2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Cooperative Lane Change Motion Planning of Connected and Automated Vehicles: A Stepwise Computational Framework *

Bai Li, Yue Zhang, Youmin Zhang, and Ning Jia

Abstract—**This paper focuses on the scheme of cooperative lane change motion planning of multiple connected and automated vehicles, so as to minimize the time for lane change while penalizing large steering angles subject to hard collision avoidance constraints. Nominally this scheme should be formulated in a centralized way with the constraints of all the vehicles considered simultaneously. In order to facilitate the numerical solving process of this centralized optimization problem, we propose a stepwise computation framework. Starting with a sub-problem with all of the collision avoidance constraints removed, a sequence of sub-problems are defined by adding back the removed collision avoidance constraints gradually until the original problem takes shape in the end. The optimum of one sub-problem is always used as the initial guess when solving the next sub-problem. This iterative process continues until the optimum of the original problem is obtained. In this way, the difficulties in the original centralized problem are divided into multiple parts, and every progress made to address the partial difficulties is "solidified" by the initial guess.**

I. Introduction

The incessant developments in automated driving have brought opportunities to improve traffic conditions in terms of increasing traffic throughput, enhancing mobility, and reducing traffic accidents, congestion, and greenhouse gas emissions [1]. An automated driving system mainly consists of three modules, namely, environment perception, motion planning, and control strategies. Herein, motion planning plays a critical role in making driving decisions, thereby reflecting the intelligence level of an automated driving system [2]. In general, obtaining the optimal motion planning result costs a long CPU time, which makes online solution unavailable. Thus the prevalent motion planners commonly sacrifice solution optimality for computational speed [3]. Although the real-time performance of a motion planning approach is essential for on-line applications, the solution quality deserves attention as well.

Multi-vehicle motion planning (MVMP) is a new paradigm in the connected and automated vehicles (CAVs) domain. MVMP is challenging because large volumes of interactions among the vehicles need to be simultaneously considered. Although there have been many MVMP methods, most of them are inherently doing joint planning rather than simultaneous planning, so that the cooperation potential in a multi-vehicle team cannot be fully exploited [4]. Concretely, few of the existing MVMP methods can 1) consider the kinematic capability of multiple vehicles simultaneously; 2) fully avoid collisions among the vehicles; and 3) handle different tasks/scenarios in a unified way [1,5]. These limitations motivate us to find the generic and optimal solutions to MVMP problem, whereby the cooperative potential among the CAVs is thoroughly exploited. This paper focuses on the cooperative lane change application. Lane change serves as a basic driving behavior in performing higher-level actions such as merging and bypassing. Many car accidents result from inappropriate lane change operations because the non-cooperative driving behaviors of human drivers are difficult to recognize, predict, and react [6].

This study aims to find offline optimal solutions to generic MVMP problems in terms of cooperative lane changes. To this end, a stepwise computation framework is proposed, which is the main contribution of this paper. Although the proposed method cannot be used on the road, the derived offline solutions serve as benchmarks for evaluating the solution quality of other prevalent motion planners.

II. MVMP Problem Formulation

The concerned MVMP problem is about minimizing the time and steering energy for cooperative lane changes, subject to the kinematic constraints, and environmental constraints.

A. Vehicle Kinematics

The kinematics of vehicle i_j is described as

$$\begin{cases} \dfrac{\mathrm{d}x_{i\_j}(t)}{\mathrm{d}t} = v_{i\_j}(t)\cdot\cos\theta_{i\_j}(t) \\ \dfrac{\mathrm{d}y_{i\_j}(t)}{\mathrm{d}t} = v_{i\_j}(t)\cdot\sin\theta_{i\_j}(t) \\ \dfrac{\mathrm{d}v_{i\_j}(t)}{\mathrm{d}t} = a_{i\_j}(t) \\ \dfrac{\mathrm{d}\theta_{i\_j}(t)}{\mathrm{d}t} = \dfrac{v_{i\_j}(t)\cdot\tan\phi_{i\_j}(t)}{\mathrm{L_w}} \\ \dfrac{\mathrm{d}\phi_{i\_j}(t)}{\mathrm{d}t} = \omega_{i\_j}(t) \end{cases}, \ t\in[t_0,t_\mathrm{f}]. \quad (1)$$

Herein, t denotes time, t_0 denotes the process starting time,

* This work was supported by the National Natural Science Foundation of China under Grant 61573282 and 71671123, Shaanxi Province Natural Science Foundation under Grant 2015JZ020, and the Natural Sciences and Engineering Research Council of Canada.

Bai Li is with the Shaanxi Key Laboratory of Complex System Control and Intelligent Information Processing, Xi'an University of Technology, Xi'an 710048, China (e-mail: libai@zju.edu.cn).

Yue Zhang is with the Division of Systems Engineering, and Center for Information and Systems Engineering, Boston University, Boston, USA (e-mail: joycez@bu.edu).

Youmin Zhang is with the Department of Mechanical, Industrial and Aerospace Engineering, Concordia Institute of Aerospace Design and Innovation, Concordia University, Montreal, Canada (phone: 514-848-2424 ext. 5225; fax: 514-848-3175; e-mail: ymzhang@encs.concordia.ca).

Ning Jia is with the Institute of Systems Engineering, Tianjin University, Tianjin, China (e-mail: jia_ning@tju.edu.cn).

t_f denotes the terminal moment, $\left(x_{i\_j}(t), y_{i\_j}(t)\right)$ locates the rear wheel axis midpoint of vehicle i_j at time t (Fig. 1), $\theta_{i\_j}(t)$ denotes the orientation angle, $v_i(t)$ refers to the linear velocity of point (x_{i_j}, y_{i_j}), $a_{i\_j}(t)$ denotes the linear acceleration, $\phi_{i\_j}(t)$ denotes the steering angle of the front wheels, and $\omega_{i\_j}(t)$ denotes the derivative of $\phi_{i\_j}(t)$. In addition, as depicted in Fig. 1, the shape of the vehicle is defined in terms of the wheelbase $\mathrm{L_w}$, the front overhang length $\mathrm{L_f}$, the rear overhang length $\mathrm{L_r}$, and the vehicle width $\mathrm{L_b}$.

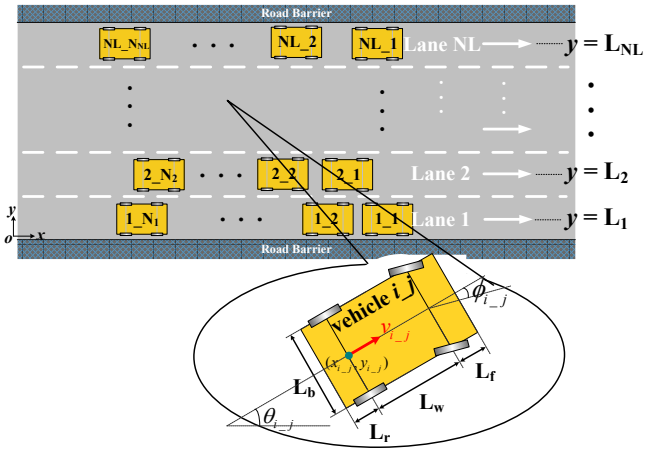

Fig. 1. Schematics on vehicle kinematics, and road scenario.

Suppose there are $\mathrm{N_V}$ CAVs traveling on NL lanes at $t = t_0$ (Fig. 1), the jth vehicle on the ith lane at t_0 is denoted as vehicle i_j, and there are totally N_k vehicles traveling on lane k at t_0 ($k = 1, \ldots, \mathrm{NL}$).

In addition to the vehicle kinematics, bounded constraints for the state/control variables are imposed:

$$\begin{cases} \left|a_{i\_j}(t)\right| \le \mathrm{a_{max}} \\ 0 \le v_{i\_j}(t) \le \mathrm{v_{max}} \\ \left|\phi_{i\_j}(t)\right| \le \Phi_{\mathrm{max}} \\ \left|\omega_{i\_j}(t)\right| \le \Omega_{\mathrm{max}} \end{cases}, \; t \in [t_0, t_f], \; i = 1, \ldots, \mathrm{NL}, j = 1, \ldots, \mathrm{N}_i, \quad (2)$$

where $\mathrm{a_{max}}$, $\mathrm{v_{max}}$, Φ_{max}, and Ω_{max} denote the upper boundaries of $\left|a_{i\_j}(t)\right|$, $v_{i\_j}(t)$, $\left|\phi_{i\_j}(t)\right|$, and $\left|\omega_{i\_j}(t)\right|$ during the entire lane change process, respectively.

B. Lane Change Scenario

At the initial moment t_0, the configuration of vehicle i_j is given:

$$\begin{aligned} &\left[x_{i\_j}(t_0), y_{i\_j}(t_0), v_{i\_j}(t_0), a_{i\_j}(t_0), \theta_{i\_j}(t_0), \omega_{i\_j}(t_0)\right] = \\ &\left[\mathrm{x}_{i\_j}, \mathrm{y}_{i\_j}, \mathrm{c}_{i\_j}, \mathrm{d}_{i\_j}, \mathrm{e}_{i\_j}, \mathrm{f}_{i\_j}\right], \; i = 1, \ldots, \mathrm{NL}, j = 1, \ldots, \mathrm{N}_i. \end{aligned} \quad (3)$$

Denoting the target lane of vehicle i_j as lane k, we expect vehicle i_j to run in a state of uniform motion at t_f:

$$\begin{aligned} &\left[y_{i\_j}(t_f), v_{i\_j}(t_f), a_{i\_j}(t_f), \theta_{i\_j}(t_f), \omega_{i\_j}(t_f)\right] = \\ &\left[\mathrm{target}_k, \mathrm{v_{same}}, 0, 0, 0\right], \; i = 1, \ldots, \mathrm{NL}, j = 1, \ldots, \mathrm{N}_i. \end{aligned} \quad (4)$$

Herein, $y = \mathrm{target}_k$ represents the central line of lane k.

C. Collision Avoidance Constraints

Collisions 1) among the vehicles, and 2) between each vehicle and either road barrier are prohibited during the lane changes. Two circles are used to represent the contour of each vehicle in the 2D plane (Fig. 2). The circle center closer to the rear wheels is denoted as $\mathrm{Cr}_i = \left(\mathrm{Cr}_{ix}, \mathrm{Cr}_{iy}\right)$, and the other is denoted as $\mathrm{Cf}_i = \left(\mathrm{Cf}_{ix}, \mathrm{Cf}_{iy}\right)$. Their locations can be determined through simple mathematics:

$$\begin{aligned} \mathrm{Cr}_{i\_jx}(t) &= x_{i\_j}(t) + \frac{\mathrm{L_w} + \mathrm{L_f} - 3\mathrm{L_r}}{4} \cdot \cos\theta_{i\_j}(t) \\ \mathrm{Cr}_{i\_jy}(t) &= y_{i\_j}(t) + \frac{\mathrm{L_w} + \mathrm{L_f} - 3\mathrm{L_r}}{4} \cdot \sin\theta_{i\_j}(t) \\ \mathrm{Cf}_{i\_jx}(t) &= x_{i\_j}(t) + \frac{3\mathrm{L_w} + 3\mathrm{L_f} - \mathrm{L_r}}{4} \cdot \cos\theta_{i\_j}(t) \\ \mathrm{Cf}_{i\_jy}(t) &= y_{i\_j}(t) + \frac{3\mathrm{L_w} + 3\mathrm{L_f} - \mathrm{L_r}}{4} \cdot \sin\theta_{i\_j}(t) \end{aligned}, \quad (5)$$

and the radius R of each circle is $\sqrt{\left(\frac{\mathrm{L_r} + \mathrm{L_w} + \mathrm{L_f}}{4}\right)^2 + \left(\frac{\mathrm{L_b}}{2}\right)^2}$.

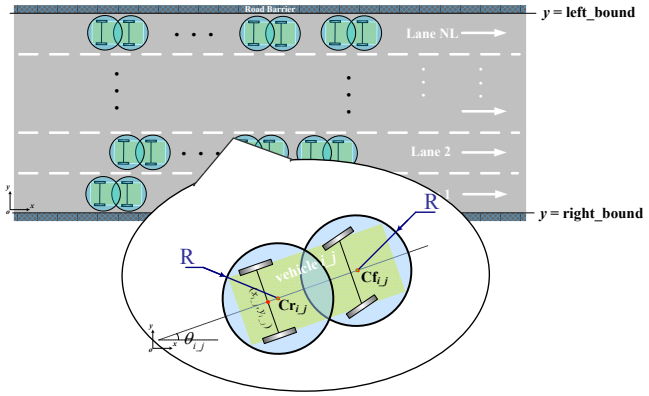

Fig. 2. Schematics on vehicle collision-avoidance constraints.

The collision between vehicle i and any other vehicle j can be avoided if

$$\begin{cases} \sqrt{\left(\mathrm{Cr}_{ix}(t) - \mathrm{Cr}_{jx}(t)\right)^2 + \left(\mathrm{Cr}_{iy}(t) - \mathrm{Cr}_{jy}(t)\right)^2} \ge 2\mathrm{R} \\ \sqrt{\left(\mathrm{Cr}_{ix}(t) - \mathrm{Cf}_{jx}(t)\right)^2 + \left(\mathrm{Cr}_{iy}(t) - \mathrm{Cf}_{jy}(t)\right)^2} \ge 2\mathrm{R} \\ \sqrt{\left(\mathrm{Cf}_{ix}(t) - \mathrm{Cr}_{jx}(t)\right)^2 + \left(\mathrm{Cf}_{iy}(t) - \mathrm{Cr}_{jy}(t)\right)^2} \ge 2\mathrm{R} \\ \sqrt{\left(\mathrm{Cf}_{ix}(t) - \mathrm{Cf}_{jx}(t)\right)^2 + \left(\mathrm{Cf}_{iy}(t) - \mathrm{Cf}_{jy}(t)\right)^2} \ge 2\mathrm{R} \end{cases}, \quad (6)$$

$\forall i, j = 1, \ldots, \mathrm{N_V}, \; i < j, \; t \in [t_0, t_f]$.

Besides that, collisions between vehicle i_j and the road barriers are avoided via

$$\begin{cases} \mathrm{Cr}_{i\_jy}(t) + \mathrm{R} \le \text{left\_bound} \\ \mathrm{Cr}_{i\_jy}(t) - \mathrm{R} \ge \text{right\_bound} \\ \mathrm{Cf}_{i\_jy}(t) + \mathrm{R} \le \text{left\_bound} \\ \mathrm{Cf}_{i\_jy}(t) - \mathrm{R} \ge \text{right\_bound} \end{cases}, \quad (7)$$
$$i = 1,\ldots,\mathrm{NL}, j = 1,\ldots,\mathrm{N}_i,\ t \in [t_0, t_\mathrm{f}],$$

where $y = \text{left\_bound}$ and $y = \text{right\_bound}$ represent the road barriers on both sides (Fig. 2).

D. Lane Change Objective

Let us define a cost functional J that trades off between minimizing time and steering energy:

$$J = (t_\mathrm{f} - t_0) + \lambda \cdot \int_{t=t_0}^{t_\mathrm{f}} \left(\sum_{i=1}^{\mathrm{NL}} \sum_{j=1}^{\mathrm{N}_i} \left(\phi_{i\_j}(t) \right)^2 \right) \cdot \mathrm{d}t, \quad (8)$$

wherein $\lambda > 0$ is the coefficient for trade-off. This formulation penalizes larger changes in the steering angles of the CAVs and longer time for completing the lane changes.

E. Overall Problem Statement

To summarize, a dynamic optimization problem is formulated as follows:

$$\begin{aligned} &\min\ (8) \\ &\text{s.t. Eqs. } (1) - (7). \end{aligned} \quad (9)$$

The major difficulties in (9) lie in the collision-avoidance constraints (6), the scale of which grows geometrically with $\mathrm{N_V}$.

III. Dynamic Optimization Problem Solution

A. Preliminaries

To solve (9) numerically, the orthogonal collocation direct transcription (OCDT) method is adopted, which describes the infinite-dimensional variables in (9) through a finite number of parameters, thereby converting (9) to a nonlinear programming (NLP) problem [7,8]. The interior-point method (IPM) has been widely known as being capable of solving large-scale NLPs. However, directly applying IPM does not lead to converged optima, as the difficulties in (9) are beyond the capability of IPM.

B. Motivation

Since the difficulties in (9) cannot be handled all at once, a natural thought is to divide the whole difficult constraints into multiple parts, to solve them separately, and then to summarize the partial results together. This thought inspires us to temporarily discard the collision-avoidance constraints and then gradually add them back towards the original formulation (9). Concretely, a number of sub-problems are defined in a sequence such that sub-problem (k+1) contains more collision-avoidance constraints compared to sub-problem k. Sub-problem 0, which contains none collision-avoidance constraint, is solved at step 1; thereafter, the obtained optimum is utilized as the initial guess for solving sub-problem 1 at step 2; this iterative process continues until all the constraints are added back.

If each sub-problem can be solved successfully, this stepwise process ensures to derive the optimal solution to the original problem within finite steps, as the total number of collision-avoidance constraints in the discretized NLP problem is finite.

C. Stepwise Computational Framework

A stepwise computation algorithm is proposed to facilitate the numerical solving process of (9). Suppose that $\mathrm{N_{fe}}$ equidistant finite elements are utilized when converting (9) into the corresponding NLP form via OCDT [8]. We define ($\mathrm{N_{fe}}$+1) sub-problems, namely P_0, P_1,…, $\mathrm{P}_{\mathrm{N_{fe}}}$. Herein, P_0 is defined by removing the collision-avoidance constraints in (9) throughout [t_0, t_f]. Compared with P_0, P_1 includes the collision-avoidance constraints during [t_0, t_0+($t_\mathrm{f} - t_0$)/$\mathrm{N_{fe}}$]. P_2 additionally includes the collision-avoidance constraints during the period $[t_0 + (t_\mathrm{f} - t_0)/\mathrm{N_{fe}}, t_0 + 2 \cdot (t_\mathrm{f} - t_0)/\mathrm{N_{fe}}]$. Following this, for $k \ge 3$, P_k is formed through adding the collision-avoidance constraints during $[t_0 + k \cdot (t_\mathrm{f} - t_0)/\mathrm{N_{fe}}, t_0 + (k+1) \cdot (t_\mathrm{f} - t_0)/\mathrm{N_{fe}}]$ to P_{k-1}.

At step 1, sub-problem P_0 is numerically solved using OCDT in conjunction with IPM. The derived optimal solution is recorded as the initial guess for solving the next sub-problem P_1 at step 2. The optimal solution to P_1, once obtained, is recorded as the initial guess for solving P_2 at step 3. This iterative process continues until the last sub-problem $\mathrm{P}_{\mathrm{N_{fe}}}$ (also known as the original problem (9)), is solved. The aforementioned computation procedures are summarized as the following pseudo codes.

Algorithm 1. Stepwise Computation Algorithm.

1. Solve P_0 without initial guess;
2. **If** the solving process of P_0 fails, **then**
3. | Report failure;
4. | Go to step 19;
5. **End if**
6. Record the obtained optimum of P_0 as the initial guess;
7. **For** k = 1 : ($\mathrm{N_{fe}}$-1) **do**
8. | Solve P_k with the current initial guess;
9. | **If** the solving process of P_k succeeds, **then**
10. | | Update the current initial guess as the obtained optimum of P_k;
11. | **End if**
12. **End for**
13. Solve P_k with the current initial guess;
14. **If** the solving process of $\mathrm{P}_{\mathrm{N_{fe}}}$ fails, **then**
15. | Report failure;
16. | Go to step 19;
17. **End if**
18. Output the optimum of $\mathrm{P}_{\mathrm{N_{fe}}}$;
19. Exit.

In this way, the difficulties in the original problem (9) are divided into multiple parts, and every progress made to address the partial difficulties will be solidified by the initial guess [4, 9–13].

IV. Simulation Results and Discussions

Simulations were conducted in the AMPL environment [14] and executed on an Intel Core i5-4460T CPU that runs at 1.90 × 2 GHz. A software package of IPM, namely IPOPT, is utilized [15]. Some parametric settings are listed in Table I and II, while others are listed in Appendix.

Table I. User-specific parametric notations and settings.

Parameter	Description	Setting
N_{fe}	Number of finite elements in OCDT	20
L_f	Vehicle front overhang length	0.960 m
L_w	Vehicle wheelbase	2.800 m
L_r	Vehicle rear overhang length	0.929 m
L_b	Vehicle width	1.942 m
N_V	Number of CAVs	12
NL	Number of lanes of the road	4
N_i	Number of CAVs in lane i at t_0, $i=1,...,N_V$	3
λ	Trade-off parameter in (8)	10
v_{max}	Upper bound of $v_i(t)$, $i=1,...,N_V$	15.0 m/s
Φ_{max}	Upper bound of $\lvert\phi_i(t)\rvert$, $i=1,...,N_V$	0.576 rad
a_{max}	Upper bound of $\lvert a_i(t)\rvert$, $i=1,...,N_V$	0.5 m/s^2
ω_{max}	Upper bound of $\lvert\omega_i(t)\rvert$, $i=1,...,N_V$	0.3 rad/s
v_{same}	Common velocity at t_f	10.0 m/s
left_bound	Left boundary of the road	13.125 m
right_bound	Right boundary of the road	-1.875 m
$target_1$	Location of lane 1	0 m
$target_2$	Location of lane 2	3.75 m
$target_3$	Location of lane 3	7.50 m
$target_4$	Location of lane 4	11.25 m

Table II. Simulation case setup and results.

Case#	Lane change targets	Feature	Optimized J
1	Vehicles {6,9,10} travel in lane 1; Vehicles {1,2,11} travel in lane 2; Vehicles {5,7,8,12} travel in lane 3; Vehicles {3,4} travel in lane 4.	Random	7.376
2	Vehicles {1,2,6,9,10,12} travel in lane 1; Vehicles {11} travel in lane 2; Vehicles {5,7,8} travel in lane 3; Vehicles {3,4} travel in lane 4.	Unbalanced	7.578
3	Vehicles {7,8,9} travel in lane 1; Vehicles {10,11,12} travel in lane 2; Vehicles {1,2,3} travel in lane 3; Vehicles {4,5,6} travel in lane 4.	Highly conflicting	7.608

The optimized lane-change trajectories of Cases 1–3 are depicted in Figs. 3–5. To assist understanding, a video is provided at https://youtu.be/Bx6dxHohKUE. In Figs. 3–5, each lane-change vehicle travels monotonically from the original lane to the target lane, which contributes to a relatively smooth trajectory. To further investigate this phenomenon, we additionally conducted a comparative experiment with a reduced λ ($\lambda=1$) on the basis of Case 1. The optimized trajectories are shown in Fig. 6. Compared with Fig. 3, Fig. 6 contains several lane-change vehicles which do not travel monotonically between their original and goal lanes. This difference is due to the decrease of λ, which indicates that the smoothness of the lane-change trajectories become less important in the optimization objective (8) whereas the completion time ($t_f - t_0$) becomes more important. As it turns out, nonetheless, the completion time increases from 7.025 to 7.065 when λ reduces from 10 to 1. This means the optimal solution depicted in Fig. 6 is a local optimum rather than a global one. Therefore, the global optimization capability of the proposed methodology still deserves improvement.

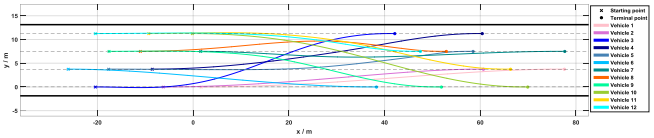

Fig. 3. Optimized lane-change trajectories of Case 1 ($t_f - t_0 = 7.025$ s).

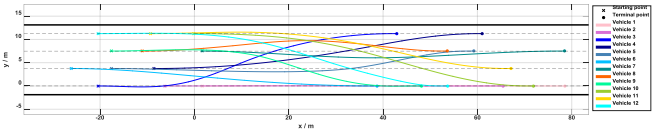

Fig. 4. Optimized lane-change trajectories of Case 2 ($t_f - t_0 = 7.098$ s).

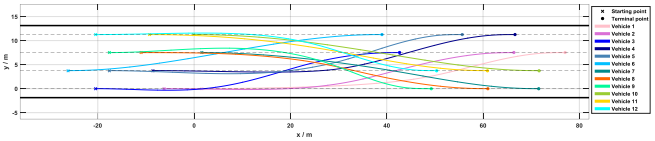

Fig. 5. Optimized lane-change trajectories of Case 3 ($t_f - t_0 = 6.999$ s).

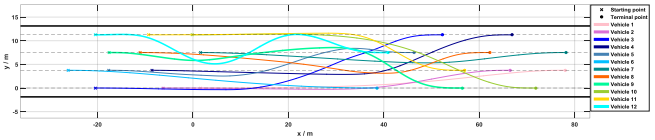

Fig. 6. Optimized lane-change trajectories of Case 1 with $\lambda=1$ ($t_f - t_0 = 7.065$ s).

To show the efficiency of the proposed stepwise computation algorithm, a few algorithms similar with Algorithm 1 are defined in Table III.

Table III. Definitions of Algorithms 2–6.

Algorithm	Description
Algorithm 2	Same as Algorithm 1, except that the sub-problem solution pathway is reduced as $P_0, P_1, P_3, P_5, ..., P_{19}, P_{20}$.
Algorithm 3	Same as Algorithm 1, except that the sub-problem solution pathway is reduced as $P_0, P_1, P_4, P_7, ..., P_{19}, P_{20}$.
Algorithm 4	Same as Algorithm 1, except that the sub-problem solution pathway is reduced as $P_0, P_1, P_5, P_9, P_{13}, P_{17}, P_{20}$.
Algorithm 5	Same as Algorithm 1, except that the sub-problem solution pathway is reduced as $P_0, P_1, P_9, P_{17}, P_{20}$.
Algorithm 6	Same as Algorithm 1, except that sub-problem P_k is formed through adding the collision-avoidance constraints during $[t_f - k\cdot(t_f - t_0)/N_{fe}, t_f - (k-1)\cdot(t_f - t_0)/N_{fe}]$ to P_{k-1} ($k=1,...,20$).

As it turns out, Algorithm 6 fails after consuming 5653.37 seconds while Algorithms 1–5 succeed in deriving the optimal solution to the original problem. The CPU time spent

on the execution of Algorithms 1–5 is shown in Fig. 7.

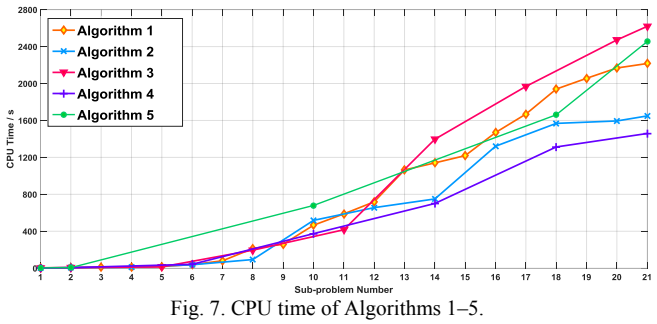

Fig. 7. CPU time of Algorithms 1–5.

Compared with Algorithm 1, Algorithm 6 gradually adds the collision-avoidance constraints in a reverse way (i.e. from the terminal moment towards the starting moment). Note that each x_{i_j} is not specified at t_f, thus the satisfaction to (4) is easier than (3). This means that $x_{i\_j}(t_f)$ would change drastically during the sequential sub-problem-solving process in adopting Algorithm 6. The failure of Algorithm 6 is intuitively understandable because "tail wags the dog" is far more difficult than "dog wags the tail". Viewing the performance of Algorithms 2–5, one can notice that skipping the sub-problems in part can shorten the gross CPU time, but skipping too many sub-problems may not be beneficial.

V. Conclusions

This paper has introduced a centralized and optimal MVMP method for cooperative lane changes of CAVs. A stepwise computational framework is proposed to facilitate the numerical solving process of the centralized MVMP problem.

The efficiency of the proposed algorithm is validated via simulations. As our future work, the global optimization capability will be seriously investigated. Furthermore, the time efficiency of Algorithm 1 could be improved via developing a self-adaptive sub-problem skipping strategy.

Appendix

The initial location of each CAV is listed in the following table.

Table A. CAV locations at t_0.

Parameter	Value	Parameter	Value
(x_{1_1}, y_{1_1})	(1.626, 0)	(x_{3_1}, y_{3_1})	(1.596, 7.50)
(x_{1_2}, y_{1_2})	(–6.234, 0)	(x_{3_2}, y_{3_2})	(–11.045, 7.50)
(x_{1_3}, y_{1_3})	(–20.425, 0)	(x_{3_3}, y_{3_3})	(–17.603, 7.50)
(x_{2_1}, y_{2_1})	(–8.557, 3.75)	(x_{4_1}, y_{4_1})	(–0.150, 11.25)
(x_{2_2}, y_{2_2})	(–17.634, 3.75)	(x_{4_2}, y_{4_2})	(–9.295, 11.25)
(x_{2_3}, y_{2_3})	(–26.139, 3.75)	(x_{4_3}, y_{4_3})	(–20.447, 11.25)

Preliminary source codes of this work are provided at https://www.researchgate.net/publication/324862623.

Acknowledgments

The authors are sincerely grateful to the four anonymous reviewers for their valuable comments. Bai Li thanks Bo'er Tao and Dr. Zhijiang Shao for the inspirations during this study.

References

[1] B. Li, Y. M. Zhang, Y. Feng, Y. Zhang, Y. Ge and Z. Shao, "Balancing computation speed and quality: A decentralized motion planning method for cooperative lane changes of connected and automated vehicles," *IEEE Transactions on Intelligent Vehicles*, accepted, 2018.

[2] C. Katrakazas, M. Quddus, W. H. Chen, and L. Deka, "Real-time motion planning methods for automated on-road driving: State-of-the-art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 416–442, 2015.

[3] N. Dadkhah, and B. Mettler, "Survey of motion planning literature in the presence of uncertainty: Considerations for UAV guidance," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1, pp. 233–46, 2012.

[4] B. Li, Y. M. Zhang, Z. Shao, and N. Jia, "Simultaneous versus joint computing: A case study of multi-vehicle parking motion planning," *Journal of Computational Science*, vol. 20, pp. 30–40, 2017.

[5] M. During, and K. Lemmer, "Cooperative maneuver planning for cooperative driving," *IEEE Intelligent Transportation Systems Magazine*, vol. 8, no. 3, pp. 8–22, 2016.

[6] C. Bax, P. Leroy, and M. P. Hagenzieker, "Road safety knowledge and policy: A historical institutional analysis of the Netherlands," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 25, pp. 127–136, 2014.

[7] B. Li, and Z. Shao, "A unified motion planning method for parking an autonomous vehicle in the presence of irregularly placed obstacles," *Knowledge-Based Systems*, vol. 86, pp. 11–20, 2015.

[8] L. T. Biegler. *Nonlinear programming: concepts, algorithms, and applications to chemical processes*, SIAM, 2010.

[9] B. Li, K. Wang, and Z. Shao, "Time-optimal maneuver planning in automatic parallel parking using a simultaneous dynamic optimization approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 11, pp. 3263–3274, 2016.

[10] B. Li, and Z. Shao, "An incremental strategy for tractor-trailer vehicle global trajectory optimization in the presence of obstacles," In *Proc. 2015 IEEE International Conference on Robotics and Biomimetics*, pp. 1447–1452, 2015.

[11] B. Li, Y. M. Zhang, and Z. Shao, "Spatio-temporal decomposition: A knowledge-based initialization strategy for parallel parking motion optimization," *Knowledge-Based Systems*, vol. 107, pp. 179–196, 2016.

[12] B. Li, Z. Shao, Y. M. Zhang, and P. Li, "Nonlinear programming for multi-vehicle motion planning with Homotopy initialization strategies," In *Proc. 13th IEEE Conference on Automation Science and Engineering*, pp. 118–123, 2017.

[13] B. Li, N. Jia, W. Meng, and Y. M. Zhang, "Incrementally constrained dynamic optimization: A computational framework for lane change motion planning of connected and automated vehicles," *Journal of Intelligent Transportation Systems*, submitted for review.

[14] R. Fourer, D. M. Gay, and B. W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. San Francisco, CA, USA: Scientific, 2003.

[15] A. Wächter, and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.