User Guide

Mike Phulsuksombati

The program provide the interactive interface to visualize a water distribution network as well as solving the optimization problem to find the optimal flow and pressure of the network that consume the least energy. The demo of the interface can be found online here.

Installation

1. Clone the project from GitHub

   ```
   git clone https://github.com/mikephul/pump_webapp.git
   cd pump_webapp
   ```

2. Install the requirement

   ```
   pip install -r requirements.txt
   ```

3. Obtain and install the mosek license file

4. Make sure that the project folder contains the .inp file that describe the water distribution network (We include the Small.inp and Big.inp with the project.)

5. Execute the flask application with the script below. This will open the flask webserver.

   ```
   python app.py Small.inp
   ```

6. Pull up any web broweser (Chrome is recommended.) You can see the application at the address below. The default port is 5000.

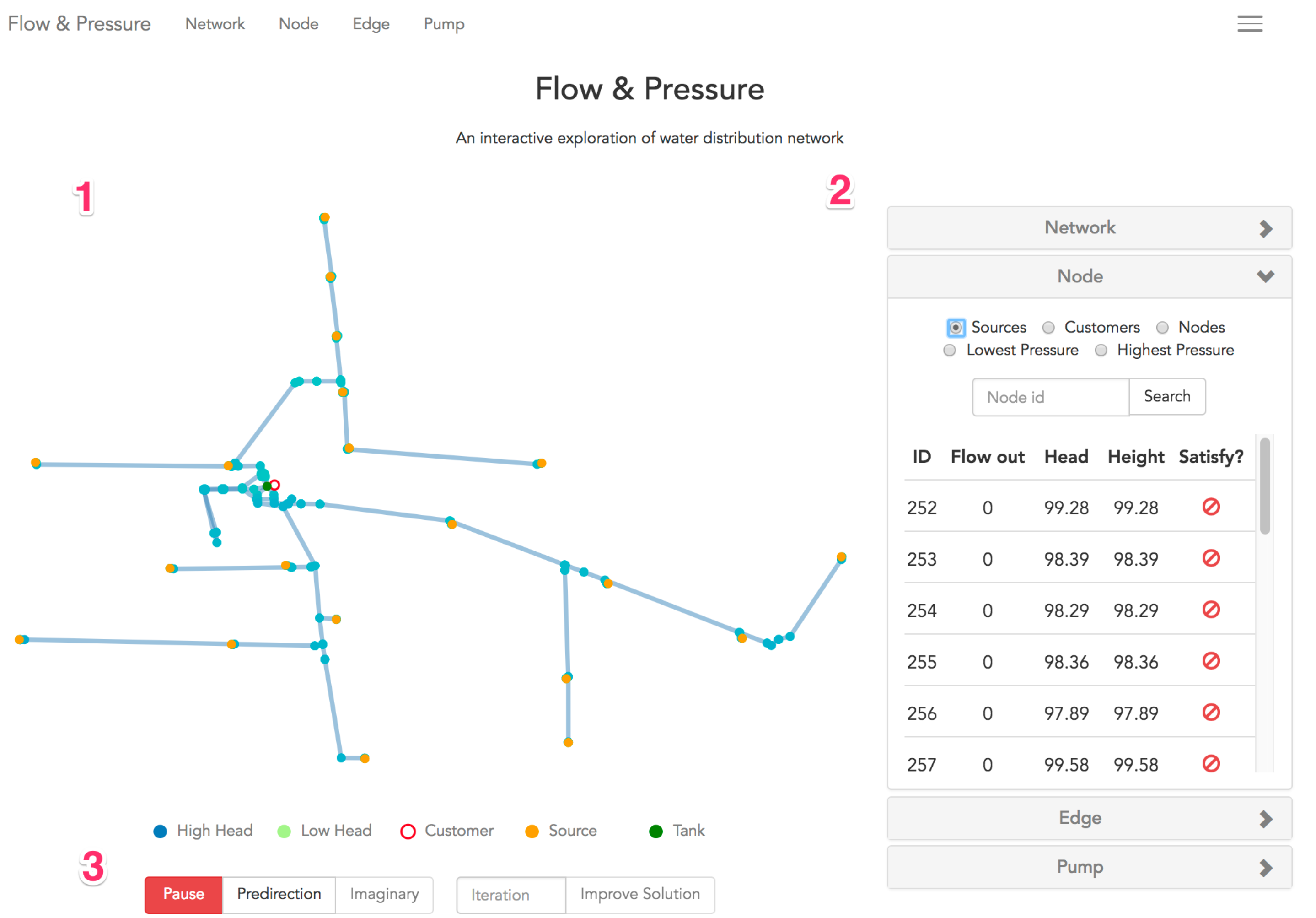
   ```
   http://localhost:5000/
   ```

   Then, you should see the following interface.

Feature

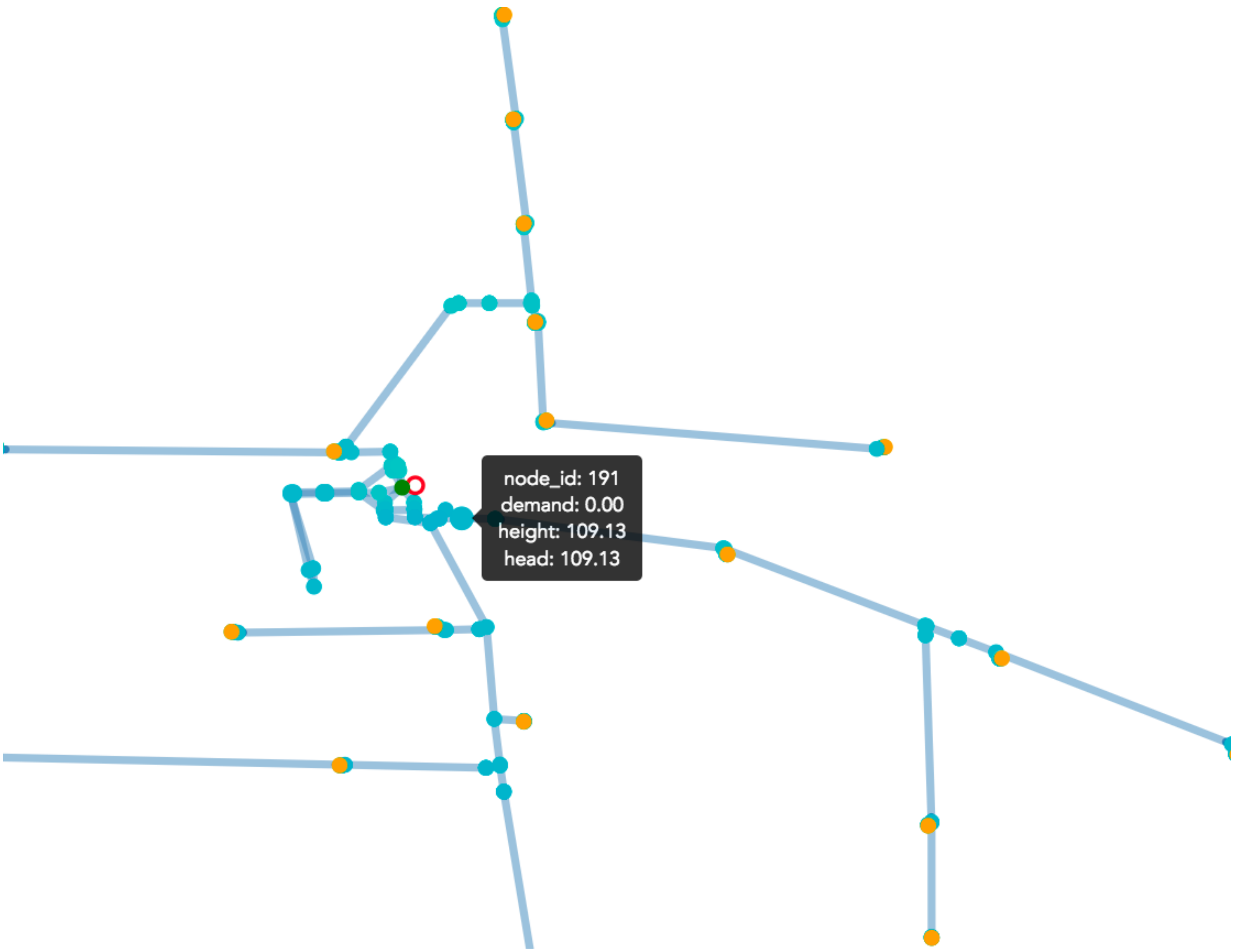
There are three main feature of the application

1. Network Geometry
2. Network Information (Node and Edge)
3. Solver actions



1. Network Geometry

The geometry of the input network gives a sense of where each components in the network is located physically. When hover cursor over the the network, there is a tooltip that show the information of the particular node or edge.

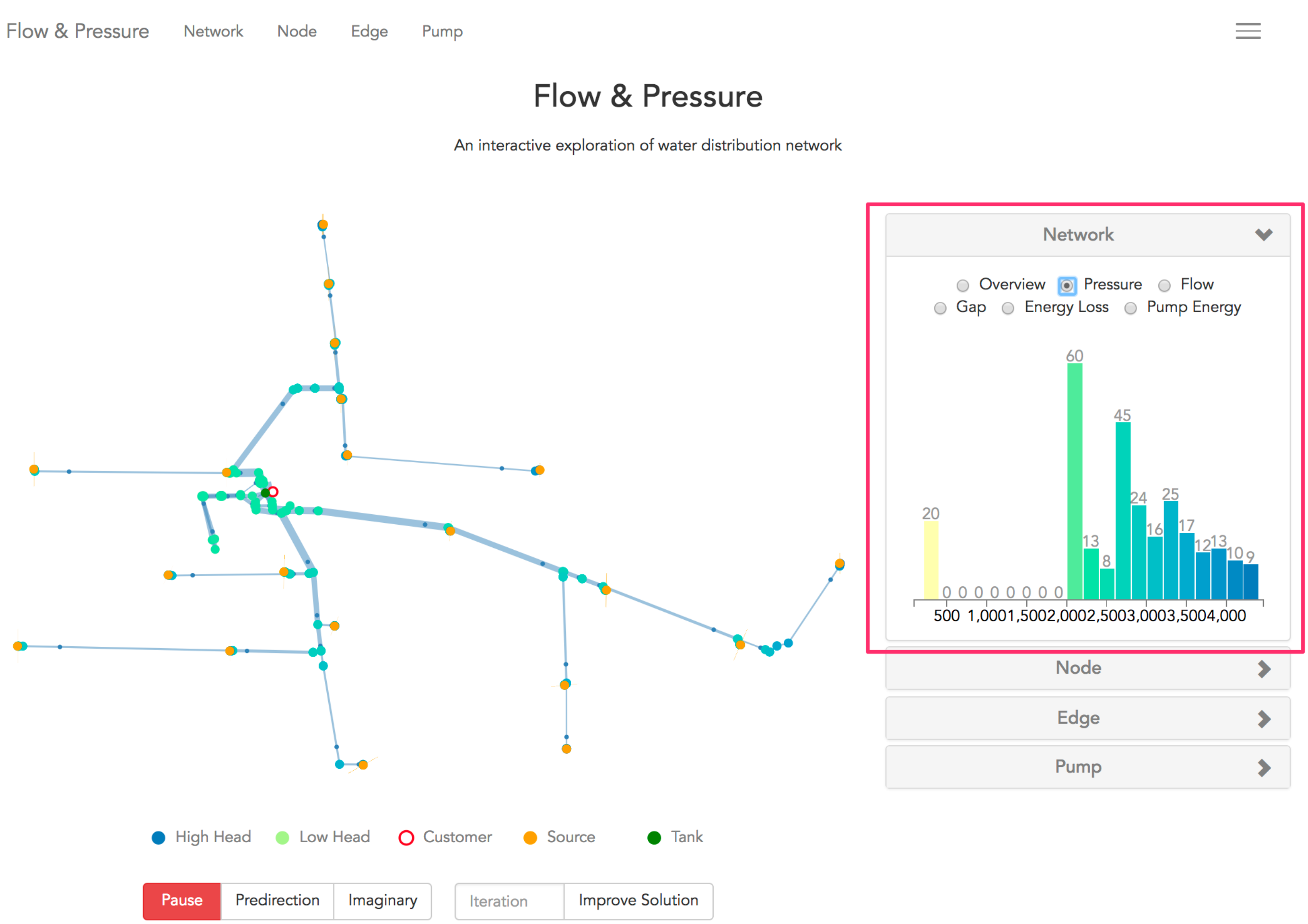


2. Network Information

You can choose to see the information of the network, nodes, edges, or pumps.

Network

- **Overview** - shows the general info about network such as number of nodes or edges, and etc.
- **Pressure** - shows the histogram of pressure at each nodes
- **Flow** - shows the histogram of flow at each edges
- **Gap** - shows how much the pressure-flow coupling constraint gap is violated at each iteration
- **Energy Loss** - shows the energy that lost within the network
- **Pump Energy** - shows the energy consumer by pumps

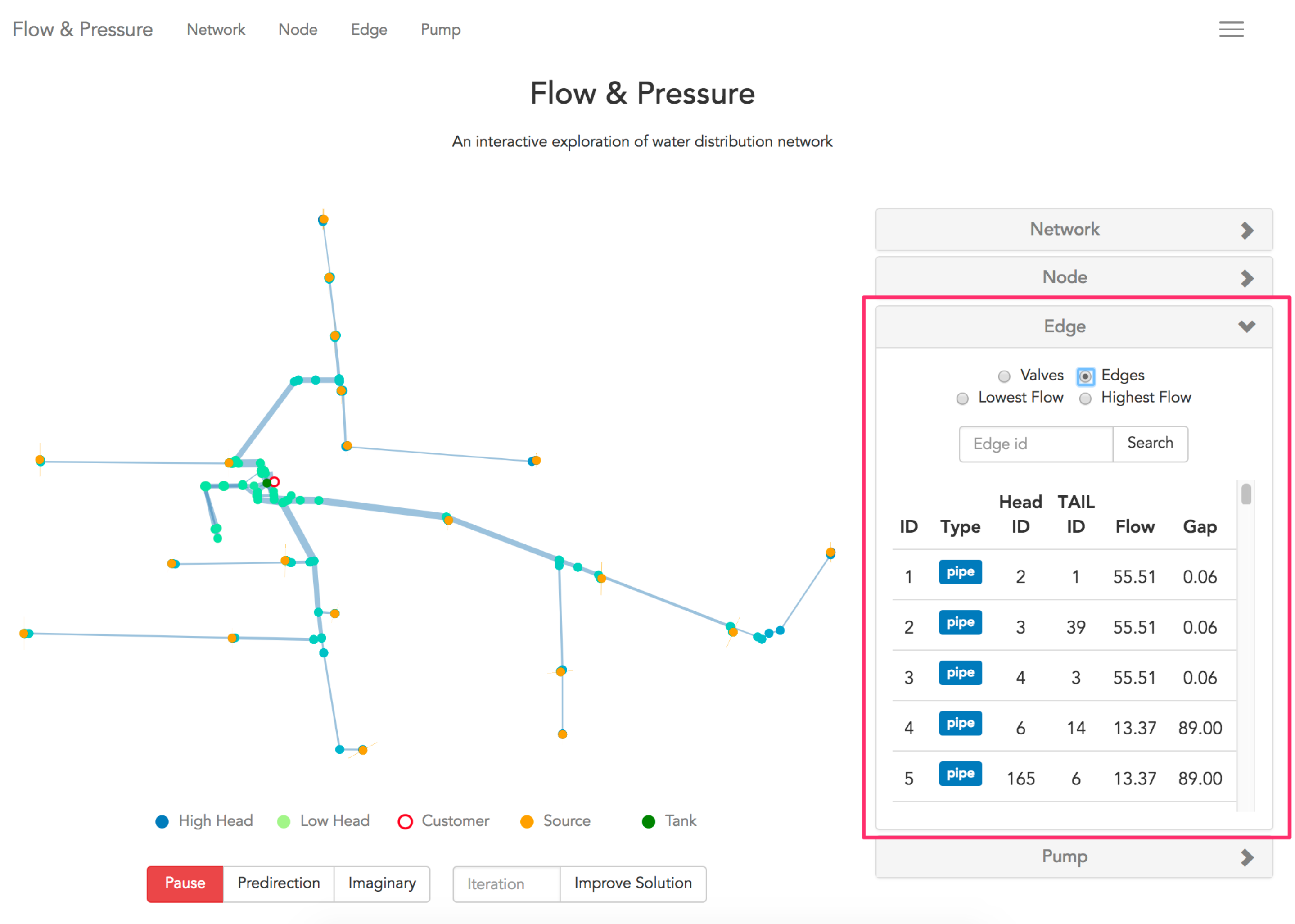


Node

- **Sources** - shows information about source node
- **Customers** - shows information about customer node
- **Nodes** - shows information of all nodes
- **Lowest Pressure** - show information of the top 5 lowest pressure nodes
- **Highest Pressure** - show information of the top 5 highest pressure nodes
- **Search by Node id** - show information of the specified node

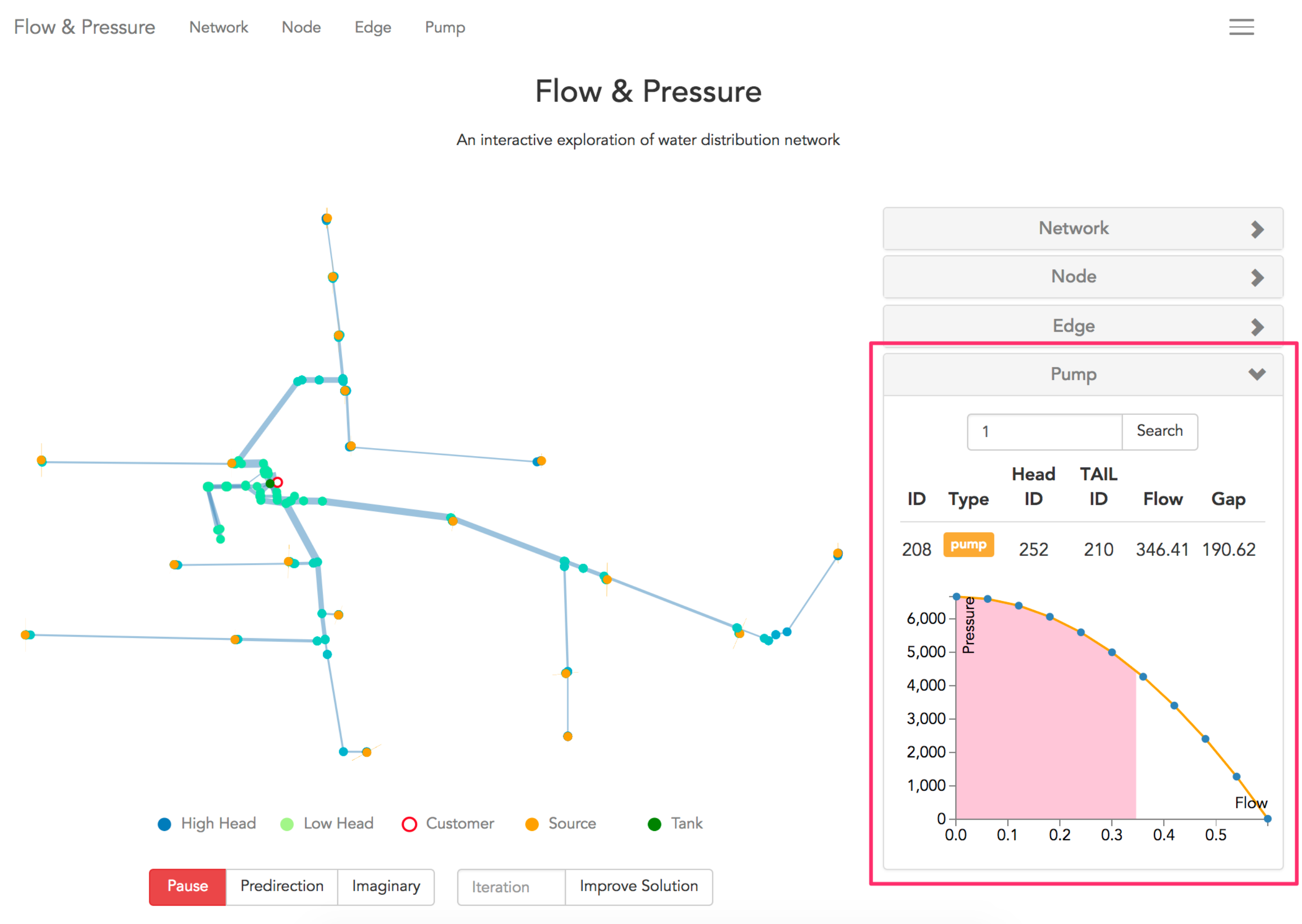
Edge

- **Valves** - shows information about valve edges
- **Edges** - shows information about edges
- **Lowest Flow** - shows information of the top 5 lowest flow edges
- **Highest Flow** - shows information of the top 5 highest flow edges
- **Search by Edge id** - shows information of the specified edge



Pump

- **Search by pump id -** shows pump characteristic curve and information of the specified pump



3. Solver Action

- **Play/Pause** - Play or pause the direction of the edges
- **Predirection** - Redirect the network to get the feasible minimum energy flow direction
- **Imaginary** - Solve the convex relaxation problem to find minimum energy flow and pressure
- **Improve Solution** - Improve the existing constraint gap from the pre-solve method to achieve better accuracy by specify the number of iteration to run

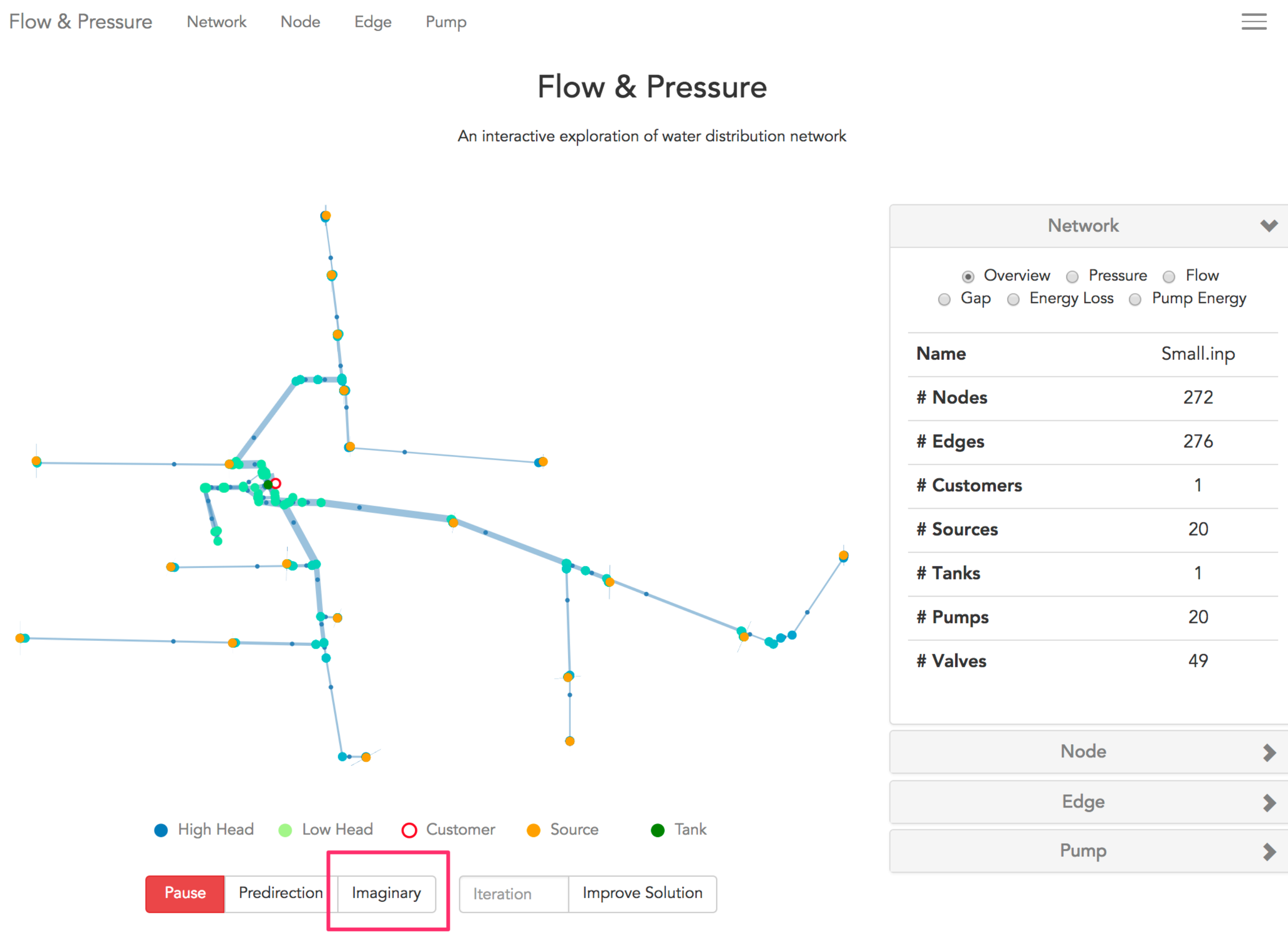
General Usage

1. Start the network visualizer with

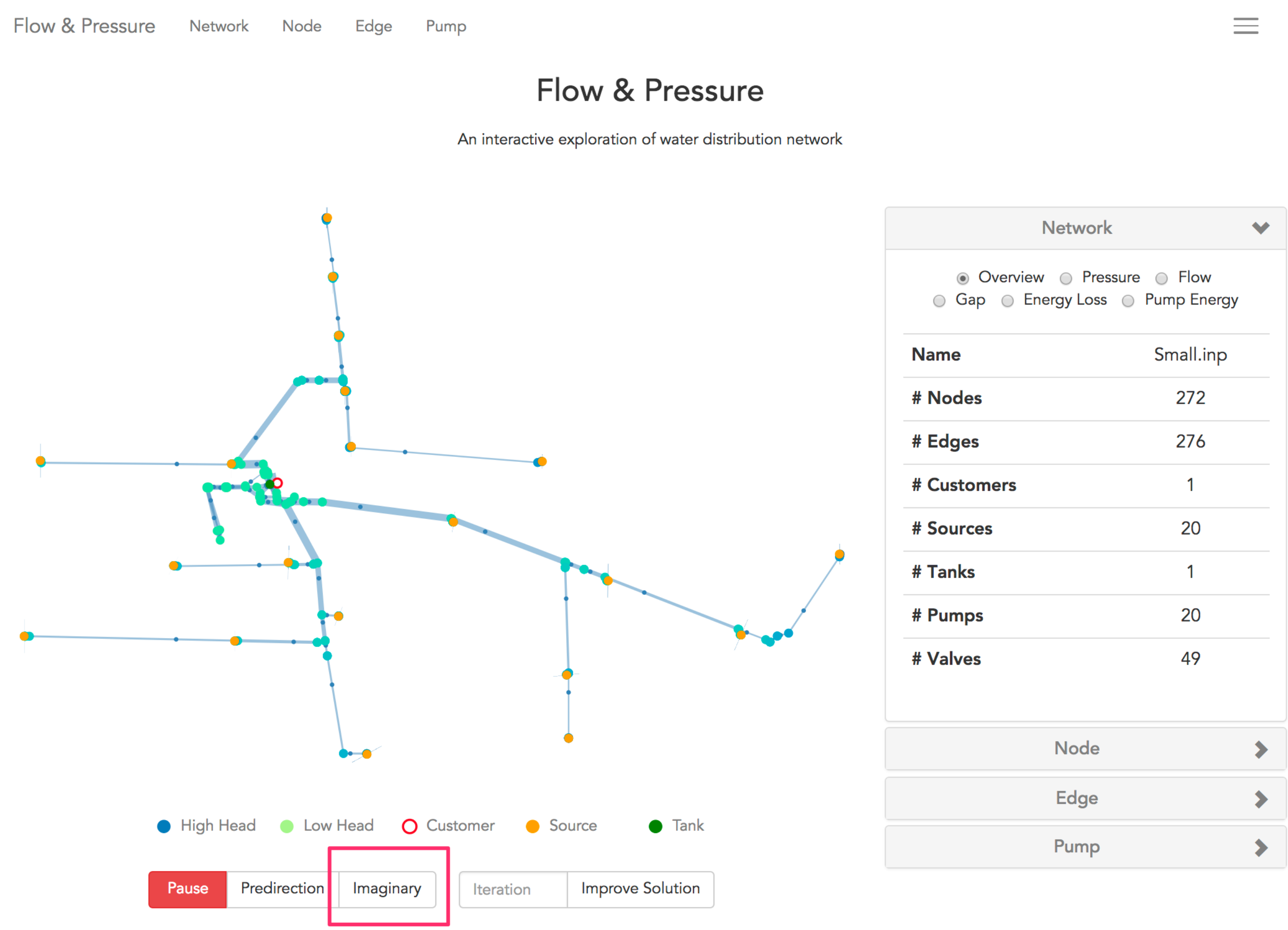
```
python app.py <inp file>
```

1. Predirection the network

1. Solve the convex relaxation problem to find minimum energy flow and pressure



1. Specify the number of iterative improvment to run. Improve the existing constraint gap from the pre-solve method to achieve better accuracy



1. View network information and the solution in the application
2. Alternately, The solution is saved as the numpy object (.npy). You can load them with

```
np.load(path/to/file)
```