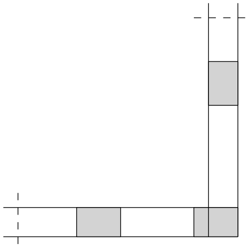
Traffic Scheduling

Jeroen van Riel

March 2024

Single intersection

Safe trajectories at isolated intersection



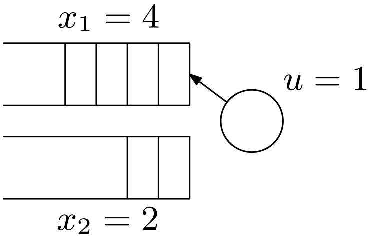
- trajectories $x(t)$ satisfy contraints

$$0 \leq x'(t) \leq v_m \tag{1a}$$
$$|x''(t)| \leq a_m \tag{1b}$$

- no collisions between vehicles
- problem of optimal control is essentially reduced to finding an optimal policy in a two-queue polling system Miculescu and Karaman (2016)

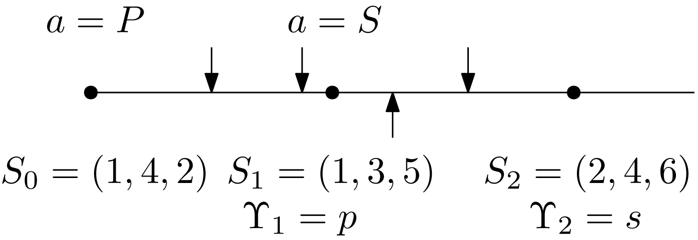
Two-queue polling system

- two queues with customers arriving as Poisson(λ_i) process
- single server alternates queues, location is $u \in \{1, 2\}$
- number of customers in queue i is denoted as x_i
- service takes p time, switch takes s time



Semi-Markov Decision Process

- Markov decision process with sojourn times Υ_n
- action space $\mathcal{A} = \{P, S, I\}$
- state space $\mathcal{S} = \{1, 2\} \times \mathbb{N}^+ \times \mathbb{N}^+$
- serving and switching are non-preemptive, so skip arrivals while serving or switching



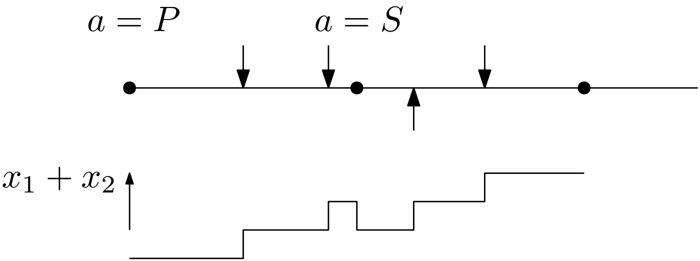
Semi-Markov Decision Process

- holding costs for customers

$$r(t) = -(x_1(t) + x_2(t)) \tag{2}$$

- total discounted reward

$$\phi_\beta = \left[\int_0^\infty e^{-\beta t} r(t) dt\right] \tag{3}$$



Optimal policies

- Hofri and Ross (1987)
  - theorem: there is an optimal exhaustive policy
  - conjecture: there is an optimal double-threshold policy
- curse of modeling
  - approximate the SMDP, then dynamic programming
  - Q-learning or similar model-free methods

General arrival processes

- extend the state space to include time since last arrival

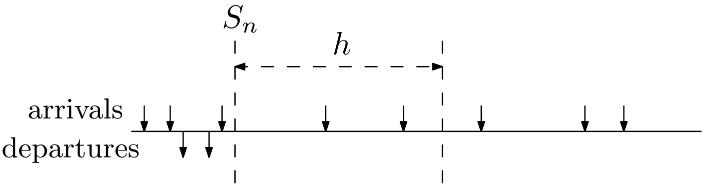
$$(u, x_1, x_2, \tau_1, \tau_2) \in \mathcal{S} = \{1, 2\} \times \mathbb{N}^+ \times \mathbb{N}^+ \times \mathbb{R}^+ \times \mathbb{R}^+ \tag{4}$$

- explicitly discretize state space
- parametric function approximation (neural net)

Knowledge of future arrivals

Knowledge of future arrivals

- extreme cases
  - full knowledge ($h = \infty$) $\Longrightarrow$ planning
  - no knowledge ($h = 0$)



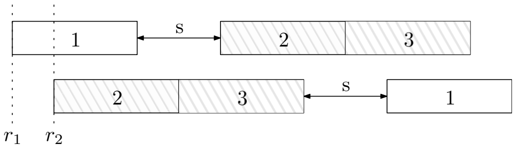
Full knowledge of future (planning)

- general solution method via MILP formulation
- r_j is arrival time
- y_j is crossing time, $C_j = y_j + p$ is completion time
- o_{jl} is order of vehicles j and l (binary decision variable)
- $\mathcal{C}$ is set of precedence constraints
- $\bar{\mathcal{D}}$ is set of conflicts (between vehicles of distinct lanes)

$$
\begin{aligned}
\text{minimize} \quad & \sum_{j=1}^{n} C_j && && \text{(5a)} \\
\text{s.t.} \quad & r_j \leq y_j && \text{for all } j = 1, \ldots, n, && \text{(5b)} \\
& C_j \leq y_l && \text{for all } (j, l) \in \mathcal{C}, && \text{(5c)} \\
& C_j + s \leq y_l + o_{jl} M && \text{for all } (j, l) \in \bar{\mathcal{D}}, && \text{(5d)} \\
& o_{jl} \in \{0, 1\} && \text{for all } (j, l) \in \bar{\mathcal{D}}. && \text{(5e)}
\end{aligned}
$$

Full knowledge of future (planning)

- example where waiting is necessary



No knowledge of future

- polling system discussed at beginning
- conjecture: waiting is required to obtain optimal policy
- action space $\mathcal{A} = \{P, S, I(\infty)\} \cup \{I(\delta) : \delta > 0\}$

Multiple intersections

Job Shop

- extension of MILP to multiple intersections is trivial
- similar to job-shop machine scheduling
- no guarantees for existence of safe trajectories
- finite buffer space between intersections

End-to-end methods

- define very general policy space
- use model-free reinforcement learning to find policies

References

Micha Hofri and Keith W. Ross. On the Optimal Control of Two Queues with Server Setup Times and Its Analysis. *SIAM Journal on Computing*, 16(2):399–420, April 1987. ISSN 0097-5397. doi: 10.1137/0216029.

David Miculescu and Sertac Karaman. Polling-systems-based Autonomous Vehicle Coordination in Traffic Intersections with No Traffic Signals, July 2016.