IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 21 October 2023, accepted 8 November 2023, date of publication 16 November 2023,
date of current version 8 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3334148

RESEARCH ARTICLE

PID Tuning Using Differential Evolution With Success-Based Particle Adaptations

VICTOR PARQUE [1,2], (Member, IEEE), AND ALAA KHALIFA [3]
[1]Department of Modern Mechanical Engineering, Waseda University, Shinjuku, Tokyo 169-8555, Japan
[2]Department of Mechatronics and Robotics, Egypt-Japan University of Science and Technology, New Borg-El Arab City, Alexandria 21934, Egypt
[3]Department of Industrial Electronics and Control Engineering, Faculty of Electronic Engineering, Menoufia University, Menouf 32952, Egypt

Corresponding author: Victor Parque (parque@aoni.waseda.jp)

This work was supported in part by Waseda University and Japan International Cooperation Agency (JICA) under Egypt-Japan University of Science and Technology (E-JUST) project framework.

ABSTRACT Proportional-Integral-Derivative (PID) is a simple and intuitive feedback-based control mechanism being useful to track set points and to reject disturbances. A key question in gradient-free optimization is to ascertain whether the class of optimization algorithms based on the difference of vectors generalize reasonably well to tackle a large class of PID control problems. For generalization and practical purposes, it would be desirable to render algorithms being able to tune PID controllers over a diverse and large set of control problems/tasks with minimal human intervention (self-adaptation features), and under tight computational budgets. In this paper, aiming to fill the above-mentioned gap, we propose and investigate the effectiveness of a new class of algorithm based on the difference of vectors and self-adaptation mechanisms for PID tuning. As such, we introduce a new class of Differential Evolution with success-based Particle Adaptations (DEPA), which unifies the principles of difference of vectors, particle schemes and trial/parameter adaptation through archive (memory) mechanisms. Our computational simulations using a large/relevant set of 25 control problem instances (tracking of linear, nonlinear, continuous, and discontinuous trajectories in motor position control, motor velocity control, magnetic levitation, inverted pendulum, crane stabilization), and the comparisons with a large set of closely related optimization algorithms, and their extended adaptive variants (23 optimization algorithms in total) has shown the outperforming benefits of the proposed approach in convergence performance under tight function evaluation budgets (1000 function evaluations). Also, the experiments on a real-world inverted pendulum device show the potential for transferability of the learned gains to unseen situations during training. Furthermore, we evaluated the algorithmic extension and the generalization towards diverse fitness landscapes in the CEC 2017 benchmark suite, showing the attractive/outperforming performance overall problem instances. In particular, the proposed framework performed better in 558 control instances when using ISE as a performance metric, and in 358 control instances when using IAE, ITAE and ITSE as performance metrics. Also, the algorithmic extension for general optimization landscapes performed better than the related algorithms in 182, 215 and 235 problem instances of CEC 2017 benchmark suite for 10, 30 and 50 dimensions, respectively. Our obtained results have the potential to further advance towards developing efficient and self-adaptive optimization algorithms based on the difference of vectors, which may find use in a wider set of optimization and control problems.

INDEX TERMS PID control, PID tuning, differential evolution, motor velocity control, motor position control, magnetic levitation, inverted pendulum, crane stabilization, optimization.

The associate editor coordinating the review of this manuscript and approving it for publication was Min Wang.

I. INTRODUCTION

A. BACKGROUND

Proportional-Integral-Derivative (PID) is a feedback-based control mechanism useful to stabilize regulated systems [1]. Its ubiquitous use in industrial systems is mainly due to the simplicity and intuitiveness of its operation through Digital Control mechanisms and Programmable Logic Controllers (PLCs) [2]. Also, the reliance on parameters that encode the proportional nature of the error gap, the accumulation of error, and the variability of error allow the precise modulation of logical, digital, and physical systems [3]. PID-based control is independent of the dynamics of the system, being attractive to allow the adaptive hybrids seamlessly [4], [5], [6]. As such, PID is often used in servo control to track a set point and in regulatory control to reject disturbances.

Ziegler-Nichols (Z-N) [7] and Cohen-Coon [8] are popular schemes for PID tuning. In Z-N method, the plant is often modeled as a first-order lag plus delay (FOLPD) system whose parameters are estimated by a tangent and point, and PID parameters are derived by formulae. Also, there exist more than a thousand tuning rules to minimize tailored plant conditions and control performance criteria [9]. In the context of optimization, particle filtering improves the computation complexity [10], constrained optimization enables the inclusion of robustness constraints [11], and the ensembles of Kalman Filters use fewer particles while allowing a slight increase in computational time for complex transfer functions [12]. A comparison of tuning rules for fractional-order PID control is provided by [13]. Furthermore, derivative-based and interior-point algorithms such as [14] approach PID control and parameter tuning by iterative sampling.

B. CHALLENGES

PID parameter optimization (or sometimes referred as parameter tuning) is key for the effective and the robust control performance in user-defined tasks in manipulators [15], magnetic levitation [16], industrial plants [17], higher order systems [18], and other industrial settings. Nonetheless, tuning the parameters of PID control systems is challenging due to the computationally expensive evaluations of real-world surrogates and the requirement for fast adaptation to new control tasks [6], [19]. From the viewpoint of desirable performance in user-defined control tasks, systems governed by PID controllers are required to undergo several design cycles to find suitable parameter configurations, as in greenhouse climate control, in which the Levenberg-Marquardt optimization searches for desirable parameters of heating, ventilation, fogging, and CO_2 [20]. It would be desirable to find relevant and generalizable PID parameters by efficient and self-adaptive optimization algorithms, regardless of the dynamics of the controlled system.

C. LITERATURE REVIEW

Generally speaking, the class of gradient-free population-based optimization heuristics have the potential to find high-performing PID parameters regardless of the dynamics of the surrogate model. Several nature-inspired heuristics for PID parameter optimization have emerged in the field in recent years. For ease of reference, Table 1 and Table 2 summarize the related works from recent years and the seminal works from the 90s in the field of PID parameter tuning and optimization. Table 1 and Table 2 describe the references, ordered by year, and both the kind of optimization algorithms and the control problem tackled in each study. The acronyms denoting the algorithms are described in appendix A. Here, for instance, the seminal works in the 90s related to PID tuning used Genetic Algorithms (GA), such as in [21] and [22], to tackle nonlinearities in industrial practice. Since then, several algorithms have been studied/proposed, among which Particle Swarm Optimization (PSO) and Differential Evolution (DE) schemes have received systematic attention from the community. For instance, within the class of swarm-inspired algorithm, [23] used PSO for PID optimization in hypersonic vehicle control, [24] used Quantitative Feedback Theory (QFT) and PSO for a DC-DC converter, [25] used Kriging surrogates and PSO for fractional order PID in a production-inventory control system, [26] used PSO in magnetic-levitation control, [27] approached the electro-hydraulic servo control system, [28] combined GA and PSO for Gaussian adaptive PID control of a DC-DC converter, [29] modified the inertia weight of PSO as a piecewise nonlinear function to consider the effects of PID parameters on control response, [30] used a fractional order PSO in which the velocity term implements a non-integer order equation to smooth the transition and exploration of the search space.

Among the tackled control problems in Table 1-2, the voltage regulation, the position/velocity control in motors and machines have received attention as well. For instance, the works in [5], [29], [32], [51], [60], [70], and [77] approached the voltage regulation control problem, and the works in [31], [36], [54], and [68] tackled the motor velocity control problem. Most of the above-mentioned works tackled the PID tuning problem by a single-objective optimization, whereas a few works such as [35], [76], [80], and [86] used multi-objective approaches, such as [76] which used a multi-objective state transition algorithm for PID-based goethite process control.

Furthermore, since heuristics use distinct forms of stochastic sampling of the search space, it has been often a desirable practice to compare several optimization heuristics to address the performance of optimized PID controllers. By observing at the algorithms in Table 1 and Table 2, [70] and [87] studied the largest number of heuristics: [87] used eight DE mutation schemes within the context of a parallel robot, and [70] used eight nature-inspired algorithms within the context of automatic voltage regulation. The particular attention in transfer functions involving the voltage regulation control problems has enabled the practice of re-using optimized parameters and ease the task of comparing the performance of several heuristics, such as in [51], [60], [70], and [77].

TABLE 1. Recent works on PID parameter optimization (part 1).

Year	Ref.	Algorithm	Control Instances
2023	[31]	QGAs (3), PSOs (2)	Motor velocity control
2023	[32]	MPA, MPSEDA	Voltage regulator (AVR)
2023	[33]	SSQPSO	Transfer function (spring-mass system)
2023	[34]	PSO	Temperature control (cooling coil)
2023	[35]	NSDE-R	Temperature control (reactor model)
2023	[36]	TS	Motor velocity control
2023	[37]	SADE, AVOA	Temperature control (parabolic trough)
2023	[38]	AOA, GTO, SOA, PPA, HGTOAOA	Particle position control
2023	[39]	WOA (2), FFA	Rotary pendulum (position control)
2023	[40]	MH, CS, PSO	Mass-spring damper
2023	[41]	SCSO	Motor position control
2023	[42]	GA	Smoke control in tunnel fires
2023	[43]	MOTEO, PSO, GOA	Altitude control (quadrotor)
2022	[44]	WCA, IWCA, IWCA-CS, PSO, CS	Clock synchronization
2022	[45]	DE	Suspension system (quarter car)
2022	[46]	PSO	pH control (water and fertilizer)
2022	[47]	D3QN	Vehicle velocity control (AVL chassis)
2022	[24]	PSO	DC-DC (boost) converter
2022	[48]	AMSGrad	Level control (2-tank)
2022	[49]	DEs (4)	Circle tracking (cartesian robot)
2022	[50]	MFO	DC-DC converter (inductor)
2022	[51]	MSFA, SFA, SCA, SA, CAS, PSO, NSGA-II	Voltage regulator (AVR)
2022	[52]	DRL	Level-flow control
2022	[53]	MMA, MA, PSO, SSA, WOA	Transfer function (water turbine speed)
2022	[54]	PSOs (3), WOA, SSO	Motor velocity control
2021	[25]	PSO, GWO, ALO, ACO, GA	Transfer functions (3), Production-Inventory (1)
2021	[55]	FMO, PSO	Transfer functions (3)
2021	[56]	MM-MADRL, GA, PSO, SSA, DE	Quadcopter altitude control
2021	[57]	DEs (4), PSO (2)	Inverted Pendulum, Magnetic Levitation
2021	[58]	DEs (4), PSO (2)	Inverted Pendulum
2021	[59]	GD	Response time (cloud software)
2021	[60]	SCA, SA, CAS, PSO, GA, BBO, ABC	Voltage regulator (AVR)
2021	[61]	ACO	DC-DC converter
2020	[62]	GD	Bulk resumption (ore mining)
2020	[23]	PSO	Hypersonic vehicle in the pitch plane
2020	[30]	PSOs (2), GA	Magnetic levitation
2020	[28]	PSO, GA	DC–DC converter

TABLE 2. Recent works on PID parameter optimization (part 2).

Year	Ref.	Algorithm	Control Instances
2020	[63]	QSLP, WOA, SLP, PSO	Electro-hydraulic servo
2020	[64]	PSO	Canonical tank system
2020	[65]	ALO	Transfer functions
2020	[66]	BBBC, PSO	Spatial inverted pendulum
2020	[67]	ACO	Magnetic levitation
2019	[27]	PSOs (2), GA	Central position control (electro-hydraulics)
2019	[68]	CHASO, ASO, GWO, IWO, SFS	Motor velocity control
2019	[69]	PSO	Power allocation
2019	[70]	WOA, ABC, PSO, DE, others	Voltage regulator
2019	[71]	ISCA	Load frequency control (power system)
2019	[72]	GWO, PSO, GA	Load frequency control
2019	[73]	GA, ES, DE, CS	Trajectory control (quadrotor)
2019	[74]	FFA, PSO, GA	Power thermal system
2019	[75]	ALPSOs (2), HKA, DE, ABC	Transfer functions (3)
2018	[76]	MOSTA, NSGAII, SPEA2, PESA2	Goethite process
2018	[77]	ACONM, PSO ABC, DE, GA	Voltage regulator (AVR)
2018	[78]	ACO, DE	Direct torque control with space vector modulation (DTC-SVM)
2018	[79]	DE, PSO	Inverted Pendulum
2018	[80]	MODE-II	Gasifier control (pressure, temperature, bed mass, calorific)
2018	[81]	DEs (2)	Trajectory tracking (four bar linkage)
2018	[82]	PSO	Position, speed control (aircraft refuel)
2017	[83]	ABC	Power flow (photo voltaic battery system)
2017	[84]	GWO, PSO, GA	Servo motor position control
2017	[85]	DEs (2)	Robot position control (SCARA)
2017	[29]	PSOs (2), GA	Transfer functions (electric drive, voltage regulator)
2016	[86]	NSGA-II	Active magnetic bearings (AMBs)
2016	[87]	DE mutations (8)	Parallel robot
2016	[88]	BMA	Trajectory tracking (mobile robot)
2013	[5]	GA, PSO	Automatic voltage regulator
2013	[89]	ACO, GA	Pressure level (tank)
2011	[26]	PSO	Maglev system
2011	[90]	DEs(2), GA	Binary Wood–Berry distillation column
2010	[91]	GA, EP, PSO, ACO	Transfer functions
1998	[92]	DE	Transfer functions
1994	[21]	GA	pH neutralization
1992	[22]	GA	Transfer functions

Also, since nature-inspired heuristics are prone to stagnate in local optima regions of the search space, it has been often a desirable practice to evaluate the performance of heuristics over distinct control instances. A control instance implies the task of tracking a user-defined trajectory. Among the listed works in Table 1-2, [85] and [64] studied the largest number of control instances: whereas [85] evaluated two DE schemes in the context of eight control instances of robot position control (SCARA), [64] used a PSO-based heuristic in eight control instances of a canonical tank system. Other works evaluated six control instances in pressure level (tank) [89], temperature control (cooling coil) [34], smoke control in tunnel fires [42], bulk resumption (ore mining) [62], and power allocation [69].

Furthermore, natural and biological processes were used as referents to devise new heuristics for PID tuning. For instance, [88] mimicked the behaviour of microbial evolution for PID tuning in line-tracing robot context, [66] used a theory of the evolution of the universe to render the Bing Bang-Big Crunch (BBBC) algorithm for spatial inverted pendulum, [84] used the model of social hierarchy and hunting dynamics of grey wolves to render the Grey Wolf Optimization (GWO) for tuning a Takagi-Sugeno-Kang PI-Fuzzy control, [55] used fractional order fish migration for PID parameter optimization of transfer functions, [56] proposed a modified monkey-multiagent deep reinforcement learning for PID tuning in quadrotor position control, [68] used the model of force interactions in atoms through the Lennard-Jones potential to render the atom search optimization (ASO) for fractional-order PID tuning in motor velocity control.

Also, hybrids with Neural Networks and Fuzzy Logic for PID tuning have been proposed. For instance, [47] used a deep reinforcement learning (D3QN) for a robot driver system, [48] used artificial hydrocarbon network trained with backpropagation for a two-tank system, [93], [94] proposed the hybrid with neural networks for inverted pendulum, and [85] used a three-layer neural network optimized by DE. Reference [95] used Fuzzy-PID in formation control and Takagi-Sugeno Fuzzy inference, [44] studied the hybrid between Fuzzy-PID, wolf colony algorithm and cuckoo search for smart grid, [45] tackled the Fuzzy-PID control with online optimization by DE for the semi-active suspension system, [46] used the hybrid between Fuzzy Logic, PID control and PSO-based parameter optimization for pH control in water and fertilizer, [63] used a hybrid between a swarm learning process (SLP) and Q-learning for weight updating SLP through a deterministic rule, [64] used a single variable for online robustness for a PID-control of a canonical tank system.

D. SCOPES

The optimization schemes based on the difference of vectors by DE have received systematic and consistent attention in the community. Solutions sampled from referent vectors in the population/archive are often useful to render attractive directions of the basins of the search landscape [96].

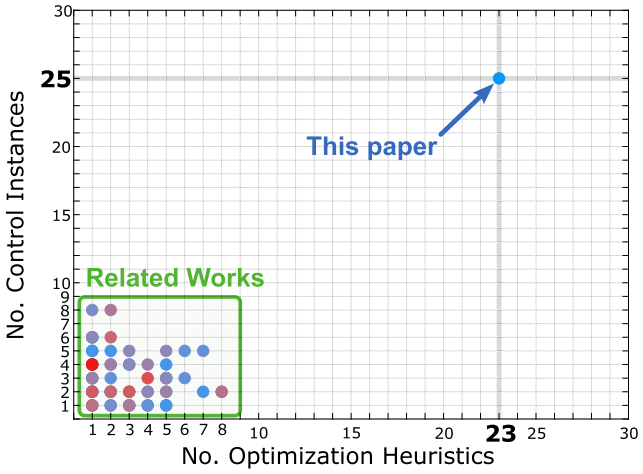


FIGURE 1. Number of studied heuristics and control instances in the recent related works. Spheres with blue (red) color denote works in recent (early) years.

This paper focuses in PID tuning/optimization based on DE for a class of control problems related to DC motor control, DC motor velocity control, magnetic levitation, inverted pendulum, and crane stabilization. Since the early works on control of transfer functions by a DE/RAND/1 mutation [92], several control instances and improved heuristics have been explored. For instance, [90] decoupled tackled the multivariable control in a context of distillation column model in which the scaling factor of a DE algorithm was normalized by Zaslavsky [97] mapping function. Reference [49] showed that DE algorithms attained the lowest positional error for PID-based visual control laws in robot manipulator. Reference [87] used dynamic constraints in DE within the context of control of a parallel robot, [37] used the strategy-adaptation DE (SADE) for temperature control of a parabolic trough, [45] used DE with Fuzzy Logic for PID tuning of a semi-active suspension system (quarter car), [35] used the many-objective DE for level temperature control of a rector model, [81] proposed the distance-based diversity enhancing mutation in DE for PID-based trajectory tracking of a four-bar linkage mechanism, [80] used the multi-objective differential evolution with spherical pruning for PID-based gasifier control (pressure, temperature, bed mass and calorific value).

E. MOTIVATION

A key aspect in nature-inspired optimization is to ascertain whether the class of optimization algorithms based on the difference of vectors generalize well to tackle a large class of PID-based control problems. Although the rigorous studies using synthetic mathematical functions have attracted the attention of the community [98], [99], [100], [101], [102], [103], [104], [105], [106], [107], the study of the convergence performance in relevant and diverse control problems/tasks

has remained elusive. The related works mentioned in Table 1 and Table 2 have so far explored the performance of a few algorithms over a selected number of control problems, most of which are in the form of transfer functions. To show the contemporary status of studies in PID tuning by nature-inspired optimization algorithms, Fig. 1 shows the number of studied optimization algorithms and the number of tackled control problem scenarios. For instance, [91] studied the performance of 4 algorithms (GA, EP, PSO, and ACO) for PID control tuning of 3 transfer functions; and since then, both the number of studied gradient-free optimization algorithms and control problem instances has remained relatively low. However, for generalization and practical purposes, it would be desirable to design algorithms that are able to tune PID controllers over a diverse set of control problems with minimal human intervention (self-adaptation features), and under tight computational budgets.

F. CONTRIBUTIONS

In this paper, aiming to fill the above-mentioned gap, we propose and investigate the effectiveness of a new class of algorithm based on the difference of vectors and self-adaptation mechanisms for PID tuning. Therefore, to investigate the above-mentioned line of inquiry, our focus/contribution is as follows:

- We tackle PID tuning problems by a new class of Differential Evolution with success-based Particle Adaptations (DEPA), which integrates principles of difference of vectors, particle schemes and adaptation mechanisms through archive (memory) means to enable the suitable adjustment of potential solution vectors and sampling parameters.
- We enlarge the scope of evaluation/study of the related/recent works, as shown by Fig. 1, through computational simulations over a large set of control problems (25 control instances considering the tracking of linear, nonlinear, continuous, and discontinuous trajectories in DC motor position control, DC motor velocity control, magnetic levitation, inverted pendulum, crane stabilization), compare the convergence performance with a large set of closely related optimization algorithms, and develop the extended adaptive variants of the closely related works (23 optimization algorithms were evaluated in total based on the below mentioned algorithms).
  - DERAND: DE/rand/1/bin Strategy [96],
  - DEBEST: DE/best/1/bin Strategy [96],
  - DESPS: DE with Speciation Strategy [104],
  - JADE: Adaptive DE with External Archive [108],
  - SHADE: Success-History based Adaptive DE [109],
  - RBDE: Rank-based Differential Evolution [110],
  - DEGL: DE with Local and Global Neighborhoods [111],
  - DESIM: DE with Similarity Based Mutation [112],
  - DCMAEA: Differential CMAE [113],
  - OBDE: Opposition Based DE [114],
  - PSO: Particle Swarm Optimization [115],
- Our results show (1) the competitive/outperforming convergence performance of the proposed algorithm under tight function evaluation budgets, and (2) the feasibility to attain reasonable control performances over 25 control instances considering the tracking of linear, nonlinear, continuous, and discontinuous trajectories in DC motor position control, DC motor velocity control, magnetic levitation, inverted pendulum, crane stabilization. Example real-world experiments on an inverted pendulum device show the potential for transferability of the learned gains to unseen environments.
- We evaluated the algorithmic extension/generalization towards further optimization domains in the CEC 2017 benchmark suite, which considers single-objective optimization problems with unimodality, multimodality, hybrid functions and compositions. Computational experiments show the attractive/outperforming performance overall problem instances.

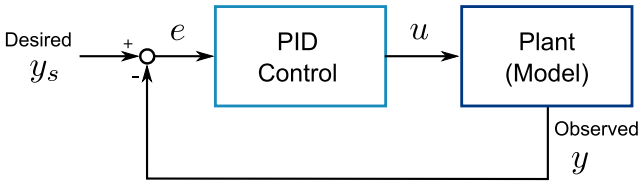


FIGURE 2. Basic concept of PID.

G. ORGANIZATION

The rest of the paper is organized as follows:

- section II briefly introduces PID and the corresponding fitness functions,
- section III introduces the new class of Differential Evolution with success-based Particle Adaptations (DEPA),
- section IV describes and discusses the computational simulations and the example experiments on hardware,
- section V describes the further extensions towards further optimization problems, and
- section VI concludes the paper.

II. PID CONTROL AND FITNESS FUNCTION

This section briefly describes the basic concept of PID control and the corresponding (fitness) performance functions.

PID control is often used to track a user-defined set point/trajectory profile of actuators, machines and feedback-based systems in general. Fig. 2 shows the basic idea of the key elements in a PID-based control system, in which the system to be controlled is modeled by the plant, the user-defined profile for the plant is encoded by y_s, observations from the plant are encoded by y. The overall goal is to render control policies u to render desirable performance in terms of the error term e.

A PID control law in continuous form is expressed by

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt}, \tag{1}$$

$$e(t) = y_s(t) - y(t), \tag{2}$$

where t denotes the simulation/execution time bounded by $[0, T_s]$ for upper bound on modeling/execution time T_s, $u(t)$ is the control signal provided to the plant, k_p, k_i, k_d are coefficients (gains) for the proportional, integral and derivative terms of (1), $y_s(t)$ is the set point value, $y(t)$ is the measurement from the plant, and $e(t)$ describes the error between the set point and the measured value. PID gains are often encoded by a vector

$$\boldsymbol{x} = [k_p \; k_i \; k_d]. \tag{3}$$

The (fitness) performance function of a PID-based control system is often measured with the following performance metrics

$$\text{IAE}(\boldsymbol{x}) = \int_0^{T_s} |e(t)| dt, \tag{4}$$

$$\text{ITAE}(\boldsymbol{x}) = \int_0^{T_s} t|e(t)| dt, \tag{5}$$

$$\text{ITSE}(\boldsymbol{x}) = \int_0^{T_s} te(t)^2 dt, \tag{6}$$

$$\text{ISE}(\boldsymbol{x}) = \int_0^{T_s} e(t)^2 dt, \tag{7}$$

IAE denotes the integral of the absolute of error, ITAE denotes the integral of the time multiplied absolute error, ITSE denotes the integral of the time multiplied squared error, and ISE denotes the integral of the squared error. In this paper, we use the above-mentioned as fitness functions.

III. DIFFERENTIAL EVOLUTION WITH SUCCESS-BASED PARTICLE ADAPTATIONS (DEPA)

Having described the basic concept of PID, this section introduces the key concepts behind the proposed new class of Differential Evolution with success-based Particle Adaptations (DEPA).

The idea of using difference of vectors [96] and particle schemes [115] to generate solutions by sampling through a population of individuals has triggered new forms of selection and self-adaptation in gradient-free and nature-inspired optimization algorithms. This paper unifies the principles of difference of vectors, particle schemes and adaptation mechanisms into a general and integrated form, and presents new sampling schemes for gradient-free population-based optimization. We extend the related literature by integrating adaptation mechanisms through memory schemes to enable the tracking of both successful/potential solution vectors and sampling parameters. In a nutshell, solution vectors $\boldsymbol{x}_i$ are generated by iterative sampling based on the following equations:

$$\boldsymbol{x}_{i,g+1} = \begin{cases} \boldsymbol{u}_{i,g}, & \text{if } f(\boldsymbol{u}_{i,g}) < f(\boldsymbol{x}_{i,g}) \\ \boldsymbol{x}_{i,g}, & \text{otherwise} \end{cases} \tag{8}$$

$$\boldsymbol{u}_{i,g} = \boldsymbol{x}^r_{i,g} + \boldsymbol{v}^*_{i,g} \tag{9}$$

$$\boldsymbol{v}^*_{i,g} = \omega \boldsymbol{v}_{i,g} + F_i(\boldsymbol{x}_{\text{pbest},g} - \boldsymbol{x}^r_{i,g})r_1 + G_i(\boldsymbol{x}_{\text{gbest},g} - \boldsymbol{x}^r_{i,g})r_2 \tag{10}$$

$$\boldsymbol{v}_{i,g+1} = \begin{cases} \boldsymbol{v}^*_{i,g}, & \text{if } f(\boldsymbol{u}_{i,g}) < f(\boldsymbol{x}_{i,g}) \\ \boldsymbol{v}_{i,g}, & \text{otherwise} \end{cases} \tag{11}$$

where $\boldsymbol{x}_{i,g}$ denotes the i-th individual/solution vector in the g-th generation/iteration; and $f(\boldsymbol{x}_{i,g})$ denotes the fitness function of the i-th individual/solution. Here, solutions are elements of a population set $\boldsymbol{x}_i \in \mathcal{P}$, $i = 1, 2, \ldots, N$, with $N = |\mathcal{P}|$. For ease of reference, and along with the general notations of DE [96] and PSO [115], $\boldsymbol{u}_{i,g}$ is a trial vector; $\boldsymbol{v}_{i,g}$ is a velocity vector associated to the i-th individual; $\boldsymbol{v}^*_{i,g}$ is a trial velocity vector; ω is a smoothing coefficient on the velocity vector; F_i, G_i are scaling factors associated with the i-th individual; $\boldsymbol{x}_{\text{pbest},g}$ is the best solution of the i-th individual up to generation g; $\boldsymbol{x}_{\text{gbest},g}$ is the best individual overall the population $\mathcal{P}$ up to generation g; $\boldsymbol{x}^r_{i,g}$ is the referential vector related to the i-th individual; and r_1, r_2 are random numbers with uniform distribution $U[0, 1]$. In the above formulation, individuals $\boldsymbol{x}_{i,g}$ from the population $\mathcal{P}$ and their respective velocities $\boldsymbol{v}_{i,g}$ are initialized randomly; and the reference vector $\boldsymbol{x}^r_{i,g}$ is sampled/chosen uniformly randomly either from the population $\mathcal{P}$ or from an archive (memory) $\mathcal{A}$, the latter of which serves the role of providing a diversification mechanism by storing potential solution vectors from the search space in case detrimental mutations occur [104], as follows:

$$\boldsymbol{x}^r_{i,g} \in_R \begin{cases} \mathcal{P}, \text{ if } q_i < Q \\ \mathcal{A}, \text{ otherwise} \end{cases} \tag{12}$$

where q_i is a counter of unsuccessful trials and Q is a user-defined threshold. The selection mechanism in (8) is successful whenever the trial solution vector $\boldsymbol{u}_{i,g}$ is better than the current solution $\boldsymbol{x}_{i,g}$, thus $f(\boldsymbol{u}_{i,g}) < f(\boldsymbol{x}_{i,g})$. As such, after initializing the archive $\mathcal{A}$ with the population $\mathcal{P}$, it is updated with subsequent successful trial vectors $\boldsymbol{u}_{i,g}$, as follows

$$\mathcal{A}_g = \begin{cases} \mathcal{P}, \text{ if } g = 0 \\ \mathcal{A}_g \cup \{\boldsymbol{u}_{i,g}\}, \text{ if } g > 0 \text{ and } f(\boldsymbol{u}_{i,g}) < f(\boldsymbol{x}_{i,g}) \end{cases} \tag{13}$$

in the above formulation, the size of $\mathcal{A}$ is set to a factor of the population size to prevent the uncontrolled growth of the archive (memory) $\mathcal{A}$, that is $|\mathcal{A}| = \lambda|\mathcal{P}|$ for constant λ, and newer elements joining $\mathcal{A}$ replace older ones to allow tracking attractive trial vectors in recent/later generations.

Inspired by adaptive parameter adaptation schemes [116], each individual $\boldsymbol{x}_{i,g}$ at generation/iteration g is associated with scaling factors F_i and G_i, each of which is sampled from the Cauchy distributions

$$F_i \sim \text{Cauchy}(\mathcal{M}^{a_i}_F, \sigma^2_F) \tag{14}$$

$$G_i \sim \text{Cauchy}(\mathcal{M}^{b_i}_G, \sigma^2_G) \tag{15}$$

where Cauchy(μ, σ^2) denotes the values sampled from a Cauchy distribution with location parameter μ and scale parameter σ^2 [108], the parameter $\mathcal{M}_F^{a_i}$ denotes the a_i-th element of an h-dimensional tuple (memory) $\mathcal{M}_F$, and the positive integers $a_i,\ b_i$ is sampled uniformly from the range $[1, h]$, $a_i \sim U[1, h],\ b_i \sim U[1, h]$, $h = N$. When the values of F_i and G_i are negative, they are regenerated until positive values are obtained.

The elements of the archives $\mathcal{M}_F,\ \mathcal{M}_G$ are updated by using the weighted Lehmer mean that considers not only the potential/useful parameters that led to successful mutations, but also the contribution to fitness improvements [109]. After a successful selection in (8), when $f(\boldsymbol{u}_{i,g}) < f(\boldsymbol{x}_{i,g})$ in (8), the coefficients F_i and G_i associated to the i-th individual are recorded into corresponding (archive) sets S_F and S_G, with $|S_F| = |S_G| = n$ denoting the number of fitness improvements through successful mutations/selections; and at the end of each generation/iteration, the mean values $\mathcal{M}_F, \mathcal{M}_G$ are updated with the weighted Lehmer mean as follows

$$\mathcal{M}_F^k = \frac{\sum_{j=1}^{n} w_j \cdot S_{F,j}^2}{\sum_{j=1}^{n} w_j \cdot S_{F,j}},\ k \in [1, h] \tag{16}$$

$$\mathcal{M}_G^k = \frac{\sum_{j=1}^{n} w_j \cdot S_{G,j}^2}{\sum_{j=1}^{n} w_j \cdot S_{G,j}},\ k \in [1, h] \tag{17}$$

$$w_j = \frac{\triangle f_j}{\sum_{s=1}^{n} \triangle f_s} \tag{18}$$

$$\triangle f_j = \left| f(\boldsymbol{u}_{i,g}) - f(\boldsymbol{x}_{i,g}) \right| \tag{19}$$

where w_j is the weight associated to a *successful* trial vector $\boldsymbol{u}_{i,g}$; $\triangle f_j$ denotes the j-th improvement of the fitness function by trial vector $\boldsymbol{u}_{i,g}$ from respect to current solution $\boldsymbol{x}_{i,g}$; $\mathcal{M}_F^j,\ \mathcal{M}_G^j$ denote the weighted mean of scaling parameters F_i, G_i respectively. The key motivation of using two different archives $\mathcal{M}_F^j,\ \mathcal{M}_G^j$ is to track/log distinct scaling parameters that lead to successful mutations through best referent solution vectors $\boldsymbol{x}_{\text{pbest},g}$ and $\boldsymbol{x}_{\text{gbest},g}$, respectively. The above update mechanisms imply that the scaling parameters F_i, G_i are updated considering concomitant normalized improvements during the search procedure. The pseudocode of the complete DEPA algorithm is outlined in Algorithm 1.

In the flow of the overall algorithm, in every generation, (1) we compute scaling parameters F_i and G_i associated to each individual; (2) generate trial vectors accordingly; (3) perform mutation and selection mechanisms and, whenever successful mutations occur, we store potential reference vectors and scaling parameters into their corresponding archive (memory) mechanisms; and (4) update the memory archives considering useful parameters that led to successful fitness improvements. The procedure is repeated until the maximum number of fitness evaluations ($MaxFEs$) is reached. The complexity is estimated as $O(MaxFEs.(n + N(f + D)))$, in which $O(f)$ is the complexity of evaluating the fitness function and D is the dimensionality of solution vector $\boldsymbol{x}$. DEPA unifies the principles of difference of vectors, stochastic particle schemes and parameter adaptation mechanisms into an integrated formulation where solutions are rendered through potential referent vectors, and where memory-based adaptation enables to track useful parameters that lead to fitness improvements. Compared to related works in archive-based self-adaptive DE [104], [108], [109], DEPA extends the sampling mechanism of new solutions by using the difference of best vectors and by considering the archive of trial vectors

Algorithm 1 DEPA

$FEs = 0,\ g = 0,\ q_i = 0,\ k = 0$;
Generate a set of N individuals randomly as initial population set $\mathcal{P}$;
Initialize the archive $\mathcal{A}$ from the population set $\mathcal{P}$;
Initialize the tuples $\mathcal{M}_F^k$ and $\mathcal{M}_G^k$;
$FEs = FEs + N$;
while $FEs \leq MaxFEs$ **do**
  $g = g + 1$;
  $S_F = \{\},\ S_G = \{\}$;
  for $i = 1$ **to** N **do**
    Sample $a_i \sim U[1, h],\ b_i \sim U[1, h]$;
    Generate F_i and G_i using (14)-(15);
  end
  Find out the best individual $\boldsymbol{x}_{\text{gbest},g}$ overall $\mathcal{P}$;
  for $i = 1$ **to** N **do**
    Update the i-th best individual $\boldsymbol{x}_{\text{pbest},g}$;
    Generate the trial vector $\boldsymbol{u}_{i,g}$ using (9)-(12);
    if $f(\boldsymbol{u}_{i,g}) < f(\boldsymbol{x}_{i,g})$ **then**
      $\boldsymbol{x}_{i,g+1} = \boldsymbol{u}_{i,g},\ \boldsymbol{v}_{i,g+1} = \boldsymbol{v}_{i,g}^*$;
      $\boldsymbol{u}_{i,g} \rightarrow \mathcal{A}$;
      Delete an element from $\mathcal{A}$ if $|\mathcal{A}| > \lambda|\mathcal{P}|$;
      $F_i \rightarrow S_F,\ G_i \rightarrow S_G$;
      $q_i = 0$;
    else
      $\boldsymbol{x}_{i,g+1} = \boldsymbol{x}_{i,g}$;
      $\boldsymbol{v}_{i,g+1} = \boldsymbol{v}_{i,g}$;
      $q_i = q_i + 1$;
    end
    $FEs = FEs + 1$;
  end
  if $S_F \neq \{\} \wedge S_G \neq \{\}$ **then**
    $k = k + 1$;
    Update $\mathcal{M}_F^k$ and $\mathcal{M}_G^k$ using (16) and (17);
    Set $k = 1$ if $k > h$;
  end
end

that led to successful mutations through Eq. (8) - Eq. (13). Also, DEPA uses two different archives $\mathcal{M}_F^j$, $\mathcal{M}_G^j$ to track and adapt distinct scaling parameters that render successful trial vectors through best referent solution vectors $\boldsymbol{x}_{\text{pbest},g}$ and $\boldsymbol{x}_{\text{gbest},g}$, respectively. Furthermore, compared to successful parent selection schemes [57], [58], [79], [104], DEPA extends the adaptation mechanisms through memory schemes to enable the tracking of successful/potential solution vectors $\boldsymbol{u}_{i,g}$ as well as velocities $\boldsymbol{v}_{i,g}^*$ and scaling parameters F_i, G_i through Eq. (11) and Eq. (14) - Eq. (19). In the next section, we rigorously compare the performance of the proposed DEPA algorithm to the related works in the context of PID parameter tuning.

IV. COMPUTATIONAL SIMULATIONS AND EXPERIMENTS ON HARDWARE

Having described the proposed new class of Differential Evolution with success-based Particle Adaptations (DEPA), this section describes the performance evaluations and benchmarks of DEPA using computational and experimental studies in a relevant set of control systems and tasks.

Thus, to evaluate the performance of the algorithm in finding feasible PID gains, we conducted computational simulations to evaluate the convergence ability of the algorithms, as well as the control performance. In this section we first describe the control problems tackled in this paper, and then describe our obtained results rendered from computational experiments, and finally present an example of the performance on hardware to show the control performance on unseen environments during training.

A. MOTOR POSITION CONTROL

The control of the angle of rotation of a DC motor considers the RLC circuit model as shown in Fig. 3, as follows

$$L_a \frac{d}{dt} i(t) + R_a i(t) + \frac{1}{C} \int i(t) dt = V(t), \tag{20}$$

where

- $V(t)$ is the input voltage,
- $v_a(t)$ is the voltage at both ends in the capacitor,
- R_a is the effective resistance of the combined load, source and components,
- L_a is the motor inductance (inductor component), and
- C is the capacitance coefficient.

Considering linear relations

$$T(t) = k_T i(t), \tag{21}$$

$$v_a(t) = k_E \frac{d\theta}{dt}, \tag{22}$$

- θ is the angle of rotation of the motor,
- T is the motor torque,
- k_T is a torque constant,
- k_E is the electromotive force constant.

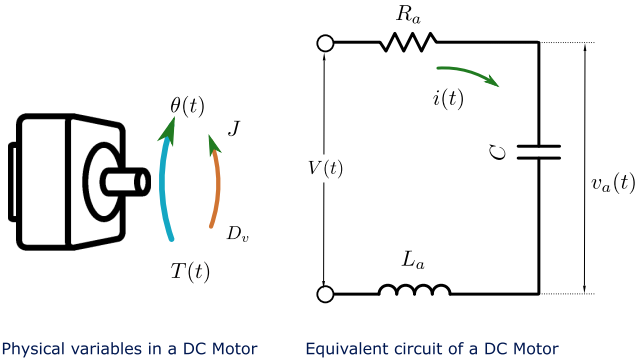


FIGURE 3. DC motor model.

The governing equations of the DC motor system can be obtained from Newton's Law and by combining (20)-(22):

$$\frac{L_a J}{k_T} \frac{d^3\theta}{dt} + \left(\frac{L_a D_v + R_a J}{k_T}\right) \frac{d^2\theta}{dt} + \left(\frac{R_a D_v}{k_T} + k_E\right) \frac{d\theta}{dt} = V(t), \tag{23}$$

- J is the inertia moment, and
- D_v is the mechanical friction (viscous coefficient).

For initial angular position θ_0, the motor position control aims at tracking a user-defined profile $\theta_s(t)$. As such, the tracking error is defined by

$$e(t) = \theta_s(t) - \theta(t), \tag{24}$$

were the gains of a PID control framework considers

$$\boldsymbol{x}_{mp} = [k_p^{mp}\ k_i^{mp}\ k_d^{mp}], \tag{25}$$

B. MOTOR VELOCITY CONTROL

It is possible to obtain the governing equations for motor velocity from (23) as follows

$$\frac{L_a J}{k_T} \frac{d^2\omega}{dt} + \left(\frac{L_a D_v + R_a J}{k_T}\right) \frac{d\omega}{dt} + \left(\frac{R_a D_v}{k_T} + k_E\right) \omega = V(t) \tag{26}$$

where $\omega = \frac{d\theta}{dt}$ denotes the angular velocity of the motor.

Thus, for initial angular velocity configuration in θ_0 and ω_0, the velocity control aims at tracking a user-defined speed profile $\omega_s(t)$. Similar to position control, the tracking error is defined by

$$e(t) = \omega_s(t) - \omega(t), \tag{27}$$

where the PID gains encoding the proportional, integral and derivative gains are

$$\boldsymbol{x}_{mv} = [k_p^{mv}\ k_i^{mv}\ k_d^{mv}], \tag{28}$$

C. MAGNETIC LEVITATION

Being used in magnet-based trains, fans of desktop computers, bearings, pumps and turbines, the magnetic levitation systems are appealing for vibration damping. Generally speaking, a magnetic levitation system consists of a coil

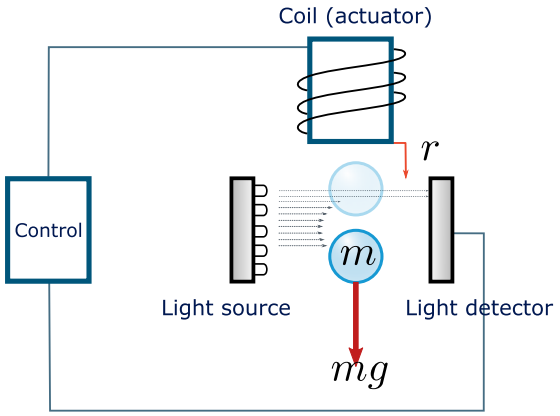


FIGURE 4. Magnetic levitation model.

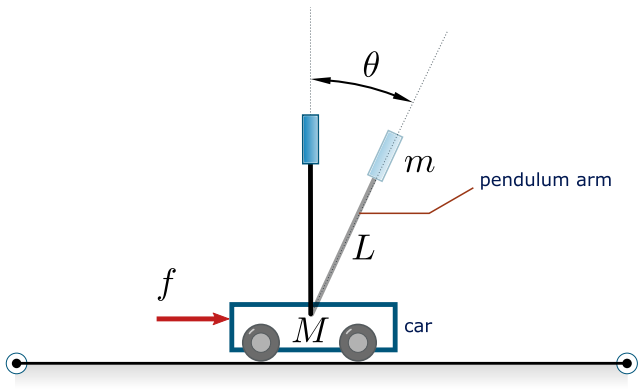


FIGURE 5. Inverted pendulum model.

mounted on an actuator applying electrical current to a metal-based levitating component (e.g. sphere). As such, considering the downward side as positive displacement as shown by Fig. 4, the governing equation of a levitating component is

$$m\ddot{r} = mg - k\frac{i^2}{r^2}, \tag{29}$$

- m is the mass of the levitating component,
- k is an electromagnet parameter (coil-dependent),
- i is the inductor current,
- r is the vertical displacement of the levitating component (spacing between the electromagnet and the levitating component).

The above ignores non-ideal factors such as hysteresis, saturation of the core, and eddy currents. Since the current i is proportional to the control voltage u, it is reasonable to define the following relationship

$$i = k_1 u, \tag{30}$$

and simplifying terms considering negligible perturbations around the equilibrium, the dynamics become

$$\ddot{r} = \frac{-2gi}{i_0} + \frac{2gx}{r_0}, \tag{31}$$

where i_0 and r_0 denote the equilibrium state and g is the gravity constant. The control framework considers moving the levitating component to track a user-defined displacement trajectory $r_s(t)$; thus the tracking error is defined by

$$e(t) = r_s(t) - r(t), \tag{32}$$

The search space for parameter tuning is defined by a 3-dimensional tuple encoding proportional-integral-derivative gains of the gap spacing

$$\boldsymbol{x}_{ml} = [k_p^{ml}\ k_i^{ml}\ k_d^{ml}], \tag{33}$$

D. INVERTED PENDULUM

Used in applications requiring vertical stabilization such as two-wheeled personal transporters (segway), building structures, and aircraft landings, the stabilization of a self-lifting arm attached to a car (cart) as shown by Fig. 5 is a nonlinear control problem whose governing equations are often expressed by

$$(m+M)\ddot{r} + b\dot{r} + mL\ddot{\theta}\cos\theta - mL\dot{\theta}^2\sin\theta = f, \tag{34}$$

$$(I+mL^2)\ddot{\theta} - mgL\sin\theta + mL\ddot{r}\cos\theta + d\dot{\theta} = 0, \tag{35}$$

- m is the pole mass,
- M is the car (cart) mass,
- L is the length of the arm,
- g is the gravity constant,
- I is the moment of inertia of the arm,
- b is the friction coefficient of the car,
- d is the damping coefficient of the pendulum,
- f is the dragging force,
- r is the car position,
- θ is the inclination angle of the pendulum arm.

For initial configuration (orientation) of the pendulum arm at θ_0 and the position of the car at $r_0 = 0$, the inverted pendulum stabilizes the arm at $\theta = 0$ (the vertical position) and car position at $r = 0$ by controlling the position of the car and the arm (by moving the car in the horizontal direction and rotating the arm). As such, the tracking error is defined by

$$e_c(t) = \underbrace{r_s(t) - r(t)}_{\text{car}} \tag{36}$$

$$e_a(t) = \underbrace{\theta_s(t) - \theta(t)}_{\text{arm}}. \tag{37}$$

where e_c denotes the error term in the car position, and e_a denotes the error term in the arm angular position; the nature of the control instance of the inverted pendulum requires that $r_s = 0$m and $\theta_s = 0$rad. The control framework using PID considers combining the control signals from the car and the arm as follows $u(t) = u^c(t) + u^a(t)$ where the script u^c (u^a) denotes the control signal for the car (arm). The gains are

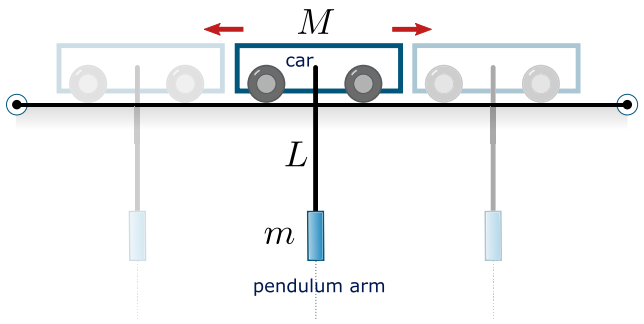


FIGURE 6. Crane stabilization model.

encoded by

$$\boldsymbol{x}_c = [k_p^c \; k_i^c \; k_d^c], \tag{38}$$

$$\boldsymbol{x}_a = [k_p^a \; k_i^a \; k_d^a] \tag{39}$$

E. CRANE STABILIZATION

Being used to stabilize moving loads in construction, logistics and ship building, the crane stabilization problem is appealing for ensuring safety in transporting loads attached to a cable or arm. As such, the stabilization of the crane considers the control of the car of the inverted pendulum to track a user-defined profile $r_s(t)$ while setting the arm at the fixed position $\theta = \pi$, as shown by Fig. 6. The governing equations of the crane stabilization problem are defined by Eq. (34) - (35).

For initial configuration at $\theta_0 = \pi$rad and $r_0 = 0$, the tracking error is defined by the errors from the car and the arm following (36) and (37), with $\theta_s = \pi$rad (vertical position, pointing downwards). The control framework using PID considers combining the control signals from the car and the arm as follows

$$u(t) = u^c(t) - u^a(t), \tag{40}$$

where u^c (u^a) denotes the control signal for the cart (arm). Aligned with the inverted pendulum problem, a PID-control for the crane stabilization problem encodes the gains with (38) - (39).

F. SIMULATION IMPLEMENTATIONS

Having described the models of target systems to be controlled, this subsection describes the key computational considerations and parameters to simulate a relevant set of diverse control profiles/tasks.

The corresponding governing equations of modeled and controlled systems were evaluated and numerically solved through Matlab 2021b by using Simulink instances which correspond to simulation interfaces of existing real-world systems [117], [118], [119]. The parameter configuration for all the modeled/controlled systems are described by Table 3. The motivation behind using the parameter set in Table 3 is due to the reasonable alignment with existing real-world units of servo motors, magnetic levitation, inverted pendulum and crane stabilization [117], [118], [119]. Through several simulation instances, the following solvers rendered feasible solutions efficiently: the fifth order Dormand-Prince formula (ode5) for magnetic levitation, inverted pendulum, and crane model stabilization, and a quasi-constant step size solver with numerical Jacobian estimations (ode15s) for motor position and velocity control. The modeling/simulation time T_s for each controlled system is also shown in Table 3, with the simulation time step $dt = 0.01$s. Also, for each type of system, we used a set of control profiles that correspond to tasks inspired by real-world phenomena. As such, we modeled each control profile by mathematical functions aiming to portray distinct behaviors of the system that consider linear, nonlinear, continuous, and discontinuous transitions, as shown by Table 3.

- For DC motor position (velocity) control problems [117], the set variable is the angular rotation θ_s (angular velocity ω_s), and five instances model the behavior of θ_s (ω_s) as a function of time as rendered in Table 3.
- For magnetic levitation [118], the goal is to move the position r of the ball to reach the nonlinear control profiles given by instance 1 to instance 5.
- For inverted pendulum [119], the goal is to move the car in the horizontal direction to reach the angular position of the pendulum arm at $\theta = 0$rad (vertical position of the arm, pointing upwards) and car position at $r = 0$, given initial conditions on the configuration of the pendulum arm at $\theta_0 = \{0.05,\ 0.2,\ 0.35,\ 0.5,\ 0.65\}$rad.
- For crane stabilization [119], the goal is to move the position r of the car to reach the nonlinear control profiles through instances 1 to instances 5 while keeping the angular of the pendulum arm fixed at $\theta_s = \pi$.

Each model instance considers five control profiles (tasks), thus the total number of tackled control instance problems is 25 (= 5 systems × 5 instances). Furthermore, the performance/fitness function considers the key control performance metrics such as IAE, ITAE, ITSE, ISE defined by (4)-(7), each of which is computed by trapezoidal numerical integration. Models of inverted pendulum and crane stabilization use a two-error term defined by (36) and (37), as such, the performance/fitness composes two terms. For instance, to compute IAE metric for inverted pendulum and crane stabilization, we compute, and numerically estimate by trapezoidal integration the composition of the following:

$$\text{IAE}(\boldsymbol{x}_c, \boldsymbol{x}_a) = \text{IAE}(\boldsymbol{x}_c) + \text{IAE}(\boldsymbol{x}_a).$$

And to compute ITAE and ITSE metrics for inverted pendulum and crane stabilization, we compose the following:

$$\text{ITAE}(\boldsymbol{x}_c, \boldsymbol{x}_a) = \text{ITAE}(\boldsymbol{x}_c) + \text{ITAE}(\boldsymbol{x}_a).$$
$$\text{ITSE}(\boldsymbol{x}_c, \boldsymbol{x}_a) = \text{ITSE}(\boldsymbol{x}_c) + \text{ITSE}(\boldsymbol{x}_a).$$

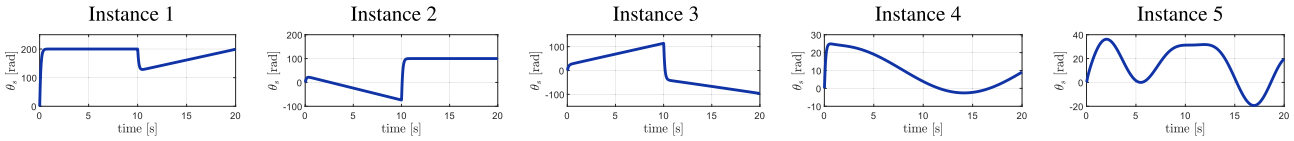
The composition of other performance/fitness metrics such as the ones defined in (4)-(7) would follow a similar procedure.

TABLE 3. System parameters and control profiles.

1. DC Motor Position Control

Moment of inertia $J = 1.4 \cdot 10^{-5}\text{kg}\,\text{m}^2$, torque constant $k_T = 0.52\text{Nm/A}$, the electromotive force constant $k_E = 0.057\text{Vs/rad}$, viscous coefficient $D_v = 10^{-6}\text{Nms/rad}$, electrical circuit resistance $R_a = 2.5\Omega$, electrical inductance $L_a = 0.0025\text{mH}$, PID gains $k_p^{mp}, k_i^{mp}, k_d^{mp} \in [0, 0.05]$, modeling time $T_s = 20\text{s}$.

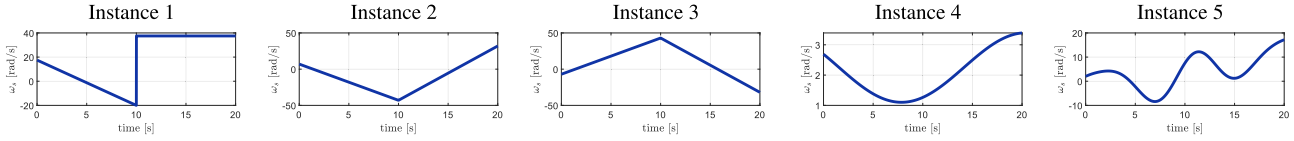
Profiles for DC Motor Position θ_s



2. DC Motor Velocity Control

Moment of inertia $J = 10^{-5}\text{kg}\,\text{m}^2$, torque constant $k_T = 0.52\text{Nm/A}$, the electromotive force constant $k_E = 0.057\text{Vs/rad}$, viscous coefficient $D_v = 10^{-6}\text{Nms/rad}$, electrical circuit resistance $R_a = 2.5\Omega$, electrical inductance $L_a = 0.0025\text{mH}$, PID gains $k_p^{mv}, k_i^{mv}, k_d^{mv} \in [0, 0.05]$, modeling time $T_s = 20\text{s}$.

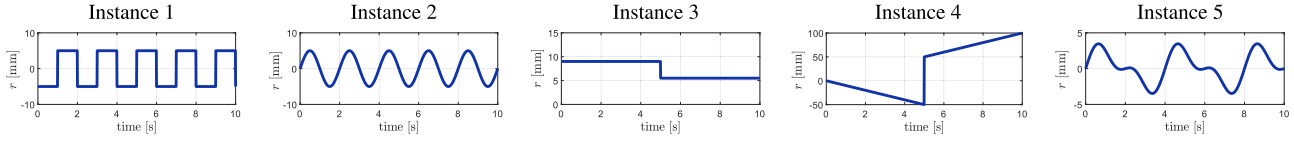
Profiles for DC Motor Velocity ω_s



3. Magnetic Levitation Model

Equilibrium of current $i_0 = 0.8\text{A}$, equilibrium of position $r_0 = 9\text{mm}$, PID gains $k_p^{ml} \in [0,5], k_i^{ml} \in [0,5], k_d^{ml} \in [0, 0.15]$, modeling time $T_s = 10\text{s}$.

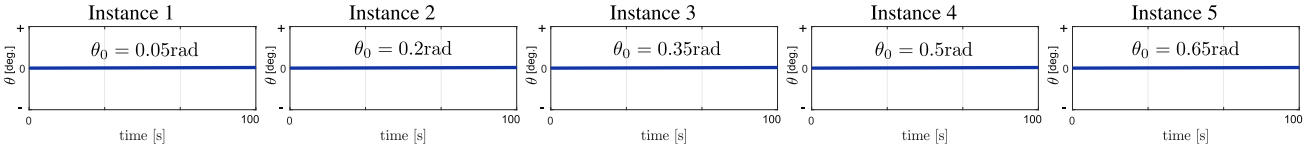
Profiles for Control r_s



4. Inverted Pendulum Model

Pendulum mass $m = 0.23\text{kg}$, cart mas $M = 2.4\text{kg}$, (pendulum) arm length $L = 0.4\text{m}$, moment of inertia of the (pole) arm $I = 0.099\text{kg}\,\text{m}^2$, cart friction coefficient $b = 0.05\text{Ns/m}$, pendulum damping coefficient $d = 0.005\text{Nms/rad}$, PID gains $k_p^c \in [1,20]$, $k_i^c \in [0,15]$, $k_d^c \in [0,15]$, $k_p^p \in [0,50]$, $k_i^p \in [0,50]$, $k_d^p \in [0,50]$, modeling time $T_s = 30\text{s}$.

Profiles for Control θ_s



All control instances imply reaching the set angle $\theta = 0\text{rad}$, given initial conditions on the configuration of the pendulum arm for $\theta_0 = 0.05\text{rad}$ (Instance 1), $\theta_0 = 0.2\text{rad}$ (Instance 2), $\theta_0 = 0.35\text{rad}$ (Instance 3), $\theta_0 = 0.5\text{rad}$ (Instance 4), $\theta_0 = 0.65\text{rad}$ (Instance 5).

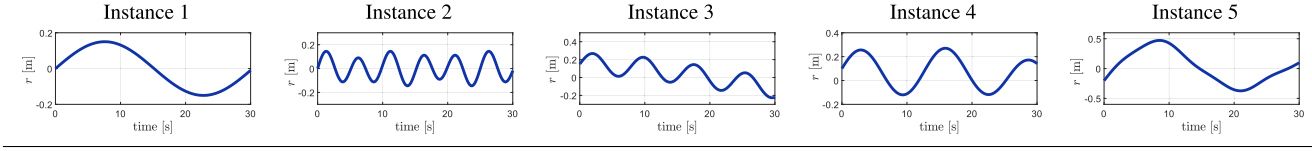
5. Crane Stabilization Model

DC Motor: Moment of inertia $J = 1.4 \cdot 10^{-5}\text{kg}\,\text{m}^2$, torque constant $k_T = 0.05\text{Nm/A}$, the electromotive force constant $k_E = 0.05\text{Vs/rad}$, viscous coefficient $D = 10^{-6}\text{Nms/rad}$, electrical circuit resistance $R_a = 2.5\Omega$, electrical inductance $L_a = 0.0025\text{mH}$.

Pendulum mass $m = 0.2\text{kg}$, cart mas $M = 2.3\text{kg}$, (pendulum) arm length $L = 0.3\text{m}$, moment of inertia of the (pole) arm $I = 0.099\text{kg}\,\text{m}^2$, cart friction coefficient $b = 5 \cdot 10^{-5}\text{Ns/m}$, pendulum damping coefficient $d = 0.005\text{Nms/rad}$, initial pendulum angle $\theta_0 = \pi$.

Modeling time $T_s = 30\text{s}$, PID gains $k_p^c \in [1,20]$, $k_i^c \in [0,15]$, $k_d^c \in [0,15]$, $k_p^p \in [0,50]$, $k_i^p \in [0,50]$, $k_d^p \in [0,50]$.

Profiles for Control r_s



G. BENCHMARK HEURISTICS

Having described the key considerations and parameters to simulate control profiles/tasks, this subsection describes the set of DE-based benchmark heuristics used in evaluations and overall comparisons.

To explore the performance frontiers of gradient-free metaheuristics and evaluate the ability to tackle computationally expensive PID-based control problems through population-based approaches being inherently native to parallelization, we used the following heuristics:

- DEPA and DEP: the proposed algorithm with and without parameter adaptation. Whereas DEPA (section III) implements the adaptation mechanisms (14)-(19), DEP forgoes such adaptation mechanisms and uses fixed parameters F, G for all individuals in the population.
- DERAND and DERAND_A: DE with DE/rand/1/bin mutation strategy [96], this algorithm is considered highly exploratory, and implements the random-based difference of vectors.
- DEBEST and DEBEST_A: DE/best/1/bin strategy [96]; due to using a best reference solution vector, this algorithm is regarded as having a high degree of exploitation strategy.
- DESPS and DESPS_A: DE with successful parent selection strategy [104]; due to using archive-based adaptation of reference vectors, this algorithm is known to avoid stagnation in local optima.
- SHADE: Success-History based adaptive Differential Evolution [109]; this algorithm uses a success-based history adaptation of scaling parameters based on weighted Lehmer mean.
- PSO and PSO_A: Particle Swarm Optimization [115]; this algorithm is used as a closely related work due to the use of particle schemes.
- RBDE and RBDE_A: Rank-based Differential Evolution [110]; this algorithm uses a adaptive selection pressure using the Whitley distribution.
- JADE and JADE_A: Adaptive Differential Evolution with External Archive [108]; this algorithm uses an archive-based success-based history adaptation of scaling parameters based on weighted arithmetic mean.
- DEGL and DEGL_A: DE with Local and Global Neighborhoods [111]; this algorithm uses adaptation mechanism using best local and global reference vectors, this algorithm is used as referent due to the use of global/local best.
- DESIM and DESIM_A: Differential Evolution with Similarity Based Mutation [112], this algorithm uses adaptation based on similarity to the best individual in the population.
- DCMAEA and DCMAEA_A: Differential Covariance Matrix Adaptation Evolutionary Algorithm [113], this algorithm implements/hybridizes the Covariance Matrix operators into the DE formulation, showing a balanced adaptation of explorative and exploitative behaviors.
- OBDE and OBDE_A: Opposition Based Differential Evolution [114], this algorithm implements the opposition operator in initialization/mutations, having shown the attractive performance compared to an adaptive DE.

In the above, the " A" subscript implies that the heuristic implements the adaptation mechanism based on the weighted composition of potential parameters that lead to successful mutations/crossover as well as fitness improvements [109]. The key motivation of the above is to enable fair comparisons across adaptive mechanisms in PID tuning. DEPA extends the above-mentioned adaptation principle through (14)-(19), whereas heuristic SHADE inherently implements such adaptation scheme, yet using a distinct notion of difference of vectors. As such, for fair comparisons, we implemented the adaptation of both scaling parameters F_i and crossover rate CR_i for each individual across all the above-mentioned DE-based algorithms [96], [104], [108], [110], [111], [112], [113], [114]. It is noteworthy to mention that JADE_A implements the weighted Lehmer mean that considers the contributions in fitness improvements, instead of the original weighted arithmetic mean. Also, since PSO does not use crossover rate CR [115], PSO_A extends the adaptation of scaling parameters over the personal best (pbest) and the global best (gbest), akin to the behaviour of F_i and G_i in (10). By counting both the original and the extended adaptive versions of each of the above mentioned algorithm, we evaluated the performance of 23 optimization algorithms in total, including the proposed algorithms DEPA and DEP.

Furthermore, our motivation in using the above-described algorithms is to evaluate distinct forms of initialization, selection pressure, parameter adaptation, archive history, similarity-based sampling, and the performance frontiers of different forms of adaptation of exploration-exploitation mechanisms based on the difference of vectors for PID controller tuning. In addition to the class of DE variants, we included PSO as a benchmark due to the fact of using the concept of particle scheme. Although it is possible to use local-based search methods and nonlinear constrained optimization approaches such as Sequential Quadratic Programming, their performance is constrained by the ability to identify an initial guess, the differentiability of the objective function, the exploitative nature of sampling the search space, and the computational cost involved in computing gradients when expensive simulations are involved. The evaluation of gradient-based and local heuristics is out of the scope of this paper, whose study and potential hybridization to the above-mentioned metaheuristics is left for future work in our agenda.

H. HEURISTICS WITH FIXED PARAMETERS

Having described the overall set of DE-based benchmark heuristics for overall comparisons in subsection IV-G, this subsection describes the convergence and control performance of the subset of optimization algorithms forgoing the parameter adaptation mechanisms, that is the

original/unmodified versions without the " A" subscript as described in section IV-G.

Therefore, in the first set of experiments, we studied the convergence performance using different population sizes and a fixed set of scaling and crossover parameters in DE and PSO heuristics. As such, in this section we used DEP, which is the heuristic forgoing the parameter adaptation mechanisms presented in (14)-(19), as described in section IV-G. For benchmark comparisons, we used the original algorithms described in section IV-G, that is the original/unmodified versions without the " A" subscript as described in section IV-G. Thus, for parameter configuration, we used the probability of crossover $CR = 0.5$, scaling factor $F = 0.7$, population sizes $N = \{10, 50, 100\}$ for all DE-based algorithms, which are used in related literature [104]. The bias term in RBDE $\rho = 2$, $\omega = 0.5$, $F_i = c_1 = 0.5$ weight on pbest, and $G_i = c_2 = 2$ as weight on gbest in PSO and DEP, respectively, and the termination criterion is set to $MaxFEs = 5000$ function evaluations. In this section, we used ISE as the fitness/performance function for all algorithms and control instances. Also, due to the stochastic nature of the above-mentioned metaheuristics, 30 independent runs were evaluated for each algorithm and each configuration. Other parameters followed the suggested values of the above-mentioned references. Considering that each algorithm used three types of population size, the total number of algorithm configurations is 36, which is the result of 12 algorithms $\times$ 3 types of population. The key motivations for using the above parameters are as follows:

- Crossover probability with $CR = 0.5$ and $\omega = 0.5$ implement the equal importance and consideration to historical search directions up to the current number of iterations t.
- Weights $c_1 = 0.5$ and $c_2 = 2$ enable the higher preference for global best compared to the local best while sampling the search space; thus PSO and DEP algorithms consider the higher selective pressure towards the global best.
- Small population size $N = 10$ and number of evaluations up to 5000 allow evaluating the (frontier) performance of the gradient-free algorithms to find feasible gains under relatively tight computational budgets.
- Relatively large population, $N = 100$, allows to evaluate the ability of generating close to optimal solutions at the early stage (due to the stochastic nature of random initialization).
- Using distinct population sizes, as $N = \{10, 50, 100\}$, allows to evaluate the performance and feasibility of quick convergence of each mode of sampling and selection pressure in the above-mentioned algorithms.

To show the convergence ability of the studied algorithms, Table 4 - Table 8 show the overall convergence performance of all studied algorithms under distinct population sizes, evaluated on each control instance. In these figures, the x-axis denotes the number of function evaluations, and the y-axis denotes the value of the fitness/performance metric. The convergence figures in Table 4 - Table 8 represent the mean over 30 independent runs, and curves denote the evolution of the fitness/performance of the heuristic as new solutions are sampled. The legend of the convergence curves is encoded in the bottom of each figure.

By observing the convergence performance of all algorithms overall control instances, we observe that using relatively large population size is beneficial in obtaining reasonable convergence; however, we also observe that it is possible to obtain competitive fitness/performance convergence with smaller population sizes, whereby attaining the desirable/reasonable convergence at 500 - 1500 function evaluations. For instance, by observing the results in motor velocity control Table 5, convergence occurs in the range of 500 - 1500 function evaluations; yet convergence on instances 1 - 3 becomes more challenging for some algorithms in smaller population sizes. On the other hand, by observing the results on inverted pendulum control in Table 7, we note that convergence occurs in the range of 500 - 4000 function evaluations, yet it is possible to obtain competitive fitness/performance convergence in the range of 500 - 1500 function evaluations across all control instances. Although the above-mentioned similar observations can be obtained on crane stabilization problem, results in Table 8 show the highest variability across convergence of algorithms. This observation pinpoints the difficulty in attaining convergence in crane stabilization control problems across independent runs.

In order to compare the performance of the algorithm across independent runs, we performed statistical comparisons using the Wilcoxon rank-sum test at 5% significance level; as such, by conducting pairwise statistical significance tests between algorithms, it is straightforward to compare fitness/performance metrics across independent runs to evaluate when an algorithm is significantly better(+), equal(=), or worse(−) compared to another algorithm. Since we evaluate 12 algorithms, we conducted 132 pairwise comparisons per control instance and population size. Then, it is possible to count the number of times an algorithm is significantly better(+), equal(=), or worse(−) compared to other algorithms overall control instances. Table 9 shows the summary of the statistical comparisons overall algorithms and control instances. Here, in the top, Table 9 shows whether a heuristic performs better (bars in blue), similarly to (bars in cyan), or worse than (bars in yellow) other algorithms across all control instances. As such, the x-axis of Table 9 shows the algorithm instance, ordered by rank from left to right, and the y-axis shows the count of the number of instances.

By observing the comparative results from Table 9, we note that DEP performs better than other algorithms in 558 times across all control instances, achieving the highest rank overall control problems and instance runs; on the other hand, OBDE underperforms in 564 times, achieving the lowest rank among all algorithms. For detailed reference over distinct classes of control problems, Table 9 also shows the statistical comparisons per type of control problem; for instance,

TABLE 4. Convergence performance of DC motor position control.

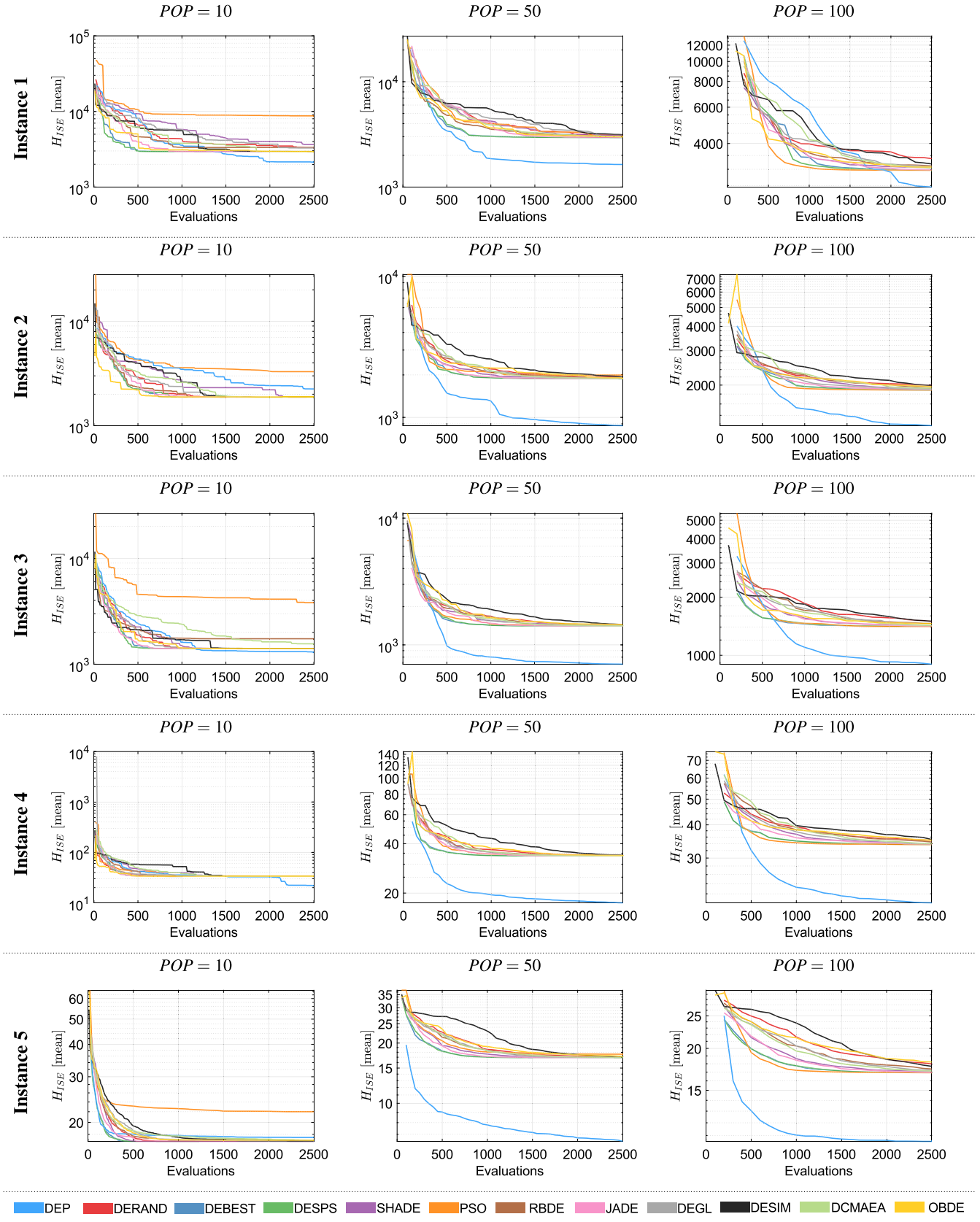


TABLE 5. Convergence performance of DC motor velocity control.

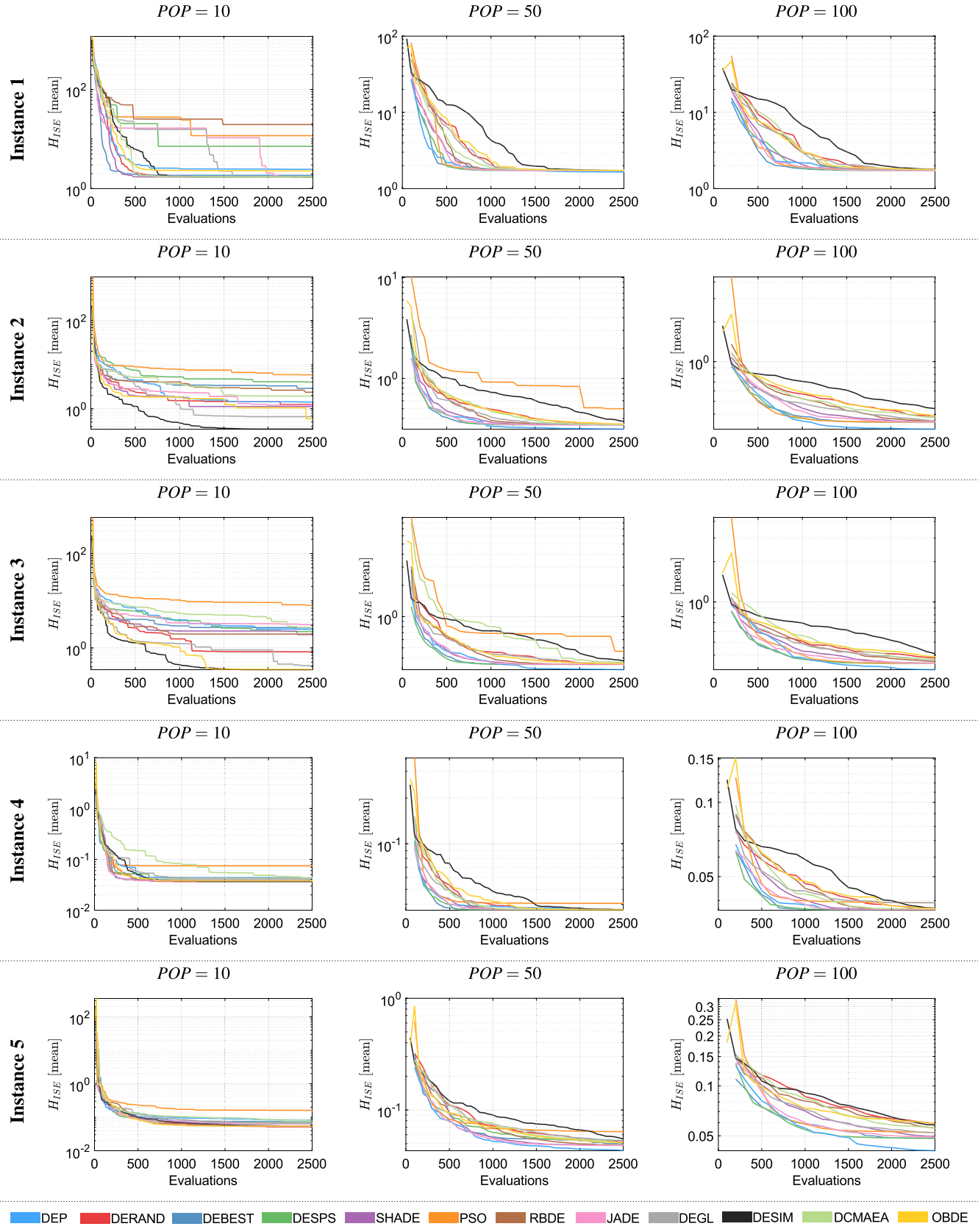


TABLE 6. Convergence performance of magnetic levitation control.

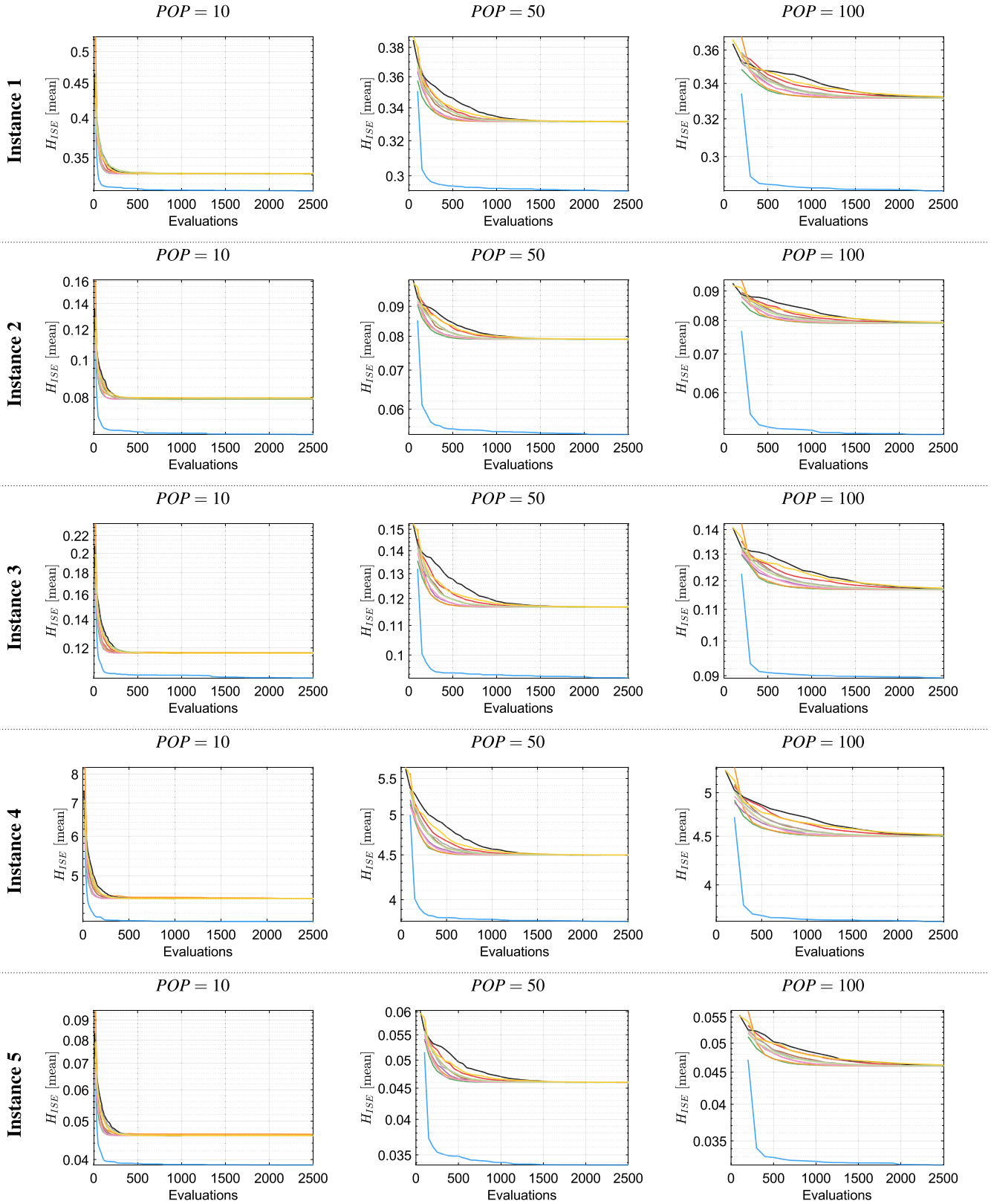


TABLE 7. Convergence performance of **inverted pendulum** control.

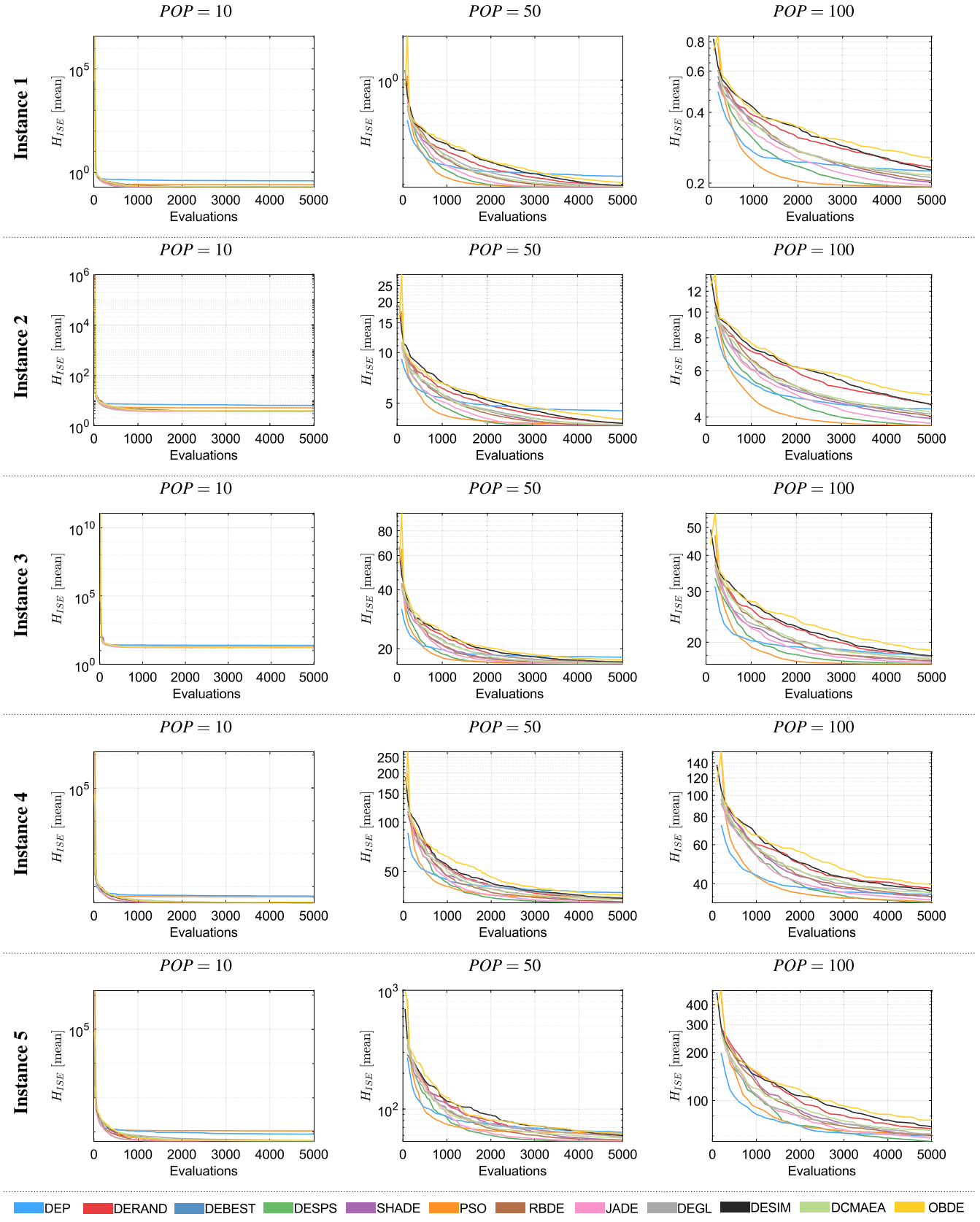


TABLE 8. Convergence performance of crane stabilization control.

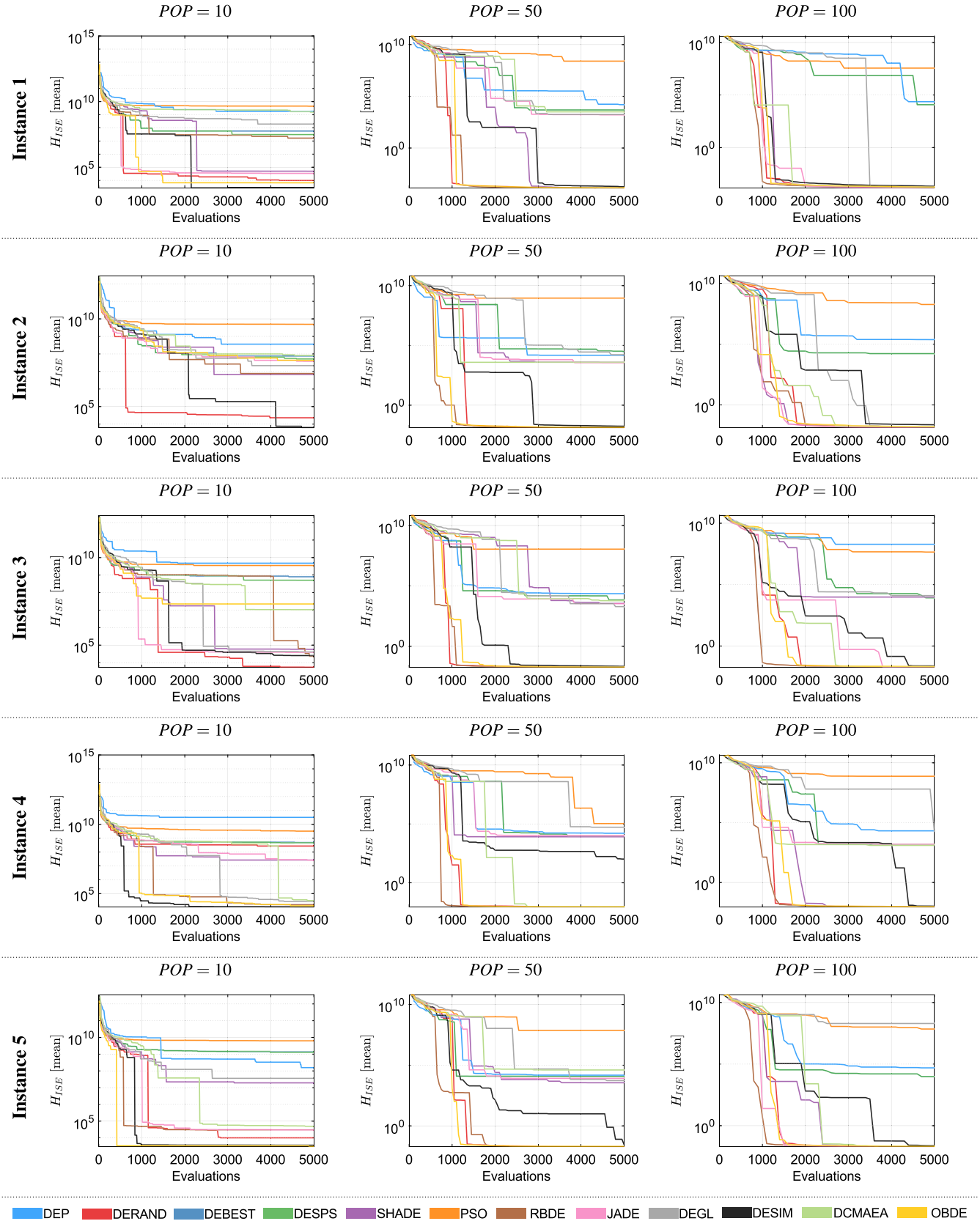
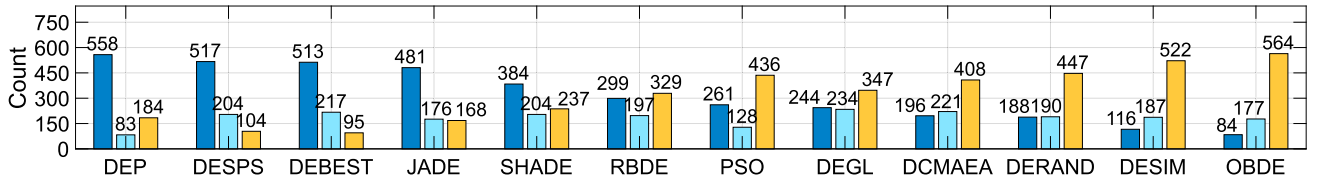
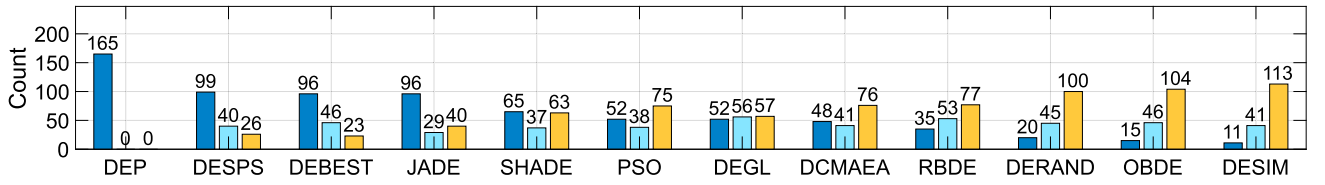


TABLE 9. Summary of statistical comparisons.

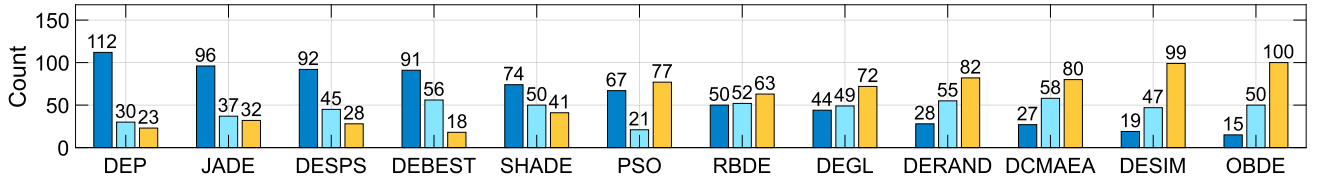
Overall Comparison



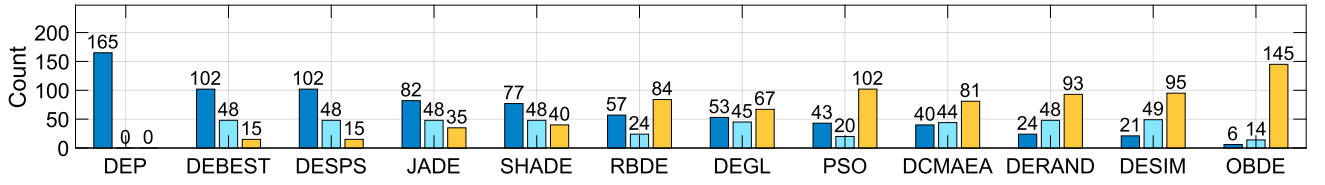
1. DC Motor Position Control



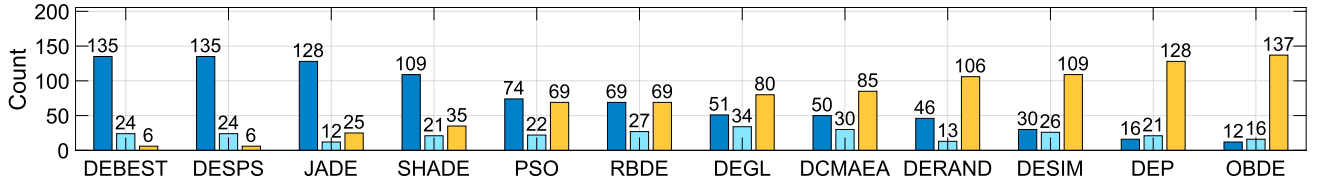
2. DC Motor Velocity Control



3. Magnetic Levitation



4. Inverted Pendulum



5. Crane Stabilization

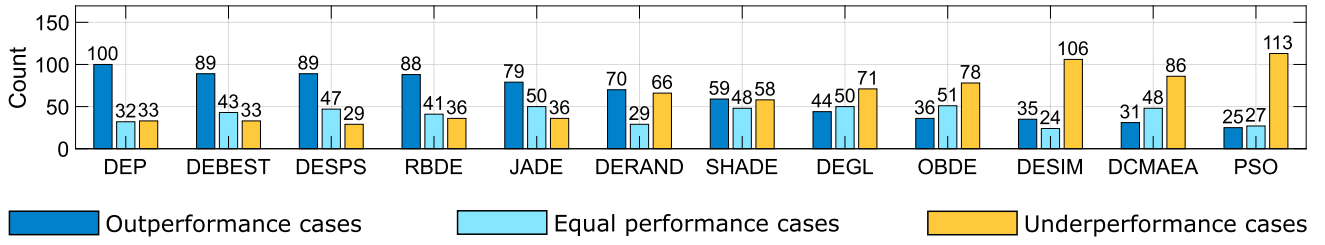
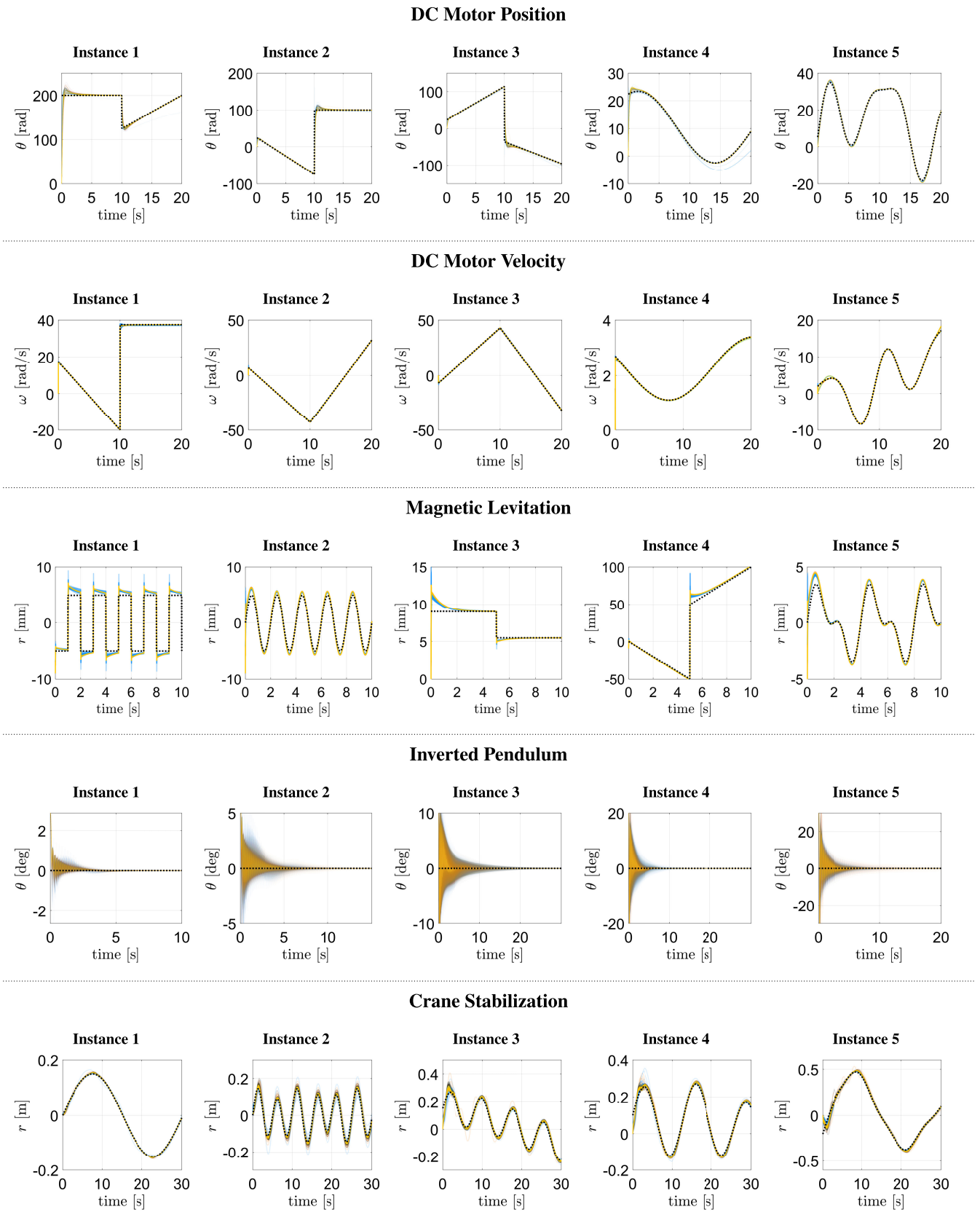


TABLE 10. Performance of best obtained gains overall algorithms.



we observe that DEP performs better than other algorithms in 165 times in the motor position control problem, showing no instances of equal nor underperformance (numbers in zero corresponding to equal/underperformance cases). However, DESIM underperforms in 113 of cases, attaining the lowest rank across all the motor position control instances. Overall, DEP shows the attractive performance over most control problem instances, whereas algorithms with an explorative behaviour such as DERAND and OBDE underperform compared across all control problems.

By observing the order of ranks from left to right in Table 9, we observe that the second best performer is DESPS, followed by DEBEST, JADE and SHADE. The fact of both DEP and DESPS using archives to track potential solutions through successful mutations/selection clarifies the importance of using memory-based and alternatives schemes in case detrimental mutations occur along the search procedure. However, the particle-based difference of vectors in DEP outperforms the conventional DE-based difference of vectors. Furthermore, although DEP in this section does not use parameter adaptation schemes, it outperforms the implicit parameter adaptation approaches from JADE and SHADE. Although JADE and SHADE follow similar parameter adaptation principles, they are unable to outperform the archive-based algorithms and a DE-based algorithm based on an exploitative strategy.

In order to show the tracking performance across control instances and independent runs, Table 10 shows the best control profile responses that correspond to optimized PID gains by each algorithm. Here, for each algorithm, we computed the best response out of 3 types of population sizes and 30 independent runs. Thus, due to comparing 12 optimization algorithms, 25 control instances, Table 10 shows 300 ($= 12 \times 25$) response curves depicting the performance of 300 elite optimized PID gains. The x-axis of all figures in Table 10 show the simulation time t in seconds, and the y-axis shows the observed state of the system. As such, the y-axis shows the angular rotation of the motor θ for a motor position control problem, the angular velocity of the motor ω for a motor velocity control instance, the displacement r of the levitating object for magnetic levitation, the orientation of the pendulum arm θ for an inverted pendulum control problem, and the position of the cart r for crane stabilization. Also, for ease of reference and clarity of plots, we use alpha-transparency to show the overlapping nature of obtained responses across algorithms. The set trajectory is depicted with a dashed style, whereas the achieved response trajectory rendered from the obtained PID control is depicted with a continuous line. By observing the results of Table 10, we note the reasonable control performance across control instances. Although we observe overshooting and noticeable oscillations at the early stages of the change of set trajectories, we observe the reasonable tracking of the set trajectories within a few seconds, irrespective of the challenging nature of the task. The above-mentioned observations show the amenability of the obtained gains based on the difference of vectors, suggesting the ability to compute feasible PID configurations within a small number of function evaluations.

I. HEURISTICS WITH ADAPTATION MECHANISMS

Having described the overall set of DE-based benchmark heuristics for overall comparisons in subsection IV-G, and the performance of optimization algorithms forgoing the parameter adaptation mechanisms in subsection IV-H, this subsection describes the control and convergence performance of the subset of optimization algorithms using the parameter adaptation mechanisms, that is the modified versions with the " A" subscript as described in section IV-G.

Therefore, in a second set of experiments, we studied the convergence performance using the adaptive versions of the list of algorithms mentioned in section IV-G. As such, in this section we used DEPA, which is the heuristic using the parameter adaptation mechanisms presented in (14)-(19). For fair comparisons across benchmark algorithms, we used the extended heuristics described in section IV-G, that is the modified versions with the " A" subscript, as described in section IV-G. Also, to complement our evaluations presented in section IV-H, we used IAE, ITAE and ITSE as the fitness/performance functions for all algorithms and control instances. Furthermore, we used 10 independent runs for each algorithm execution/configuration, the population size $N = 10$, and the termination criterion at $MaxFEs = 1000$ function evaluations. Other parameters followed the suggested values of the above-mentioned references.

To show the convergence of the studied algorithms, Table 11 - Table 15 show the overall convergence performance of all adaptive algorithms under distinct fitness/performance metrics, evaluated across each control instance. In these figures, the x-axis shows the number of function evaluations, and the y-axis denotes the value of the fitness/performance metric. The convergence figures in Table 11 - Table 15 are ordered by fitness/performance metric (horizontal direction) and control instance (vertical direction). Also, the convergence curves represent the mean over 10 independent runs. The corresponding legend of the heuristic is presented at the bottom of each figure. By observing the convergence performances, we observe that it is possible to obtain reasonable convergence behaviours across fitness/performance metrics and control instances. Although most control problems show the suitability for quick convergence in the order of 250 - 500 function evaluations (e.g., magnetic levitation control in Table 13), the convergence results in crane stabilization in Table 15 show the highest variability across control instances and fitness/performance metrics, pinpointing the difficulty to realize the effective sampling of the search space.

To evaluate the comparative performance across independent control instances, we conducted statistical comparisons using the Wilcoxon rank-sum test at 5% significance level. Table 16 shows the summary of the statistical comparisons overall adaptive algorithms and fitness/performance metrics. Following a similar organization to section IV-H, the top of

TABLE 11. Convergence performance using parameter adaptation in DC motor position control.

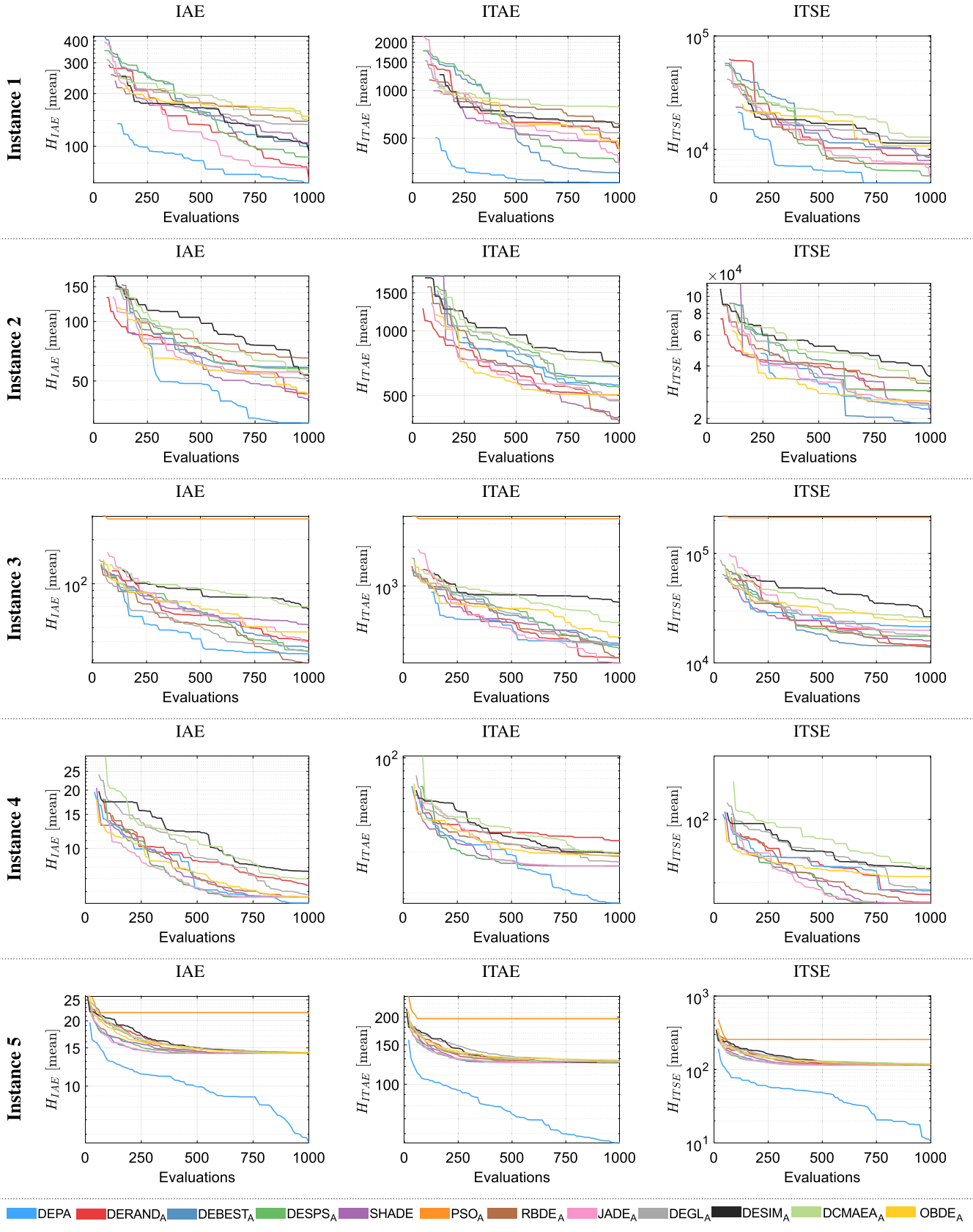


TABLE 12. Convergence performance using parameter adaptation in DC motor velocity control.

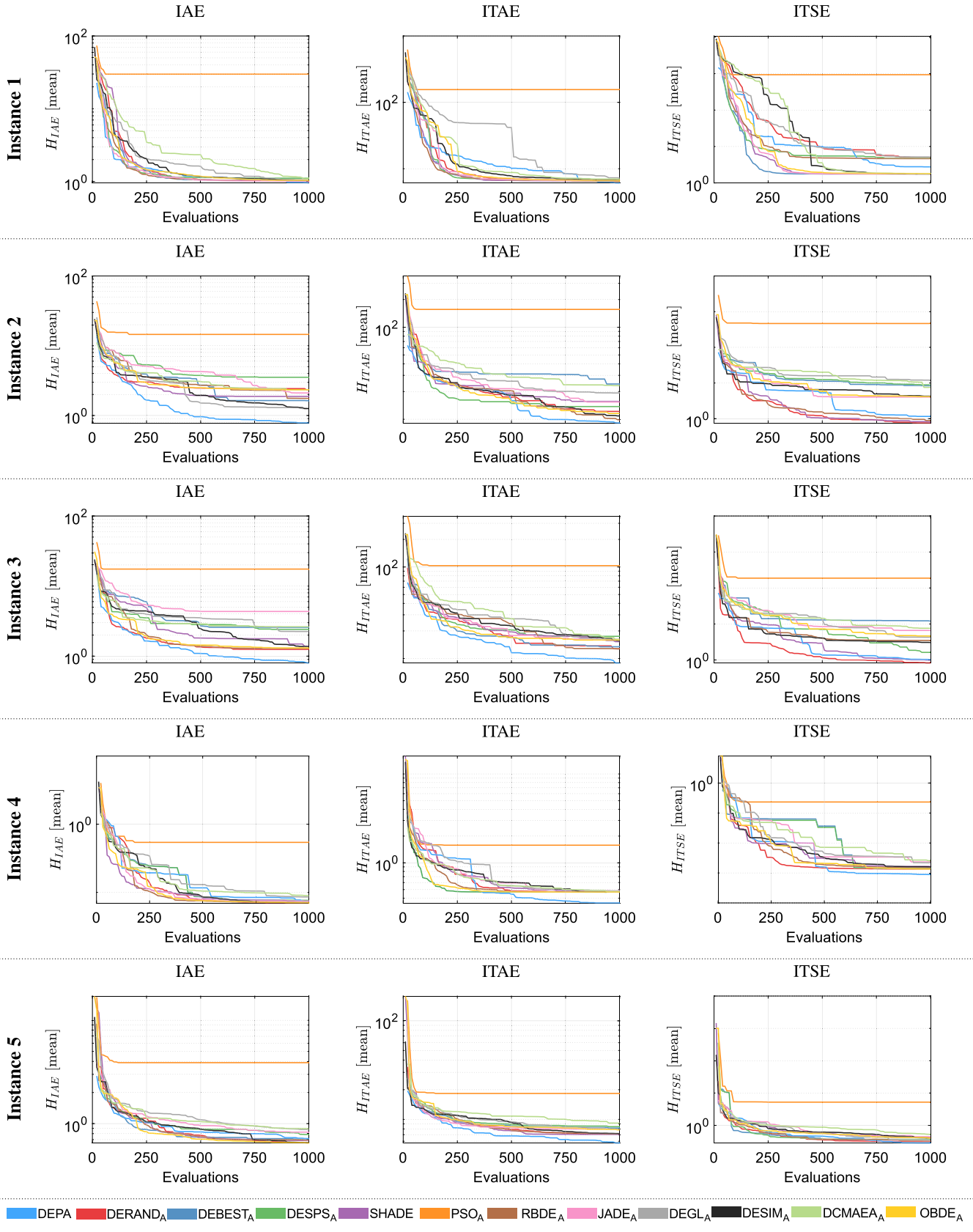


TABLE 13. Convergence performance using parameter adaptation in magnetic levitation control.

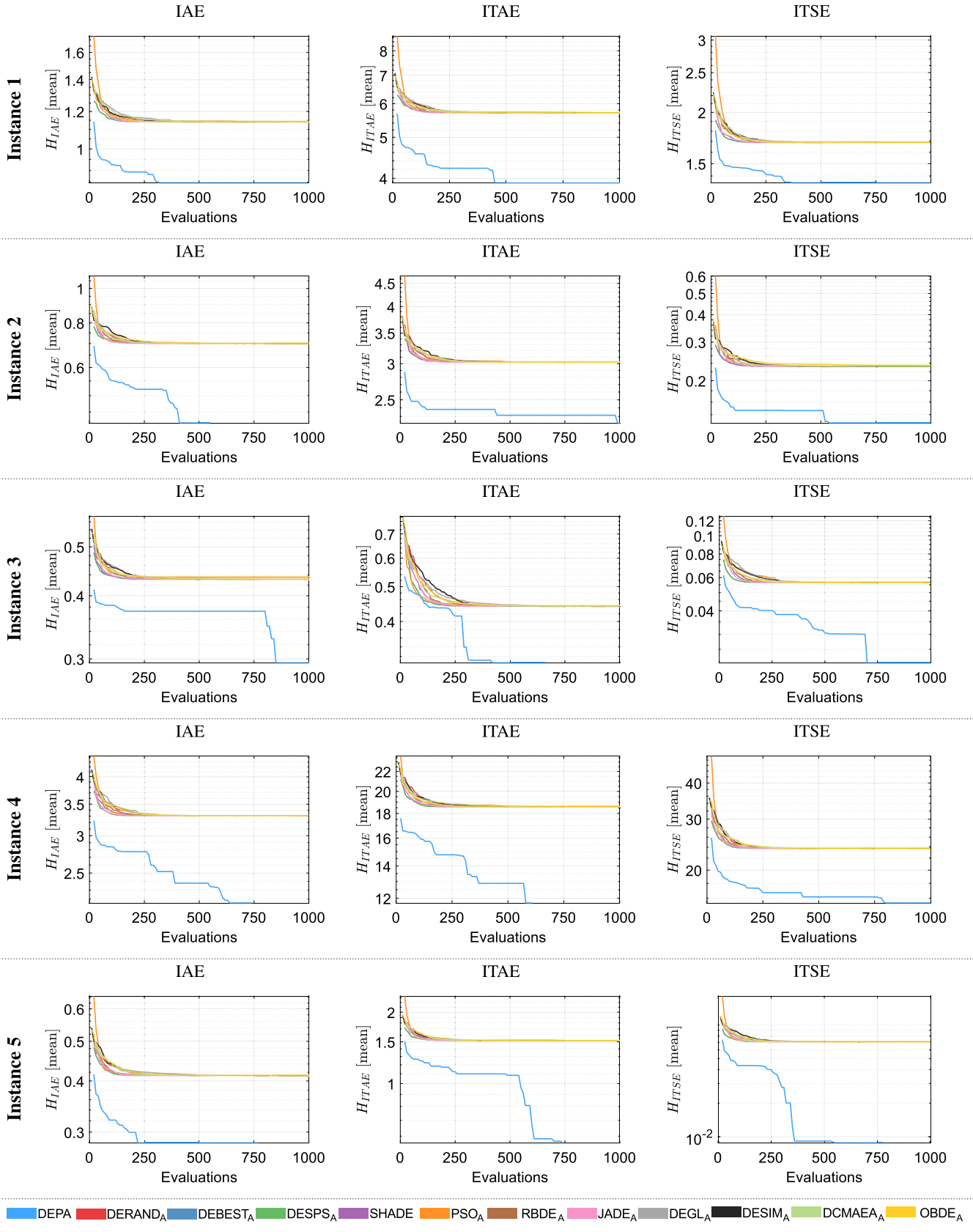


TABLE 14. Convergence performance using parameter adaptation in inverted pendulum control.

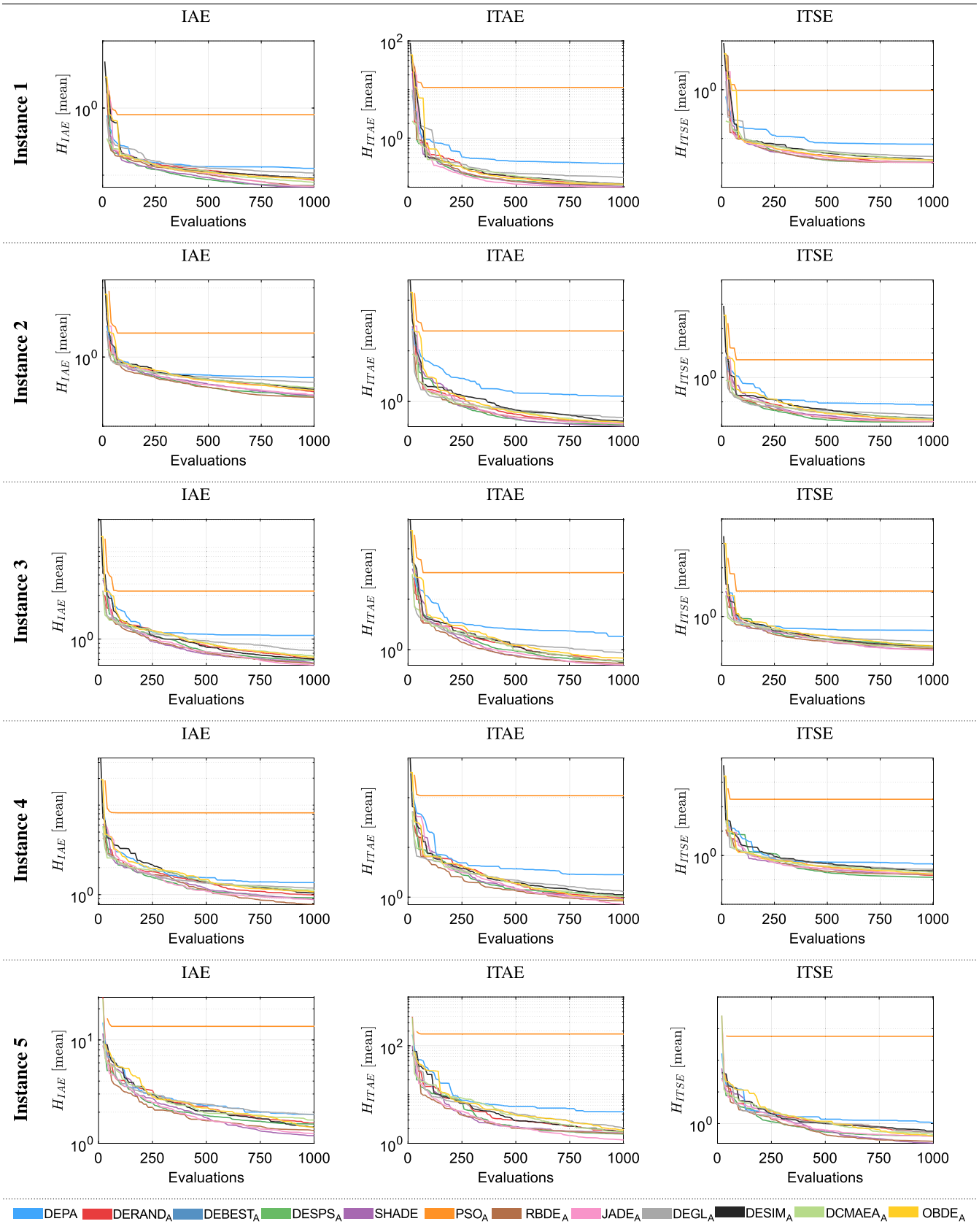


TABLE 15. Convergence performance using parameter adaptation in crane stabilization control.

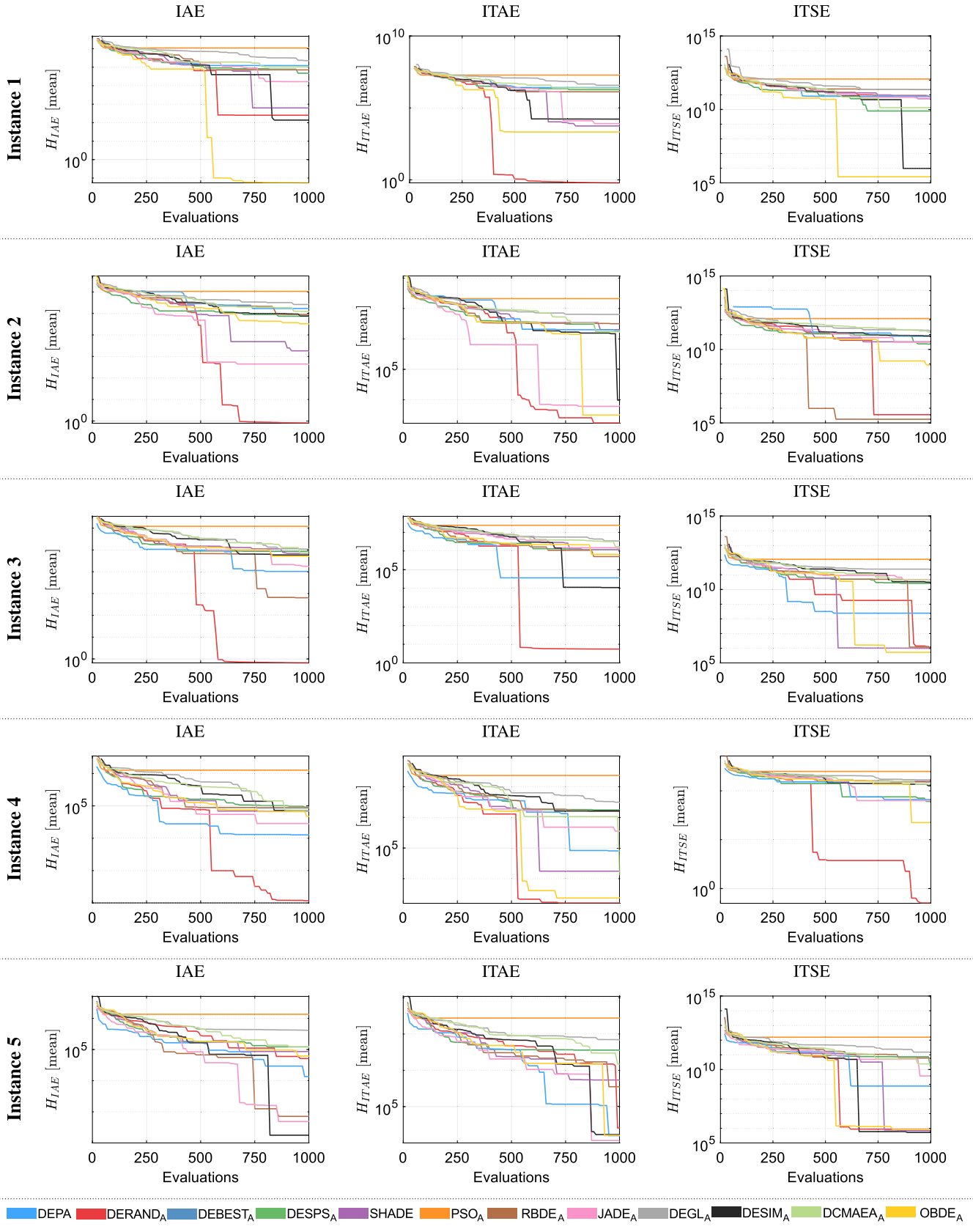
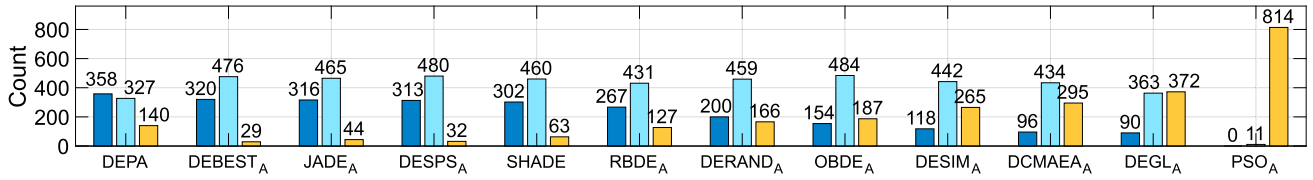
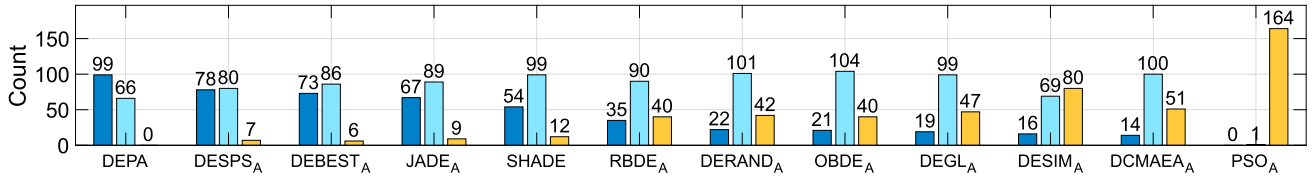


TABLE 16. Summary of statistical comparisons considering IAE, ITAE and ITSE and **parameter adaptation** in all algorithms.

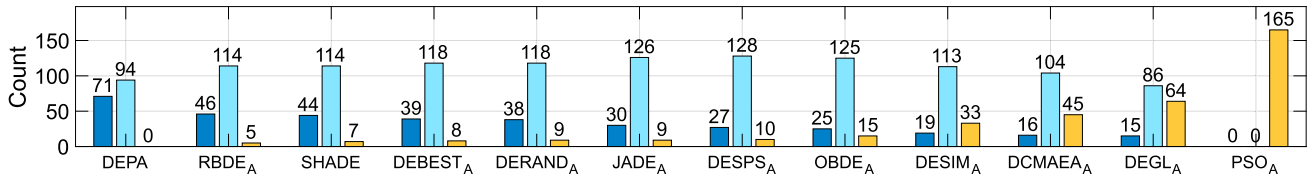
Overall Comparison



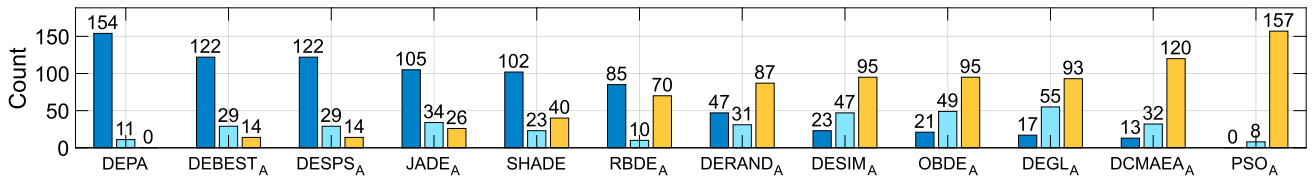
1. DC Motor Position Control



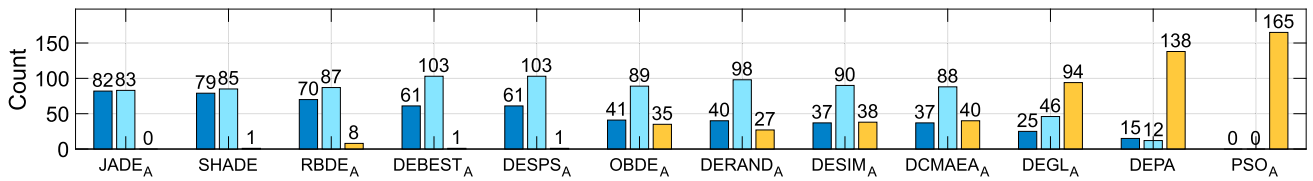
2. DC Motor Velocity Control



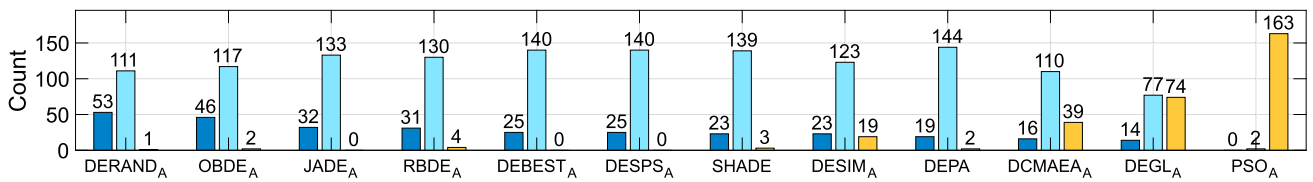
3. Magnetic Levitation



4. Inverted Pendulum



5. Crane Stabilization



Outperformance cases Equal performance cases Underperformance cases

TABLE 17. Performance of best obtained gains overall algorithms for DC motor position control.

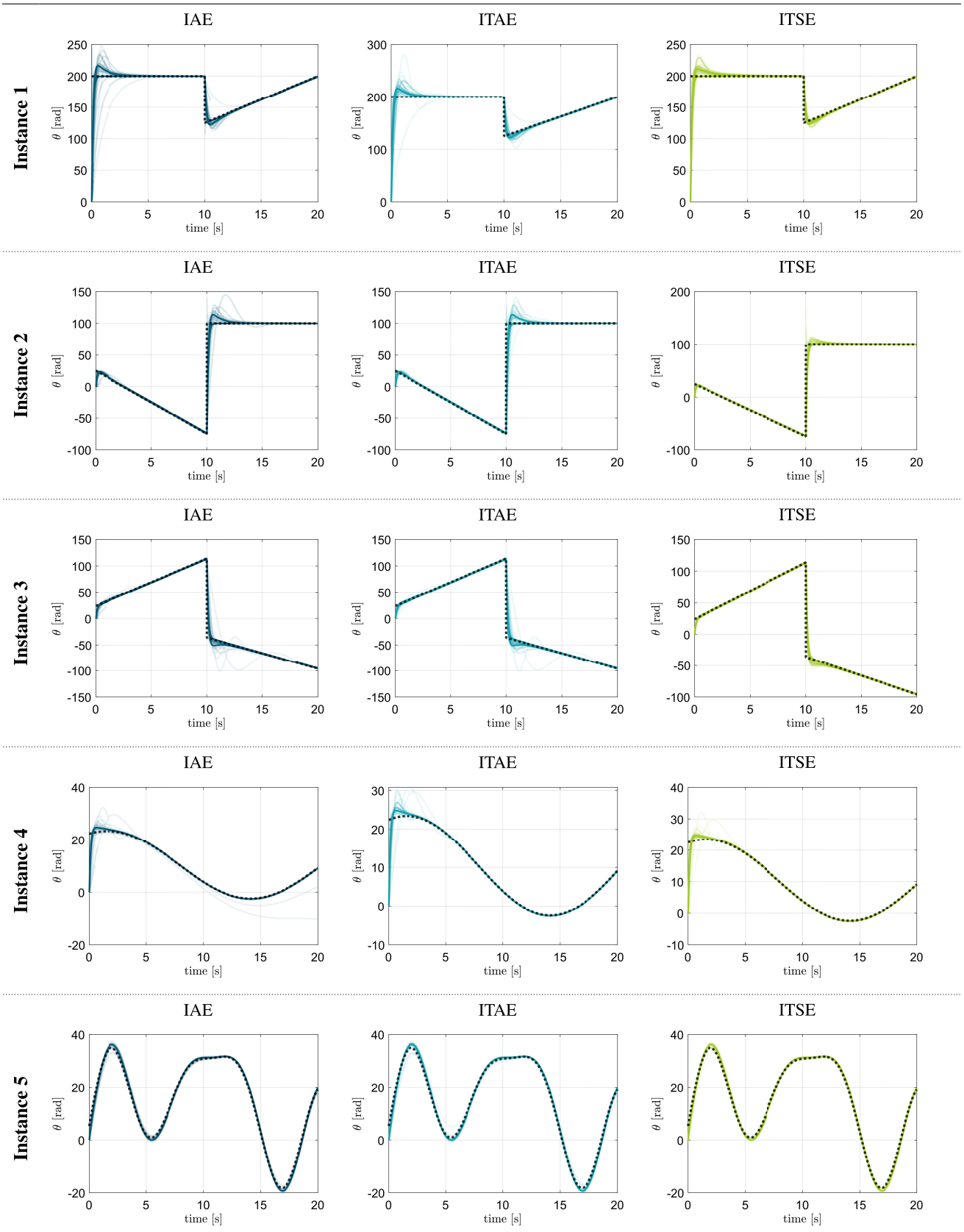


TABLE 18. Performance of best obtained gains overall algorithms for DC motor velocity control.

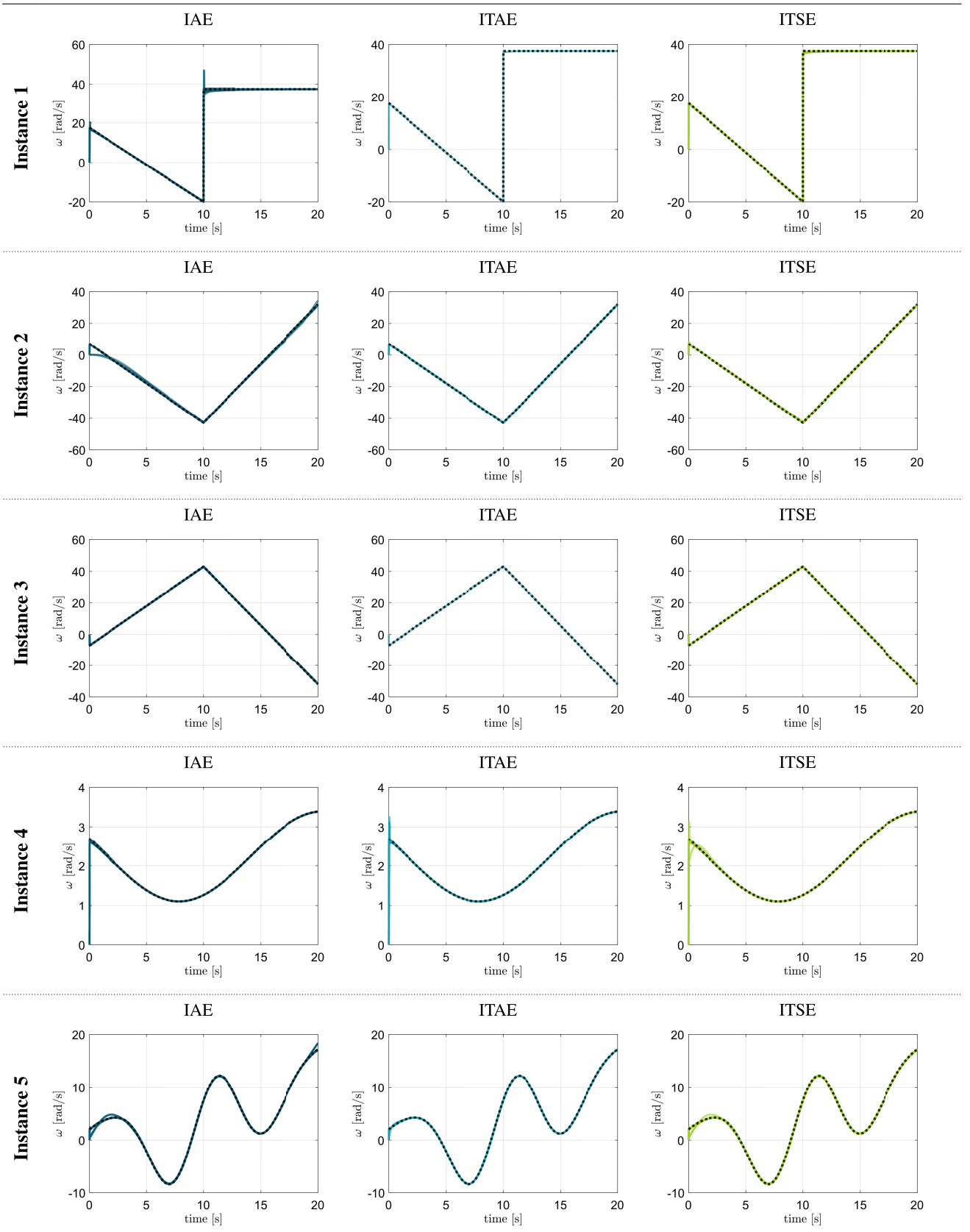


TABLE 19. Performance of best obtained gains overall algorithms for magnetic levitation control.

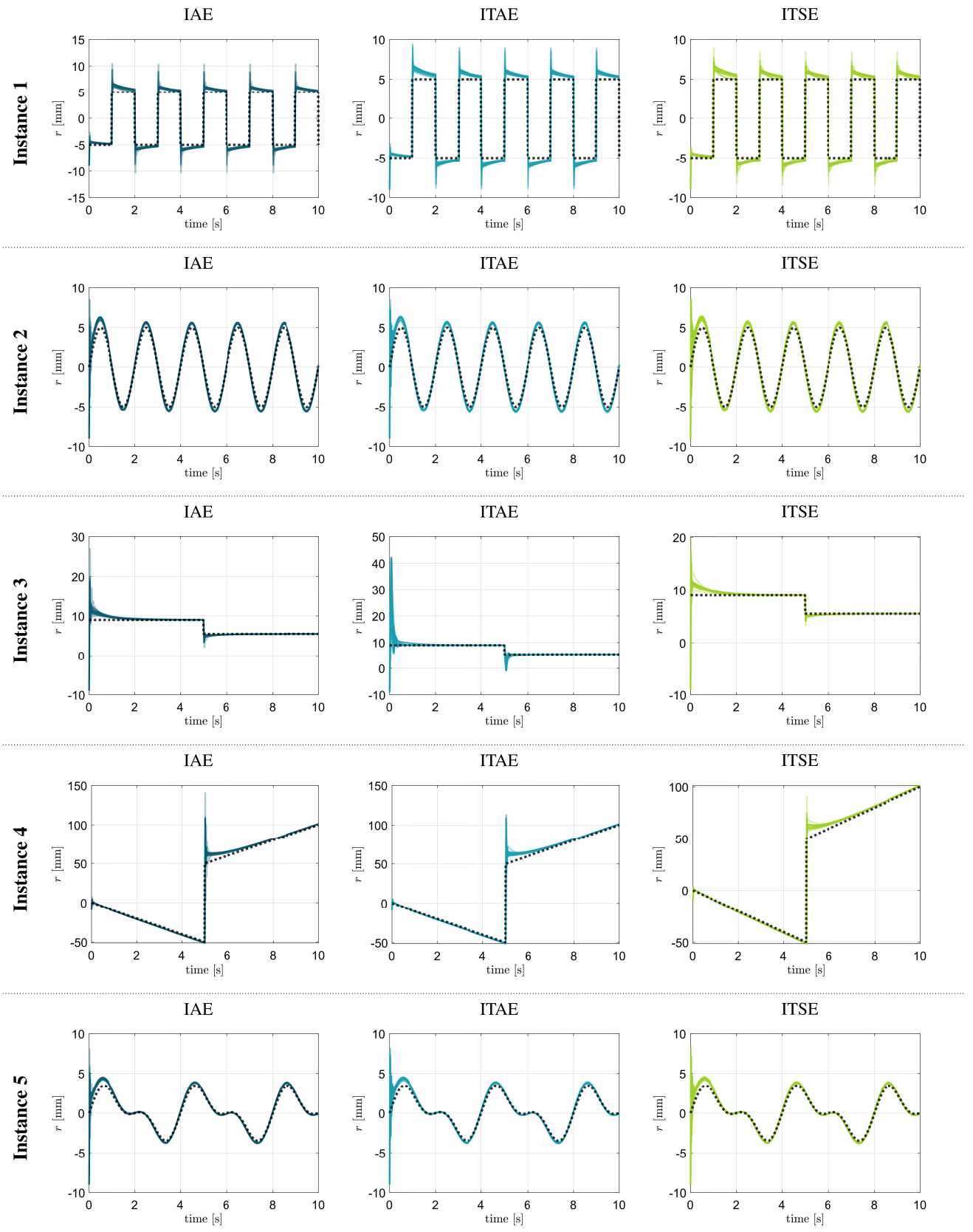


TABLE 20. Performance of best obtained gains overall algorithms for **inverted pendulum control.**

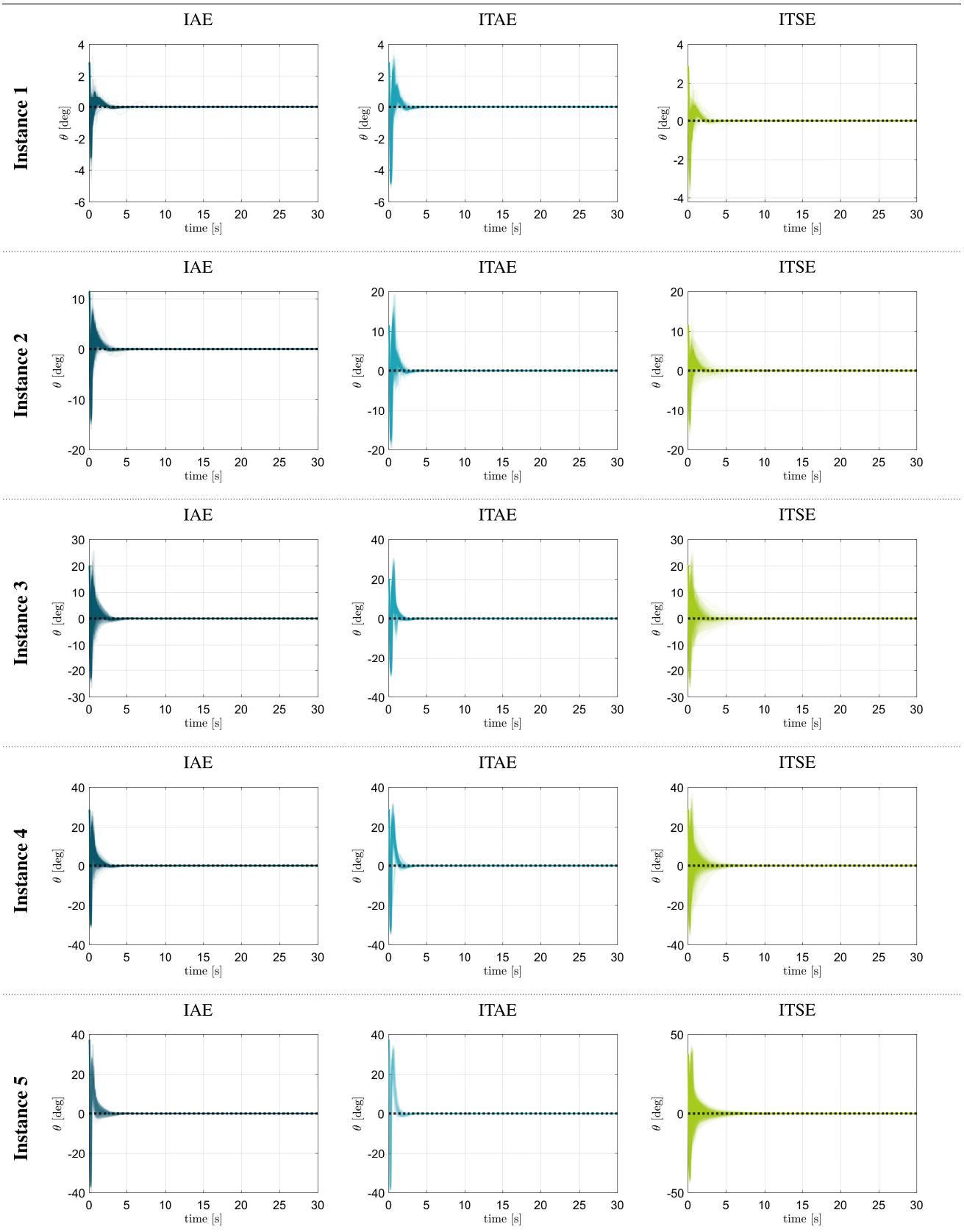


TABLE 21. Performance of best obtained gains overall algorithms for crane stabilization control.

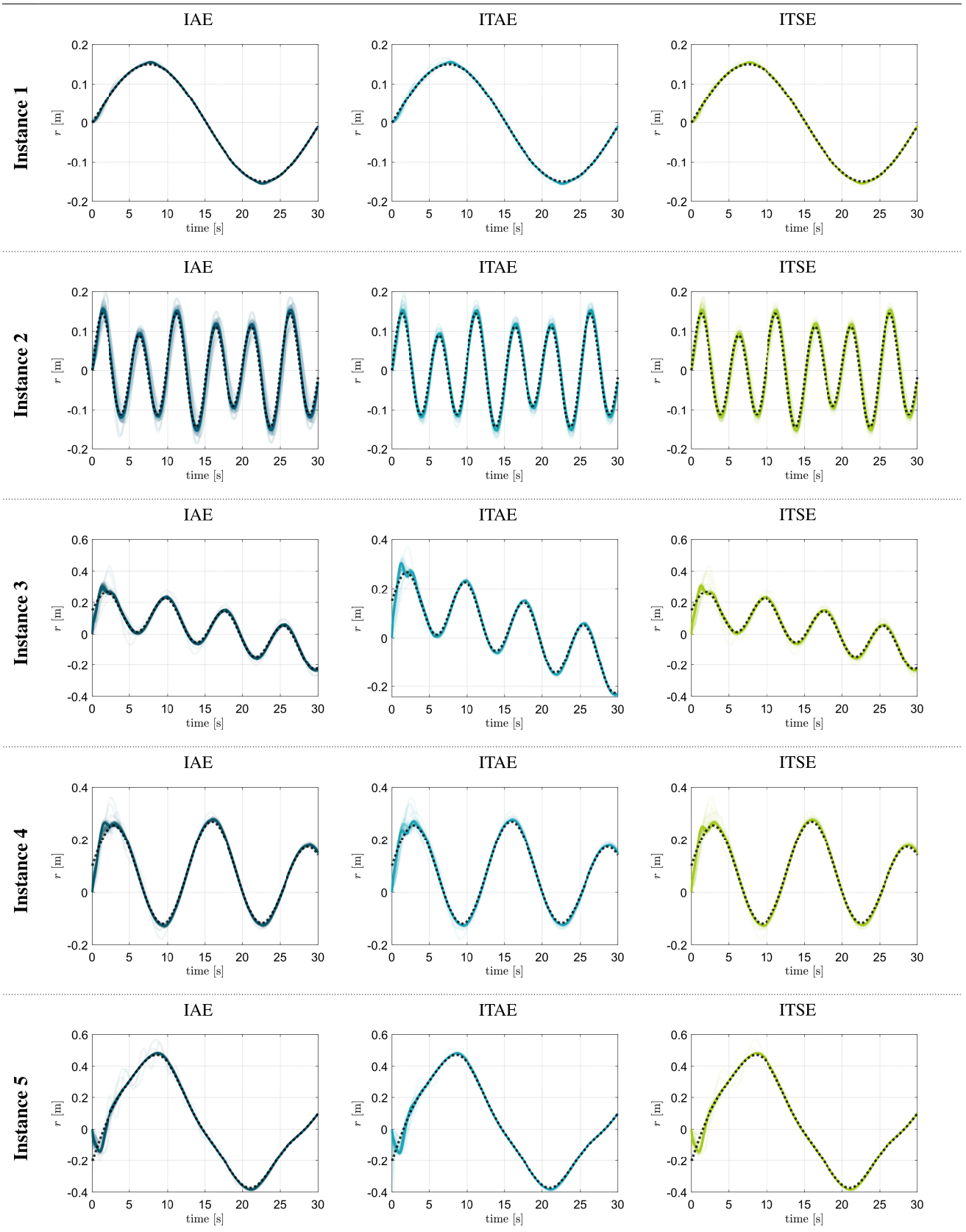


Table 16 shows whether a heuristic performs better (bars in blue), similarly to (bars in cyan), or worse than (bars in yellow) other algorithms across all control instances and performance metrics. Thus, the x-axis of each figure in Table 16 shows the adaptive algorithm instance, ordered by rank from left to right, and the y-axis shows the count of the number of times/cases.

By observing the comparative results from Table 16, we note that DEPA performs better than other algorithms in 358 times/cases across control instances and performance metrics, achieving the highest rank overall control instances. We also note that PSO_A underperforms in 814 of the times, achieving the lowest rank among all algorithms. In line with the organization section IV-H, Table 16 shows a detailed comparison for distinct classes of control problems. For instance, we observe that DEPA performs better than other algorithms in 99 (71) times in motor position (velocity) control problem, showing no instance of underperformance (numbers in zero corresponding to underperformance cases). Although DEPA performs similar to other algorithms in the crane stabilization problem (144 times), we observe that DEPA underperforms in inverted pendulum control instances (138 cases). This observation pinpoints the difficulty of the landscape for the type of DEPA-based algorithms. By observing the results overall control instances and fitness/performance metrics, DEPA shows the attractive performance over most control problem instances and fitness/performance functions, whereas algorithms using composition of global-local donors/difference of vectors such as DEGL_A and PSO_A underperform compared across all control problems and fitness/performance metrics.

By observing the order of ranks of the comparative performances in Table 16, from left side (showing the best heuristic) towards the right side (showing the heuristics with largest number of underperformance cases), the overall attractive performances of DEPA is followed by the meritorious performances of DEBEST_A, JADE_A, DESPS_A and SHADE. On the other hand, we can also note that PSO_A underperforms overall control instances. Whereas DEBEST_A is an algorithm known for its exploitative behaviour [112], its adaptive extension through the archive-based parameter adaptation is meritorious against other algorithms with highly exploitative features such as RBDE_A. We also note that algorithms with exploitative features such as DEBEST_A and DESPS_A outperformed other exploration and exploration-exploitation switching mechanisms such as DERAND_A, DESIM_A and OBDE_A, each of which was unable to counter-balance the convergence of performance of other variants.

Also, the modified JADE_A follows the same mutation principle (current-to-pbest/1/bin strategy) and the same adaptation principle compared to SHADE; however, JADE uses a less explorative approach when generating the best referent vectors for mutation. Furthermore, although DEPA and DESPS_A use similar parameter adaptation principles, the use of particle adaptation schemes in DEPA is meritorious when considering the overall control instances. Yet, the use of archives to track potential solutions through successful mutations/selection mechanisms in DEPA and DESPS_A clarifies the usefulness of using archive-based schemes to counteract detrimental mutations during sampling. The exploitation mechanism of DEPA relies on the adaptive importance of the direction towards the global and local best, yet DEPA has a regulated and sporadic stagnation-avoidance mechanism that allows for a possible exploration of the search space through a historical archive. It is noteworthy to mention that each of the above-mentioned algorithms embeds a form of exploitative behaviour along with a degree explorative mechanism. We argue that algorithms with a higher degree of exploitation and regulated exploration of the search space, such as DEPA, DEBEST_A, JADE_A, DESPS_A and SHADE_A have the potential to tackle the search space of PID optimization in the scope of motor control, magnetic levitation, inverted pendulum and crane stabilization problems.

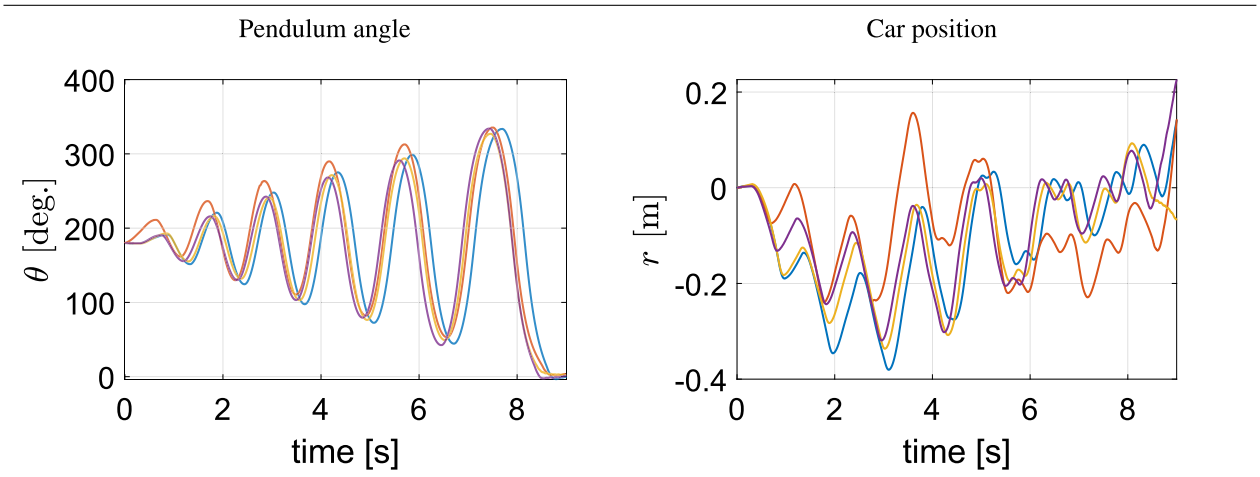
Following a similar principle in section IV-H, in order to show the profile of tracking performance across control instances and distinct fitness/performance metrics, Table 17 - Table 21 show the best control profile responses that correspond to optimized PID gains by each algorithm. The x-axis of all figures in Table 17 - Table 21 show the simulation time t in seconds, and the y-axis shows the observed state of the system. Also, for ease of reference, we use alpha-transparency to render the most common behaviours of obtained responses across algorithms. We use the same style to render plots of control profiles: the set trajectory is shown with a dashed style, and the achieved response trajectory from the PID control is shown by a continuous line. Due to using distinct performance metrics, response profiles are rendered with different, yet distinctive colors. By observing the results of Table 17 - Table 21, we note the reasonable control performance overall control instances and fitness/performance metrics. By observing the response results from Table 17 - Table 21, it is possible to note overshooting and oscillation behaviours, mainly during the early stages of the simulation and during the change of the non-continuous set of trajectories. Yet, the set trajectories can be generally tracked with reasonable performance.

The above-mentioned observations pinpoint the feasibility of attaining competitive convergence for PID parameter tuning. Investigating the performance of the proposed algorithm in the corresponding diverse class of hardware environments and devising new operators for many-objective optimization (e.g., inclusion of robustness constraints as an additional objective, or combination of multiple fitness/performance metrics), and the suitable switching of exploration-exploitation mechanisms that lead to efficient convergence on a broader class of control problems and dynamic scenarios are on our agenda.

J. EXPERIMENTS ON HARDWARE

Having described the convergence and control performance through computational experiments (simulations) in subsection IV-H and subsection IV-I, this subsection describes the

TABLE 22. Performance of the inverted pendulum device.



control performance and the experiments in hardware when using optimized PID gains on an inverted pendulum device.

For simplicity, we evaluated four arbitrary obtained gains on a cart-pendulum device by Feedback Model 33-936S [119] (which is related to the scope of study). In order to evaluate the performance in unseen environments during training, we used the initial angle $\theta_0 = \pi$rad, which is far from the initial set in the range $\theta_0 \in [0.05, 0.65]$rad used during training (simulation). Table 22 shows the performance of the pendulum angle θ and the car position r. By looking at Table 22, we can note the oscillations in the arm and the car to attain the desirable state at $\theta = 0$rad (vertical position) over four independent trials, each using distinct learned gains. In order to show a glimpse of the pendulum states, Table 23 shows the transition of the states from initial configuration (top-left) to the attained desirable state (bottom-right). Our results portray the potential for transferability of the learned gains to unseen environments.

V. COMPUTATIONAL EXPERIMENTS ON SYNTHETIC MATHEMATICAL FUNCTIONS

Having described the performance and benchmark comparisons in control systems/tasks in section IV, this section describes the performance in a different class of optimization problems derived from the evolutionary computing community.

Thus, in this section, we considered the potential extension/generalization to other problem domains and use the CEC 2017 benchmark set [120] as a key reference to tackle single-objective benchmark functions with varying landscapes such as unimodality, multimodality, hybrid functions and their compositions. Since the formulations in section III were inspired by/towards the suitable exploitation in expensive PID control problems, the suitable adaptation between exploration and exploitation in the difference of vectors and overall dimensions becomes essential to ensure the reasonable convergence performance for a diverse set of problem domains such as the CEC 2017 benchmark set [120]. In particular, we consider the following potential extension of DEPA to consider both exploitative and explorative vectors into the sampling mechanisms, and their adaptation mechanisms. For simplicity and ease of reference, we describe the full algorithm, as follows:

$$\boldsymbol{x}_{i,g+1} = \begin{cases} \boldsymbol{u}_{i,g}, & \text{if } f(\boldsymbol{u}_{i,g}) < f(\boldsymbol{x}_{i,g}) \\ \boldsymbol{x}_{i,g}, & \text{otherwise} \end{cases} \tag{41}$$

$$\boldsymbol{u}_{i,g} = \boldsymbol{x}_{i,g} + \boldsymbol{v}^*_{i,g} \tag{42}$$

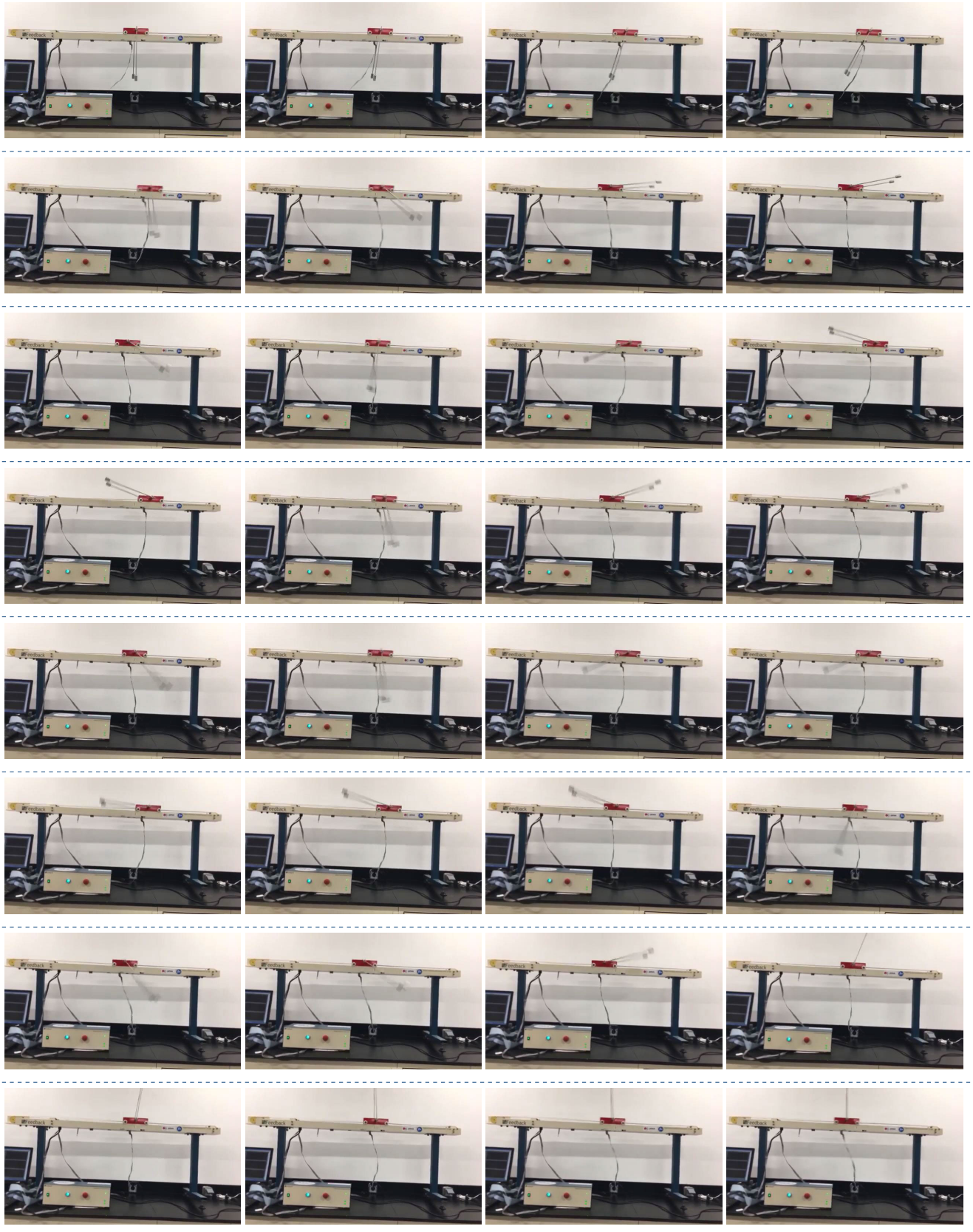
$$\boldsymbol{v}^*_{i,g} = \omega \boldsymbol{v}_{i,g} + F_i(\boldsymbol{x}_{\text{pbest},g} - \boldsymbol{x}_{i,g})\boldsymbol{r}_i + F_i(\boldsymbol{x}_i^{r_1} - \boldsymbol{x}_i^{r_2})\boldsymbol{r}_i \tag{43}$$

$$\boldsymbol{v}_{i,g+1} = \begin{cases} \boldsymbol{v}^*_{i,g}, & \text{if } f(\boldsymbol{u}_{i,g}) < f(\boldsymbol{x}_{i,g}) \\ \boldsymbol{v}_{i,g}, & \text{otherwise} \end{cases} \tag{44}$$

where $\boldsymbol{x}_{i,g}$ denotes the i-th individual/solution vector in the g-th generation/iteration; and $f(\boldsymbol{x}_{i,g})$ denotes the fitness function of the i-th individual/solution. Along with the general notations of section III, $\boldsymbol{u}_{i,g}$ is a trial vector, $\boldsymbol{v}^*_{i,g}$ is a trial velocity vector associated to the i-th individual, ω is a smoothing coefficient on the velocity vector, and F_i is a scaling factor associated with the i-th individual. Here, the major differences consist in the following:

- We implement (43) to allow the suitable balance between exploitation and exploration and the fair comparisons with the related adaptation mechanisms in the CEC 2017 benchmark set [108], [109], [116], [120]. As such, we allow $\boldsymbol{x}_{i,g} = \boldsymbol{x}^r_{i,g}$ to become the reference solution vector, enable a single step parameter $F_i = G_i$ for individual i, and compute $\boldsymbol{x}_{\text{pbest},g}$ as a best arbitrary solution from the top $100p\%$ solutions at generation g, for small $p \in [0, 1]$. In the above, $\boldsymbol{x}_{\text{pbest},g} - \boldsymbol{x}_{i,g}$ implies the vector favoring exploitation, whereas $\boldsymbol{x}_i^{r_1} - \boldsymbol{x}_i^{r_2}$ implies the vector favoring exploration in which $r_1 \neq r_2$.

TABLE 23. Performance of the experiment on inverted pendulum.



- Furthermore, we generalize the scaling coefficients as a binary vector $\boldsymbol{r}_i \in \{0,1\}^D$, for dimensionality D, to consider the suitable adaptation of steps in each dimension, as follows:

$$r_{i,j} = \begin{cases} 1, \text{ if } r_u < CR_i \mid j = jrand \\ 0, \text{ otherwise} \end{cases} \tag{45}$$

- The reference vectors $\boldsymbol{x}_i^{r_1}$ and $\boldsymbol{x}_i^{r_2}$ are sampled/chosen uniformly randomly either from the population $\mathcal{P}$ or from an archive (memory) $\mathcal{A}$, as follows:

$$\boldsymbol{x}_{i,g}^{r_1},\ \boldsymbol{x}_{i,g}^{r_2} \in_R \begin{cases} \mathcal{P}, \text{ if } q_i < Q \\ \mathcal{A}, \text{ otherwise} \end{cases} \tag{46}$$

where q_i is a counter of unsuccessful trials and Q is a user-defined threshold. The archive $\mathcal{A}$ accumulates the successful trial solution vectors $\boldsymbol{u}_{i,g}$ following (13).

Furthermore, inspired by parameter adaptation schemes [116], and following similar adaptation principles to (14) - (15), each individual $\boldsymbol{x}_{i,g}$ at generation/iteration g is associated with scaling factors F_i and crossover rate CR_i, each of which is sampled from the Cauchy and Normal distributions:

$$F_i \sim \text{Cauchy}(\mathcal{M}_F^{a_i}, \sigma_F^2) \tag{47}$$

$$CR_i \sim \mathcal{N}(\mathcal{M}_{CR}^{a_i}, \sigma_{CR}^2) \tag{48}$$

where Cauchy(μ, σ^2) denotes the values sampled from a Cauchy distribution with location parameter μ and scale parameter σ^2 [108], $\mathcal{N}(\mu, \sigma^2)$ denotes values sampled from the normal distribution with mean μ and variance σ^2, the parameters $\mathcal{M}_F^{a_i}$ and $\mathcal{M}_{CR}^{a_i}$ denote the a_i-th element of h-dimensional tuples $\mathcal{M}_F$ and $\mathcal{M}_{CR}$, respectively. The positive integer $a_i \sim U[1,h]$, $h = N$, $F_i \in (0,1]$, and $CR_i \in [0,1]$.

Following the archive of successful parameters in section III, the parameters F_i and CR_i associated to the i-th individual are recorded into corresponding (archive) sets S_F and S_{CR}, in which $n = |S_F| = |S_{CR}|$ denotes the number of fitness improvements through successful mutations/selections. As such, the elements of the archives $\mathcal{M}_F$ are updated by the weighted Lehmer mean following (16),

$$\mathcal{M}_F^k = \frac{\sum_{j=1}^{n} w_j \cdot S_{F,j}^2}{\sum_{j=1}^{n} w_j \cdot S_{F,j}},\ k \in [1,h], \tag{49}$$

whereas the elements of $\mathcal{M}_{CR}$ are updated by the weighted mean [109], [121]:

$$\mathcal{M}_{CR}^k = \sum_{j=1}^{n} w_j \cdot S_{CR,j},\ k \in [1,h] \tag{50}$$

where w_j is the weight associated to a *successful* trial vector $\boldsymbol{u}_{i,g}$ following (18). The key motivation of using the archives $\mathcal{M}_F^j$, $\mathcal{M}_{CR}^j$ is to track/log parameters that lead to successful mutations through selective scaling and difference of vectors.

Algorithm 2 DEPA$_2$

```
FEs = 0, g = 0, q_i = 0, k = 0;
Generate a set of N individuals randomly as initial population set P;
Initialize the archive A from the population set P;
Initialize the tuples M_F^k and M_G^k;
FEs = FEs + N;
while FEs ≤ MaxFEs do
    g = g + 1;
    S_F = {}, S_G = {};
    for i = 1 to N do
        Sample a_i ~ U[1, h];
        Generate F_i using (47);
        Generate CR_i using (48);
    end
    for i = 1 to N do
        Generate the trial vector u_{i,g} using (42)-(46);
        if f(u_{i,g}) < f(x_{i,g}) then
            x_{i,g+1} = u_{i,g}, v_{i,g+1} = v*_{i,g};
            u_{i,g} → A;
            Delete an element from A if |A| > λ|P|;
            F_i → S_F, CR_i → S_CR;
            q_i = 0;
        else
            x_{i,g+1} = x_{i,g};
            v_{i,g+1} = v_{i,g};
            q_i = q_i + 1;
        end
        FEs = FEs + 1;
    end
    if S_F ≠ {} ∧ S_G ≠ {} then
        k = k + 1;
        Update M_F^k and M_CR^k using (49) and (50);
        Set k = 1 if k > h;
    end
end
```

The pseudocode of the extended algorithm, labeled as DEPA$_2$, is outlined in Algorithm 2. As such, in every generation, (1) we compute parameters F_i and CR_i associated to each individual; (2) generate trial vectors accordingly; (3) perform mutation and selection mechanisms and, whenever successful mutations/selections occur, we store potential trial vectors and scaling/mutation parameters into their corresponding archive (memory) mechanisms; and (4) update the memory archives considering useful parameters that led to successful fitness improvements. The procedure is repeated until the maximum number of fitness evaluations ($MaxFEs$) is reached. The complexity is estimated as $O(MaxFEs.(n + N(f + D + N^2)))$, in which $O(f)$ is the time complexity of evaluating the fitness function, D is the dimensionality of the solution vector $\boldsymbol{x}$, and the term $O(N^2)$ is due to the quick sort algorithm. DEPA$_2$ extends the principles of difference of vectors and success-based parameter adaptation mechanisms

into a formulation where solutions are rendered by current-to-pbest-like sampling, and where memory-based adaptation enables to track useful trial vectors and parameters that lead to fitness improvements. Compared to related works in archive-based self-adaptive DE [104], [108], [109], [116], DEPA_2 extends the sampling mechanism of new solutions by perturbing the difference to the pbest vectors, and by considering the archive of successful trial vectors that lead to successful mutations through (41) - (46). Compared with parameter adaptation mechanisms [108], [109], DEPA_2 logs the successful trial vectors $\boldsymbol{u}_{i,g}$ into the archive A, and distinguishes the source of exploration in the difference $\boldsymbol{x}_i^{r_1} - \boldsymbol{x}_i^{r_2}$. Also, DEPA_2 distinguishes between the archive $\mathcal{A}$ and the population $\mathcal{P}$ to render the referent vectors $\boldsymbol{x}_{i,g}^{r_1}$ and $\boldsymbol{x}_{i,g}^{r_2}$, whereas the existing approaches use the union of $\mathcal{P} \cup \mathcal{A}$ to render a potential reference vector. Furthermore, compared to successful parent selection schemes [57], [58], [79], [104], DEPA_2 extends not only the adaptation mechanisms through memory schemes to enable the tracking of successful trial solution vectors and sampling parameters F_i and CR_i, but also extends the current-to-pbest strategy at (43) to consider adaptive reference vectors $\boldsymbol{x}_{i,g}^{r_1}$, $\boldsymbol{x}_{i,g}^{r_2}$ from population and archive. In the next section, we rigorously compare the performance of the proposed DEPA_2 algorithm to the related works in the context of CEC 2017 benchmark set.

To show the performance of the proposed/studied algorithms in the CEC 2017 benchmark set [120], for simplicity and clarity of exposition, we labeled here the objective functions (and problems) with $f_1, f_3, f_4, \ldots, f_{30}$,[1] in which f_i implies the mex function $\mathsf{cec17\_func}(\boldsymbol{x}, i)$ for variable $\boldsymbol{x}$ and $i \in [1, 30]$. We used the original implementations from [120], which extends the mex-based compilations of 30 objective functions in Matlab. Furthermore, we used dimensions $D = \{10,\ 30,\ 50\}$ to evaluate the performance and scalability of the evaluated algorithms. And, following the general settings of the benchmark suite, the maximum number of function evaluations is set as $D \times 10^4$. For fairness of comparisons, we used the class of algorithms extending the adaptation mechanisms, population size $N = 50$ and 30 independent runs for each algorithm. Other parameters for DEPA_2 include $Q = 1$, and $p = 0.05$, which are set to induce a high selection pressure during sampling.

In order to show the convergence performance of the proposed/studied algorithms, Table 24 - Table 26 show the convergence features of all studied functions. Here, the x-axis show the number of functions evaluations, whereas the y-axis shows the value of the mean fitness function over 30 independent runs. Labels (colors) of each algorithm are shown at the bottom side of each plot in Table 24 - Table 26. Thus, by observing the convergence figures in Table 24 - Table 26, we can note the following facts:

- Table 24 shows that convergence in problems with dimensions $D = 10$ occur in the order of 2×10^4 - 4×10^4 function evaluations, whereas, the convergence behaviour of problems of higher dimensions ($D = 30$ and $D = 50$) occur in the range 1×10^5 - 2×10^5 evaluations.
- Overall dimensions, there exists a subset of problems which require small number of function evaluations for convergence, such as f_4, f_{15}, f_{19} and f_{28}.
- Overall dimensions, all algorithms are able to converge to the basins of the landscapes within the allocated number of function evaluations.

To outline the statistical performance of the evaluated algorithms across the objective functions, Table 27 - Table 29 show the summary of mean converged error values and standard deviations over independent runs. Here, the symbol ▲/▼/◆ shows DEPA_2 is significantly better, worse or similar to the algorithm in the column by considering the Wilcoxon pairwise statistical comparative tests over independent runs. For each function, the mean and standard deviation is provided, and the bottom part of the grid shows the count of the number of problems in which DEPA_2 is significantly better, worse or similar when compared to other algorithms. By observing the results from Table 27 - Table 29, we observe the following facts:

- On problems with $D = 10$ dimensions, the counts on instances depicted by ▲ show that DEPA_2 either outperforms or performs equally well compared to most algorithms overall problems. On the other hand, the counts depicted by ▼ show that DEPA_2 underperforms in low number of cases (e.g. 9 cases when compared to $\mathsf{DESIM}_\mathsf{A}$).
- The results on $D = 30$ and $D = 50$ show a similar trend to the above-mentioned case, in which DEPA_2 shows the larger number of cases of outperformance for most of the problems.

To evaluate the comparative performance across all problem instances, we performed pair-wise statistical comparisons among all algorithms instances by using the Wilcoxon rank-sum test at 5% significance level. Table 9 shows the summary of the statistical comparisons overall algorithms and problem dimensions. Following the same principle described in section IV-H, Table 9 shows whether a heuristic performs significantly better (bars in blue), similarly to (bars in cyan), or worse than (bars in yellow) other algorithms across all problem instances and dimensions. As such, the x-axis of each figure in Table 9 presents the algorithm instance, ordered by best rank from left to right, and the y-axis shows the count of the number of cases. By observing the comparative results from Table 9, we can observe that DEPA_2 performs better than other algorithms in 182, 215 and 235 problem instances for $D = 10$, $D = 30$ and $D = 50$, respectively, achieving the highest rank overall problem instances. Conversely, PSO_A underperforms in 318, 319 and 317 problem instances for each respective dimension, achieving the lowest rank among all algorithms. By observing the results overall dimensions, DEPA_2 shows the attractive performance over most problem instances,

[1] CEC'17 benchmark suite disabled the use of function f_2.

TABLE 24. Convergence performance using parameter adaptation in CEC 2017 benchmark, $D = 10$.

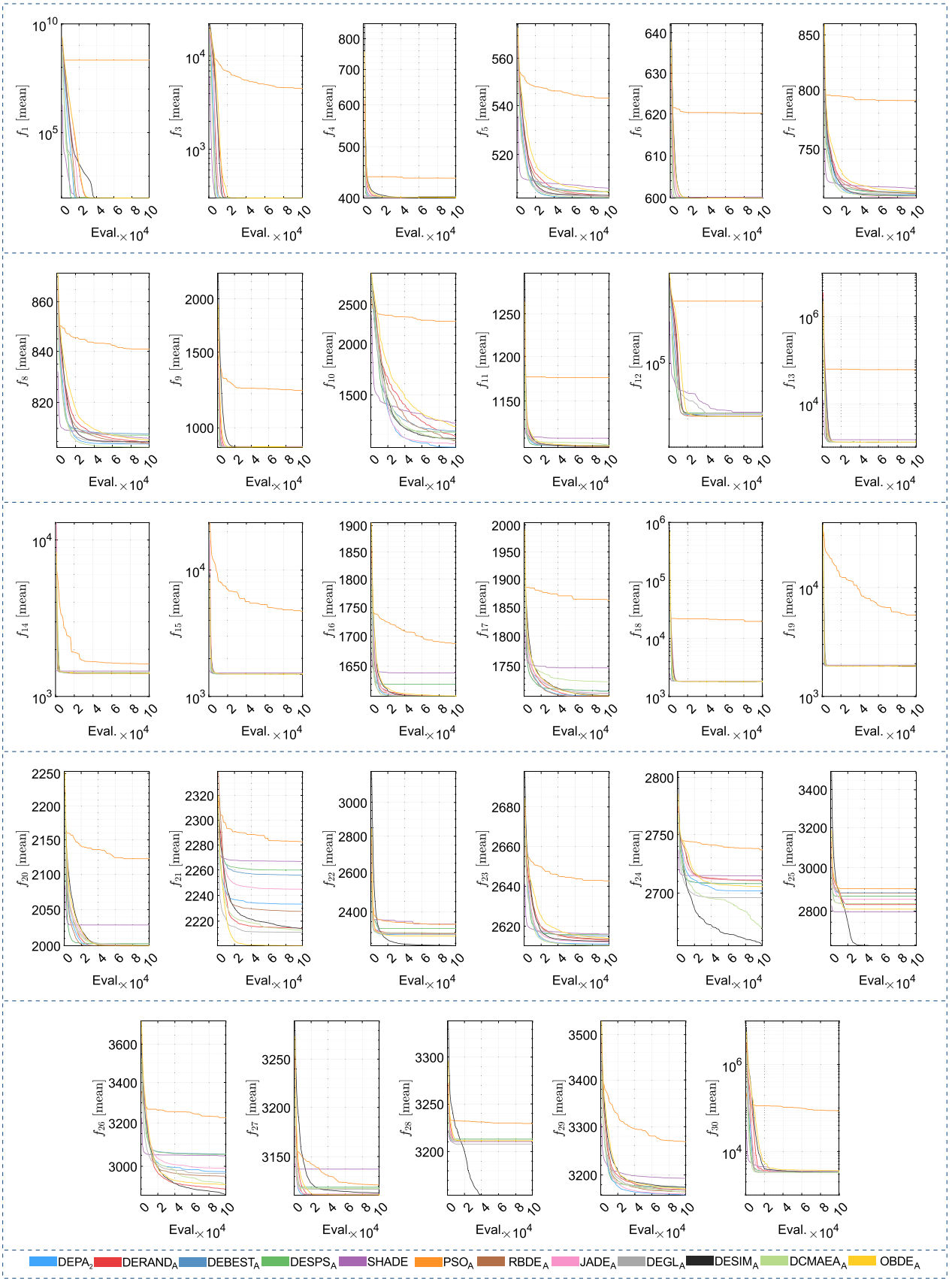


TABLE 25. Convergence performance using parameter adaptation in CEC 2017 benchmark, $D = 30$.

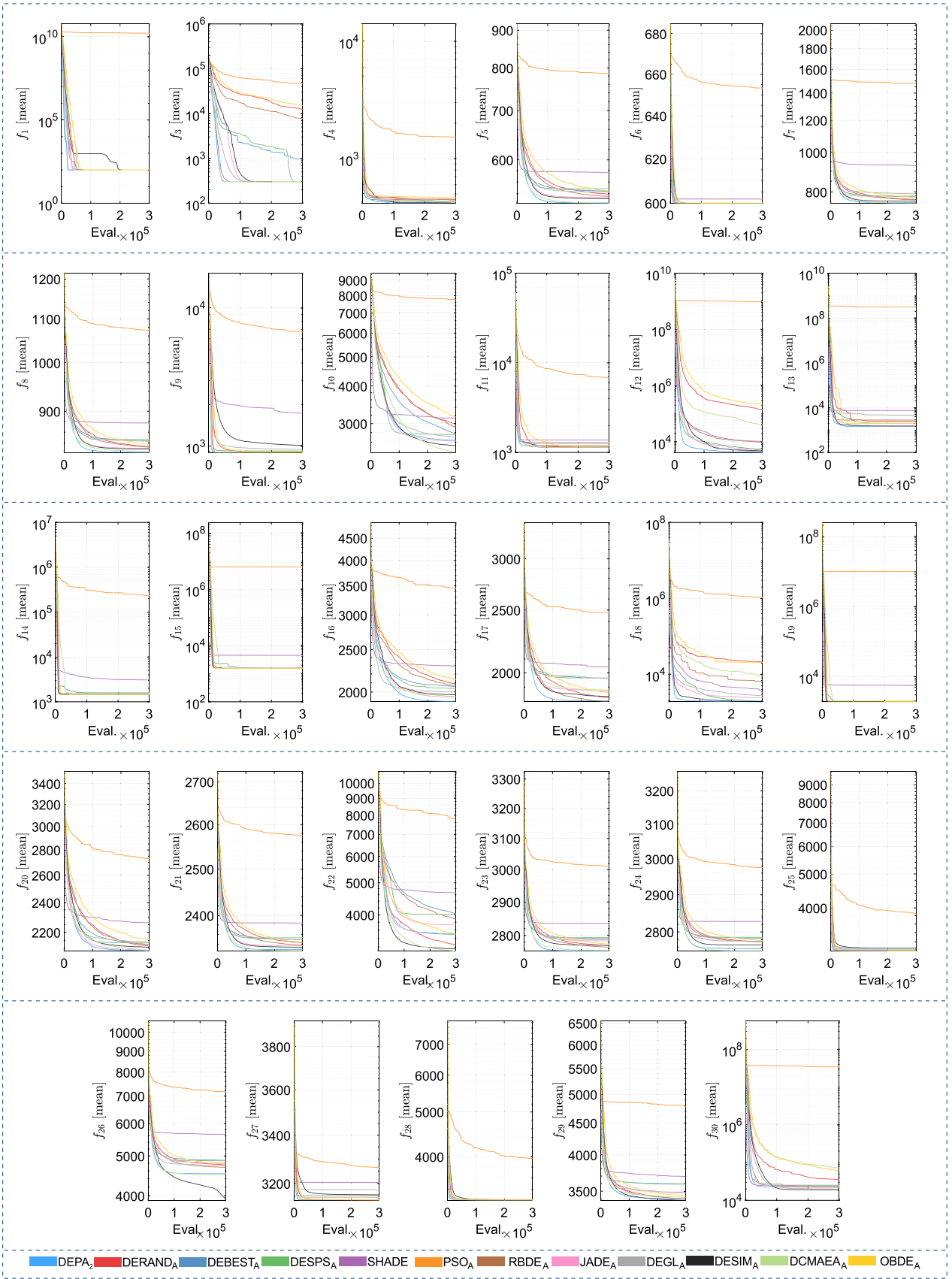


TABLE 26. Convergence performance using parameter adaptation in CEC 2017 benchmark, $D = 50$.

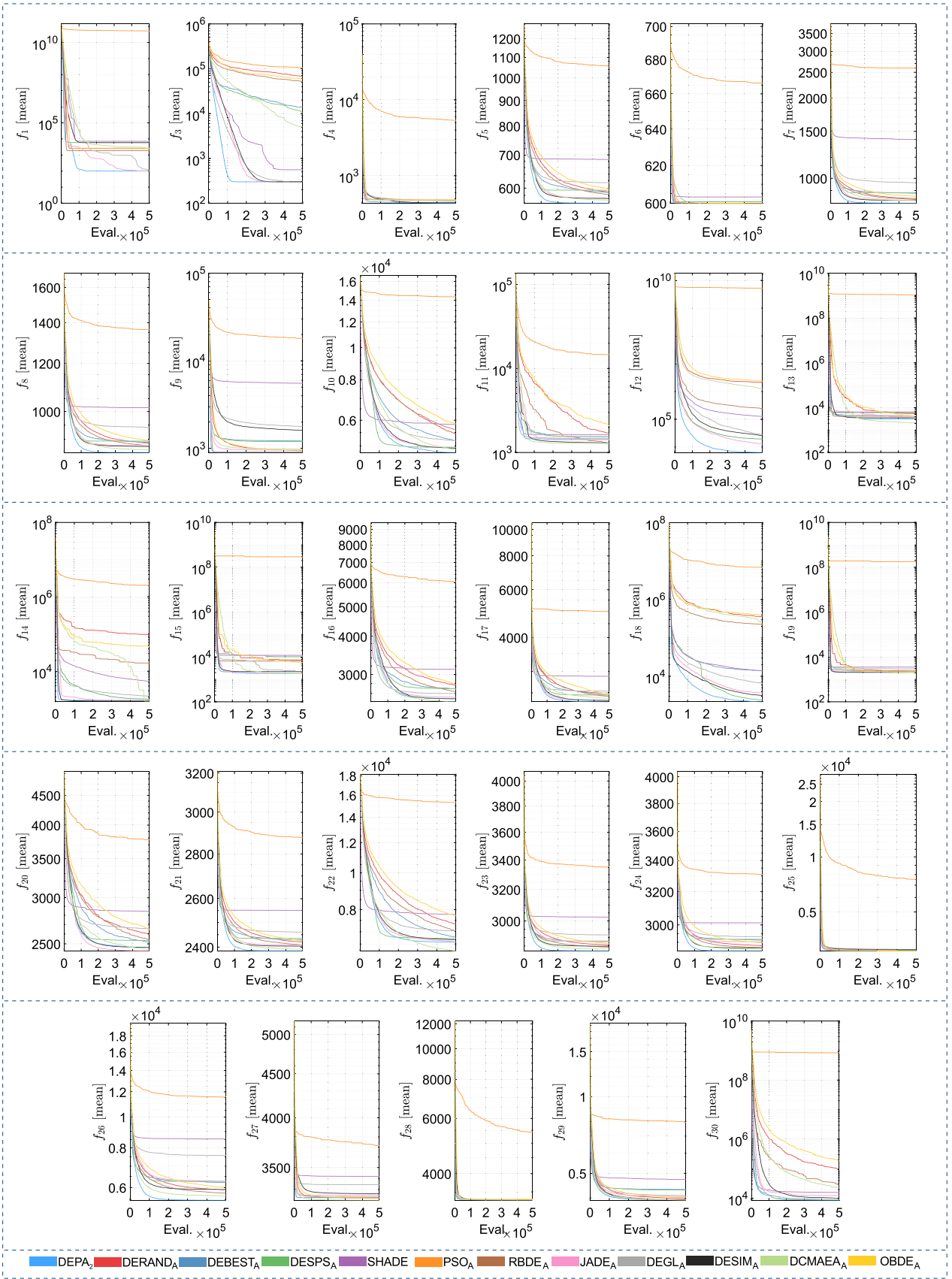


TABLE 27. Results of computational experiments on CEC 2017 benchmark functions for $D = 10$.

Fun.	$DEPA_2$	$DERAND_A$	$DEBEST_A$	$DESPS_A$	$SHADE_A$	PSO_A
f_1	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00 ♦	0.00E+00 ± 0.00E+00 ♦	0.00E+00 ± 0.00E+00 ♦	3.63E-12 ± 1.80E-11 ▲	2.16E+08 ± 2.96E+08 ▲
f_3	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00 ♦	0.00E+00 ± 0.00E+00 ♦	0.00E+00 ± 0.00E+00 ♦	2.67E-13 ± 9.81E-13 ♦	4.20E+03 ± 1.20E+03 ▲
f_4	5.32E-01 ± 1.38E+00	0.00E+00 ± 0.00E+00 ▼	1.33E+00 ± 1.91E+00 ♦	1.33E+00 ± 1.91E+00 ♦	1.20E+00 ± 1.86E+00 ▲	3.59E+01 ± 2.51E+01 ▲
f_5	3.04E+00 ± 1.09E+00	3.91E+00 ± 1.04E+00 ▲	5.40E+00 ± 4.24E+00 ♦	5.54E+00 ± 4.00E+00 ♦	6.84E+00 ± 5.68E+00 ▲	4.32E+01 ± 5.26E+00 ▲
f_6	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00 ♦	4.52E-06 ± 1.94E-05 ▲	4.52E-06 ± 1.94E-05 ▲	2.06E-01 ± 5.12E-01 ▲	2.02E+01 ± 9.13E+00 ▲
f_7	1.32E+01 ± 1.10E+00	1.43E+01 ± 9.55E-01 ▲	1.52E+01 ± 4.98E+00 ♦	1.52E+01 ± 5.17E+00 ♦	1.82E+01 ± 8.05E+00 ▲	9.09E+01 ± 2.03E+01 ▲
f_8	3.96E+00 ± 1.25E+00	4.44E+00 ± 1.07E+00 ♦	7.90E+00 ± 5.48E+00 ▲	7.36E+00 ± 5.10E+00 ▲	6.24E+00 ± 3.65E+00 ▲	4.09E+01 ± 6.69E+00 ▲
f_9	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00 ♦	2.98E-03 ± 1.63E-02 ♦	2.98E-03 ± 1.63E-02 ♦	2.92E+00 ± 1.05E+01 ▲	3.22E+02 ± 1.66E+02 ▲
f_{10}	1.17E+02 ± 8.68E+01	1.87E+02 ± 6.99E+01 ▲	2.23E+02 ± 1.71E+02 ▲	2.16E+02 ± 1.72E+02 ▲	2.79E+02 ± 2.24E+02 ▲	1.27E+03 ± 2.11E+02 ▲
f_{11}	4.28E-01 ± 7.40E-01	1.08E+00 ± 8.04E-01 ▲	2.38E+00 ± 2.46E+00 ▲	2.38E+00 ± 2.46E+00 ▲	9.92E+00 ± 1.60E+01 ▲	7.64E+01 ± 3.64E+01 ▲
f_{12}	2.67E+02 ± 1.40E+02	6.07E+01 ± 9.46E+01 ▼	4.40E+02 ± 2.44E+02 ▲	4.40E+02 ± 2.44E+02 ▲	5.42E+02 ± 3.22E+02 ▲	1.60E+07 ± 1.83E+07 ▲
f_{13}	2.99E+00 ± 2.28E+00	5.16E+00 ± 2.05E+00 ▲	8.16E+00 ± 3.67E+00 ▲	8.16E+00 ± 3.67E+00 ▲	1.84E+02 ± 2.13E+02 ▲	5.92E+04 ± 4.84E+04 ▲
f_{14}	1.14E+00 ± 3.82E+00	1.42E+00 ± 1.00E+00 ▲	1.44E+01 ± 9.90E+00 ▲	1.37E+01 ± 1.00E+01 ▲	5.21E+01 ± 4.51E+01 ▲	2.08E+02 ± 1.01E+02 ▲
f_{15}	4.00E-01 ± 3.37E-01	9.21E-01 ± 6.04E-01 ▲	2.49E+00 ± 2.08E+00 ▲	2.49E+00 ± 2.08E+00 ▲	3.38E+01 ± 3.07E+01 ▲	3.23E+03 ± 2.42E+03 ▲
f_{16}	6.07E-01 ± 2.05E+00	6.51E-01 ± 1.04E+00 ▲	2.01E+01 ± 3.59E+01 ▲	2.01E+01 ± 3.59E+01 ▲	3.85E+01 ± 5.53E+01 ▲	8.72E+01 ± 4.38E+01 ▲
f_{17}	1.43E+00 ± 5.14E+00	3.96E-01 ± 3.46E-01 ▲	9.04E+00 ± 2.31E+01 ▲	8.41E+00 ± 2.30E+01 ▲	4.68E+01 ± 6.08E+01 ▲	1.62E+02 ± 8.08E+01 ▲
f_{18}	5.12E+00 ± 8.42E+00	4.54E-01 ± 5.20E-01 ▼	1.18E+01 ± 1.04E+01 ▲	1.18E+01 ± 1.04E+01 ▲	4.13E+01 ± 1.60E+01 ▲	1.79E+04 ± 1.18E+04 ▲
f_{19}	8.66E-02 ± 2.50E-01	3.71E-02 ± 5.15E-02 ♦	1.09E+00 ± 7.43E-01 ▲	1.08E+00 ± 7.43E-01 ▲	3.73E+01 ± 5.79E+01 ▲	3.68E+03 ± 6.21E+03 ▲
f_{20}	2.08E-02 ± 7.92E-02	0.00E+00 ± 0.00E+00 ♦	2.75E+00 ± 6.04E+00 ▲	2.75E+00 ± 6.04E+00 ▲	2.85E+01 ± 4.84E+01 ▲	1.22E+02 ± 4.08E+01 ▲
f_{21}	1.33E+02 ± 5.17E+01	1.14E+02 ± 3.35E+01 ♦	1.56E+02 ± 5.71E+01 ♦	1.60E+02 ± 5.73E+01 ♦	1.67E+02 ± 5.97E+01 ▲	1.83E+02 ± 6.30E+01 ▲
f_{22}	9.13E+01 ± 2.70E+01	9.84E+01 ± 9.57E+00 ▲	9.61E+01 ± 2.07E+01 ▲	1.19E+02 ± 1.31E+02 ▲	1.41E+02 ± 1.82E+02 ▲	1.39E+02 ± 3.73E+01 ▲
f_{23}	3.11E+02 ± 9.08E-01	3.13E+02 ± 1.49E+00 ▲	3.15E+02 ± 5.14E+00 ▲	3.16E+02 ± 5.07E+00 ▲	3.16E+02 ± 4.25E+00 ▲	3.43E+02 ± 6.30E+00 ▲
f_{24}	3.02E+02 ± 3.80E+01	3.11E+02 ± 1.98E+00 ▲	3.08E+02 ± 3.97E+01 ▲	3.08E+02 ± 3.97E+01 ▲	3.15E+02 ± 3.97E+00 ▲	3.37E+02 ± 1.32E+01 ▲
f_{25}	3.82E+02 ± 7.75E+01	3.29E+02 ± 1.28E+02 ▼	3.67E+02 ± 1.07E+02 ♦	3.67E+02 ± 1.07E+02 ♦	2.96E+02 ± 1.52E+02 ♦	4.01E+02 ± 9.84E+01 ▲
f_{26}	3.76E+02 ± 1.14E+02	3.00E+02 ± 1.14E+02 ♦	4.55E+02 ± 1.26E+02 ▲	4.54E+02 ± 1.25E+02 ▲	4.47E+02 ± 1.30E+02 ▲	6.26E+02 ± 2.37E+02 ▲
f_{27}	4.12E+02 ± 6.73E-01	4.11E+02 ± 8.52E-01 ▼	4.19E+02 ± 2.79E+01 ▲	4.19E+02 ± 2.79E+01 ▲	4.37E+02 ± 3.26E+01 ▲	4.21E+02 ± 4.92E+00 ▲
f_{28}	4.11E+02 ± 2.89E-13	4.11E+02 ± 2.89E-13 ♦	4.13E+02 ± 8.07E+00 ♦	4.13E+02 ± 8.07E+00 ♦	4.10E+02 ± 2.27E+01 ▲	4.29E+02 ± 1.16E+01 ▲
f_{29}	2.59E+02 ± 5.87E+00	2.70E+02 ± 7.81E+00 ▲	2.75E+02 ± 2.20E+01 ▲	2.75E+02 ± 2.25E+01 ▲	2.93E+02 ± 3.24E+01 ▲	3.68E+02 ± 5.60E+01 ▲
f_{30}	3.21E+02 ± 3.00E+01	3.65E+02 ± 1.74E+02 ♦	3.94E+02 ± 8.75E+01 ▲	3.94E+02 ± 8.75E+01 ▲	5.59E+02 ± 2.87E+02 ▲	8.36E+04 ± 7.36E+04 ▲
▲	Significantly better	13	20	20	27	29
▼	Significantly worse	5	0	0	0	0
♦	=	11	9	9	2	0

Fun.	$RBDE_A$	$JADE_A$	$DEGL_A$	$DESIM_A$	$DCMAEA_A$	$OBDE_A$
f_1	0.00E+00 ± 0.00E+00 ♦	0.00E+00 ± 0.00E+00 ♦	0.00E+00 ± 0.00E+00 ♦	0.00E+00 ± 0.00E+00 ♦	0.00E+00 ± 0.00E+00 ♦	0.00E+00 ± 0.00E+00 ♦
f_3	0.00E+00 ± 0.00E+00 ♦	0.00E+00 ± 0.00E+00 ♦	0.00E+00 ± 0.00E+00 ♦	0.00E+00 ± 0.00E+00 ♦	0.00E+00 ± 0.00E+00 ♦	0.00E+00 ± 0.00E+00 ♦
f_4	0.00E+00 ± 0.00E+00 ▼	1.33E-01 ± 7.28E-01 ♦	1.46E+00 ± 1.95E+00 ▲	0.00E+00 ± 0.00E+00 ▼	6.64E-01 ± 1.51E+00 ♦	8.05E-01 ± 5.48E-01 ▲
f_5	3.37E+00 ± 9.31E-01 ♦	4.01E+00 ± 1.33E+00 ▲	3.49E+00 ± 1.38E+00 ♦	4.06E+00 ± 1.38E+00 ▲	3.15E+00 ± 1.06E+00 ♦	5.21E+00 ± 1.40E+00 ▲
f_6	0.00E+00 ± 0.00E+00 ♦	0.00E+00 ± 0.00E+00 ♦	7.01E-13 ± 3.84E-12 ♦	0.00E+00 ± 0.00E+00 ♦	0.00E+00 ± 0.00E+00 ♦	0.00E+00 ± 0.00E+00 ♦
f_7	1.42E+01 ± 7.27E-01 ▲	1.09E+01 ± 3.77E+00 ♦	1.38E+01 ± 9.81E-01 ▲	1.24E+01 ± 3.17E+00 ♦	1.22E+01 ± 7.16E-01 ▼	1.56E+01 ± 1.07E+00 ▲
f_8	4.31E+00 ± 1.84E+00 ♦	4.86E+00 ± 1.45E+00 ▲	4.19E+00 ± 1.82E+00 ♦	4.89E+00 ± 1.42E+00 ▲	2.84E+00 ± 1.38E+00 ▼	5.43E+00 ± 1.23E+00 ▲
f_9	0.00E+00 ± 0.00E+00 ♦	0.00E+00 ± 0.00E+00 ♦	0.00E+00 ± 0.00E+00 ♦	0.00E+00 ± 0.00E+00 ♦	0.00E+00 ± 0.00E+00 ♦	0.00E+00 ± 0.00E+00 ♦
f_{10}	1.56E+02 ± 7.43E+01 ♦	1.38E+02 ± 9.49E+01 ♦	1.61E+02 ± 1.05E+02 ♦	1.74E+02 ± 9.44E+01 ▲	1.63E+02 ± 9.70E+01 ▲	2.53E+02 ± 9.67E+01 ▲
f_{11}	4.34E-01 ± 5.73E-01 ♦	1.64E+00 ± 8.92E-01 ▲	2.02E+00 ± 1.26E+00 ▲	1.07E+00 ± 7.70E-01 ▲	2.51E+00 ± 1.74E+00 ▲	1.61E+00 ± 7.39E-01 ▲
f_{12}	8.28E+01 ± 1.11E+02 ▼	3.15E+02 ± 1.38E+02 ♦	4.49E+02 ± 2.01E+02 ▲	7.21E+01 ± 8.09E+01 ▼	3.61E+02 ± 1.87E+02 ♦	4.42E+01 ± 6.93E+01 ▼
f_{13}	5.09E+00 ± 1.71E+00 ▲	2.98E+00 ± 2.10E+00 ♦	3.69E+00 ± 3.99E+00 ♦	1.76E+00 ± 2.16E+00 ▼	9.86E+00 ± 1.33E+01 ▲	4.31E+00 ± 2.30E+00 ▲
f_{14}	1.37E+00 ± 1.26E+00 ▲	6.67E+00 ± 9.32E+00 ▲	9.70E+00 ± 9.45E+00 ▲	1.69E+00 ± 1.67E+00 ▲	2.30E+01 ± 7.48E+00 ▲	1.42E+00 ± 1.10E+00 ▲
f_{15}	9.14E-01 ± 6.30E-01 ▲	6.10E-01 ± 6.14E-01 ♦	2.20E+00 ± 2.44E+00 ▲	8.69E-01 ± 5.02E-01 ▲	6.18E+00 ± 5.58E+00 ▲	9.19E-01 ± 7.46E-01 ▲
f_{16}	6.51E-01 ± 2.06E+00 ♦	4.44E-01 ± 3.04E-01 ▲	1.91E-01 ± 1.42E-01 ♦	3.46E-01 ± 1.78E-01 ▲	2.65E-01 ± 1.45E-01 ♦	9.80E-01 ± 1.19E+00 ▲
f_{17}	1.22E-01 ± 1.73E-01 ♦	5.29E+00 ± 7.82E+00 ▲	4.56E+00 ± 4.95E+00 ▲	1.50E+00 ± 1.56E+00 ▲	2.36E+01 ± 2.35E+01 ▲	8.49E-01 ± 4.70E-01 ▲
f_{18}	4.25E-01 ± 5.56E-01 ▼	4.45E+00 ± 7.99E+00 ♦	1.29E+01 ± 1.10E+01 ▲	8.17E-02 ± 1.39E-01 ▼	2.07E+01 ± 1.15E+01 ▲	1.98E-01 ± 3.63E-01 ▼
f_{19}	1.94E-02 ± 2.80E-02 ♦	4.36E-02 ± 2.99E-02 ▲	5.40E-01 ± 1.03E+00 ▲	4.03E-02 ± 1.90E-02 ▲	3.28E+00 ± 4.23E+00 ▲	7.55E-02 ± 6.00E-02 ▲
f_{20}	4.16E-02 ± 1.08E-01 ♦	6.67E-01 ± 3.65E+00 ▲	1.03E+00 ± 3.61E+00 ▲	0.00E+00 ± 0.00E+00 ♦	2.19E-01 ± 3.38E-01 ▲	0.00E+00 ± 0.00E+00 ♦
f_{21}	1.28E+02 ± 4.65E+01 ♦	1.45E+02 ± 5.51E+01 ♦	1.11E+02 ± 3.40E+01 ♦	1.14E+02 ± 2.08E+01 ▲	1.13E+02 ± 3.29E+01 ♦	1.01E+02 ± 1.83E+00 ♦
f_{22}	9.39E+01 ± 2.34E+01 ♦	9.68E+01 ± 1.78E+01 ▲	9.06E+01 ± 2.62E+01 ▲	3.90E+01 ± 3.03E+01 ▼	9.84E+01 ± 8.66E+00 ♦	8.34E+01 ± 3.79E+01 ♦
f_{23}	3.14E+02 ± 3.30E+00 ▲	3.12E+02 ± 1.15E+00 ▲	3.11E+02 ± 4.02E+00 ♦	3.13E+02 ± 1.19E+00 ▲	3.10E+02 ± 2.04E+00 ▼	3.14E+02 ± 1.86E+00 ▲
f_{24}	3.11E+02 ± 2.21E+00 ▲	3.11E+02 ± 1.64E+00 ▲	2.96E+02 ± 5.33E+01 ▲	2.56E+02 ± 6.69E+01 ♦	2.70E+02 ± 7.20E+01 ▼	3.05E+02 ± 3.49E+01 ▲
f_{25}	3.81E+02 ± 7.68E+01 ♦	3.53E+02 ± 1.15E+02 ♦	3.33E+02 ± 1.31E+02 ♦	1.48E+02 ± 1.02E+02 ▼	3.33E+02 ± 1.43E+02 ▲	3.08E+02 ± 1.38E+02 ▼
f_{26}	3.56E+02 ± 1.29E+02 ♦	3.91E+02 ± 1.20E+02 ♦	3.68E+02 ± 1.28E+02 ♦	2.74E+02 ± 6.36E+01 ▼	3.18E+02 ± 1.27E+02 ♦	3.19E+02 ± 1.26E+02 ♦
f_{27}	4.12E+02 ± 8.43E-01 ♦	4.12E+02 ± 8.08E-01 ▲	4.17E+02 ± 5.67E+00 ▲	4.14E+02 ± 2.39E+00 ▲	4.18E+02 ± 6.84E+00 ▲	4.12E+02 ± 9.02E-01 ♦
f_{28}	4.11E+02 ± 2.89E-13 ♦	4.11E+02 ± 3.38E-01 ♦	4.08E+02 ± 2.03E+01 ♦	3.55E+02 ± 9.51E+01 ▼	4.11E+02 ± 2.89E-13 ♦	4.11E+02 ± 2.89E-13 ♦
f_{29}	2.66E+02 ± 8.14E+00 ▲	2.61E+02 ± 7.93E+00 ♦	2.69E+02 ± 8.94E+00 ▲	2.75E+02 ± 9.92E+00 ▲	2.64E+02 ± 6.87E+00 ▲	2.72E+02 ± 7.59E+00 ▲
f_{30}	3.25E+02 ± 7.50E+01 ♦	3.23E+02 ± 3.83E+01 ♦	3.05E+02 ± 4.88E+01 ♦	3.18E+02 ± 2.68E+02 ▼	2.92E+02 ± 4.33E+01 ▼	6.32E+02 ± 8.34E+02 ♦
▲	7	12	14	13	12	15
▼	3	0	0	9	5	3
♦	19	17	15	7	12	11

TABLE 28. Results of computational experiments on CEC 2017 benchmark functions for $D = 30$.

Fun.	$DEPA_2$	$DERAND_A$	$DEBEST_A$	$DESPS_A$	$SHADE_A$	PSO_A
f_1	0.00E+00 ± 0.00E+00 ♦	0.00E+00 ± 0.00E+00 ♦	3.17E-12 ± 1.33E-11 ▲	3.10E-12 ± 1.33E-11 ▲	5.67E-09 ± 2.20E-08 ▲	1.64E+10 ± 3.27E+09 ▲
f_3	6.82E-14 ± 2.60E-13	1.23E+04 ± 1.93E+04 ▲	5.50E+02 ± 3.01E+03 ▲	2.39E-05 ± 1.31E-04 ▲	3.06E-11 ± 4.51E-11 ▲	4.55E+04 ± 7.96E+03 ▲
f_4	1.85E+01 ± 2.53E+01	4.92E+01 ± 3.06E+01 ▲	3.01E+01 ± 2.65E+01 ▲	3.01E+01 ± 2.65E+01 ▲	3.26E+01 ± 3.27E+01 ▲	1.13E+03 ± 2.43E+02 ▲
f_5	2.46E+01 ± 4.66E+00	3.89E+01 ± 5.81E+00 ▲	4.43E+01 ± 1.79E+01 ▲	4.82E+01 ± 1.71E+01 ▲	7.64E+01 ± 2.19E+01 ▲	2.86E+02 ± 2.00E+01 ▲
f_6	4.60E-06 ± 1.15E-05	0.00E+00 ± 0.00E+00 ▼	5.21E-02 ± 7.18E-02 ▲	5.21E-02 ± 7.18E-02 ▲	1.86E+00 ± 5.28E+00 ▲	5.33E+01 ± 4.41E+00 ▲
f_7	5.30E+01 ± 4.30E+00	6.70E+01 ± 4.67E+00 ▲	7.78E+01 ± 2.29E+01 ▲	8.11E+01 ± 2.18E+01 ▲	2.28E+02 ± 7.60E+01 ▲	7.79E+02 ± 1.23E+02 ▲
f_8	2.28E+01 ± 4.80E+00	3.36E+01 ± 6.03E+00 ▲	4.42E+01 ± 1.95E+01 ▲	4.65E+01 ± 1.67E+01 ▲	7.74E+01 ± 3.00E+01 ▲	2.72E+02 ± 1.68E+01 ▲
f_9	4.98E-01 ± 5.74E-01	1.33E+00 ± 1.44E+00 ♦	2.34E+01 ± 2.83E+01 ▲	2.26E+01 ± 2.86E+01 ▲	8.35E+02 ± 6.58E+02 ▲	5.80E+03 ± 7.71E+02 ▲
f_{10}	1.64E+03 ± 2.45E+02	1.92E+03 ± 2.20E+02 ▲	1.76E+03 ± 4.63E+02 ♦	1.76E+03 ± 6.05E+02 ♦	2.13E+03 ± 3.87E+02 ▲	6.76E+03 ± 4.53E+02 ▲
f_{11}	9.37E+01 ± 5.03E+01	4.61E+01 ± 2.60E+01 ▼	1.99E+02 ± 9.04E+01 ▲	1.99E+02 ± 9.04E+01 ▲	2.75E+02 ± 1.21E+02 ▲	5.79E+03 ± 1.00E+03 ▲
f_{12}	2.94E+03 ± 2.12E+03	1.41E+05 ± 2.13E+05 ▲	4.22E+03 ± 2.87E+03 ▲	4.22E+03 ± 2.87E+03 ▲	9.13E+03 ± 6.88E+03 ▲	9.71E+08 ± 6.34E+08 ▲
f_{13}	1.40E+02 ± 1.13E+02	1.14E+03 ± 1.55E+03 ▲	7.16E+02 ± 9.05E+02 ▲	7.16E+02 ± 9.05E+02 ▲	6.19E+03 ± 5.95E+03 ▲	3.09E+08 ± 3.93E+08 ▲
f_{14}	9.19E+01 ± 3.93E+01	4.03E+01 ± 1.44E+01 ▼	1.84E+02 ± 6.93E+01 ▲	1.84E+02 ± 6.93E+01 ▲	1.67E+03 ± 1.65E+03 ▲	2.34E+05 ± 1.20E+05 ▲
f_{15}	7.59E+01 ± 3.38E+01	2.93E+01 ± 1.85E+01 ▼	1.04E+02 ± 4.18E+01 ▲	1.04E+02 ± 4.18E+01 ▲	2.97E+03 ± 4.44E+03 ▲	6.19E+06 ± 1.03E+07 ▲
f_{16}	3.00E+02 ± 1.34E+02	4.87E+02 ± 1.33E+02 ▲	4.58E+02 ± 2.45E+02 ▲	4.37E+02 ± 2.57E+02 ▲	6.94E+02 ± 1.88E+02 ▲	1.86E+03 ± 2.54E+02 ▲
f_{17}	1.10E+02 ± 8.08E+01	1.29E+02 ± 7.06E+01 ♦	2.61E+02 ± 1.21E+02 ▲	2.62E+02 ± 1.20E+02 ▲	3.44E+02 ± 1.55E+02 ▲	7.78E+02 ± 1.69E+02 ▲
f_{18}	8.04E+01 ± 4.28E+01	1.94E+04 ± 3.42E+04 ▲	1.47E+02 ± 6.23E+01 ▲	1.47E+02 ± 6.23E+01 ▲	1.54E+03 ± 3.14E+03 ▲	1.05E+06 ± 4.24E+05 ▲
f_{19}	5.33E+01 ± 2.04E+01	2.02E+01 ± 7.39E+00 ▼	8.62E+01 ± 2.86E+01 ▲	8.62E+01 ± 2.86E+01 ▲	3.71E+03 ± 4.38E+03 ▲	9.86E+06 ± 7.47E+06 ▲
f_{20}	8.26E+01 ± 3.45E+01	1.12E+02 ± 3.50E+01 ▲	1.20E+02 ± 7.59E+01 ▲	1.33E+02 ± 8.65E+01 ▲	2.63E+02 ± 1.58E+02 ▲	7.30E+02 ± 7.92E+01 ▲
f_{21}	2.29E+02 ± 5.24E+00	2.43E+02 ± 6.34E+00 ▲	2.49E+02 ± 1.75E+01 ▲	2.53E+02 ± 1.43E+01 ▲	2.84E+02 ± 2.27E+01 ▲	4.75E+02 ± 1.36E+01 ▲
f_{22}	1.29E+03 ± 8.90E+02	1.05E+03 ± 1.07E+03 ♦	1.84E+03 ± 7.70E+02 ▲	1.81E+03 ± 8.19E+02 ▲	2.47E+03 ± 9.22E+02 ▲	5.61E+03 ± 2.25E+03 ▲
f_{23}	4.54E+02 ± 5.98E+00	4.70E+02 ± 7.13E+00 ▲	4.91E+02 ± 2.90E+01 ▲	4.93E+02 ± 2.90E+01 ▲	5.35E+02 ± 4.52E+01 ▲	7.10E+02 ± 1.29E+01 ▲
f_{24}	3.56E+02 ± 6.22E+00	3.74E+02 ± 8.56E+00 ▲	3.83E+02 ± 1.65E+01 ▲	3.86E+02 ± 1.56E+01 ▲	4.28E+02 ± 3.45E+01 ▲	5.76E+02 ± 1.71E+01 ▲
f_{25}	5.34E+02 ± 3.28E+01	5.18E+02 ± 1.66E+01 ♦	5.34E+02 ± 4.20E+01 ♦	5.34E+02 ± 4.20E+01 ♦	5.63E+02 ± 4.98E+01 ▲	1.35E+03 ± 1.56E+02 ▲
f_{26}	1.93E+03 ± 3.13E+02	2.16E+03 ± 6.40E+01 ▲	2.28E+03 ± 2.26E+02 ▲	2.27E+03 ± 2.25E+02 ▲	3.05E+03 ± 8.01E+02 ▲	4.57E+03 ± 5.94E+02 ▲
f_{27}	4.34E+02 ± 6.17E+00	4.33E+02 ± 4.21E+00 ♦	4.51E+02 ± 1.76E+01 ▲	4.51E+02 ± 1.76E+01 ▲	5.05E+02 ± 3.38E+01 ▲	5.65E+02 ± 3.81E+01 ▲
f_{28}	4.25E+02 ± 1.35E+01	4.22E+02 ± 1.54E+01 ♦	4.24E+02 ± 2.14E+01 ♦	4.24E+02 ± 2.14E+01 ♦	4.29E+02 ± 2.26E+01 ♦	1.16E+03 ± 1.84E+02 ▲
f_{29}	5.01E+02 ± 6.31E+01	5.01E+02 ± 6.66E+01 ♦	6.97E+02 ± 2.23E+02 ▲	6.91E+02 ± 2.25E+02 ▲	7.96E+02 ± 1.81E+02 ▲	1.88E+03 ± 2.68E+02 ▲
f_{30}	1.96E+04 ± 1.13E+04	3.25E+04 ± 2.42E+04 ♦	2.22E+04 ± 1.26E+04 ▲	2.22E+04 ± 1.26E+04 ▲	2.06E+04 ± 8.54E+03 ▲	3.40E+07 ± 2.96E+07 ▲
▲	Significantly better	15	26	26	28	29
▼	Significantly worse	5	0	0	0	0
♦	=	9	3	3	1	0

Fun.	$RBDE_A$	$JADE_A$	$DEGL_A$	$DESIM_A$	$DCMAEA_A$	$OBDE_A$
f_1	0.00E+00 ± 0.00E+00 ♦	0.00E+00 ± 0.00E+00 ♦	8.11E-11 ± 4.39E-10 ▲	2.30E-12 ± 5.46E-12 ▲	0.00E+00 ± 0.00E+00 ♦	1.11E-06 ± 5.79E-06 ▲
f_3	7.38E+03 ± 1.52E+04 ▲	1.02E-13 ± 3.12E-13 ♦	1.21E-04 ± 5.88E-04 ▲	7.66E-08 ± 4.16E-07 ▲	0.00E+00 ± 0.00E+00 ♦	1.48E+04 ± 2.53E+04 ▲
f_4	4.33E+01 ± 3.08E+01 ▲	3.53E+01 ± 2.68E+01 ▲	1.95E+01 ± 2.52E+01 ♦	1.60E+01 ± 2.66E+01 ♦	3.42E+01 ± 3.35E+01 ♦	6.48E+01 ± 2.80E+01 ▲
f_5	3.52E+01 ± 5.04E+00 ▲	3.34E+01 ± 5.58E+00 ▲	4.28E+01 ± 9.17E+00 ▲	3.15E+01 ± 6.70E+00 ▲	2.40E+01 ± 5.18E+00 ♦	4.42E+01 ± 6.40E+00 ▲
f_6	0.00E+00 ± 0.00E+00 ▼	5.48E-05 ± 1.81E-04 ▲	2.13E-02 ± 5.49E-02 ♦	0.00E+00 ± 0.00E+00 ▼	6.71E-13 ± 1.50E-12 ▼	0.00E+00 ± 0.00E+00 ▼
f_7	6.49E+01 ± 4.51E+00 ▲	5.87E+01 ± 4.96E+00 ▲	9.15E+01 ± 1.54E+01 ▲	5.92E+01 ± 5.40E+00 ▲	5.03E+01 ± 6.24E+00 ▼	7.51E+01 ± 4.71E+00 ▲
f_8	3.24E+01 ± 6.77E+00 ▲	3.01E+01 ± 6.08E+00 ▲	4.32E+01 ± 8.11E+00 ▲	2.90E+01 ± 5.00E+00 ▲	2.50E+01 ± 5.07E+00 ♦	3.81E+01 ± 4.73E+00 ▲
f_9	3.33E-02 ± 1.16E-01 ▼	1.26E+00 ± 1.57E+00 ▲	4.80E+01 ± 4.49E+01 ▲	1.15E+02 ± 6.61E+01 ▲	2.75E-02 ± 1.51E-01 ▼	3.78E+00 ± 5.16E+00 ▲
f_{10}	1.97E+03 ± 2.76E+02 ▲	1.62E+03 ± 2.16E+02 ♦	1.71E+03 ± 2.44E+02 ♦	1.54E+03 ± 2.64E+02 ♦	1.40E+03 ± 2.55E+02 ▼	2.16E+03 ± 3.20E+02 ▲
f_{11}	4.51E+01 ± 2.83E+01 ▼	1.51E+02 ± 5.37E+01 ▲	1.92E+02 ± 7.04E+01 ▲	6.94E+01 ± 2.92E+01 ♦	9.61E+01 ± 4.57E+01 ♦	5.42E+01 ± 3.86E+01 ▼
f_{12}	8.41E+03 ± 1.60E+04 ▲	3.77E+03 ± 2.32E+03 ▲	3.24E+03 ± 1.95E+03 ♦	3.48E+03 ± 2.98E+03 ♦	3.97E+04 ± 5.85E+04 ♦	2.13E+05 ± 2.73E+05 ▲
f_{13}	1.50E+03 ± 1.94E+03 ▲	1.97E+02 ± 2.58E+02 ♦	3.27E+03 ± 4.38E+03 ▲	2.26E+02 ± 1.25E+02 ▲	7.30E+02 ± 9.03E+02 ▲	6.85E+02 ± 1.24E+03 ▲
f_{14}	4.97E+01 ± 1.60E+01 ▼	1.17E+02 ± 4.88E+01 ▲	1.28E+02 ± 3.40E+01 ▲	5.84E+01 ± 1.59E+01 ▼	9.22E+01 ± 2.19E+01 ♦	4.00E+01 ± 8.52E+00 ▼
f_{15}	5.74E+01 ± 1.52E+02 ▼	7.69E+01 ± 3.47E+01 ♦	5.72E+01 ± 2.66E+01 ▼	3.25E+01 ± 1.50E+01 ▼	3.45E+01 ± 1.39E+01 ▼	1.92E+01 ± 9.21E+00 ▼
f_{16}	4.67E+02 ± 1.58E+02 ▲	3.47E+02 ± 1.71E+02 ♦	4.00E+02 ± 1.47E+02 ▲	3.72E+02 ± 1.24E+02 ♦	3.52E+02 ± 1.29E+02 ♦	5.35E+02 ± 1.41E+02 ▲
f_{17}	1.03E+02 ± 5.08E+01 ♦	1.74E+02 ± 8.76E+01 ▲	2.63E+02 ± 7.57E+01 ▲	1.39E+02 ± 8.27E+01 ♦	1.81E+02 ± 8.96E+01 ▲	1.62E+02 ± 6.70E+01 ▲
f_{18}	4.68E+03 ± 1.73E+04 ▲	6.25E+02 ± 2.29E+03 ▲	9.05E+02 ± 1.51E+03 ▲	1.17E+02 ± 9.02E+01 ♦	7.11E+03 ± 1.09E+04 ♦	1.86E+04 ± 3.08E+04 ▲
f_{19}	2.47E+01 ± 1.02E+01 ▼	5.16E+01 ± 1.97E+01 ♦	5.83E+01 ± 2.35E+01 ♦	2.33E+01 ± 1.36E+01 ▼	4.44E+01 ± 1.55E+01 ♦	1.48E+01 ± 5.28E+00 ▼
f_{20}	1.20E+02 ± 4.14E+01 ▲	8.87E+01 ± 3.32E+01 ♦	1.39E+02 ± 5.19E+01 ▲	1.06E+02 ± 4.37E+01 ▲	8.55E+01 ± 3.08E+01 ♦	1.53E+02 ± 6.39E+01 ▲
f_{21}	2.38E+02 ± 8.19E+00 ▲	2.36E+02 ± 6.62E+00 ▲	2.49E+02 ± 9.51E+00 ▲	2.34E+02 ± 6.05E+00 ▲	2.26E+02 ± 5.20E+00 ▼	2.44E+02 ± 8.86E+00 ▲
f_{22}	1.69E+03 ± 9.90E+02 ▲	1.53E+03 ± 7.59E+02 ♦	1.31E+03 ± 1.09E+03 ♦	9.58E+02 ± 1.06E+03 ♦	9.11E+02 ± 9.62E+02 ♦	1.30E+03 ± 1.26E+03 ♦
f_{23}	4.65E+02 ± 7.86E+00 ▲	4.84E+02 ± 1.33E+01 ▲	4.86E+02 ± 3.55E+01 ▲	4.70E+02 ± 1.54E+01 ▲	4.56E+02 ± 9.46E+00 ♦	4.75E+02 ± 9.36E+00 ▲
f_{24}	3.73E+02 ± 9.75E+00 ▲	3.75E+02 ± 9.40E+00 ▲	3.76E+02 ± 1.24E+01 ▲	3.65E+02 ± 9.73E+00 ▲	3.50E+02 ± 5.35E+00 ▼	3.77E+02 ± 9.15E+00 ▲
f_{25}	5.16E+02 ± 2.35E+01 ♦	5.20E+02 ± 2.56E+01 ♦	5.66E+02 ± 4.67E+01 ▲	5.68E+02 ± 6.35E+01 ▲	5.54E+02 ± 2.89E+01 ▲	5.11E+02 ± 1.83E+01 ▼
f_{26}	2.09E+03 ± 7.03E+01 ▲	2.22E+03 ± 8.65E+01 ▲	2.14E+03 ± 9.01E+02 ▲	1.30E+03 ± 6.85E+02 ▼	1.90E+03 ± 4.69E+02 ♦	2.18E+03 ± 7.75E+01 ▲
f_{27}	4.34E+02 ± 4.34E+00 ♦	4.45E+02 ± 1.24E+01 ▲	4.75E+02 ± 2.07E+01 ▲	4.55E+02 ± 1.62E+01 ▲	4.36E+02 ± 6.31E+00 ♦	4.34E+02 ± 3.89E+00 ♦
f_{28}	4.25E+02 ± 1.47E+01 ♦	4.30E+02 ± 1.88E+01 ▲	4.22E+02 ± 2.00E+01 ♦	4.22E+02 ± 1.96E+01 ♦	4.18E+02 ± 1.21E+01 ▼	4.26E+02 ± 1.32E+01 ♦
f_{29}	5.04E+02 ± 7.84E+01 ♦	5.77E+02 ± 7.93E+01 ▲	5.82E+02 ± 7.17E+01 ▲	4.84E+02 ± 7.91E+01 ♦	5.20E+02 ± 6.41E+01 ♦	5.40E+02 ± 5.60E+01 ▲
f_{30}	1.92E+04 ± 1.04E+04 ♦	2.06E+04 ± 8.72E+03 ♦	1.70E+04 ± 5.92E+03 ♦	1.59E+04 ± 5.70E+03 ▼	6.85E+04 ± 3.36E+04 ▲	5.76E+04 ± 3.75E+04 ▲
▲	16	18	20	13	4	20
▼	6	0	1	6	8	6
♦	7	11	8	10	17	3

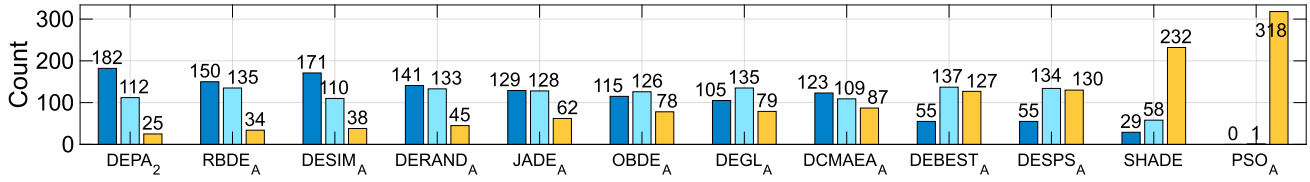
TABLE 29. Results of computational experiments on CEC 2017 benchmark functions for $D = 50$.

Fun.	$DEPA_2$	$DERAND_A$	$DEBEST_A$	$DESPS_A$	$SHADE_A$	PSO_A
f_1	3.36E-10 ± 1.11E-09	2.40E+03 ± 2.94E+03 ▲	5.79E+03 ± 4.45E+03 ▲	5.79E+03 ± 4.45E+03 ▲	7.05E+03 ± 4.60E+03 ▲	5.25E+10 ± 5.68E+09 ▲
f_3	9.19E-13 ± 4.07E-13	6.79E+04 ± 6.12E+04 ▲	1.35E+04 ± 2.76E+04 ▲	1.01E+04 ± 2.03E+04 ▲	2.58E+02 ± 6.98E+02 ▲	1.05E+05 ± 1.22E+04 ▲
f_4	3.45E+01 ± 3.60E+01	8.08E+01 ± 1.00E+01 ▲	6.90E+01 ± 3.53E+01 ▲	6.90E+01 ± 3.53E+01 ▲	8.05E+01 ± 3.97E+01 ▲	4.97E+03 ± 8.15E+02 ▲
f_5	5.94E+01 ± 1.03E+01	8.94E+01 ± 1.20E+01 ▲	8.80E+01 ± 2.20E+01 ▲	9.43E+01 ± 2.09E+01 ▲	1.85E+02 ± 3.72E+01 ▲	5.57E+02 ± 1.91E+01 ▲
f_6	3.93E-03 ± 1.10E-02	0.00E+00 ± 0.00E+00 ▼	7.94E-01 ± 1.06E+00 ▲	7.94E-01 ± 1.06E+00 ▲	3.29E+00 ± 7.69E+00 ▲	6.62E+01 ± 3.04E+00 ▲
f_7	1.06E+02 ± 1.05E+01	1.38E+02 ± 7.86E+00 ▲	1.79E+02 ± 3.52E+01 ▲	1.84E+02 ± 3.26E+01 ▲	6.99E+02 ± 2.36E+02 ▲	1.89E+03 ± 1.64E+02 ▲
f_8	5.63E+01 ± 8.22E+00	8.69E+01 ± 7.80E+00 ▲	8.96E+01 ± 2.92E+01 ▲	9.41E+01 ± 3.07E+01 ▲	2.14E+02 ± 5.51E+01 ▲	5.63E+02 ± 2.07E+01 ▲
f_9	1.45E+01 ± 1.12E+01	4.40E+01 ± 4.44E+01 ▲	3.20E+02 ± 3.58E+02 ▲	3.01E+02 ± 3.54E+02 ▲	4.64E+03 ± 1.77E+03 ▲	1.72E+04 ± 1.82E+03 ▲
f_{10}	3.78E+03 ± 3.39E+02	4.44E+03 ± 2.73E+02 ▲	4.18E+03 ± 8.45E+02 ▲	3.96E+03 ± 1.06E+03 ◆	4.82E+03 ± 4.77E+02 ▲	1.33E+04 ± 5.61E+02 ▲
f_{11}	3.51E+02 ± 1.38E+02	5.67E+02 ± 5.76E+02 ◆	4.46E+02 ± 1.45E+02 ▲	4.34E+02 ± 1.67E+02 ▲	5.34E+02 ± 5.22E+02 ◆	1.35E+04 ± 1.85E+03 ▲
f_{12}	5.15E+03 ± 3.51E+03	2.13E+06 ± 2.06E+06 ▲	1.81E+04 ± 1.17E+04 ▲	1.81E+04 ± 1.17E+04 ▲	1.24E+05 ± 1.47E+05 ▲	5.25E+09 ± 1.57E+09 ▲
f_{13}	1.86E+03 ± 2.12E+03	4.06E+03 ± 3.65E+03 ▲	4.71E+03 ± 5.03E+03 ▲	4.71E+03 ± 5.03E+03 ▲	5.12E+03 ± 3.79E+03 ▲	1.09E+09 ± 6.71E+08 ▲
f_{14}	2.56E+02 ± 7.55E+01	9.62E+04 ± 1.80E+05 ◆	3.91E+02 ± 1.28E+02 ▲	3.91E+02 ± 1.28E+02 ▲	3.95E+03 ± 3.40E+03 ▲	1.95E+06 ± 9.83E+05 ▲
f_{15}	3.24E+02 ± 1.15E+02	5.87E+03 ± 8.71E+03 ▲	8.72E+03 ± 1.36E+04 ▲	8.72E+03 ± 1.36E+04 ▲	1.03E+04 ± 1.52E+04 ▲	2.93E+08 ± 2.79E+08 ▲
f_{16}	8.99E+02 ± 1.52E+02	1.16E+03 ± 1.95E+02 ▲	1.09E+03 ± 3.21E+02 ▲	1.09E+03 ± 3.12E+02 ▲	1.52E+03 ± 3.85E+02 ▲	4.45E+03 ± 4.02E+02 ▲
f_{17}	6.17E+02 ± 1.55E+02	7.55E+02 ± 1.32E+02 ▲	7.94E+02 ± 2.80E+02 ▲	7.53E+02 ± 2.90E+02 ▲	1.18E+03 ± 2.81E+02 ▲	3.29E+03 ± 3.78E+02 ▲
f_{18}	3.59E+02 ± 2.57E+02	3.52E+05 ± 4.14E+05 ▲	1.21E+04 ± 6.19E+04 ▲	5.76E+02 ± 3.69E+02 ▲	1.20E+04 ± 1.06E+04 ▲	6.87E+06 ± 2.93E+06 ▲
f_{19}	1.31E+02 ± 4.91E+01	4.88E+02 ± 5.73E+02 ▲	2.78E+02 ± 2.69E+02 ▲	2.77E+02 ± 2.69E+02 ▲	1.61E+03 ± 1.23E+03 ▲	1.76E+08 ± 1.38E+08 ▲
f_{20}	4.65E+02 ± 1.54E+02	5.93E+02 ± 1.50E+02 ▲	5.19E+02 ± 1.77E+02 ◆	5.33E+02 ± 2.37E+02 ◆	8.44E+02 ± 2.80E+02 ▲	1.77E+03 ± 1.57E+02 ▲
f_{21}	2.84E+02 ± 1.33E+01	3.20E+02 ± 9.68E+00 ▲	3.29E+02 ± 2.58E+01 ▲	3.35E+02 ± 2.39E+01 ▲	4.51E+02 ± 5.74E+01 ▲	7.78E+02 ± 2.95E+01 ▲
f_{22}	4.37E+03 ± 5.10E+02	5.13E+03 ± 3.08E+02 ▲	4.58E+03 ± 4.82E+02 ◆	4.52E+03 ± 8.24E+02 ◆	5.57E+03 ± 5.88E+02 ▲	1.31E+04 ± 1.73E+03 ▲
f_{23}	5.20E+02 ± 1.69E+01	5.55E+02 ± 1.31E+01 ▲	5.74E+02 ± 4.18E+01 ▲	5.79E+02 ± 4.09E+01 ▲	7.24E+02 ± 6.88E+01 ▲	1.05E+03 ± 2.53E+01 ▲
f_{24}	4.51E+02 ± 1.28E+01	4.81E+02 ± 1.39E+01 ▲	5.13E+02 ± 2.81E+01 ▲	5.17E+02 ± 3.16E+01 ▲	6.12E+02 ± 5.55E+01 ▲	9.10E+02 ± 1.51E+01 ▲
f_{25}	5.84E+02 ± 4.62E+01	5.66E+02 ± 1.21E+01 ◆	5.80E+02 ± 4.10E+01 ◆	5.80E+02 ± 4.10E+01 ◆	6.10E+02 ± 5.11E+01 ▲	5.03E+03 ± 6.26E+02 ▲
f_{26}	2.87E+03 ± 1.60E+02	3.30E+03 ± 1.76E+02 ▲	3.63E+03 ± 3.69E+02 ▲	3.67E+03 ± 3.66E+02 ▲	5.93E+03 ± 1.21E+03 ▲	8.95E+03 ± 2.85E+02 ▲
f_{27}	5.28E+02 ± 1.53E+01	5.23E+02 ± 1.44E+01 ◆	5.29E+02 ± 2.21E+01 ◆	5.29E+02 ± 2.21E+01 ◆	7.19E+02 ± 9.27E+01 ▲	1.01E+03 ± 1.23E+02 ▲
f_{28}	4.99E+02 ± 2.63E+01	5.10E+02 ± 1.24E+01 ◆	5.08E+02 ± 2.60E+01 ▲	5.08E+02 ± 2.60E+01 ▲	4.91E+02 ± 3.22E+01 ◆	2.62E+03 ± 3.05E+02 ▲
f_{29}	1.03E+03 ± 1.61E+02	1.07E+03 ± 1.33E+02 ◆	1.42E+03 ± 3.74E+02 ▲	1.40E+03 ± 4.35E+02 ▲	1.83E+03 ± 4.37E+02 ▲	5.04E+03 ± 5.11E+02 ▲
f_{30}	5.47E+03 ± 8.30E+02	9.57E+04 ± 1.16E+05 ▲	5.49E+03 ± 1.81E+03 ◆	5.49E+03 ± 1.81E+03 ◆	1.27E+04 ± 5.20E+03 ▲	8.39E+08 ± 5.51E+08 ▲
▲	Significantly better	22	24	23	27	29
▼	Significantly worse	1	0	0	0	0
◆	=	6	5	6	2	0

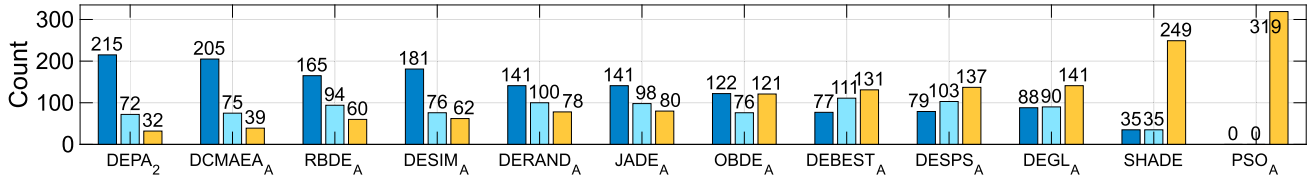
Fun.	$RBDE_A$	$JADE_A$	$DEGL_A$	$DESIM_A$	$DCMAEA_A$	$OBDE_A$
f_1	1.77E+03 ± 1.82E+03 ▲	3.61E+00 ± 1.74E+01 ▲	1.35E+01 ± 5.80E+01 ▲	5.76E+03 ± 4.85E+03 ▲	2.22E+03 ± 3.48E+03 ▲	2.58E+03 ± 2.81E+03 ▲
f_3	5.12E+04 ± 5.62E+04 ▲	2.16E-06 ± 1.16E-05 ▲	7.16E-01 ± 2.27E+00 ▲	8.21E-04 ± 1.44E-03 ▲	4.73E+03 ± 1.40E+04 ▼	5.88E+04 ± 6.94E+04 ▲
f_4	8.26E+01 ± 7.11E-13 ▲	6.85E+01 ± 2.62E+01 ▲	6.15E+01 ± 4.32E+01 ▲	4.23E+01 ± 2.73E+01 ▲	7.34E+01 ± 2.41E+01 ▲	8.08E+01 ± 9.78E+00 ▲
f_5	8.27E+01 ± 9.32E+00 ▲	7.29E+01 ± 1.12E+01 ▲	1.15E+02 ± 1.44E+01 ▲	7.19E+01 ± 1.18E+01 ▲	6.74E+01 ± 1.34E+01 ▲	9.88E+01 ± 8.66E+00 ▲
f_6	1.60E-09 ± 8.75E-09 ▼	1.96E-03 ± 4.07E-03 ◆	1.56E-03 ± 7.13E-03 ▼	0.00E+00 ± 0.00E+00 ▼	2.59E-08 ± 4.26E-08 ▼	0.00E+00 ± 0.00E+00 ▼
f_7	1.33E+02 ± 1.11E+01 ▲	1.27E+02 ± 1.36E+01 ▲	2.63E+02 ± 4.97E+01 ▲	1.26E+02 ± 1.52E+01 ▲	1.25E+02 ± 2.32E+01 ▲	1.55E+02 ± 1.04E+01 ▲
f_8	7.89E+01 ± 7.30E+00 ▲	7.95E+01 ± 1.30E+01 ▲	1.43E+02 ± 2.47E+01 ▲	7.60E+01 ± 1.05E+01 ▲	6.51E+01 ± 1.51E+01 ▲	9.63E+01 ± 1.15E+01 ▲
f_9	8.13E-01 ± 1.65E+00 ▼	5.57E+01 ± 6.30E+01 ▲	8.85E+02 ± 4.33E+02 ▲	7.16E+02 ± 2.55E+02 ▲	7.78E+01 ± 1.07E+02 ◆	5.65E+01 ± 6.20E+01 ◆
f_{10}	4.60E+03 ± 3.84E+02 ▲	3.93E+03 ± 2.90E+02 ◆	4.19E+03 ± 4.41E+02 ▲	3.92E+03 ± 2.83E+02 ◆	3.87E+03 ± 3.52E+02 ◆	4.84E+03 ± 3.34E+02 ▲
f_{11}	1.94E+02 ± 1.90E+02 ▼	3.21E+02 ± 1.95E+02 ◆	3.67E+02 ± 7.27E+01 ◆	2.01E+02 ± 7.05E+01 ▼	1.85E+02 ± 6.26E+01 ▼	9.75E+02 ± 9.27E+02 ◆
f_{12}	2.42E+05 ± 5.03E+05 ▲	1.17E+04 ± 6.99E+03 ▲	2.71E+04 ± 3.08E+04 ▲	2.41E+04 ± 2.13E+04 ▲	1.16E+06 ± 8.16E+05 ▲	2.39E+06 ± 2.59E+06 ▲
f_{13}	2.84E+03 ± 3.21E+03 ◆	2.26E+03 ± 2.21E+03 ◆	2.95E+03 ± 2.53E+03 ▲	2.22E+03 ± 2.07E+03 ◆	8.20E+02 ± 4.61E+02 ▼	2.86E+03 ± 3.30E+03 ◆
f_{14}	1.53E+04 ± 7.77E+04 ▲	2.62E+02 ± 1.08E+02 ◆	7.56E+02 ± 1.31E+03 ▲	1.78E+02 ± 6.56E+01 ▼	1.12E+02 ± 2.48E+01 ▼	4.58E+04 ± 1.35E+05 ▼
f_{15}	5.04E+03 ± 5.51E+03 ▲	3.76E+02 ± 2.02E+02 ◆	7.46E+02 ± 8.93E+02 ▲	7.04E+02 ± 1.74E+03 ◆	3.14E+02 ± 4.70E+02 ▼	4.22E+03 ± 6.42E+03 ▲
f_{16}	1.04E+03 ± 1.67E+02 ▲	9.27E+02 ± 1.82E+02 ◆	9.96E+02 ± 2.68E+02 ◆	8.96E+02 ± 1.75E+02 ◆	8.41E+02 ± 2.39E+02 ◆	1.19E+03 ± 2.00E+02 ▲
f_{17}	7.23E+02 ± 1.19E+02 ▲	6.48E+02 ± 1.67E+02 ◆	8.34E+02 ± 2.02E+02 ▲	6.42E+02 ± 1.44E+02 ◆	6.85E+02 ± 1.73E+02 ◆	7.78E+02 ± 1.58E+02 ▲
f_{18}	2.13E+05 ± 3.09E+05 ▲	1.85E+03 ± 1.73E+03 ▲	4.67E+03 ± 5.92E+03 ▲	1.35E+03 ± 6.86E+02 ▲	2.28E+05 ± 2.31E+05 ▲	3.96E+05 ± 4.34E+05 ▲
f_{19}	4.58E+02 ± 3.71E+02 ▲	2.24E+02 ± 2.61E+02 ▲	1.03E+03 ± 9.24E+02 ▲	1.52E+02 ± 7.44E+01 ◆	1.32E+02 ± 7.94E+01 ◆	2.26E+02 ± 2.51E+02 ◆
f_{20}	5.40E+02 ± 1.07E+02 ▲	4.30E+02 ± 1.41E+02 ◆	6.52E+02 ± 1.52E+02 ▲	4.62E+02 ± 1.34E+02 ◆	4.41E+02 ± 1.23E+02 ◆	6.47E+02 ± 1.67E+02 ▲
f_{21}	3.09E+02 ± 1.14E+01 ▲	3.09E+02 ± 1.43E+01 ▲	3.60E+02 ± 2.06E+01 ▲	3.04E+02 ± 1.34E+01 ▲	2.92E+02 ± 1.98E+01 ◆	3.26E+02 ± 1.21E+01 ▲
f_{22}	4.83E+03 ± 9.44E+02 ▲	4.39E+03 ± 4.41E+02 ◆	4.84E+03 ± 9.63E+02 ▲	4.47E+03 ± 9.24E+02 ◆	4.03E+03 ± 1.38E+03 ◆	5.52E+03 ± 3.19E+02 ▲
f_{23}	5.42E+02 ± 1.32E+01 ▲	5.75E+02 ± 2.10E+01 ▲	6.15E+02 ± 3.13E+01 ▲	5.48E+02 ± 1.30E+01 ▲	5.31E+02 ± 1.71E+01 ▲	5.64E+02 ± 1.31E+01 ▲
f_{24}	4.71E+02 ± 1.43E+01 ▲	5.03E+02 ± 1.67E+01 ▲	5.33E+02 ± 2.78E+01 ▲	4.69E+02 ± 1.21E+01 ▲	4.62E+02 ± 2.52E+01 ▲	4.98E+02 ± 1.43E+01 ▲
f_{25}	5.59E+02 ± 2.22E+00 ◆	5.72E+02 ± 2.43E+01 ◆	6.23E+02 ± 4.41E+01 ▲	6.18E+02 ± 4.00E+01 ▲	5.57E+02 ± 6.18E+00 ▼	5.68E+02 ± 9.54E+00 ▲
f_{26}	3.16E+03 ± 1.34E+02 ▲	3.69E+03 ± 2.23E+02 ▲	4.96E+03 ± 9.84E+02 ▲	3.31E+03 ± 2.14E+02 ▲	3.05E+03 ± 2.47E+02 ▲	3.41E+03 ± 1.37E+02 ▲
f_{27}	5.34E+02 ± 2.04E+01 ◆	5.49E+02 ± 2.72E+01 ▲	6.42E+02 ± 8.47E+01 ▲	5.63E+02 ± 4.02E+01 ▲	5.06E+02 ± 1.27E+01 ▼	5.25E+02 ± 1.15E+01 ◆
f_{28}	5.13E+02 ± 1.37E+01 ▲	5.06E+02 ± 1.80E+01 ◆	4.89E+02 ± 3.23E+01 ◆	5.03E+02 ± 2.48E+01 ◆	5.00E+02 ± 2.21E+01 ◆	5.10E+02 ± 1.15E+01 ◆
f_{29}	1.02E+03 ± 2.36E+02 ◆	1.15E+03 ± 1.90E+02 ▲	1.19E+03 ± 2.36E+02 ▲	1.03E+03 ± 1.01E+02 ◆	1.02E+03 ± 1.24E+02 ◆	1.11E+03 ± 1.13E+02 ◆
f_{30}	2.54E+04 ± 5.01E+04 ▲	9.28E+03 ± 5.29E+03 ▲	5.91E+03 ± 1.09E+03 ◆	6.99E+03 ± 4.29E+03 ▲	1.90E+04 ± 6.80E+03 ▲	1.85E+05 ± 2.40E+05 ▲
▲	22	17	24	16	11	20
▼	3	0	1	3	8	2
◆	4	12	4	10	10	7

TABLE 30. Summary of statistical comparisons on CEC 2017 considering parameter adaptation in all algorithms.

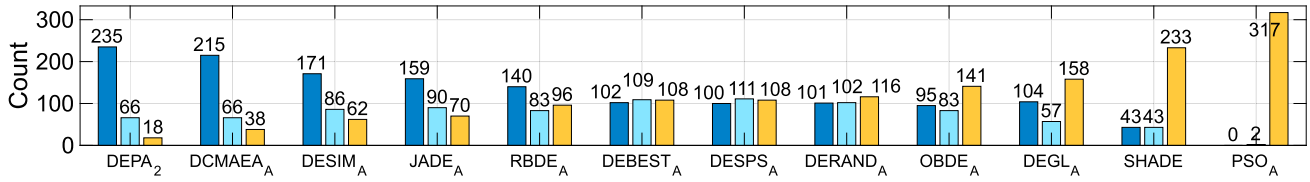
1. Summary of statistical comparisons for $D = 10$



2. Summary of statistical comparisons for $D = 30$



3. Summary of statistical comparisons for $D = 50$



seconded by algorithms using mutations based on similarity (DESIM$_A$), rank-based selection mechanisms (RBDE$_A$), and mutation based on covariance matrix adaptation (DCMAEA$_A$).

The above-mentioned results show the potential to tackle optimization problems with diverse landscapes and dimensionalities. Further research on gradient-free optimization using different benchmark sets and the study of large-scale problems are in our agenda. For generalization purposes, it is desirable that the optimization algorithm operates over a diverse, and possibly dynamic and changing, set of control and optimization problems with minimal human intervention, thus self-adaptation of sampling parameters becomes key. Also, the studied algorithms are mainly single-objective optimization heuristics, that is they are inherently unable to consider performance and robustness. Although a straightforward approach consists in composing performance and constraints into a single objective function (e.g., weighted sum), the approach opens questions as to how to suitably compose performance and robustness functions, how to choose suitable parameters for the composition functions, and how to suitably select from the Pareto frontier. Such questions are better addressed by multi-objective heuristics and the recent variants. The study of multi-objective and constraint satisfaction techniques is part of our future work.

VI. CONCLUSION

In this paper, we proposed an approach to tackle the PID parameter tuning problem by a new class of Differential Evolution algorithm with success-based Particle Adaptations (DEPA, in short). The basic idea of the proposed algorithm is to integrate the principles of difference of vectors, particle schemes and adaptation mechanisms by using particle-based sampling and adaptation through archive (memory) schemes to enable the tracking of both potential solution vectors and sampling parameters through the search process. As such, an advantageous feature of the proposed approach is the self-adaptation under tight computational budgets and the ability to select potential referent vectors depending on success-based selections.

In the context of PID tuning, we extend the scope of analysis in terms of number of optimization and control instances. The rigorous computational simulations over a large set of control problems (25 control instances considering the control trajectories with linear, nonlinear, continuous, and discontinuous state transitions in DC motor position control, DC motor velocity control, magnetic levitation control, inverted pendulum, crane stabilization), and the comparisons with a large set of closely related optimization algorithms, and their extended adaptive variants (23 optimization algorithms were evaluated in total) has shown

the competitive/outperforming convergence performance of the proposed algorithm under tight function evaluation budgets (in the order of 1000 function evaluations). And the experiments on a real-world inverted pendulum device show the potential for transferability of the learned gains to unseen situations.

In particular, the overall statistical results on control performance suggest that:

- the proposed framework using a fixed set of parameters performs better in 558 control instances when using ISE as performance metric and population sizes of 10, 50 and 100,
- the proposed approach using adaptation mechanisms outperformed in 358 control instances compared to other algorithms for PID control tasks using IAE, ITAE and ITSE as performance metrics, outperforming other related adaptation-based optimization algorithms,
- on DC motor position control, DC motor velocity control, magnetic levitation, inverted pendulum, and crane stabilization the proposed approach outperformed other related adaptation-based optimization algorithms in 99, 71, 154, 15 and 19 control instances,
- the proposed approach performed better than other algorithms in motor position and velocity control; however, DEPA performed similar to other algorithms in the crane stabilization problem (in 144 cases), and underperformed in inverted pendulum control instances (138 cases).

The above-mentioned observations pinpoint the merits and the ability of DEPA-based algorithms for control applications. Overall control instances, DEPA shows the attractive performance over most control problem instances whereas algorithms using composition of global-local donors such as DEGL_A and PSO_A underperform compared across all problems and fitness/performance metrics.

Furthermore, we evaluated the algorithmic extension and the generalization towards further optimization landscapes considering the CEC 2017 benchmark suite, which models single-objective fitness functions with unimodality, multimodality, hybrid functions and compositions. The computational experiments show the attractive/outperforming performance overall problem instances. The statistical comparisons show that the extension performs better than other related algorithms in 182, 215 and 235 problem instances in 10, 30 and 50 dimensions.

Our proposed approach is useful to suggest PID parameters within tight computational budgets attaining reasonable control performances. The analysis of the convergence using different population size (a key parameter in population-based algorithms) has been presented in the paper. Yet, since the extended algorithms using adaptation mechanisms adapt the step size F and the crossover rate CR, the sensitivity analysis to such parameters becomes unnecessary.

An important line of future research is to develop continual learning algorithms for general control problems.

TABLE 31. Heuristic acronyms.

Acronym	Algorithm
ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
ACONM	Ant Colony with Nelder Mead
ALO	Ant Lion Optimization
ALPSO	Augmented Lagrangian PSO
AMSGrad	Adaptive Moment Estimation Algorithm ADAM
AOA	Arithmetic Optimization Algorithm
ASO	Atom Search Optimization
ATS	Adaptive Tabu Search
AVOA	African Vultures Optimization Algorithm
BBBC	Bing Bang-Big Crunch
BBO	Biogeography-based Optimization
BMA	Bacterial Memetic Algorithm
CAS	Chaotic Ant Swarm
CHASO	Chaotic Atom Search Optimization
CS	Cuckoo Search
D3QN	Dueling Double Deep Q Network
DE	Differential Evolution
DRL	Deep Reinforcement Learning
EOA	Equilibrium Optimizer Algorithm
ES	Evolution Strategies
FFA	Firefly Algorithm
FMO	Fish Migration Optimization
GA	Genetic Algorithm
GOA	Grasshopper Optimization Algorithm
GTO	Gorilla Troop Optimization
GWO	Grey Wolf Optimizer
HAOAGTO	Hybrid of AOA and GTO
HKA	Heuristic Kalman Algorithm
IQGA	Quantum Genetic Algorithm
ISCA	Improved Sine Cosine Algorithm
IWCA	Improved Wolf Colony Algorithm
IWCA-CS	Improved WCA and CS
IWO	Invasive Weed Optimization
MA	Monkey Algorithm
MA	Mayfly Algorithm
MFO	Moth Flame Optimization
MH	Metaheuristic Algorithm
MMA	Mayfly Algorithm with Median Position
MM-MADRL	Modified Monkey-Multiagent DRL

TABLE 31. *Continued.* **Heuristic acronyms.**

MODE-II	Multiobjective Differential Evolution - II
MOSTA	Multi-Objective State Transition Algorithm
MOTEO	Thermal Exchange Optimization
MPA	Marine Predators Algorithm
MPASEDA	MPA with Safe Experimentation Dynamics
MSFA	Modified Smoothed Function Algorithm
NSDE-R	Many objective DE with reference points
NSGAII	Non-dominated Sorting Genetic Algorithm II
PESA2	Modified Pareto Envelop-based Selection II
PPA	Parasitism Predation Algorithm
PSO	Particle Swarm Optimization
QGA	Quantum Genetic Algorithm
QSLP	Q-learning with Swarm Learning Process
SA	Simulated Annealing
SADE	Self-Adaptive Differential Evolution
SaS	Salvia Swarm Algorithm
SCA	Sine Cosine Algorithm
SCSO	Sand Cat Swarm Optimization
SFA	Smoothed Function Algorithm
SFS	Stochastic Fractal Search
SLP	Swarm Learning Process
SOA	Seagull Optimization Algorithm
SPEA2	Strength Pareto Evolutionary Algorithm II
SSA	Sparse Search Algorithm
SSO	Social Spider Optimization Algorithm
SSQPSO	Sequential Switching Quadratic PSO
TS	Tabu Search
WCA	Wolf Colony Algorithm
WOA	Whale Optimization Algorithm

The optimization-based and gradient-free approaches to exploring control rules has the potential to learn a repertoire of behaviours not seen by optimal control and classical reinforcement learning. Another line of future work is to explore the potential adaptation mechanisms for discrete and combinatorial problems. Furthermore, studying the many-objective schemes, robustness constraints, the large-scale domains (hundreds and thousands of dimensions), and the improved switching of exploration and exploitation through modular decompositions for a wider set of control and optimization problems is in our agenda.

APPENDIX A
ACRONYMS

See Table 31.

REFERENCES

[1] P. Rocco, "Stability of PID control for industrial robot arms," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 606–614, Aug. 1996.

[2] Y. Li, K. H. Ang, and G. C. Y. Chong, "Patents, software, and hardware for PID control: An overview and analysis of the current art," *IEEE Control Syst.*, vol. 26, no. 1, pp. 42–54, Feb. 2006.

[3] A. Visioli, *Practical PID Control*. London, U.K.: Springer-Verlag, 2006.

[4] S. Skoczowski, S. Domek, K. Pietrusewicz, and B. Broel-Plater, "A method for improving the robustness of PID control," *IEEE Trans. Ind. Electron.*, vol. 52, no. 6, pp. 1669–1676, Dec. 2005.

[5] H. M. Hasanien, "Design optimization of PID controller in automatic voltage regulator system using Taguchi combined genetic algorithm method," *IEEE Syst. J.*, vol. 7, no. 4, pp. 825–831, Dec. 2013.

[6] M. Abdelwahab, V. Parque, A. M. R. Fath Elbab, A. A. Abouelsoud, and S. Sugano, "Trajectory tracking of wheeled mobile robots using Z-number based fuzzy logic," *IEEE Access*, vol. 8, pp. 18426–18441, 2020.

[7] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *J. Dyn. Syst., Meas., Control*, vol. 115, no. 2, pp. 220–222, 1993.

[8] W. K. Ho, O. P. Gan, E. B. Tay, and E. L. Ang, "Performance and gain and phase margins of well-known PID tuning formulas," *IEEE Trans. Control Syst. Technol.*, vol. 4, no. 4, pp. 473–477, Jul. 1996.

[9] R. Vilanova and A. Visioli, *PID Control in the Third Millennium: Lessons Learned and New Approaches*. London, U.K.: Springer-Verlag, 2012.

[10] J. Li, T. Chai, L. Fan, L. Pan, and J. Gong, "PID controller tuning using particle filtering optimization," in *Proc. 3rd Int. Symp. Syst. Control Aeronaut. Astronaut.*, Jun. 2010, pp. 66–69.

[11] H. Panagopoulos, K. Astrom, and T. Hagglund, "Design of PID controllers based on constrained optimization," in *Proc. Amer. Control Conf.*, 1999, pp. 3858–3862.

[12] D. A. Plaza-Guingla, R. M. Idrovo, A. J. Valencia, and C. S. Lopez, "Enhancing the performance of the particle filtering optimization algorithm for the tuning of PID controllers," in *Proc. 5th Int. Conf. Control, Mechtron. Autom.* New York, NY, USA: Association for Computing Machinery, 2017, pp. 95–99.

[13] I. Birs, C. Muresan, M. Mihai, E. Dulf, and R. De Keyser, "Tuning guidelines and experimental comparisons of sine based autotuning methods for fractional order controllers," *IEEE Access*, vol. 10, pp. 86671–86683, 2022.

[14] T. A. M. Euzébio, M. T. D. Silva, and A. S. Yamashita, "Decentralized PID controller tuning based on nonlinear optimization to minimize the disturbance effects in coupled loops," *IEEE Access*, vol. 9, pp. 156857–156867, 2021.

[15] J. L. Meza, V. Santibanez, R. Soto, and M. A. Llama, "Fuzzy self-tuning PID semiglobal regulator for robot manipulators," *IEEE Trans. Ind. Electron.*, vol. 59, no. 6, pp. 2709–2717, Jun. 2012.

[16] Q. Chen, Y. Tan, J. Li, and I. Mareels, "Decentralized PID control design for magnetic levitation systems using extremum seeking," *IEEE Access*, vol. 6, pp. 3059–3067, 2018.

[17] R. Mahadeva, M. Kumar, S. P. Patole, and G. Manik, "PID control design using AGPSO technique and its application in TITO reverse osmosis desalination plant," *IEEE Access*, vol. 10, pp. 125881–125892, 2022.

[18] C.-L. Lee and C.-C. Peng, "Analytic time domain specifications PID controller design for a class of 2^{nd} order linear systems: A genetic algorithm method," *IEEE Access*, vol. 9, pp. 99266–99275, 2021.

[19] R. P. Borase, D. K. Maghade, S. Y. Sondkar, and S. N. Pawar, "A review of PID control, tuning methods and applications," *Int. J. Dyn. Control*, vol. 9, no. 2, pp. 818–827, Jun. 2021.

[20] Y. Su, Q. Yu, and L. Zeng, "Parameter self-tuning PID control for greenhouse climate control problem," *IEEE Access*, vol. 8, pp. 186157–186171, 2020.

[21] P. Wang and D. Kwok, "Optimal design of PID process controllers based on genetic algorithms," *Control Eng. Pract.*, vol. 2, no. 4, pp. 641–648, 1994.

[22] P. Wang and D. P. Kwok, "Auto-tuning of classical PID controllers using an advanced genetic algorithm," in *Proc. Int. Conf. Ind. Electron., Control, Instrum., Autom.*, vol. 3, 1992, pp. 1224–1229.

[23] H. Zhong, W. Bai, W. Huang, and Z. Zheng, "Adaptive PID controller design method based on PSO," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Aug. 2020, pp. 4501–4506.

[24] T. Kobaku, R. Poola, and V. Agarwal, "Design of robust PID controller using PSO-based automated QFT for nonminimum phase boost converter," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 12, pp. 4854–4858, Dec. 2022.

[25] A. Parnianifard, V. Rezaie, S. Chaudhary, M. A. Imran, and L. Wuttisittikulkij, "New adaptive surrogate-based approach combined swarm optimizer assisted less tuning cost of dynamic production-inventory control system," *IEEE Access*, vol. 9, pp. 144054–144066, 2021.

[26] R.-J. Wai, J.-D. Lee, and K.-L. Chuang, "Real-time PID control strategy for Maglev transportation system via particle swarm optimization," *IEEE Trans. Ind. Electron.*, vol. 58, no. 2, pp. 629–646, Feb. 2011.

[27] L. Jia and X. Zhao, "An improved particle swarm optimization (PSO) optimized integral separation PID and its application on central position control system," *IEEE Sensors J.*, vol. 19, no. 16, pp. 7064–7071, Aug. 2019.

[28] E. D. P. Puchta, H. V. Siqueira, and M. dos S. Kaster, "Optimization tools based on metaheuristics for performance enhancement in a Gaussian adaptive PID controller," *IEEE Trans. Cybern.*, vol. 50, no. 3, pp. 1185–1194, Mar. 2020.

[29] J. Chen, M. N. Omidvar, M. Azad, and X. Yao, "Knowledge-based particle swarm optimization for PID controller tuning," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Oct. 2017, pp. 1819–1826.

[30] D. S. Acharya, B. Sarkar, and D. Bharti, "A fractional order particle swarm optimization for tuning fractional order PID controller for magnetic levitation plant," in *Proc. 1st IEEE Int. Conf. Meas., Instrum., Control Autom. (ICMICA)*, Jun. 2020, pp. 1–6.

[31] H. Wang, S. Xu, and H. Hu, "PID controller for PMSM speed control based on improved quantum genetic algorithm optimization," *IEEE Access*, vol. 11, pp. 61091–61102, 2023.

[32] M. Z. Mohd Tumari, M. A. Ahmad, M. H. Suid, and M. R. Hao, "An improved marine predators algorithm-tuned fractional-order PID controller for automatic voltage regulator system," *Fractal Fractional*, vol. 7, no. 7, p. 561, Jul. 2023.

[33] Y. Luo, Z. Wang, H. Dong, J. Mao, and F. E. Alsaadi, "A novel sequential switching quadratic particle swarm optimization scheme with applications to fast tuning of PID controllers," *Inf. Sci.*, vol. 633, pp. 305–320, Jul. 2023.

[34] A. Ambroziak and A. Chojecki, "The PID controller optimisation module using fuzzy self-tuning PSO for air handling unit in continuous operation," *Eng. Appl. Artif. Intell.*, vol. 117, 2023, Art. no. 105485.

[35] A. Javadian, N. Nariman-Zadeh, and A. Jamali, "Evolutionary design of marginally robust multivariable PID controller," *Eng. Appl. Artif. Intell.*, vol. 121, May 2023, Art. no. 105938.

[36] J. Pakdeeto, S. Wansungnoen, K. Areerak, and K. Areerak, "Optimal speed controller design of commercial BLDC motor by adaptive Tabu search algorithm," *IEEE Access*, vol. 11, pp. 79710–79720, 2023.

[37] A. Goel, R. Mahadeva, S. P. Patole, and G. Manik, "Dynamic modeling and controller design for a parabolic trough solar collector," *IEEE Access*, vol. 11, pp. 33381–33392, 2023.

[38] E. S. Ghith and F. A. A. Tolba, "Tuning PID controllers based on hybrid arithmetic optimization algorithm and artificial gorilla troop optimization for micro-robotics systems," *IEEE Access*, vol. 11, pp. 27138–27154, 2023.

[39] A. Nagarajan and A. A. Victoire, "Optimization reinforced PID-sliding mode controller for rotary inverted pendulum," *IEEE Access*, vol. 11, pp. 24420–24430, 2023.

[40] D. F. Zambrano-Gutierrez, J. M. Cruz-Duarte, J. G. Avina-Cervantes, J. C. Ortiz-Bayliss, J. J. Yanez-Borjas, and I. Amaya, "Automatic design of metaheuristics for practical engineering applications," *IEEE Access*, vol. 11, pp. 7262–7276, 2023.

[41] A. Muqeet, A. Israr, M. H. Zafar, M. Mansoor, and N. Akhtar, "A novel optimization algorithm based PID controller design for real-time optimization of cutting depth and surface roughness in finish hard turning processes," *Results Eng.*, vol. 18, Jun. 2023, Art. no. 101142.

[42] Y. Hong, C. Fu, and B. Merci, "Optimization and determination of the parameters for a PID based ventilation system for smoke control in tunnel fires: Comparative study between a genetic algorithm and an analytical trial-and-error method," *Tunnelling Underground Space Technol.*, vol. 136, Jun. 2023, Art. no. 105088.

[43] A. Ansarian and M. J. Mahmoodabadi, "Multi-objective optimal design of a fuzzy adaptive robust fractional-order PID controller for a nonlinear unmanned flying system," *Aerosp. Sci. Technol.*, vol. 141, Oct. 2023, Art. no. 108541.

[44] X. Zhang, S. Zhang, W. Dong, and K. Wang, "A novel time synchronization method for smart grid based on improved wolf colony algorithm-cuckoo search optimized fuzzy PID controller," *IEEE Access*, vol. 10, pp. 116959–116971, 2022.

[45] H. Ahmed, A. As'arry, A. A. Hairuddin, M. Khair Hassan, Y. Liu, and E. C. U. Onwudinjo, "Online DE optimization for fuzzy-PID controller of semi-active suspension system featuring MR damper," *IEEE Access*, vol. 10, pp. 129125–129138, 2022.

[46] C. Fu, X. Ma, and L. Zhang, "Fuzzy-PID strategy based on PSO optimization for pH control in water and fertilizer integration," *IEEE Access*, vol. 10, pp. 4471–4482, 2022.

[47] J. Park, H. Kim, K. Hwang, and S. Lim, "Deep reinforcement learning based dynamic proportional-integral (PI) gain auto-tuning method for a robot driver system," *IEEE Access*, vol. 10, pp. 31043–31057, 2022.

[48] J. Sánchez-Palma and J. L. Ordoñez-Ávila, "A PID control algorithm with adaptive tuning using continuous artificial hydrocarbon networks for a two-tank system," *IEEE Access*, vol. 10, pp. 114694–114710, 2022.

[49] P. Sanchez-Sanchez, J. G. Cebada-Reyes, A. Ruiz-Garcia, A. Montiel-Martinez, and F. Reyes-Cortes, "Differential evolution algorithms comparison used to tune a visual control law," *IEEE Access*, vol. 10, pp. 46028–46042, 2022.

[50] E. Yusubov and L. Bekirova, "A moth-flame optimized robust PID controller for a SEPIC in photovoltaic applications," *IFAC-PapersOnLine*, vol. 55, no. 11, pp. 120–125, 2022.

[51] R. Mok and M. A. Ahmad, "Fast and optimal tuning of fractional order PID controller for AVR system based on memorizable-smoothed functional algorithm," *Eng. Sci. Technol., Int. J.*, vol. 35, Nov. 2022, Art. no. 101264.

[52] N. P. Lawrence, M. G. Forbes, P. D. Loewen, D. G. McClement, J. U. Backström, and R. B. Gopaluni, "Deep reinforcement learning with shallow controllers: An experimental application to PID tuning," *Control Eng. Pract.*, vol. 121, Apr. 2022, Art. no. 105046.

[53] G. Lei, X. Chang, Y. Tianhang, and W. Tuerxun, "An improved mayfly optimization algorithm based on median position and its application in the optimization of PID parameters of hydro-turbine governor," *IEEE Access*, vol. 10, pp. 36335–36349, 2022.

[54] W. Assawinchaichote, C. Angeli, and J. Pongfai, "Proportional-integral-derivative parametric autotuning by novel stable particle swarm optimization (NSPSO)," *IEEE Access*, vol. 10, pp. 40818–40828, 2022.

[55] B. Guo, Z. Zhuang, J.-S. Pan, and S.-C. Chu, "Optimal design and simulation for PID controller using fractional-order fish migration optimization algorithm," *IEEE Access*, vol. 9, pp. 8808–8819, 2021.

[56] H. Zhang, W. Assawinchaichote, and Y. Shi, "New PID parameter autotuning for nonlinear systems based on a modified monkey–multiagent DRL algorithm," *IEEE Access*, vol. 9, pp. 78799–78811, 2021.

[57] V. Parque, "A differential particle scheme with successful parent selection and its application to PID control tuning," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Kraków, Poland, Jul. 2021, pp. 522–529.

[58] V. Parque, "A differential particle scheme and its application to PID parameter tuning of an inverted pendulum," in *Proc. GECCO Genetic Evolutionary Comput. Conf., Companion*, Lille, France, K. Krawiec, Ed. Jul. 2021, pp. 1937–1943.

[59] M. Sabuhi, N. Mahmoudi, and H. Khazaei, "Optimizing the performance of containerized cloud software systems using adaptive PID controllers," *ACM Trans. Auton. Adapt. Syst.*, vol. 15, no. 3, pp. 1–27, Aug. 2021.

[60] M. S. Ayas and E. Sahin, "FOPID controller with fractional filter for an automatic voltage regulator," *Comput. Electr. Eng.*, vol. 90, Mar. 2021, Art. no. 106895.

[61] S. Arun and T. Manigandan, "Design of ACO based PID controller for zeta converter using reduced order methodology," *Microprocessors Microsyst.*, vol. 81, Mar. 2021, Art. no. 103629.

[62] J. P. Moura, J. V. F. Neto, E. F. M. Ferreira, and E. M. A. Filho, "On the design and analysis of structured-ANN for online PID-tuning to bulk resumption process in ore mining system," *Neurocomputing*, vol. 402, pp. 266–282, Aug. 2020.

[63] J. Pongfai, X. Su, H. Zhang, and W. Assawinchaichote, "PID controller autotuning design by a deterministic Q-SLP algorithm," *IEEE Access*, vol. 8, pp. 50010–50021, 2020.

[64] B. Verma and P. K. Padhy, "Robust fine tuning of optimal PID controller with guaranteed robustness," *IEEE Trans. Ind. Electron.*, vol. 67, no. 6, pp. 4911–4920, Jun. 2020.

[65] R. Pradhan, S. K. Majhi, J. K. Pradhan, and B. B. Pati, "Optimal fractional order PID controller design using ant lion optimizer," *Ain Shams Eng. J.*, vol. 11, no. 2, pp. 281–291, Jun. 2020.

[66] J.-J. Wang and T. Kumbasar, "Optimal PID control of spatial inverted pendulum with big bang–big crunch optimization," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 3, pp. 822–832, May 2020.

[67] A. Mughees and S. A. Mohsin, "Design and control of magnetic levitation system by optimizing fractional order PID controller using ant colony optimization algorithm," *IEEE Access*, vol. 8, pp. 116704–116723, 2020.

[68] B. Hekimoglu, "Optimal tuning of fractional order PID controller for DC motor speed control via chaotic atom search optimization algorithm," *IEEE Access*, vol. 7, pp. 38100–38114, 2019.

[69] L. R. R. dos Santos, F. R. Durand, and T. Abrão, "Adaptive PID scheme for OCDMA next generation PON based on heuristic swarm optimization," *IEEE Syst. J.*, vol. 13, no. 1, pp. 500–510, Mar. 2019.

[70] A. M. Mosaad, M. A. Attia, and A. Y. Abdelaziz, "Whale optimization algorithm to tune PID and PIDA controllers on AVR system," *Ain Shams Eng. J.*, vol. 10, no. 4, pp. 755–767, Dec. 2019.

[71] K. S. Rajesh and S. S. Dash, "Load frequency control of autonomous power system using adaptive fuzzy based PID controller optimized on improved sine cosine algorithm," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 6, pp. 2361–2373, Jun. 2019.

[72] M. Ghanamijaber, "A hybrid fuzzy-PID controller based on gray wolf optimization algorithm in power system," *Evolving Syst.*, vol. 10, no. 2, pp. 273–284, Jun. 2019.

[73] S.-E.-I. Hasseni, L. Abdou, and H.-E. Glida, "Parameters tuning of a quadrotor PID controllers by using nature-inspired algorithms," *Evol. Intell.*, vol. 14, no. 1, pp. 61–73, Mar. 2021.

[74] K. Jagatheesan, B. Anand, S. Samanta, N. Dey, A. S. Ashour, and V. E. Balas, "Design of a proportional-integral-derivative controller for an automatic generation control of multi-area power thermal systems using firefly algorithm," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 2, pp. 503–515, Mar. 2019.

[75] E. S. A. Shahri, A. Alfi, and J. A. T. Machado, "Fractional fixed-structure H_∞ controller design using augmented Lagrangian particle swarm optimization with fractional order velocity," *Appl. Soft Comput.*, vol. 77, pp. 688–695, Apr. 2019.

[76] X. Zhou, J. Zhou, C. Yang, and W. Gui, "Set-point tracking and multi-objective optimization-based PID control for the goethite process," *IEEE Access*, vol. 6, pp. 36683–36698, 2018.

[77] M. J. Blondin, J. Sanchis, P. Sicard, and J. M. Herrero, "New optimal controller tuning method for an AVR system using a simplified ant colony optimization with a new constrained Nelder–Mead algorithm," *Appl. Soft Comput.*, vol. 62, pp. 216–229, Jan. 2018.

[78] B. L. G. Costa, C. L. Graciola, B. A. Angélico, A. Goedtel, and M. F. Castoldi, "Metaheuristics optimization applied to PI controllers tuning of a DTC-SVM drive for three-phase induction motors," *Appl. Soft Comput.*, vol. 62, pp. 776–788, Jan. 2018.

[79] V. Parque, A. Khalifa, and T. Miyashita, "Towards stagnation-free particle swarm optimization and its application to PID control tuning," in *Proc. SICE Int. Symp. Control Syst.*, 2018, pp. 1–12.

[80] V. H. Alves Ribeiro and G. R. Meza, "Multi-objective PID controller tuning for an industrial gasifier," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, pp. 1–6.

[81] A. Rodíguez-Molina, M. G. Villarreal-Cervantes, and M. Aldape-Pérez, "Adaptive control for the four-bar linkage mechanism based on differential evolution," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, pp. 1–7.

[82] A. Panday and J. Pedro, "Particle swarm-optimised PID control of a receiver aircraft during aerial refuelling," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, pp. 1–8.

[83] A. Baldini, L. Ciabattoni, R. Felicetti, F. Ferracuti, A. Freddi, and A. Monteriù, "Nonlinear control of a photovoltaic battery system via ABC-tuned dynamic surface controller," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2017, pp. 1985–1991.

[84] R.-E. Precup, R.-C. David, and E. M. Petriu, "Grey wolf optimizer algorithm-based tuning of fuzzy control systems with reduced parametric sensitivity," *IEEE Trans. Ind. Electron.*, vol. 64, no. 1, pp. 527–534, Jan. 2017.

[85] N. N. Son, C. V. Kien, and H. P. H. Anh, "A novel adaptive feed-forward-PID controller of a SCARA parallel robot using pneumatic artificial muscle actuator based on neural network and modified differential evolution algorithm," *Robot. Auto. Syst.*, vol. 96, pp. 65–80, Oct. 2017.

[86] C. Wei and D. Söffker, "Optimization strategy for PID-controller design of AMB rotor systems," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 3, pp. 788–803, May 2016.

[87] M. G. Villarreal-Cervantes and J. Alvarez-Gallegos, "Off-line PID control tuning for a planar parallel robot using DE variants," *Exp. Syst. Appl.*, vol. 64, pp. 444–454, Dec. 2016.

[88] B. Zahn, I. Ucherdzhiev, J. Szeles, J. Botzheim, and N. Kubota, "Optimization of a proportional-summation-difference controller for a line-tracing robot using bacterial memetic algorithm," in *Proc. Intell. Robot. Appl.*, 2016, pp. 362–372.

[89] M. Ünal, A. Ak, V. Topuz, and H. Erdal, *Optimization of PID Controllers Using Ant Colony and Genetic Algorithms*. Berlin, Germany: Springer-Verlag, 2013.

[90] L. D. S. Coelho and M. W. Pessôa, "A tuning strategy for multivariable PI and PID controllers using differential evolution combined with chaotic Zaslavskii map," *Exp. Syst. Appl.*, vol. 38, no. 11, pp. 13694–13701, May 2011.

[91] B. Nagaraj and N. Murugananth, "A comparative study of PID controller tuning using GA, EP, PSO and ACO," in *Proc. Int. Conf. Commun. Control Comput. Technol.*, Oct. 2010, pp. 305–313.

[92] S.-L. Cheng and C. Hwang, "Designing PID controllers with a minimum IAE criterion by a differential evolution algorithm," *Chem. Eng. Commun.*, vol. 170, no. 1, pp. 83–115, Jan. 1998.

[93] S. Omatu, T. Fujinaka, and M. Yoshioka, "Neuro-PID control for inverted single and double pendulums," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, 2000, pp. 2685–2690.

[94] S. Omatu and M. Yoshioka, "Self-tuning neuro-PID control and applications," in *Proc. IEEE Int. Conf. Syst., Man, Cybern., Comput. Cybern. Simulation*, 1997, pp. 1985–1989.

[95] V. P. Tran, F. Santoso, M. A. Garratt, and I. R. Petersen, "Distributed formation control using fuzzy self-tuning of strictly negative imaginary consensus controllers in aerial robotics," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 5, pp. 2306–2315, Oct. 2021.

[96] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, Dec. 1997.

[97] G. M. Zaslavsky, "The simplest case of a strange attractor," *Phys. Lett. A*, vol. 69, no. 3, pp. 145–147, Dec. 1978.

[98] X. Ji, Y. Zhang, D. Gong, and X. Sun, "Dual-surrogate-assisted cooperative particle swarm optimization for expensive multimodal problems," *IEEE Trans. Evol. Comput.*, vol. 25, no. 4, pp. 794–808, Aug. 2021.

[99] D. Yazdani, R. Cheng, D. Yazdani, J. Branke, Y. Jin, and X. Yao, "A survey of evolutionary continuous dynamic optimization over two decades—Part A," *IEEE Trans. Evol. Comput.*, vol. 25, no. 4, pp. 609–629, Aug. 2021.

[100] D. Yazdani, R. Cheng, D. Yazdani, J. Branke, Y. Jin, and X. Yao, "A survey of evolutionary continuous dynamic optimization over two decades—Part B," *IEEE Trans. Evol. Comput.*, vol. 25, no. 4, pp. 630–650, Aug. 2021.

[101] J.-S. Pan, N. Liu, and S.-C. Chu, "A hybrid differential evolution algorithm and its application in unmanned combat aerial vehicle path planning," *IEEE Access*, vol. 8, pp. 17691–17712, 2020.

[102] Z. Meng, Y. Chen, and X. Li, "Enhancing differential evolution with novel parameter control," *IEEE Access*, vol. 8, pp. 51145–51167, 2020.

[103] L. Cui, Q. Huang, G. Li, S. Yang, Z. Ming, Z. Wen, N. Lu, and J. Lu, "Differential evolution algorithm with tracking mechanism and backtracking mechanism," *IEEE Access*, vol. 6, pp. 44252–44267, 2018.

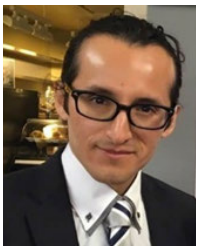
[104] S.-M. Guo, C.-C. Yang, P.-H. Hsu, and J. S.-H. Tsai, "Improving differential evolution with a successful-parent-selecting framework," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 717–730, Oct. 2015.

[105] L. Deng, H. Sun, and C. Li, "JDF-DE: A differential evolution with Jrand number decreasing mechanism and feedback guide technique for global numerical optimization," *Appl. Intell.*, vol. 51, no. 1, pp. 359–376, 2021.

[106] G. Sun, C. Li, and L. Deng, "An adaptive regeneration framework based on search space adjustment for differential evolution," *Neural Comput. Appl.*, vol. 33, pp. 9503–9519, Jan. 2021.

[107] K. Miao and Z. Wang, "Neighbor-induction and population-dispersion in differential evolution algorithm," *IEEE Access*, vol. 7, pp. 146358–146378, 2019.

[108] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, Oct. 2009.
[109] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 71–78.
[110] A. M. Sutton, M. Lunacek, and L. D. Whitley, "Differential evolution and non-separability: Using selective pressure to focus search," in *Proc. 9th Annu. Conf. Genetic Evol. Comput.* New York, NY, USA: ACM, Jul. 2007, pp. 1428–1435.
[111] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 526–553, Jun. 2009.
[112] E. Segredo, E. Lalla-Ruiz, and E. Hart, "A novel similarity-based mutant vector generation strategy for differential evolution," in *Proc. Genetic Evol. Comput. Conf.* New York, NY, USA: ACM, Jul. 2018, pp. 881–888.
[113] S. Ghosh, S. Das, S. Roy, S. K. M. Islam, and P. N. Suganthan, "A differential covariance matrix adaptation evolutionary algorithm for real parameter optimization," *Inf. Sci.*, vol. 182, no. 1, pp. 199–219, Jan. 2012.
[114] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 64–79, Feb. 2008.
[115] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw.*, vol. 4, 1995, pp. 1942–1948.
[116] R. Tanabe and A. Fukunaga, "Reviewing and benchmarking parameter control methods in differential evolution," *IEEE Trans. Cybern.*, vol. 50, no. 3, pp. 1170–1184, Mar. 2020.
[117] *Precision Modular Servo Control Experiments*, document 33-927S Ed01 122006, Feedback Instrum., Crowborough, U.K., 2006.
[118] *Magnetic Levitation Control Experiments*, document 33-942S Ed01 122006, Feedback Instrum., Crowborough, U.K., 2006.
[119] *Digital Pendulum Control Experiments*, document 33-936S Ed01 122006, Feedback Instrum., Crowborough, U.K., 2006.
[120] A. Kumar, K. V. Price, A. W. Mohamed, A. A. Hadi, and P. N. Suganthan. *Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization*. Accessed: Sep. 3, 2023. [Online]. Available: https://github.com/P-N-Suganthan/CEC2017-BoundContrained
[121] F. Peng, K. Tang, G. Chen, and X. Yao, "Multi-start JADE with knowledge transfer for numerical optimization," in *Proc. IEEE Congr. Evol. Comput.*, May 2009, pp. 1889–1895.



VICTOR PARQUE (Member, IEEE) received the B.Sc. degree in systems engineering from National Central University, in 2004, the M.B.A. degree from the Graduate School of Business Administration, Esan University, in 2009, and the Ph.D. degree from the Graduate School of Information, Production and Systems, Waseda University, in 2011. He is currently an Associate Professor with the Department of Modern Mechanical Engineering, Waseda University, and a JSUC Tokunin Professor with the Egypt-Japan University of Science and Technology. He was a Postdoctoral Fellow with the Department of Mechanical Engineering, Toyota Technological Institute, in 2012 and 2014, and an Assistant Professor with Waseda University, in 2014 and 2018. He has authored more than 100 articles in journals, conferences, and book chapters. He is actively involved in research collaborations with both industry and academia. His research interests include the principles and applications of learning and intelligent systems in design engineering, mechatronics, robotics, optimization, planning, and control. He was honored as a Finalist in the Hummies Awards for Human-Competitive Results, in 2018. He is a member of IEEE (CIS, RAS, IES, and SMC), ACM (SIGAI and SIGEVO), and Robotics Society of Japan (RSJ).

ALAA KHALIFA received the B.Sc. degree in industrial electronics and control engineering from the Faculty of Electronic Engineering, Menoufia University, Al Menoufia, Egypt, in 2010, and the M.Sc. and Ph.D. degrees in mechatronics and robotics engineering from the Egypt-Japan University of Science and Technology, New Borg El-Arab City, Alexandria, Egypt, in 2015 and 2018, respectively. His current research interests include tendon-driven manipulators, modeling and manipulation of deformable linear objects (DLOs), surgical robots, parallel robots, teleoperation systems, optimization techniques, and control system design.