2018 21st International Conference on Intelligent Transportation Systems (ITSC)
Maui, Hawaii, USA, November 4-7, 2018

Enhanced Delta-tolling: Traffic Optimization via Policy Gradient Reinforcement Learning

Hamid Mirzaei*, Guni Sharon†, Stephen Boyles‡, Tony Givargis* and Peter Stone§
*Department of Computer Science, University of California Irvine, Irvine, CA 92617, USA
†Department of Computer Science & Engineering, Texas A&M University, College Station, TX 77843, USA
‡Civil, Architectural and Environmental Engineering, The University of Texas at Austin, Austin, TX 78712, USA
§Department of Computer Science, University of Texas at Austin, Austin, TX 78712, USA
mirzaeib@uci.edu, gunisharon@gmail.com, sboyles@mail.utexas.edu, givarigs@uci.edu, pstone@cs.utexas.edu

***Abstract*—In the micro-tolling paradigm, a centralized system manager sets different toll values for each link in a given traffic network with the objective of optimizing the system's performance. A recently proposed micro-tolling scheme, denoted Δ-*tolling*, was shown to yield up to 32% reduction in total travel time when compared to a no-toll scheme. Δ-*tolling*, computes a toll value for each link in a given network based on two global parameters: β which is a proportional parameter and R which controls the rate of toll change over time. In this paper, we propose to generalize Δ-*tolling* such that it would consider different R and β parameters for each link. a policy gradient reinforcement learning algorithm is used in order to tune this high-dimensional optimization problem. The results show that such a variant of Δ-*tolling* far surpasses the original Δ-*tolling* scheme, yielding up to 38% reduced system travel time compared to the original Δ-*tolling* scheme.**

I. Introduction

Advancements in connected and automated vehicle technology present many opportunities for highly optimized traffic management mechanisms [1]. One such mechanism, *micro-tolling*, has been the focus of a line of recently presented studies [2, 3, 4]. In the micro-tolling paradigm, tolls can be charged on many or all network links, and changed frequently in response to real-time observations of traffic conditions. Toll values and traffic conditions can then be communicated to vehicles which might change routes in response, either autonomously, or by updating directions given to the human driver. A centralized system manager is assumed to set toll values with the objective of optimizing the traffic flow. Many methods for computing such tolls were presented over the last century most of which made very specific assumptions regarding the underlying traffic model. For instance, assuming that demand is known or fixed [5], assuming that links' capacity is known or fixed, assuming that the user's value of time (VOT) is homogeneous [6], assuming traffic follows specific latency functions [7], or assuming traffic patterns emerge instantaneously [8].

A recent line of work [2, 3] suggested a new tolling scheme denoted Δ-*tolling*. Unlike previous tolling schemes, Δ-*tolling* makes no assumptions regarding the demand, links' capacity, users' VOT, and specific traffic formation models. Δ-*tolling* sets a toll for each link equal to the difference (denoted Δ) between its current travel time and free flow travel time multiplied by a proportionality parameter β. The rate of change in toll values between successive time steps is controlled by another parameter R. Despite being extremely simple to calculate, Δ-*tolling* was shown to yield optimal system performance under the stylized assumptions of a macroscopic traffic model using the Bureau of Public Roads (BPR) type latency functions [9]. Moreover, Δ-*tolling* presented significant improvement in total travel time and social welfare across markedly different traffic models and assumptions. In fact, the simple working principle of Δ-*tolling* is what allows it to act as a model-free mechanism. Whereas the original Δ-*tolling* algorithm required a single β and R parameter for the entire network, the main contribution of this paper is a generalization of Δ-*tolling* to accommodate separate parameter settings for each link in the network. While conceptually straightforward, we demonstrate that doing so enables significant performance improvements in realistic traffic networks.

The increased representational power of Enhanced Δ-*tolling* compared to Δ-*tolling* does come at the cost of necessitating that many more parameters be tuned. A secondary contribution of this paper is a demonstration that policy gradient reinforcement learning methods can be leveraged to set tune these parameters effectively. Our detailed empirical study in Section V validates our claim that Enhanced Δ-*tolling* has the potential to improve upon the already impressive results of Δ-*tolling* when it comes to incentivizing self-interested agents to coordinate towards socially optimal traffic flows.

II. Problem definition and terminology

We consider a scenario where a set of agents must be routed across a traffic network given as a directed graph, $G(V, E)$. Each agent a is affiliated with a source node, $s_a \in V$, a target node, $t_a \in V$, a departure time, d_a, and a VOT, c_a (the agent's monetary value for a delay of one unit of time).

Agents are assumed to be self-interested and, hence, follow the least cost path leading from s_a to t_a. The cost of a path, p, for an agent, a, is a function of the path's latency, l_p, and tolls along it, τ_p. Formally, $cost(p, a) = l_p \cdot c_a + \tau_p$. The value of

978-1-7281-0322-8/18/$31.00 ©2018 IEEE

time, c_a, is assumed to be constant per agent. Although this assumption might not hold in real-world, it follows common practice in the transportation literature [3, 10, 11].

Since traffic is dynamically evolving, travel times and toll values might change over time, agents are assumed to continually re-optimize their chosen route. As a result, an agent might change its planned route at every node along its path. Each link in the network, $e \in E$, is affiliated with a dynamically changing toll value τ_e where for any path, p, $\tau_p = \sum_{e \in p} \tau_e$. Moreover, each link is affiliated with a latency l_e representing the travel time on link e. Similar to τ_e, l_e is dynamically changing as a function of the traffic state.

The objective of the system manager is to assign tolls such that if each agent maximizes its own self interest, the system behavior will maximize social welfare. Denoting the latency suffered by agent a as l_a, social welfare is defined as $\sum_a l_a \cdot c_a$.[1] The system manager addresses the micro-tolling assignment problem which is defined as follows.

Given: L^i - the vector of links' latencies at time step i.
Output: τ^{i+1} - the vector of tolls applied to each link at the next time step.
Objective: Optimize social welfare.
Assumption: Agents are self interested i.e., they travel the least cost path ($\arg\min_p\{cost(p, a)\}$) leading to their assigned destination (t_a).

III. Background and Related work

The approach suggested in this paper for solving the micro-tolling assignment problem builds on two previously presented algorithms: Δ-tolling, and Finite Difference policy Gradient Reinforcement Learning (RL).

A. Delta-tolling

It is well known that charging each agent an amount equivalent to the cost it inflicts on all other agents, also known as *marginal-cost tolling*, results in optimal social welfare [7].

Applying a marginal-cost tolling scheme, when differentiable latency functions are not assumed, requires knowing in advance the marginal delay that each agent will impose on all others. This, in turn, requires knowledge of future demand and roadway capacity conditions, as well as counterfactual knowledge of the network states without each driver.

Δ-*tolling* [2, 3] was recently suggested as a model-free scheme for evaluating marginal cost tolling. It requires observing only the latency (travel time) on each link and makes no assumption on the underlying traffic model. Δ-*tolling* involves charging a toll on each link proportional to its delay (the difference between observed and free-flow travel times). Δ-*tolling* requires tuning of only two parameters: a proportionality constant (β), and a smoothing parameter (R) used to damp transient spikes in toll values.

Algorithm 1 describes the toll value update process of Δ-*tolling*. For each link, Δ-*tolling* first computes the difference (Δ) between its current latency (l_e^i) and its free flow travel time (denoted by T_e). We use i to denote the current time step. Next, the toll for link e at the next time step (τ_e^{i+1}) is updated to be a weighted average of Δ times beta and the current toll value. The weight assigned to each of the two components is governed by the R parameter ($0 < R \leq 1$).

Algorithm 1: Updating tolls according to Δ-*tolling*.

```
1 while true do
2     for each link e ∈ E do
3         Δ ← l_e^i − T_e
4         τ_e^{i+1} ← R(βΔ) + (1 − R)τ_e^i
5     i ← i + 1
```

The R parameter determines the rate in which toll values react to observed traffic conditions. When $R = 1$ the network's tolls respond immediately to changes in traffic on the one hand but leave the system susceptible to oscillation and spikes on the other hand. By contrast, as $R \to 0$ the tolls are stable, but are also unresponsive to changes in traffic conditions.

Sharon et al. [2, 3] showed that the performance of Δ-*tolling* is sensitive to the values of both the R and β parameters. Their empirical study suggests that values of $\beta = 4$ and $R = 10^{-4}$ result in the best performance. However, they do not present a procedure for optimizing these parameters and relay on brute force search for finding the optimal values through trial and error.

B. Policy gradient RL

Policy gradient RL is a general purpose optimization method that can be used to learn a parameterized policy based on on-line experimental data. While there are several different methods for estimating the gradient of the policy performance with respect to the parameters [12], one of the most straightforward, and the one we use in this paper, is *Finite Difference Policy Gradient RL* (FD-PGRL) [13] which is based on finite differences. In this subsection we review the methods and formulations presented in [13].

FD-PGRL is presented in Algorithm 2. Under this framework, the policy is parameterized using the parameter vector $\pi = [\theta_1, \ldots, \theta_N]^\intercal$. The algorithm starts with the initial parameters $\pi^0 = [\theta_1^0, \ldots, \theta_N^0]^\intercal$ (line 1). At each step k, the policy gradient is estimated by running a set of randomly generated policies $\Pi^k = \{\pi_1^k, ..., \pi_M^k\}$ (lines 5- 7) where each policy is defined as:

$$\pi_m^k = [\theta_1^{k-1} + \delta_{1,m}^k, \ldots, \theta_N^{k-1} + \delta_{N,m}^k]^\intercal, \qquad (1)$$

where $\delta_{n,m}^k \in \{-\epsilon_n, 0, \epsilon_n\}$. The generated policies in (1) are obtained by randomly changing each parameter from the previous policy by a small ϵ_n, relative to θ_n. The cost of each newly created policy, π_m^k, is observed and denoted by c_m^k (lines 8- 9).

To estimate the policy gradient, the policy set in (1) is partitioned to three subsets (lines 11- 14) for each dimension depending on whether the change in the policy in that

[1] The tolls are not included in the calculation of social welfare, because we assume that toll revenues are transfer payments which remain internal to society.

Algorithm 2: Finite Difference Policy Gradient RL

1 $\pi^0 \leftarrow [\theta_1^0, \ldots, \theta_N^0]^\intercal$;
2 $k \leftarrow 0$;
3 **while** *improving* **do**
4 $\quad k \leftarrow k+1$;
5 $\quad$ generate $\Pi^k = \{\pi_1^k, ..., \pi_M^k\}$,
6 $\quad\quad \pi_m^k = [\theta_1^{k-1} + \delta_{1,m}^k, \ldots, \theta_N^{k-1} + \delta_{N,m}^k]^\intercal$,
7 $\quad\quad \delta_{n,m}^k \sim Uniform\{-\epsilon_n, 0, \epsilon_n\}$;
8 $\quad$ **for** *each* $m \in \{1, \ldots, M\}$ **do**
9 $\quad\quad c_m^k \leftarrow run(\pi_m^k)$;
10 $\quad$ **for** *each* $n \in \{1, \ldots, N\}$ **do**
11 $\quad\quad$ partition Π^k to
12 $\quad\quad\quad \Pi_{-\epsilon,n}^k = \{\pi_m^k : \delta_{n,m}^k = -\epsilon\}$,
13 $\quad\quad\quad \Pi_{0,n}^k = \{\pi_m^k : \delta_{n,m}^k = 0\}$,
14 $\quad\quad\quad \Pi_{+\epsilon,n}^k = \{\pi_m^k : \delta_{n,m}^k = \epsilon\}$;
15 $\quad\quad \overline{c}_{-\epsilon,n}^k \leftarrow average(c_m^k : \pi_m^k \in \Pi_{-\epsilon,n}^k)$;
16 $\quad\quad \overline{c}_{0,n}^k \leftarrow average(c_m^k : \pi_m^k \in \Pi_{0,n}^k)$;
17 $\quad\quad \overline{c}_{+\epsilon,n}^k \leftarrow average(c_m^k : \pi_m^k \in \Pi_{+\epsilon,n}^k)$;
18 $\quad\quad$ **if** $\overline{c}_{-\epsilon,n}^k < \overline{c}_{0,n}^k$ **&** $\overline{c}_{+\epsilon,n}^k < \overline{c}_{0,n}^k$ **then**
19 $\quad\quad\quad a_n^k \leftarrow 0$;
20 $\quad\quad$ **else**
21 $\quad\quad\quad a_n^k \leftarrow \overline{c}_{+\epsilon,n}^k - \overline{c}_{-\epsilon,n}^k$;
22 $\quad \pi^k \leftarrow \pi^{k-1} - \eta \frac{A^k}{|A^k|}$,
23 $\quad\quad A^k = [a_1^k, \ldots, a_N^k]^\intercal$;

dimension is negative, positive or zero, that is the three subsets are:

$$\pi_m^k \in \begin{cases} \Pi_{-\epsilon,n}^k = \{\pi_m^k : \delta_{n,m}^k = -\epsilon\} \\ \Pi_{0,n}^k = \{\pi_m^k : \delta_{n,m}^k = 0\} \\ \Pi_{+\epsilon,n}^k = \{\pi_m^k : \delta_{n,m}^k = \epsilon\}. \end{cases} \tag{2}$$

The average costs of above policy subsets are denoted by $\overline{c}_{-\epsilon,n}^k$, $\overline{c}_{0,n}^k$ and $\overline{c}_{+\epsilon,n}^k$ (lines 15- 17). The adjustment vector $A^k = [a_1^k, \ldots, a_N^k]^\intercal$ can be constructed by the following equation for each dimension (lines 18- 21):

$$a_n^k = \begin{cases} 0, & \text{if} \quad \overline{c}_{-\epsilon,n}^k < \overline{c}_{0,n}^k \text{and} \quad \overline{c}_{+\epsilon,n}^k < \overline{c}_{0,n}^k \\ \overline{c}_{+\epsilon,n}^k - \overline{c}_{-\epsilon,n}^k & \text{otherwise} \end{cases} \tag{3}$$

The adjustment vector A^k is normalized and multiplied by a constant step size η to update the parameter vector at the end of each step k (lines 22- 23).

Unlike other policy gradient methods that rely on within-episode reward signals to search for an optimal policy, or those in which the agent must learn the policy with no prior knowledge of a reasonably-performing starting policy (for example [14] and [15]), in the method employed in this paper, the policy is parameterized with a finite set of parameters and the overall system performance at each episode is optimized using an empirical estimate of the policy gradient based on finite differences. This approach is well-suited for the traffic optimization problem for two reasons. First, the agent can leverage an existing policy with reasonable system performance. Second, the agent is required to proceed towards the optimal policy only by slight changes of the policy parameters in contrast to approaches in which randomized exploration policies can be executed more freely. Our empirical study suggests that considering such slight changes results in a total cost that is within an acceptable bound. Furthermore, using other RL methods to learn actual tolls in real-time instead of Δ-*tolling* parameters requires modeling traffic as Markov Decision Process which is a challenging task (see [16]).

IV. Enhanced Delta-tolling

We now present the main contribution of this paper, the Enhanced Δ-*tolling* mechanism for solving the micro-tolling assignment problem. Enhanced Δ-*tolling* extends the Δ-*tolling* mechanism that is presented in Section III-A. Δ-*tolling* uses two global variables that are used to set tolls on every link in the network. Since different links possess different attributes e.g., capacity, length, speed limit, etc. optimizing the β and R parameters per link can potentially yield greater benefits (higher social welfare, lower total travel time). However, doing so would require optimizing a set of $2|E|$ parameters instead of only two. Optimizing such a high dimensional function cannot be done efficiently in a brute force way.

This paper introduces Enhanced Δ-*tolling* which extends Δ-*tolling* by first, considering unique β and R parameters per link and second, incorporating policy gradient RL for optimizing these parameters.

In order to apply policy gradient RL (specifically FD-PGRL, as described in Section III-B), the traffic assignment policy that maps the current state of the traffic to the appropriate actions, which are assigning tolls to each link of the network, should be parameterized. Since the Δ-*tolling* scheme, inherently implemented a policy that takes into account the real-time state of the traffic by assigning tolls proportional to the current links delay, we only use RL policy gradient method to optimize the performance metric at the end of each traffic cycle. Therefore, we define the cost to be the total travel time at the end of each day and consider the following three parametrization of Δ-*tolling*:

$$\begin{aligned} \pi_R &= [\beta, R_1, \ldots, R_n] \\ \pi_\beta &= [R, \beta_1, \ldots, \beta_n] \\ \pi_{R,\beta} &= [R_1, \ldots, R_n, \beta_1, \ldots, \beta_n] \end{aligned} \tag{4}$$

The experimental results presented by Sharon et al. [3] suggest that there is some correlation between the optimally performing β and R values. However, no conclusions were presented regarding how they correlate and their individual impact on the convergence rate in a parameter tuning procedure.

As the relation between the β and R parameters remains unclear, we consider three variants of Enhanced Δ-*tolling*based on the parameterized policies listed in (4):

EΔ-*tolling*$_\beta$: this variant uses a global R parameter and link specific β parameters ($|E|+1$ parameters in total). It

should perform well under the assumption that there is a correlation between the best performing β and R values and when FD-PGRL estimates the gradient over link specific β parameters more accurately than it does for link specific R parameters.

EΔ-*tolling*$_R$: this variant uses a global β parameter and link specific R parameters ($|E|+1$ parameters in total). It should perform well under the assumption that there is a correlation between the best performing β and R values and when FD-PGRL estimates the gradient over link specific R parameters more accurately than it does for link specific β parameters.

EΔ-*tolling*$_{\beta,R}$: this variant uses link specific β and R parameters ($2|E|$ parameters in total). It should perform best if there is no correlation between the best performing β and R values and if sufficient computation time is given (converging on $2|E|$ parameters is usually slower than on $|E|+1$).

V. Empirical study

Our experimental evaluation focuses on real-life road networks. Traffic is evaluated using the cell transmission model (CTM) [17, 18] which is a discrete, explicit solution method for the hydrodynamic theory of traffic flow proposed in [19] and [20].

CTM is frequently used in dynamic traffic assignment. The time step used in this model is typically short, on the order of a few seconds. When used with Enhanced Δ-*tolling*, this allows for a truly adaptive toll which can be updated based on observed traffic conditions.

A. Scenario specification

Demand model: demand is given as a trip table, where every entry is affiliated with a single agent (a) and specifies: a source node (s_a), a target node (t_a), and a departure time step (i_a).

Agent model: let l_p^i be the sum of latency along path p during time step i and let τ_p^i be the sum of tolls along p during time step i. When agent a reaches a diverge node n at time step i all paths (P_{nt}) leading from n to destination t_a are considered. Agent a is assigned the minimal cost path i.e., $\arg\min_{p\in P_{nt}}\{\tau_p^i + l_p^i \cdot c_a\}$.

B. Experiments and results

For running CTM we used the DTA simulator [21] implemented in Java. Whenever a vehicle is loaded onto the network, it is assigned a VOT randomly drawn from a Dagum distribution with parameters $\hat{a} = 22020.6$, $\hat{b} = 2.7926$, and $\hat{c} = 0.2977$, reflecting the distribution of personal income in the United States [22, 23].[2]

The step size in FD-RPGS , η, is 0.4. The policy perturbation parameter, ϵ (see Line 2 in Algorithm 2) is set to 0.01 and the number of policy runs at each step, M, is 60 for all the experiments. These values presented best performance overall. Our empirical study focuses on three traffic scenarios:

[2]The simulation settings were chosen to be identical to those presented in [3].

(a) Sioux Falls (b) Austin (c) San Antonio

Fig. 1: Maps of traffic networks used in the experiments

Sioux Falls: [24] — this scenario is common in the transportation literature [25], and consists of 76 directed links, 24 nodes (intersections) and 28,835 trips spanning 3 hours.

Downtown Austin: [26] — this network consists of 1,247 directed links, 546 nodes and 62,836 trips spanning 2 hours during the morning peak.

Uptown San Antonio: this network consists of 1,259 directed links, 742 nodes and 223,479 trips spanning 3 hour during the morning peak.

The networks affiliated with each scenario are depicted in Figure 1. All of these traffic scenarios are available online at: https://goo.gl/SyvV5m

1) System performance: Our first set of results aims to evaluate the performance of the different variants of Enhanced Δ-*tolling*, by comparing them with each other and basic Δ-*tolling*. Figure 2 presents normalized values of total latency summed over all trips (top figure) and social welfare that is the summation of costs, i.e., latency times VOT, over all agents (bottom figure). The values are normalized according to the system's performance when no tolls are applied. Table I presents the total latency and social welfare performance when applying no-tolls (representing the value of 1.0 in Figure 2).

The results present a clear picture in which Δ-*tolling* improves on applying no tolls in both total latency and social welfare. EΔ-*tolling*$_\beta$ further improve the system's performance and both EΔ-*tolling*$_R$ and EΔ-*tolling*$_{\beta,R}$ achieve the best performance.

The fact that EΔ-*tolling*$_R$ results in system performance which is similar to EΔ-*tolling*$_{\beta,R}$ suggests that there is a correlation between the best performing β and R values. The slight superiority of of EΔ-*tolling*$_R$ comparing to EΔ-*tolling*$_{\beta,R}$ is due to faster convergence which will be discussed later in this section. The fact that EΔ-*tolling*$_\beta$ performs worse than EΔ-*tolling*$_R$ suggests that policy FD-PGRL estimates the gradient over link specific R parameters more accurately than it does for link specific β parameters.

2) Convergence rate: applying EΔ-*tolling* to real-life traffic raises two concerns:

1) Convergence rate - the system should converge to a good solution with as few learning iterations as possible.
2) Worst case performance - during the learning process EΔ-*tolling* should perform at least as well as Δ-*tolling*.

	Sioux Falls	Austin	San Antonio
Latency (hr)	11,859	21,590	26,362
cost ($)	353,169	637,086	780,739

TABLE I: Average total latency and total generalized cost when applying no tolls.

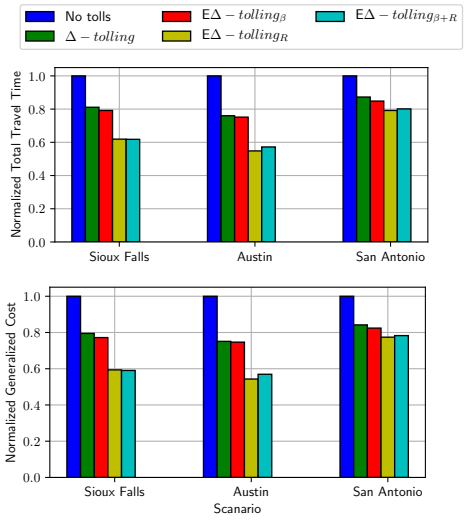


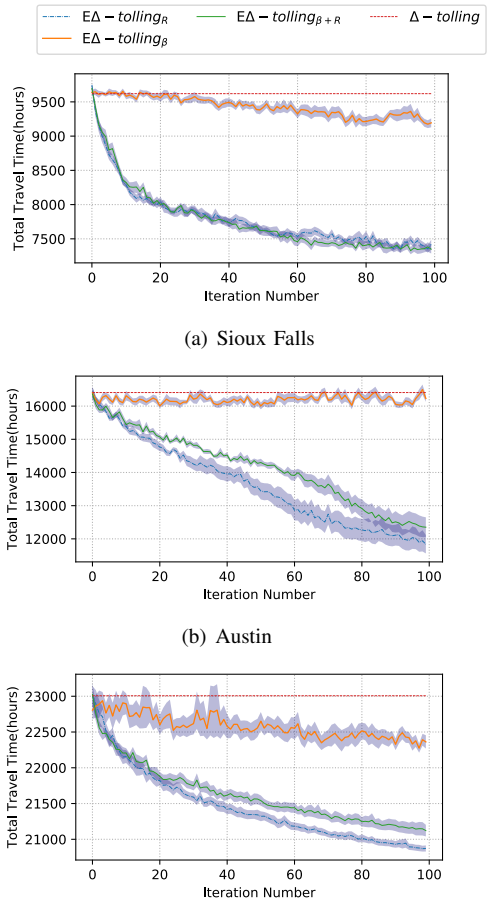
Fig. 2: Total Travel Time and Total Generalized Cost for different tolling schemes and scenarios.

Figure 3 presents the system performance w.r.t total latency (y-axis) versus learning iteration step (x-axis) for each of our three scenarios and every EΔ-*tolling* variant. The error regions are obtained using 10 different runs of the algorithm for each example and EΔ-*tolling* variant and they show the standard error of the average performance in each iteration. Results for basic Δ-*tolling* are also included for comparison. The results are consistent with each other, showing that EΔ-*tolling*$_R$ performs best overall w.r.t convergence rate.

Table II presents the area under the curve for each scenario and EΔ-*tolling* variant. These results give a quantitative comparison of the convergence rates. We learn that EΔ-*tolling*$_R$ has the best overall performance with a total AUC of 4,285,353. Nonetheless, EΔ-*tolling*$_{\beta,R}$ performs better on the Sioux Falls scenario. All the experiments are initialized with $\beta = 4$ and $R = 10^{-4}$ for all the links. A set of experiments (not presented) with different starting parameter values show that the performance is sensitive to the initial settings. However, the mentioned default starting values ($\beta = 4$ and $R = 10^{-4}$) perform relatively well across all scenarios and EΔ-*tolling* variants.

Scheme	S. Falls	Austin	S. Antonio	Total
Δ-*tolling*	962,000	1,640,900	2,300,700	4,903,600
EΔ_{β}	943,076	1,619,928	2,257,830	4,820,834
EΔ_{R}	779,990	**1,360,861**	**2,144,502**	**4,285,353**
E$\Delta_{\beta+R}$	**777,469**	1,415,094	2,162,006	4,354,569

TABLE II: Area under the convergence curves from Figure 3.



(a) Sioux Falls

(b) Austin

(c) San Antonio

Fig. 3: System performance w.r.t total latency (y-axis) versus learning iteration step (x-axis) for different scenarios and EΔ-*tolling* variants

VI. Discussion and Future Work

The promising experimental results reported in Section V suggest that EΔ-*tolling* can have practical applications where traffic optimization is performed constantly and in real-time through manipulations to the R and or β parameters. Nonetheless, implementation of EΔ-*tolling* raises several practical issues that must first be addressed.

Limitations: EΔ-*tolling* is limited in its convergence rate. General traffic patterns might change frequently, preventing EΔ-*tolling* from advancing in a promising direction. Practitioners must evaluate the convergence rate of EΔ-*tolling* versus the rate in which traffic patterns change in order to determine the applicability of EΔ-*tolling* in a specific network.

Assumptions: EΔ-*tolling*assumes that all agents traversing the network are self-interested and responsive to tolls in real time. Real world scenarios might violate these assumptions and the trends observed in our results cannot be assumed in such cases.

Practical aspects of EΔ-*tolling* present many promising directions for future work. Since the convergence rate of EΔ-*tolling* plays an important role in determining its applica-

bility, one promising direction for future work is developing heuristics and utilizing advanced RL methods to guide the gradient exploration towards promising directions in order to facilitate faster learning.

Examining the effects of partial compliance to tolls is another promising direction. Building on recent study that examines the effects of partial compliance on similar micro-tolling schemes [27], studying the practical impacts of partial compliance on EΔ-*tolling* is a promising direction to pursue.

VII. Conclusion

This paper introduced Enhanced Δ-*tolling*, a micro-tolling assignment scheme that builds on the previously suggested Δ-*tolling* scheme. The previously suggested Δ-*tolling* scheme makes use of two global parameters, β and R, to tune the system for optimized performance (minimal total latency or maximal social welfare). Enhanced Δ-*tolling* generalizes Δ-*tolling* in two complementary ways. First, recognizing that different links in the network have different attributes (length, capacity, speed limit) Enhanced Δ-*tolling* considers individual β and R parameters per link. Second, given the resulting large parameter set (twice the number of links), Enhanced Δ-*tolling* suggests a policy gradient RL approach for tuning these parameters. Experimental results suggest that tuning the R parameter while keeping a global β parameter performs best overall (w.r.t total latency, social welfare, worst case performance, and convergence rates).

VIII. Acknowledgements

This work was supported in part by the National Science Foundation under NSF grant number 1563652.

References

[1] M. Amir and T. Givargis, "Hybrid state machine model for fast model predictive control: Application to path tracking," in *Proceedings of the 36th International Conference on Computer-Aided Design*, ser. ICCAD '17. IEEE Press, 2017, pp. 185–192.

[2] G. Sharon, J. P. Hanna, T. Rambha, M. W. Levin, M. Albert, S. D. Boyles, and P. Stone, "Real-time adaptive tolling scheme for optimized social welfare in traffic networks," in *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2017)*, May 2017.

[3] G. Sharon, M. W. Levin, J. P. Hanna, T. Rambha, S. D. Boyles, and P. Stone, "Network-wide adaptive tolling for connected and automated vehicles," *Transportation Research Part C*, vol. 84, pp. 142–157, September 2017.

[4] H. Chen, B. An, G. Sharon, J. P. Hanna, P. Stone, C. Miao, and Y. C. Soh, "Dyetc: Dynamic electronic toll collection for traffic congestion alleviation," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI-18)*, February 2018.

[5] S. Lu, "Sensitivity of static traffic user equilibria with perturbations in arc cost function and travel demand," *Transportation Science*, vol. 42, no. 1, pp. 105–123, 2008.

[6] H. Yang, Q. Meng, and D.-H. Lee, "Trial-and-error implementation of marginal-cost pricing on networks in the absence of demand functions," *Transportation Research Part B: Methodological*, vol. 38, no. 6, pp. 477–493, 2004.

[7] A. C. Pigou, *The Economics of Welfare*. Palgrave Macmillan, 1920.

[8] M. J. Beckmann, C. B. McGuire, and C. B. Winston, *Studies in the Economics of Transportation*. New Haven, CT: Yale University Press, 1956.

[9] Y. Sheffi, "Urban transportation network," *Equilibrium analysis with mathematical programming methods, Prentice Hall*, 1985.

[10] R. B. Dial, "Minimal-revenue congestion pricing part I: A fast algorithm for the single-origin case," *Transportation Research Part B*, vol. 33, pp. 189–202, 1999.

[11] T. A. Roughgarden, "Selfish routing," Ph.D. dissertation, Cornell University, 2002.

[12] J. Peters and S. Schaal, "Policy gradient methods for robotics," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2006, October 9-15, 2006, Beijing, China*, 2006.

[13] N. Kohl and P. Stone, "Policy gradient reinforcement learning for fast quadrupedal locomotion," in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation, ICRA 2004,.*

[14] S. El Bsat, H. Bou-Ammar, and M. E. Taylor, "Scalable multitask policy gradient reinforcement learning." in *AAAI*, 2017, pp. 1847–1853.

[15] S. Levine and P. Abbeel, "Learning neural network policies with guided policy search under unknown dynamics," in *Advances in Neural Information Processing Systems*, 2014, pp. 1071–1079.

[16] A. L. Bazzan, "Opportunities for multiagent systems and multiagent reinforcement learning in traffic control," *Autonomous Agents and Multi-Agent Systems*, vol. 18, no. 3, p. 342, 2009.

[17] C. F. Daganzo, "The cell transmission model: a dynamic representation of highway traffic consistent with the hydrodynamic theory," *Transportation Research Part B*, vol. 28, no. 4, pp. 269–287, 1994.

[18] ——, "The cell transmission model, part II: network traffic," *Transportation Research Part B*, vol. 29, no. 2, pp. 79–93, 1995.

[19] M. Lighthill and G. Whitham, "On kinematic waves. II. A theory of traffic flow on long crowded roads," *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, pp. 317–345, 1955.

[20] P. Richards, "Shock waves on the highway," *Operations Research*, vol. 4, no. 1, pp. 42–51, 1956.

[21] Y.-C. Chiu, J. Bottom, M. Mahut, A. Paz, R. Balakrishna, T. Waller, and J. Hicks, "Dynamic traffic assignment: A primer," *Transportation Research E-Circular*, no. E-C153, 2011.

[22] P. Lukasiewicza, K. Karpioa, and A. Orlowskia, "The models of personal incomes in USA," in *Proceedings of the 5th Symposium on Physics in Economics and Social Sciences*, Warsaw, Poland, 2012.

[23] L. Gardner, H. Bar-Gera, and S. D. Boyles, "Development and comparison of choice models and tolling schemes for high-occupancy/toll (HOT) facilities," *Transportation Research Part B*, vol. 55, pp. 142–153, 2013.

[24] L. J. LeBlanc, E. K. Morlok, and W. P. Pierskalla, "An efficient approach to solving the road network equilibrium traffic assignment problem," *Transportation Research*, vol. 9, no. 5, pp. 309–318, 1975.

[25] M. W. Levin and S. D. Boyles, "Intersection auctions and reservation-based control in dynamic traffic assignment," in *Transportation Research Board 94th Annual Meeting*, no. 15-2149, 2015.

[26] M. W. Levin, M. Pool, T. Owens, N. R. Juri, and S. T. Waller, "Improving the convergence of simulation-based dynamic traffic assignment methodologies," *Networks and Spatial Economics*, vol. 15, no. 3, pp. 655–676, 2015.

[27] G. Sharon, M. Albert, S. B. Tarun Rambha, and P. Stone, "Traffic optimization for a mixture of self-interested and compliant agents," in *32th AAAI Conference on Artificial Intelligence (AAAI-18)*, 2018.