For edge disjoint shortest paths, Yen's shortest path and edge disjoint widest paths:

Relevant file:
lightning_routing/simulations/path_based_lp_global/kshortestpaths.py

Example usage:
For computing 4 edge disjoint widest shortest paths on a scale free graph with constant channel capacities run python kshortestpaths.py --graph_type scale_free --credit_type uniform --path_type ksp_yen --max_num_paths 4

Computed paths are stored in lightning_routing/simulations/path_based_lp_global/paths folder.

Command line arguments and options:
graph_type: scale_free, small_world, txt
credit_type: uniform, random, txt
path_type: ksp_yen, ksp_edge_disjoint, kwp_edge_disjoint
max_num_paths: any positive integer; maximum number of paths to consider between each source, destination pair

txt here indicates the topology and channel capacities are to be read from a .txt file.

Quirks (to be cleaned up later):
1. If graph_type is set to txt, then the relevant .txt filepath must be entered in line 221 of kshortestpaths.py.
2. If graph_type is set to either scale_free or small_world, then the graph size must be entered in line 194 of kshortestpaths.py.

For oblivious routing paths:

Prerequisite:
Install Yates from https://cornell-netlab.github.io/yates/installation/#source.

Relevant files:
lightning_routing/simulations/oblivious_routing/get_yates_inputs.py
lightning_routing/simulations/oblivious_routing/process_yates_outputs.py

Example usage:
To compute oblivious paths on a lnd graph (given by a .txt file) do the following steps:
1. Run python get_yates_inputs.py --graph_type txt --credit_type txt. This creates three files lightning.dot, lightning.hosts and lightning.txt in the oblivious_routing directory.

2. Next do, scp lightning.dot ~/yates/data/topologies/, and scp lightning.hosts ~/yates/data/hosts/, and scp lightning.txt ~/yates/data/demands/actual/, and scp lightning.txt ~/yates/data/demands/predicted/.
3. From the ~/yates folder, run yates data/topologies/lightning.dot data/demands/actual/lightning.txt data/demands/predicted/lightning.txt data/hosts/lightning.hosts -raeke -budget 4, where budget denotes the number of paths.
4. Now scp ~/yates/data/results/lightning/paths/raeke_0 ~/lightning_routing/simulations/oblivious_routing.
5. Lastly from ~/lightning_routing/simulations/oblivious_routing directory run python process_yates_outputs.py --graph_size 102.

Command line arguments and options for get_yates_inputs.py:
graph_type: scale_free, small_world, txt
credit_type: uniform, random, txt

Command line arguments and options for process_yates_outputs.py:
graph_size: positive integer

Quirks (to be cleaned up later):
1. If graph_type is scale_free or small_world, the graph size must be entered in line 50 of get_yates_inputs.py.
2. If graph_type is txt, then the relevant .txt filepath must be entered in line 76 of get_yates_inputs.py.
3. The filename of the paths file copied from ~/yates/data/results/lightning/paths/ (e.g., raeke_0) must be entered in line 48 of process_yates_outputs.py.