, the local

$$\frac{L(P_i^D,\theta_i,\alpha,\beta,\gamma,\delta,\sigma)}{\partial P_i^D} = 0 \qquad (14)$$

It is noted that it is very difficult to find optimal value of all of the Lagrange multipliers by solving (13) and (14) directly. So, in this project, all of the Lagrange multipliers will be solved by using gradient method. For each iteration, the Lagrange multipliers will be updated using a step size (learning rate) for each iteration.

$$\begin{aligned} \alpha &= \{\alpha + s_1(P_i^D - P_{max})\}^+ \\ \beta &= \{\beta + s_2(\theta_i - 1)\}^+ \\ \gamma &= \{\gamma + s_3(T_i^D - T_{min}^D)\}^+ \\ \delta &= \{\delta + s_4(T_i^C - T_{min}^C)\}^+ \\ \sigma &= \{\sigma + s_5(P_i^R - P_{thresold}^1)\}^+ \end{aligned} \qquad (15)$$

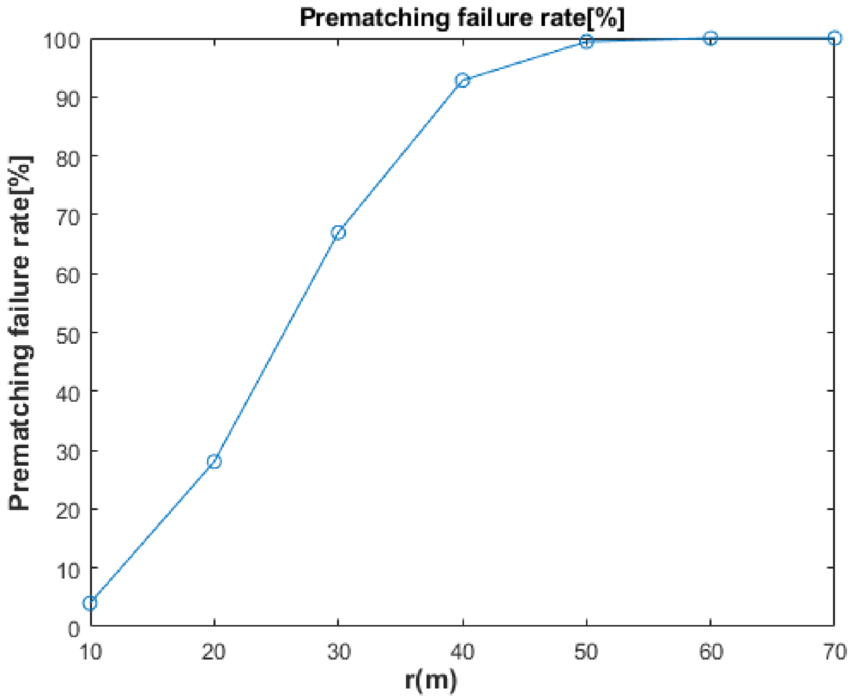
Where s_1, s_2, s_3, s_4, s_5 are the corresponding step size based on relevant constraints. The symbol $\{M\}^+ = \max\{0, M\}$. The step size is a parameter determining how will a working model change according to error every time some parameters of the model are changed. It is quite important to establish the proper step size before running a model, for example, if it was set too big, the model will diverge and if it is set too small, the model will converge very slowly [19]. It is usually established based on the objective function and tested parameters, in this project, they are all set to 10^{-5}.

For the initialization problem of the Lagrange multipliers, the method is based on ***Comparison of Various Learning Rate Scheduling Techniques on Convolutional Neural Network***.

Implement two test functions for pre-matching function for communication distance and number of D2D links

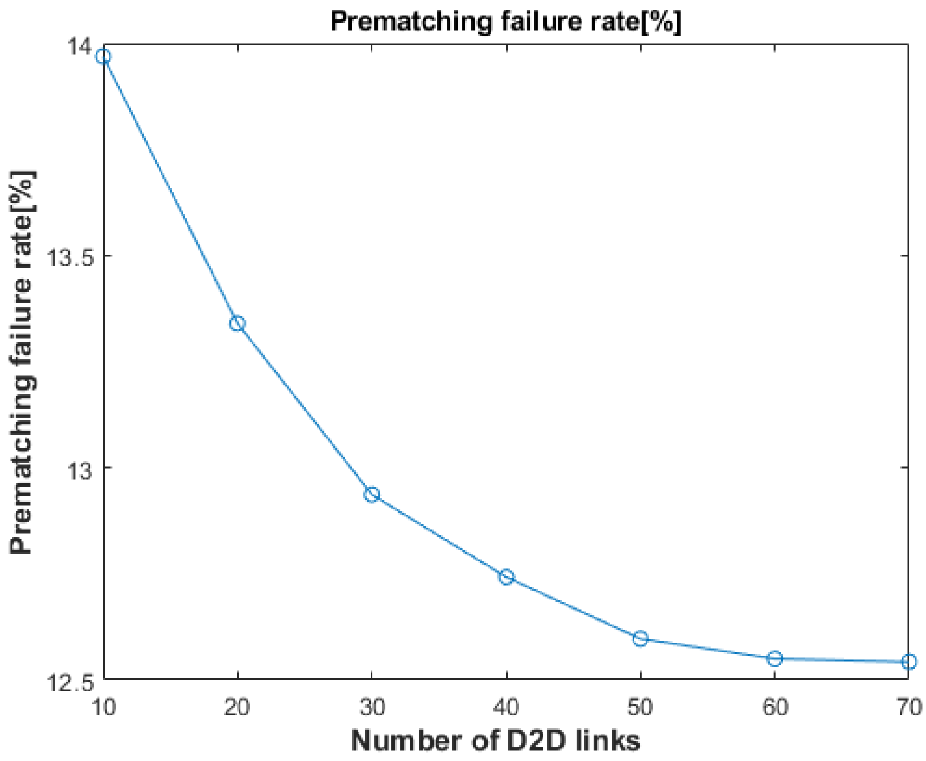
1.pre-matching test at different communication distance:

```
function test_prematching_distance()
Pkc=0.1995262315;
Pth1=10*10^(-6);
Pmax=0.1995262315;
Tmin=2;
total=[];
for p=10:10:70
fail_rate=[];
for i=1:100
[D2D,CUE]=system_model(30,30,p);
for j=1:100
[Sid,InfD,EhaD]=Prematch(D2D,CUE,Pkc,Pth1,Pmax,Tmin,p);
fail=size(InfD,2)/30;
fail_rate(i,j)=fail;
end
end
fail_rate_total=sum(fail_rate(:))/10000;
total(end+1)=fail_rate_total;
p
end
number=10:10:70;
total=100*total;
plot(number,total,'-o');
title('Prematching failure rate[%]');
xlabel('r(m)');
ylabel('Prematching failure rate[%]');
saveas(gcf,[pwd '/simulation_results/Prematch_versus_distance.fig']);
end
```



2.pre-matching test at different number of D2D links

```
function test_premtaching_number()
Pkc=0.1995262315;
Pth1=10*10^(-6);
Pmax=0.1995262315;
Tmin=2;
total=[];
for p=10:10:70
fail_rate=[];
for i=1:100
[D2D,CUE]=system_model(p,p,15);
for j=1:100
[Sid,InfD,EhaD]=Prematch(D2D,CUE,Pkc,Pth1,Pmax,Tmin,15);
fail=size(InfD,2)/p;
fail_rate(i,j)=fail;
end
end
fail_rate_total=sum(fail_rate(:))/10000;
total(end+1)=fail_rate_total;
p
end
number=10:10:70;
total=100*total;
total
plot(number,total,'-o')
title('Prematching failure rate[%]')
xlabel('Number of D2D links')
ylabel('Prematching failure rate[%]')
saveas(gcf,[pwd '/simulation_results/Prematch_versus_number.fig']);
end
```



The result from my reference paper:

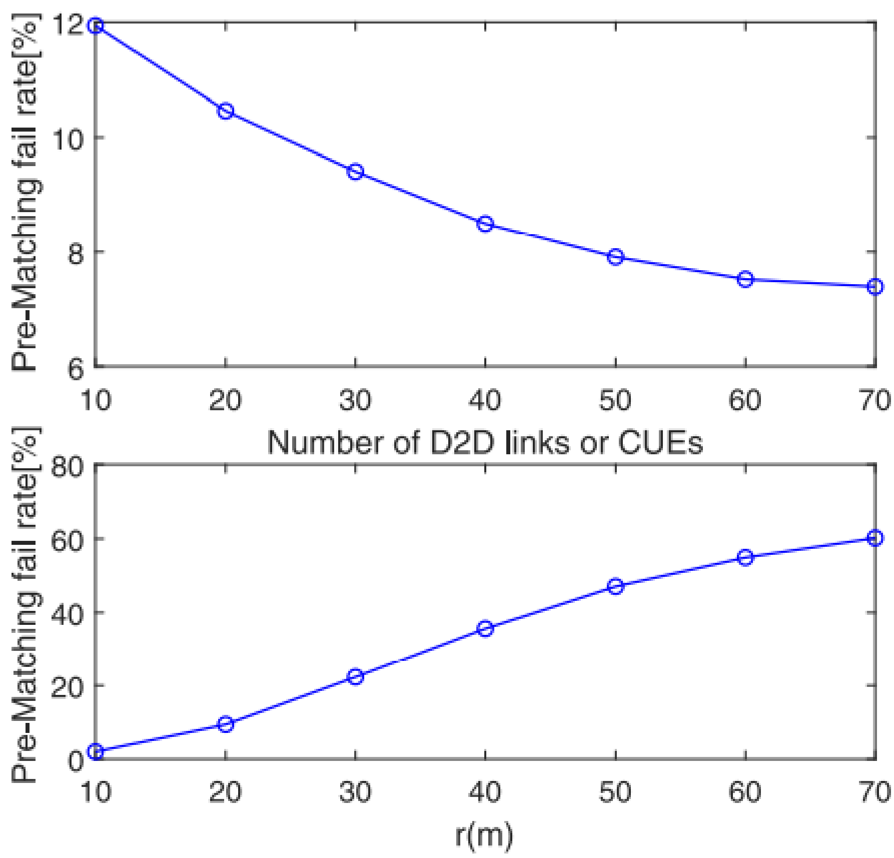


Fig. 3. PMFR versus the number of D2D links (or CUEs) and D2D communication distance.

Compared to the result of my reference paper, the trend is very similar.