The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Traffic Optimization for a Mixture of Self-Interested and Compliant Agents

Guni Sharon,[1]* Michael Albert,[2]* Tarun Rambha,[3]* Stephen Boyles,[4] Peter Stone[1]
[1]Department of Computer Science, University of Texas at Austin, Austin, TX 78712, USA
[2]Department of Computer Science, Duke University, Durham, NC 27708, USA
[3]Civil and Environmental Engineering, Cornell University, Ithaca, NY 14853, USA
[4]Civil, Architectural and Environmental Engineering, The University of Texas at Austin, Austin, TX 78712, USA
gunisharon@gmail.com, malbert@cs.duke.edu, tr244@cornell.edu, sboyles@mail.utexas.edu, pstone@cs.utexas.edu

Abstract

This paper focuses on two commonly used path assignment policies for agents traversing a congested network: *self-interested routing*, and *system-optimum routing*. In the self-interested routing policy each agent selects a path that optimizes its own utility, while the system-optimum routing agents are assigned paths with the goal of maximizing system performance. This paper considers a scenario where a centralized network manager wishes to optimize utilities over all agents, i.e., implement a system-optimum routing policy. In many real-life scenarios, however, the system manager is unable to influence the route assignment of all agents due to limited influence on route choice decisions. Motivated by such scenarios, a computationally tractable method is presented that computes the minimal amount of agents that the system manager needs to influence (compliant agents) in order to achieve system optimal performance. Moreover, this methodology can also determine whether a given set of compliant agents is sufficient to achieve system optimum and compute the optimal route assignment for the compliant agents to do so. Experimental results are presented showing that in several large-scale, realistic traffic networks optimal flow can be achieved with as low as 13% of the agent being compliant and up to 54%.

Introduction

In multiagent systems, there are generally two paradigms of interaction. Centralized control paradigms assume that a single decision making entity is able to dictate the actions of all the agents, thus leading them to a coordinated social optimum. Decentralized control paradigms, on the other hand, assume that each agent selects its own actions, and while it is in principle possible for them to act altruistically, they are generally assumed to be self-interested.

In this paper, we consider a routing scenario in which a subset of agents are controlled centrally (*compliant agents*), while the remaining are *self-interested agents*. We model the system as a Stackelberg routing game (Yang, Zhang, and Meng 2007) in which the decision maker for the centrally controlled agents is the leader, and the self-interested agents are the followers. In this paper, we provide a computationally tractable methodology for 1) determining the maximum number of self-interested agents that a system can tolerate at optimal flow, 2) determining whether a given subset of centrally controlled agents are sufficient to achieve system optimum (SO), and 3) computing the actions the leader should prescribe to a sufficient set of compliant agents in order to achieve SO.

A known fact in routing games is that agents seeking to minimize their private latency need not minimize the total system's latency (Pigou 1920; Roughgarden and Tardos 2002). That is, self-interested agents may reach a user equilibrium (UE) that is not optimal from a system perspective. However, if all agents are assigned paths with minimum system marginal cost then the system will achieve optimal performance (Pigou 1920; Beckmann, McGuire, and Winsten 1956; Dietrich 1969).

Therefore, from a system manager perspective, it is desirable that all agents traversing a network would strictly utilize minimal marginal cost paths, even if such paths are not of minimum latency for an individual agent. However, in many important scenarios, it will not be possible to enforce path assignment on all agents, but it may be possible to affect the behavior of a subset (the compliant agents). As a motivating example, consider an opt-in tolling system where drivers are given positive incentives to enroll but, in exchange, they will be subject to tolls that affect their route choice (Sharon et al. 2017a; 2017b). Another relevant example is virtual private network (VPN) path allocation. While each packet within the VPN might be self-interested, a pro-social network manager might allocate virtual paths that are different from those preferred by the self-interested packets (Fingerhut, Suri, and Turner 1997; Duffield et al. 1999).

We show that, in the general case, computing the optimal assignment of compliant agents is NP-hard. Therefore, we focus on the specific scenario where the portion of compliant agents is sufficiently large to achieve SO. We present a novel *linear program* (LP) representation for computing the maximal portion of self-interested agents that allow the system to achieve SO and to determine whether a given set of compliant agents is sufficient to achieve SO. Furthermore, we provide a method to tractably compute the flow assignment for the compliant agents such that SO performance is guaranteed.

Experimental results, performed using a standard traffic

*These authors contributed equally.

simulator, are provided and demonstrate that the number of compliant agents necessary to achieve system optimum can be a relatively small percentage of total flow (between 13% and 53%).

Motivation

Recent advances in GPS based tolling technology (Numrich, Ruja, and Voß 2012) open the possibility of implementing micro-tolling systems in which specific tolls are charged for the use of links within a road network. Such 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. Setting tolls appropriately can influence self-interested drivers to prefer paths with minimum system marginal cost and thus, lead to improved system performance (Sharon et al. 2017a; 2017b).

Unfortunately, political factors deter public officials from allowing such a micro-tolling scheme to be realized. Road pricing is known to cause a great deal of public unrest and is thus opposed by governmental institutions (Schaller 2010). To tackle this issue and avoid public unrest, we suggests an *opt-in* micro-tolling system where, given some initial monetary sign-up incentive, drivers choose to opt-in to the system and be charged for each journey they take based on their chosen route. The vehicles belonging to such drivers would need to be equipped with a GPS device as well as a computerized navigation system. Given the toll values and driver's value of time, the navigation system would suggest a minimal cost route where the cost is a function of the travel time and tolls.

While addressing the issue of political acceptance, an opt-in system would result in traffic that is composed of a mixture of self-interested and compliant agents (compliant in the sense that the system manager can influence their route choice). Such a scenario raises some practical questions which are the focus of this paper, namely, what portion of self-interested agents can the system tolerate while still reaching optimum performance? The answer to this question can help practitioners to determine both the level and the targeting of incentives in an opt-in system.

Problem definition and terminology

The terminology in this paper follows that of Roughgarden and Tardos (2002). We review the relevant concepts and notation in this section.

The flow model

The flow model in this work is composed of a directed graph $G(V, E)$, and a demand function $R(s,t) \rightarrow \mathbb{R}^+$ mapping a pair of vertices $s, t \in V^2$ to a non-negative real number representing the required amount of flow between source, s, and target, t.[1] An instance of the flow model is a $\{G, R\}$ pair.

$\mathcal{P}_{s,t}$ denotes the set of acyclic paths from s to t. Define $\mathcal{P}$ as the collection of all $\mathcal{P}_{s,t}$ (i.e., $\cup_{s,t \in V^2} \mathcal{P}_{s,t}$). The variable f_p represents the flow volume assigned to path p. Similarly, f_e is the flow volume assigned to link e. By definition, the flow on each link (f_e) equals the summation of flows on all paths of which e is a part. Define the system flow vector as $f = \text{vect}\{f_p\}$. f is said to be *feasible* if for all $s, t \in V^2$, $\sum_{p \in \mathcal{P}_{s,t}} f_p = R(s,t)$.

Each link $e \in E$ has a latency function $l_e(f_e)$ which, given a flow volume (f_e), returns the latency (travel time) on e. Following Roughgarden and Tardos (2002) we make the following assumption:

Assumption 1. *The latency function $l_e(f_e)$ is non-negative, differentiable, and non-decreasing for each link $e \in E$.*

The latency of a simple path p for a given flow f, is defined as $l_p(f) = \sum_{e \in p} l_e(f_e)$. A feasible flow f is defined as a *user equilibrium* (UE) if for every $s, t \in V^2$ and $p_a, p_b \in \mathcal{P}_{s,t}$ with $f_{p_a} > 0$ it holds that $l_{p_a}(f) \leq l_{p_b}(f)$ (see Lemma 2.2 in (Roughgarden and Tardos 2002)). In other words, at UE, no amount of flow can be rerouted to a path with lower latency when the rest of the flow is fixed.

Define the system cost associated with link e as $c_e(f_e) = l_e(f_e) f_e$, the cost of a path p as $c_p(f) = \sum_{e \in p} c_e(f_e)$ and the cost of a flow f as $c(f) = \sum_{e \in E} c_e(f_e)$. Define $c'_e(x) = \frac{d}{dx} c_e(x)$ and $c'_p(f) = \sum_{e \in p} c'_e(f_e)$. A feasible flow f is defined as a *system optimum* (SO) flow if for every $s, t \in V^2$ and $p_a, p_b \in \mathcal{P}_{s,t}$ with $f_{p_a} > 0$ it holds that ${c'}_{p_a}(f) \leq c'_{p_b}(f)$ (see Lemma 2.5 in (Roughgarden and Tardos 2002)). In other words, at SO, the benefit from reducing the flow along any path is always less than or equal to the cost of adding the same amount of flow to a parallel, alternative path. We follow Roughgarden and Tardos (2002), and make the following assumption:

Assumption 2. *The cost function $c_e(f_e)$ is convex for each link $e \in E$.*

Assumptions 1 and 2 imply that the set of SO flows correspond to the set of solutions of a convex program where the objective is to minimize $c(f) = \sum_{e \in E} c_e(f_e)$ (see Roughgarden and Tardos (2002) Corollary 2.7).

Problem Definition

The focus of this paper is a scenario where the demand is partitioned into self-interested and compliant agents. We define two types of controllers that assign paths to all of the agents. These controllers are viewed as players in a Stackelberg game (Yang, Zhang, and Meng 2007).

- SO-controller - Stackelberg leader, the SO-controller aspires to assign paths to the compliant subset of agents that, taking into account the self-interested agents' reaction, optimizes the systems performance (i.e. minimizes total latency). We refer to flow assigned by the SO-controller as *compliant flow*.
- UE-controller - Stackelberg follower, considering the compliant agents' path assignment as fixed, the UE-controller assigns paths to the self-interested agents, the

[1]The demand between any source and target, $R(s,t)$, can be viewed as an infinitely divisible set of agents (also known as a non-atomic flow (Rosenthal 1973)).

UE flow, such that a state of user equilibrium (as defined above) is achieved.[2]

The problems addressed in this paper are:

1. Given an instance of the flow model $\{G, R\}$, what is the maximum amount of self-interested agents that can be assigned to the UE controller and still permit the optimal flow.
2. Given a set of compliant agents and an instance of the flow model $\{G, R\}$, can the SO controller assign paths to them in such a way that the system achieves SO.
3. If SO is achievable, how should the SO-controller assign the compliant flow. Equivalently, what is the optimal Stackelberg equilibrium.

To the best of our knowledge, this work is the first to answer these questions in a general setting.

Related Work

Previous work examined mixed equilibrium scenarios where traffic is composed of: UE and Cournot-Nash (CN) controllers. A CN-controller assigns flows to a given subset of the demand with the aim of minimizing the total travel time only for that subset. For instance, a logistic company with many trucks can be viewed as a CN-controller.

It was shown that the equilibrium for a mixed UE, CN scenario is unique and can be computed using a convex program (Haurie and Marcotte 1985; Yang and Zhang 2008). On the other hand, no tractable algorithm is known for computing the optimal Stackelberg equilibrium for scenarios that also include a SO-controller.

Korilis et. al. (1997) examined mixed equilibrium scenarios that do include a SO-controller. In their work, a technique for computing a solution for the above questions #1 and #3 was suggested for specific types of flow models. Their technique was proven to work for networks with a common source and a common target with any number of parallel links. Moreover, the latency functions were assumed to be of a very specific form (linear function with a capacity bound). As a result, their solution is not applicable when general networks with arbitrary latency functions are considered.

Other work (Roughgarden 2004; Immorlica et al. 2009) studied a variant of the scheduling problem where infinitesimal jobs must be assigned to a set of shared machines each of which is affiliated with a non-negative, differentiable, and non-decreasing latency function that, given the machine load, specify the amount of time needed to complete a job. When considering a scenario where part of the jobs are assigned to machines by a UE-controller while the rest are assigned by a SO-controller, they show it is NP-hard to compute the optimal Stackelberg equilibrium (Roughgarden 2004). Their problem can be viewed as a special case of our problem, specifically a network with a single source and target with multiple parallel links between them. Given that in this, more restrictive setting, computing the optimal Stackelberg equilibrium is intractable, the same yet general question in our setting will also be computationally intractable.

[2]The UE enforced by the UE-controller applies only for the self-interested subset of agents. That is, no **self-interested** agent can benefit from unilaterally deviating from its assigned path.

Computing the Maximal UE Flow

Given that finding the optimal Stackelberg equilibrium is NP-hard for an arbitrary size of compliant flow, this work focuses on scenarios where the size of the compliant flow is sufficient to achieve SO. As we will show, finding the optimal Stackelberg equilibrium can be done in polynomial time for such cases. In this section, we will present a computationally tractable method to compute the maximal UE flow given an instance of a flow model $\{G, R\}$, and we will provide a method to check, for a given level of compliant flow, whether SO is achievable.

We define r^*_{UE} as the maximal amount of demand comprised of self-interested agents that the system can tolerate and still achieve SO. Additionally, we define $r^*_{s,t}$ as the amount of demand from source s to target t that is assigned to the UE-controller. That is, computing r^*_{UE} is equivalent to maximizing $\sum_{s,t} r^*_{s,t}$.

We can cast the problem of maximizing $\sum_{s,t} r^*_{s,t}$ as an optimization problem, specifically a *linear program* (LP). Assigning values to all variables of type $r^*_{s,t}$ must follow some constraints. Specifically, the UE flow from each origin to each destination must be both a *subflow* of the SO flow, and must follow a least latency path.

Definition 1 (Subflow of flow f)**.** *For a directed graph $G(V, E)$ and demand function R, a flow f^* is a* subflow of flow *f if for all links $e \in E$, $0 \leq f^*_e \leq f_e$ and for each pair of nodes $s, t \in V^2$, there exists $0 \leq r_{s,t} \leq R(s, t)$ such that*

$$\sum_{e \in out(s)} f^*_e - \sum_{e \in in(s)} f^*_e = \sum_t r_{s,t}$$

and

$$\sum_{e \in in(t)} f^*_e - \sum_{e \in out(t)} f^*_e = \sum_s r_{s,t}.$$

A path p, leading from vertex s to vertex t, is said to be *zero reduced cost* if there is no other path, p', leading from s to t with lower latency or lower marginal cost.

Definition 2 (Zero reduced cost path)**.** *For a flow model $\{G, R\}$, a* zero reduced cost path *with regard to flow assignment f is a path $p \in \mathcal{P}_{s,t}$ such that $\forall\, p' \in \mathcal{P}_{s,t} : l_p(f) \leq l_{p'}(f)$ and $c'_p(f) \leq c'_{p'}(f)$. A link, e, is defined as a* zero reduced cost link*, with respect to source s, if it is part of any zero reduced cost path originating from s and terminating at t for some origin-destination pair $(s, t) \in V^2$. We denote the set of zero reduced cost links with respect to source s as E^s_{RC}*

We require that the UE flow (flow routed by the UE-controller) is routed solely via zero reduced cost links/paths. This is because the UE controller can only assign flow to minimal latency paths (otherwise self-interested agents would deviate). the UE flow is also required to follow minimal marginal cost paths else it cannot be a subflow of the SO flow.

Note that it is sufficient to only consider whether or not a link e is part of a reduced cost path from the origin s to *some* destination t (not a specific t) because either link e is along a reduced cost path from (s, t), or there is no path only along links in E^s_{RC} that includes e.

We can efficiently compute the set of zero reduced cost links for any origin destination pair (s, t) by applying uniform cost search from s to t and marking all links that are part of optimal paths, once with regard to minimal total latency $(\arg\min_{p \in \mathcal{P}_{s,t}}(l_p(f^{SO}))$, and second with regard to minimal marginal cost $(\arg\min_{p \in \mathcal{P}_{s,t}}(c'_p(f^{SO}))$.

Let the constant f^{SO} denote the flow vector at a SO solution.[3] The SO flow is not unique when latency functions are non-decreasing, and the maximal amount of UE flow permitted may, in general, depend on the specific SO flow. Therefore, we must efficiently search over the space of SO flows. This is possible due to the following lemmas.

Lemma 1. *For any two flows that achieve SO, f^{SO} and $\hat{f}^{SO}$, $l_e(f^{SO}_e) = l_e(\hat{f}^{SO}_e)$.*

Proof. Given Assumption 2, a SO flow is the solution to a convex program (Roughgarden and Tardos 2002). The solutions to a convex program form a convex set. Suppose that there are two flows that both achieve SO, but for which $f^{SO}_e \neq \hat{f}^{SO}_e$. Then $c_e(f_e) = l_e(f_e)f_e$ must be a linear function between f^{SO}_e and $\hat{f}^{SO}_e$ (to see this, note that any convex combination of f^{SO} and $\hat{f}^{SO}$ is also an SO solution, but if $c_e(f_e)$ is not linear, then the total system travel time would be strictly less, a contradiction). Since $l_e(f_e)$ is a non-decreasing function, the only way for $c_e(f_e)$ to be linear is for $l_e(f_e)$ to be constant between f^{SO}_e and $\hat{f}^{SO}_e$. □

Lemma 2. *The set of zero reduced cost paths is identical for all SO solutions.*

Proof. By Lemma 1, all SO flows have the same latency on each link, so the SO solutions can differ by at most flows along a set of links with constant latency over the range of which the two flows differ on those links. Since we assume that the latency functions are differentiable, the derivatives of the latency function are zero over the range at which they are constant. Therefore, $c'_e(f_e) = l_e(f_e) + f_e l'_e(f_e)$ is constant over the range as well. This implies that any path that is reduced cost in one flow is also reduced cost in the other flow, since the latency functions and $c'_e(f_e)$ are constant for every link e. □

Define the constant $\bar{f}^{SO}_e = \sup\{f : l_e(f) = l_e(f^{SO}_e)\}$, i.e. $\bar{f}^{SO}_e$ is the largest flow value such that the latency on link e is equal to the latency at a SO solution. Note that if l_e is strictly increasing at f^{SO}_e, then $\bar{f}^{SO}_e = f^{SO}_e$. However, if l_e is constant at f^{SO}_e, then $\bar{f}^{SO}_e > f^{SO}_e$.

Given that the zero reduced cost paths are the same for *all SO* flows (Lemma 2), and any SO flow has the same latency on all links (Lemma 1), it will be sufficient to only search over flows that are less than $\bar{f}^{SO}_e$ on each link $e \in E$.

[3]A SO flow can be efficiently computed as a solution to a convex program (Roughgarden and Tardos 2002; Dial 2006).

For each vertex, s, and link, e, define variable x^s_e denoting the amount of UE flow originating from source s that is assigned to link e. Let $in(v)$ denote the set of links for which v is the tail vertex and $out(v)$ the set of links for which v is the head vertex.

Definition 3. *For a given flow model $\{G, R\}$, the UE* linear program *is:*

$$\max_{r^*_{s,t}, x^s_e} \sum_{s,t \in V^2} r^*_{s,t} \tag{1}$$

subject to

$$r^*_{s,t} \leq R(s,t) \qquad \forall\, s,t \in V^2 \tag{2}$$

$$\sum_{e \in out(s)} x^s_e = \sum_{t \in V} r^*_{s,t} \qquad \forall\, s \in V \tag{3}$$

$$\sum_{e \in in(t)} x^s_e - \sum_{e \in out(t)} x^s_e = r^*_{s,v} \qquad \forall\, s,t \in V^2 \tag{4}$$

$$\sum_s x^s_e \leq \bar{f}^{SO}_e \qquad \forall\, e \in E,\ s \in V \tag{5}$$

$$x^s_e \geq 0,\ r^*_{s,t} \geq 0 \qquad \forall\, s,t \in V,\, e \in E \tag{6}$$

$$x^s_e = 0 \qquad \forall\, s \in V,\, e \in E \setminus E^s_{RC} \tag{7}$$

The flow $f^{UE}_e = \sum_v x^v_e$ defined by a feasible solution to the UE linear program (given constraints (2)-(7)*) is a UE* subflow. *The flow defined by an optimal solution to the UE linear program is an* optimal *UE* subflow.

Note that the number of variables is $|\{\forall s \in V, \forall t \in V, \forall e \in E : r^*_{s,t}, x^s_e\}| = O(|V|^2 + |V||E|)$, and the number of constraints is also $O(|V|^2 + |V||E|)$. Therefore, since the number of variables and constraints are polynomial in the flow model, the optimal solution to the UE linear program can be computed in polynomial time (Karmarkar 1984).

Theorem 1. *A UE subflow, f^{UE}, defined by a feasible solution to the UE linear program is a subflow of a SO flow.*

Proof. First, note that by equations (2)–(4), the UE subflow, f^{UE}_e, satisfies flow conservation constraints. Equation (2) states that the flow along all zero reduced cost paths from origin s to destination t must be less then total demand for (s, t). Then equations (3) and (4) state that the flow out of node v must either be due to the demand generated by node v or the flow into it, minus the flow that reaches v as a destination. Therefore, f^{UE}_e is a subflow of a feasible flow.

What must be shown is that there must exist a SO flow, f^{SO}, such that $f^{UE}_e \leq f^{SO}_e$ for all e. If e is such that l_e is strictly increasing at an SO solution, and therefore will be strictly increasing at all SO solutions by Lemma 1, then $f^{SO}_e = \bar{f}^{SO}_e$ and constraint (5) guarantees this claim. Let E' be the set of links such that the latency function is constant at a SO flow. Therefore, it only needs to be shown that there exists a SO solution, f, such that for $e \in E'$, $f^{UE}_e \leq f^{SO}_e$.

Suppose that there existed a set of links $e \in E'$ such that for all SO flows f^{SO}, $f^{UE}_e > f^{SO}_e$. Let $\hat{f}^{SO}$ be an SO flow. Then there must exist an origin destination pair

(s, t) such that there are two sets of paths $\mathcal{P}_>, \mathcal{P}_< \subset \mathcal{P}_{s,t}$ for which for all $p \in \mathcal{P}_>$, $f_p^{UE} > \hat{f}_p^{SO}$, and for all $p' \in \mathcal{P}_<$, $f_{p'}^{UE} < \hat{f}_{p'}^{SO}$ and all paths only differ by links in E'. This is because the total flow between any origin-destination is larger in the SO flow by equation (2). Moreover, $\sum_{p \in \mathcal{P}_>}(f_p^{UE} - \hat{f}_p^{SO}) \leq \sum_{p' \in \mathcal{P}_<}(\hat{f}_p^{SO} - f_p^{UE})$ since the flow along non-constant latency links constrains the total flow. Move $\sum_{p \in \mathcal{P}_>}(f_p^{UE} - \hat{f}_p^{SO})$ units of flow from paths in set $\mathcal{P}_>$ to paths in set $\mathcal{P}_<$ in the SO flow $\hat{f}^{SO}$. Denote the new flow by f'. The total travel time for f' cannot increase because the flow has only increased on constant latency links, and the new flow does not exceed $\bar{f}_e^{SO}$ on any link. The total travel time also cannot have decreased because $\hat{f}^{SO}$ was an SO flow, so f' is also an SO flow. Continue this procedure until there does not exist a link $e \in E'$ for which f_e^{UE} exceeds the transformed SO flow. Then we have constructed an SO flow, f, in which, for all links $e \in E$, $f_e^{UE} \leq f_e$, a contradiction. □

Lemma 3. *For a network $\{G, R\}$, let f^* be a subflow of a feasible flow f. Then the flow f' such that $f'_e = f_e - f^*_e$ is also a subflow of f.*

Proof. First, $0 \leq f'_e \leq f_e$, by the definition of a subflow. Now set $r'_{s,t} = R(s,t) - r^*_{s,t}$. Then for all $s, t \in V^2$, $\sum_{e \in out(s)} f'_e - \sum_{e \in in(s)} f'_e = \sum_t (R(s,t) - r^*_{s,t}) = \sum_t r'_{s,t}$, and similarly for $\sum_{e \in in(t)} f'_e - \sum_{e \in out(t)} f'_e$ □

Theorem 2. *The optimal value of the UE linear program for a network instance $\{G, R\}$ is the maximum amount of UE agents that the network can support and achieve SO.*

Proof. First, by Theorem 1, there exists an SO flow such that the optimal UE subflow, f^{UE}, is a subflow of the SO flow, and by Lemma 3, there exists a subflow of compliant agents that can achieve the SO solution. Moreover, by the definition of the UE linear program and Lemma 2, the UE flow is only along zero reduced cost paths. By the definition of zero reduced cost paths, all UE agents are willing to take the assigned paths. Therefore, the SO solution is achievable with the UE flow, and there is some volume of UE flow that is equal to the objective of the UE linear program.

Now, suppose that there was another UE flow assignment, f', for which compliant flow could be assigned in such a way that the SO total system travel time was achieved and the total UE flow volume was larger than the value returned by the UE linear program. Note that this flow assignment (f') must be a subflow of some SO flow, f. Moreover, by the definition of UE flow and the fact that all paths in a SO solution are minimum marginal cost paths, all paths assigned with a UE flow greater than zero must be a zero reduced cost path. Therefore, the flow f' satisfies the equations (2)-(6), and since the UE linear program returns the optimal UE flow assignment under these constraints, this is a contradiction. □

While we've demonstrated that we can compute the *maximal* UE flow that permits an SO solution given the appropriate assignment of the compliant flow, it is likely that a more common problem would be to determine, for a given set of compliant agents, whether or not it is possible to achieve SO with that set. Our methodology also provides an answer to this question, as the following Corollary demonstrates.

Corollary 1. *For a given network instance $\{G, R\}$ and given a set of compliant demand, $r^C_{s,t}$, from each origin destination pair $s, t \in V^2$, there exists a compliant flow f^C such that the network achieves SO if and only if there exists an x^s_e for all $s \in V$ and $e \in E$ such that $r^{UE}_{s,t} = R(s,t) - r^C_{s,t}$ and x^s_e are a solution to the UE linear program.*

Proof. By Theorem 1, any solution to the UE linear program defines a subflow of an SO flow. Therefore, if $r^{UE}_{s,t}$ and x^s_e is a solution, there exists an assignment of the compliant flow that achieves SO.

Moreover, if there exists an assignment of the complaint flow, f^C, such that a UE subflow with demands $r^{UE}_{s,t}$ achieves system optimum, then the UE flow is only along zero reduced cost paths by definition of UE flow and SO, and the UE subflow is feasible. Therefore, the decomposed UE flow satisfies the constraints of the linear program. □

Flow Assignment for Compliant Agents

Given that we can now determine both the maximal amount of UE flow that a system can tolerate and achieve system optimum and, for a given set of compliant agents, whether or not a system can achieve optimum, we are only left with assigning the compliant flow to paths. This section tackles the question of how to assign paths to a, sufficiently large, set of compliant agents such that SO is achieved.

The methodology from the previous section immediately suggests a solution. Given a network instance $\{G, R\}$, suppose that we have compliant demand equal to $r^C_{s,t}$ for all $s, t \in V^2$. Then we must find a SO flow, f^{SO}, such that $r^C_{s,t}$ and $r^{UE}_{s,t} = R(s,t) - r^C_{s,t}$ permit subflows of the SO solution. Such a SO flow must exist by Theorem 1 and Corollary 1.

The first step is to compute the UE subflow, f^{UE}, given UE demand. From the previous section: this exists and is computationally tractable. Any feasible subflow, f^C, with demand $r^C_{s,t}$ such that the total flow along link e satisfies $f^C_e + f^{UE}_e \leq \bar{f}_e^{SO}$ has latency equal to the SO solution, and the flow $f^C_e + f^{UE}_e$, by Lemma 1, is an SO solution.

We can compute f^C with the following linear program:

$$\max_{f^C_e} 1$$

subject to

$$\sum_{e \in out(v)} f^C_e - \sum_{e \in in(v)} f^C_e = \sum_t (r^C_{v,t}) \qquad \forall\, v \in V$$

$$\sum_{e \in in(v)} f^C_e - \sum_{e \in out(v)} f^C_e = \sum_s (r^C_{s,v}) \qquad \forall\, v \in V$$

$$0 \leq f^C_e \leq \bar{f}^{SO}_e - f^{UE}_e \qquad \forall\, e \in E$$

We know that a solution to the above linear program exists and it can be computed tractably.

The final step is to decompose the compliant flow, f^C, into a per path assignment for each origin-destination pair

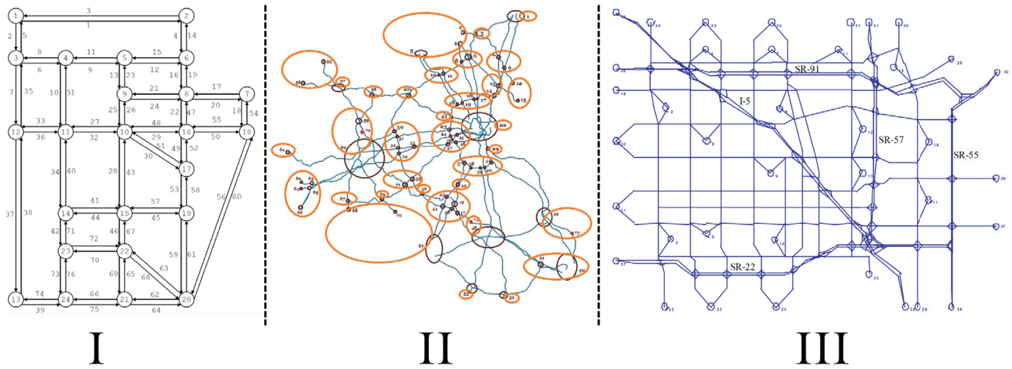

Figure 1: Three representative network topologies: I - Sioux Falls, SD, II - Eastern Massachusetts (Ellipsoids represent different zones), III - Anaheim, CA.

(s, t) in order to assign individual agents to a path. This can be done in time $O(|V||E|)$ using standard flow decomposition algorithms (see Section 3.5 of Ahuja, Magnanti, et. al. (1993) for a discussion).

Experimental Results

We are interested in the viability of *opt-in* micro-tolling schemes to more efficiently utilize road networks. As such, we haven undertaken an empirical study to investigate the minimal amount of compliant flow required for SO (r^*_{UE}) in six realistic traffic scenarios over actual road networks.

Scenarios

Each traffic scenario is defined by the following attributes:

1. The road network, $G(V, E)$, specifying the set of vertices and links where each link is affiliated with a length, capacity and speed limit. Networks are, following standard practice, partitioned into traffic analysis zones (TAZs) and each zone contains a node belonging to V called the centroid. All traffic originating and terminating within the zone is assumed to enter and leave the network at the centroid.
2. A trip table which specifies the traffic demand between pairs of centroids. The demand function R between nodes other than centroids is set to zero.

The following benchmark scenarios were chosen both for their diversity of topology and traffic volume and their widespread use within the traffic literature: Sioux Falls, Eastern Massachusetts, Anaheim, Chicago Sketch, Philadelphia, and Chicago-regional. All traffic scenarios are available at: https://github.com/bstabler/TransportationNetworks. Figure 1 depicts three representative network topologies (the three smallest networks).

The Traffic Model

A macroscopic model was used in order to evaluate traffic formation. Macroscopic models calculate the UE in a given scenario using algorithm B (Dial 2006). For all scenarios, the model assumed that travel times follow the *Bureau of Public Roads* (BPR) function (Moses and Mtoi 2017) with the commonly used parameters $\beta = 4$, $\alpha = 0.15$. The SO solution is computed by replacing the latency functions with $c'_e(x)$ and using algorithm B to obtain the equilibrium solution (Dial 1999). Since solving for the UE and SO solutions requires solving a convex program (Dial 2006), we only solve them to a certain precision. To measure convergence, given an assignment of agents to paths, we define the average excess cost (AEC) as the average difference between the travel times on paths taken by the agents and their shortest alternative path. The algorithm terminates when the AEC is less than 1E-12 minutes (except for Chicago-regional for which 1E-10 was used due to the size of the network). Therefore, a minimum marginal cost path is only a minimum up to a threshold.

A link e is defined to be zero reduced cost with respect to s if it carries flow originating at s in the SO solution (i.e., the link belongs to a minimum marginal cost path) and if the difference between the least latency path that includes e and the least latency unrestricted path, both leading from s to the head vertex of e, is less than a threshold T.

The threshold T is defined as follows: for each origin s and link e we calculate the least marginal cost path (c') leading from s to the head vertex of e at the SO solution. We do this once while restricting the path to include e and once without such restriction. The difference between these two values is stored and T is set to be the maximum of these difference across all the links and origins in the network.

Results

Table 1 presents the percentage of flow that must be compliant in order to guarantee an SO solution for six different traffic scenarios. Each scenario is affiliated with the number of vertices, links, and zones comprising the affiliated road network as well as the number of trips that make up the affiliated demand.

The columns "UE TTT" and "SO TTT" represent the total travel time (in minutes) over all agents for the case where 100% of the agents are controlled by the UE controller (UE

Scenario	Vertices	Links	Zones	Total Flow	*UE* TTT	*SO* TTT	% Improve	Threshold	% compliant
Sioux Falls	24	76	24	360,600	7,480,225	7,194,256	3.82	6.19E-11	**13.04**
Eastern MA	74	258	74	65,576	28,181	27,323	3.04	3.04E-13	**19.73**
Anaheim	416	914	38	104,694	1,419,913	1,395,015	1.75	8.05E-11	**19.76**
Chicago S	933	2,950	387	1,260,907	18,377,329	17,953,267	2.31	9.14E-10	**27.29**
Philadelphia	13,389	40,003	1525	18,503,872	335,647,106	324,268,465	3.39	4.20E-09	**49.59**
Chicago R	12,982	39,018	1790	1,360,427	33,656,964	31,942,956	5.09	4.14E-07	**53.34**

Table 1: Required fraction of compliant agents given as "% compliant" for different scenarios along with network specifications for each scenario: number of vertices, links and zones followed by the Total Travel Time (TTT) at UE (0% compliant agents) and SO (100% compliant agents). The percentage of improvement of the SO TTT over the UE TTT is given as "% improve".

solution) and when 100% of the agents are controlled by the SO controller (SO solution) respectively. The percentage of improvement in total travel time between UE TTT and SO TTT is also shown under "% improve".

The percentage of required compliant flow (formally $r^*_{UE}/|R|$ where $|R| = \sum_{s,t} R(s,t)$) as computed by the UE linear program (Definition 3) is presented for each scenario under "% compliant".[4]

The results suggest that as the size of the network (i.e., the number of nodes and vertices) increases, a greater fraction of compliant travelers are needed to ensure the network achieves system optimum. This appears to be due to an increasing number of used paths at the SO solution as the network size increases. As the number of paths grow, the set of zero reduced cost paths grows more slowly, and, therefore, a higher percentage of compliant agents is required.

Summary

This paper discussed a scenario where a set of agents traverse a congested network, while a centralized network manager is seeking to optimize the flow (minimizes total latency) by influencing the route assignment of a set of compliant agents. A methodology was presented for computing the minimal volume of traffic flow that needs to be compliant in order to reach a state of optimal traffic flow. Moreover, the methodology extends to inferring which agents should be compliant and how exactly the compliant agents should be assigned to paths. Experimental results demonstrate that the required percentage of agents that are compliant is small for some scenarios but can be greater than 50% in others.

Going forward, it would be worthwhile to explore the possibility of approximation algorithms for assigning compliant flow when the UE flow volume is too large to achieve a state of system optimum. Given that the optimal solution to this problem is known to be NP-hard, an efficient approximation algorithm would be a useful tool as opt-in network routing systems are implemented. Further, in order to limit the necessary opt-in incentives, there is work needed to develop systems that target particularly influential users to opt-in to these systems.

Acknowledgements

A portion of this work has taken place in the Learning Agents Research Group (LARG) at UT Austin. LARG research is supported in part by NSF (IIS-1637736, IIS-1651089, IIS-1724157), Intel, Raytheon, and Lockheed Martin. Peter Stone serves on the Board of Directors of Cogitai, Inc. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research. The authors would like to thank the Texas Department of Transportation for supporting this research under project 0-6838, Bringing Smart Transport to Texans: Ensuring the Benefits of a Connected and Autonomous Transport System in Texas. The authors would also like to acknowledge the support of the Data-Supported Transportation Operations & Planning Center and the National Science Foundation under Grant No. 1254921. Finally, we wish to acknowledge the help of Michael Levin and Josiah Hana to this project.

[4]Statistical analysis for Table 1 is not presented, as the macroscopic model is deterministic.

References

Ahuja, R. K.; Magnanti, T. L.; and Orlin, J. B. 1993. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1 edition.

Beckmann, M.; McGuire, C. B.; and Winsten, C. B. 1956. *Studies in the Economics of Transportation*. Yale University Press.

Dial, R. B. 1999. Network-optimized road pricing: Part i: a parable and a model. *Informs* 47.

Dial, R. B. 2006. A path-based user-equilibrium traffic assignment algorithm that obviates path storage and enumeration. *Transportation Research Part B: Methodological* 40(10):917–936.

Dietrich, B. 1969. Über ein Paradoxon aus der Verkehrsplanung. *Unternehmensforschung* 12:258–268.

Duffield, N. G.; Goyal, P.; Greenberg, A.; Mishra, P.; Ramakrishnan, K. K.; and van der Merive, J. E. 1999. A flexible model for resource management in virtual private networks. In *ACM SIGCOMM Computer Communication Review*, volume 29, 95–108. ACM.

Fingerhut, J. A.; Suri, S.; and Turner, J. S. 1997. Designing least-cost nonblocking broadband networks. *Journal of Algorithms* 24(2):287–309.

Haurie, A., and Marcotte, P. 1985. On the relationship between Nash-Cournot and Wardrop equilibria. *Networks* 15(3):295–308.

Immorlica, N.; Li, L. E.; Mirrokni, V. S.; and Schulz, A. S. 2009. Coordination mechanisms for selfish scheduling. *Theoretical Computer Science* 410(17):1589–1598.

Karmarkar, N. 1984. A new polynomial-time algorithm for linear programming. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, STOC '84, 302–311. New York, NY, USA: ACM.

Korilis, Y. A.; Lazar, A. A.; and Orda, A. 1997. Achieving network optima using stackelberg routing strategies. *IEEE/ACM Transactions on Networking (TON)* 5(1):161–173.

Moses, R., and Mtoi, E. T. 2017. Calibration and evaluation of link congestion functions. *Journal of Transportation Technologies* 4(2).

Numrich, J.; Ruja, S.; and Voß, S. 2012. Global navigation satellite system based tolling: state-of-the-art. *NETNOMICS: Economic Research and Electronic Networking* 13(2):93.

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

Rosenthal, R. W. 1973. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory* 2(1):65–67.

Roughgarden, T., and Tardos, É. 2002. How bad is selfish routing? *Journal of the ACM (JACM)* 49(2):236–259.

Roughgarden, T. 2004. Stackelberg scheduling strategies. *SIAM Journal on Computing* 33(2):332–350.

Schaller, B. 2010. New York Citys congestion pricing experience and implications for road pricing acceptance in the United States. *Transport Policy* 17(4):266–273.

Sharon, G.; Hanna, J. P.; Levin, M. W.; Rambha, T.; Albert, M.; Boyles, S. D.; and Stone, P. 2017a. 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-17)*.

Sharon, G.; Levin, M. W.; Hanna, J. P.; Rambha, T.; Boyles, S. D.; and Stone, P. 2017b. Network-wide adaptive tolling for connected and automated vehicles. *Transportation Research Part C* 84:142–157.

Yang, H., and Zhang, X. 2008. Existence of anonymous link tolls for system optimum on networks with mixed equilibrium behaviors. *Transportation Research Part B: Methodological* 42(2):99–112.

Yang, H.; Zhang, X.; and Meng, Q. 2007. Stackelberg games and multiple equilibrium behaviors on networks. *Transportation Research Part B: Methodological* 41(8):841–861.