IEEE Transactions on Industrial Electronics, Volume: 67, Issue 12, December 2020, pp. 10809 - 10821
DOI: 10.1109/TIE.2019.2962482

Multi-Objective Optimal Parking Maneuver Planning of Autonomous Wheeled Vehicles

Runqi Chai, *Member, IEEE,* Antonios Tsourdos, *Member, IEEE,* Al Savvaris, *Member, IEEE,* Senchun Chai, *Senior Member, IEEE,* Yuanqing Xia, *Senior Member, IEEE,* and C. L. Philip Chen, *Fellow, IEEE,*

***Abstract*—This paper proposes a computational trajectory optimization framework for solving the problem of multi-objective automatic parking motion planning. Constrained automatic parking maneuver problem is usually difficult to solve because of some practical limitations and requirements. This problem becomes more challenging when multiple objectives are required to be optimized simultaneously. The designed approach employs a swarm intelligent algorithm to produce the trade-off front along the objective space. In order to enhance the local search ability of the algorithm, a gradient operation is utilized to update the solution. In addition, since the evolutionary process tends to be sensitive with respect to the flight control parameters, a novel adaptive parameter controller is designed and incorporated in the algorithm framework such that the proposed method can dynamically balance the exploitation and exploration. The performance of using the designed multi-objective strategy is validated and analyzed by performing a number of simulation and experimental studies. The results indicate that the present approach can provide reliable solutions and it can outperform other existing approaches investigated in this paper.**

***Index Terms*—Trajectory optimization, automatic parking, trade-off front, adaptive parameter controller.**

I. Introduction

TRAJECTORY or motion planning of autonomous vehicles has received significant attentions over the last two decades due to its importance in the autonomous control module [1], [2]. A motion planner can automatically produce a feasible path for the vehicle from a given starting point to a desired terminal position. Early works on this topic mainly focus on the development of geometric path planners or smooth decomposition-based path planners such as the spline-based planner and the cubic polynomial-inspired planner [3], [4]. In the recent five years, there has been a growing interest in generating the vehicle path by using optimization-based approaches [5], [6]. One main advantage of using these optimization-based planning methods is that different types of vehicular or environmental limitations which are usually problematic for geometric and smooth decomposition-based planners, can be modelled into constraints and entailed in the optimization formulation. Due to this reason, we give more attentions to the application of this kind of technique.

The mission scenario studied in this paper focuses on the optimal parallel parking of an autonomous wheeled vehicle. In most of existing works, the parking path design usually targets one single performance index. For example, in [7] the authors minimized the parking position error by using a feedforward direct method. In [8], the primary goal is to shape a smooth parallel parking trajectory such that the curvature variation indicator can be minimized. Furthermore, the authors in [3] selected the path length as the main objective and calculated the shortest path of an automatic parking problem. However, in practical situations, all the aforementioned individual objectives should be optimized simultaneously. This inspires the development of multi-objective optimal parking maneuver methods which will be mainly focused in the present study.

It is important to note that in [9], the authors proposed and applied a genetic algorithm (GA)-based multi-objective parking algorithm to produce trajectories for a car-like autonomous vehicle. In their work, the final position and oriental angle errors were selected as two mission objectives and they were integrated through aggregation. In addition, a multi-objective motion planner using η^3-spline techniques was developed in [10] to minimize the parking path length as well as the curvature variation. Similar with [9], these two objectives were integrated through aggregation and optimized via a GA-based method. In these two studies, the mission scenario was extended from single-objective case to multi-objective case, and effective parking trajectories were obtained. However, there are still some remaining issues of these works. For example, in [9] there is no constraint for the instantaneous curvature as well as the parking slot boundary, which must frequently be considered during the real parking environment. Besides, since the geometric motion planner developed in [10] can be treated as a point-to-point framework, it may have difficulty in dealing with parking cases when multiple backward and forward motions are needed. As a result, a new design of multi-objective parking algorithm is highly demanded to tackle these issues and offer an effective alternative.

To address the multi-objective parking optimization problems, the evolutionary multi-objective optimization (EMO) algorithm is a powerful and promising tool because of its ability in searching Pareto-optimal solutions in a single trial. It is important to note that usually for a multi-objective engineering problem, there is no single solution that can optimize all the mission objectives simultaneously, as the objectives might conflict with each other. Hence, mission planners usually aim to find a set of compromised solutions and then make a decision based on their preferences. Among EMOs, the multi-objective particle swarm optimization (MOPSO) algorithm is usually recognised as an effective and intelligent swarm-based optimization technique. The earliest MOPSO was proposed in [11] and further investigated in [12]. A review of achievements on developing this algorithm can be found in [13]. As for the practical implementations of MOPSO, contributions can also be reviewed in the literature. For instance, a humanoid robot online navigation problem was successfully solved in [14], wherein a preference-guided MOPSO algorithm was designed and applied to produce the compromised solution. Similarly, a PSO-based multi-objective optimization strategy was proposed in [15] in order to maximize the output force and efficiency of a planar motor while minimizing the cost at the same time. Although all the aforementioned MOPSO-based methods have

R. Chai, A. Tsourdos and A. Savvaris are with the School of Aerospace, Transport and Manufacturing, Cranfield University, UK, e-mail: (r.chai@cranfield.ac.uk), (a.tsourdos@cranfield.ac.uk), and (a.savvaris@cranfield.ac.uk).

S. Chai and Y. Xia are with the school of Automation, Beijing Institute of Technology, Beijing, China, e-mail: (chaisc97@163.com), (xia_yuanqing@bit.edu.cn).

C. L. P. Chen is with the Faculty of Science and Technology, University of Macau, Macau 999078, China, and also with the Department of Navigation, Dalian Maritime University, Dalian 116026, China, and with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China, e-mail: (philip.chen@ieee.org).

© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

the potential to be applied in the multi-objective parking maneuver planning problem, there are still some issues for implementations. One important issue is that the use of MOPSO will introduce some additional parameters to control the evolution direction. Poor selection of these parameters can significantly degrade the diversity and optimality of the final solution set. Besides, the convergence speed of the algorithm is also largely affected by these parameters. Hence a proper treatment with respect to these algorithm control parameters is required.

The main contributions of this study are summarised below:

1) A new multi-objective parking maneuver optimization model is established with the consideration of instantaneous curvature constraint and parking slot boundary constraints.
2) A hybrid metric-based adaptive gradient MOPSO approach (denoted as HM-AGMOPSO) is proposed. The main novel part of the proposed HM-AGMOPSO lies in its control parameter update component, where an adaptive flight controller is developed to dynamically balance the exploitation and exploration of the algorithm.
3) A local gradient operation is embedded in the overall algorithm framework so as to make further progress during the iteration.
4) A number of simulation studies are presented to illustrate the effectiveness and superiority of the proposed method, while experimental tests are provided to validate the reliability of the pre-planned parking trajectory.

The remainder of this paper is organized as follows. Sec II illustrates the multi-objective optimization model of the autonomous vehicle parking problem. Then the method used to solve the problem is detailed in Sec III. Experimental results and comparative studies are demonstrated in Sec IV and Sec V, while the conclusion is drawn in Sec VI.

II. Multi-Objective Model of the Autonomous Vehicle Parking Problem

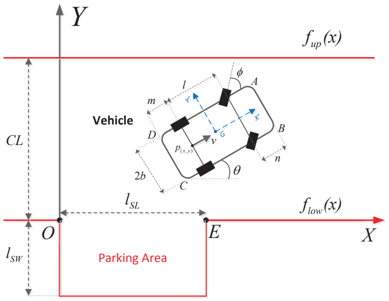


Fig. 1: Illustration of the parking mission

TABLE I: Notations for variables

p_x, p_y:	Central point of the rear wheel
v, α:	Velocity and acceleration of the vehicle
θ:	Oriental angle
ϕ:	Steering angle of the steering wheel
η:	The jerk (derivative of the acceleration)
ω:	Angular velocity of the front wheel
t:	Time point
l:	Length between the front and rear wheels
m:	The rear overhang
n:	The front overhang
l_{SL}:	Length of the parking slot
l_{SW}:	Width of the parking slot
CL:	Width of the road
$2b$:	Width of the vehicle

In this section, the mathematical formulation of the investigated multi-objective problem (MOP) is discussed. At the beginning, the wheeled vehicle equations of motion will be introduced in Sec II.A. Following that, different types of constraints including the state/control path constraints and collision-free constraints are formulated in Sec II.B. Subsequently, in order to take more of the real-world requirements into account, multiple performance indices are established and detailed in Sec II.C.

A. Vehicle Equations of Motion

To describe the movement of the wheeled vehicle and the automatic parking scenario, a detailed mission illustration, together with the vehicle equations of motion, is summarised in Fig.1 and Eq.(1).

$$\begin{cases} \dot{p_x}(t) = v(t)\cos(\theta(t)) \\ \dot{p_y}(t) = v(t)\sin(\theta(t)) \\ \dot{v}(t) = \alpha(t) \\ \dot{\alpha}(t) = \eta(t) \\ \dot{\theta}(t) = v(t)\tan(\phi(t))/l \\ \dot{\phi}(t) = \omega(t) \end{cases} \tag{1}$$

The notations of all the mission/vehicle-dependent variables appeared in Fig.1 and Eq.(1) are tabulated in Table I. The vehicle is considered as a front-steering rigid body and the sideslip is ignored. For the sake of brevity, the state variables are abbreviated as $x = [p_x, p_y, v, \alpha, \theta, \phi]^T$, whereas the controls are written as $u = [\eta, \omega]^T$, respectively. The reason for making η as the control input in the trajectory design phase is to smooth the planned acceleration and speed profiles [5], [16]. The actual control inputs applied to the vehicle in the experiments are v and ω [17], [18]. This will be further shown in the experiment section of this paper.

B. Different Types of Constraints

During the parking movement process, different types of constraints are required to be considered. For example:

1). State/control path constraints: The magnitude of the state and control variables should be limited and this can be expressed by

$$\begin{array}{rr} p_x^{min} \leq p_x(t) \leq p_x^{max} & p_y^{min} \leq p_y(t) \leq p_y^{max} \\ v^{min} \leq v(t) \leq v^{max} & \alpha^{min} \leq \alpha(t) \leq \alpha^{max} \\ \theta^{min} \leq \theta(t) \leq \theta^{max} & \phi^{min} \leq \phi(t) \leq \phi^{max} \\ d_\alpha^{min} \leq \eta(t) \leq d_\alpha^{max} & d_k^{min} \leq \dot{k}(t) \leq d_k^{max} \end{array} \tag{2}$$

where $k = \tan(\theta)/l$ represents the instantaneous curvature. Its derivative value can be computed by $\dot{k} = \omega/(l\cos^2(\theta))$. It is worth noting that imposing a path constraint on the jerk variable η can effectively smooth the acceleration evolution. On the other hand, the path constraint of $\dot{k}$ can avoid non-smooth segments on the vehicle trajectory, thus decreasing discomfort to passengers.

2). Parking slot and terminal constraints: As specified in Fig.1, certain constraints should be assigned at the terminal time instant t_f to place the vehicle in the desired parking slot. To formulate them, the four corner points of the vehicle (ABCD) should be used:

$$\begin{cases} A_x = p_x + \cos(\theta)(l+n) - b\sin(\theta) \\ A_y = p_y + \sin(\theta)(l+n) + b\cos(\theta) \\ B_x = p_x + \cos(\theta)(l+n) + b\sin(\theta) \\ B_y = p_y + \sin(\theta)(l+n) - b\cos(\theta) \\ C_x = p_x - m\cos(\theta) + b\sin(\theta) \\ C_y = p_y - m\sin(\theta) - b\cos(\theta) \\ D_x = p_x - m\cos(\theta) - b\sin(\theta) \\ D_y = p_y - m\sin(\theta) + b\cos(\theta) \end{cases} \tag{3}$$

Then the terminal constraints are given by:

$$\begin{array}{lll} A_y(t_f) \leq 0, & B_y(t_f) \leq 0, & C_y(t_f) \leq 0 \\ D_y(t_f) \leq 0, & v(t_f) = 0, & \alpha(t_f) = 0 \end{array} \tag{4}$$

Note that the last two equality constraints can be treated as a full stop condition. Apart from Eq.(4), to successfully park the vehicle, the following parking area constraints are formulated:

$$\begin{array}{l} f_{low}(A_x) \leq A_y \leq f_{up}(A_x) \\ f_{low}(B_x) \leq B_y \leq f_{up}(B_x) \\ f_{low}(C_x) \leq C_y \leq f_{up}(C_x) \\ f_{low}(D_x) \leq D_y \leq f_{up}(D_x) \end{array} \tag{5}$$

in which $f_{low}(\cdot)$ stands for the lower boundary of the parking slot and is designed as $f_{low}(x) = -(H(x)+H(x-l_{SL}))l_{SW}$. The upper boundary of the road is modeled as $f_{up}(x) = C_L$. Here $H(\cdot)$ stands for the Heaviside step function.

3). Collision-free limitations: From Fig.1, it is clear that the vehicle has the probability to collide with the edge points of the parking slot (e.g., points O and E). Hence, collision-free constraints are required to be considered. Different from the discontinuous collision-free constraint formulation used in [5], we apply a continuous version in this paper. More precisely, this is achieved by restricting the corner points O and E are always located outside the vehicle rectangular area $ABCD$, which can be described as:

$$\begin{array}{l} S_{AOB} + S_{BOC} + S_{COD} + S_{AOD} \geq S_{ABCD} \\ S_{AEB} + S_{BEC} + S_{CED} + S_{AED} \geq S_{ABCD} \end{array} \tag{6}$$

where S denotes the area operation.

Remark 1. In [5], the authors modeled the collision-free constraints by transforming the corner points O and E from the coordinate XOY to the vehicle's body frame $X'GY'$ (as indicated in Fig.1). Then the collision avoidance is achieved by restricting:

$$\begin{array}{ll} |O_x^{'}| \geq (n+l+m)/2, & \text{if} \quad |O_y^{'}| \leq b \\ |E_x^{'}| \geq (n+l+m)/2, & \text{if} \quad |E_y^{'}| \leq b \end{array} \tag{7}$$

where $(O_x^{'}, O_y^{'})$ and $(E_x^{'}, E_y^{'})$ are coordinates of O and E in the $X'GY'$ frame. It is obvious that Eq.(7) is discontinuous, and the optimization algorithm might struggle due to the intuitive application of this equation.

C. Mission Objectives

Early investigations on automatic parking trajectory planning problems normally target one single objective. However, in practice, multiple objectives might need to be frequently considered during the maneuver phrase. In this study, we consider three objectives so as to capture more realities of the problem. The first objective to be optimized is the path length so that the vehicle can park along a shortest trajectory. Besides, since the path smoothness is directly related to the comfort of the passengers and riders, a path smoothness indicator is defined and minimized. Furthermore, to measure the quality of the parking, a parking achievement indicator is used. The aim for optimizing this indicator is that it is desired to place the geometric center of the vehicle overlapping the geometric center of the parking slot with a zero oriental angle. Consequently, the three objectives applied for experiments are:

$$\begin{array}{l} \min J_1 = \int_0^{t_f} v dt \\ \min J_2 = \int_0^{t_f} \dot{k} dt \\ \min J_3 = E_f \end{array} \tag{8}$$

where $t_f \in [t_f^{min}, t_f^{max}]$ is a variable. E_f is given by:

$$E_f = (G_x(t_f) - G_x^*)^2 + (G_y(t_f) - G_y^*)^2 + (\theta(t_f) - \theta^*)^2 \tag{9}$$

In Eq.(9), (G_x, G_y) are the geometric center of the vehicle and it can be computed by $G_x = (p_x + (((n+l+m)\cos\theta)/2))$; $G_y = (p_y + (((n+l+m)\sin\theta)/2))$. From Eq.(9), the three components have different scales. It is noteworthy that poor scaling can have significant impacts on the convergence performance of the optimization process. Hence, we normalize the coordinates and angle in (9) via $\bar{G}_x = G_x/l_{SL}$, $\bar{G}_y = G_y/l_{SW}$ and $\bar{\theta} = \theta/\pi$, respectively. Based on all the definitions stated above, the overall multi-objective parking movement planning model can be written as:

$$\begin{array}{ll} \text{minimize} & J = [J_1, J_2, J_3] \\ \text{subject to} & \dot{p_x}(t) = v(t)\cos(\theta(t)) \\ & \dot{p_y}(t) = v(t)\sin(\theta(t)) \\ & \dot{v}(t) = a(t) \\ & \dot{\alpha}(t) = \eta(t) \\ & \dot{\theta}(t) = v(t)\tan(\phi(t))/l \\ & \dot{\phi}(t) = \omega(t) \\ & A_y(t_f) \leq 0, B_y(t_f) \leq 0, C_y(t_f) \leq 0 \\ & D_y(t_f) \leq 0, v(t_f) = 0, a(t_f) = 0 \\ & f_{low}(A_x) \leq A_y \leq CL, f_{low}(B_x) \leq B_y \leq CL \\ & f_{low}(C_x) \leq C_y \leq CL, f_{low}(D_x) \leq D_y \leq CL \\ & S_{AOB} + S_{BOC} + S_{COD} + S_{AOD} \geq S_{ABCD} \\ & S_{AEB} + S_{BEC} + S_{CED} + S_{AED} \geq S_{ABCD} \\ & p_x^{min} \leq p_x(t) \leq p_x^{max}, p_y^{min} \leq p_y(t) \leq p_y^{max} \\ & v^{min} \leq v(t) \leq v^{max}, \alpha^{min} \leq \alpha(t) \leq \alpha^{max} \\ & \theta^{min} \leq \theta(t) \leq \theta^{max}, \phi^{min} \leq \phi(t) \leq \phi^{max} \\ & d_\alpha^{min} \leq \eta(t) \leq d_\alpha^{max}, d_k^{min} \leq \dot{k}(t) \leq d_k^{max} \end{array} \tag{10}$$

where $J = [J_1, J_2, J_3] \in \Re^3$ is the vector of objectives, whereas $u = [\eta(t), \omega(t)]^T$ is the decision variable of problem (10). To optimize problem (10), we apply the concept of Pareto-optimal, which is outlined by the following definitions.

Definition 1. A vector of decision variable $u_1 = [\eta_1, \omega_1]$ is dominated by another one u_2 if and only if $\forall M \in \{1, 2, 3\}$, $J_M(u_2) \leq J_M(u_1)$, and $\exists M \in \{1, 2, 3\}$, $J_M(u_2) < J_M(u_1)$. The domination relationship is denoted by $u_1 \prec u_2$.

Definition 2. A vector of decision variable u^* is Pareto-optimal if it is not dominated by other candidates among the feasible region $\mathscr{F}$.

Definition 3. The Pareto-optimal set $\mathscr{P}$ as well as the Pareto-optimal front P can be written as:

$$\begin{cases} \mathscr{P} = \{u \in \mathscr{F} | u \text{ is Pareto-optimal}\} \\ P = \{J(u) \in \Re^3 | u \in \mathscr{P}\} \end{cases}$$

From Definitions 1 and 2, there may exist a set of Pareto-optimal solutions for an MOP. Hence, the goal for solving an MOP becomes determining $\mathscr{P}$ from $\mathscr{F}$. In other words, when addressing the MOP, we aim to find not a single but rather a set of Pareto-optimal solutions.

Remark 2. It is noteworthy that some objective functions given by Eq.(8) are considered as constraints in other works. For example, the third objective function was considered as hard equality constraints in [5] and [16]. However, in this paper we are interested in designing a swarm-intelligence-based method to optimize the parking motion. The consideration of hard equality constraints will put more pressures on the evolution process for finding feasible solutions. Alternatively, considering J_3 as an objective can reduce the number of equality constraints, thereby easing the burden of finding feasible solutions.

III. Methodology

In order to address the multi-objective automatic parking mission given by Eq.(10), this paper applies an HM-AGMOPSO algorithm. This method can be treated as a global optimization technique as it uses the principle of survival of fitness in order to locate the optimal solution. Compared with gradient-based algorithms, the proposed method tends to have stronger ability to explore the entire searching space rather than be attracted to a local optimal point. Besides, the unique features of the HM-AGMOPSO method in comparison with traditional MOPSO and NSGA-II are that a local gradient operation as well as a hybrid metrics-based adaptive controller of flight parameters are designed and embedded in the algorithm. The aim for carrying on a local gradient search is to enhance the exploitation around the current solution. Besides, using the hybrid metrics-based adaptive controller can adjust the flight control variables so as to balance the local exploitation and global exploration. Priori to introducing in detail these two mechanisms, it is worth recalling some basis of the MOPSO for completeness reasons.

A. Basic MOPSO

In MOPSO, each particle among the swarm can be regarded as a candidate solution defined on the solution space and it has a position vector μ as well as a velocity vector ν. These two vectors can be written as:

$$\begin{array}{c} \mu_j(s) = [\mu_{j,1}(s), \mu_{j,2}(s) ..., \mu_{j,D}(s)] \\ \nu_j(s) = [\nu_{j,1}(s), \nu_{j,2}(s) ..., \nu_{j,D}(s)] \end{array} \tag{11}$$

where $s = 1, 2, ..., S^{max}$ denotes the index of the generation, while $j = 1, 2, ..., N_j$ is the index of the particle boxed by the size of the swarm N_j. Since the parking system inputs are time-dependent, we firstly discretize the time domain $[0, t_f]$ using a set of temporal nodes $\{t_k\}_{k=0}^{N_k-1}$, where $t_0 = 0$ and $t_{k+1} = t_k + t_f/N_k$. Then each particle is encoded as a vector representing a potential control sequence at $\{t_k\}$ and $D = 2 \cdot N_k + 1$ stands the dimensional index of each particle. More precisely, the following equation is used to present the actual content of the particle and the swarm:

$$\begin{array}{ll} \mu_j & = [\eta_j(t_0), ..., \eta_j(t_{N_k-1}), \omega_j(t_0), ..., \omega_j(t_{N_k-1}), t_f] \\ \text{Swarm} & = [\mu_1, ..., \mu_{N_j}] \end{array} \tag{12}$$

All particles in the first generation are obtained by randomly initializing all decision variables within the specified lower and upper control bounds given by Eq.(2). During the solution-finding process, the best position of the jth particle and the best position among the current swarm in the sth generation are recorded as $q_j(s)$ and $g(s)$, respectively. According to the definitions of $q_j(s)$ and $g(s)$, the velocity vector of the jth particle can then be updated by:

$$\begin{array}{rl} \nu_j(s+1) = & \varpi \nu_j(s) + r_1 c_1 (q_j(s) - \mu_j(s)) \\ & r_2 c_2 (g(s) - \mu_j(s)) \end{array} \tag{13}$$

In Eq.(13), several control parameters are introduced to generate the new velocity vector. ϖ represents the inertia weight, whereas r_1 and r_2 are two random constants uniformly distributed on $[0, 1]$. c_1 and c_2 can be treated as two acceleration parameters associated with the cognitive component and the social component, respectively. Using the updated velocity vector, the new position vector for the jth particle can be computed by:

$$\mu_j(s+1) = \mu_j(s) + \nu_j(s+1) \tag{14}$$

From Eq.(13) and Eq.(14), it is obvious that the particle uses the positional information of its own movement and the selected global leader to update the position and velocity vectors. Besides, the previous best position $q_j(s)$ should be re-recorded via:

$$q_j(s) = \begin{cases} q_j(s-1) & \text{if} \quad \mu_j(s) \prec q_j(s-1) \\ \mu_j(s) & \text{if} \quad \mu_j(s) \succ q_j(s-1) \\ \text{rand}\{\mu_j(s), q_j(s-1)\} & \text{if} \quad \mu_j(s) \nprec \nsucc q_j(s-1) \end{cases} \tag{15}$$

Here, the concept of Pareto optimality is adopted. $\mu_j(s) \prec q_j(s-1)$ means $\mu_j(s)$ is dominated by $q_j(s-1)$ and it should be replaced. On the other hand, the notation $\mu_j(s) \nprec \nsucc q_j(s-1)$ means the two are mutually non-dominated. In this case, one of them will be selected randomly (e.g., rand$\{\mu_j(s), q_j(s-1)\}$).

Defining $A(s) = [a_1(s), a_2(s), ..., a_{N_A}(s)]$ as the external archive (external non-dominated set), to update the archive $A(s)$, we need the previous archive $A(s-1)$ and $q_j(s)$ information. That is, if $q_j(s)$ is non-dominated by any individuals in $A(s-1)$, then $A(s) = A(s-1) \cup q_j(s)$. If $q_j(s)$ is dominated by an individual in $A(s-1)$, $q_j(s)$ is discarded. If there are some individuals in $A(s-1)$ that are dominated by $q_j(s)$, the $A(s-1)$ firstly removes elements that are dominated by $q_j(s)$ and then augments by $q_j(s)$.

B. Multi-Objective Gradient Operation

To strengthen the local exploitation capability of the MOPSO approach, a local gradient search is applied. The Jacobian vector of J with respect to μ can be written as $\nabla_{\mu_j} J_i(\mu_j(s))$=$[\frac{\partial J_i(\mu_j(s))}{\partial \mu_{j,1}(s)}, ..., \frac{\partial J_i(\mu_j(s))}{\partial \mu_{j,D}(s)}]$, $i = 1, 2, ...M$. Here, M is the number of objectives. Based on the Jacobian vector, a local descent direction can be determined:

$$e_j = -(\textstyle\sum_{i=1}^{M} w_i \frac{\nabla_{\mu_j} J_i(\mu_j(s))}{\|\nabla_{\mu_j} J_i(\mu_j(s))\|}), \textstyle\sum_{i=1}^{M} w_i = 1 \tag{16}$$

From Eq.(16), e_j can be understood as a linear combination of the steepest descent direction for all objectives. Subsequently, the particle can be updated by applying:

$$\bar{\mu}_j(s) = \mu_j(s) + \Delta\delta e_j \tag{17}$$

in which $\Delta\delta$ is the step length along the direction e_j. Notice that the gradient operation is applied to the current population and the improved particles are integrated back to the swarm. Moreover, the gradient operation is applied once every generation.

C. Handling of Mission Constraints

For most engineering optimization problems, different types of mission constraints are commonly required to be considered simultaneously when optimizing the objectives. For the parking movement planning problem (10), we apply a direct transcription method to deal with the path and collision-free constraints. This method, named multi-objective constraint handling (MOCH), is based on the principle of multi-objective optimization and it reformulates the constrained multi-objective problem to an unconstrained equivalent form. To do this, the total amount of constraint violation for each particle is computed. Subsequently, this value is assigned as the additional mission objective, which means the dimension of the objective is increased by one. For example, the additional objective J_{M+1} can be defined by:

$$J_{M+1} = \min(V_G + V_I) \tag{18}$$

V_G and V_I are, respectively, the total violation degrees for the inequality constraints $g_i(\cdot) > z_i, i = 1, ..., N_G$ and the equality constraints $I_i(\cdot) = 0, i = 1, ..., N_I$. To calculate their values, we apply $V_G = \sum_{i=1}^{N_G} \langle \bar{g}_i(\cdot) \rangle$ and $V_I = \sum_{i=1}^{N_I} |I_i(\cdot)|$. Here, $\bar{g}_i = (g_i(\cdot)/z_i) - 1$. The notation $\langle \bar{g}_i \rangle$ returns $|\bar{g}_i|$ if $\bar{g}_i < 0$. Otherwise, it returns zero. Compared with other constraint handling methods, the MOCH does not introduce additional parameters and can be easily implemented.

D. Controller Design of Flight Parameters

In traditional MOPSO, the flight parameters (ϖ, c_1 and c_2) are specified as fixed constants [11]. According to experimental results presented in the literature [19], [20], it can be summarised that a smaller ϖ, a smaller c_1 and a larger c_2 tend to result in better local exploitation. By contrast, the larger ϖ and c_1, together with a smaller c_2, can improve the global exploration ability of the algorithm. Based on these conclusions, in this subsection, we are interested in designing an adaptive controller to automatically tune the flight parameters. Inspired by the work presented in [19], the tuning is achieved by applying a hybrid strategy that takes into consideration the hypervolume (HV) and spacing (SP) information.

Until now, many multi-objective performance indicators have been designed to measure the quality of the obtained Pareto set [21], [22]. Detailed analysis and definitions of these metrics can be found in [23]. One type of indicator that has been widely applied is the distance-based metrics. These metrics calculate the distance between the true Pareto-optimal set and the approximated set obtained via a given method. However, for the considered parking problem, the true Pareto-set is unknown. Therefore, we apply the volume- and diversity-based measures (such as the HV and SP indicators [23]). The HV value can be computed by:

$$H(s+1) = Leb(\bigcup_{a \in A} [J_1(a), R_1] \times \cdots \times [J_M(a), R_M]) \qquad (19)$$

Here $Leb(\cdot)$ denotes the Lebesgue measure. $R = [R_1, ..., R_M]$ is the reference point. From Eq.(19), H can be understood as the union of all the rectangular areas and it reflects both the distribution and convergence of the archive. Besides, to further quantify the degree of the distribution, an SP metric is used, which can be written as:

$$SP(s+1) = (\frac{1}{N_A - 1} \sum_{j=1}^{N_A} (\bar{l}(s+1) - l_j(s+1))^2)^{0.5} \qquad (20)$$

in which $\bar{l}(\cdot)$ stands for the average minimum Manhattan distance (MD) of particles, whereas $l_j(\cdot)$ is the minimum MD of the jth particle. Notice that the SP equation is identical to the one formally defined in [24], and the distance measure used is different from the minimum Euclidean distance. From Eq.(20), if the solutions are near uniformly spread, then the resulting distance measure tends to be small. As a result, it is desired to find a set of Pareto-optimal solution with a smaller SP value.

The HV and SP values will be used as the primary metrics to design the flight parameter controller. Specifically, if the inputs of the controller are $H(s+1)$ and $SP(s+1)$, then the outputs can be written as:

$$\varpi(s+1) = \begin{cases} \varpi(s), & \text{if H(s+1)>H(s), SP(s+1)<SP(s);} \\ \varpi(s)\Delta_1(s), & \text{if H(s+1)>H(s), SP(s+1)>SP(s);} \\ \varpi(s)(\Delta_2(s)+1), & \text{if H(s+1)<H(s), SP(s+1)>SP(s);} \\ \varpi(s)(\frac{1}{2}\Delta_3(s)+1), & \text{if H(s+1)<H(s), SP(s+1)<SP(s).} \end{cases} \qquad (21)$$

$$c_1(s+1) = \begin{cases} c_1(s), & \text{if H(s+1)>H(s), SP(s+1)<SP(s);} \\ c_1(s)\Delta_1(s), & \text{if H(s+1)>H(s), SP(s+1)>SP(s);} \\ c_1(s)(\Delta_2(s)+1), & \text{if H(s+1)<H(s), SP(s+1)>SP(s);} \\ c_1(s)(\frac{1}{2}\Delta_3(s)+1), & \text{if H(s+1)<H(s), SP(s+1)<SP(s).} \end{cases} \qquad (22)$$

$$c_2(s+1) = \begin{cases} c_2(s), & \text{if H(s+1)>H(s), SP(s+1)<SP(s);} \\ c_2(s)(\Delta_1(s)+1), & \text{if H(s+1)>H(s), SP(s+1)>SP(s);} \\ c_2(s)(\Delta_2(s)), & \text{if H(s+1)<H(s), SP(s+1)>SP(s);} \\ c_2(s)(\frac{1}{2}\Delta_3(s)+1), & \text{if H(s+1)<H(s), SP(s+1)<SP(s).} \end{cases} \qquad (23)$$

where $\Delta_1(s) = \max\{\frac{H(s)}{H(s+1)}, \frac{SP(s)}{SP(s+1)}\}$, $\Delta_2(s) = \max\{\frac{H(s+1)}{H(s)}, \frac{SP(s)}{SP(s+1)}\}$ and $\Delta_3(s) = \min\{\frac{H(s+1)}{H(s)}, \frac{SP(s+1)}{SP(s)}\}$. $\varpi(s+1)$, $c_1(s+1)$ and $c_2(s+1)$ are the updated inertia weight and acceleration parameters, respectively.

Remark 3. It is worth noting that in the proposed adaptive strategy, the worst case scenario is $H(s+1) < H(s)$ and $SP(s+1) > SP(s)$. This reveals that the obtained solutions is losing diversity and extensiveness. Therefore, the global exploration should be emphasized. On the other hand, if the HV value is increasing and SP indicator is shrinking (e.g., $H(s+1) > H(s)$ and $SP(s+1) < SP(s)$), it can be expected that the Pareto front is converging and uniformly distributed. As a result, we keep the flight parameters unchanged.

Remark 4. One potential problem of the SP indicator is that it does not suffice to infer evenness distribution of the solutions, as it does not take into account gaps in the Pareto front. For instance, an hypothetical Pareto front $[(x_1, y_1), (x_1 + \epsilon, y_1 - \epsilon), (x_2, y_2), (x_2 + \epsilon, y_2 - \epsilon)]$ would have a near optimal SP value without regard of the actual positions of the four point. Therefore, follow-up research can be encouraged to deal with this issue.

Algorithm 1 Framework of the HM-AGMOPSO

```
Input: Control parameters of the algorithm $\varpi(0)$, $c_1(0)$, $c_2(0)$, $r_1$, $r_2$, $N_j$, $N_k$,
and $S^{max}$;
Step 1: Initialize the velocity $\nu$ and position $\mu$ vector;
/*Main Loop*/
Step 2: Compute the objective values for each particle;
Step 3: Perform the local gradient operation to update the current swarm;
Step 4: Perform the nondominant sorting;
Step 5: Record the non-dominated solutions in $A(s)$;
Step 6: Find the global best $g(s)$ from $A(s)$;
Step 7: Compute the HV($s$) and SP($s$) indicators;
Step 8: Perform the flight parameter controller via Eqs.(21)-
        (23) to update $(\varpi(s), c_1(s), c_2(s))$;
Step 9: Update $\nu$ and $\mu$ via Eq.(13) and Eq.(14);
Step 10: Check if $S > S^{max}$ is satisfied
        if not, $S = S + 1$ and go back to Step 2;
/*End main iteration*/
Output: The final Pareto set;
```

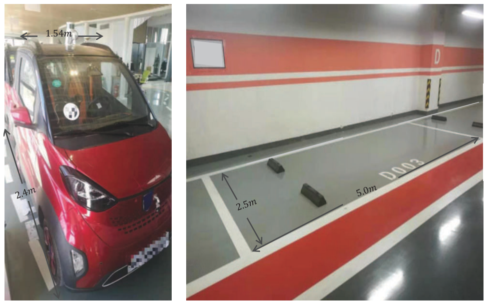
E. Overall Algorithm Structure

According to the above statement and analysis, a pseudocode of the proposed HM-AGMOPSO is constructed in order to better describe the overall procedures of the algorithm (see Algorithm 1).

IV. Multi-Objective Parking Maneuver Results

A. Parameter/Scenario Specification

In this subsection, all the parameters/variables used to generate the multi-objective optimal parking maneuver results are specified. The simulation results were obtained based on a real autonomous vehicle (e.g., illustrated in Fig.2(a)) under a real parking scenario. The vehicle is 2.4m long and 1.54m wide. The values for m, l, and n are set as 0.4m, 1.45m and 0.55m, respectively. As for the parking slot, its size is illustrated in Fig.2(b), where l_{SL}, l_{SW}, and CL are 5.0m, 2.5m, and 4.0m, respectively.



a) Autonomous Vehicle

b) Parking Slot

Fig. 2: Autonomous vehicle and parking space

Regarding the vehicle state variables, their ranges are demonstrated in Table II. Two control path constraints associated with the jerk variable η and the curvature derivative $\dot{k}$ are $\eta \in [-0.5, 0.5]$ and $\dot{k} \in [-0.6, 0.6]$, respectively. The initial condition of the vehicle is $[p_x(0), p_y(0), v(0), \alpha(0), \theta(0), \phi(0)]$=[$9.0m, 2.0m, 0m/s, 0m/s^2$, $0°, 0°$].

TABLE II: Ranges of state variables

Variables	Ranges	Variables	Ranges
p_x	$[-10, 15]$m	p_y	$[-2, 3.5]$m
v	$[-2, 2]$m/s	α	$[-0.75, 0.75]$m/s^2
θ	$[-180°, 180°]$	ϕ	$[-33°, 33°]$

On the other hand, the control variables with regard to the proposed HM-AGMOPSO are illustrated in Table III. These values are used to start the optimization algorithm [16] and they are dynamically adjusted via the strategy developed in Sec III.D.

TABLE III: Control parameters for the HM-AGMOPSO

Control variables	Values/ranges	Control variables	Values/ranges
$\varpi(0)$	$(1+r_1)/2$	r_1, r_2	$[0,1]$
$c_1(0)$	1.49445	N_j	200
$c_2(0)$	1.49445	S^{max}	500
t_f	$[10,50]$	N_k	40

B. Effectiveness of the Constraint Handling Strategy

In this subsection, the performance of applying MOCH-based MOEAs for solving the optimal parking maneuver problem is tested. Three test cases are firstly established and performed. For example: Case i, $i=1,2,3$, stands for minimizing J_i with the consideration of all constraints. The algorithm proposed in this paper is compared to seven approaches of the literature, optimizing these three mission cases. Specifically, the algorithms selected for the comparative study are the penalty function-based GA (denoted as PF-GA) [25], the PF-based PSO algorithm [26], the PF-based artificial bee colony algorithm (PF-ABC) [27], the infeasible rejection GA (IR-GA) [28], the infeasible rejection PSO algorithm (IR-PSO) [28], the MNSGA-II algorithm [29] and the MOEA/D approach [30], respectively. The first five algorithms are typical single-objective optimizers, while the last two algorithms are Pareto-based MOEAs using MOCH.

It should be noted that Case 3 was also considered in [9], where two mission objectives, the coordinate errors and the oriental angle error, were optimized via PF-GA through aggregation. Therefore, we re-perform this mission case using PF-GA and compare the result with the proposed method. Since the implementation of heuristic algorithms might introduce some randomness, it is not reliable to only present the experimental result in one single trial. Therefore, all the techniques were performed for 20 trials independently and the statistical results are tabulated in Table IV.

From the results presented in Table IV, it can be seen that compared with other techniques investigated in this paper, the multi-objective approaches using MOCH strategy can generally produce better solutions. This can be explained that different from other constraint handling strategies which infeasible solutions are always considered worse than the feasible one, the MOCH strategy may accept some infeasible solutions to the next iteration. Sometimes infeasible candidates might contain valuable information and they can be used to guide the evolution direction toward more promising results. If these infeasible solutions are simply removed, it may cause convergence issues especially when a problem contains disconnected feasible regions. Therefore, we suggest that for the optimal parking problem researched in this paper, it is advantageous to consider the total amount of constraint violation as a new objective function.

TABLE IV: Statistical results of Cases 1-3

Different	Average optimal values		
methods	Case 1 (J_1)	Case 2 (J_2)	Case 3 (J_3)
PF-GA	9.1420	1.3088	0.0154
IR-GA	9.0775	1.2927	0.0153
PF-PSO	8.9924	1.3057	0.0146
IR-PSO	9.0996	1.2887	0.0146
PF-ABC	9.1420	1.4219	0.0246
MOEA/D	8.9901	1.2850	0.0146
MNSGA-II	8.9878	1.2797	0.0145
HM-AGMOPSO	8.9680	1.2746	0.0145

To provide more indications with respect to the accuracy and distribution of the obtained solutions, final results for the three mission cases are displayed by box plots in Fig.3.

Fig.3 reflects the ranges of the solution, including the non-averaged best and worst values. These results further confirm that the performance of multi-objective solvers with MOCH strategy tends to be superior to that of other solvers for the three single-objective mission cases. Moreover, it is clear from Fig.3 that final results generated via the proposed method for Case 2 are much better than in any other case. Since the aim of Case 2 is to minimize the smoothness indicator, the use of the gradient operation can provide improvement to the objective and decrease the $\dot{k}$ path constraint violation value at the same time, which in turn improves the convergence of the algorithm.

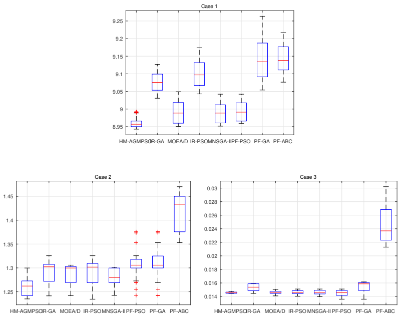


Fig. 3: Box plots of final results

C. Bi-Objective Optimal Parking Results

In this subsection, we perform the bi-objective case study so as to demonstrate the trade-off parking results computed by applying the HM-AGMOPSO algorithm. For example,

- Case 4: minimizing J_1 and J_2.
- Case 5: minimizing J_2 and J_3.
- Case 6: minimizing J_1 and J_3.

Cases 4-6 were performed using the MOCH strategy with V as the additional objective, thereby resulting in a three-objective formulation. The performance of the HM-AGMOPSO algorithm is compared against two Pareto-based multi-objective evolutionary algorithms such as the MNSGA-II algorithm [29] and the MOEA/D approach [30]. It should be noticed that the mission case studied in [10] is equivalent to Case 4 considered in this paper. More precisely, in [10] the authors optimized J_1 and J_2 (e.g., through aggregation) using PF-GA. Therefore, it is worth adding a comparison between the proposed method and the PF-GA method for optimizing all the multi-objective mission cases considered in this paper (denoted as PF-MOGA).

The Pareto front solutions for Cases 4-6 are visualized in Fig.4, from where it can be observed that minimizing the path length and minimizing the path smoothness indicator are two contradicting objectives. This can be explained that if the vehicle wants to have a smoother parking trajectory, the variation of the instantaneous curvature, which is mainly affected by the angular velocity of the vehicle, will be decreased. This means that the vehicle will not use the maximum allowable control and the manoeuver time will be longer. Consequently, the total path length will be increased. Similarly, in Case 5 minimizing the path smoothness and optimizing the parking achievement indicator display a contradicting relationship. To park the vehicle in a desirable position, sharply varied control signals are usually required and this might result in an increase with respect to the smoothness indicator.

D. Three-Objective Optimal Parking Results

In this subsection, the three mission objectives are considered simultaneously (denoted as Case 7). Similar with Cases 4-6, Case 7 was performed using the MOCH strategy with V as the additional objective (e.g., $J_4=\min(V_G+V_I)$). The resulting four-objective

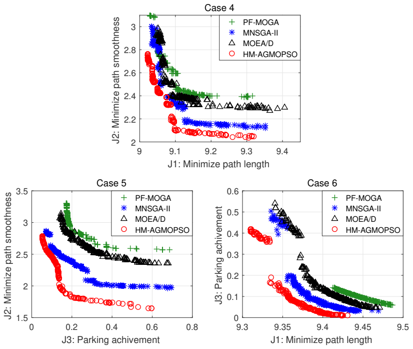


Fig. 4: Pareto solution for Cases 4-6

formulation was solved by applying different algorithms stated previously.

The obtained Pareto results are displayed in Fig.5, where the solutions are projected onto two plane (e.g., J_1 versus J_2 and J_2 versus J_3). From the trade-off front, it is clear to know that how the optimal path length result reduces the path smoothness indicator and how the optimal path smoothness solution affects the optimality of the parking achievement objective. These solutions (along the obtained front) are considered equally good, but in practical situations, only one optimized solution is needed. Therefore, a proper compromise between extreme solutions should be usually made. Based on the obtained Pareto-optimal solutions, the intermediate point $[J_1, J_2, J_3]$=$[9.10, 2.10, 0.41]$ can be recognised as the best compromised solution. This point is obtained by searching the solution p_i among the final Pareto set P via $p_i^* = \arg\max_{p_i \in P} \sum_{i=1}^{3}(1 - \frac{J_i(p_i) - J_i^*}{J_i^{max} - J_i^*})$. Here J_i^* and J_i^{max} are the extreme (best and worst) values of J_i. This procedure can be understood as searching a solution along the obtained Pareto front such that the distance between this point and the origin can be minimized. Similar procedures can also be applied to Cases 4-6 to calculate the compromised solutions.

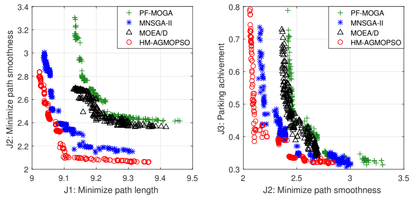


Fig. 5: Pareto front for Case 7

E. Analysis of Results

Regarding the performance of different algorithms (e.g., the Pareto-based MOEAs and the aggregation approach PF-MOGA), as can be seen from Fig.4, the proposed HM-AGMOPSO can generally perform better than its counterparts for mission Cases 4-6. This conclusion can be reflected by the obtained Pareto front solution. That is, the MNSGA-II, MOEA/D, and MOGA solutions/results are generally covered by the HM-AGMOPSO solutions. In addition, the Pareto-results produced by HM-AGMOPSO tend to be more uniform than the others in the objective space. This can be attributed to the implementation of the adaptive flight parameter controller which can balance the exploitation and exploration of the evolution process, thus making the solution more optimal and uniform. According to the Case 7 results shown in Fig.5, the Pareto solutions produced by MNSGA-II, MOEA/D and MOGA are again covered by the solution calculated via the proposed method. Moreover, the HM-AGMOPSO solutions tend to be more well-distributed than other techniques.

Remark 5. It should be noted that even if most of literature algorithms are single-objective optimizers, they still can provide useful baselines in multi-objective contexts. For example, if we directly apply PF-PSO algorithm to optimize individual objective J_1 in Case 4, the values for J_1 and J_2 are 8.971 and 3.784, respectively. Based on the result presented in Fig.4, although J_2 is significantly sacrificed, this solution point (J_1, J_2) can be treated as an extreme solution. However, if we apply this algorithm to optimize a normalised aggregation of J_1 and J_2 to find a compromised solution, the final values of (J_1, J_2) are dominated by the solutions obtained via the Pareto-based MOEAs. Similar results can also be obtained for the other four single-objective optimizers investigated in this paper. Therefore, it is more suitable to apply Pareto-based MOEAs to address the multi-objective parking maneuver planning cases.

In order to analyze the quality of the obtained Pareto solution and the performance of different multi-objective solvers, a statistical study was carried out with respect to the HV value for different mission cases. To compute the HV indicator, the obtained Pareto-optimal solutions are firstly normalized. This step is achieved by placing all non-dominated solutions found via different algorithms in one set. Then we extract the minimum and maximum values for each objective and normalize all the solutions. The reference points for mission Cases 4-6 are set to $R = [1.1, 1.1]$, while the reference points for Case 7 is assigned as $R = [1.1, 1.1, 1.1]$. Then the statistical HV results for Case 4-7 are summarised in Table V, where all results were averaged on 20 independent trials.

TABLE V: Best, average and worst HV results for Cases 4-7

	HV			
Case No.	PF-MOGA [10]	MNSGA-II [29]	MOEA/D [30]	HM-AGMOPSO
4	0.4565 0.4563 0.4561	0.5448 0.5446 0.5442	0.4931 0.4930 0.4928	**0.5761** **0.5758** **0.5756**
5	0.6037 0.6031 0.6026	0.8664 0.8650 0.8641	0.6936 0.6928 0.6923	**0.9442** **0.9429** **0.9412**
6	0.7769 0.7767 0.7766	0.8172 0.8169 0.8166	0.8018 0.8014 0.8011	**0.8375** **0.8373** **0.8370**
7	0.4563 0.4556 0.4552	0.5433 0.5427 0.5423	0.4617 0.4614 0.4610	**0.5648** **0.5646** **0.5641**

According to the data presented in Table V, it is obvious that the HM-AGMOPSO technique designed in this paper is able to achieve the highest HV values among the four multi-objective solvers. Moreover, all the Pareto-based MOEAs can statistically outperform the aggregation-based optimizer for the considered parking maneuver planning cases.

To further study the robustness and convergence of different Pareto-based MOEAs, the evolution of the HV value along the iterations is given concerns. Fig.6 displays the HV evolution histories for Case 4-7. From Fig.6, it is obvious that compared with other Pareto-based MOEAs, the proposed method tends to converge to more optimal solution sets in relatively-fewer number of iterations for all the considered mission cases. This further confirms the superiority of applying the proposed strategy.

In addition, to contrast the significance of the Pareto-optimal solutions obtained via different multi-objective optimizers, the Wilcoxon signed-ranks test is adopted (encouraged and demonstrated in [31]). This is a typical non-parametric statistical hypothesis test and it has the capability to rank the difference in performance between various multi-objective optimization methods studied in this section. The test was carried out with respect to the HV performance metric and the results are tabulated in Table VI. In this table, the p-values

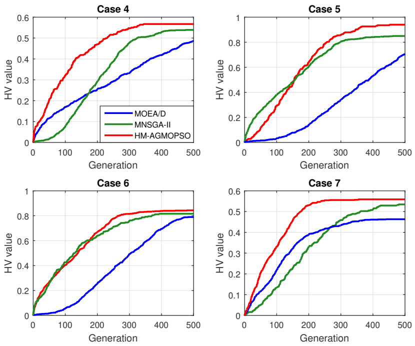


Fig. 6: HV evolution histories for Case 4-7

and h-values calculated for all pairwise comparisons concerning HM-AGMOPSO are reported. It is worth remarking that $h = 1$ reflects that a significant improvement in terms of the HV performance can be achieved with $\varepsilon = 0.05$ (level of significance). As can be observed from Table VI, the proposed method demonstrates a significant improvement over other multi-objective optimizers for the considered mission cases. These statistical results further improve the confidence in the verification of enhanced performance of the proposed strategy.

TABLE VI: Non-parametric statistical test results

Pairwise comparison	Case 4		Case 5	
	p-value	h-value	p-value	h-value
HM-AGMOPSO vs PF-MOGA	<e-04	1	<e-04	1
HM-AGMOPSO vs MNSGA-II	<e-04	1	<e-04	1
HM-AGMOPSO vs MOEA/D	<e-04	1	<e-04	1
Pairwise comparison	Case 6		Case 7	
	p-value	h-value	p-value	h-value
HM-AGMOPSO vs PF-MOGA	4.88e-04	1	<e-04	1
HM-AGMOPSO vs MNSGA-II	4.88e-04	1	<e-04	1
HM-AGMOPSO vs MOEA/D	4.88e-04	1	<e-04	1

TABLE VII: Best, average and worst HV results via MOPSO-based methods

	HV			
Case No.	MOPSO	GMOPSO	HM-AMOPSO	HM-AGMOPSO
4	0.5017	0.5114	0.5538	0.5761
	0.5014	0.5113	0.5536	0.5758
	0.5012	0.5111	0.5533	0.5756
5	0.7227	0.7538	0.8829	0.9442
	0.7225	0.7537	0.8827	0.9429
	0.7223	0.7536	0.8826	0.9412
6	0.7816	0.7927	0.8205	0.8375
	0.7815	0.7924	0.8203	0.8373
	0.7813	0.7922	0.8201	0.8370
7	0.4624	0.4836	0.5516	0.5648
	0.4621	0.4835	0.5513	0.5646
	0.4619	0.4834	0.5512	0.5641

Though a better performance can be achieved via the proposed method, it is not clear whether the use of the gradient operation or the self-adaptation strategy will have contributions to the algorithm. To address this concern, experiments were performed to compare the results obtained using the approach with no adjustment (e.g., the original MOPSO), the approach with only the gradient operation (denoted as GMOPSO), and the approach with only the self adaptation strategy (denoted as HM-AMOPSO). The aim of carrying out this analysis is to study and appreciate the contributions made by each individual component.

The corresponding HV results for different mission cases are tabulated in Table VII, from where it can be observed that both the gradient operation and the flight parameter controller can have positive influences on the algorithm performance for all the considered mission cases. Furthermore, compared with the gradient operation, the use of the flight parameter controller can lead to a more significant improvement to the algorithm.

V. Experimental Results

Apart from the simulation studies, it is also necessary to carry out a set of experiments in order to verify the reliability of the planned parking trajectory obtained via the proposed HM-AGMOPSO method.

A. Tracking of Planned Trajectory

To fulfill the automatic parking, a controller should be used so as to track the pre-planned parking movement. The control method used in this work is a simple application of the result in [17]. The control input to be applied to the vehicle is $u = [v, \omega]$. Denoting the pre-planned trajectory as $[p_x^r, p_y^r, \theta^r, \phi^r]$, based on Eq.(1), the tracking error vector $[p_x^e, p_y^e, \theta^e, \phi^e]$ follows the equation:

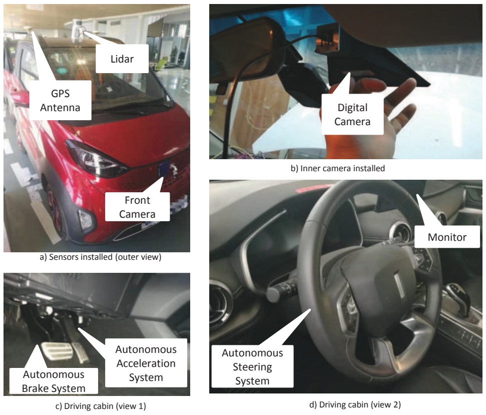
$$\begin{bmatrix} \dot{p}_x^e \\ \dot{p}_y^e \\ \dot{\theta}^e \\ \dot{\phi}^e \end{bmatrix} = \begin{bmatrix} v\cos\theta - v^r\cos\theta^r \\ v\sin\theta - v^r\sin\theta^r \\ v\tan\phi - v^r\tan\phi^r \\ \omega - \omega^r \end{bmatrix} \tag{24}$$

where $[v^r, \omega^r]$ is the reference input. The aim for the controller is to produce an input vector such that the current vehicle's motion can track the desired trajectory. That is, $(p_x^e, p_y^e, \theta^e, \phi^e)$ should be bounded and converge to a small neighborhood of zero. According to [17], a simple control law can be designed in the form of:

$$\begin{cases} v = (v^r\cos\theta^r - \chi(p_x^e))/\cos\theta \\ \omega = \dot{\varphi}/(1+\varphi^2) \\ \varphi = (-\sigma p_y^e - \omega - \beta\theta^e + v^r\tan\phi^r)/v \end{cases} \tag{25}$$

where $\chi, \sigma, \beta > 0$.

B. Experimental Setup



a) Sensors installed (outer view)

b) Inner camera installed

c) Driving cabin (view 1)

d) Driving cabin (view 2)

Fig. 7: Experimental setup: Sensors and Systems

The experiment was performed on the autonomous vehicle illustrated in Fig.2. This intelligent vehicle uses the drive-by-wire technology and is equipped with sensors for mapping and localization. Fig.7(a) and Fig.7(b) illustrate some important sensors installed on the vehicle, while Fig.7(c) and Fig.7(d) demonstrate the driving cabin of the vehicle. Besides, to achieve a high computational performance, the NVIDIA Jetson TX2 kit (dual-core NVIDIA Denver2 and quad-core ARM Cortex-A57, 8GB 128-bit LPDDR4 memory and integrated 256-core NVIDIA Pascal GPU) is applied as the onboard computer of the autonomous vehicle. As for the control box, a graphical illustration is given by Fig.8, where the in-vehicle

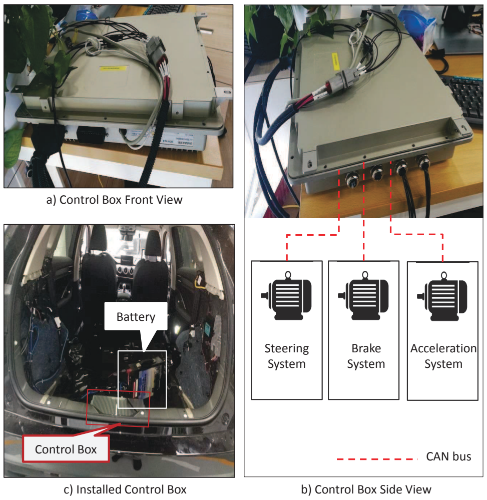


Fig. 8: Experimental setup: Control box

communication between the control box and three control subsystems is achieved via the controller area network (CAN). The battery is used as the power supplier of the control box (as shown in Fig.8(c)).

C. Results and Discussions

The real-time capability as well as the impact of initial position perturbations are firstly analyzed. To boost the processing speed of the optimization algorithm proposed in this paper, we apply a homogeneous parallel computing structure [14]. Specifically, the idea is to separate the original population and archive to several subpopulations and subarchives. In this way, the entire optimization task can be equally distributed to a number of available processors and processed in parallel, thereby significantly reducing the required processing time. A graphical demonstration regarding this homogeneous parallel computing structure is given by Fig.9, from where it is obvious that for each processor, the HM-AGMOPSO optimization process is executed and updates will be made on the subpopulation and subarchive. The sizes of the subpopulation and subarchive, along with the number of processor, are assigned as 40, 25 and 6, respectively. Other algorithm-related parameters remain the same as specified in Table III. Using this configuration, we have tested the algorithm over 100 trials and the average as well as the worst-case processing times are 2.714s and 3.117s, respectively. These results are almost 50 times lesser than the one obtained using a single processor, which confirms the advantage of applying the parallel computing structure.

The control algorithm is then applied to track the pre-planned trajectory for Case 4. The initial condition was perturbed by $[p_x(0), p_y(0), \theta(0)] = [p_x(0), p_y(0), \theta(0)] + [\delta p_x, \delta p_y, \delta\theta]$, where $|\delta p_x| \leq 0.4$m, $|\delta p_y| \leq 0.2$m and $|\delta\theta| \leq 0.5^\circ$. 500 Monte-Carlo tests were performed and the error evolution profiles are displayed in Figs.10(a)-(b), while the distribution and histogram regarding the average execution time per control action are plotted in Fig.10(c) and Fig.10(d). From Figs.10(a)-(b), it is obvious that although the initial condition perturbations can diverge the actual trajectory from the reference, the error will be steered to around zero via the control. Moreover, based on the algorithm execution results presented in Fig.10(c) and Fig.10(d), the real-time applicability can be justified.

In terms of the real-world experiments, Fig.11 demonstrates the automatic parking experimental results. Specifically, Fig.11 illustrates the actual parking movements of the autonomous vehicle for mission Case 4. From the test, although the pre-planned path is not accurately followed, the vehicle can successfully fulfill the parking task for the mission without colliding with the slot boundary. This further confirms the reliability of the proposed trajectory planning method and the produced parking trajectory. Videos were also submitted to prove this conclusion in real parking situations [1].

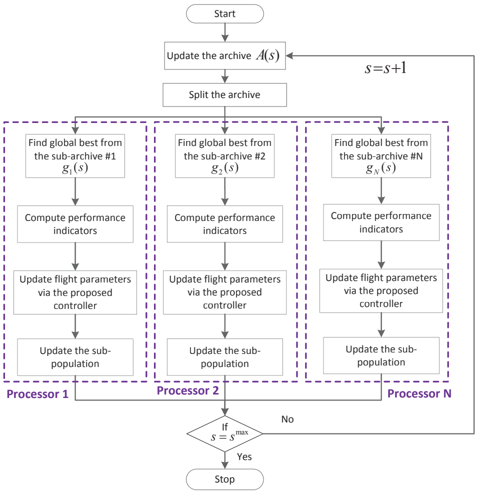

Start
Update the archive $A(s)$
$s=s+1$
Split the archive
Find global best from the sub-archive #1 $g_1(s)$
Find global best from the sub-archive #2 $g_2(s)$
Find global best from the sub-archive #N $g_N(s)$
Compute performance indicators
Update flight parameters via the proposed controller
Update the sub-population
Processor 1
Processor 2
Processor N
If $s = s^{max}$
No
Yes
Stop

Fig. 9: HM-AGMOPSO parallel computing structure

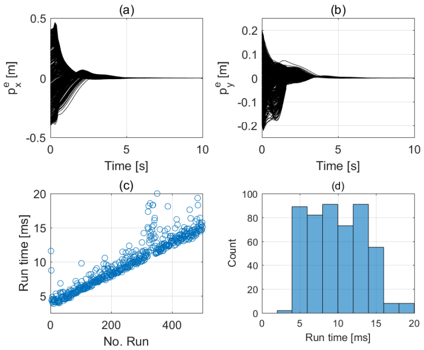


Fig. 10: 500 real-time Monte-Carlo test

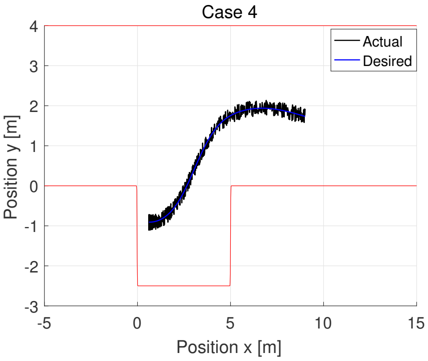


Fig. 11: Actual parking movement result

VI. Conclusion

In this investigation, a HM-AGMOPSO approach was designed and implemented to solve the multi-objective wheeled vehicle parking motion planning problem. To improve the local search ability and better control the evolution process of the algorithm, a local

[1] https://youtu.be/OQLIKZrX_Zg or https://v.youku.com/v_show/id_XNDA5OTgxODQ2MA

gradient operation, along with a novel flight parameter controller, was designed and embedded in the MOPSO framework. A detailed experimental study was performed so as to demonstrate the key features of the multi-objective solutions and the reliability of the HM-AGMOPSO. From the simulation results, we have obtained that:

1) Considering the total constraint violation as an additional objective can effectively improve the quality of the final results for the investigated problem (supported by the results in Table IV and Fig.3).
2) For optimizing multi-objective parking mission cases, the proposed HM-AGMOPSO algorithm is able to produce non-dominated results with more uniform and optimal distribution (supported by the results in Table V and Figs.4-6).
3) The proposed gradient operation and flight parameter controller can indeed provide contributions to the algorithm (supported by the data in Table VII).

Field tests were performed and the results confirm the availability of using the proposed design in real-wold applications. Thus, we suggest using the proposed algorithm to offer an effective Pareto-optimal solution for the multi-objective parking trajectory optimization problem

References

[1] B. Tian, W. Fan, R. Su, and Q. Zong, "Real-time trajectory and attitude coordination control for reusable launch vehicle in reentry phase," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 3, pp. 1639–1650, 2015.

[2] C. Tang, Y. Wang, S. Wang, R. Wang, and M. Tan, "Floating autonomous manipulation of the underwater biomimetic vehicle-manipulator system: Methodology and verification," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 6, pp. 4861–4870, 2018.

[3] T. Fraichard and A. Scheuer, "From Reeds and Shepp's to continuous-curvature paths," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 1025–1035, 2004.

[4] T.-C. Liang, J.-S. Liu, G.-T. Hung, and Y.-Z. Chang, "Practical and flexible path planning for car-like mobile robot using maximal-curvature cubic spiral," *Robotics and Autonomous Systems*, vol. 52, no. 4, pp. 312–335, 2005.

[5] 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.

[6] P. Zips, M. Bock, and A. Kugi, "Optimisation based path planning for car parking in narrow environments," *Robotics and Autonomous Systems*, vol. 79, pp. 1–11, 2016.

[7] W. Liu, Z. Li, L. Li, and F. Y. Wang, "Parking like a human: A direct trajectory planning solution," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3388–3397, 2017.

[8] S. Upadhyay and A. Ratnoo, "Continuous-curvature path planning with obstacle avoidance using four parameter logistic curves," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 609–616, 2016.

[9] D. Maravall and J. de Lope, "Multi-objective dynamic optimization with genetic algorithms for automatic parking," *Soft Computing*, vol. 11, no. 3, pp. 249–257, 2007.

[10] G. Lini, A. Piazzi, and L. Consolini, "Multi-optimization of η^3-splines for autonomous parking," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, 2011, Conference Proceedings, pp. 6367–6372.

[11] C. A. C. Coello and M. S. Lechuga, "MOPSO: a proposal for multiple objective particle swarm optimization," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02*, vol. 2, Conference Proceedings, pp. 1051–1056 vol.2.

[12] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, 2004.

[13] M. Reyes-Sierra and C. A. C. Coello, "Multi-objective particle swarm optimizers: A survey of the state-of-the-art," *International Journal of Computational Intelligence Research*, vol. 2-3, pp. 287–308, 2006.

[14] K. B. Lee, H. Myung, and J. H. Kim, "Online multiobjective evolutionary approach for navigation of humanoid robots," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 9, pp. 5586–5597, 2015.

[15] L. Guo, H. Zhang, M. Galea, J. Li, and C. Gerada, "Multiobjective optimization of a magnetically levitated planar motor with multilayer windings," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 6, pp. 3522–3532, 2016.

[16] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, and Y. Xia, "Two-stage trajectory optimization for autonomous ground vehicles parking maneuver," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2018.

[17] A. Ailon, N. Berman, and S. Arogeti, "On controllability and trajectory tracking of a kinematic vehicle model," *Automatica*, vol. 41, no. 5, pp. 889–896, 2005.

[18] F. You, R. Zhang, G. Lie, H. Wang, H. Wen, and J. Xu, "Trajectory planning and tracking control for autonomous lane change maneuver based on the cooperative vehicle infrastructure system," *Expert Systems with Applications*, vol. 42, no. 14, pp. 5932–5946, 2015.

[19] R. Chai, A. Savvaris, A. Tsourdos, and S. Chai, "Multi-objective trajectory optimization of space manoeuvre vehicle using adaptive differential evolution and modified game theory," *Acta Astronautica*, vol. 136, pp. 273–280, 2017.

[20] H. Han, W. Lu, and J. Qiao, "An adaptive multiobjective particle swarm optimization based on multiple adaptive methods," *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2754–2767, 2017.

[21] J. Knowles and D. Corne, "On metrics for comparing nondominated sets," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02*, vol. 1, Conference Proceedings, pp. 711–716 vol.1.

[22] T. Okabe, Y. Jin, and B. Sendhoff, "A critical survey of performance indices for multi-objective optimisation," in *The 2003 Congress on Evolutionary Computation, 2003. CEC '03.*, vol. 2, Conference Proceedings, pp. 878–885 Vol.2.

[23] N. Riquelme, C. V. Lucken, and B. Baran, "Performance metrics in multi-objective optimization," in *2015 Latin American Computing Conference (CLEI)*, Conference Proceedings, pp. 1–11.

[24] J. R. Schott, "Fault tolerant design using single and multicriteria genetic algorithm optimization," *MSc Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts*, 1995.

[25] J. Liu, K. L. Teo, X. Wang, and C. Wu, "An exact penalty function-based differential search algorithm for constrained global optimization," *Soft Computing*, vol. 20, no. 4, pp. 1305–1313, 2016.

[26] J. J. Kim and J. J. Lee, "Trajectory optimization with particle swarm optimization for manipulator motion planning," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 620–631, 2015.

[27] H. Duan and S. Li, "Artificial bee colony based direct collocation for reentry trajectory optimization of hypersonic vehicle," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 1, pp. 615–626, 2015.

[28] Y. G. Woldesenbet, G. G. Yen, and B. G. Tessema, "Constraint handling in multiobjective evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 514–525, 2009.

[29] B. Ji, X. Yuan, and Y. Yuan, "Modified nsga-ii for solving continuous berth allocation problem: Using multiobjective constraint-handling strategy," *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2885–2895, 2017.

[30] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[31] J. Derrac, S. Garcia, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

Runqi Chai (S'15-M'18) received the B.S. degree in information and computing science from the North China University of Technology, Beijing, China, in 2015 and the Ph.D. degree in Aerospace Engineering from Cranfield University, Cranfield, U.K, in August 2018. He is currently a research fellow at Cranfield University. His research interests include trajectory optimization, networked control systems, multi-agent control systems, and autonomous vehicle motion planning.

Antonios Tsourdos (M'99) obtained a Ph.D. on nonlinear robust missile autopilot design and analysis from Cranfield University, in 1999. He is a Professor of Autonomous Systems and Control with Cranfield University. He was appointed Head of the Autonomous Systems Group in 2007, Head of the Centre of Autonomous and Cyber-Physical Systems in 2012 and Director of Research - Aerospace, Transport and Manufacturing in 2015. He leads the research theme on autonomous systems within the School of Aerospace, Transport and Manufacturing at Cranfield University. He has diverse expertise in both unmanned and autonomous vehicles as well as networked systems.He conducts basic and applied research in the fields of guidance, control and navigation for single and multiple unmanned autonomous vehicles as well as research on cyber-physical systems.

Al Savvaris (M'08) received the M.Eng. degree in aerospace systems engineering from the University of Hertfordshire, Hertfordshire, U.K., in 1998 and the Ph.D. degree in radiowave propagation and system design from the University of South Wales, Pontypridd, U.K., in 2004. His research interests include trajectory optimization, networked control systems, and aircraft control systems.

Senchun Chai (M'19-SM'19) received the B.S and Master degree from Beijing Institute of Technology, Beijing, China from 1997 to 2004 and the Ph.D. degree in Networked Control System from University of South Wales, Pontypridd, U.K., in 2007.

He is currently an associated professor of School of Automation with Beijing Institute of Technology. His current research interests focus on flight control system, networked control systems, embedded systems and multi-agent control systems.

Yuanqing Xia (M'15-SM'16) was born in Anhui Province, China, in 1971. He received the B.S. degree from the Department of Mathematics, Chuzhou University, Chuzhou, China, in 1991, the M.S. degree in fundamental mathematics from Anhui University, Wuhu, China, in 1998, and the Ph.D. degree in control theory and control engineering from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 2001. His current research interests are in the fields of networked control systems, robust control and signal processing, active disturbance rejection control and flight control. He has published 8 monographs with Springer and Wiley, and more than 200 papers in journals. He has obtained Second Award of the Beijing Municipal Science and Technology (No. 1) in 2010, Second National Award for Science and Technology (No. 2) in 2011, and Second Natural Science Award of The Ministry of Education (No. 1) in 2012. He is a Deputy Editor of the Journal of the Beijing Institute of Technology, Associate Editor of Acta Automatica Sinica, Control Theory and Applications, the International Journal of Innovative Computing, Information and Control, and the International Journal of Automation and Computing.

C. L. Philip Chen (S'88-M'88-SM'94-F'07) received the M.S. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 1985 and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1988.

He is a Chair Professor with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau, China. Being a Program Evaluator of the Accreditation Board of Engineering and Technology Education in the U.S., for computer engineering, electrical engineering, and software engineering programs, he successfully architects the University of Macau’ s Engineering and Computer Science programs receiving accreditations from Washington/Seoul Accord through Hong Kong Institute of Engineers (HKIE), of which is considered as his utmost contribution in engineering/computer science education for Macau as the former Dean of the Faculty. His current research interests include systems, cybernetics, and computational intelligence.

Dr. Chen was a recipient of the 2016 Outstanding Electrical and Computer Engineers Award from his alma mater, Purdue University, after he graduated from the University of Michigan at Ann Arbor, Ann Arbor, MI, USA. He was the IEEE SMC Society President from 2012 to 2013 and is currently a Vice President of Chinese Association of Automation (CAA). He is the Editor-in-Chief of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS and an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS and IEEE TRANSACTIONS ON CYBERNETICS. He was the Chair of TC 9.1 Economic and Business Systems of International Federation of Automatic Control from 2015 to 2017. He is a fellow of AAAS, IAPR, CAA, and HKIE.