Problem Statement:

- **Personalized PageRank Model**
  - Seed node
  - Random surfer to neighbors with probability \((1 - \alpha)\)
  - Jump to seed nodes with probability \(\alpha\)
  - The stable probability can be computed by the seed nodes.

- **Problem**
  - **Input**: Given a graph \(G\), a set of seed nodes \(P\), and teleport probability \(\alpha\).
  - **Output**: Find the Personalized PageRank Vector (PPV) \(r_P\) which is computed as
    \[
    r_P = (1 - \alpha)A^T r_P + \alpha u_P,
    \]
    where
    - \(A^T\) is the normalized adjacency matrix,
    - \(u_P\) is the user preference vector.

- **Challenge**
  - **Exactness**: Most existing methods focus on approximate PPV computation, exact PPV is hard to compute.
  - **Parallel**: It is hard to design scalable distributed algorithm to compute PPV that works in iteration.
  - **Costs**: It requires high time, space and network costs for distributed graph computation.

Approaches:

- **Graph Partition Based Algorithm**
  - If we choose the hub nodes that can separate the graph, the size of partial vector can be bounded inside a subgraph.

- **Hierarchical Graph Partition Based Algorithm**
  - The partial vector computation in a subgraph is to compute a "local" PPV. We can further partition the subgraph recursively.

Background:

- **From PPV to random tours**
  - PPV scores can be computed by random tours
  - Example: there are 3 random tours from \(u_i\) to \(u_j\)
    - \(t_1: u_1 \rightarrow u_2 \rightarrow u_3\)
    - \(t_2: u_1 \rightarrow u_3 \rightarrow u_4 \rightarrow u_5\)
    - \(t_3: u_2 \rightarrow u_4 \rightarrow u_5\)
  - The PPV score can be computed by adding up the weight of all possible random tours.
    \[
    r_{u_i}(u_j) = P(t_1) + P(t_2) + P(t_3)
    \]
    \[
    P(t) = \alpha(1 - \alpha)^{w(t)} \prod_{u \in t} \frac{1}{\text{in}(u)}
    \]

- **Random Tour Decomposition**
  - If we select some nodes to be hub nodes
    1. The random tours can be decomposed by these hub nodes.
    2. Result in two types of tours
      - **Partial vector**: tours passing through no hub nodes
        \[
        p_{u_i} = P(t_1) + P(t_2) = P(u_1 \rightarrow u_4) + P(u_1 \rightarrow u_2 \rightarrow u_3)
        \]
      - **Skeleton vector**: tours stop at a hub node
        \[
        s_{u_i} = P(t_2) + P(t_3) = P(u_1 \rightarrow u_3) + P(u_1 \rightarrow u_2 \rightarrow u_3) + P(u_1 \rightarrow u_2 \rightarrow u_5)
        \]
    - All possible tours can be constructed by partial vectors and skeleton vectors.

- **Example**: Consider a tour \(u_1 \rightarrow u_2 \rightarrow u_3\)
  - In skeleton vector of \(u_1\)
  - In partial vector of \(u_1\)

Experimental Results:

- **Baselines**
  - Approximate: FastPPV [Fanwei Zhu, VLDB 2013]
  - Exact: Power iteration
  - Graph Processing Systems: Pregel [Da Yan, VLDB 2014], Blogel [Da Yan, VLDB 2014]