本科毕业论文答辩
Bachelor Thesis Defense

自动驾驶车辆的分布式协调方案

Tree Structured Traffic Coordination with Decentralized Optimization

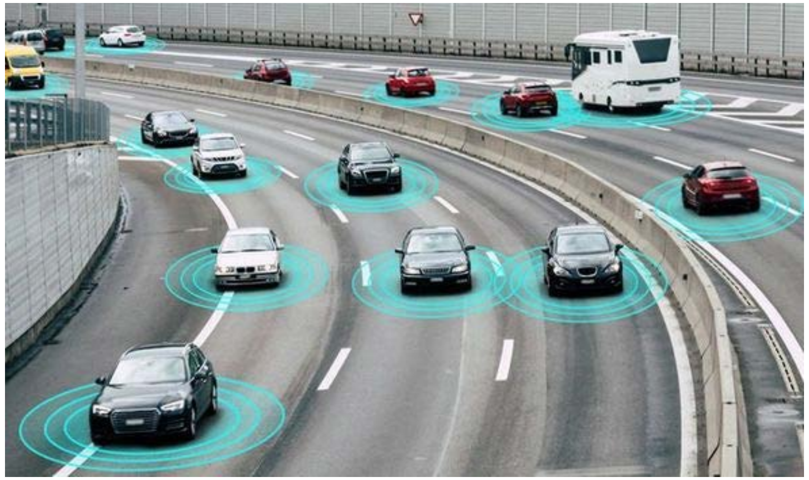
方笑天 Xiaotian Fang
Supervisor: Boris Houska
June 5, 2020

Outline

上海科技大学
ShanghaiTech University

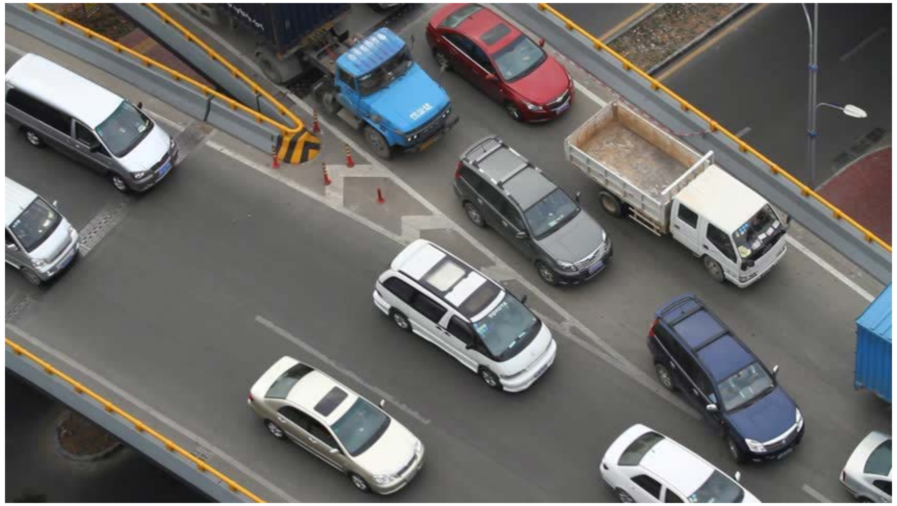
Motivation & Background

coordinate vehicles

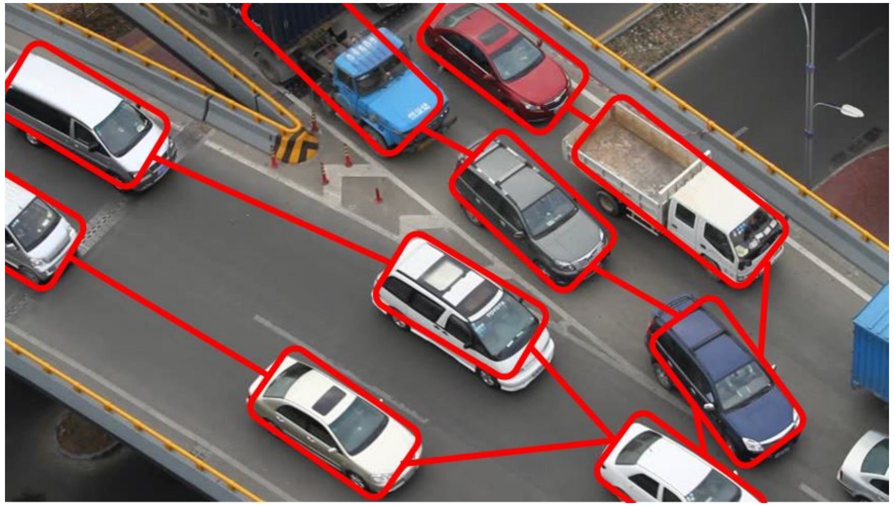


- Alleviate traffic congestion
- Autonomous driving

Motivation & Background



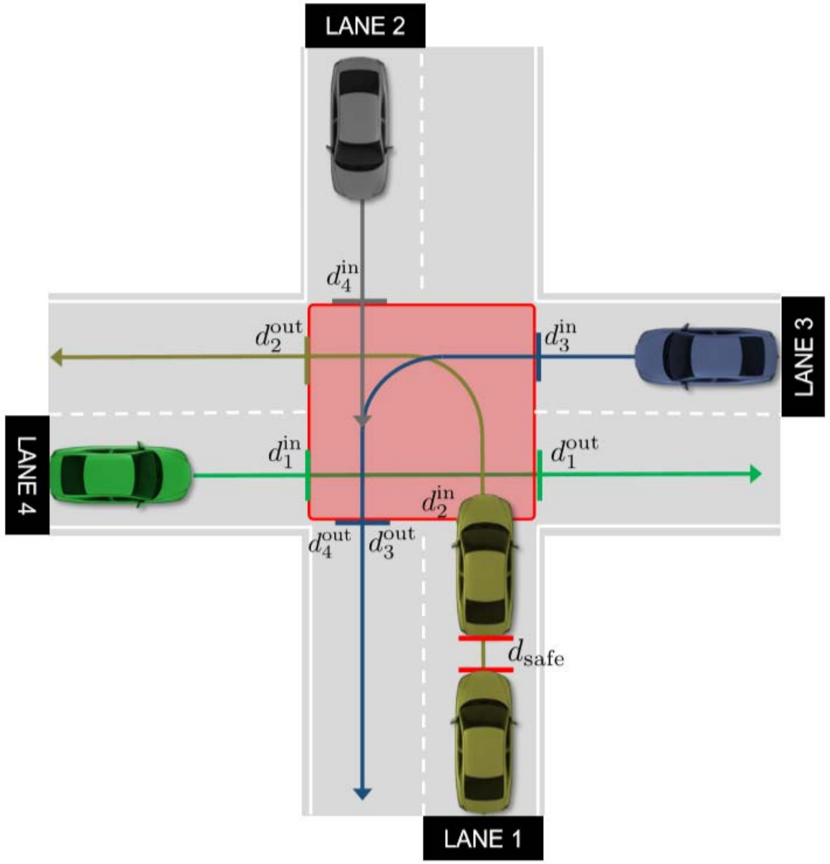
Traffic Coordination



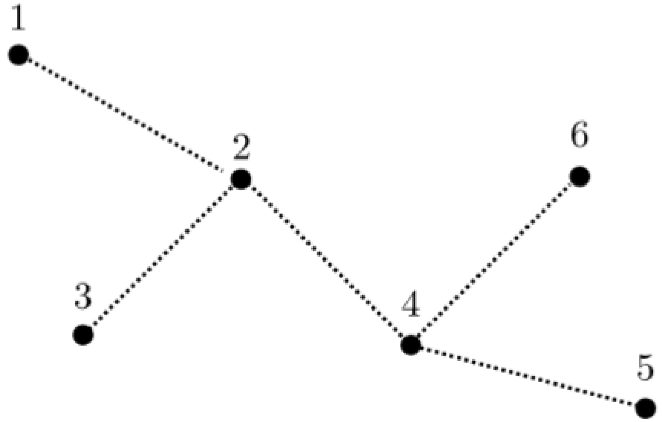
Network Structure
Tree-structured Network

Motivation & Background

上海科技大学
ShanghaiTech University



+



$$V^* = \min_x \sum_{i \in \mathcal{N}} F_i(x_i) \quad \text{s.t.} \quad \begin{cases} \forall (i,j) \in \mathcal{E}^+, \\ S_{i,j} x_i = S_{j,i} x_j \ . \end{cases}$$

Y. Jiang, M. Zanon, R. Hult, and B. Houska. IFAC 2017
Distributed algorithm for optimal vehicle coordination at traffic intersections.

J. Shi, Y. Zheng, Y. Jiang, M. Zanon, R. Hult, and B. Houska. CDC 2018
Distributed control algorithm for vehicle coordination at traffic intersections.

Y. Jiang, D. Kouzoupis, H.Yin, M. Diehl, and B. Houska.
Decentralized Optimization over Tree Graphs.
arXiv preprint arXiv:1910.09206.

System Model

上海科技大学 ShanghaiTech University

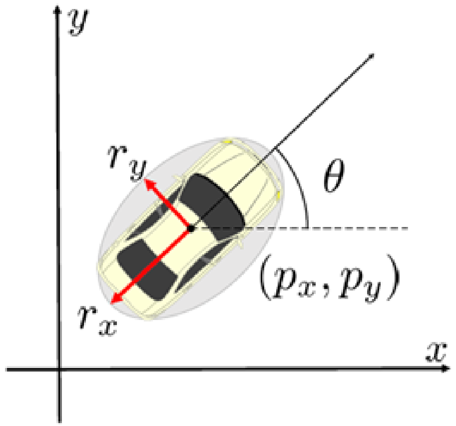
- Vehicle Kinematics

$$x_{i,k+1} = \underbrace{\begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix}}_{A} \cdot x_{i,k} + \underbrace{\begin{bmatrix} 0 \\ \frac{h^2}{2} \end{bmatrix}}_{B} \cdot u_{i,k}$$

$$x_{i,k} = \begin{bmatrix} s_{i,k} \\ v_{i,k} \end{bmatrix}$$

$$v_i^{min} \leq v_{i,k} \leq v_i^{max}$$
$$u_i^{min} \leq u_{i,k} \leq u_i^{max}$$

- Vehicle Shape



Shape ellipse: $\mathcal{E}(p_{i,k}, \mathcal{Q}(\theta_{i,k}))$

- Local to Global coordinates

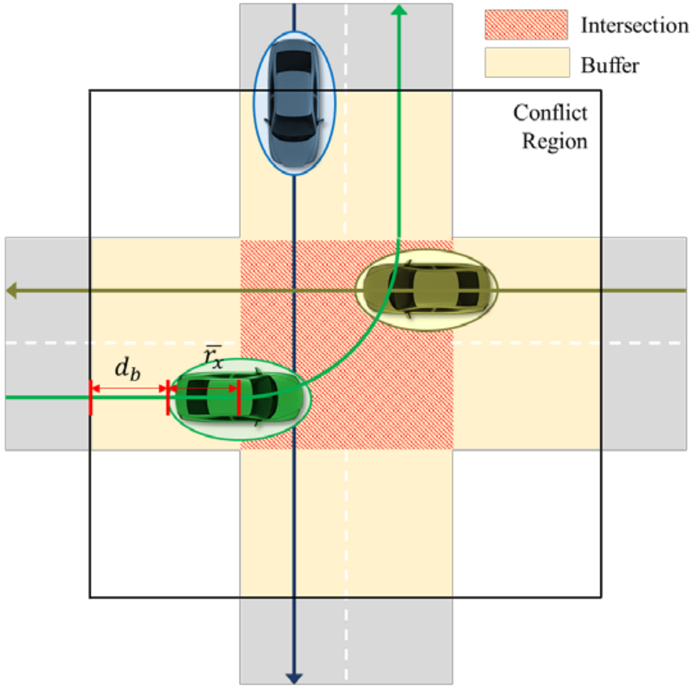
$$p_i(s_{i,k}) = \begin{bmatrix} p_{i,x}(s_{i,k}) \\ p_{i,y}(s_{i,k}) \end{bmatrix} = \begin{bmatrix} \sum_{l=0}^{n_p} \alpha^{x,l} B_n^{x,l}(s_{i,k}) \\ \sum_{l=0}^{n_p} \alpha_n^{y,l} B_n^{y,l}(s_{i,k}) \end{bmatrix}$$

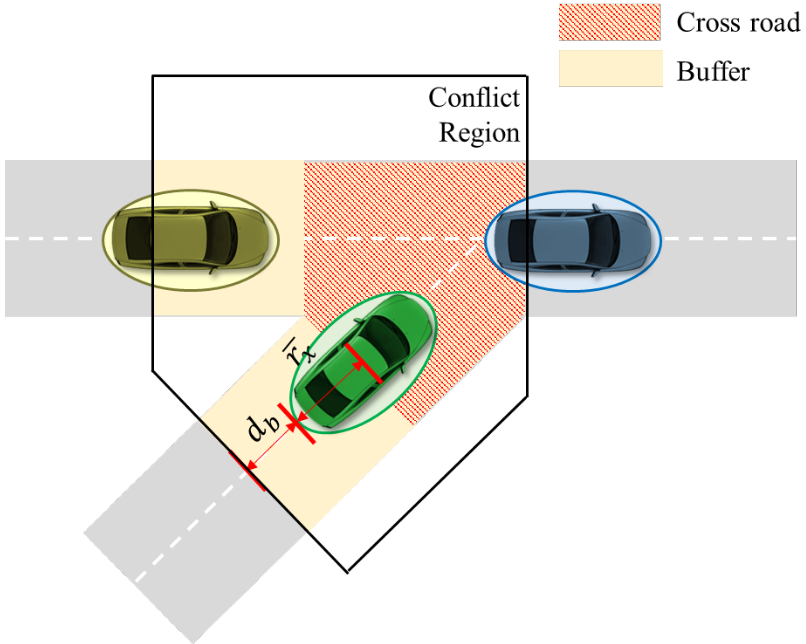
$$\mathcal{Q}(\theta_{i,k}) = \begin{bmatrix} r_x^2 \cos^2\theta_{i,k} + r_y^2 \sin^2\theta_{i,k} & 0.5sin2\theta_{i,k}(r_x^2 - r_y^2) \\ 0.5sin2\theta_{i,k}(r_x^2 - r_y^2) & r_x^2 \sin^2\theta_{i,k} + r_y^2 \cos^2\theta_{i,k} \end{bmatrix}$$

Collision Avoidance
Conflict Region

$$\mathcal{C}_i = [s_i^{in}, s_i^{out}]$$

outside: Rear-end Collision
inside: Rear-end and Side Collision





$$l_b = d_b + \bar{r}_x$$

d_b: minimum braking distance
$\bar{r}_x$: longest semi-axis

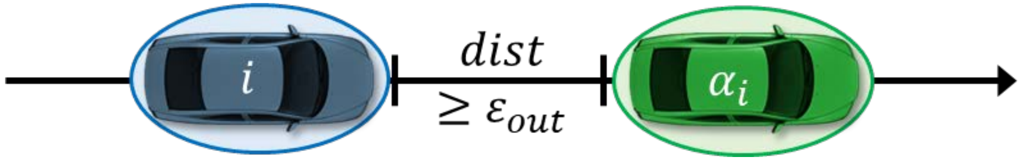
Collision Avoidance

Rear-end and Side Collision Avoidance

- Rear-end Collision Avoidance

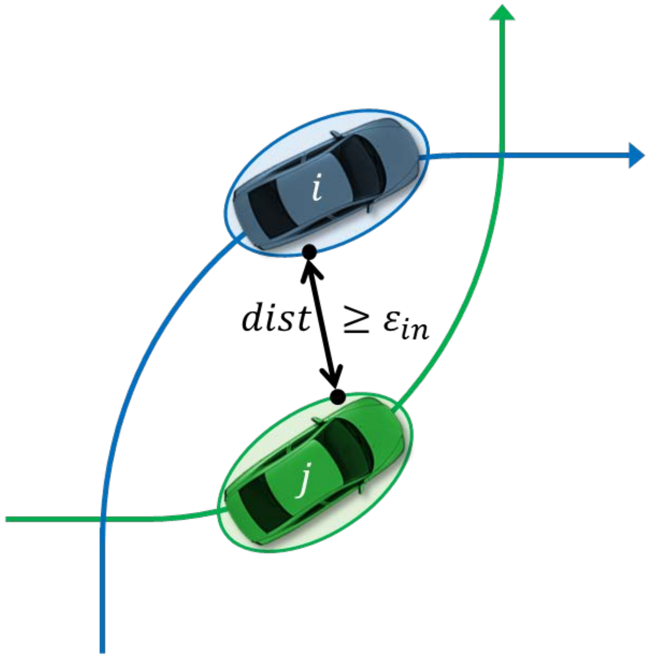
  Assumption: Paths are straight outside conflict region

  $$\delta_{i,a_i,k} = s_{i,k} - s_{a_i,k} - r_{i,x} - r_{a_i,x} \geq \varepsilon_{out}$$



- Side Collision Avoidance

  $$dist\left(\mathcal{E}\left(p_{i,k}, \mathcal{Q}_i(\theta_{i,k})\right), \mathcal{E}\left(p_{j,k}, \mathcal{Q}_j(\theta_{j,k})\right)\right) \geq \varepsilon_{in}$$



Collision Avoidance

Distance between Ellipses

$$dist\big(\mathcal{E}(q_1, Q_1), \mathcal{E}(q_2, Q_2\,)\big) \geq d_{12}$$

$$\Updownarrow$$

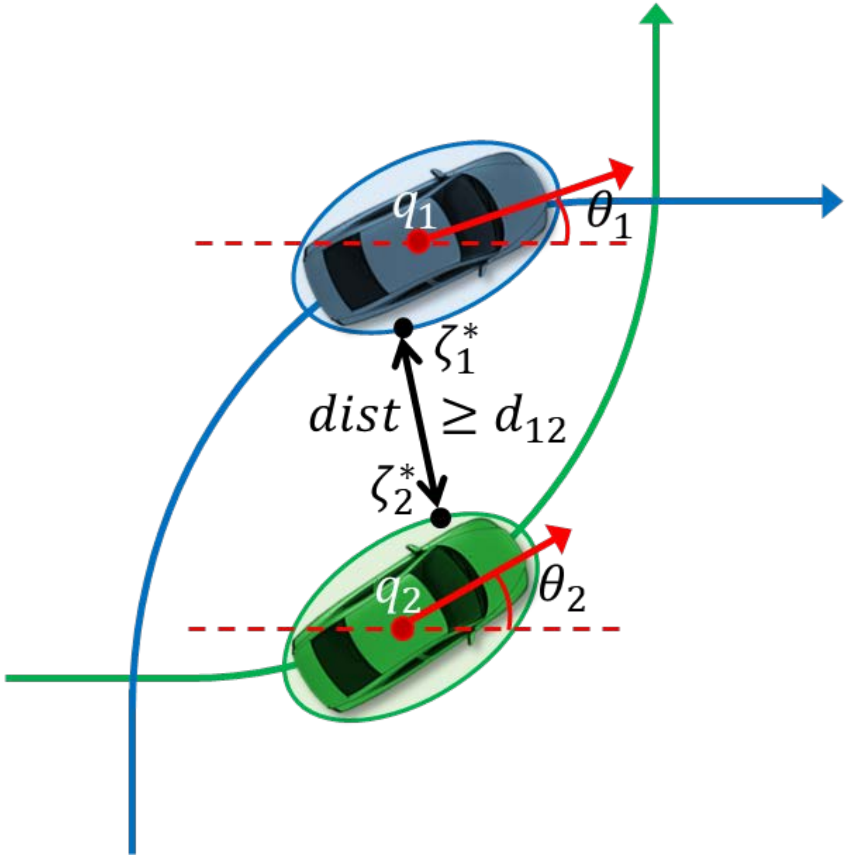
exist $\delta = (\delta_1, \delta_2) \in \mathbb{R}^2$ satisfy

$$\begin{cases} ||\zeta_1(\delta) - \zeta_2(\delta)||_2 \geq d_{12} \\ (\zeta_1(\delta) - q_1)^T Q_1^{-1}(\zeta_1(\delta) - q_1) = 1 \\ (\zeta_2(\delta) - q_2)^T Q_2^{-1}(\zeta_2(\delta) - q_2) = 1 \\ \delta \geq 0 \end{cases}$$

where

$$\zeta_1(\delta) = [(I + \delta_2 Q_2^{-1})(I + \delta_1 Q_1^{-1}) - I]^{-1} [\delta_1(I + \delta_2 Q_2^{-1}) Q_1^{-1} q_1 + \delta_2 Q_2^{-1} q_2]$$

$$\zeta_2(\delta) = [(I + \delta_1 Q_1^{-1})(I + \delta_2 Q_2^{-1}) - I]^{-1} [\delta_2(I + \delta_1 Q_1^{-1}) Q_2^{-1} q_2 + \delta_1 Q_1^{-1} q_1]$$



Problem Formulation

Centralized Formulation

上海科技大学 ShanghaiTech University

$$\min_{x,u} \sum_{i\in\mathcal{V}} J_i(x_i,u_i) = \left\|x_i - x_{ref}\right\|_Q^2 + \|u_i\|_R^2$$

Objective

s.t. $\forall i \in \mathcal{V}$

$$x_{i,0} = \hat{x}_i \,, x_{i,k+1} = Ax_{i,k} + Bu_{i,k} \,, \ \forall k \in \mathbb{Z}_0^{K-1}$$

Kinematics

$$h_i\left(x_{i,k}, u_{i,k}\right) \le 0 \,, \ \forall k \in \mathbb{Z}_0^K$$

State and Control Constraints

$$\begin{cases} \text{if } \hat{x}_i \notin \mathcal{C}_i \,, \ \forall k \in \mathbb{Z}_0^K \\ \delta_{i,a_i,k} \le \varepsilon_{out} \end{cases}$$

Rear-end Collision Avoidance

$$\begin{cases} \text{if } \hat{x}_i \in \mathcal{C}_i \,, \forall j \in \mathcal{V}_i^b \,, \ \forall k \in \mathbb{Z}_0^K \\ G\left(\delta_{i,j,k}, p_{i,k}, p_{j,k}, \theta_{i,k}, \theta_{j,k}, \varepsilon_{in}\right) \le 0 \end{cases}$$

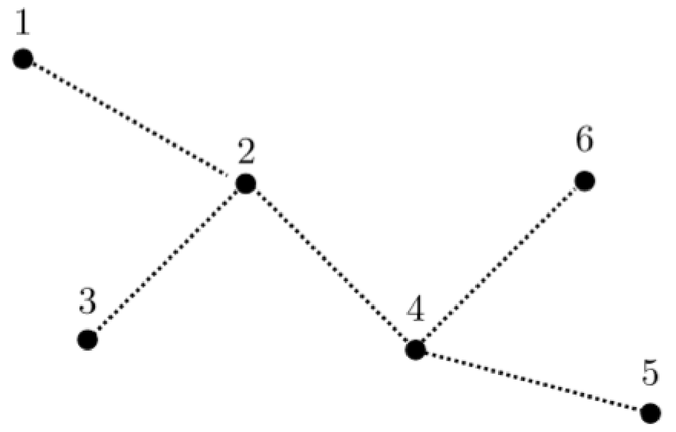
Inside Conflict Region Collision Avoidance

Problem Formulation

Tree-structed Optimization Problem

上海科技大学 ShanghaiTech University

Tree graph $(\mathcal{N}, \mathcal{E})$: $\begin{cases} \text{Nodes set, } \mathcal{N} = \{1, \dots, N\} \\ \text{Edges set, } \mathcal{E} \subseteq \mathcal{N} \times \mathcal{N} \end{cases}$



$\mathcal{N}_i$: neighbors of i−th node

$\mathcal{L} \coloneqq \{i \in \mathcal{N} \mid |\mathcal{N}_i| \leq 1\}$

$\mathcal{E}^+ \coloneqq \{(i,j) \in \mathcal{E} \mid i < j\}$

$\mathcal{C}_i \coloneqq \{j \in \mathcal{N}_i \mid j > i\}$

$\pi_i \coloneqq \{j \in \mathcal{N}_i \mid j < i\}$

$$V^* = \min_x \sum_{i \in N} F_i(x_i) \quad \text{s.t.} \begin{cases} \forall (i,j) \in \mathcal{E}^+ \\ S_{i,j} x_i = S_{j,i} x_j \end{cases}$$

Problem Formulation

Dynamic Programming

上海科技大学 ShanghaiTech University

Backward sweep: leaves to root

- Each node receives V_j from children $\mathcal{C}_i$. Then, construct V_i by

$$V_i(p) = \min_{x_i} F_i(x_i) + \sum_{j \in \mathcal{C}_i} V_i(S_{i,j} x_i) \quad s.t.\ S_{\mathrm{i},\pi_i} x_i = p \tag{1}$$

  and sent it to parent π_i.

- The root solves

$$V^* = \min_{x_1} F_1(x_1) + \sum_{j \in \mathcal{C}_1} V_j(S_{1,j} x_1)$$

Forward sweep: root to leaves

- Each node receives $S_{\pi_i,i} x_{\pi_i}$ from parent and solve (1) by set $p = S_{\pi_i,i} x_{\pi_i}$ to get x_i^*, then send it to their children.

Disadvantages:
- V_i required to be constructed explicitly for p.
- Greatly affected by curse of dimensionality.

Problem Formulation

Model function

上海科技大学 ShanghaiTech University

- For $i \in \mathcal{L} \backslash \{1\}$, the model function

$$W_{i,\pi_i}(p) \approx V_i(p) = \min_{x_i} F_i(x_i) \quad \text{s.t. } S_{i,\pi_i} x_i = p$$

is constructed.

- For $i \notin \mathcal{L}$, we construct

$$W_{i,k}(p) \approx \Omega_{i,k}(p) \approx V_i(p)$$

based on

$$\Omega_{i,k}(p) = \min_{x_i} F_i(x_i) + \sum_{j \in \mathcal{N}_i \backslash \{k\}} W_{j,i}(S_{i,j} x_i) \quad \text{s.t. } S_{i,k} x_i = p$$

Example (cubic model):

$$W_{i,k}(p) = \frac{1}{2} p^T H_{i,k} p + g_{i,k}^T + \sigma_{i,k} \left\| p - S_{i,k} y_i^* \right\|^3 + \text{const}$$

with $$H_{i,k} = \nabla^2 \Omega_{i,k}(S_{i,k} y_i^*), \quad g_{i,k} = \nabla \Omega_{i,k}\left(S_{i,k} y_i^*\right) - H_{i,k} S_{i,k} y_i^*$$

Problem Formulation

Multi-sweep Method

上海科技大学 ShanghaiTech University

Initialization

- Set all nodes to backward mode.
- If $i \neq 1$, choose an initial model function $W_{\pi_i,i}$.

Repeat on every node $i \in \mathcal{N}$:

- **IF backward**
  - Wait for $W_{j,i}$ from all children $j \in \mathcal{C}_i$.
  - If $i = 1$, switch to forward and **BREAK**
  - Solve

$$\min_{y_i} F_i(y_i) + \sum_{j \in \mathcal{N}_i} W_{j,i}(S_{i,j} y_i) \qquad (2)$$

    and construct $W_{i,\pi_i} \approx \Omega_{i,\pi_i}$.
  - Send W_{i,π_i} to parent and witch to forward.
- **ELSE (forward)**
  - If $i \neq 1$, wait for $W_{\pi_i,i}$ from the parent.
  - Solve (2) and construct $W_{i,j} \approx \Omega_{i,j}$.
  - Send $W_{i,j}$ to all nodes $j \in \mathcal{C}_i$, switch to backward.

Problem Formulation

Tree-structed Reformulation

$$\min_{x,u} \sum_{i\in\mathcal{V}} J_i(x_i,u_i) = \|x_i - x_{ref}\|_Q^2 + \|u_i\|_R^2$$

$$\begin{aligned}
\text{s.t. } & \forall i \in \mathcal{V} \\
& x_{i,0} = \hat{x}_i \,, x_{i,k+1} = Ax_{i,k} + Bu_{i,k} \,,\ \forall k \in \mathbb{Z}_0^{K-1} \\
& h_i(x_{i,k}, u_{i,k}) \le 0 \,,\ \forall k \in \mathbb{Z}_0^K \\
& \begin{cases} \text{if } \hat{x}_i \notin \mathcal{C}_i \,,\ \forall k \in \mathbb{Z}_0^K \\ \delta_{i,a_i,k} \le \varepsilon_{out} \end{cases} \\
& \begin{cases} \text{if } \hat{x}_i \in \mathcal{C}_i \,, \forall j \in \mathcal{V}_i^b \,,\ \forall k \in \mathbb{Z}_0^K \\ G(\delta_{i,j,k}, p_{i,k}, p_{j,k}, \theta_{i,k}, \theta_{j,k}, \varepsilon_{in}) \le 0 \end{cases}
\end{aligned}$$

- construct new $x_i = \left(s_i, v_i, [s_j]_{j\in\mathcal{N}_i}, u_i\right)$.
- use single shooting to eliminate kinematic constraints.
- decoupled objective can be given by

$$F_i(x_i) = \min_{x_i} J_i(x_i) + \sum_{j\in\mathcal{N}_i} \mu(s_i, s_j)$$

with log barrier function μ.

$$\Rightarrow \quad V^* = \min_x \sum_{i\in N} F_i(x_i) \quad \text{s.t. } \begin{cases} \forall (i,j) \in \mathcal{E}^+ \\ S_{i,j}x_i = S_{j,i}x_j \end{cases}$$

Distributed MPC Scheme

Algorithm Distributed MPC for Traffic Coordination

Initialization:

1) choose suitable discretization accuracies $h \geq 0$ and $K \geq 0$.
2) use B-splines to fit predetermined paths for all $i \in \mathcal{V}$.

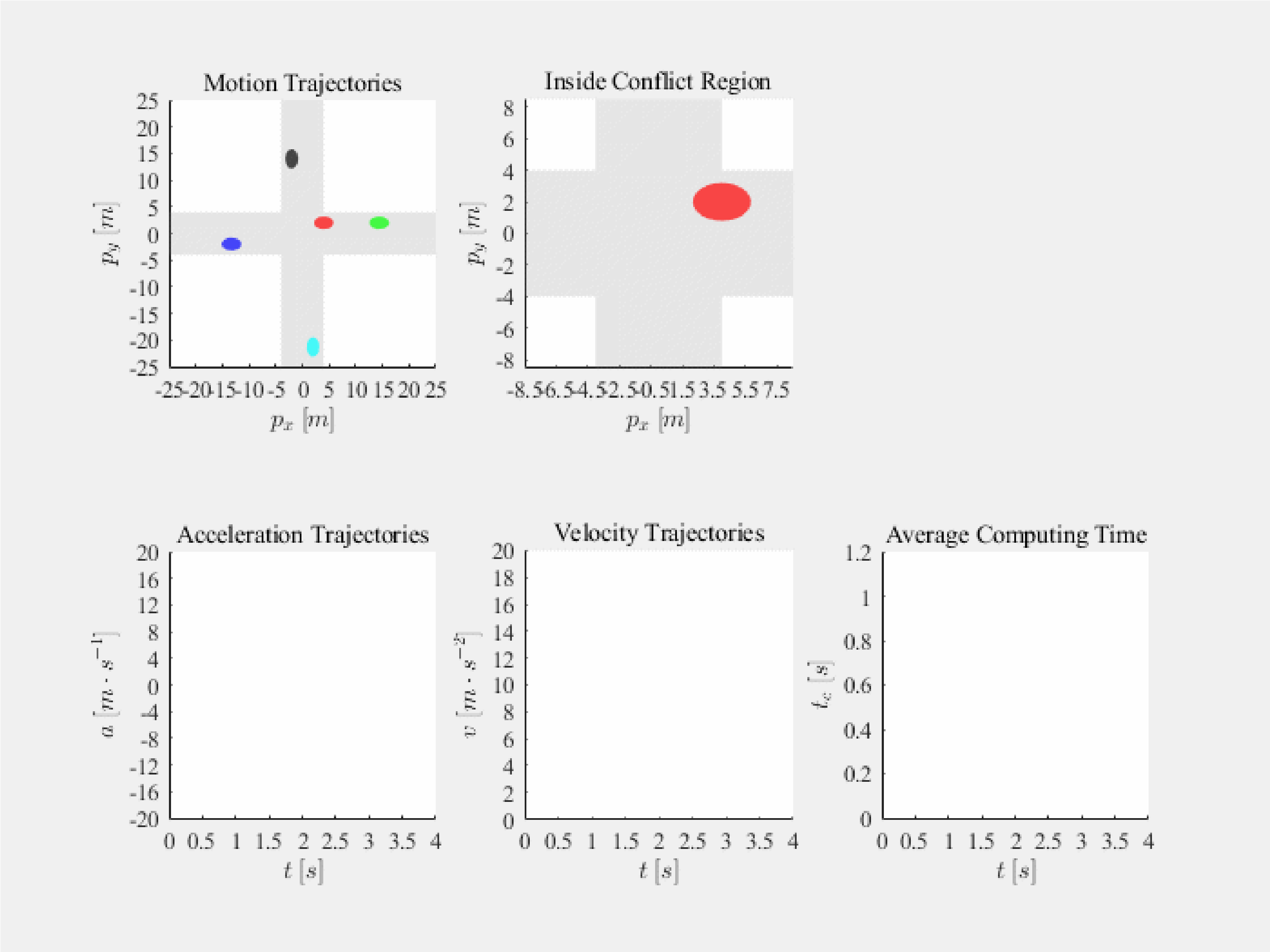
Online:

1) each vehicle i measures $\widehat{x}_i$.
2) for vehicle i whose $s_{i,0} \notin \mathcal{C}_i$, communicate with its neighbors to update a_i.
3) for vehicle i whose $s_{i,0} \in \mathcal{C}_i$, communicate with its neighbors to update $\mathcal{V}_i^b$.
4) construct tree graph of all coordinating vehicles.
5) each vehicle i construct x_i, objective $F_i(x_i)$ and connectivity matrixes $S_{i,j}$.
6) use multi-sweep method to solve tree-structed decentralized problem.
7) each vehicle i applies the control $u_{i,0}^*$ during the first time interval $[0, h]$.

Case Study

Intersection Scenario

上海科技大学
ShanghaiTech University

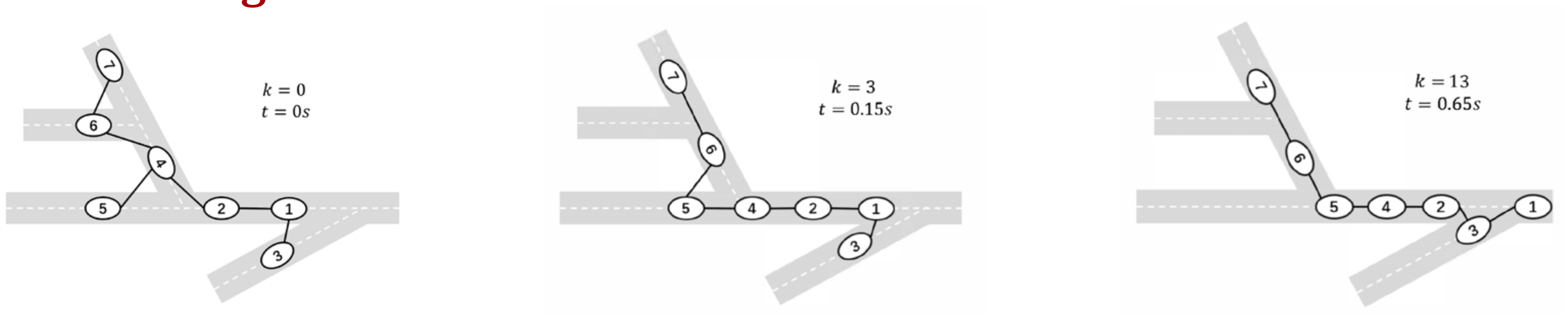


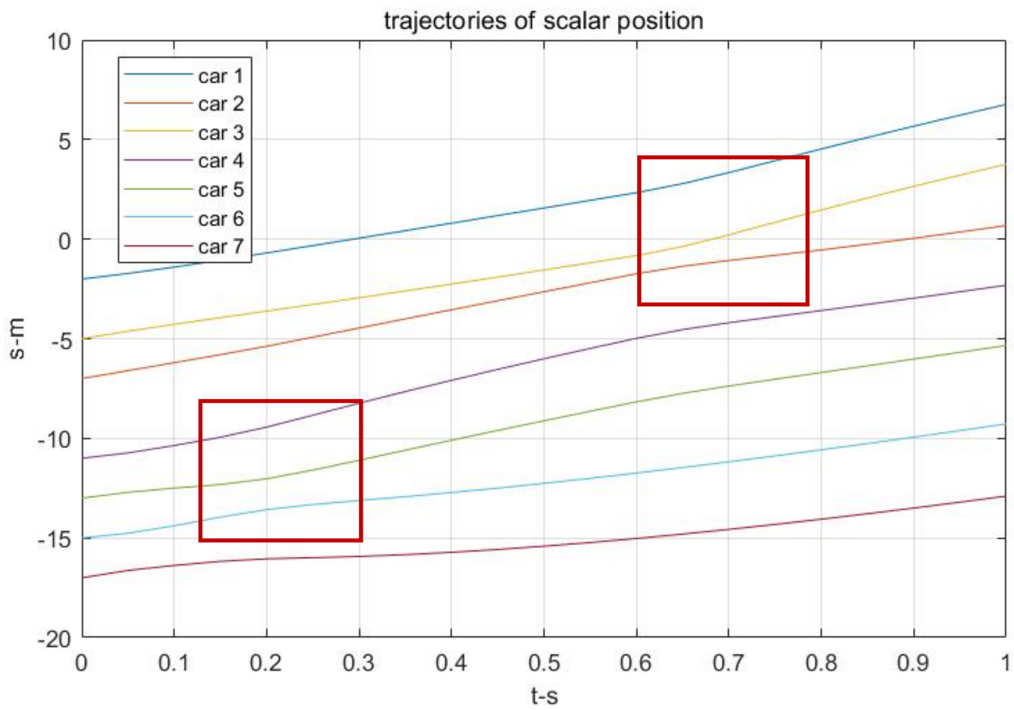
Parameters			
r_x	$2m$	r_y	$1.3m$
ε_{in}	$0.2m$	ε_{out}	$1m$
Q	I	R	$0.2I$
v_{ref}	$10m/s$	v_{max}	$20m/s$
a_{min}	$-30m/s^2$	a_{max}	$30m/s^2$

- Relatively complex scenario
- Good utilization in space
- High cost in computation

Case Study

Lanes merge Scenario





- No considering the shape of vehicles
- Relatively simple scenario
- High efficiency in computation
  - one backward + forward for QP problem
  - computational cost increases in linear rate

Conclusion

上海科技大学 ShanghaiTech University

01 Space Utilization
#agents per unit area

Balance ⟺

Efficiency
computational cost

02 Network structed decentralized optimization algorithm is pretty suitable for (traffic) coordination.

03 **Feasibility** maybe be the biggest problem for traffic coordination. A complete strategy for the whole traffic system (not some parts) is needed.

上海科技大学
ShanghaiTech University

Thank you for your attention.

Sincere thanks to Prof. Houska,
Dr. Jiang and Jiahe Shi.

Q&A

END