Tolling for MDP Congestion Games

authors

I. Introduction

[1] Although a single population congestion game has been extensively analyzed(CITATION), we adopt the perspective of a global planner to enforce state space constraints without altering individual agent's behaviour.

Other's work: What have others done in this?

In order to enforce global constraints, we treat the routing game as a finite horizon, potential mean-field game that can be solved with a Markov Decision Process(MDP). Using the game reward as planner's control, it can be shown that state constraints can be reformulated as rewards, which in turn motivate game players to achieve a Wardrop Equilibrium that satisfies the constraints.

In particular, we reformulate state space constraints as a toll or flow based congestion control. Interestingly, it can be shown that a flow based control will not satisfy any state-space constraint by itself. The tolls are then applied to two scenarios:

1) Ride-share drivers attempting to maximize individual monetary return.
2) Parking scenario where all drivers are attempting to park as quickly as possible.

II. Notation

Behcet vs Yoyo vs Einsum

III. Problem Formulation

A. Markov decision processes

Consider the following linear programming formulation of finite horizon MDP [2]

$$\begin{aligned}
\underset{\substack{y_t\\ t\in\mathcal{T}}}{\min.} & \sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} y_t(s,a)c_t(s,a)\\
\text{s.t.} & \sum_{a\in\mathcal{A}} y_{t+1}(s',a) = \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\Gamma(s'\mid s,a)y_t(s,a),\\
& \sum_{a\in\mathcal{A}} y_0(s,a) = p_0(s),\\
& y_t(s,a)\geq 0, \quad \forall s,s'\in\mathcal{S}, a\in\mathcal{A}, t\in\mathcal{T}
\end{aligned} \tag{1}$$

where $\mathcal{S}$ and $\mathcal{A}$ denote respectively the set of states and actions, $\mathcal{T}=\{0,\ldots,|\mathcal{T}|\}$ denote the time step, $y_t(s,a)$ and $c_t(s,a)$ denote respectively the probability and cost of state-action pair (s,a), $\Gamma(s'|s,a)$ denotes transition probability from state s to state s' given action a, $p_0(s)$ denotes the probability that the decision processes starts from state s. Notice that the initial and dynamics constraints in (1) implies that

$$\begin{aligned}
\sum_{s\in\mathcal{S}} y_0(s,a) &= \sum_{s\in\mathcal{S}} p_0(s) = 1,\\
\sum_{s'\in\mathcal{S}} y_{t+1}(s',a) &= \sum_{s'\in\mathcal{S}}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\Gamma(s'\mid s,a)y_t(s,a)\\
&= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} y_t(s,a).
\end{aligned}$$

Hence by induction we know that the nomalization constraints on $y_t(s,a)$ are automatically satisfied for all $t\in\mathcal{T}$.

The KKT conditions of problem (1) are composed of three components: primal and dual feasibility

$$\begin{aligned}
\sum_{a\in\mathcal{A}} y_{t+1}(s',a) &= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\Gamma(s'\mid s,a)y_t(s,a),\\
\sum_{a\in\mathcal{A}} y_0(s,a) &= p_0(s),\\
y_t(s,a) &\geq 0\\
\mu_t(s,a) &\geq 0,
\end{aligned} \tag{2}$$

complementary slackness

$$\mu_t(s,a)y_t(s,a) = 0 \tag{3}$$

and vanishing gradient

$$\begin{aligned}
V_T(s) &= c_T(s,a) - \mu_T(s,a),\\
V_t(s) &= c_t(s,a) + \sum_{s'\in\mathcal{S}}\Gamma(s'\mid s,a)V_{t+1}(s')\\
&\quad - \mu_t(s,a),
\end{aligned} \tag{4}$$

It is straightforward to verify that the following *value iteration* satisfies conditions in (3) and(4) simultaneously,

$$\begin{aligned}
V_T(s) &= \underset{a\in\mathcal{A}}{\min.}\ c_T(s,a)\\
V_t(s) &= \underset{a\in\mathcal{A}}{\min.}\ c_t(s,a) + \sum_{s'\in\mathcal{S}}\Gamma(s'\mid s,a)V_{t+1}(s')
\end{aligned} \tag{5}$$

If we define the Q-value function for a state-action pair

(s,a) as

$$\begin{aligned} Q_T(s,a) =& c_T(s,a) \\ Q_t(s,a) =& c_t(s,a) + \sum_{s'\in\mathcal{S}} \Gamma(s' \mid s,a) \min_{a\in\mathcal{A}} Q_{t+1}(s',a) \end{aligned} \tag{6}$$

Then the dual variable $\mu_t(s,a)$ gives the inefficiency of $Q_t(s,a)$, *i.e.*,

$$\mu_t(s,a) = Q_t(s,a) - V_t(s,a).$$

B. Markov decision processes with density and flow rate constraints

Consider the following constrained Markov decision processes

$$\begin{aligned} \min_{\substack{y_t\\ t\in\mathcal{T}}}.& \sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} y_t(s,a)c_t(s,a) \\ \text{s.t.}& \sum_{a\in\mathcal{A}} y_{t+1}(s',a) = \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \Gamma(s' \mid s,a) y_t(s,a), \\ & \sum_{a\in\mathcal{A}} y_0(s,a) = p_0(s), \\ & y_t(s,a) \geq 0, \\ & \underline{p}_t(s) \leq \sum_{a\in\mathcal{A}} y_t(s,a) \leq \overline{p}_t(s), \\ & \underline{r}_t(s) \leq \sum_{a\in\mathcal{A}} (y_{t+1}(s,a) - y_t(s,a)) \leq \overline{r}_t(s), \\ & \forall s,s'\in\mathcal{S}, a\in\mathcal{A}, t\in\mathcal{T} \end{aligned} \tag{7}$$

where $[\underline{p}_t(s), \overline{p}(s)]$ and $[\underline{r}_t(s), \overline{r}_t(s)]$ denote respectively the desired interval of population and flow rate in state s. The KKT conditions of problem (7) is as follows. Primal and dual feasibility: in addition to (2) we have the following

$$\underline{\tau}_t(s) \geq 0, \overline{\tau}_t(s) \geq 0 \quad \underline{\delta}_t(s) \geq 0, \overline{\delta}_t(s) \geq 0, \tag{8}$$

complementary slackness: in addition to (3), we have the following

$$\begin{aligned} & \overline{\tau}_t(s) \left(\sum_{a\in\mathcal{A}} y_t(s,a) \right) = \overline{\tau}_t(s)\overline{p}_t(s), \\ & \underline{\tau}_t(s) \left(\sum_{a\in\mathcal{A}} y_t(s,a) \right) = \underline{\tau}_t(s)\underline{p}_t(s), \\ & \overline{\delta}_{t+1}(s) \left(\sum_{a\in\mathcal{A}} (y_{t+1}(s,a) - y_t(s,a)) - \overline{r}_t(s) \right) = 0, \\ & \underline{\delta}_{t+1}(s) \left(\underline{r}_t(s) - \sum_{a\in\mathcal{A}} (y_{t+1}(s,a) - y_t(s,a)) \right) = 0, \end{aligned} \tag{9}$$

and vanishing gradient

$$\begin{aligned} V_T(s) =& c_T(s,a) - \mu_T(s,a) + \overline{\tau}_T(s) - \underline{\tau}_T(s) \\ & + \overline{\delta}_T(s) - \underline{\delta}_T(s), \\ V_t(s) =& c_t(s,a) + \sum_{s'\in\mathcal{S}} \Gamma(s' \mid s,a) V_{t+1}(s') \\ & - \mu_t(s,a) + \overline{\tau}_t(s) - \underline{\tau}_t(s) \\ & + (\overline{\delta}_t(s) - \overline{\delta}_{t+1}(s)) - (\underline{\delta}_t(s) - \underline{\delta}_{t+1}(s)). \end{aligned} \tag{10}$$

Theorem 1 (Exact penalty)**.** *[3, Prop. 5.4.5] Let $y_t^\star, \mu_t^\star, V_t^\star, \overline{\tau}_t^\star, \underline{\tau}_t^\star, \overline{\delta}_t^\star, \underline{\delta}_t^\star$ satisfy the KKT conditions* (2)*,* (3)*,* (8)*,* (9) *and* (10)*. The solutions to problem* (7) *are the same as the follow optimization problem*

$$\begin{aligned} \min_{\substack{y_t\\ t\in\mathcal{T}}}.& \sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} y_t(s,a)c_t(s,a) \\ & + \sum_{s\in\mathcal{S}} \Bigg(\overline{\tau}_t(s) \left[\sum_{a\in\mathcal{A}} y_t(s,a) - \overline{p}_t(s) \right]_+ \\ & + \underline{\tau}_t(s) \left[\underline{p}_t(s) - \sum_{a\in\mathcal{A}} y_t(s,a) \right]_+ \\ & + \overline{\delta}_t(s) \left[\sum_{a\in\mathcal{A}} (y_{t+1}(s,a) - y_t(s,a)) - \overline{r}_t(s) \right]_+ \\ & + \underline{\delta}_t(s) \left[\underline{r}_t(s) - \sum_{a\in\mathcal{A}} (y_{t+1}(s,a) - y_t(s,a)) \right]_+ \Bigg) \\ \textit{s.t.}& \sum_{a\in\mathcal{A}} y_{t+1}(s',a) = \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \Gamma(s' \mid s,a) y_t(s,a), \\ & \sum_{a\in\mathcal{A}} y_0(s,a) = p_0(s), \\ & y_t(s,a) \geq 0, \quad \forall s,s'\in\mathcal{S}, a\in\mathcal{A}, t\in\mathcal{T} \end{aligned} \tag{11}$$

where

$$\begin{aligned} & \overline{\tau}_t(s) > \overline{\tau}_t^\star(s), \quad \underline{\tau}_t(s) > \underline{\tau}_t^\star(s), \\ & \overline{\delta}_t(s) > \overline{\delta}_t^\star(s), \quad \underline{\delta}_t(s) > \underline{\delta}_t^\star(s). \end{aligned} \tag{12}$$

IV. Convergence

V. Numerical examples

A. Simulation: Ride-sharing game

To demonstrate the usage of exact penalty as a toll generation method, we consider a game scenario that ride-sharing drivers might play in Seattle, seeking to optimize their fares. The city is abtracted an undirected, connect graph, where neighbourhoods are nodes, and drivers may traverse adjacent nodes as specified by edges (shown in Figure 1).

At each node, the driver can choose from several actions. The first action, a_r, is to wait for a random rider

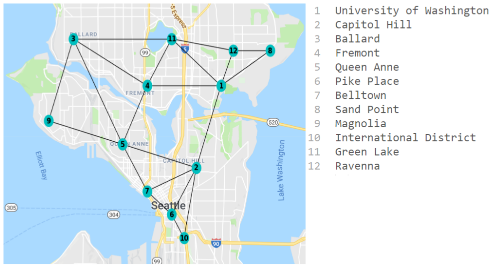

Fig. 1: (a) Illustration of Seattle Neighborhoods .

and transition to whatever node that rider wants to go to. We assume that the driver will always pick up a rider, although it may take long time if there already exists many drivers doing the same. In addition, the passenger's destination will appear as random to the driver. At each node, a constant percentage of riders want to travel to each of the other nodes. This model would be useful for ride sharing services where drivers are simply assigned riders or taxi drivers who queue at transportation hubs.

The driver can also choose to transition without a rider. The action of transitioning to node s' without a rider is $a_{s'}$. In general, this would result in the driver paying the travel costs without receiving a fare (however, there could also be a small possibility that the driver will find a customer along the way).

The rewards for taking each action (and then transitioning from state s to state s') is given by

$$R_t[s', s, a_r]\Big(z_t\Big) = M_t[s', s] \tag{13}$$

$$- C_t^{\text{trav}}[s', s] - C_t^{\text{wait}} \cdot z_t[s, a_r] \tag{14}$$

$$R_t[s', s, a_{s'}] = -C_t^{\text{trav}}[s', s] \tag{15}$$

where $M_t[s', s]$ is the monetary cost for transitioning from state s to s', $C_t^{\text{trav}}[s', s]$ is the travel cost from state s to s', and C_t^{wait} is the coefficient of the cost of waiting for a rider. We compute these various parameters as

$$M_t[s', s] = \big(\text{Rate}\big) \cdot \big(\text{Dist}\big) \tag{16}$$

$$C_t^{\text{trav}}[s', s] = \tau \underbrace{\big(\text{Dist}\big)}_{\text{mi}} \underbrace{\big(\text{Vel}\big)^{-1}}_{\text{hr/mi}} + \underbrace{\begin{pmatrix}\text{Fuel}\\\text{Price}\end{pmatrix}}_{\$/\text{gal}} \underbrace{\begin{pmatrix}\text{Fuel}\\\text{Eff}\end{pmatrix}^{-1}}_{\text{gal/mi}} \underbrace{\big(\text{Dist}\big)}_{\text{mi}} \tag{17}$$

$$C_t^{\text{wait}} = \tau \cdot \Bigg(\underbrace{\begin{matrix}\text{Customer}\\\text{Demand Rate}\end{matrix}}_{\text{rides/hr}}\Bigg)^{-1} \tag{18}$$

and τ is a time-money tradeoff parameter which we take to be \$27/hr.

The values independent of specific transitions are listed in the table below.

Rate	Velocity	Fuel Price	Fuel Eff
\$6 /mi	8 mph	\$2.5/gal	20 mi/gal

B. Circling for parking

Another application of this model is determining the optimal strategies for urban drivers looking for places to park. A sample set of city blocks shown in Figure 2a. We solve the problem on the dual graph as it allows us more freedom to restrict certain transitions (left turns, U-turns, etc.). The actions associated with each state are either park, a_p, or make one one of the allowed turns stipulated in the following table (dependent on the intersection).

Park	Straight	L-turn	R-turn	U-turn
a_p	a_s	a_l	a_r	a_u

The expected waiting time for a given member of the population is related to the rate at which spots become available and the probability that a driver will park given there is a parking spot open. This last component depends on the number of other drivers on the block face. For the probability of finding an empty spot, we draw inspiration from a queuing model where customers are selected randomly from a queue. Each block face s has c_s spots and each spot becomes available at a rate of γ_s (in units of car/min). The total cost of driving by each block face is given by

$$R_t[s, \ell_j(x) = M_j p_j + \Big((C_j)_{\text{wait}} + M_j\Big)(1 - p_j) \tag{19}$$

Here, M_j is the cost of parking and $(C_j)_{\text{wait}}$ is the cost of waiting for a space which we take to be the $(C_j)_{\text{wait}} = \tau \Delta t_j$ where τ is a time vs. money tradeoff parameter and Δt_j is the average amount of time spent on block face j. The probability of an individual driver getting a parking spot, p_j, is given by

$$p_j = \Big(1 - e^{-c_j \gamma_j \Delta t_j}\Big) \frac{1}{1 + x_j} \tag{20}$$

This equation consists of two parts. The first is the probability that a space opens up in the time the driver spends on the block face. The second part is the probability that an individual driver of the population on the edge gets that space. This second term is 1 when there is 0 mass on the edge, $\frac{1}{2}$ when there is a mass of 1 other driver on the edge, $\frac{1}{3}$ when there is a mass of 2 other drivers on the edge, etc.

The full loss function can be rewritten as

$$\ell_j(x) = \underbrace{\Big(M_j + (C_j)_{\text{wait}}\Big)}_{a_j} \underbrace{-(C_j)_{\text{wait}}\Big(1 - e^{-c_j \gamma_j \Delta t_j}\Big)}_{b_j} \frac{1}{1 + x_j} \tag{21}$$

The potential function is then given by

$$F(x) = \sum_j \int_0^{x_j} \ell_j(u)\ du = \sum_j \Big[a_j x_j + b_i \ln\big(1 + x_j\big) \Big] \tag{22}$$

We use the following parameter values (Table. V-B) for all streets, while the number of parking spaces are given by Table. V-B.

Δt_j (min)	γ_j (1/hr)	M_j ($)	τ ($/min)
0.5	1/120	5	0.5

Street	1-6	7-8	9-10	11-14	15-16	17-18	19 -24
Num. spaces (c_j)	20	10	20	10	20	10	20

The streets are numbered in Figure 2a.

VI. Conclusions

References

[1] E. Altman, *Constrained Markov decision processes*. CRC Press, 1999, vol. 7.

[2] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[3] D. P. Bertsekas, *Nonlinear programming*. Athena scientific Belmont, 1999.

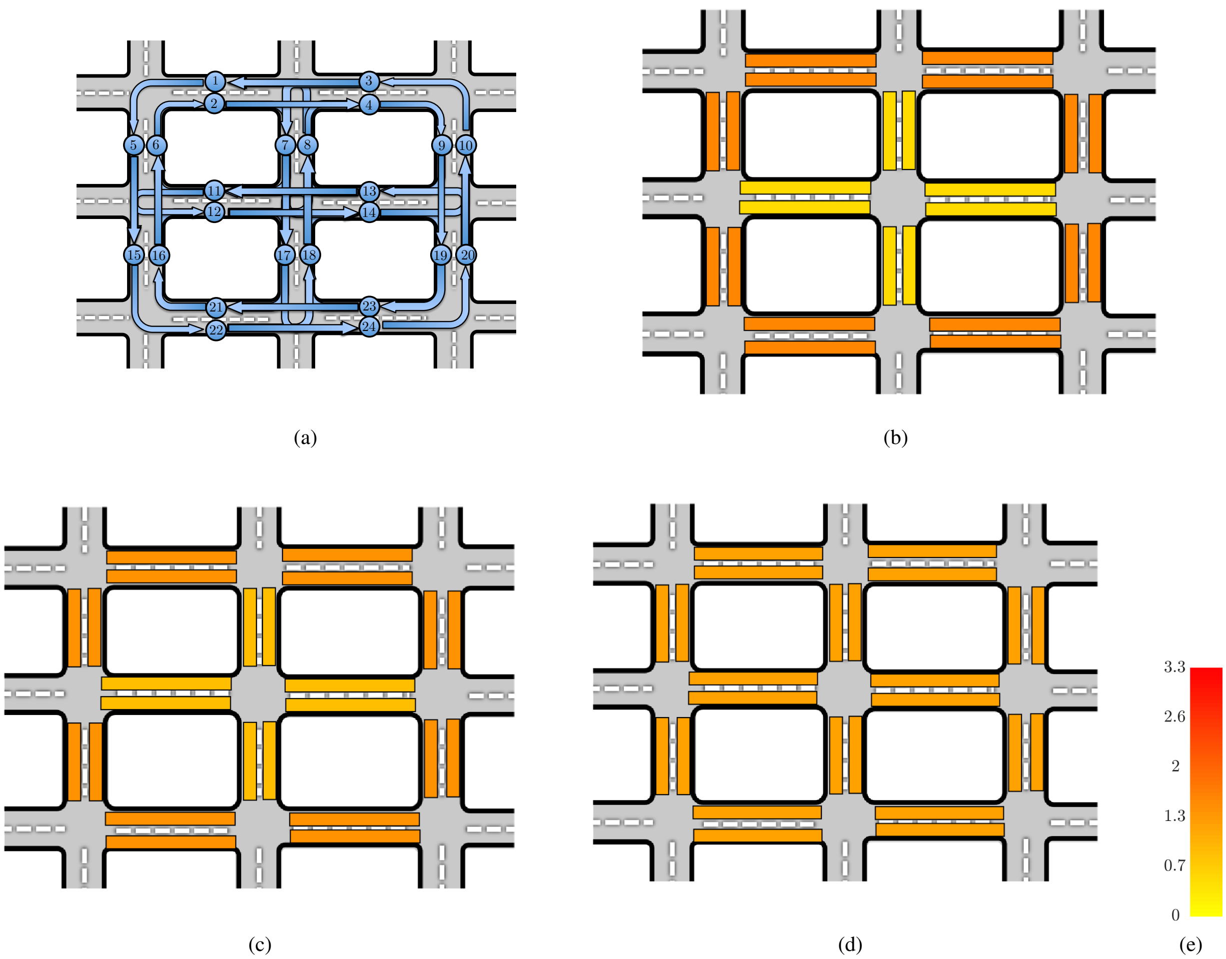

Fig. 2: (a) Block faces and allowed transitions. (b) Wardrop equilibrium. (c) Social optimum. (d) Uniform turns. (e) Scale (cars).