Date:12/10/2021

Still working on the paper ***Resource and Power Allocation in SWIPT-Enabled Device-to-Device Communications Based on a Nonlinear Energy Harvesting Model***. Record some notes for the piecewise linear EH model mentioned in this paper, and methods for transforming the linear programming model into nonlinear programming model.

Technical term

- **Piecewise linear functions**

  It is a function whose graph consists of straight line segments. it is a function of which the each piecewise is linear.

$$f(x) = \begin{cases} 2x & x <= 2 \\ -x+3 & x > 2 \end{cases}$$

  For the linear EH model mentioned in this paper:

$$EH_i^D = \begin{cases} 0 & P_i^R \in [P_{tn}^0, P_{th}^1] \\ k_j P^R + b_j, & P_i^R \in [P_{th}^j, P_{th}^{j+1}], j \in 1, \ldots, L-1 \\ P_{max}^{EH} & P_i^R \in [P_{th}^L, P_{th}^{L+1}] \end{cases}$$

  Where the EH_i^D is the power harvested by D2D receiver i , and P_i^R is the received power for EH at D2D receiver i when sharing the RB with CUE k, which can be expressed as:

$$P_i^R = \lambda_i^e (P_i^D + P_k^C h_{k,i} + N_0)$$

  Note that $P_{th} = \{P_{th} | 1 \leq j \leq L+1\}$ is the set of thresholds on P_i^R for $L+1$ linear segments. The k_j and b_j are the coefficients and the intercept of the linear function in the j_{th} segment. P_{th}^1 denotes the minimum received power requirement for activating the **RF EH** circuit, which is also the circuit sensitivity of the EH circuit, and te P_{max}^{EH} is the maximum power the **RF EH** circuit can harvest.

  So the above piecewise linear EH model shows the **different amount of energy that the system can harvest at different segment.**

- **Maximization of energy efficiency(EE)for SWIPT-enabled D2D links**

  The final equation of **Energy Efficiency(EE)** for D2D links can be expressed by

$$EE_i^D = \frac{T_i^D}{EC_i^D} = \frac{log_2(1 + \frac{P_i^D h_i^D}{(P_k^C h_{k,i} + N_0) + \frac{N_1}{1-\lambda_i^e}})}{P_i^D + 2P_{cir} - EH_i^D}$$

  As shown in the equation, if i want to find the maximum value of the Energy Efficiency, it is all about finding a optimal value for the transmission power at D2D link i (P_i^D), the harvest energy from the system(EH_i^D), the power splitting ratio(λ_i^e).

  And in this paper, the transmission power for **CUE link is constant for simulation。**

  Accordingly, In this paper the EE maximization problem was formulated as

$$
\begin{aligned}
\mathbf{P1}: \quad & \max_{\{P_i^D, \lambda_e^i, i \in Eha^D\}} \quad EE_i^D \\
\text{s.t.} \quad & \mathrm{C1}: 0 < P_i^D \leq P_{\max} \\
& \mathrm{C2}: 0 \leq \lambda_i^e \leq 1 \\
& \mathrm{C3}: T_i^D \geq \mathrm{T}_{\min}^{\mathrm{D}} \\
& \mathrm{C4}: T_k^C \geq \mathrm{T}_{\min}^{\mathrm{C}} \\
& \mathrm{C5}: P_i^R \geq P_{th}^1 \\
& \mathrm{C6}: P_{th}^j \leq P_i^R \leq P_{th}^{j+1}, \quad j \in 0, \ldots, L \quad (12)
\end{aligned}
$$

- **non-linear programming**

  In mathematics, **nonlinear programming** (**NLP**) is the process of solving an optimization problem where some of the constraints or the objective function are nonlinear

- **linear programming:**

  Linear programming is a simple technique where we **depict** complex relationships through linear functions and then find the optimum points. The important word in the previous sentence is depicted. The real relationships might be much more complex – but we can simplify them to linear relationships.

  As shown in the figure 1, it is a very classic example for people to use LP to save on fuel and time and find the shortest route.

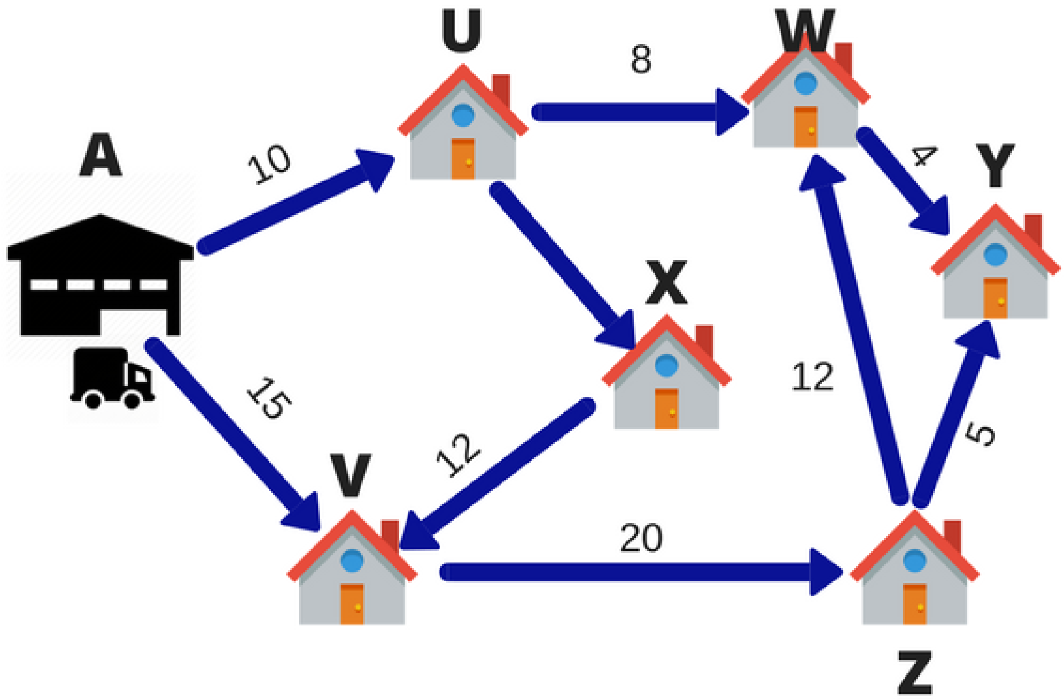

Figure 1: A simple example for linear programming

Notation definition

- $argmax_S f = \{x \in f(s) \quad for \quad all \quad s \in S\}$:$argmax$ is the set of points x for which $f(x)$ attains the function's largest value(if it exists). $Argmax$ may be the empty set, a singleton, or contain multiple elements.

Algorithm

- The **Outer Loop Algorithm**

Algorithm 2 TLEEIA—Outer Loop Algorithm

Input: $Eha^D, S_i^D, \lambda_{i,j}^e, P_{i,j}^D, EE_{i,j}^D$

Output: $P_i^{D*}, \lambda_i^{e*}, EE_i^{D*}$.

```
for i ∈ Eha^D do
    for k ∈ S_i^D do
        for j = 1 : N_max do
            j* = arg max_j {EE^D_{i,1}, ..., EE^D_{i,j}, ..., EE^D_{i,N_max}}.
            Obtain P_i^{D*} = P^D_{i,j*}, λ_i^{e*} = λ^e_{i,j*}, EE_i^{D*} = EE^D_{i,j*}.
        end for
    end for
end for
```

$$j^* = \arg\max_j \{EE_{i,1}^D, \ldots, EE_{i,j}^D, \ldots, EE_{i,N_{max}}^D\}.$$

$$P_i^{D*} = P_{i,j^*}^D,\ \lambda_i^{e*} = \lambda_{i,j^*}^e,\ EE_i^{D*} = EE_{i,j^*}^D.$$