Development of Topics in Electrical Engineering 1 - "Learning to Route" Restoring Paper Results

Report

Written by:

Ido Yehezkel 204397368

Table of Contents

Introduction

This report is about restoring the achieved results from the paper: *Valadarsky, A., Schapira, M., Shahaf, D., & Tamar, A. (2017, November). Learning to route*.
In *Proceedings of the 16th ACM workshop on hot topics in networks* (pp. 185-191).

The paper's writers interduce a new approach of how to use machine learning techniques in order to solve one of the fundamental network control problem, traffic routing.
Two approaches were evaluated: traffic patterns predictions based on the history traffic using supervised learning and solving the routing control problem directly using reinforcement learning agent.
The supervised learning models is a good start point but didn't achieve noticeable results, on the other hand, the reinforcement learning techniques achieved much better results and create new direction of research.

This report is focusing on restoring the presented results and conclusions using reinforcement learning techniques. The motivation is getting new knowledge about routing algorithms (Optimal Routing, Oblivious Routing), new heuristics methods, useful programming packages, but the main goal is to develop a new direction for a continue research.

Definition of the Problem

Network topology- following the paper, for the control routing problem the network topology is represent by directional graph which is based on unidirectional graph with edges' capacity function (in Mb/s).
The directional graph is simply made by considering each edge as independent bidirectional link.

Synthetic Traffic Generation – to examine the developed techniques a synthetic traffic generation is necessary.
The traffic demands is represent by square matrix where the cell i, j is the flow demand from node i to node j .
Another property of the matrix is sparsity, which is the percentage of source, destination pairs that include traffic, for example, a topology with 10 nodes includes 90 difference pairs of source destination, with sparsity of 30% only 27 randomly chosen pairs are included in the traffic.
Two different types of traffic are generated for evaluation:

- **Gravity Traffic**: traffic with correlation to the links' capacities connected to the source node and the destination node, calculated by the formula:
$$\frac{\left[\text{source out links' capacity}\right]\times\left[\text{destination out links' capacity}\right]}{\text{total nodes out links' capacity}}$$
- **Bimodal Traffic:** each flow demand of the traffic is sampled by some probability (biased coin flip) from two independent gaussian distributions. One distribution represents mice flows and the other elephants' flows.
For all the evaluations the gaussians distributions are: $N\left(400,20\right)$ for elephant flows and $N\left(150,20\right)$ for mice flows.

The Goal: finding routing scheme for the traffic demands with the objective of minimization of maximum link utilization, or the minimax problem, also referred to as load balancing flows in the network.

Baselines - Optimal Routing

To evaluate the models and techniques that were developed by the paper's writers they define a reference baseline which is based on **load balancing**.

The optimal routing criteria is defined as the **minimization of maximum link utilization**, minimize the most congested link, the mathematic expression is:

$$\min_{e\in E}\left\{\max\frac{f_e}{C_e}\right\}$$

C_e – capacity of link e

f_e – total flow in link e

These criteria can be formulated as an optimization problem as follow:

$$Objective: Minimize\ r$$

$$\sum_i\sum_{j\neq i} g_e(i,j) \le C_e \cdot r \quad \forall \mathrm{e} \in Edges$$

where $g_e(i,j)$ is a fraction of demened $i \to j$ flows on link e

The routing scheme constartions:

For each existing demend $i \to j$:

$$\sum_{e\in IN(v)} g_e(i,v) - \sum_{e\in OUT(v)} g_e(i,v) = demend(i,v) \Big| v = j, \text{destination constraint}$$

$$\sum_{e\in OUT(v)} g_e(v,j) - \sum_{e\in IN(v)} g_e(v,j) = demend(v,j) \Big| v = i, \text{source constraint}$$

$$\sum_{e\in OUT(v)} g_e(i,j) - \sum_{e\in IN(v)} g_e(i,j) = 0 \Big| v \neq i, j, \text{transit constraint}$$

$$g_e(i,j) \ge 0 \quad \forall i,j$$

$$r \ge 0$$

As this optimization problem define, all the constraints are linear expressions therefore, a linear programing solver (like "IBM – CPLEX" or "Gurobi") can be used to solve it, ("Gurobi" had been used, because it easy to set multiple objectives much easier to prevent the tool create unnecessary flows in loops).

The linear programming problem includes $O\left(|edges| \times |nodes|^2\right)$ variables and $O\left(|edges| + |nodes|^2\right)$ constrains.

Baseline - Optimal Oblivious Routing

Another baseline the writers used is **optimal oblivious routing**. As its name implies the oblivious routing **is not traffic patterns depended but only a topology depended**, this routing technique was represented in several papers in early 2000s.

The oblivious performance ratio of routing scheme f is define as follows:

$$CONGESTION(f, D) = \max_{e \in E} \frac{Flow(e, f, D)}{C_e}$$

Most congestion edge w.r.t routing scheme f, demend D and capacity C.

$$OBLIV - PERF - RATIO(f) = \sup_D \frac{CONGESTION(f, D)}{OPT(D)}$$

Worst demends matrix congestion w.r.t routing scheme f normalized by optimal routing congestion.

$$OBLIVE - OPT(G) = \min_f OBLIV - PERF - RATIO(f)$$

Best routing scheme f of topology G regrdless the traffic demends

Using the result of the paper *Applegate, D., & Cohen, E. (2006). Making routing robust to changing traffic demands: algorithms and evaluation. IEEE/ACM Transactions on Networking, 14(6), 1193-1206,* the optimal oblivious routing problem can be formulated as a single optimization problem as follow:

$Objective: Minimize\ r$

f is valid routing scheme

$\forall edges\ e:$

$$\sum_{h \in E} C_h \cdot \pi_e(h) \le r$$

$$\forall \text{ pairs } i \to j: \ \frac{f_e(i, j)}{C_e} \le p_e(i, j)$$

$\forall$ node i, $\forall$ edge a=(j,k):

$$\pi_e(a) + p_e(i, j) - p_e(i, k) \ge 0$$

$\forall$ edge h $\in$ E : $\pi_e(h) \ge 0$

$\forall$ node i: $p_e(i, i) = 0$

$\forall$ node i, j: $p_e(i, j) \ge 0$

As one can see, because all the constrains are linear, this also a linear programming problem with $O\left(|edges| \times |nodes|^2\right)$ variables and $O\left(|edges|^2 \times |nodes|\right)$ constrains, where $\pi_e(h)$ is a variable that represent an exist weight that for every pair of edges $e, h: \sum_{h \in E} C_h \cdot \pi_e(h) \le r$ and $p_e(i, j)$ represent the length of the shortest path from node i to node j according the edge weights $\pi_e(h)$.

Restoring the Baseline Results

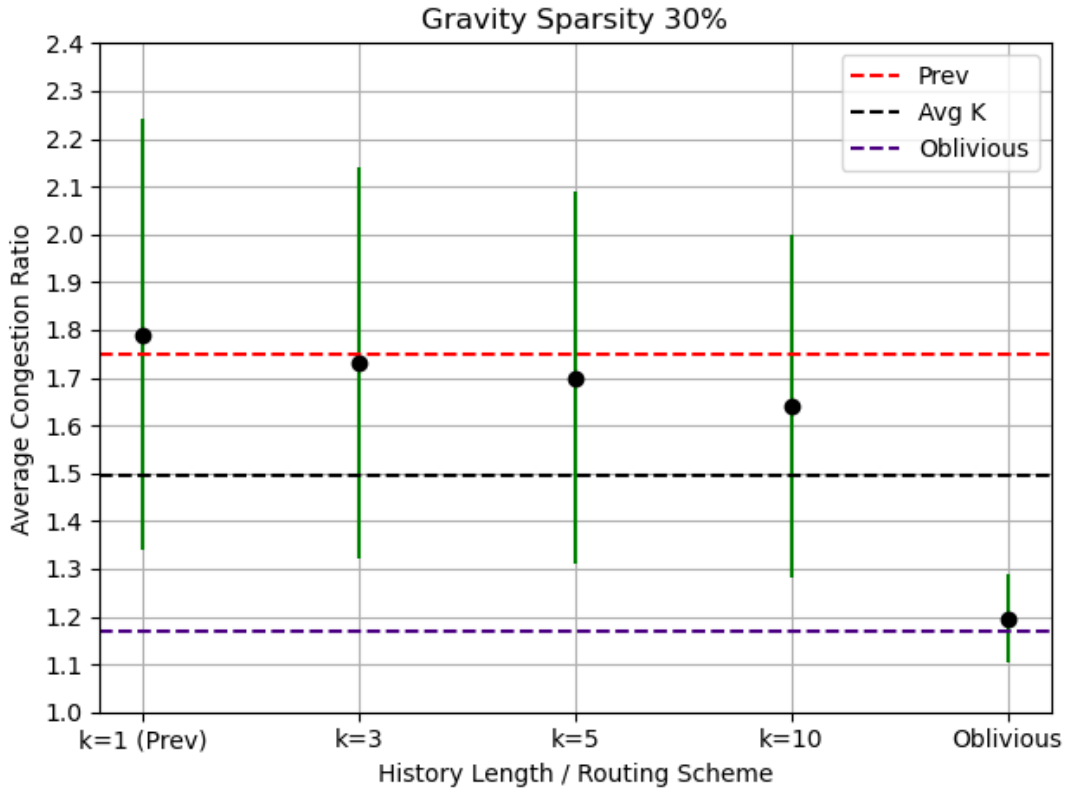
Before restoring the results of the reinforcement learning agent all the baselines should be restored.
Because the agent is taking a decision by observing the traffic history the writers created a similar reference baseline that also use traffic history, by observing the last K matrices and calculate an average traffic matrix and route the next new traffic matrix by the optimal routing scheme of the average one (create the routing scheme in advance, before new traffic arrives).
One of the critic challenges needed to be considered is that it can be happens that a flow exists in the current routed new matrix but not in the average one (based on history, this is more common in low matrix sparsity), the solution is to use an ECMP policy with equal weights, so those first appear flows are equally divided between all shortest paths between the source and destination.
The result are normalized with the most congested link utilization when applying the optimal routing scheme.

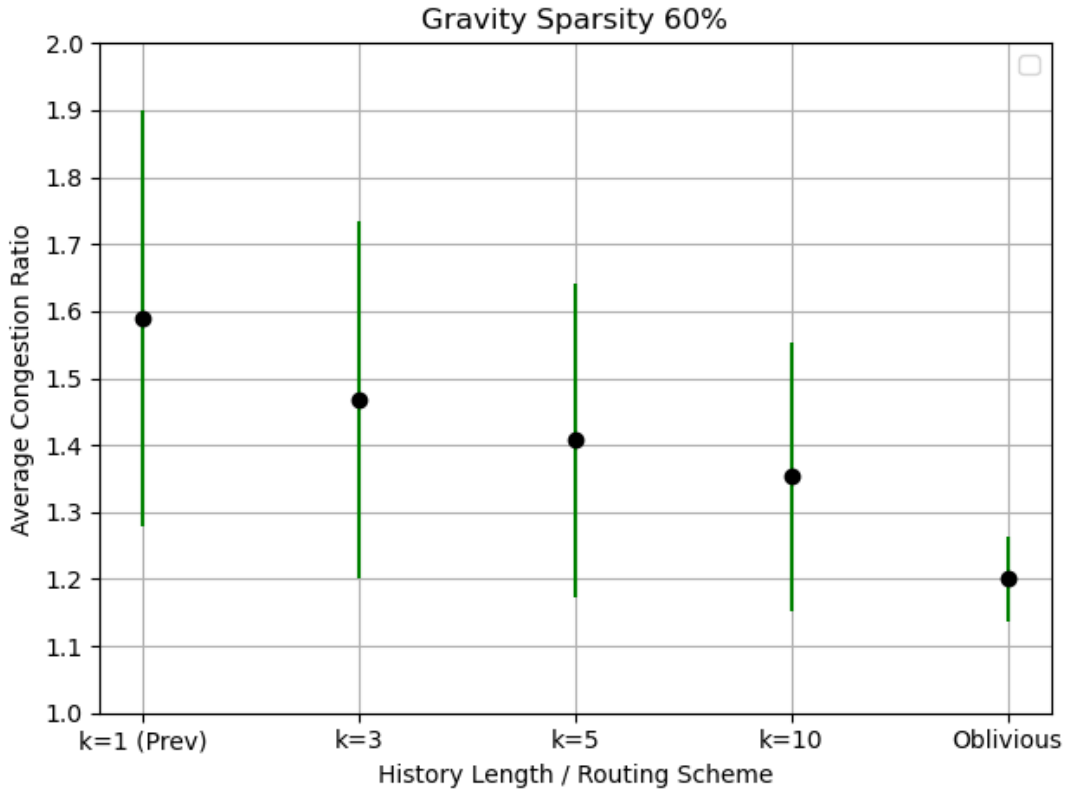
Evaluation

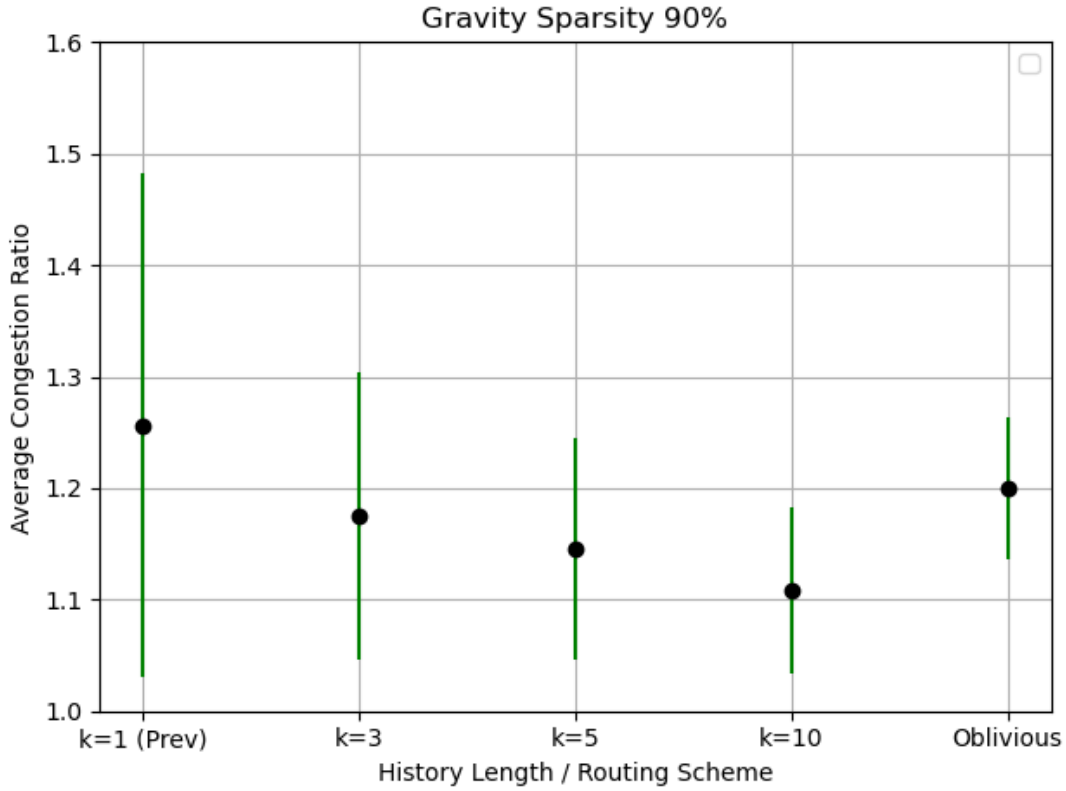
Similar to the paper, a 12-node topology with 26 edges (the original topology includes duplicate edges, this represented as double the capacity for those edges) and constant link capacity of 10,000 Mb.
20,000 traffic matrices dataset had been used in order to get the results, for example, for K=5 matrices 1,2,3,4,5 are used for calculate an average matrix then finding the optimal routing scheme for it and apply it on matrix 6.

(The dashed lines are approximations of the results from the paper, the vertical green lines are the standard deviation).

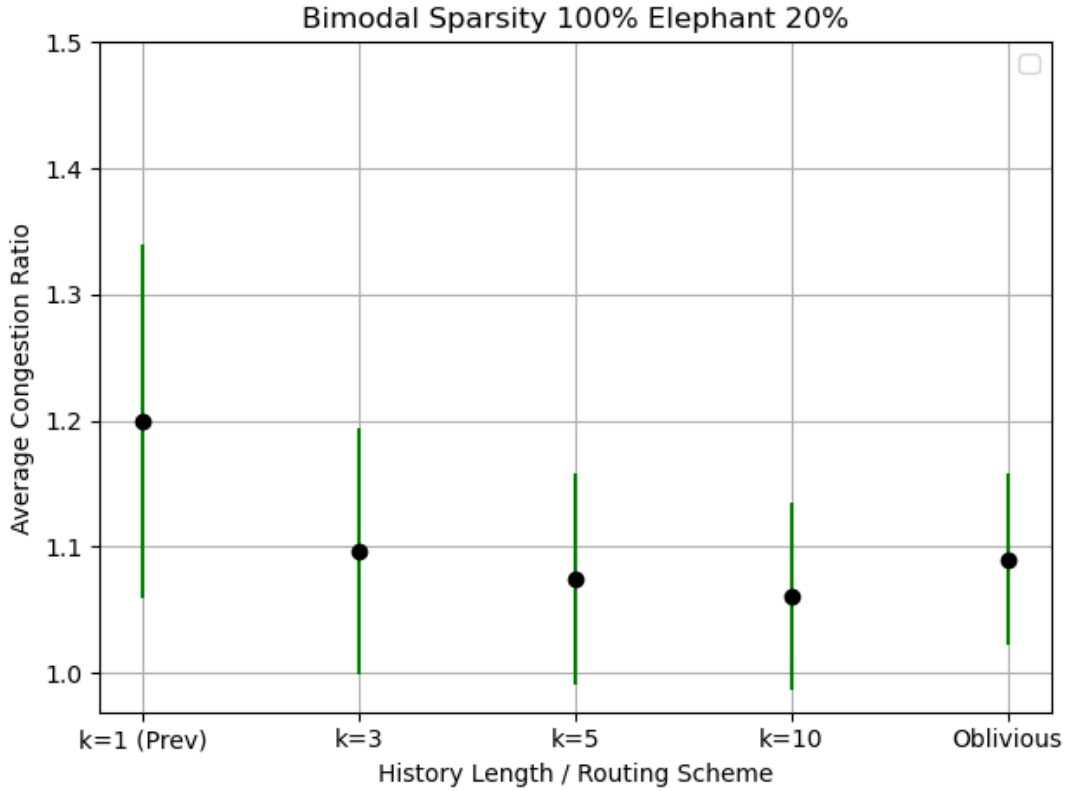
Gravity Traffic:

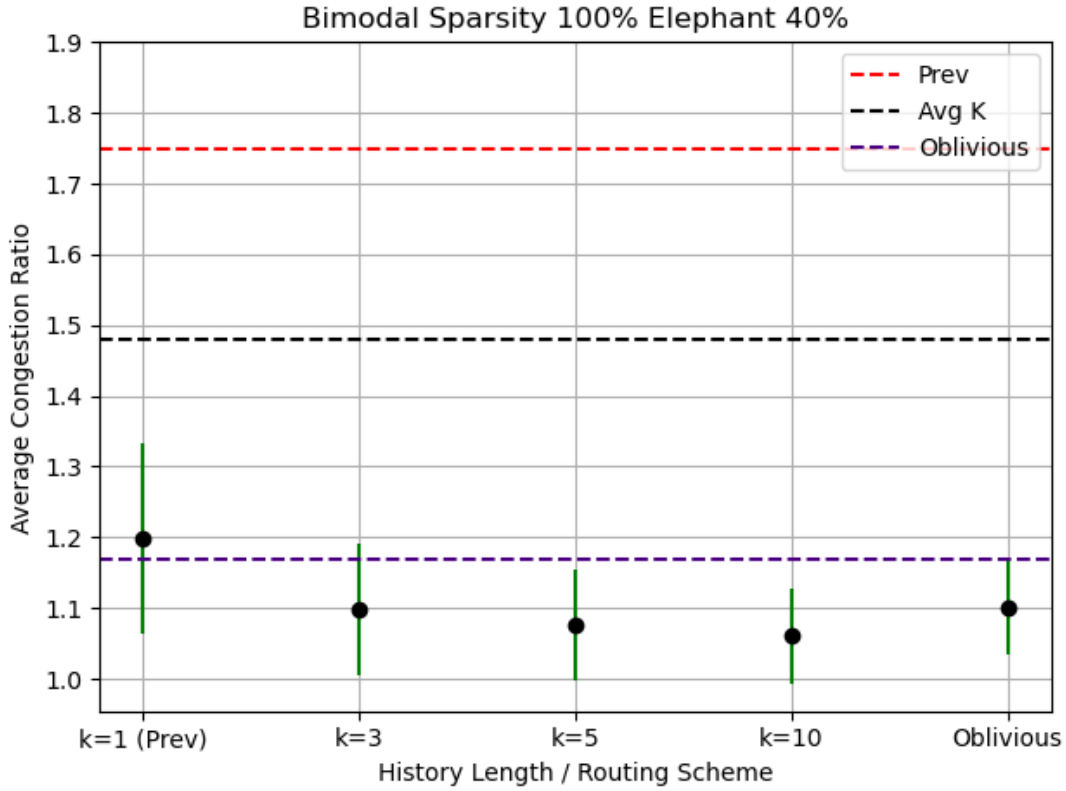


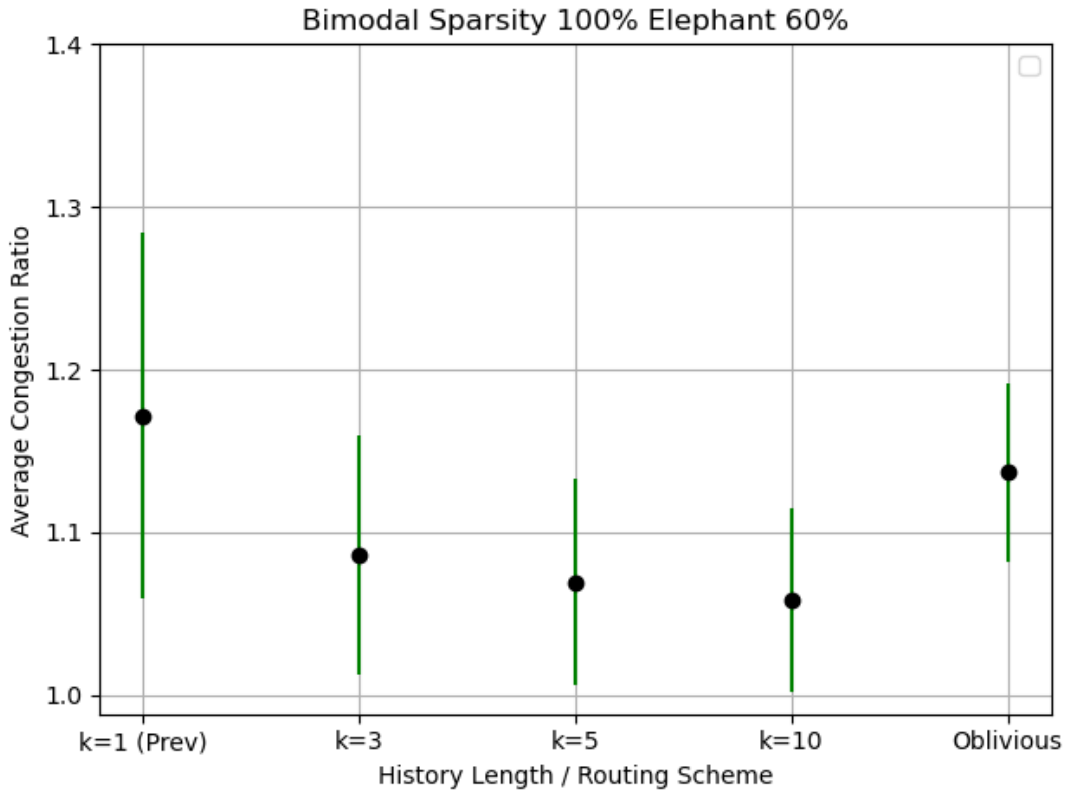




Bimodal Traffic:




Bimodal Sparsity 100% Elephant 40%
Prev
Avg K
Oblivious
Average Congestion Ratio
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1.0
k=1 (Prev)
k=3
k=5
k=10
Oblivious
History Length / Routing Scheme


Bimodal Sparsity 100% Elephant 60%
Average Congestion Ratio
1.4
1.3
1.2
1.1
1.0
k=1 (Prev)
k=3
k=5
k=10
Oblivious
History Length / Routing Scheme

Restoring the Reinforcement Learning Results

The outstanding result of the paper is the prove of concept that an RL agent can be useful to produce good routing scheme that minimize the congestion ratio.

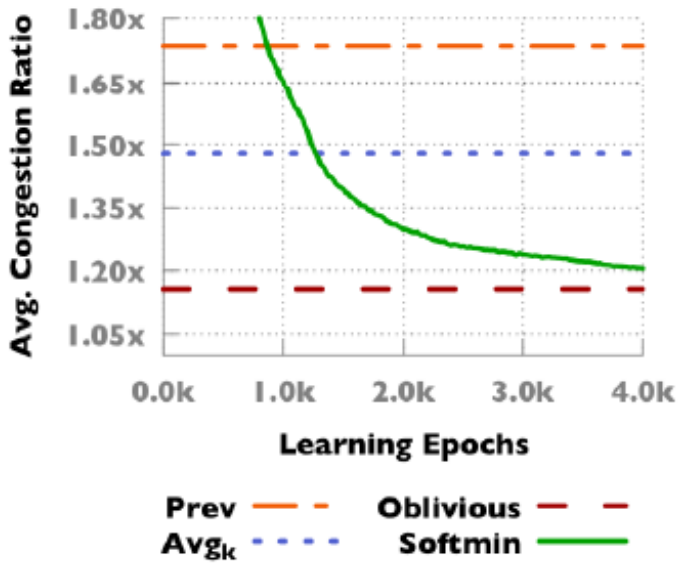
The writers set up an environment that simulate the network nodes and links and every timestep a new traffic matrix with new demands is routed by the agent.

The goal of the process of learning is the agent learns a **map** from a **history of traffic matrices to weights** for each network link which used for routed the future traffic matrix. Using those weights, the environment calculates shortest path to each destination of demands from every other node. Finally, plug these paths' costs and each link weight to Soft-Min function to calculate for each node the percentage of flow carried by each leaving link. The final step is to run each demand of flow around the network links until all of it (99.99%) reaches to its destination and calculate the most congested edge for the final congestion ratio of the traffic matrix.

Evaluation

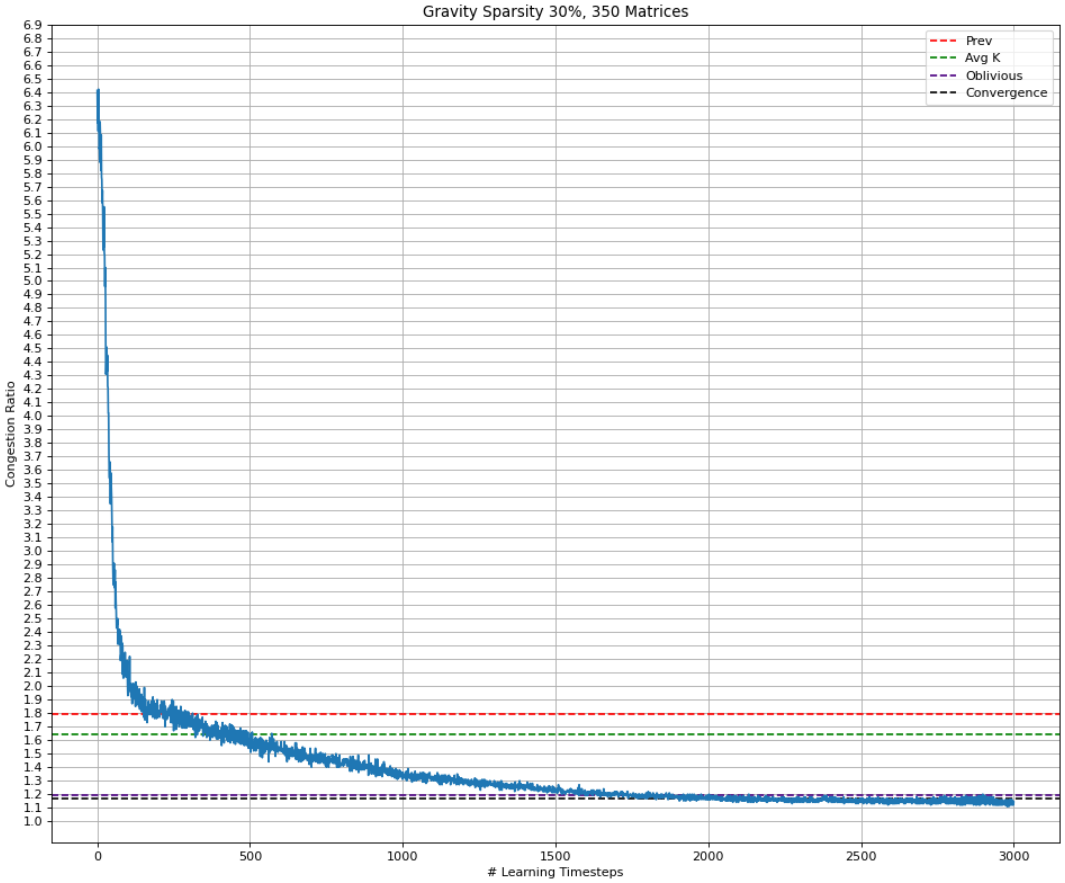
Similar to the paper, a 12-node topology with 26 edges (the original topology includes duplicate edges, this represented as double the capacity for those edges) and constant link capacity of 10,000 Mb.
The evaluated agent parameters are: history length of 10 (as paper), discount factor equal to 0 and episode length of 1. Because flows are not continued more than one timestep (demand flow can't start in time t and continue to time $t+1$), therefore the assumption that a myopic approach to minimize the congestion of the current traffic matrix is a good start because current agent decisions has no effect on future ones.

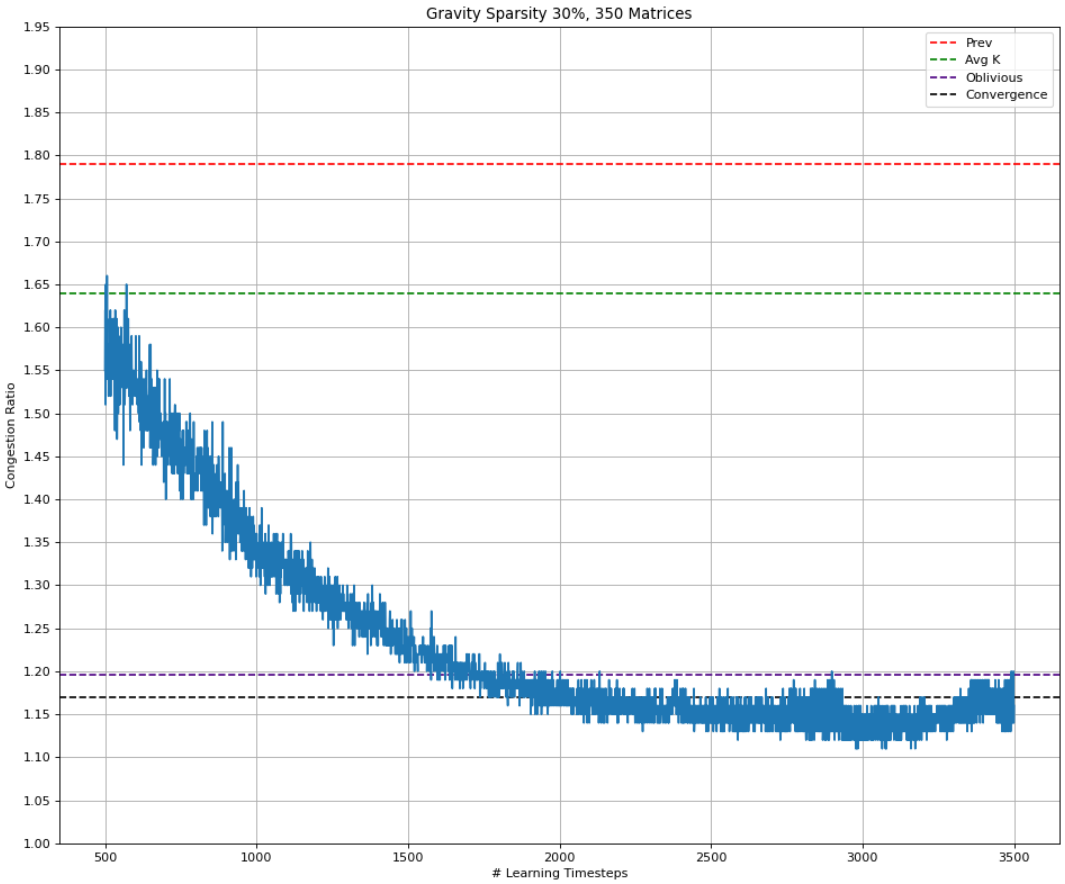
Gravity Traffic Paper results:



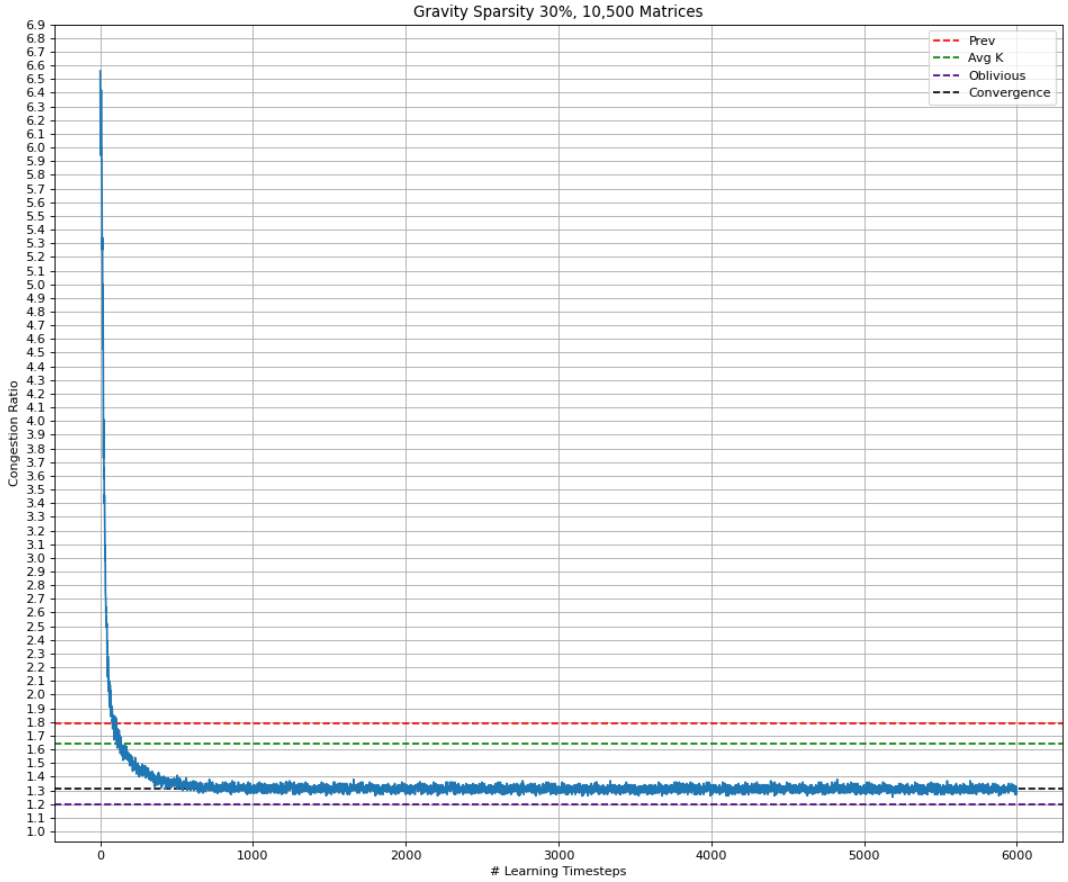
(a) *Congestion ratio for sparse ($p = 0.3$) gravity DM sequences*

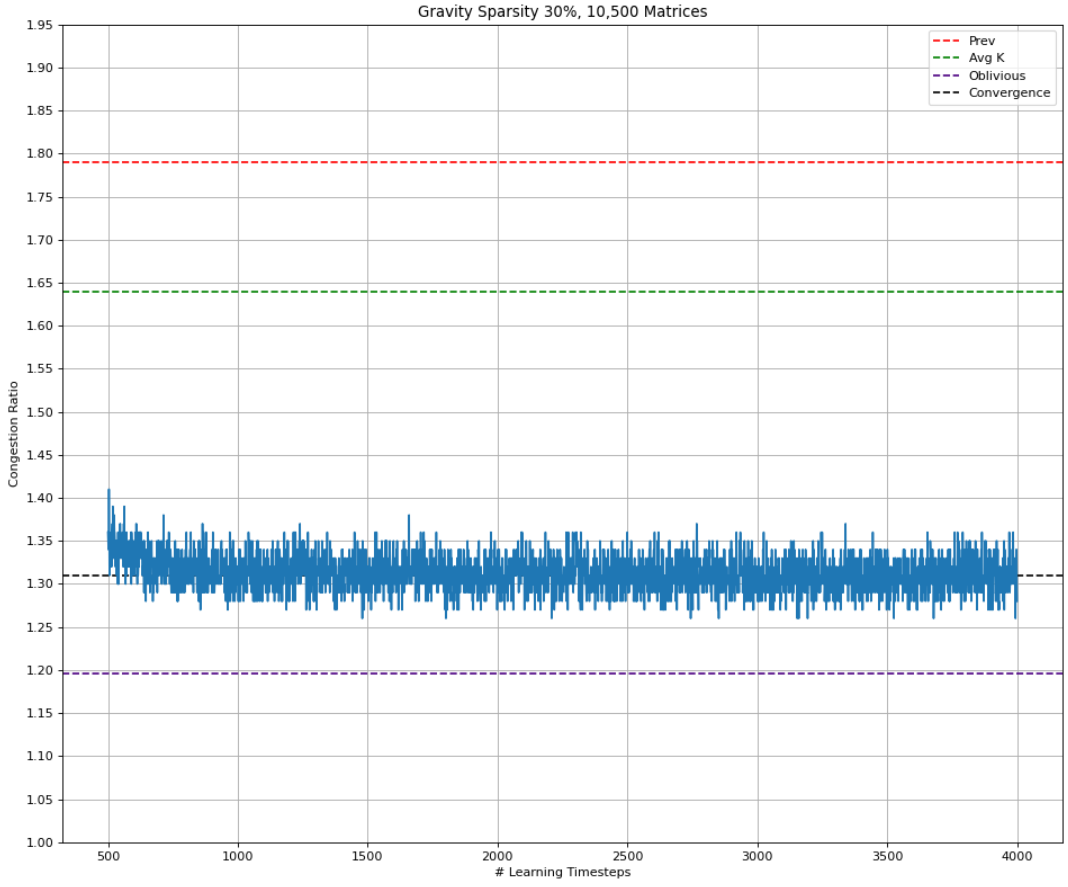
Gravity Traffic with 350 different TMs:



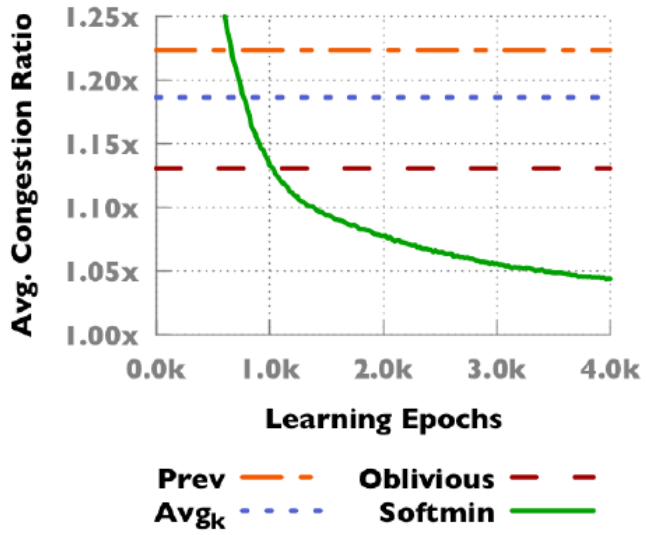

Gravity Sparsity 30%, 350 Matrices
Prev
Avg K
Oblivious
Convergence
Congestion Ratio
Learning Timesteps
1.95
1.90
1.85
1.80
1.75
1.70
1.65
1.60
1.55
1.50
1.45
1.40
1.35
1.30
1.25
1.20
1.15
1.10
1.05
1.00
500
1000
1500
2000
2500
3000
3500

Gravity Traffic with 10,500 different TMs:



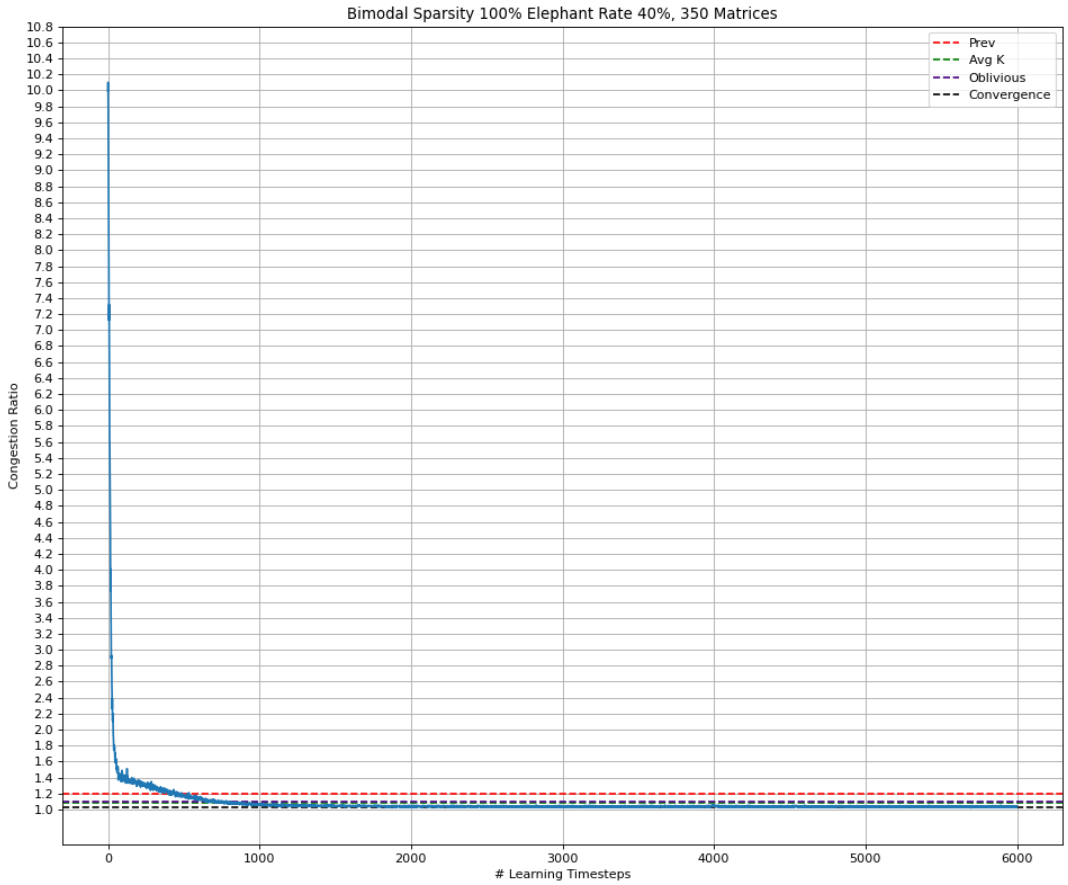


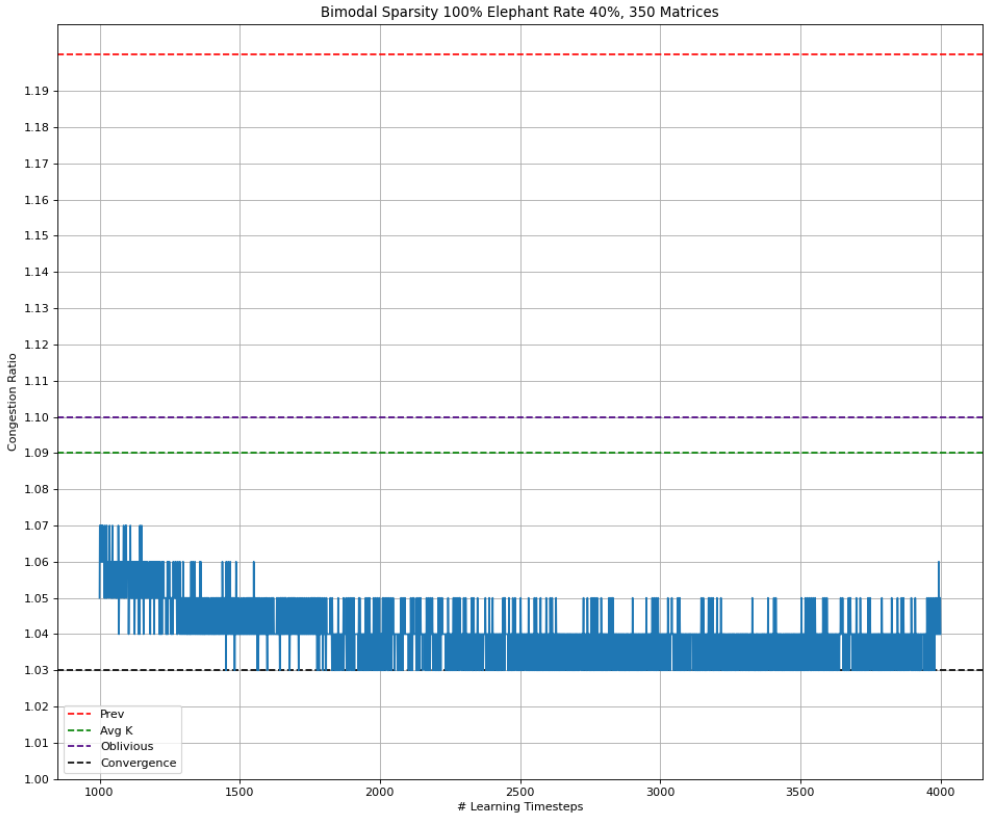
Bimodal Traffic Paper results:



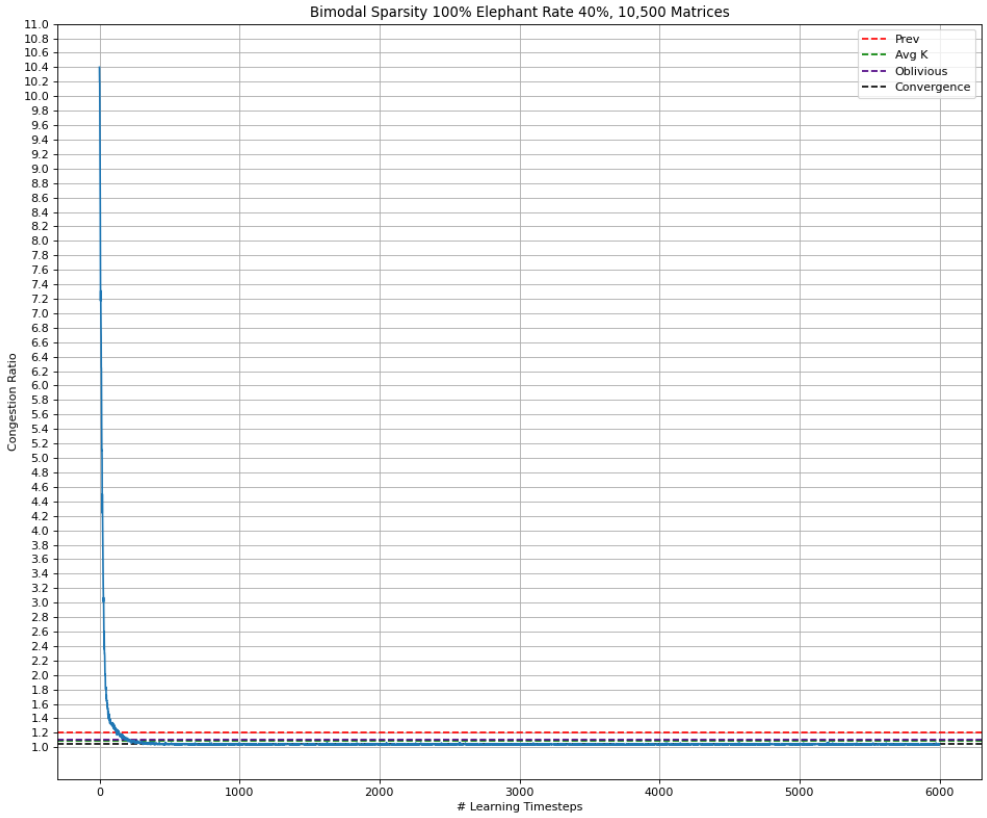
(b) *Congestion ratio for non-sparse* ($p = 1.0$) *bimodal DM sequences with 40% elephant flows*

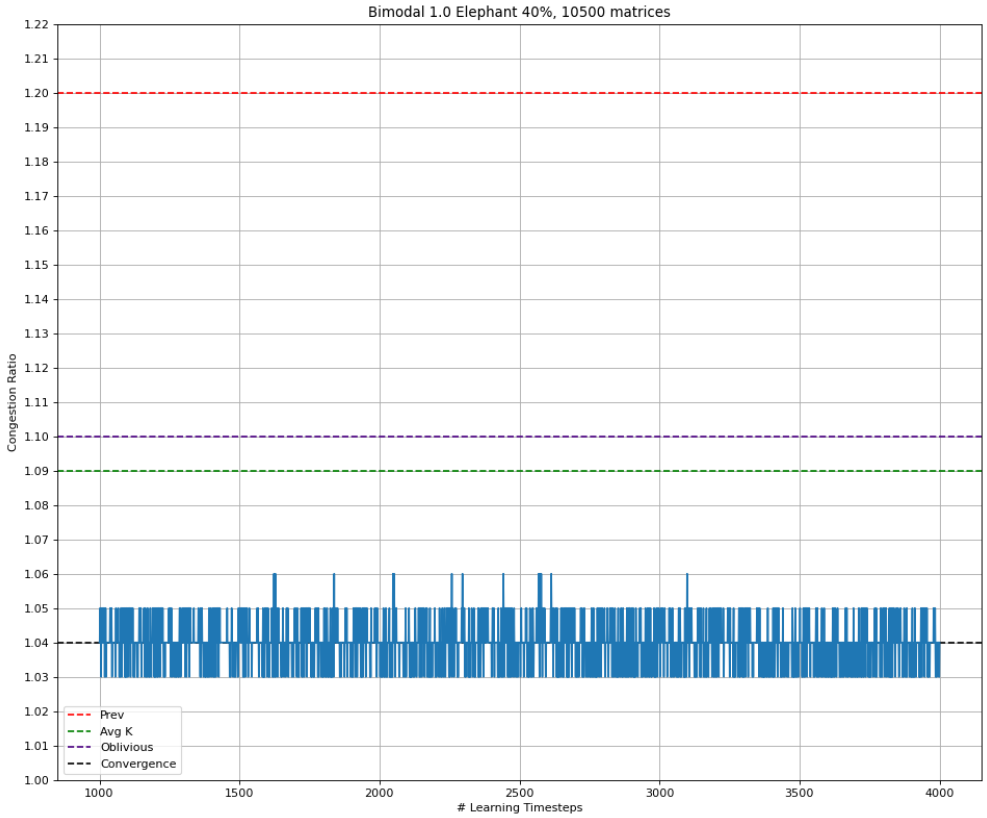
Bimodal Traffic with 350 different TMs:




Bimodal Sparsity 100% Elephant Rate 40%, 350 Matrices
Congestion Ratio
Learning Timesteps
Prev
Avg K
Oblivious
Convergence

Bimodal Traffic with 10,500 different TMs:





Conclusions and Summary

First, I must say that restoring the results of the paper help me, as a young researcher, to understand much better the algorithms and methods of evaluation I am about to use in my own research. Moreover, the practical experience with the Python packages for this task help me to understand better how to write a RL environment and how to configure an agent for example what are the hyperparameters.

While restoring the results and after examining the outputs figures I notice some conclusions that may be helpful to next research directions.
As seen in the baselines results, the oblivious routing scheme is becoming less effective when the sparsity of the matrix is higher, I think the reason for that is because with higher sparsity the average matrix includes more relevant flows that appears in the future traffic demands, make the average one a better approximation of the future one.
The way of routing first seen flows with ECMP can be done in more sophisticated way, for example each link weight is $\frac{1}{Capacity}$, the motivation is to preferred paths with high link capacities.

When examine the bimodal traffic evaluations specifically, the effectiveness of the percentage of an elephants flows on the congestion is low (the reason may be because the means of the gaussians distributions for mice and elephant should be difference with factor scale, like 10).

The reinforcement learning agent achieves results like presented in the paper and it seems the convergence is faster; the reason may be because the change in the learning algorithm from TRPO to PPO which is newer.

The same trends the paper shows can be seen also here, for gravity with low sparsity (30%) the oblivious is still better than the agent, but for the bimodal traffic the agent beats all the baselines. The reasons for better performance in bimodal traffic are the high sparsity of 100%, the agent learning from all pairs all the time (something less to learn) and because we set low standard deviation for the distributions the values of demands are close to each other's. another evidence for that statement is that there is no change in the performance when number of different traffic matrices to be learned from is increasing.

To summarize, I enjoy examining and restoring the results from the paper, that helped me a lot to start get some knowledge about the topics my research is about to be. Also, I would like to thank Asaf Valadarsky (one of the paper writers) that help me and contribute parts of his code.

References

- Valadarsky, A., Schapira, M., Shahaf, D., & Tamar, A. (2017, November). Learning to route. In Proceedings of the 16th ACM workshop on hot topics in networks (pp. 185-191).
- Applegate, D., & Cohen, E. (2006). Making routing robust to changing traffic demands: algorithms and evaluation. IEEE/ACM Transactions on Networking, 14(6), 1193-1206.
- Azar, Y., Cohen, E., Fiat, A., Kaplan, H., & Räcke, H. (2004). Optimal oblivious routing in polynomial time. Journal of Computer and System Sciences, 69(3), 383-394.
- Roughan, Matthew, et al. "Experience in measuring internet backbone traffic variability: Models metrics, measurements and meaning." Teletraffic Science and Engineering. Vol. 5. Elsevier, 2003. 379-388.