Applied Intelligence (2024) 54:7229–7277
https://doi.org/10.1007/s10489-024-05502-1

Nonlinear crossing strategy-based particle swarm optimizations with time-varying acceleration coefficients

Keigo Watanabe[1] · Xiongshi Xu[2]

Accepted: 2 May 2024 / Published online: 3 June 2024

Abstract
In contemporary particle swarm optimization (PSO) algorithms, to efficiently explore global optimum solutions, it is common practice to set the inertia weight to monotonically decrease over time for stability, while allowing the two acceleration coefficients, representing cognitive and social factors, to adopt decreasing or increasing functions over time, including random variations. However, there has been little discussion on a unified design approach for these time-varying acceleration coefficients. This paper presents a unified methodology for designing monotonic decreasing or increasing functions to construct nonlinear time-varying inertia weight and two acceleration coefficients in PSO, along with a control strategy for exploring global optimum solutions. We first construct time-varying coefficients by linearly amplifying well-posed monotonic functions that decrease or increase over normalized time. Here, well-posed functions ensure satisfaction of specified conditions at the initial and terminal points of the search process. However, many of the functions employed thus far only satisfy well-posedness at either the initial or terminal points of the search time, prompting the proposal of a method to adjust them to virtually meet specified initial or terminal points. Furthermore, we propose a crossing strategy where the developed cognitive and social acceleration coefficients intersect within the search time interval, effectively guiding the search process by pre-determining crossing values and times. The performance of our Nonlinear Crossing Strategy-based Particle Swarm Optimization (NCS-PSO) is evaluated using the CEC2014 (Congress on Evolutionary Computation in 2014) benchmark functions. Through comprehensive numerical comparisons and statistical analyses, we demonstrate the superiority of our approach over seven conventional algorithms. Additionally, we validate our approach, particularly in a drone navigation scenario, through an example of optimal 3D path planning. These contributions advance the field of PSO optimization techniques, providing a robust approach to addressing complex optimization problems.

Keywords Nonlinear crossing strategy · Particle swarm optimization · Time-varying acceleration coefficients · Global optimization · 3D path planning

1 Introduction

Particle Swarm Optimization (PSO) by Kennedy and Eberhart [40] has been around 27 years since its first paper was published in 1995. Recently, Nayak et al. [56] surveyed the history of PSO spanning over 25 years, covering more than 450 literature sources. In their survey paper, PSO is highlighted as one of the most researched natural-inspired algorithms, extensively used not only by researchers but also by practitioners, together with discussing numerous improved variants and application domains. Furthermore, they point out that PSO forms the foundation of many recent metaheuristics.

Due to the simplicity of its structure and high performance, PSO has been applied in various fields:

- Control Engineering: PSO is used for parameter tuning in control systems [61, 63, 69], solving optimal control problems, and applications such as robot control [7, 16,

✉ Keigo Watanabe
watanabe@sys.okayama-u.ac.jp

Xiongshi Xu
xuxiongshi1107@126.com

[1] Graduate School of Environmental, Life, Natural Science and Technology, Okayama University, 3-1-1 Tsushima-naka, Kita-ku, Okayama City 700-8530, Okayama, Japan

[2] Graduate School of Mechanical Science and Engineering, Huazhong University of Science and Technology, East 3rd District, Guanshan Street, Wuhan City 430074, Hubei Province, China

52], path planning [6, 23, 24], and driver assistance systems in automobiles [10, 72, 108].

- Machine Learning: PSO is employed for training machine learning models like neural networks [21, 32, 106], support vector machines [14, 28, 71, 80], or Bayesian networks [5, 30, 48]. It is used for tasks such as hyperparameter optimization and feature selection.
- Image Processing: PSO finds applications in image segmentation, feature extraction, and image optimization. Examples include medical image processing [38, 43, 68, 102] and remote sensing [42, 57, 77, 87].
- Combinatorial Optimization: PSO is utilized to solve combinatorial optimization problems, including the traveling salesman problem [62, 81, 104, 105], and placement problems [26, 31, 51, 76, 96, 100].
- Power Systems: PSO plays a role in power network planning and control [46, 54, 66, 67, 91], demand management [41, 53, 99], and optimal placement of renewable energy sources [2, 20, 49, 55].
- Other Applications: PSO is also used in data mining [9, 78, 85].

The search ability of the algorithm is greatly affected by three control parameters [29, 64, 97]: inertia weight, cognitive acceleration coefficient, and social acceleration coefficient. Early studies argued for stable solutions based on the assumption that inertia weight and acceleration coefficients should be constant values. Later, Shi and Eberhart [74] pointed out the effect of a linearly decreasing function of inertia weights with respect to time (or generation), although without an expression, and Ratnaweera et al. [65] extended a similar linear time-varying concept of an inertial weight to acceleration coefficients, and introduced a "crossing strategy" in which a monotonically decreasing cognitive acceleration coefficient and a monotonically increasing social acceleration coefficient are crossed during exploration to balance local exploitation and global exploration in exploration. However, the functional expressions of an inertia weight and acceleration coefficients are not identical, and the latter uses a simplified expression of the former. Chatterjee and Siarry [11] proposed a nonlinear function based on a linear decreasing function representation due to Shi and Eberhart [74] for faster solution convergence and better search accuracy. At about the same time, Zhang et al. [101] proposed nonlinear inertia weights with a logical switching rule, and Chen et al. [12] used nonlinear time-varying inertia weights with the square of the normalized time as a basic function. Xin et al. [92] proposed a multi-stage linearly-decreasing inertia weight method to achieve a better balance between global and local search, using the inertia weight representation used by Ratnaweera et al. [65]. Thus, various other nonlinear time-varying functions, exponential functions, and arbitrary adaptive functions have been proposed (see also a survey paper by Shami et al. [70]).

On the other hand, the cognitive and social acceleration coefficients were often set to constant values of 1.5 or 2.0 in early studies [15, 40, 73], so that there has been little discussion on its importance compared to the inertia weight. However, the time-varying inertia parameters inspired the proposal of linear time-varying acceleration coefficients [1, 65]. Subsequently, a nonlinear version of those of Ratnaweera et al. [65] as found in [86], those using various trigonometric functions [13, 36, 94], those of applying evaluation functions or related quantities to construct nonlinear time-varying acceleration coefficients [75, 90, 93], and even those using random acceleration coefficients [17, 34, 89, 107] have been proposed.

It should be noted, however, that there has been little systematic discussion on how to design or construct time-varying acceleration coefficients, including inertia weights. In this study, we present a unified design method for a nonlinear time-varying inertia weight or acceleration coefficients based on several time-related nonlinear functions, which are scaled linearly while satisfying initial and terminal boundary values. First, the normalized time is assumed to be the independent variable, and the basic functions are assumed to be monotonically decreasing or monotonically increasing functions that satisfy known initial and terminal values. However, when the basis function does not satisfy the specified boundary values (i.e., it is ill-posed), we propose to pre-compute fictitious boundary values for the inertia weight or acceleration coefficients and modify the inertia weight or acceleration coefficients so that they satisfy the actual specified boundary conditions.

Next, it is shown that in the crossing strategy where the cognitive acceleration coefficients and social acceleration coefficients are crossed during the search, if the relationship between the basic function and the ill-posed or well-posed boundary value is obtained, the crossing value and crossing time can be known in advance by using it. In particular, for asymmetric crossing strategies where the initial and terminal errors of the two acceleration coefficients are different in magnitude, it is pointed out that the crossing time of the two acceleration coefficients can be changed according to the ratio of the initial error. Furthermore, by introducing a parameter in the basic function that adjusts the shape of the function, it is possible to shift the crossing time in a symmetric crossing strategy where the magnitudes of the initial and terminal errors of the two acceleration coefficients are equal by changing this parameter, resulting in the ability to adjust the timing of convergence to the global best value. The results show that the convergence timing to the global best value can be adjusted by changing this parameter.

The effectiveness of the proposed algorithms (i.e., nonlinear crossing strategy-based PSOs: NCS-PSOs) is evaluated on CEC2014 (Congress on Evolutionary Computation in 2014) benchmark functions. That is, statistical analysis, Wilcoxon test, and Friedman test are used to evaluate optimization performance of NCS-PSOs, and the experimental results are compared with four state-of-the-art metaheuristics, as well as three classical PSO variants. A practical constraint programming of NCS-PSOs is studied for dealing with a 3D path optimization problem.

The structure of this paper is described below. Section 2 reviews how to design monotonically decreasing time-varying parameters that satisfy specified initial and terminal boundary values by linearly amplifying a linear basic function using monotonically increasing or decreasing functions with respect to normalized time. The basic functions are assumed to be well-posed, satisfying both boundary values, and monotonicity is guaranteed. In Section 3, when extending this concept to nonlinear basic functions that do not satisfy the boundary conditions, we show how to introduce fictitious boundary values. Section 4 derives the relationship between the basic functions and the boundary values in the crossing-strategy method for the two acceleration coefficients, so as to obtain the crossing values and crossing times for basic functions. Specific crossing times for the 10 selected basic functions are given in Section 5. In Section 6, the effectiveness of the proposed algorithm is evaluated on CEC2014 benchmark functions. After explaining a path optimization problem in Section 7, the effectiveness of the proposed method is also verified in Section 8 through an example of an actual optimization problem, in which six different nonlinear time-varying acceleration coefficients are used in 3D path planning for a drone, one using a series of normalized time as the basic function and several exponential functions with Napier's constant as the base. The conclusion is given in Section 9.

2 Review of PSO and the design method of inertia weight with a linear time-varying function

2.1 Review of PSO

PSO is a search algorithm based on a group, and starts in the initial group of the solution which is called particles and which occurred at random. Each particle i in PSO is characterized by the position $\boldsymbol{x}_i(t) = [x_{i1}(t)\ x_{i2}(t)\ ,\ldots,x_{iN}]^T$ and the velocity $\boldsymbol{v}_i(t) = [v_{i1}(t)\ v_{i2}(t)\ ,\ldots,v_{iN}]^T$, where $i = 1,\ldots,M$, M makes the number of particles and N denotes the number of dimensions of search space. The optimal solution is performed by the following algorithms.

$$\boldsymbol{v}_i(t+1) = w(t)\boldsymbol{v}_i(t) + c_1(t)\boldsymbol{r}_{1i}(t)\circ[\boldsymbol{p}_i(t) - \boldsymbol{x}_i(t)] + c_2(t)\boldsymbol{r}_{2i}(t)\circ[\boldsymbol{g}(t) - \boldsymbol{x}_i(t)] \tag{1}$$

$$\boldsymbol{x}_i(t+1) = \boldsymbol{x}_i(t) + \boldsymbol{v}_i(t+1) \tag{2}$$

Here, $w(t)$ is the inertia weight, $c_1(t)$ and $c_2(t)$ are respectively the cognitive and social coefficients (or acceleration coefficients), and $\boldsymbol{r}_{1i}(t)$ and $\boldsymbol{r}_{2i}(t)$ are two random vectors within [0, 1] taken from a uniform probability distribution. Moreover, the "$\circ$" sign is an operator denoting a product for every element between vectors; $\boldsymbol{p}_i(t) = [p_{i1}(t)\ p_{i2}(t)\ ,\ldots,p_{iN}(t)]^T$ is the past best position vector of the ith particle; and $\boldsymbol{g}(t) = [g_1(t)\ g_2(t)\ ,\ldots,g_N(t)]^T$ is the best position vector discovered by all the old particles.

2.2 Review of how to make a monotonically decreasing function with any known boundary values

Here, according to a monotonically increasing or decreasing basic function that includes a time variable, let us review how to make an inertia weight to be a monotonically decreasing function with any known initial and terminal values. However, the time is assumed to be normalized by the maximum time $t_{\max}$. For example, consider a linear function about time t, which is described by

$$y(t) = t/t_{\max} \triangleq f(t, t_{\max}) \tag{3}$$

Since this function is a monotonically increasing function whose initial and terminal values fill respectively $f(t_0, t_{\max}) = 0$ and $f(t_{\max}, t_{\max}) = 1$ (i.e., it has well-posed boundary values), if $t_0 = 0$, in order to make a monotonically decreasing function, let us transform it into

$$y(t) = 1 - t/t_{\max} = 1 - f(t, t_{\max}) \tag{4}$$

Then, this monotonically decreasing function is scaled up to the function $w(t)$ whose initial constant point is w_s and terminal constant point is w_e. To this end, multiplying (4) by $(w_s - w_e)$ gives

$$y_1(t) = (w_s - w_e)[1 - f(t, t_{\max})] \tag{5}$$

where this $y_1(t)$ becomes $w_s - w_e$ if $t = 0$, whereas 0 if $t = t_{\max}$. In addition, letting $y_2(t) = y_1(t) + w_e$ yields that

$$y_2(t) = (w_s - w_e)[1 - f(t, t_{\max})] + w_e \triangleq w(t) \tag{6}$$

This clearly becomes w_s if $t = 0$, while w_e if $t = t_{\max}$.

On the other hand, applying a monotonically decreasing function $g(t, t_{\max})$ with well-posed boundary values, such as $g(t_0, t_{\max}) = 1$ and $g(t_{\max}, t_{\max}) = 0$, to make $w(t)$ like the previous one, it follows, from the above result, immediately that

$$w(t) = (w_s - w_e)g(t, t_{\max}) + w_e \tag{7}$$

Note that, from the equation of a line passing through two points $(0, w_s)$ and $(t_{\max}, w_e)$, (6) can also be derived as

$$w(t) - w_s = \frac{(w_e - w_s)}{t_{\max}}t \tag{8}$$

Rewriting the expression that uses a monotonically increasing function $t/t_{\max}$ in this equation by the expression that uses a monotonically decreasing function $(1 - t/t_{\max})$, i.e.,

$$w(t) = w_1(1 - t/t_{\max}) + w_2 \tag{9}$$

comparing them, it follows readily that $w_1 = w_s - w_e$ and $w_2 = w_e$.

3 Calculation of fictitious boundary values

Since $w(t)$, $c_1(t)$ and $c_2(t)$ are constituted in this paper using the form of time-varying parameter $a(t)$ which is described later, consider designing or selecting $f(t, t_{\max})$ or $g(t, t_{\max})$ so that it may serve as a monotonically decreasing function or a monotonically increasing function.

$$a(t) = (a_s - a_e)[1 - f(t, t_{\max})] + a_e \tag{10}$$

$$\quad = (a_s - a_e)g(t, t_{\max}) + a_e \tag{11}$$

Here, if $f(t, t_{\max})$ is considered as a monotonically increasing function with a well-posed boundary value, it should be satisfied such that $f(t_0, t_{\max}) = 0$ and $f(t_{\max}, t_{\max}) = 1$. Similarly, if $g(t, t_{\max})$ is regarded as a monotonically decreasing function with a well-posed boundary value, it should be subjected to $g(t_0, t_{\max}) = 1$ and $g(t_{\max}, t_{\max}) = 0$. Note here that when the monotonic increasing function $f(t)$ is considered, if $(a_s - a_e) > 0$ according to (10), then $1 - f(t)$ becomes a monotonically decreasing function, resulting in a monotonically decreasing parameter for $a(t)$, whereas $if (a_s - a_e) < 0$, then $a(t)$ becomes a monotonically increasing parameter instead. Similarly, for the monotonic decreasing function $g(t)$, if $(a_s - a_e) > 0$ according to (11), then $a(t)$ immediately becomes a monotonically decreasing parameter, where $if (a_s - a_e) < 0$, it becomes a monotonically increasing parameter instead.

However, in the selection of realistic $f(\cdot)$ or $g(\cdot)$, we often encounter the case that a terminal boundary value is not fulfilled though an initial boundary value is generally fulfilled, or that a terminal boundary value is fulfilled vice versa, but an initial boundary value is not fulfilled. In that case, the time-varying parameter in the above form becomes $a(t)|_{t=t_0} \neq a_s$ or $a(t)|_{t=t_{\max}} \neq a_e$, so that the exact estimate or setup of the parameter value in the starting point or the terminal point can not be performed. When the basic functions, $f(\cdot)$ and $g(\cdot)$, are ill-posed which do not fulfill the appointed boundary values, reset the actual initial boundary value to be specified as a_{ini} or the actual terminal boundary value to be given as a_{ter}. Then, the original a_s and a_e are regarded as a fictitious initial value or a fictitious termination value, respectively, and they are only set up as the intermediate variables, i.e., they are just accompanied to the equation of the parameter $a(t)$.

3.1 For the case of an initially ill-posed boundary value

When the basic function is to be a monotonically increasing function $f(t, t_{\max})$ with $f(t_0, t_{\max}) \neq 0$: For this case, it follows from (10) that

$$a(t)|_{t=t_0} = (a_e - a_s)f(t_0, t_{\max}) + a_s \overset{\triangle}{=} a_{ini}$$

or

$$a_{ini} = [1 - f(t_0, t_{\max})]a_s + f(t_0, t_{\max})a_e$$

Given a_{ini}, solving it about a_s yields

$$a_s = \frac{a_{ini} - a_e f(t_0, t_{\max})}{1 - f(t_0, t_{\max})} \tag{12}$$

Thus, regarding a_s as a fictitious initial boundary value, it can be realized that $a(t)|_{t=t_0} = a_{ini}$.

When the basic function is to be a monotonically decreasing function $g(t, t_{\max})$ with $g(t_0, t_{\max}) \neq 1$: For this case, it follows from (11) that

$$a(t)|_{t=t_0} = (a_s - a_e)g(t_0, t_{\max}) + a_e \overset{\triangle}{=} a_{ini}$$

or

$$a_{ini} = a_s g(t_0, t_{\max}) - a_e[g(t_0, t_{\max}) - 1]$$

Given a_{ini}, solving it about a_s yields

$$a_s = \frac{a_{ini} + a_e[g(t_0, t_{\max}) - 1]}{g(t_0, t_{\max})} = \frac{a_{ini}}{g(t_0, t_{\max})} + a_e\left(1 - \frac{1}{g(t_0, t_{\max})}\right) \tag{13}$$

Thus, regarding a_s as a fictitious initial boundary value, the given value a_{ini} can be assigned as the actual initial value such that $a(t)|_{t=t_0} = a_{ini}$, instead of using a_s.

3.2 For the case of a terminally ill-posed boundary value

When the basic function is to be a monotonically increasing function $f(t, t_{\max})$ with $f(t_{\max}, t_{\max}) \neq 1$: For this case, it follows from (10) that

$$a(t)|_{t=t_{\max}} = (a_e - a_s) f(t_{\max}, t_{\max}) + a_s \overset{\triangle}{=} a_{ter}$$

or

$$a_{ter} = [1 - f(t_{\max}, t_{\max})]a_s + f(t_{\max}, t_{\max})a_e$$

Given a_{ter}, solving it about a_e yields

$$\begin{aligned} a_e &= \frac{a_{ter} - [1 - f(t_{\max}, t_{\max})]a_s}{f(t_{\max}, t_{\max})} \\ &= \frac{a_{ter}}{f(t_{\max}, t_{\max})} + a_s \left(1 - \frac{1}{f(t_{\max}, t_{\max})}\right) \end{aligned} \tag{14}$$

Thus, regarding a_e as a fictitious terminal boundary value, it can be realized that $a(t)|_{t=t_{\max}} = a_{ter}$.

When the basic function is to be a monotonically decreasing function $g(t, t_{\max})$ with $g(t_{\max}, t_{\max}) \neq 0$: For this case, it follows from (11) that

$$a(t)|_{t=t_{\max}} = (a_s - a_e) g(t_{\max}, t_{\max}) + a_e \overset{\triangle}{=} a_{ter}$$

or

$$a_{ter} = a_s g(t_{\max}, t_{\max}) + a_e[1 - g(t_{\max}, t_{\max})]$$

Given a_{ter}, solving it about a_e yields

$$a_e = \frac{a_{ter} - a_s g(t_{\max}, t_{\max})}{1 - g(t_{\max}, t_{\max})} \tag{15}$$

Thus, regarding a_e as a fictitious terminal boundary value, the given value a_{ter} can be assigned as the actual terminal value such that $a(t)|_{t=t_{\max}} = a_{ter}$, instead of using a_e.

3.3 Examples of fictitious boundary values

Some fictitious boundary values are shown in several basic functions.

3.3.1 $G(t, t_{\max}) = \tan^{-1}(\frac{t_{\max}-t}{t_{\max}})$

This function is of used for constituting $w(t)$ or $c_1(t)$ in Jiang et al. [36], and

$$\tan^{-1}\left(\frac{t_{\max} - t}{t_{\max}}\right)\bigg|_{t=0} = \frac{\pi}{4}, \quad \tan^{-1}\left(\frac{t_{\max} - t}{t_{\max}}\right)\bigg|_{t=t_{\max}} = 0$$

Since

$$\frac{d\left[\tan^{-1}\frac{t_{\max}-t}{t_{\max}}\right]}{dt} = -\frac{1}{t_{\max}} \frac{1}{1 + \left(\frac{t_{\max}-t}{t_{\max}}\right)^2} < 0$$

it is a monotonically decreasing function whose initial boundary value is ill-posed. Thus, it is found from (13) that the fictitious initial boundary value is reduced to

$$\begin{aligned} a_s &= \frac{a_{ini}}{\frac{\pi}{4}} + a_e\left(1 - \frac{1}{\frac{\pi}{4}}\right) \\ &= \frac{4}{\pi}(a_{ini} - a_e) + a_e \end{aligned} \tag{16}$$

Note that this function can have a well-posed boundary value, if it is modified as $\frac{4}{\pi}\tan^{-1}(\frac{t_{\max}-t}{t_{\max}})$ by visual inspection.

3.3.2 $G(t, t_{\max}) = \cos(\pi t/t_{\max})$

This function is used in Yang et al. [94] to make time-varying parameters $c_1(t)$ and $c_2(t)$. Since

$$\frac{d\cos(\pi t/t_{\max})}{dt} = -\frac{\pi}{t_{\max}}\sin(\pi t/t_{\max}) < 0$$

it is found that this function is to be a monotonically decreasing function, but the terminal boundary value is reduced to be ill-posed:

$$\cos(\pi t/t_{\max})|_{t=0} = 1, \quad \cos(\pi t/t_{\max})|_{t=t_{\max}} = -1$$

Thus, from (15), the fictitious terminal boundary value results in

$$a_e = \frac{a_{ter} - a_s(-1)}{1 - (-1)} = \frac{a_{ter} + a_s}{2} \tag{17}$$

Note that this function can take a well-posed terminal boundary value, if it is modified as $\frac{1}{2}[\cos(\pi t/t_{\max}) + 1]$ by some rearrangements.

3.3.3 $F(t, t_{\max}) = e^{-(t_{\max}-t)/t_{\max}}$

From the fact that

$$\frac{dF(t, t_{\max})}{dt} = \frac{1}{t_{\max}} F(t, t_{\max}) > 0$$

this monotonically increasing function provides $F(0, t_{\max}) = 1/e$ and $F(t_{\max}, t_{\max}) = 1$, so that the initial boundary value is to be ill-posed. Thus, from (12), the fictitious initial boundary value beomes

$$a_s = \frac{a_{ini} - a_e/e}{1 - 1/e} = \left(\frac{e}{e-1}\right) a_{ini} - \frac{a_e}{e-1} \tag{18}$$

3.3.4 $G(t, t_{\max}) = e^{-t_{\max}/(t_{\max}-t)}$

Since

$$\frac{dG(t, t_{\max})}{dt} = -\frac{1}{t_{\max}(1 - t/t_{\max})^2} G(t, t_{\max}) < 0$$

this monotonically decreasing function provides $G(0, t_{\max}) = 1/e$ and $G(t_{\max}, t_{\max}) = 0$, so that it has an ill-posed initial boundary value. Therefore, it is found from (13) that the fictitious initial boundary value results in

$$a_s = \frac{a_{ini}}{1/e} + a_e\left(1 - \frac{1}{1/e}\right) = e a_{ini} + a_e(1 - e) \tag{19}$$

3.3.5 $G(t, t_{\max}) = e^{-\alpha t/t_{\max}}$

Becuase of

$$\frac{dG(t, t_{\max})}{dt} = -\frac{\alpha}{t_{\max}} G(t, t_{\max}) < 0$$

this monotonically decreasing function gives $G(0, t_{\max}) = 1$ and $G(t_{\max}, t_{\max}) = 1/e^{\alpha}$, so that the terminal boundary value is not well-posed. Thus, it is found from (15) that the fictitious terminal boundary value is reduced to

$$a_e = \frac{a_{ter} - a_s/e^{\alpha}}{1 - 1/e^{\alpha}} = \left(\frac{e^{\alpha}}{e^{\alpha}-1}\right) a_{ter} - \frac{a_s}{e^{\alpha}-1} \tag{20}$$

3.3.6 $F(t, t_{\max}) = e^{-\alpha t_{\max}/t}$

Since

$$\frac{dF(t, t_{\max})}{dt} = -\frac{\alpha t_{\max}}{t^2} F(t, t_{\max}) > 0$$

this monotonically increasing function provides $F(0, t_{\max}) = 0$ and $F(t_{\max}, t_{\max}) = 1/e^{\alpha}$, so that the terminal boundary value is ill-posed. Therefore, from (14), the fictitious terminal boundary value is found to be

$$a_e = \frac{a_{ter}}{1/e^{\alpha}} + a_s\left(1 - \frac{1}{1/e^{\alpha}}\right) = e^{\alpha} a_{ter} + a_s(1 - e^{\alpha}) \tag{21}$$

4 Relationship between basic function and boundary values when adopting the crossing strategy of acceleration coefficients

In the crossing strategy proposed in this paper using nonlinear time-varying decreasing function $c_1(t)$ and increasing function $c_2(t)$ which are acceleration coefficients, the crossing time t_{cross} and crossing value c_{cross} are introduced as the amount of information in connection with the timing of intersection. Therefore, the relationship between the basic function and boundary values is derived in advance here when adopting a crossing strategy. It is roughly classified into four cases, depending on whether the basic function to be used is monotonically increasing or monotonically decreasing, or whether the initial boundary condition is ill-posed or the terminal boundary condition is ill-posed.

4.1 When the basic function is a monotonically increasing function with an ill-posed initial boundary condition

4.1.1 An asymmetric case about a crossing axis

When the acceleration coefficients $c_1(t)$ and $c_2(t)$ are asymmetrical about a crossing axis, using a fictitious initial boundary value in which the basic function is monotonically increasing with an ill-posed initial boundary condition, i.e., $f(t_0, t_{\max}) \neq 0$, $c_1(t)$ and $c_2(t)$ are respectively expressed as follows:

$$c_{1s} = \frac{c_{1ini} - c_{1e} f(t_0, t_{\max})}{1 - f(t_0, t_{\max})} \tag{22}$$

$$\begin{aligned} c_1(t) &= (c_{1e} - c_{1s}) f(t, t_{\max}) + c_{1s} \\ &= \left(c_{1e} - \frac{c_{1ini} - c_{1e} f(t_0, t_{\max})}{1 - f(t_0, t_{\max})}\right) f(t, t_{\max}) + c_{1s} \\ &= \left(\frac{c_{1e} - c_{1ini}}{1 - f(t_0, t_{\max})}\right) f(t, t_{\max}) \\ &\quad + \frac{c_{1ini} - c_{1e} f(t_0, t_{\max})}{1 - f(t_0, t_{\max})} \end{aligned} \tag{23}$$

$$c_{2s} = \frac{c_{2ini} - c_{2e} f(t_0, t_{\max})}{1 - f(t_0, t_{\max})} \tag{24}$$

$$
\begin{aligned}
c_2(t) &= (c_{2e} - c_{2s}) f(t, t_{\max}) + c_{2s} \\
&= \left(c_{2e} - \frac{c_{2ini} - c_{2e} f(t_0, t_{\max})}{1 - f(t_0, t_{\max})} \right) f(t, t_{\max}) + c_{2s} \\
&= \left(\frac{c_{2e} - c_{2ini}}{1 - f(t_0, t_{\max})} \right) f(t, t_{\max}) \\
&\quad + \frac{c_{2ini} - c_{2e} f(t_0, t_{\max})}{1 - f(t_0, t_{\max})} \qquad (25)
\end{aligned}
$$

Equating (23) and (25), assuming that $f(t_0, t_{\max}) \neq 0$ and $f(t_0, t_{\max}) \neq 1$, it obtains

$$
\begin{aligned}
&[(c_{2e} - c_{1e}) + (c_{1ini} - c_{2ini})] f(t, t_{\max}) \\
&\quad = (c_{1ini} - c_{2ini}) + (c_{2e} - c_{1e}) f(t_0, t_{\max})
\end{aligned}
$$

consequently it follows that

$$
f(t, t_{\max}) = \beta + (1 - \beta) f(t_0, t_{\max}) \qquad (26)
$$

Here,

$$
\beta = \frac{c_{12ini}}{c_{21e} + c_{12ini}} \qquad (27)
$$

in which c_{12ini} denotes the actual initial error between two acceleration coefficients $c_1(t)$ and $c_2(t)$, c_{21e} is its terminal error, which are given by

$$
c_{12ini} = c_{1ini} - c_{2ini}, \quad c_{21e} = c_{2e} - c_{1e} \qquad (28)
$$

where $c_{12ini} \neq 0$ and $c_{21e} \neq 0$. Note here that $c_1(t)$ and $c_2(t)$ are asymmetrical means also that the initial acceleration coefficient error differs from the termination acceleration coefficient error. If $f(t_0, t_{\max}) \equiv 0$, i.e., the initial boundary condition is well-posed, then (26) is reduced to

$$
f(t, t_{\max}) = \beta \qquad (29)
$$

Before discussing the symmetry of the crossing axis for the coefficients $c_1(t)$ and $c_2(t)$, we establish the relationship between the acceleration coefficient errors c_{12ini} at the initial point and c_{21e} at the terminal point to achieve this symmetry. In general, for the two time-varying coefficients $c_1(t)$ and $c_2(t)$ to be axisymmetric with respect to the time (or generation) axis for crossing strategies, they must intersect at least once along the time axis. This intersection should exhibit the property of symmetric movement, where the symmetry axis (crossing axis) becomes the perpendicular bisector of the line connecting corresponding points. The conditions for ensuring the symmetry of acceleration coefficients when the initial values are ill-posed can be categorized into three cases, introducing the ratio of acceleration coefficient errors at the initial point and the terminal point as:

$$
\eta_{ii} = \frac{c_{12ini}}{c_{21e}}
$$

1. Case of $\eta_{ii} = 1$ (i.e., $c_{12ini} = c_{21e}$): Additionally, the following conditions are necessary:

$$
c_{1ini} - c_{cross} = c_{cross} - c_{2ini} \quad \text{or} \quad c_{2e} - c_{cross} = c_{cross} - c_{1e}
$$

2. Case of $\eta_{ii} > 1$: Further conditions are required:

$$
c_{1ini} - c_{cross} = c_{cross} - c_{2ini}; \quad c_{2e} - c_{cross} = c_{cross} - c_{1e}
$$

where $c_{1ini} > c_{2e}$ or $c_{1e} > c_{2ini}$.

3. Case of $0 < \eta_{ii} < 1$: Additional conditions apply:

$$
c_{1ini} - c_{cross} = c_{cross} - c_{2ini}; \quad c_{2e} - c_{cross} = c_{cross} - c_{1e}
$$

where $c_{2e} > c_{1ini}$ or $c_{2ini} > c_{1e}$. In this discussion, we primarily focus on the case where $\eta_{ii} = 1$. Furthermore, when the initial values are well-posed, we can design two axisymmetric acceleration coefficients by setting $c_{1ini} = c_{1s}$, $c_{2ini} = c_{2s}$, and $c_{12ini} = c_{12s}$ (i.e., $c_{12s} = c_{1s} - c_{2s}$) in the above equations.

4.1.2 A symmetric case about a crossing axis

When $c_1(t)$ and $c_2(t)$ are symmetric about the crossing axis, or the initial acceleration coefficient error and the terminal acceleration error are equivalent, it obtains the condition of $c_{12ini} = c_{21e}$, so that (26) becomes

$$
f(t, t_{\max}) = \frac{1}{2}(1 + f(t_0, t_{\max})) \qquad (30)
$$

Moreover, if the initial boundary condition is well-posed, then it follows that

$$
f(t, t_{\max}) = \frac{1}{2} \qquad (31)
$$

4.1.3 Crossing value

Here, a crossing value c_{cross} is solved, when using a crossing strategy. Substituting the relationship between the basic function and the initial boundary value in the asymmetrical crossing method, i.e., (26) into (23), letting $c_1(t)$ at that time

be c_{cross}, it follows that

$$\begin{aligned} c_{cross} &= \frac{(c_{1e}-c_{1ini})[\beta+(1-\beta)f(t_0,t_{\max})]}{1-f(t_0,t_{\max})} \\ &\quad +\frac{c_{1ini}-c_{1e}f(t_0,t_{\max})}{1-f(t_0,t_{\max})} \\ &= \frac{(c_{1e}-c_{1ini})\beta+c_{1ini}}{1-f(t_0,t_{\max})} \\ &\quad +\frac{[(c_{1e}-c_{1ini})(1-\beta)-c_{1e}]f(t_0,t_{\max})}{1-f(t_0,t_{\max})} \end{aligned}$$

Noting that $(c_{1e}-c_{1ini})(1-\beta)-c_{1e}=-[(c_{1e}-c_{1ini})\beta+c_{1ini}]$, the above equation is reduced to

$$\begin{aligned} c_{cross} &= \frac{[(c_{1e}-c_{1ini})\beta+c_{1ini}](1-f(t_0,t_{\max}))}{1-f(t_0,t_{\max})} \\ &= (c_{1e}-c_{1ini})\beta+c_{1ini} \end{aligned} \tag{32}$$

Since $c_{1ini}=c_{2e}$ and $\beta=1/2$ in a symmetrical crossing strategy, it follows that

$$c_{cross}=\frac{1}{2}(c_{1ini}+c_{1e})\equiv\frac{1}{2}(c_{1e}+c_{2e}) \tag{33}$$

or using the fact of $c_{1e}=c_{2ini}$, it is found that

$$c_{cross}=\frac{1}{2}(c_{1ini}+c_{2ini})\equiv\frac{1}{2}(c_{2ini}+c_{2e}) \tag{34}$$

Figure 1 summarizes the fictious initial values a_s, crossing time t_{cross} and crossing value c_{cross} when the monotonically increasing function $f(t,t_{\max})$ is initially ill-posed. However, t_{cross} must be obtained after the concrete $f(t,t_{\max})$ is given. When $f(t,t_{\max})$ is initially well-posed, it should take that $f(t_0,t_{\max})\equiv 0$. Note also that the symmetric cross strategy denotes only for the case of $c_{12ini}=c_{21e}$.

4.2 When the basic function is a monotonically decreasing function with an ill-posed initial boundary condition

4.2.1 An asymmetric case about a crossing axis

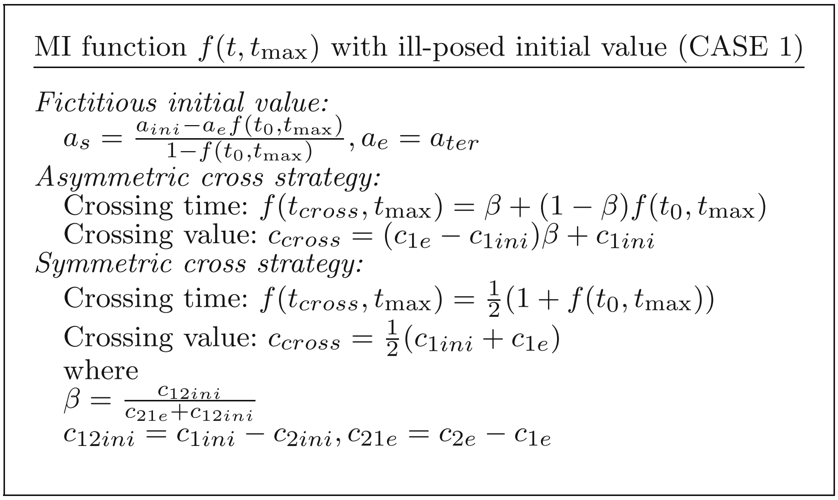
When the acceleration coefficients $c_1(t)$ and $c_2(t)$ are asymmetrical about a crossing axis, using a fictitious initial boundary value in which the basic function is monotonically decreasing with an ill-posed initial boundary condition, i.e., $g(t_0,t_{\max})\neq 1$ and $g(t_0,t_{\max})\neq 0$, $c_1(t)$ and $c_2(t)$ are respectively expressed as follows:

$$c_{1s}=\frac{c_{1ini}}{g(t_0,t_{\max})}+c_{1e}\left(1-\frac{1}{g(t_0,t_{\max})}\right) \tag{35}$$

$$\begin{aligned} c_1(t) &= (c_{1s}-c_{1e})g(t,t_{\max})+c_{1e} \\ &= \left(\frac{c_{1ini}}{g(t_0,t_{\max})}+c_{1e}-\frac{c_{1e}}{g(t_0,t_{\max})}-c_{1e}\right) \\ &\quad \times g(t,t_{\max})+c_{1e} \\ &= (c_{1ini}-c_{1e})g(t,t_{\max})/g(t_0,t_{\max})+c_{1e} \end{aligned} \tag{36}$$

$$c_{2s}=\frac{c_{2ini}}{g(t_0,t_{\max})}+c_{2e}\left(1-\frac{1}{g(t_0,t_{\max})}\right) \tag{37}$$

$$\begin{aligned} c_2(t) &= (c_{2s}-c_{2e})g(t,t_{\max})+c_{2e} \\ &= \left(\frac{c_{2ini}}{g(t_0,t_{\max})}+c_{2e}-\frac{c_{2e}}{g(t_0,t_{\max})}-c_{2e}\right) \\ &\quad \times g(t,t_{\max})+c_{2e} \\ &= (c_{2ini}-c_{2e})g(t,t_{\max})/g(t_0,t_{\max})+c_{2e} \end{aligned} \tag{38}$$



MI function $f(t,t_{\max})$ with ill-posed initial value (CASE 1)

Fictitious initial value:

$a_s=\frac{a_{ini}-a_e f(t_0,t_{\max})}{1-f(t_0,t_{\max})}, a_e=a_{ter}$

Asymmetric cross strategy:

Crossing time: $f(t_{cross},t_{\max})=\beta+(1-\beta)f(t_0,t_{\max})$

Crossing value: $c_{cross}=(c_{1e}-c_{1ini})\beta+c_{1ini}$

Symmetric cross strategy:

Crossing time: $f(t_{cross},t_{\max})=\frac{1}{2}(1+f(t_0,t_{\max}))$

Crossing value: $c_{cross}=\frac{1}{2}(c_{1ini}+c_{1e})$

where

$\beta=\frac{c_{12ini}}{c_{21e}+c_{12ini}}$

$c_{12ini}=c_{1ini}-c_{2ini}, c_{21e}=c_{2e}-c_{1e}$

Fig. 1 The fictitious initial value a_s, crossing time t_{cross} and crossing value c_{cross}, when the monotonic-increasing (MI) function $f(t,t_{\max})$ is initially ill-posed

Equating (36) and (38), assuming $g(t_0, t_{\max}) \neq 0$, it obtains that

$$\begin{aligned}(c_{1ini} - c_{1e})g(t, t_{\max})/g(t_0, t_{\max}) + c_{1e} \\ = (c_{2ini} - c_{2e})g(t, t_{\max})/g(t_0, t_{\max}) + c_{2e}\end{aligned}$$

or

$$\begin{aligned}(c_{1ini} - c_{2ini})g(t, t_{\max})/g(t_0, t_{\max}) \\ +(c_{2e} - c_{1e})g(t, t_{\max})/g(t_0, t_{\max}) = c_{2e} - c_{1e}\end{aligned}$$

In the sequel, if $c_{12ini} \neq 0$ and $c_{21e} \neq 0$, then it is found that

$$g(t, t_{\max})/g(t_0, t_{\max}) = 1 - \beta \tag{39}$$

When $g(t_0, t_{\max}) \equiv 1$, i.e., the initial boundary condition is well-posed, it is easy to find that (39) is reduced to

$$g(t, t_{\max}) = 1 - \beta \tag{40}$$

4.2.2 A symmetric case about a crossing axis

When $c_1(t)$ and $c_2(t)$ are symmetric about the crossing axis, or the initial acceleration coefficient error and the terminal acceleration error are equivalent, i.e., $c_{12ini} = c_{21e}$, it is easy to find that (39) becomes

$$g(t, t_{\max})/g(t_0, t_{\max}) = \frac{1}{2} \tag{41}$$

Additionally, if the initial boundary condition is well-posed, i.e., $g(t_0, t_{\max}) \equiv 1$, then the above equation is reduced to

$$g(t, t_{\max}) = \frac{1}{2} \tag{42}$$

4.2.3 Crossing value

Here, the crossing value c_{cross} is derived. Substituting the relationship between the basic function and the initial boundary value in the asymmetrical crossing method, i.e., (39) into (36), letting $c_1(t)$ in that time be c_{cross}, it is obtained that

$$\begin{aligned}c_{cross} &= (c_{1ini} - c_{1e})(1 - \beta)g(t_0, t_{\max})/g(t_0, t_{\max}) + c_{1e} \\ &= c_{1ini}(1 - \beta) - c_{1e}(1 - \beta) + c_{1e} \\ &= c_{1ini}(1 - \beta) + c_{1e}\beta \\ &= (c_{1e} - c_{1ini})\beta + c_{1ini}\end{aligned} \tag{43}$$

Since $c_{1ini} = c_{2e}$ and $\beta = 1/2$ in the symmetrical crossing strategy, it is easy to find that

$$c_{cross} = \frac{1}{2}(c_{1ini} + c_{1e}) \equiv \frac{1}{2}(c_{1e} + c_{2e}) \tag{44}$$

or from the fact of $c_{1e} = c_{2ini}$ that

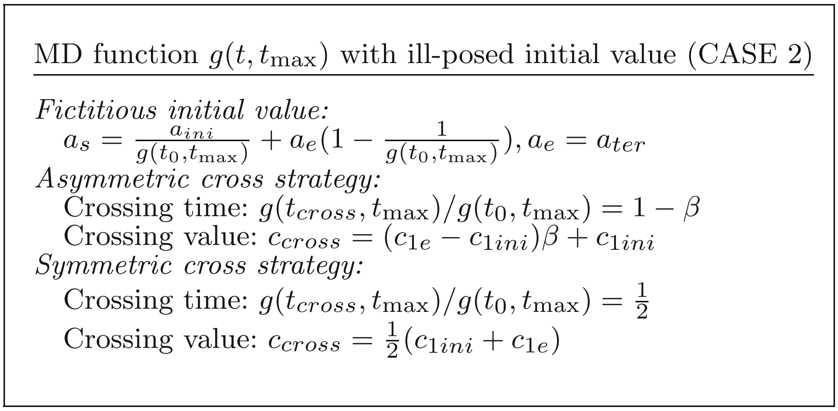
$$c_{cross} = \frac{1}{2}(c_{1ini} + c_{2ini}) \equiv \frac{1}{2}(c_{2ini} + c_{2e}) \tag{45}$$

Figure 2 summarizes the fictitious initial values a_s, crossing time t_{cross} and crossing value c_{cross} when the monotonically decreasing function $g(t, t_{\max})$ is initially ill-posed. However, t_{cross} must be obtained after the concrete $g(t, t_{\max})$ is given. When $g(t, t_{\max})$ is initially well-posed, it should take that $g(t_0, t_{\max}) \equiv 1$.

4.3 When the basic function is a monotonically decreasing function with an ill-posed terminal boundary condition

4.3.1 An asymmetric case about a crossing axis

When the acceleration coefficients $c_1(t)$ and $c_2(t)$ are asymmetrical about a crossing axis, using a fictitious terminal boundary value in which the basic function is monotonically decreasing with an ill-posed terminal boundary condition, i.e., $g(t_{\max}, t_{\max}) \neq 0$, $c_1(t)$ and $c_2(t)$ are respectively



MD function $g(t, t_{\max})$ with ill-posed initial value (CASE 2)

Fictitious initial value:
$a_s = \frac{a_{ini}}{g(t_0, t_{\max})} + a_e(1 - \frac{1}{g(t_0, t_{\max})}), a_e = a_{ter}$
Asymmetric cross strategy:
Crossing time: $g(t_{cross}, t_{\max})/g(t_0, t_{\max}) = 1 - \beta$
Crossing value: $c_{cross} = (c_{1e} - c_{1ini})\beta + c_{1ini}$
Symmetric cross strategy:
Crossing time: $g(t_{cross}, t_{\max})/g(t_0, t_{\max}) = \frac{1}{2}$
Crossing value: $c_{cross} = \frac{1}{2}(c_{1ini} + c_{1e})$

Fig. 2 The fictitious initial value a_s, crossing time t_{cross} and crossing value c_{cross}, when the monotonic-decreasing (MD) function $g(t, t_{\max})$ is initially ill-posed

expressed as follows:

$$c_{1e} = \frac{c_{1ter} - c_{1s} g(t_{\max}, t_{\max})}{1 - g(t_{\max}, t_{\max})} \tag{46}$$

$$\begin{aligned} c_1(t) &= (c_{1s} - c_{1e}) g(t, t_{\max}) + c_{1e} \\ &= \left(c_{1s} - \frac{c_{1ter} - c_{1s} g(t_{\max}, t_{\max})}{1 - g(t_{\max}, t_{\max})} \right) g(t, t_{\max}) + c_{1e} \\ &= \left(\frac{c_{1s} - c_{1ter}}{1 - g(t_{\max}, t_{\max})} \right) g(t, t_{\max}) \\ &\quad + \frac{c_{1ter} - c_{1s} g(t_{\max}, t_{\max})}{1 - g(t_{\max}, t_{\max})} \end{aligned} \tag{47}$$

$$c_{2e} = \frac{c_{2ter} - c_{2s} g(t_{\max}, t_{\max})}{1 - g(t_{\max}, t_{\max})} \tag{48}$$

$$\begin{aligned} c_2(t) &= (c_{2s} - c_{2e}) g(t, t_{\max}) + c_{2e} \\ &= \left(c_{2s} - \frac{c_{2ter} - c_{2s} g(t_{\max}, t_{\max})}{1 - g(t_{\max}, t_{\max})} \right) g(t, t_{\max}) + c_{2e} \\ &= \left(\frac{c_{2s} - c_{2ter}}{1 - g(t_{\max}, t_{\max})} \right) g(t, t_{\max}) \\ &\quad + \frac{c_{2ter} - c_{2s} g(t_{\max}, t_{\max})}{1 - g(t_{\max}, t_{\max})} \end{aligned} \tag{49}$$

Equating (47) with (49), assuming that $g(t_{\max}, t_{\max}) \neq 0$ and $g(t_{\max}, t_{\max}) \neq 1$, it follows that

$$\begin{aligned} &[(c_{1s} - c_{2s}) + (c_{2ter} - c_{1ter})] g(t, t_{\max}) \\ &\quad = (c_{2ter} - c_{1ter}) + (c_{1s} - c_{2s}) g(t_{\max}, t_{\max}) \end{aligned}$$

consequently, it results in

$$g(t, t_{\max}) = (1 - \gamma) + \gamma g(t_{\max}, t_{\max}) \tag{50}$$

where

$$\gamma = \frac{c_{12s}}{c_{12s} + c_{21ter}} \tag{51}$$

in which c_{12s} denotes the initial error of two acceleration coefficients $c_{1(t)}$ and $c_{2(t)}$, and c_{21ter} is its actual terminal error, which are respectively given by

$$c_{12s} = c_{1s} - c_{2s}, \quad c_{21ter} = c_{2ter} - c_{1ter} \tag{52}$$

Here, assume that $c_{12s} \neq 0$ and $c_{21ter} \neq 0$. If $g(t_{\max}, t_{\max}) \equiv 0$, i.e., the terminal boundary condition is well-posed, then (50) becomes

$$g(t, t_{\max}) = 1 - \gamma \tag{53}$$

In a manner similar to the case of ill-posed initial boundary conditions, we also present analogous conditions to achieve symmetry for the two acceleration coefficients here. When the terminal value is ill-posed, three cases can be considered by introducing the ratio of acceleration coefficient errors at the initial point and at the terminal point as:

$$\eta_{ti} = \frac{c_{12s}}{c_{21ter}}$$

The following conditions apply for each case:

1. When $\eta_{ti} = 1$ (i.e., $c_{12s} = c_{21ter}$), the additional requirement is:

$$c_{1s} - c_{cross} = c_{cross} - c_{2s} \quad \text{or} \quad c_{2ter} - c_{cross} = c_{cross} - c_{1ter}$$

2. When $\eta_{ti} > 1$, the additional conditions are:

$$c_{1s} - c_{cross} = c_{cross} - c_{2s} \quad \text{and } c_{2ter} - c_{cross} = c_{cross} - c_{1ter}$$

   where $c_{1s} > c_{2ter}$ or $c_{1ter} > c_{2s}$.
3. When $0 < \eta_{ti} < 1$, the additional conditions are:

$$c_{1s} - c_{cross} = c_{cross} - c_{2s} \quad \text{and } c_{2ter} - c_{cross} = c_{cross} - c_{1ter}$$

where $c_{2ter} > c_{1s}$ or $c_{2s} > c_{1ter}$. In this paper, we primarily focus on the case where $\eta_{ti} = 1$.

4.3.2 A symmetric case about a crossing axis

When $c_1(t)$ and $c_2(t)$ are symmetric about the crossing axis, or the initial acceleration coefficient error and the terminal acceleration error are equivalent, i.e., $c_{12s} = c_{21ter}$, it is easy to find that (50) becomes

$$g(t, t_{\max}) = \frac{1}{2}(1 + g(t_{\max}, t_{\max})) \tag{54}$$

In addition, if the terminal boundary condition is well-posed, then it is easy to find that

$$g(t, t_{\max}) = \frac{1}{2} \tag{55}$$

4.3.3 Crossing value

The crossing value in the crossing strategy is derived here. Directly substituting the relationship between the basic function and the terminal boundary value in the asymmetric crossing method, i.e., (50) into (47), letting $c_1(t)$ in that time

be c_{cross}, it follows that

$$\begin{aligned} c_{cross} &= \frac{(c_{1s} - c_{1ter})[(1-\gamma) + \gamma g(t_{\max}, t_{\max})]}{1 - g(t_{\max}, t_{\max})} \\ &\quad + \frac{c_{1ter} - c_{1s} g(t_{\max}, t_{\max})}{1 - g(t_{\max}, t_{\max})} \\ &= \frac{(c_{1s} - c_{1ter})(1-\gamma) + c_{1ter}}{1 - g(t_{\max}, t_{\max})} \\ &\quad + \frac{[(c_{1s} - c_{1ter})\gamma - c_{1s}] g(t_{\max}, t_{\max})}{1 - g(t_{\max}, t_{\max})} \end{aligned}$$

Here, noting that $(c_{1s} - c_{1ter})\gamma - c_{1s} = -[(c_{1s} - c_{1ter})(1-\gamma) + c_{1ter}]$, the above equation is reduced to

$$\begin{aligned} c_{cross} &= \frac{[(c_{1s} - c_{1ter})(1-\gamma) + c_{1ter}](1 - g(t_{\max}, t_{\max}))}{1 - g(t_{\max}, t_{\max})} \\ &= (c_{1s} - c_{1ter})(1-\gamma) + c_{1ter} \\ &= (c_{1ter} - c_{1s})\gamma + c_{1s} \end{aligned} \tag{56}$$

Since $c_{1ter} = c_{2s}$ and $\gamma = 1/2$ in the symmetric crossing strategy, it is easy to find that

$$c_{cross} = \frac{1}{2}(c_{1ter} + c_{1s}) \equiv \frac{1}{2}(c_{1s} + c_{2s}) \tag{57}$$

or from $c_{1s} = c_{2ter}$ that

$$c_{cross} = \frac{1}{2}(c_{1ter} + c_{2ter}) \equiv \frac{1}{2}(c_{2ter} + c_{2s}) \tag{58}$$

Figure 3 summarizes the fictitious terminal value a_e, crossing time t_{cross} and crossing value c_{cross} when the monotonically decreasing function $g(t, t_{\max})$ is terminally ill-posed. However, t_{cross} must be obtained after the concrete $g(t, t_{\max})$ is given. When $g(t, t_{\max})$ is terminally well-posed, it should take that $g(t_{\max}, t_{\max}) \equiv 0$. Note also that the symmetric cross strategy denotes only for the case of $c_{12s} = c_{21ter}$.

4.4 When the basic function is a monotonically increasing function with an ill-posed terminal boundary condition

4.4.1 An asymmetric case about a crossing axis

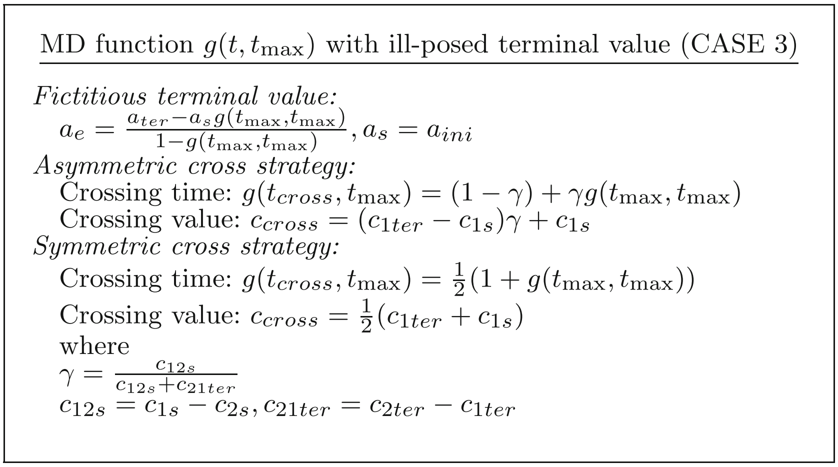
When the acceleration coefficients $c_1(t)$ and $c_2(t)$ are asymmetrical about a crossing axis, using a fictitious terminal boundary value in which the basic function is monotonically increasing with an ill-posed terminal boundary condition, i.e., $f(t_{\max}, t_{\max}) \neq 1$ and $f(t_{\max}, t_{\max}) \neq 0$, $c_1(t)$ and $c_2(t)$ are respectively expressed as follows:

$$c_{1e} = \frac{c_{1ter}}{f(t_{\max}, t_{\max})} + c_{1s}\left(1 - \frac{1}{f(t_{\max}, t_{\max})}\right) \tag{59}$$

$$\begin{aligned} c_1(t) &= (c_{1e} - c_{1s}) f(t, t_{\max}) + c_{1s} \\ &= \left(\frac{c_{1ter}}{f(t_{\max}, t_{\max})} + c_{1s} - \frac{c_{1s}}{f(t_{\max}, t_{\max})} - c_{1s}\right) \\ &\quad \times f(t, t_{\max}) + c_{1s} \\ &= (c_{1ter} - c_{1s}) f(t, t_{\max}) / f(t_{\max}, t_{\max}) + c_{1s} \end{aligned} \tag{60}$$

$$c_{2e} = \frac{c_{2ter}}{f(t_{\max}, t_{\max})} + c_{2s}\left(1 - \frac{1}{f(t_{\max}, t_{\max})}\right) \tag{61}$$

$$\begin{aligned} c_2(t) &= (c_{2e} - c_{2s}) f(t, t_{\max}) + c_{2s} \\ &= \left(\frac{c_{2ter}}{f(t_{\max}, t_{\max})} + c_{2s} - \frac{c_{2s}}{f(t_{\max}, t_{\max})} - c_{2s}\right) \\ &\quad \times f(t, t_{\max}) + c_{2s} \\ &= (c_{2ter} - c_{2s}) f(t, t_{\max}) / f(t_{\max}, t_{\max}) + c_{2s} \end{aligned} \tag{62}$$



MD function $g(t, t_{\max})$ with ill-posed terminal value (CASE 3)

Fictitious terminal value:

$a_e = \frac{a_{ter} - a_s g(t_{\max}, t_{\max})}{1 - g(t_{\max}, t_{\max})}, a_s = a_{ini}$

Asymmetric cross strategy:

Crossing time: $g(t_{cross}, t_{\max}) = (1-\gamma) + \gamma g(t_{\max}, t_{\max})$

Crossing value: $c_{cross} = (c_{1ter} - c_{1s})\gamma + c_{1s}$

Symmetric cross strategy:

Crossing time: $g(t_{cross}, t_{\max}) = \frac{1}{2}(1 + g(t_{\max}, t_{\max}))$

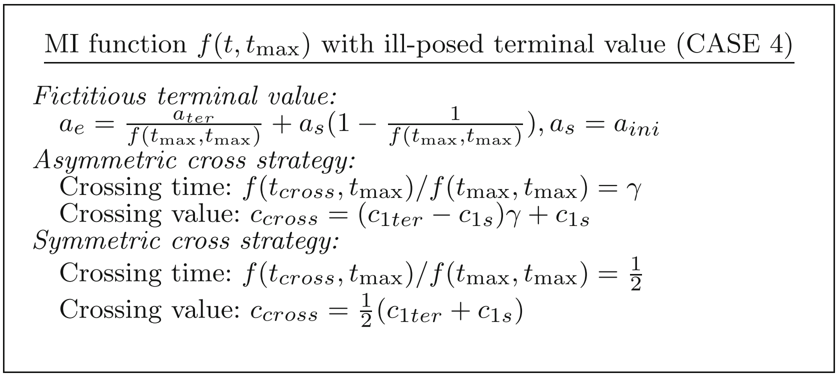
Crossing value: $c_{cross} = \frac{1}{2}(c_{1ter} + c_{1s})$

where

$\gamma = \frac{c_{12s}}{c_{12s} + c_{21ter}}$

$c_{12s} = c_{1s} - c_{2s}, c_{21ter} = c_{2ter} - c_{1ter}$

Fig. 3 The fictitious terminal value a_e, crossing time t_{cross} and crossing value c_{cross}, when the MD function $g(t, t_{\max})$ is terminally ill-posed



MI function $f(t, t_{\max})$ with ill-posed terminal value (CASE 4)

Fictitious terminal value:
$a_e = \frac{a_{ter}}{f(t_{\max}, t_{\max})} + a_s(1 - \frac{1}{f(t_{\max}, t_{\max})}), a_s = a_{ini}$
Asymmetric cross strategy:
Crossing time: $f(t_{cross}, t_{\max})/f(t_{\max}, t_{\max}) = \gamma$
Crossing value: $c_{cross} = (c_{1ter} - c_{1s})\gamma + c_{1s}$
Symmetric cross strategy:
Crossing time: $f(t_{cross}, t_{\max})/f(t_{\max}, t_{\max}) = \frac{1}{2}$
Crossing value: $c_{cross} = \frac{1}{2}(c_{1ter} + c_{1s})$

Fig. 4 The fictitious terminal value a_e, crossing time t_{cross} and crossing value c_{cross}, when the MI function $f(t, t_{\max})$ is terminally ill-posed

Equating (60) with (62), assuming that $f(t_{\max}, t_{\max}) \neq 0$, it follows that

$$(c_{1ter} - c_{1s}) f(t, t_{\max})/f(t_{\max}, t_{\max}) + c_{1s} \\ = (c_{2ter} - c_{2s}) f(t, t_{\max})/f(t_{\max}, t_{\max}) + c_{2s}$$

or

$$(c_{2ter} - c_{1ter}) f(t, t_{\max})/f(t_{\max}, t_{\max}) \\ +(c_{1s} - c_{2s}) f(t, t_{\max})/f(t_{\max}, t_{\max}) = c_{1s} - c_{2s}$$

Consequently, setting $c_{12s} \neq 0$ and $c_{21ter} \neq 0$, it is easy to find that

$$f(t, t_{\max})/f(t_{\max}, t_{\max}) = \gamma \tag{63}$$

If $f(t_{\max}, t_{\max}) \equiv 1$, i.e., the terminal boundary condition is well-posed, then (63) is simplified to

$$f(t, t_{\max}) = \gamma \tag{64}$$

4.4.2 A symmetric case about a crossing axis

When $c_1(t)$ and $c_2(t)$ are symmetric about the crossing axis, or the initial acceleration coefficient error and the terminal acceleration error are equivalent, i.e., $c_{21ter} = c_{12s}$, it is easy to find that (63) becomes

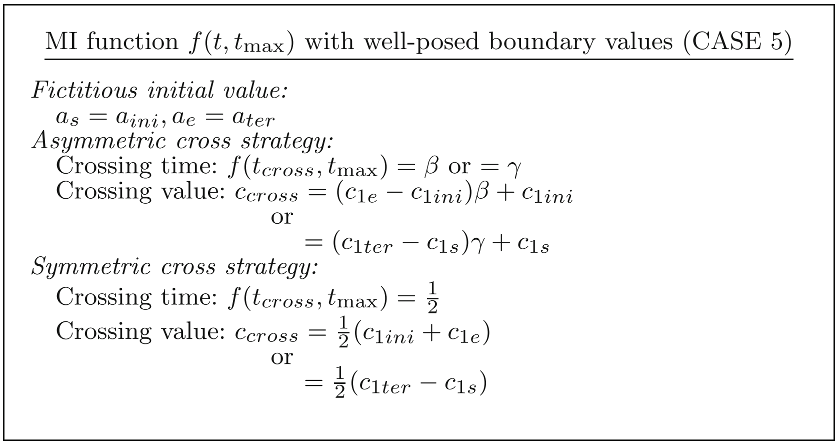
$$f(t, t_{\max})/f(t_{\max}, t_{\max}) = \frac{1}{2} \tag{65}$$

In addition, if the terminal boundary condition is well-posed, i.e., $f(t_{\max}, t_{\max}) \equiv 1$, then it is reduced to

$$f(t, t_{\max}) = \frac{1}{2} \tag{66}$$

4.4.3 Crossing value

Here, the crossing value in the crossing strategy is derived. Directly substituting the relationship between the basic function and the terminal boundary value in the asymmetric crossing method, i.e., (63) into (60), letting $c_1(t)$ in that time



MI function $f(t, t_{\max})$ with well-posed boundary values (CASE 5)

Fictitious initial value:
$a_s = a_{ini}, a_e = a_{ter}$
Asymmetric cross strategy:
Crossing time: $f(t_{cross}, t_{\max}) = \beta$ or $= \gamma$
Crossing value: $c_{cross} = (c_{1e} - c_{1ini})\beta + c_{1ini}$
or
$= (c_{1ter} - c_{1s})\gamma + c_{1s}$
Symmetric cross strategy:
Crossing time: $f(t_{cross}, t_{\max}) = \frac{1}{2}$
Crossing value: $c_{cross} = \frac{1}{2}(c_{1ini} + c_{1e})$
or
$= \frac{1}{2}(c_{1ter} - c_{1s})$

Fig. 5 The crossing time t_{cross} and crossing value c_{cross}, when the MI function $f(t, t_{\max})$ is well-posed at both boundaries

MD function $g(t, t_{\max})$ with well-posed boundary values (CASE 6)

Bondary values:

$a_s = a_{ini}, a_e = a_{ter}$

Asymmetric cross strategy:

Crossing time: $g(t_{cross}, t_{\max}) = 1 - \beta$ or $= 1 - \gamma$

Crossing value: $c_{cross} = (c_{1e} - c_{1ini})\beta + c_{1ini}$

or

$= (c_{1ter} - c_{1s})\gamma + c_{1s}$

Symmetric cross strategy:

Crossing time: $g(t_{cross}, t_{\max}) = \frac{1}{2}$

Crossing value: $c_{cross} = \frac{1}{2}(c_{1ini} + c_{1e})$

or

$= \frac{1}{2}(c_{1ter} - c_{1s})$

Fig. 6 The crossing time t_{cross} and crossing value c_{cross}, when the MD function $g(t, t_{\max})$ is well-posed at both boundaries

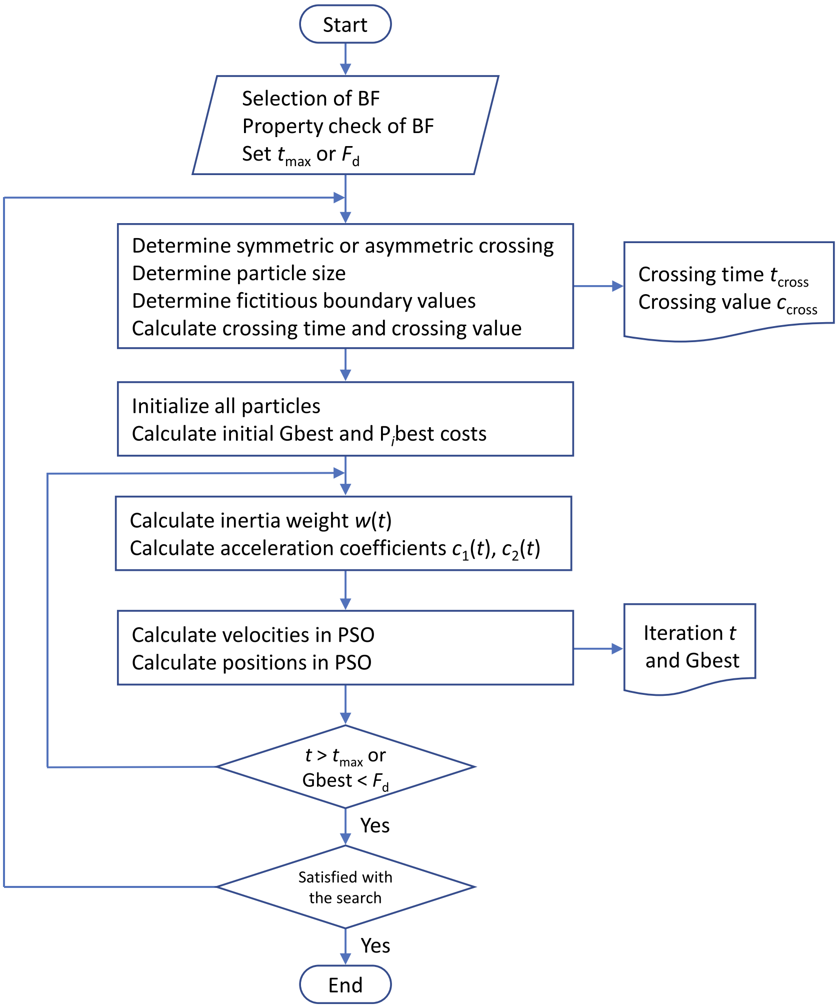

Start
Selection of BF
Property check of BF
Set $t_{\max}$ or F_d
Determine symmetric or asymmetric crossing
Determine particle size
Determine fictitious boundary values
Calculate crossing time and crossing value
Crossing time t_{cross}
Crossing value c_{cross}
Initialize all particles
Calculate initial Gbest and $P_i$best costs
Calculate inertia weight $w(t)$
Calculate acceleration coefficients $c_1(t)$, $c_2(t)$
Calculate velocities in PSO
Calculate positions in PSO
Iteration t and Gbest
$t > t_{\max}$ or Gbest < F_d
Yes
Satisfied with the search
Yes
End

Fig. 7 Schematic flowchart of the proposed cross-strategy method, where BF denotes the basic function, F_d is a desired global best cost if it is known in advance, *Gbest* is a global best cost, and P_i*best* denotes a personal best at the ith particle

be c_{cross}, it follows that

$$\begin{aligned} c_{cross} &= (c_{1ter} - c_{1s})\gamma f(t_{\max}, t_{\max})/f(t_{\max}, t_{\max}) + c_{1s} \\ &= (c_{1ter} - c_{1s})\gamma + c_{1s} \end{aligned} \tag{67}$$

Since $c_{1ter} = c_{2s}$ and $\gamma = 1/2$ in the symmetric crossing strategy, it is easy to find that

$$c_{cross} = \frac{1}{2}(c_{1ter} + c_{1s}) \equiv \frac{1}{2}(c_{1s} + c_{2s}) \tag{68}$$

or from $c_{1s} = c_{2ter}$ that

$$c_{cross} = \frac{1}{2}(c_{1ter} + c_{2ter}) \equiv \frac{1}{2}(c_{2ter} + c_{2s}) \tag{69}$$

Figure 4 shows the fictitious terminal value a_e, crossing time t_{cross}, and crossing value c_{cross} when the monotonically increasing function $f(t, t_{\max})$ is terminally ill-posed. However, t_{cross} must be obtained after the concrete $f(t, t_{\max})$ is given. When $f(t, t_{\max})$ is terminally well-posed, it should take that $f(t_{\max}, t_{\max}) \equiv 1$.

When the monotonically increasing function $f(t, t_{\max})$ is well-posed at both ends, it is a special case of Section 4.1 or Section 4.4, and Fig. 5 summarizes the boundary values and crossing information values, i.e., crossing time t_{cross} and crossing value c_{cross}. When the monotonically decreasing function $g(t, t_{\max})$ is well-posed at both ends, it is a special case of Section 4.2 or Section 4.3, and similar results are summarized in Fig. 6.

Figure 7 shows a schematic flowchart of the cross-strategy method proposed in this paper. Figures 8 and 9 show the procedure for determining fictitious boundary values, calculating crossing times and crossing values. Figure 10 shows the monotonically decreasing $w(t)$, $c_1(t)$ and monotonically increasing $c_2(t)$ formulas for which the monotonically increasing $f(t, t_{\max})$ and monotonically decreasing $g(t, t_{\max})$ basic functions are applied, respectively. Note that although theoretically there are six possible cases as shown in Figs. 1 to 6 for determining the fictitious boundary value and calculating the crossing time and crossing value according to the characteristics of the basic function, the actual program is executed only after the basic function is selected in advance and the symmetry or asymmetry strategy is determined.

5 Crossing time

Here, the crossing times are obtained for asymmetric general crossing strategies when using 10 monotonically increasing or monotonically decreasing basic functions.

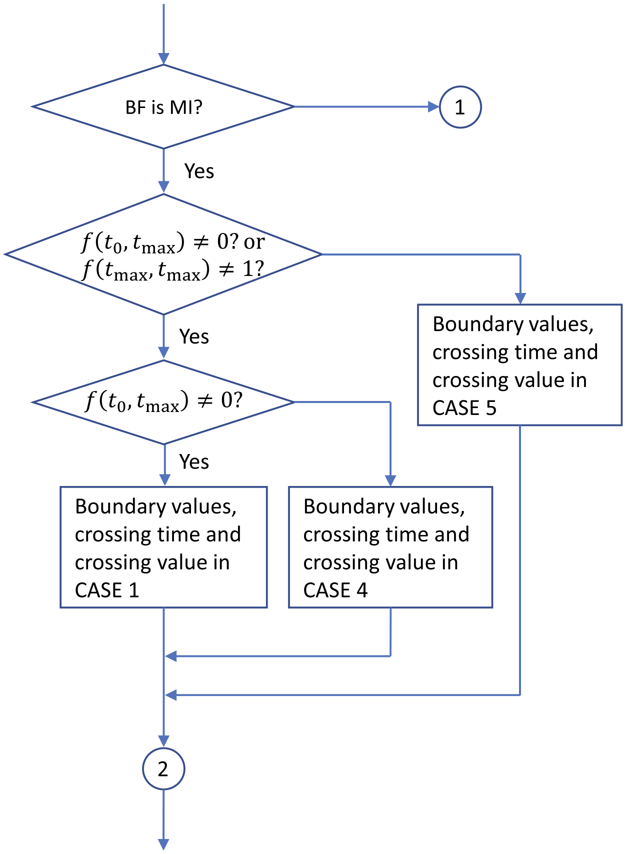

BF is MI?
1
Yes
$f(t_0, t_{\max}) \neq 0$? or $f(t_{\max}, t_{\max}) \neq 1$?
Boundary values, crossing time and crossing value in CASE 5
Yes
$f(t_0, t_{\max}) \neq 0$?
Yes
Boundary values, crossing time and crossing value in CASE 1
Boundary values, crossing time and crossing value in CASE 4
2

Fig. 8 Procedures for determining fictitious boundary values, calculating crossing times and corssing values when the basic function (BF) is monotonically increasing (MI)

5.1 $N(t, t_{\max}) = \frac{1-(t/t_{\max})}{1+s(t/t_{\max})}, -1 < s < \infty$

Since this function was used to represent Sugeno's fuzzy complement [79] and is a well-posed monotonically decreasing function satisfying $N(0, t_{\max}) = 1$, $N(t_{\max}, t_{\max}) = 0$ [45] , using the relationship between the basic function and the well-posed boundary values, t_{cross} satisfies

$$\frac{1 - (t_{cross}/t_{\max})}{1 + s(t_{cross}/t_{\max})} = 1 - \beta$$

Rearranging the above equation to obtain

$$1 - \frac{t_{cross}}{t_{\max}} = (1 - \beta)\left(1 + \frac{st_{cross}}{t_{\max}}\right)$$

or

$$\beta = \frac{t_{cross}}{t_{\max}}[s(1 - \beta) + 1]$$

it follows that

$$t_{cross} = \frac{\beta t_{\max}}{1 + s(1 - \beta)} \tag{70}$$

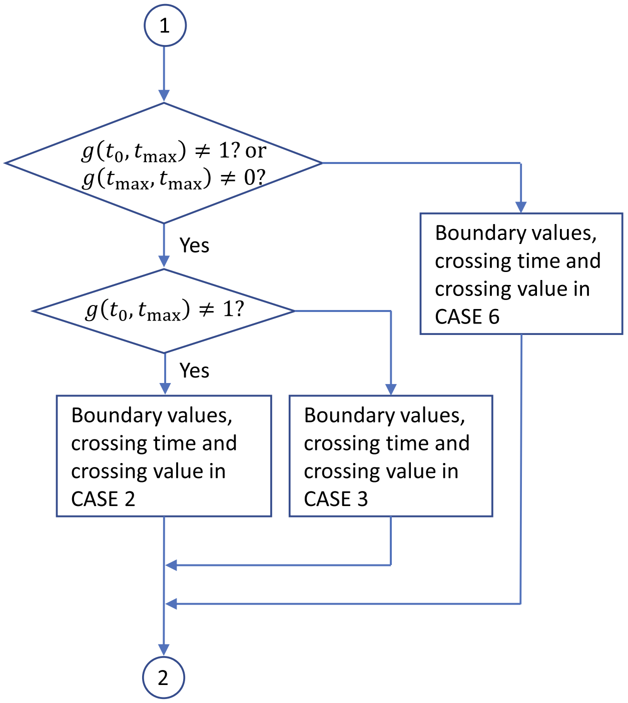

1
$g(t_0, t_{\max}) \neq 1$? or
$g(t_{\max}, t_{\max}) \neq 0$?
Yes
$g(t_0, t_{\max}) \neq 1$?
Yes
Boundary values, crossing time and crossing value in CASE 2
Boundary values, crossing time and crossing value in CASE 3
Boundary values, crossing time and crossing value in CASE 6
2

Fig. 9 Procedures for determining fictitious boundary values, calculating crossing times and corssing values when the basic function is monotonically decreasing

For a symmetric crossing, i.e., $\beta = 1/2$, (70) becomes

$$t_{cross} = \frac{0.5t_{\max}}{1 + 0.5s} \tag{71}$$

5.2 $N_g(t, t_{\max}) = [1 - (t/t_{\max})^{\alpha}]^{1/\alpha}, -1 < \alpha < \infty$

This basic function was used to represent Yager's fuzzy complement [44], which is also a monotonically decreasing function satisfying $N_g(0, t_{\max}) = 1$, $N_g(t_{\max}, t_{\max}) = 0$. Therefore, since the crossing time t_{cross} satisfies

$$[1 - (t_{cross}/t_{\max})^{\alpha}]^{1/\alpha} = 1 - \beta$$

or

$$1 - \left(\frac{t_{cross}}{t_{\max}}\right)^{\alpha} = (1 - \beta)^{\alpha}$$

it follows that

$$(t_{cross}/t_{\max})^{\alpha} = 1 - (1 - \beta)^{\alpha}$$

consequently

$$t_{cross} = t_{\max}[1 - (1 - \beta)^{\alpha}]^{1/\alpha} \tag{72}$$

Fo a symmetric crossing, i.e., $\beta = 1/2$, (72) is reduced to

$$t_{cross} = t_{\max}\left(1 - \left(\frac{1}{2}\right)^{\alpha}\right)^{1/\alpha} \tag{73}$$

For a monotonic-increasing case, i.e., $f'(t, t_{\max}) \geq 0$

Inertia weight:

$w(t) = (w_e - w_s)f(t, t_{\max}) + w_s, w_s > w_e$

Acceleration coeffiecinets:

$c_1(t) = (c_{1e} - c_{1s})f(t, t_{\max}) + c_{1s}, c_{1s} > c_{1e}$

$c_2(t) = (c_{2e} - c_{2s})f(t, t_{\max}) + c_{2s}, c_{2s} < c_{2e}$

(a) When using a monotonic-increasing function as a BF

For a monotonic-decreasing case, i.e., $g'(t, t_{\max}) \leq 0$

Inertia weight:

$w(t) = (w_s - w_e)g(t, t_{\max}) + w_e, w_s > w_e$

Acceleration coeffiecinets:

$c_1(t) = (c_{1s} - c_{1e})g(t, t_{\max}) + c_{1e}, c_{1s} > c_{1e}$

$c_2(t) = (c_{2s} - c_{2e})g(t, t_{\max}) + c_{2e}, c_{2s} < c_{2e}$

(b) When using a monotonic-decreasing function as a BF

Fig. 10 Inertia weight and acceleration coefficients with a monotonic-increasing or monotonic-decreasing function as a BF

5.3 $G(t, t_{\max}) = \tan^{-1}(\frac{t_{\max}-t}{t_{\max}})$

Since this function was a monotonically decreasing function whose initial boundary value was ill-posed in the result of fictitious boundary values, the crossing time satisfies

$$\tan^{-1}\left(\frac{t_{\max}-t_{cross}}{t_{\max}}\right) = (1-\beta)G(t_0, t_{\max}) = \frac{(1-\beta)\pi}{4}$$

so that

$$\tan\left[\frac{(1-\beta)\pi}{4}\right] = 1 - t_{cross}/t_{\max}$$

Consequently, it yields that

$$t_{cross} = t_{\max}\left\{1 - \tan\left[\frac{(1-\beta)\pi}{4}\right]\right\} \tag{74}$$

For a symmetric crossing of $\beta = 1/2$, (74) results in

$$t_{cross} = t_{\max}\left[1 - \tan\left(\frac{\pi}{8}\right)\right] \tag{75}$$

Note that, changing it into a well-posed monotonically decreasing function such as $\frac{4}{\pi}\tan^{-1}(\frac{t_{\max}-t}{t_{\max}})$, the similar result is obtained from $\frac{4}{\pi}\tan^{-1}(\frac{t_{\max}-t}{t_{\max}}) = 1 - \beta$.

5.4 $G(t, t_{\max}) = \cos(\pi\frac{t}{t_{\max}})$

This function was a monotonically decreasing function whose terminal value was ill-posed in the previous result of fictitious boundary values, so that the crossing time satisfies

$$\cos\left(\pi\frac{t_{cross}}{t_{\max}}\right) = (1-\gamma) + \gamma G(t_{\max}, t_{\max}) = (1-\gamma) - \gamma = 1 - 2\gamma$$

Deforming it such that

$$\pi\frac{t_{cross}}{t_{\max}} = \cos^{-1}(1-2\gamma)$$

consequently it follows that

$$t_{cross} = \frac{t_{\max}}{\pi}\cos^{-1}(1-2\gamma) \tag{76}$$

Here, in a symmetric crossing of $\gamma = 1/2$, (76) becomes

$$t_{cross} = \frac{t_{\max}}{\pi}\cos^{-1}(0) = \frac{t_{\max}}{2} \tag{77}$$

Note that if it is slightly modified into a well-posed monotonically decreasing function such as $\frac{1}{2}[\cos(\pi\frac{t}{t_{\max}}) + 1]$, then the similar result can be obtained directly from $\frac{1}{2}[\cos(\pi\frac{t}{t_{\max}}) + 1] = 1 - \gamma$.

5.5 $g(t, t_{\max}) = (1 - t/t_{\max})^n$

This function in Chatterjee and Siarry [11] is a well-posed monotonically decreasing function (i.e., $\{(1 - t/t_{\max})^n\}' = -\frac{n}{t_{\max}}(1-t/t_{\max})^{n-1} < 0$) taking 1 at $t = 0$ and 0 at $t = t_{\max}$, where n denotes a modulation index. Therefore, the crossing time satisfies

$$(1 - t_{cross}/t_{\max})^n = 1 - \beta$$

or

$$(1 - t_{cross}/t_{\max}) = (1-\beta)^{1/n}$$

so that

$$t_{cross} = t_{\max}[1 - (1-\beta)^{1/n}] \tag{78}$$

Note here that in a symmetric crossing of $\beta = 1/2$, (78) results in

$$t_{cross} = t_{\max}\left[1 - \left(\frac{1}{2}\right)^{1/n}\right] \tag{79}$$

5.6 $f(t, t_{\max}) = (t/t_{\max})^n$

This function in Wang and Liu [86] is a well-posed monotonically increasing function (i.e., $\{(t/t_{\max})^n\}' = \frac{n}{t_{\max}}(t/t_{\max})^{n-1} > 0$) taking 0 at $t = 0$ and 1 at $t = t_{\max}$. Then, since the crossing time satisfies

$$(t_{cross}/t_{\max})^n = \beta$$

or

$$t_{cross}/t_{\max} = (\beta)^{1/n}$$

it follows that

$$t_{cross} = t_{\max}(\beta)^{1/n} \tag{80}$$

Note here that in a symmetric crossing of $\beta = 1/2$, (80) becomes

$$t_{cross} = t_{\max}\left(\frac{1}{2}\right)^{1/n} \tag{81}$$

5.7 $F(t, t_{\max}) = e^{-(t_{\max}-t)/t_{\max}}$

When defining $f(t, t_{\max}) = t/t_{\max}$ and letting the Napier's constant be the base, this function uses $1 - f(t, t_{\max})$ to make a function $e^{-[1-f(t,t_{\max})]} = \frac{1}{e^{1-f(t,t_{\max})}} \overset{\triangle}{=} F(t, t_{\max})$. This function is monotonically increasing because $1 - f(t, t_{\max})$ is monotonically decreasing (i.e., $\{1 - f(t, t_{\max})\}' = -1/t_{\max} < 0$). However, $F(0, t_{\max}) = 1/e \neq 0$, $F(t_{\max}, t_{\max}) = 1$, which is an ill-posed boundary value. Therefore, from the relationship between the basic function and the ill-posed boundary value when the initial boundary condition of the monotonically increasing function is ill-posed, the crossing time satisfies

$$\begin{aligned} e^{-(t_{\max}-t_{cross})/t_{\max}} &= \beta + (1-\beta)F(0, t_{\max}) \\ &= \beta + (1-\beta)/e \\ &= \frac{1}{e}[\beta e + (1-\beta)] \end{aligned}$$

and moreover acting on both sides of this equation with log gives

$$\begin{aligned} -(t_{\max} - t_{cross})/t_{\max} &= \log\left\{\frac{1}{e}[\beta e + (1-\beta)]\right\} \\ &= \log[\beta e + (1-\beta)] - 1 \end{aligned}$$

Thus, from the fact that

$$t_{\max} - t_{cross} = t_{\max}\{1 - \log[\beta e + (1-\beta)]\}$$

it yields that

$$t_{cross} = t_{\max} - t_{\max}\{1 - \log[\beta e + (1-\beta)]\} \tag{82}$$

Note here that for a symmetric crossing of $\beta = 1/2$, (82) is reduced to

$$t_{cross} = t_{\max} - t_{\max}\left\{1 - \log\left(\frac{e+1}{2}\right)\right\} \tag{83}$$

5.8 $G(t, t_{\max}) = e^{-t_{\max}/(t_{\max}-t)}$

When letting $f(t, t_{\max}) = t/t_{\max}$, this is found to be a function $e^{-1/[1-f(t,t_{\max})]} = \frac{1}{e^{t_{\max}/(t_{\max}-t)}} \overset{\triangle}{=} G(t, t_{\max})$ in which the Napier's constant is used as the base, using $(1 - f(t, t_{\max}))^{-1}$ that is the reciprocal of $1 - f(t, t_{\max})$ (see also Wu et al. [89]). This function is monotonically decreasing, because $1/(1 - f(t, t_{\max}))$ is monotonically increasing, where $G(t_{\max}, t_{\max}) = 0$ is satisfied, but $G(0, t_{\max}) = 1/e \neq 1$ so that it yields an ill-posed boundary value. Therefore, using the relationship between the basic function and the ill-posed boundary when the initial boundary condition is ill-posed in a monotonically decreasing function, the crossing time satisfies

$$\begin{aligned} e^{-t_{\max}/(t_{\max}-t_{cross})} &= (1-\beta)G(0, t_{\max}) \\ &= \frac{1-\beta}{e} \end{aligned}$$

and acting on both sides of this equation with log, it obtains

$$-t_{\max}/(t_{\max} - t_{cross}) = \log(1-\beta) - 1$$

or

$$\frac{1}{t_{\max} - t_{cross}} = \frac{1 - \log(1-\beta)}{t_{\max}}$$

As a result, it follows that

$$t_{cross} = t_{\max} - \frac{t_{\max}}{1 - \log(1-\beta)} \tag{84}$$

where note that in a symmetric crossing of $\beta = 1/2$, (84) results in

$$t_{cross} = t_{\max} - \frac{t_{\max}}{1 - \log(\frac{1}{2})} \tag{85}$$

5.9 $G_L(t, t_{\max}) = e^{-\alpha t/t_{\max}}$

This function $G_L(t, t_{\max})$ was used for inertial weights in Lu et al. [50] and is a monotonically decreasing function. However, this $G_L(t, t_{\max})$ has an ill-posed boundary value because $G_L(0, t_{\max}) = 1$ and $G_L(t_{\max}, t_{\max}) = e^{-\alpha} \neq 0$. Therefore, when the terminal boundary condition of the monotonically decreasing function is ill-posed, using the relationship between the basic function and the ill-posed boundary value, it is found that the crossing time satisfies

$$\begin{aligned} e^{-\alpha t_{cross}/t_{\max}} &= (1-\gamma) + \gamma G_L(t_{\max}, t_{\max}) \\ &= (1-\gamma) + \gamma/e^{\alpha} \end{aligned}$$

and acting on both sides of this equation with log gives

$$-\alpha t_{cross}/t_{\max} = \log[(1-\gamma) + \gamma/e^{\alpha}]$$

consequently, it yields that

$$t_{cross} = -\frac{t_{\max}}{\alpha}\{\log[(1-\gamma) + \gamma/e^{\alpha}]\} \tag{86}$$

Note here that if it takes a symmetric crossing of $\gamma = 1/2$, then (86) becomes

$$t_{cross} = -\frac{t_{\max}}{\alpha}\left(\log\frac{1 + 1/e^{\alpha}}{2}\right) \tag{87}$$

5.10 $F_L(t, t_{\max}) = e^{-\alpha t_{\max}/t}$

This function $F_L(t, t_{\max})$ has the reciprocal of the exponent $t/t_{\max}$ of the previous $G_L(t, t_{\max})$ and is a monotonically increasing function. However, in this $F_L(t, t_{\max})$, $F_L(0, t_{\max}) = 0$, but $F_L(t_{\max}, t_{\max}) = e^{-\alpha} \neq 1$, so that it results in an ill-posed boundary value. Therefore, from the relationship between the fundamental function and the ill-posed boundary value when the terminal boundary condition of the monotonically increasing function is ill-posed, it is found that the crossing time satisfies

$$\begin{aligned} e^{-\alpha t_{\max}/t_{cross}} &= \gamma F_L(t_{\max}, t_{\max}) \\ &= \gamma e^{-\alpha} \end{aligned}$$

and acting on both sides of this equation with log yields

$$-\alpha t_{\max}/t_{cross} = \log(\gamma) - \alpha$$

or

$$t_{\max}/t_{cross} = 1 - \frac{\log(\gamma)}{\alpha}$$

Consequently, it follows that

$$t_{cross} = \frac{t_{\max}}{1 - \frac{\log(\gamma)}{\alpha}} \tag{88}$$

where for a symmetric crossing of $\gamma = 1/2$, it is found that (88) is reduced to

$$t_{cross} = \frac{t_{\max}}{1 - \frac{\log(\frac{1}{2})}{\alpha}} \tag{89}$$

6 Numerical experiments on CEC2014 bechmark functions

This section is the simulation experiment of the proposed algorithms. To validate the performance of local exploitation, local extremum avoidance and global exploration, the statistical analysis was performed on CEC2014 benchmark functions [47]. For comparing the relative performance of stochastic optimization algorithms, there are numerous test beds or benchmark functions [8]. Notably, recent benchmark functions include CEC2014 [47], CEC2017 [4], and CEC2020 [98]. However, focusing on CEC2014, the following differences can be observed:

- CEC2017 has a total of 29 functions, which is almost the same as the 30 functions in CEC2014. However, compared to CEC2014, CEC2017 significantly reduces the number of multimodal functions and instead increases the number of hybrid and composition functions. These benchmark problems emphasize exploring these functions from different perspectives.
- On the other hand, CEC2020 narrows its focus to a selective or specific set of benchmark problems. It includes a total of 10 functions derived from CEC2014 and CEC2017: one unimodal, two multimodal, four hybrid, and three composition functions.

In this study, we considered the simulation experiments conducted at the time, the available information (specifically, the MATLAB environment), and the ease of comparison based on the coherence of experiments and resource availability. As a result, we opted for a well-balanced set of benchmark functions from CEC2014. Three classic PSO variants, i.e., LDW-PSO [74], LCS-PSO [65], NDW-PSO [11], and four state-of-the-art metaheuristics, i.e., CSA [3], PPSO [27], POA [84], and SHO [103] were compared with our proposed NCS-PSOs.

6.1 Experimental parameter settings and conditions on benchmark functions

CEC2014 suite is the general test standard of modern algorithms, which has strong test suitability for all kinds of metaheuristic algorithms. Because of the dynamic and complexity of these benchmark functions, they are more convincing to validate the optimization performance of the proposed NCS-PSOs. This test suite includes 30 different types of functions: unimodal (CEC-1 to CEC-3), simple multimodal (CEC-4 to CEC-16), hybrid (CEC-17 to CEC-22) and composition functions (CEC-23 to CEC-30). Note that each optimal value is set as the function's alignment number times 100. The mean costs or fitness values of the experiments are recorded. The algorithms are ranked based on the results of the mean values. Also, we conduct the Wilcoxon rank-sum test at 0.05 significance level between NCS-PSOs and other algorithms. The symbol '$+$' and '$-$' mean that any NCS-PSO is significantly better, significantly worse than the compared algorithm.

The compared algorithms are listed in Table 1, where the parameter settings are listed in Table 2. Note here that such design parameters were set, referring to original papers. All algorithms run 30 times on the CEC 2014 test suite. The experiments are conducted on the 30-dimensional problems (i.e., $D = 30$), though 10, 30, 50 and 100 were prepared in the technical report [47]. The maximum number of fitness evaluations, $t_{\max}$, is set to 500. The population size M is set to be 100 for all the algorithms. The search ranges $[-100, 100]^D$ are defined for all benchmark functions. All current algorithms were coded in MATLAB 2018a and executed on a computer with Intel(R) Core(TM) i9-9900K CPU @ 3.60

Table 1 List of the compared algorithms

No.	Abbreviation	Full name	Literature
1	LDW-PSO	Linearly decreasing inertia weight PSO	Shi and Eberhart (1999) [74]
2	LCS-PSO	Linearly crossing strategy PSO	Ratnaweera et al. (2004) [65]
3	NDW-PSO	Nonlinearly decreasing inertia weight PSO	Chatterjee and Siarry (2006) [11]
4	CSA	Crow search algorithm	Askarzadeh (2016) [3]
5	PPSO	Phasor PSO	Ghasemi et al. (2019) [27]
6	POA	Pelican optimization algorithm	Trojovsky and Dehghami (2022) [84]
7	SHO	Sea horse optimizer	Zhao et al. (2023) [103]

Table 2 Parameter settings for the classic PSO variants and the advanced methaheuristics

No.	Algorithms	Parameter settings
1	LDW-PSO	Time-varying inertia weight: $w(t) = (1 - t/t_{\max})[w_s - w_e] + w_e$, $w_s = 0.9$, $w_e = 0.4$; constant acceleration coefficients, $c_1 = c_2 = 2.0$
2	LCS-PSO	Linear time-varying inertia weight: $w(t) = (1 - t/t_{\max})[w_s - w_e] + w_e$, $w_s = 0.9$, $w_e = 0.4$; Linear time-varying acceleration coefficients: $c_1(t) = \frac{t}{t_{\max}}[c_{1e} - c_{1s}] + c_{1s}$, $c_{1s} > c_{1e}$, $c_{1s} = 2.5$, $c_{1e} = 0.5$; $c_2(t) = \frac{t}{t_{\max}}[c_{2e} - c_{2s}] + c_{2s}$, $c_{2s} < c_{2e}$, $c_{2s} = 0.5$, $c_{2e} = 2.5$
3	NDW-PSO	Nonlinear time-varying inertia weight: $w(t) = (1 - t/t_{\max})^n[w_s - w_e] + w_e$, modulation index $n = 1.2$, $w_s = 0.9$, $w_e = -0.5$; Constant acceleration coefficients, $c_1 = c_2 = 2.0$
4	CSA	Flight length $fl = 2.0$; awareness probability $AP = 0.1$
5	PPSO	–
6	POA	A constant $R = 0.2$
7	SHO	Levy index $\lambda = 1.5$; Brownian scaler $l = 0.05$; spiral or Brownian motion probability in N[0, 1], $P_{r1} = 0.0$; failure probability in capturing food in U[0, 1], $P_{r2} = 0.1$

Table 3 Parameter settings for the proposed NCS-PSOs

No.	Basic function	w_s, w_e	c_{is}, c_{ie}	n or α	Crossing-time
BF-1	MDF: $(1 - t/t_{\max})^n$	$w_s = 0.99$, $w_e = 0.1$	$c_{1s} = c_{2e} = 3.0$ $c_{1e} = c_{2s} = 0.0$	$n = 0.8$	$t_{cross} = 290$[1]
BF-2	MIF: $(t/t_{\max})^n$	$w_s = 0.99$, $w_e = 0.1$	$c_{1s} = c_{2e} = 3.0$ $c_{1e} = c_{2s} = 0.0$	$n = 0.8$	$t_{cross} = 211$
BF-3	MIF: $e^{-(t_{\max}-t)/t_{\max}}$	$w_s = 0.99$, $w_e = 0.1$	$c_{1ini} = c_{2e} = 3.5$ $c_{1e} = c_{2ini} = 0.0$	–	$t_{cross} = 311$
BF-4	MDF: $e^{-t_{\max}/(t_{\max}-t)}$	$w_s = 0.99$, $w_e = 0.1$	$c_{1ini} = c_{2e} = 3.5$ $c_{1e} = c_{2ini} = 0.0$	–	$t_{cross} = 205$
BF-5	MDF: $e^{-\alpha t/t_{\max}}$	$w_s = 0.99$, $w_e = 0.1$	$c_{1s} = c_{2ter} = 3.0$ $c_{1ter} = c_{2s} = 0.0$	$\alpha = 0.1$	$t_{cross} = 244$
BF-6	MIF: $e^{-\alpha t_{\max}/t}$	$w_s = 0.99$, $w_e = 0.1$	$c_{1s} = c_{2ter} = 3.0$ $c_{1ter} = c_{2s} = 0.0$	$\alpha = 0.7$	$t_{cross} = 252$

[1] These values of crossing time were obtained from (79), (81), (83), (85), (87), (89) in the symmetric crossing strategy at $t_{\max} = 500$, where each value represents the ceiling value of real number with decimal point

Table 4 Results of the comparative methods on CEC2014 bechmark functions (CEC-1 to CEC-15)

Function		LDW-PSO	LCS-PSO	NDW-PSO	CSA	PPSO	POA	SHO	BF-1	BF-2	BF-3	BF-4	BF-5	BF-6
CEC-1	Mean	2.1547E+07	1.2785E+07	2.1464E+07	3.7641E+07	**2.4976E+06**	1.0401E+08	2.0139E+08	1.1202E+07	1.3315E+07	9.1862E+06	6.1883E+06	9.0057E+06	8.6715E+06
	Std	1.1357E+07	1.1065E+07	1.2629E+07	2.1840E+07	**1.3760E+06**	7.0622E+07	9.2085E+07	1.0124E+07	1.4184E+07	7.8380E+06	6.4255E+06	8.0813E+06	8.6019E+06
CEC-2	Mean	2.1723E+05	5.1393E+03	1.5388E+04	1.0430E+08	1.9886E+04	1.9376E+10	1.9778E+10	8.8557E+03	1.0554E+04	**5.0094E+03**	9.8171E+03	9.6308E+03	1.4332E+04
	Std	1.3063E+05	**5.6833E+03**	1.0614E+04	5.0096E+07	1.0051E+04	7.9366E+09	8.1117E+09	9.1610E+03	9.4603E+03	6.8292E+03	1.0210E+04	7.8818E+03	2.3099E+04
CEC-3	Mean	9.9919E+03	2.2532E+03	9.7605E+03	3.2506E+04	3.0915E+03	3.4463E+04	3.9465E+04	2.9761E+03	**1.7118E+03**	4.8455E+03	4.9951E+03	2.9205E+03	3.8177E+03
	Std	1.0292E+04	2.2015E+03	9.9046E+03	7.0373E+03	1.6841E+03	8.0947E+03	8.9796E+03	2.5816E+03	**1.2916E+03**	3.6992E+03	3.9384E+03	1.9014E+03	2.5773E+03
CEC-4	Mean	6.0330E+02	5.8327E+02	5.8407E+02	6.2598E+02	**4.8660E+02**	1.3744E+03	1.4747E+03	5.8877E+02	5.9662E+02	5.9219E+02	5.7955E+02	5.8539E+02	6.0058E+02
	Std	4.3718E+01	5.9865E+01	3.4574E+01	5.2114E+01	3.1190E+01	5.5330E+02	5.1186E+02	**2.9372E+01**	5.0173E+01	3.8081E+01	3.9532E+01	4.5061E+01	4.0500E+01
CEC-5	Mean	5.2092E+02	5.2091E+02	5.2096E+02	5.2102E+02	**5.2006E+02**	5.2075E+02	5.2057E+02	5.2082E+02	5.2033E+02	5.2096E+02	5.2015E+02	5.2066E+02	5.2057E+02
	Std	9.8988E-02	8.5870E-02	6.9913E-02	**5.3978E-02**	1.4969E-01	7.6185E-02	1.6410E-01	1.4880E-01	1.8050E-01	6.5548E-02	6.1056E-02	1.9155E-01	1.4113E-01
CEC-6	Mean	6.1383E+02	**6.1219E+02**	6.1342E+02	6.2340E+02	6.3148E+02	6.2674E+02	6.2772E+02	6.1453E+02	6.1548E+02	6.1369E+02	6.1324E+02	6.1616E+02	6.1908E+02
	Std	3.4605E+00	2.6789E+00	2.9037E+00	3.6143E+00	4.3404E+00	2.9260E+00	**2.4053E+00**	3.8920E+00	3.4354E+00	3.2006E+00	3.3427E+00	3.7717E+00	4.3309E+00
CEC-7	Mean	7.0049E+02	7.0001E+02	7.0006E+02	7.0228E+02	7.0006E+02	8.5553E+02	8.7481E+02	7.0002E+02	7.0002E+02	7.0001E+02	**7.0001E+02**	7.0002E+02	7.0005E+02
	Std	2.5198E-01	1.7465E-02	5.1715E-02	6.5579E-01	2.8466E-02	6.2532E+01	6.7873E+01	1.8714E-02	1.7035E-02	**1.2153E-02**	1.7657E-02	1.6232E-02	6.5721E-02
CEC-8	Mean	8.3844E+02	8.3552E+02	8.3640E+02	9.0703E+02	9.2263E+02	9.6889E+02	9.2785E+02	8.3910E+02	8.3848E+02	8.3323E+02	**8.3022E+02**	8.3599E+02	8.3626E+02
	Std	1.1608E+01	8.9342E+00	8.3963E+00	1.7749E+01	2.0990E+01	1.5719E+01	2.4664E+01	9.3263E+00	1.0861E+01	**7.8407E+00**	9.3563E+00	8.2989E+00	8.0175E+00
CEC-9	Mean	9.8893E+02	9.6159E+02	9.7214E+02	1.0240E+03	1.0731E+03	1.0976E+03	1.0922E+03	9.5678E+02	9.5595E+02	9.5794E+02	9.5462E+02	9.5456E+02	**9.5224E+02**
	Std	3.3472E+01	1.6990E+01	2.7115E+01	2.4880E+01	3.2138E+01	2.1220E+01	2.4387E+01	1.5342E+01	1.3928E+01	1.7884E+01	**1.3325E+01**	1.4283E+01	1.3967E+01
CEC-10	Mean	2.3275E+03	2.5266E+03	**2.1989E+03**	4.0819E+03	3.8463E+03	4.4727E+03	3.6243E+03	3.0516E+03	2.9903E+03	2.5090E+03	2.2052E+03	3.1109E+03	3.4447E+03
	Std	5.6976E+02	**4.2397E+02**	5.0252E+02	7.6506E+02	4.6298E+02	4.8971E+02	5.2386E+02	5.9958E+02	6.4654E+02	4.6078E+02	4.8560E+02	6.2120E+02	5.5427E+02
CEC-11	Mean	5.9591E+03	4.5444E+03	5.0647E+03	4.6958E+03	5.3690E+03	4.7501E+03	4.9533E+03	4.7303E+03	4.3568E+03	4.5633E+03	**4.2589E+03**	4.7213E+03	5.0279E+03
	Std	1.3264E+03	8.0023E+02	1.3420E+03	6.8261E+02	7.6366E+02	5.3069E+02	**4.9470E+02**	8.4360E+02	6.0249E+02	8.3467E+02	6.5080E+02	7.7629E+02	7.4090E+02
CEC-12	Mean	1.2023E+03	1.2010E+03	1.2022E+03	1.2008E+03	1.2008E+03	1.2006E+03	1.2008E+03	1.2012E+03	1.2008E+03	1.2018E+03	**1.2005E+03**	1.2009E+03	1.2009E+03
	Std	4.7444E-01	5.7875E-01	5.6213E-01	4.9084E-01	3.0166E-01	2.3772E-01	2.6955E-01	4.1529E-01	3.0616E-01	5.2984E-01	**1.9081E-01**	3.9726E-01	3.3092E-01
CEC-13	Mean	1.3004E+03	1.3004E+03	1.3005E+03	1.3005E+03	1.3006E+03	1.3027E+03	1.3035E+03	**1.3003E+03**	1.3003E+03	1.3004E+03	1.3004E+03	1.3004E+03	1.3004E+03
	Std	9.0885E-02	8.8605E-02	1.0910E-01	1.2615E-01	1.2881E-01	1.1925E+00	5.7784E-01	8.7778E-02	**7.8873E-02**	1.0380E-01	9.9792E-02	9.3709E-02	1.0185E-01
CEC-14	Mean	**1.4003E+03**	1.4003E+03	1.4004E+03	1.4003E+03	1.4003E+03	1.4543E+03	1.4600E+03	1.4003E+03	1.4004E+03	1.4003E+03	1.4004E+03	1.4004E+03	1.4003E+03
	Std	**3.8003E-02**	1.5424E-01	1.3416E-01	5.3579E-02	1.7185E-01	1.8588E+01	2.1464E+01	1.2509E-01	1.7402E-01	4.7078E-02	1.9680E-01	1.6814E-01	1.2500E-01
CEC-15	Mean	1.5143E+03	1.5068E+03	1.5134E+03	1.5209E+03	1.5435E+03	5.8280E+03	7.6533E+03	**1.5062E+03**	1.5070E+03	1.5068E+03	1.5067E+03	1.5071E+03	1.5073E+03
	Std	3.7774E+00	2.4594E+00	3.6865E+00	6.3791E+00	1.5947E+01	3.8655E+03	5.3406E+03	**1.6530E+00**	2.0348E+00	2.1168E+00	2.5050E+00	1.9906E+00	3.6841E+00

Table 5 Results of the comparative methods on CEC2014 bechmark functions (CEC-16 to CEC-30)

Function		LDW-PSO	LCS-PSO	NDW-PSO	CSA	PPSO	POA	SHO	BF-1	BF-2	BF-3	BF-4	BF-5	BF-6
CEC-16	Mean	1.6123E+03	1.6115E+03	1.6125E+03	1.6118E+03	1.6125E+03	1.6119E+03	1.6120E+03	1.6116E+03	1.6114E+03	1.6116E+03	**1.6113E+03**	1.6119E+03	1.6119E+03
	Std	4.4268E-01	7.5021E-01	4.3190E-01	4.7100E-01	5.1349E-01	**3.5367E-01**	4.2329E-01	8.0341E-01	7.3484E-01	7.3361E-01	6.2855E-01	6.3345E-01	6.6156E-01
CEC-17	Mean	1.3666E+06	7.2333E+05	1.4633E+06	4.6142E+05	**8.2628E+04**	7.2205E+05	5.4170E+06	5.5537E+05	5.3192E+05	8.4551E+05	7.9627E+05	6.8528E+05	6.6813E+05
	Std	7.9900E+05	5.4652E+05	1.1209E+06	5.4354E+05	**1.0204E+05**	6.0139E+05	3.9934E+06	4.1694E+05	3.1929E+05	5.7782E+05	7.1594E+05	5.1113E+05	6.5503E+05
CEC-18	Mean	7.6037E+03	8.0151E+03	8.1262E+03	**2.8765E+03**	4.5579E+03	1.3115E+06	2.4527E+07	9.2641E+03	1.0987E+04	8.2537E+03	8.8645E+03	7.4879E+03	8.8377E+03
	Std	6.9852E+03	5.8787E+03	5.9446E+03	**8.1227E+02**	5.0063E+03	2.6135E+06	3.3942E+07	5.8912E+03	7.1889E+03	6.6336E+03	6.8929E+03	6.2091E+03	6.8994E+03
CEC-19	Mean	1.9147E+03	**1.9098E+03**	1.9121E+03	1.9240E+03	1.9286E+03	1.9536E+03	1.9841E+03	1.9195E+03	1.9143E+03	1.9137E+03	1.9173E+03	1.9206E+03	1.9187E+03
	Std	1.8405E+01	**2.4094E+00**	1.2942E+01	1.9669E+01	2.9296E+01	2.7853E+01	3.1032E+01	2.5247E+01	1.6564E+01	1.7715E+01	2.3547E+01	2.5478E+01	2.2165E+01
CEC-20	Mean	1.1251E+04	5.9780E+03	1.4383E+04	8.0245E+03	**3.5942E+03**	1.3535E+04	2.9525E+04	5.5298E+03	5.4733E+03	5.1340E+03	6.3718E+03	5.0449E+03	5.0957E+03
	Std	6.4713E+03	3.1019E+03	8.0907E+03	3.5129E+03	**1.9907E+03**	6.2030E+03	1.2032E+04	2.6259E+03	2.5698E+03	2.4464E+03	3.2424E+03	2.3596E+03	2.6266E+03
CEC-21	Mean	3.5841E+05	9.4567E+04	6.2648E+05	9.4466E+04	**2.2517E+04**	1.7498E+05	9.4710E+05	1.3890E+05	1.1250E+05	2.0000E+05	1.4900E+05	1.9988E+05	1.3584E+05
	Std	3.9439E+05	7.6500E+04	4.8472E+05	9.7883E+04	**1.4192E+04**	1.6561E+05	1.2009E+06	1.5592E+05	7.6460E+04	1.3739E+05	1.1160E+05	3.2369E+05	1.9602E+05
CEC-22	Mean	2.5791E+03	2.5524E+03	2.5859E+03	2.6585E+03	2.9007E+03	2.6610E+03	2.7407E+03	2.4641E+03	2.4503E+03	2.4978E+03	**2.4041E+03**	2.4895E+03	2.4610E+03
	Std	1.5852E+02	1.5192E+02	1.7800E+02	1.6681E+02	2.3163E+02	1.5956E+02	1.9170E+02	**1.3933E+02**	1.6834E+02	1.4559E+02	1.4660E+02	1.6039E+02	1.6201E+02
CEC-23	Mean	2.6167E+03	2.6163E+03	2.6159E+03	2.6349E+03	2.6117E+03	**2.5000E+03**	2.6584E+03	2.6162E+03	2.6164E+03	2.6160E+03	2.6157E+03	2.6163E+03	2.6167E+03
	Std	6.3456E-01	7.2126E-01	6.0758E-01	5.4952E+00	2.1087E+01	**8.4444E-14**	1.5428E+01	7.8231E-01	7.7409E-01	4.3220E-01	2.4655E-01	8.8873E-01	1.1312E+00
CEC-24	Mean	2.6358E+03	2.6375E+03	2.6387E+03	2.6203E+03	2.6002E+03	**2.6000E+03**	2.6000E+03	2.6413E+03	2.6401E+03	2.6390E+03	2.6385E+03	2.6414E+03	2.6433E+03
	Std	6.1218E+00	5.7310E+00	4.9224E+00	5.0082E+00	2.1018E-01	**0.0000E+00**	1.6109E-04	5.6420E+00	6.5199E+00	6.2594E+00	5.2343E+00	5.1152E+00	5.4947E+00
CEC-25	Mean	2.7125E+03	2.7115E+03	2.7116E+03	2.7039E+03	2.7002E+03	**2.7000E+03**	2.7000E+03	2.7108E+03	2.7117E+03	2.7105E+03	2.7113E+03	2.7120E+03	2.7129E+03
	Std	3.5954E+00	3.0613E+00	2.8542E+00	2.6243E+00	1.0361E+00	**8.4444E-14**	3.4817E-13	2.3136E+00	3.6724E+00	2.5144E+00	3.3059E+00	3.0790E+00	3.2441E+00
CEC-26	Mean	2.7105E+03	2.7003E+03	2.7295E+03	2.7004E+03	2.7005E+03	2.7081E+03	2.7313E+03	2.7003E+03	**2.7003E+03**	2.7003E+03	2.7004E+03	2.7004E+03	2.7004E+03
	Std	3.0707E+01	7.9309E-02	6.5241E+01	1.2219E-01	1.1602E-01	2.5008E+01	4.5763E+01	7.7149E-02	**6.8551E-02**	8.9558E-02	9.6271E-02	9.6523E-02	1.1201E-01
CEC-27	Mean	3.3163E+03	3.2197E+03	3.3399E+03	3.2660E+03	3.6566E+03	**2.9012E+03**	3.5890E+03	3.1086E+03	3.1338E+03	3.1090E+03	3.1096E+03	3.1082E+03	3.1083E+03
	Std	1.3477E+02	1.1381E+02	1.3237E+02	2.5972E+02	3.8382E+02	**3.7682E+00**	2.6902E+02	6.7397E+00	6.3849E+01	6.8941E+00	1.9414E+01	6.0457E+00	5.1413E+00
CEC-28	Mean	4.3088E+03	3.9175E+03	4.0693E+03	6.0729E+03	4.6908E+03	**3.0000E+03**	5.6627E+03	3.8631E+03	3.8472E+03	3.8398E+03	3.8251E+03	3.8749E+03	3.8891E+03
	Std	5.0803E+02	1.6511E+02	3.6187E+02	9.2805E+02	5.4633E+02	**4.0716E-04**	7.4352E+02	1.2734E+02	9.0052E+01	8.6556E+01	6.6062E+01	1.3480E+02	1.2701E+02
CEC-29	Mean	2.6470E+06	4.8077E+03	2.0858E+06	1.4465E+06	7.6599E+07	5.2812E+05	5.5364E+06	4.5350E+03	4.7760E+03	4.3362E+03	**4.2755E+03**	4.5833E+03	4.6741E+03
	Std	8.1097E+06	1.2192E+03	5.4606E+06	7.8712E+06	7.4025E+07	1.2604E+06	9.4125E+06	3.9484E+02	7.9430E+02	2.9866E+02	**2.6308E+02**	2.7221E+02	5.5859E+02
CEC-30	Mean	1.1349E+04	2.0074E+04	**1.0770E+04**	6.5344E+04	6.3265E+04	5.7278E+04	1.4962E+05	1.2623E+04	1.9328E+04	1.1305E+04	1.2013E+04	1.3144E+04	2.0089E+04
	Std	6.3903E+03	1.2604E+04	**5.5189E+03**	5.5584E+04	1.1942E+05	3.7498E+04	1.4194E+05	6.6472E+03	1.3371E+04	9.5078E+03	7.5813E+03	7.7929E+03	1.3373E+04

GHz (+ 16.0 GB of implemented RAM) processor under the 64-bit Windows 10 Pro operating system. Therefore, the results were based on double-precision floating-point number (64-bit) operations using MATLAB, and were managed using a 5-digit display, including the integer part. The design parameters of our NCS-PSOs are shown in Table 3. The selection of the basic functions BF-1 to BF-6 of the proposed method, the boundary values of each time-varying parameter as design parameters, and the modulation or time-decay rate were determined after some trials, considering their use in the 3-D path planning problem in the later section. It should be noted that the basic parameter design guideline is based on the assumption that symmetric crossing of acceleration coefficients is used, with crossing values around 1.5 and crossing times around $0.5t_{\max}$, which is always the case in the linear case.

6.2 Performance analysis of NCS-PSOs on ECE2014 benchmark functions

Tables 4 and 5 show the test results of NCS-PSOs and seven comparison algorithms, where the bold entries represent the best ones among all the algorithms. It can be seen that NCS-PSOs obtain better average fitness values on 15/30 functions. In the unimodal functions, the proposed BF-2 and BF-3 achieve the best means for the functions CEC-2 and CEC-3, with the BF-2 in particular having the best mean and variance. The conventional PPSO has the best mean and variance with respect to CEC-1.

For multimodal functions CEC-7, CEC- 8, CEC-9, CEC-11, CEC-12, CEC-13, CEC-15, and CEC-16, NCS-PSOs outperform other algorithms with stronger exploration performance. Moreover, BF-1 has the minimum Std result on function CEC-4. NCS-PSOs also shows less optimization performance on most of hybrid functions, except for CEC-22. The conventional PPSO has the best on functions CEC-17, CEC-20 and CEC-21. The conventional POA provides the best on many composition functions CEC-23, CEC-24, CEC-25, CEC-27, and CEC-28, whereas the proposed BF-2 and BF-4 can obtain the best on CEC-26 and CEC-29, respectively. Thus, the test results on CEC2014 benchmark functions show that the proposed NCS-PSOs are more competitive than other algorithms on unimodal and multimodal functions.

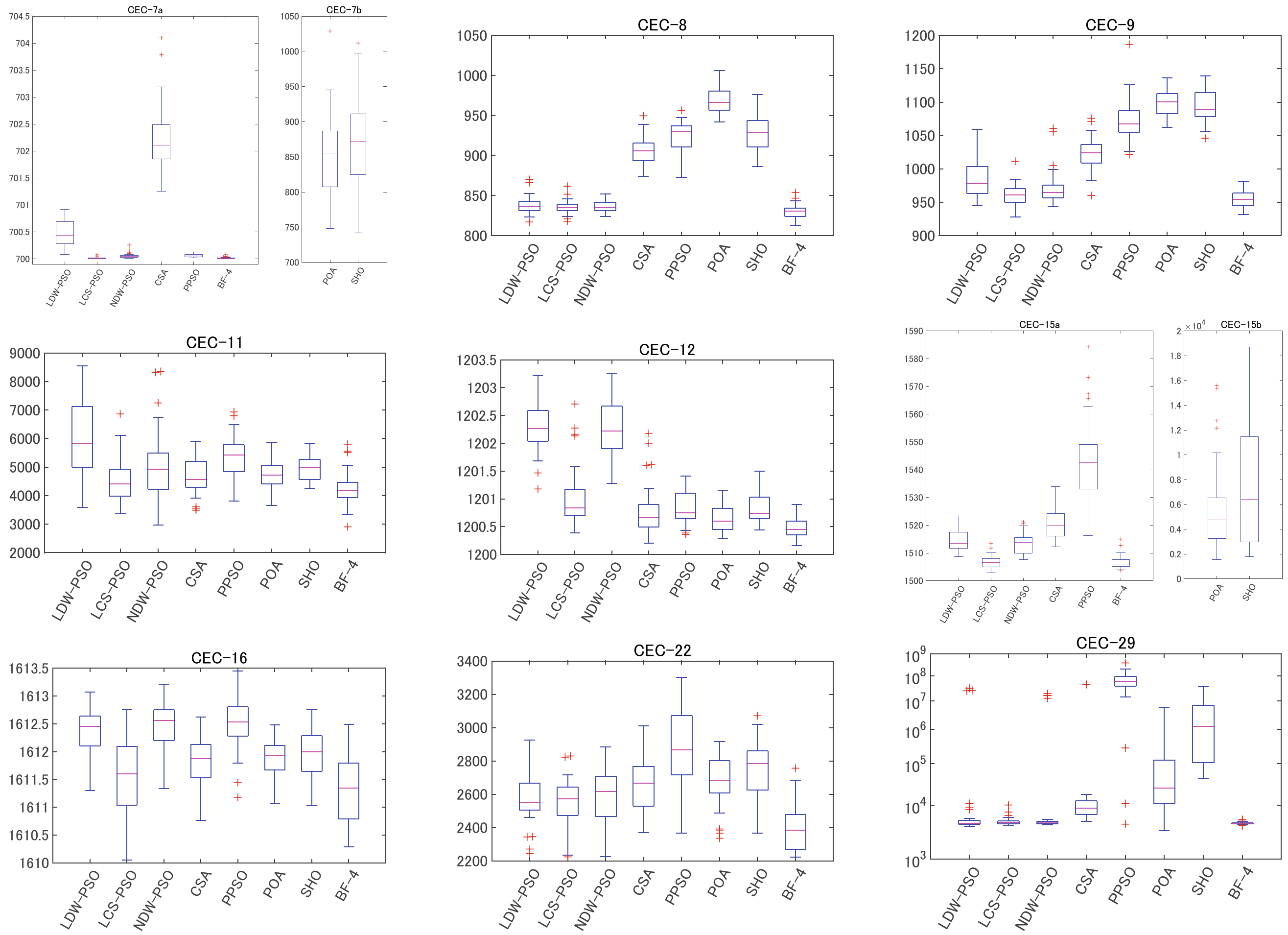


Fig. 11 Box plot of the proposed NCS-PSO and seven comparison algorithms on CEC2014 benchmark functions

Table 6 Wilcoxon rank-sum test results for BF-4 against other algorithms on CEC2014 benchmark functions

Function	BF-4 vs. LDW-PSO		BF-4 vs. LCS-PSO		BF-4 vs. NDW-PSO		BF-4 vs. CSA		BF-4 vs. PPSO		BF-4 vs. POA		BF-4 vs. SHO	
	p value	win	*p* value	win	*p* value	win	*p* value	win	*p* value	win	*p* value	win	*p* value	win
CEC-1	1.1567E-07	+	1.3832E-02	+	1.7294E-07	+	3.1967E-09	+	1.3272E-02	+	3.3384E-11	+	3.0199E-11	+
CEC-2	3.0199E-11	+	1.1882E-01	–	1.7649E-02	+	3.0199E-11	+	4.9818E-04	+	3.0199E-11	+	3.0199E-11	+
CEC-3	1.8090E-01	–	5.2640E-04	+	1.6687E-01	–	3.0199E-11	+	3.7782E-02	+	3.0199E-11	+	3.0199E-11	+
CEC-4	2.3243E-02	+	5.0114E-01	–	6.8432E-01	–	2.2539E-04	+	2.1544E-10	+	8.1527E-11	+	3.0199E-11	+
CEC-5	3.0199E-11	+	3.0199E-11	+	3.0199E-11	+	3.0199E-11	+	1.1937E-06	+	3.0199E-11	+	1.3289E-10	+
CEC-6	4.5530E-01	–	1.5367E-01	–	5.3951E-01	–	2.3715E-10	+	3.3384E-11	+	3.6897E-11	+	3.0199E-11	+
CEC-7	3.0199E-11	+	9.7052E-01	–	1.1567E-07	+	3.0199E-11	+	1.2023E-08	+	3.0199E-11	+	3.0199E-11	+
CEC-8	3.8481E-03	+	2.7086E-02	+	9.0688E-03	+	3.0199E-11	+	3.0199E-11	+	3.0199E-11	+	3.0199E-11	+
CEC-9	5.4620E-06	+	1.2597E-01	–	2.2658E-03	+	X 8.9934E-11	+	3.0199E-11	+	3.0199E-11	+	3.0199E-11	+
CEC-10	4.2039E-01	–	4.4272E-03	+	7.3940E-01	–	1.0937E-10	+	8.9934E-11	+	3.3384E-11	+	2.1544E-10	+
CEC-11	6.0460E-07	+	2.0095E-01	–	8.6844E-03	+	1.2212E-02	+	8.1975E-07	+	8.1200E-04	+	1.9963E-05	+
CEC-12	3.0199E-11	+	3.2555E-07	+	3.0199E-11	+	3.0339E-03	+	4.1178E-06	+	1.5638E-02	+	8.1975E-07	+
CEC-13	3.1830E-03	+	1.0000E+00	–	1.0277E-06	+	6.3560E-05	+	1.1023E-08	+	3.6897E-11	+	3.0199E-11	+
CEC-14	1.9579E-01	–	9.1171E-01	–	2.3985E-01	–	3.9527E-01	–	5.2014E-01	–	3.0199E-11	+	3.0199E-11	+
CEC-15	1.1737E-09	+	6.6273E-01	–	3.8249E-09	+	6.0658E-11	+	3.0199E-11	+	3.0199E-11	+	3.0199E-11	+
CEC-16	1.2541E-07	+	2.4581E-01	–	1.1023E-08	+	2.3800E-03	+	1.5581E-08	+	7.2951E-04	+	1.2477E-04	+
CEC-17	7.6973E-04	+	7.2827E-01	–	2.7548E-03	+	8.6844E-03	+	1.6947E-09	+	7.1719E-01	–	1.0702E-09	+
CEC-18	5.2014E-01	–	8.6499E-01	–	9.8231E-01	–	6.3560E-05	+	1.8575E-03	+	3.8202E-10	+	5.4941E-11	+
CEC-19	6.4142E-01	–	3.9527E-01	–	3.7904E-01	–	7.0430E-07	+	7.0430E-07	+	1.2541E-07	+	1.6947E-09	+
CEC-20	8.5641E-04	+	6.1001E-01	–	4.1178E-06	+	6.1452E-02	–	2.5974E-05	+	6.2828E-06	+	3.6897E-11	+
CEC-21	3.5012E-03	+	2.7086E-02	+	8.8829E-06	+	7.9590E-03	+	8.1014E-10	+	9.5873E-01	–	3.8053E-07	+
CEC-22	5.2650E-05	+	3.0059E-04	+	8.6634E-05	+	4.8011E-07	+	5.0723E-10	+	5.5999E-07	+	4.3106E-08	+
CEC-23	7.3803E-10	+	3.3681E-05	+	6.6273E-01	–	3.0199E-11	+	7.7387E-06	+	1.7203E-12	+	3.0199E-11	+
CEC-24	1.3345E-01	–	4.8252E-01	–	7.9585E-01	–	5.4941E-11	+ x	3.0199E-11	+	1.2118E-12	+	3.0199E-11	+
CEC-25	1.6238E-01	–	6.4142E-01	–	4.7335E-01	–	1.7769E-10	+	3.0199E-11	+	1.7203E-12	+	1.4059E-11	+
CEC-26	4.6756E-02	+	2.1506E-02	+	4.6390E-05	+	9.0490E-02	–	1.9963E-05	+	7.3846E-11	+	2.9137E-11	+
CEC-27	1.4643E-10	+	2.1327E-05	+	4.1997E-10	+	1.1737E-09	+	8.1465E-05	+	3.0010E-11	+	6.6955E-11	+
CEC-28	1.0937E-10	+	2.1506E-02	+	1.0763E-02	+	3.0199E-11	+	4.5043E-11	+	4.1109E-12	+	3.0199E-11	+
CEC-29	1.6238E-01	–	5.8282E-03	+	2.9205E-02	+	4.9752E-11	+	2.8716E-10	+	6.7650E-05	+	3.0199E-11	+
CEC-30	5.7929E-01	–	4.0330E-03	+	3.9527E-01	–	4.1997E-10	+	7.9782E-02	–	3.1589E-10	+	5.4941E-11	+
No. of ‘+’		20		14		19		27		28		28		30
No. of ‘–’		10		16		11		3		2		2		0

Figure 11 shows the boxplots for eight algorithms on CEC2014 benchmark functions, where BF-4 was selected out of six NCS-PSOs because of its superiority to other methods in NCS-PSOs. If '+' has appeared outside the upper fence, it implies that the algorithm has a poor search accuracy in 30 runs, whereas if '+' was outside the lower fence, it represents that the algorithm can be fully explored and exploited to search for the superior precision value. The linear crossing method LCS-PSO, the nonlinearly decreasing inertia weight method NDW-PSO and the proposed method BF-4 have much more '+' from the upper fence than the other conventional methods, but their medians are lower than those of the other methods, as can be seen from the visual inspection of Fig. 11.

The proposed method has the smallest difference between the upper and lower quartiles for CEC-11, CEC-12, and CEC-15. On the other hand, for other benchmark functions, there are several conventional methods with similar quartile differences to the proposed method. For example, in CEC-7, LCS-PSO and the proposed method, in CEC-8, LDW-PSO, NDW-PSO and the proposed method, and in CEC-9, LCS-PSO, NDW-PSO and the proposed method obtain similar differences between quartiles. The boxplots verify that BF-4, one of NCS-PSOs, has strong robustness and stability. Thus, BF-4 in NCS-PSOs shows superior optimization performance on CEC2014 benchmark functions.

6.3 Statistical analysis

In order to avoid accidental test results, statistical tests are performed by using the Wilcoxon rank-sum test statistical method [88] applying a significance level of 5%. Further this method is used to evaluate the effectiveness and superiority of NCS-PSOs. Each BF-i out of NCS-PSOs was used as the optimization algorithm, and pairwise comparison was made with the other seven algorithms. If the p value generated by the two algorithms is less than 5%, it indicates that there is a significant difference between the two algorithms in statistical significance. Otherwise, the difference between the two algorithms is not obvious. The Wilcoxon rank-sum tests (i.e., Mann-Whitney U tests) of NCS-PSOs were conducted on CEC2014 benchmark functions. Table 6 is the summary of test results for BF-4, where '+' represents significant difference and '−' represents poor significant difference. In this table, BF-4 conducts 210 (30 × 7) groups of experiments, among which 166 groups of data (i.e., 79%) shows significant differences. This result verifies that the optimization performance of the BF-4 algorithm in NCS-PSOs is better than the other seven comparison algorithms in the statistical sense.

Friedman rank test [22] is a nonparametric method that uses rank to implement significant differences for multiple population distributions. Friedman rank tests of 13 or eight algorithms were performed on the most challenging CEC2014 benchmark functions to compare their comprehensive average performance. Table 7 shows Friedman test results of 13 algorithms, whereas Tables 8 and 9 give the results of eight algorithms, one of which is changed from BF-1 to BF-6 in NCS-PSOs.

As is shown in Tables 7 to 9, any algorithm in NCS-PSOs has any rank among one to seven out of 13 ranks, together with verifying that one of NCS-PSOs acquires 79% in maximum and 72.4% in minimum in significant differences. Especially, BF-4 in NCS-PSOs ranked the first and the remaining algorithms in NCS-PSOs were significantly better than the other six comparison algorithms, which excluded LCS-PSO. Thus, Friedman rank tests demonstrate that the proposed NCS-PSOs are effective and stable.

7 Path optimization problem

In this paper, a 3D path planning for UAVs treated by Phung and Ha [58] is picked up, and the convergence speed and accuracy of the best solution are investigated in the difference between the initial or terminal boundary values in acceleration coefficients, and also in the difference between the crossing values (or crossing times), as well as in the variety of the basic functions for constituting nonlinear time-varying inertia weight and acceleration coefficients.

This 3D path planning for UAVs consists of each cost or constraint condition, which is described below.

7.1 Optimization of path length

Letting the jth path node in the ith particle be denoted by the coordinate point P_{ij}, which is corresponded to a way-

Table 7 Friedman rank test results for BF-1 to BF-6 and other algorithms on CEC2014 benchmark functions

Algorithm	Friedman rank test	Rank
LDW-PSO	8.6667	11
LCS-PSO	5.2333	2
NDW-PSO	8.2333	9
CSA	8.3667	10
PPSO	7.4000	8
POA	8.8333	12
SHO	11.1333	13
BF-1	5.6667	5
BF-2	5.4000	4
BF-3	5.2667	3
BF-4	4.2333	1
BF-5	5.8667	6
BF-6	6.7000	7

Table 8 Friedman rank test and Wilcoxon rank-sum test results for BF-i $(i = 1, 2, 3)$ and other algorithms on CEC2014 benchmark functions

Algorithm	LDW-PSO	LCS-PSO	NDW-PSO	CSA	PPSO	POA	SHO	BF-1
Average rank	4.7333	2.8000	4.5667	4.6333	4.6000	4.4000	5.0333	3.1000
Final rank	6	1	5	4	3	7	8	2
+/−	22/8	9/21	17/3	25/5	27/3	26/4	29/1	–
Algorithm	LDW-PSO	LCS-PSO	NDW-PSO	CSA	PPSO	POA	SHO	BF-2
Average rank	4.7667	2.8000	4.5667	4.6667	4.4333	5.0667	6.7667	2.9333
Final rank	6	1	4	5	3	7	8	2
+/−	23/7	6/24	21/9	25/5	27/3	27/3	29/1	–
Algorithm	LDW-PSO	LCS-PSO	NDW-PSO	CSA	PPSO	POA	SHO	BF-3
Average rank	4.8333	2.8000	4.6333	4.6333	4.4000	4.9667	6.7000	3.0333
Final rank	6	1	4	5	3	7	8	2
+/−	17/13	9/21	17/13	27/3	28/2	26/4	30/0	–

Table 9 Friedman rank test and Wilcoxon rank-sum test results for BF-i $(i = 4, 5, 6)$ and other algorithms on CEC2014 benchmark functions

Algorithm	LDW-PSO	LCS-PSO	NDW-PSO	CSA	PPSO	POA	SHO	BF-4
Average rank	4.8333	2.9000	4.7667	4.6667	4.4000	5.0667	6.7667	2.6000
Final rank	6	2	5	4	3	7	8	1
+/−	20/10	14/16	19/11	27/3	28/2	28/2	30/0	–
Algorithm	LDW-PSO	LCS-PSO	NDW-PSO	CSA	PPSO	POA	SHO	BF-5
Average rank	4.8000	2.7333	4.6667	4.5667	4.4000	5.0333	6.7000	3.1000
Final rank	6	1	5	4	3	7	8	2
+/−	23/7	7/23	19/11	25/5	26/4	26/4	26/4	–
Algorithm	LDW-PSO	LCS-PSO	NDW-PSO	CSA	PPSO	POA	SHO	BF-6
Average rank	4.7000	2.7000	4.6333	4.5667	4.3667	5.0000	6.6667	3.3667
Final rank	6	1	5	4	3	7	8	2
+/−	24/6	12/18	19/11	26/4	24/6	26/4	25/5	–

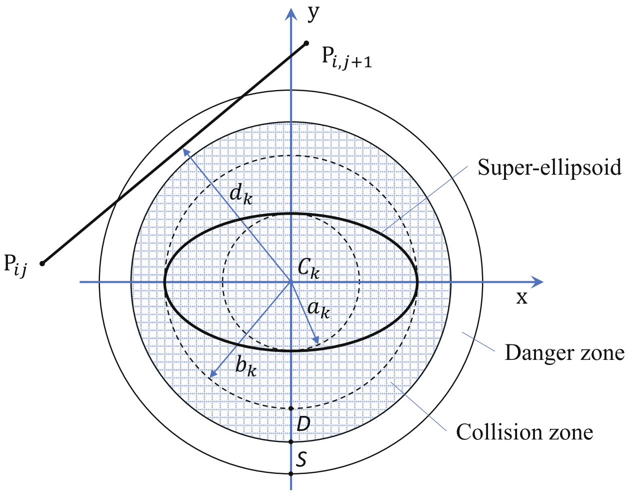


Fig. 12 Obstacle model and path vector

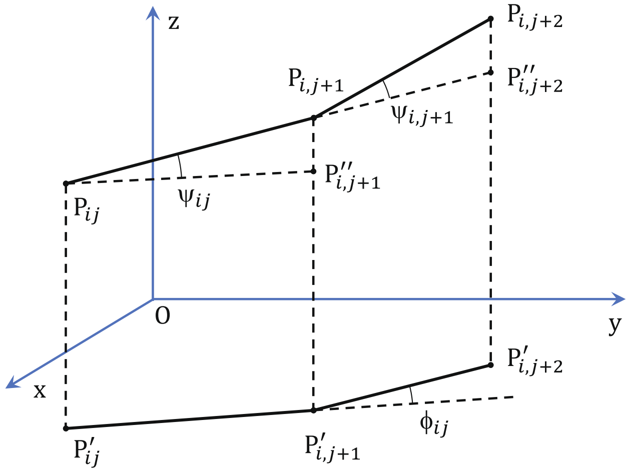


Fig. 13 Turning and climbing angles

point. When defining the Euclidean distance between two points, $\mathrm{P}_{ij}(x_{ij}, y_{ij}, z_{ij})$ and $\mathrm{P}_{i,j+1}(x_{i,j+1}, y_{i,j+1}, z_{i,j+1})$, as $\|\overrightarrow{\mathrm{P}_{ij}\mathrm{P}_{i,j+1}}\|$, the cost related to the total path length F_1 is represented by

$$F_1(X_i) = \sum_{j=1}^{n_w-1} \|\overrightarrow{\mathrm{P}_{ij}\mathrm{P}_{i,j+1}}\| \tag{90}$$

where X_i denotes the path in the ith particle and n_w is the number of way points to be passed. In this paper, X_i is expressed by a list of n_w way points and the path length is minimized in total. Instead of using $\boldsymbol{x}^i$, introducing a vector composed of 3D information, which represents the following point $\mathrm{P}_{ij}(x_{ij}, y_{ij}, z_{ij})$ as the ith particle,

$$X_i = [x_{i1}\ y_{i1}\ z_{i1}\ x_{i2}\ y_{i2}\ z_{i2} \cdots x_{iN}\ y_{iN}\ z_{iN}]^T$$

the swarm X is equivalent to a $3N \times M$ matrix such as $X = [X_1\ X_2\ \cdots X_M]$. Note that if the start and terminal points are fixed in all the paths, then any path with n_w way points is consequently reduced to be $3N$ dimension, where $N = n_w - 2$.

7.2 Modelling of obstacle and collision condition

Apart from the optimization of path planning, the planned path must secure the safety by avoiding the threats caused with the obstacles which appear in operation space, and guiding UAVs. Then, assume that there are K threats (i.e., obstacles) here, and that each projection is described by a superellipse which has the center c_k, minor-axis a_k, and major-axis b_k as shown in Fig. 12. To the obtained path segment $\|\overrightarrow{\mathrm{P}_{ij}\mathrm{P}_{i,j+1}}\|$, the related threat cost is assumed to be proportional to the distance d_k up to the center of an obstacle C_k. Moreover, letting the maximum diameter of UAVs be D and the hazarder distance to a collision zone be S, the threat cost F_2 is, for the K obstacles, reduced to

$$F_2(X_i) = \sum_{j=1}^{n_w-1} \sum_{k=1}^{K} T_k(\overrightarrow{\mathrm{P}_{ij}\mathrm{P}_{i,j+1}}) \tag{91}$$

with

$$T_k(\overrightarrow{\mathrm{P}_{ij}\mathrm{P}_{i,j+1}}) = \begin{cases} 0 & \text{if } d_k > S + D + b_k \\ (S + D + b_k) - d_k & \text{if } D + b_k < d_k \le S + D + b_k \\ \text{inf} & \text{if } d_k \le D + b_k \end{cases} \tag{92}$$

where $T_k(\overrightarrow{\mathrm{P}_{ij}\mathrm{P}_{i,j+1}})$ denotes the threat cost between any kth obstacle and the path segment $\overrightarrow{\mathrm{P}_{ij}\mathrm{P}_{i,j+1}}$. Note here that the diameter D is determined depending on the size of an UAV, but S is set to from tens of meters to hundreds of meters [58] according to the variety of obstacles and environments.

7.3 Altitude cost

The flying altitude of UAVs is different from every practical operational configurations, i.e., every applications, so that it is often constrained between the maximum and minimum heights. Now defining the minimum heights and maximum heights be $h_{\min}$ and $h_{\max}$ respectively, the altitude cost H_{ij} related to a waypoint P_{ij} is calculated by

$$H_{ij} = \begin{cases} \left| h_{ij} - \frac{h_{\max}+h_{\min}}{2} \right| & \text{if } h_{\min} \le h_{ij} \le h_{\max} \\ \text{inf} & \text{otherwise} \end{cases} \tag{93}$$

where h_{ij} denotes the flight height with respect to the ground and differs from the altitude from the sea level. This H_{ij} keeps an average height so as to penalize an outlier. By summing up H_{ij} for all waypoints, the total altitude cost is described by

$$F_3(X_i) = \sum_{j=1}^{n_w} H_{ij} \tag{94}$$

7.4 Smooth cost

The turning and climbing rates, which are essential to generate a feasible path, are evaluated as the smooth cost in UAVs [24, 58]. The turning angle ϕ_{ij} denotes the angle between $\overrightarrow{\mathrm{P}'_{ij}\mathrm{P}'_{i,j+1}}$ and $\overrightarrow{\mathrm{P}'_{i,j+1}\mathrm{P}'_{i,j+2}}$, which are the projections of two consecutive path segments $\overrightarrow{\mathrm{P}_{ij}\mathrm{P}_{i,j+1}}$ and $\overrightarrow{\mathrm{P}_{i,j+1}\mathrm{P}_{i,j+2}}$ onto the horizontal plane O_{xy} (see Fig. 13). Therefore, using a relationship between the inner product of two vectors and the angle formed by them, it yields [24] that

$$\phi_{ij} = \cos^{-1}\left(\frac{\overrightarrow{\mathrm{P}'_{ij}\mathrm{P}'_{i,j+1}} \cdot \overrightarrow{\mathrm{P}'_{i,j+1}\mathrm{P}'_{i,j+2}}}{\|\overrightarrow{\mathrm{P}'_{ij}\mathrm{P}'_{i,j+1}}\| \|\overrightarrow{\mathrm{P}'_{i,j+1}\mathrm{P}'_{i,j+2}}\|} \right) \tag{95}$$

Since the magnitude of outer product of two vectors is known to be reduced to the area of the parallelogram formed by them, it is also obtained that

$$\phi_{ij} = \sin^{-1}\left(\frac{\|\overrightarrow{\mathrm{P}'_{ij}\mathrm{P}'_{i,j+1}} \times \overrightarrow{\mathrm{P}'_{i,j+1}\mathrm{P}'_{i,j+2}}\|}{\|\overrightarrow{\mathrm{P}'_{ij}\mathrm{P}'_{i,j+1}}\| \|\overrightarrow{\mathrm{P}'_{i,j+1}\mathrm{P}'_{i,j+2}}\|} \right) \tag{96}$$

Combining them gives the following equation [58]:

$$\phi_{ij} = \tan^{-1}\left(\frac{\|\overrightarrow{\mathrm{P}'_{ij}\mathrm{P}'_{i,j+1}} \times \overrightarrow{\mathrm{P}'_{i,j+1}\mathrm{P}'_{i,j+2}}\|}{\overrightarrow{\mathrm{P}'_{ij}\mathrm{P}'_{i,j+1}} \cdot \overrightarrow{\mathrm{P}'_{i,j+1}\mathrm{P}'_{i,j+2}}} \right) \tag{97}$$

Thus, the smooth cost on turning angle is given by

$$F_4(X_i) = \sum_{j=1}^{n_w-2} \phi_{ij} \tag{98}$$

where

$$\phi_{ij} = \begin{cases} \phi_{ij} & \text{if } \phi_{ij} > \phi_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{99}$$

in which $\phi_{\max}$ is an allowable turning angle.

On the other hand, the climbing angle ψ_{ij} is coming from the angle between the path segment $\overrightarrow{\mathrm{P}_{ij}\mathrm{P}_{i,j+1}}$ and the projection of it onto the horizontal plane, i.e., $\overrightarrow{\mathrm{P}'_{ij}\mathrm{P}'_{i,j+1}}$, so that

$$\psi_{ij} = \tan^{-1}\left(\frac{z_{i,j+1} - z_{i,j}}{\|\overrightarrow{\mathrm{P}'_{ij}\mathrm{P}'_{i,j+1}}\|}\right) \tag{100}$$

Thus, if the climbing rate is defined by $d\psi_{ij} \overset{\triangle}{=} |\psi_{ij} - \psi_{i,j+1}|$, then, using the total climbing rates, the smooth cost on climbing rate is set to

$$F_5(X_i) = \sum_{j=1}^{n_w-1} d\psi_{ij} \tag{101}$$

where

$$d\psi_{ij} = \begin{cases} d\psi_{ij} & \text{if } d\psi_{ij} > d\psi_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{102}$$

in which $d\psi_{\max}$ denotes an allowable climbing rate angle.

7.5 Overall cost function

Considering the optimality, safety and feasibility constraints related to a path X_i, the overall cost function is described by the following form:

$$F(X_i) = \sum_{k=1}^{5} \eta_k F_k(X_i) \tag{103}$$

where η_k denotes the weight factor, and $F_k(X_i)$, $k = 1, ..., 5$ are the partial cost functions described above.

8 Examples of 3D path planning

8.1 Spherical vector-based PSO for UAV path planning

Instead of solving directly the waypoint vectors in the Cartesian coordinate system, $\mathrm{P}_{ij} = (x_{ij}, y_{ij}, z_{ij})$, $i = 1, ..., M$, $j = 1, ..., N$, $N = n - 2$, spherical vector-based PSO (SPSO) solves once three components in the spherical coordinate system, i.e., the magnitude $\rho \in (0, path\_length)$, the elevation angle $\psi \in (-\pi/2, \pi/2)$ and the azimuth angle $\phi \in (-\pi, \pi)$, where n denotes the number of waypoints including the initial and terminal values. Then, $\mathrm{U}_{ij} = (\rho_{ij}, \psi_{ij}, \phi_{ij})$ is transformed into a waypoint P_{ij} through the following equations:

$$x_{ij} = x_{i,j-1} + \rho_{ij} \sin\psi_{ij} \cos\phi_{ij} \tag{104}$$

$$y_{ij} = y_{i,j-1} + \rho_{ij} \sin\psi_{ij} \sin\phi_{ij} \tag{105}$$

$$z_{ij} = z_{i,j-1} + \rho_{ij} \cos\psi_{ij} \tag{106}$$

The scenario used in the following optimization is Scenario 3, in which there exist six obstacles, out of eight scenarios studied in Phung and Ha [58]. Considering any cylinders as obstacles, in other words, assuming that each threat projection is a circle, i.e., $a_k = b_k$ in previous section, the radius R_k [m] and 3D position (x_k, y_k, z_k) [m] are respectively given by

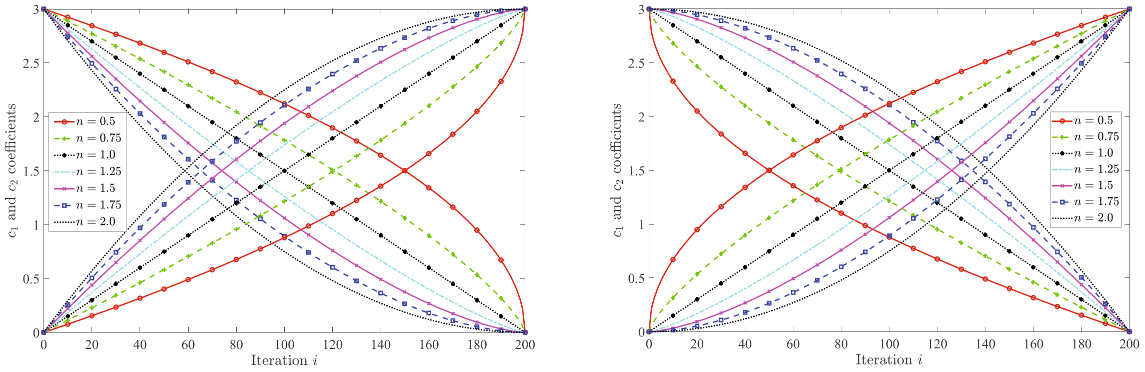
80;	(400, 500, 100) for $k = 1$
70;	(600, 200, 150) for $k = 2$
80;	(500, 350, 150) for $k = 3$
70;	(350, 200, 150) for $k = 4$
70;	(700, 550, 150) for $k = 5$
80;	(650, 750, 150) for $k = 6$

For the path to be optimized, the initial and terminal positions are assumed to be provided by (200, 100, 150) [m] and (800, 800, 150) [m], respectively, and the number of waypoints is fixed to $n_w = 12$ (i.e., the number of line segments is set to $N = 10$). The swarm size is set to $M = 500$, and the maximum number of iterations is fixed to $t_{\max} = 200$, while setting weight factors in the cost function such as $\eta_1 = 5$, $\eta_2 = \eta_4 = \eta_5 = 1$, and $\eta_3 = 10$. Note here that, for the penalty in the threat cost to judge the collision with an obstacle or in the altitude cost, the value of 15, 000 is applied to represent a finite cost value, instead of using inf. Moreover, the average cost with 20 trials and its standard deviation are evaluated to compare each other. Additionally, note that the basic function comprising the algorithm was chosen in terms of whether it has well-posed boundary conditions and

Table 10 Two crossing-time functions: The upper is for Chatterjee and Siarry [11], whereas the bottom is for Wang and Liu [86]

	Modulation index n						
	0.5	0.75	1.0	1.25	1.5	1.75	2.0
$t_{cross} = t_{\max}[1 - (\frac{1}{2})^{1/n}]$	150	121	100	86	75	66	59[1]
$t_{cross} = t_{\max}(\frac{1}{2})^{1/n}$	50	80	100	115	126	135	142

[1] Crossing time was calculated through the ceiling function, i.e., $\lceil t_{cross} \rceil$



(a) Use of a basic function by Chatterjee and Siarry [11]

(b) Use of a basic function by Wang and Liu [86]

Fig. 14 Time history of acceleration coefficients with different modulation indeces n

Table 11 Optimized results of using a basic function $(1 - t/t_{\max})^n$ (Chatterjee and Siarry [11]), where $c_{1e} = c_{2s} = 0.0$

	Modulation index n						
	0.5	0.75	1.0	1.25	1.5	1.75	2.0
(a) A case of $c_{1s} = c_{2e} = 2.0$, $c_{cross} = 1.0$							
Mean	**4684**	4742	4710	4711	4754	4752	4736
Std.	4.59	140	105	105	171	170	143
Conv. rate (%)	100	65	90	85	85	85	75
(b) A case of $c_{1s} = c_{2e} = 2.5$, $c_{cross} = 1.25$							
Mean	4727	4724	4747	4747	**4677**	**4675**	**4699**
Std.	142	145	169	168	5.65	3.78	105
Conv. rate (%)	90	90	85	85	100	100	95
(c) A case of $c_{1s} = c_{2e} = 3.0$, $c_{cross} = 1.5$							
Mean	**4696**	**4677**	**4676**	**4672**	4718	**4696**	**4694**
Std.	23.5	5.46	4.91	5.13	143	103	103
Conv. rate (%)	80	100	100	100	90	95	95
(d) A case of $c_{1s} = c_{2e} = 3.5$, $c_{cross} = 1.75$							
Mean	4787	4731	4716	4712	4754	4727	4723
Std.	29.9	131	107	105	178	152	144
Conv. rate (%)	0	50	70	85	80	85	90
(e) A case of $c_{1s} = c_{2e} = 4.0$, $c_{cross} = 2.0$							
Mean	5088	4968	4852	4833	4812	4870	4777
Std.	122	187	163	149	131	220	113
Conv. rate (%)	0	0	0	15	40	30	60

Table 12 Optimized results of using a basic function $(t/t_{\max})^n$ (Wang and Liu [86]), where $c_{1e} = c_{2s} = 0.0$

	Modulation index n						
	0.5	0.75	1.0	1.25	1.5	1.75	2.0
(a) A case of $c_{1s} = c_{2e} = 2.0$, $c_{cross} = 1.0$							
Mean	4757	4808	4714	**4692**	4709	4784	4733
Std.	175	208	103	11.2	103	195	142
Conv. rate (%)	85	60	80	75	85	75	80
(b) A case of $c_{1s} = c_{2e} = 2.5$, $c_{cross} = 1.25$							
Mean	4816	4726	4749	4703	4725	**4681**	4728
Std.	217	140	169	104	141	8.45	143
Conv. rate (%)	70	85	85	95	90	95	90
(c) A case of $c_{1s} = c_{2e} = 3.0$, $c_{cross} = 1.5$							
Mean	4766	**4673**	4723	4702	**4679**	4707	**4688**
Std.	190	4.67	144	103	13.4	108	8.98
Conv. rate (%)	80	100	90	95	95	85	90
(d) A case of $c_{1s} = c_{2e} = 3.5$, $c_{cross} = 1.75$							
Mean	4774	4751	4716	4722	4722	4729	4763
Std.	197	177	105	103	22.5	31.1	38.5
Conv. rate (%)	75	85	75	60	25	10	0
(e) A case of $c_{1s} = c_{2e} = 4.0$, $c_{cross} = 2.0$							
Mean	4853	4832	4913	4899	4934	5015	5007
Std.	216	183	162	153	141	161	104
Conv. rate (%)	0	0	0	0	0	0	0

whether it includes additional parameters such as time decay rate and modulation index.

8.2 Using of well-posed basic functions in symmetric crossing strategy

Here, let us apply well-posed basic functions to construct nonlinear time-varying inertia weight and acceleration coefficients, and consider the use of two acceleration coefficients in a symmetric crossing strategy. Therefore, the monotonically decreasing $(1-t/t_{\max})^n$ used in Chatterjee and Siarry [11] is first selected as a basic function, and constitute $w(t)$, $c_1(t)$, and $c_2(t)$ by using

$$a(t) = (a_s - a_e)(1 - t/t_{\max})^n + a_e$$

where $a_s > a_e$ so as to obtain monotonically decreasing $w(t)$ and $c_1(t)$, whereas $a_s < a_e$ so as to realize a monotonically increasing $c_2(t)$. In addition, let us select the monotonically increasing $(t/t_{\max})^n$ used in Wang and Liu [86] as another basic function, and constitute $w(t)$, $c_1(t)$, and $c_2(t)$ by applying

$$a(t) = (a_e - a_s)(t/t_{\max})^n + a_s$$

where the same conditions of a_s and a_e as used in above are required for making $w(t)$ and $c_1(t)$ be monotonically decreasing functions, while reducing $c_2(t)$ to a monotonically increasing function. Note of course that it need not obtain fictitious initial or terminal values, because these basic functions satisfy the boundary values well.

Letting the crossing strategy between $c_1(t)$ and $c_2(t)$ be symmetric, the duration from $t = 150$ ($n = 0.5$) to $t = 59$

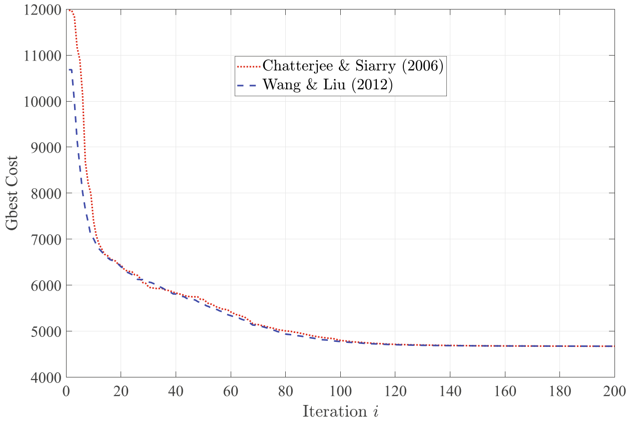
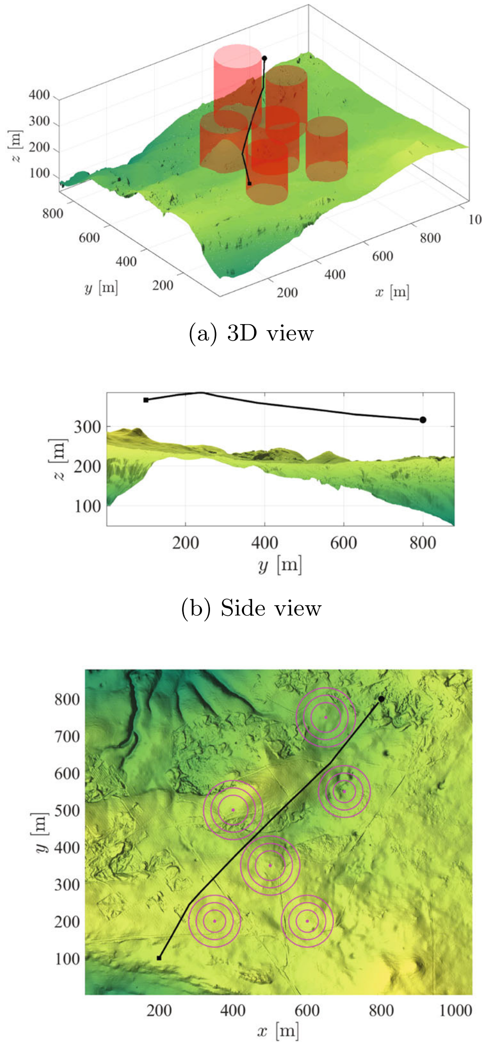


Fig. 15 Time history of gbest cost with basic functions due to Chatterjee and Siarry [11] with $n = 1.25$ and Wang and Liu [86] with $n = 0.75$, when using $c_{1s} = 3.0$, $c_{1e} = 0.0$

($n = 2.0$) was searched in the basic function due to Chatterjee and Siarry [11], as shown in Table 10, where refer to the previous section on the crossing time. On the other hand, the duration from $t = 50$ ($n = 0.5$) to $t = 142$ ($n = 2.0$) was searched in the basic function due to Wang and Liu [86]. The relationship between n and crossing time shows that t_{cross} varies inversely with the value of n in the one by Chatterjee and Siarry [11], while t_{cross} varies in direct proportion to the value of n in the one by Wang and Liu [86]. Figure 14 shows the time history of the accelerations in the difference of n for both methods, where the values of $w_s = 0.99$ and $w_e = 0.1$ were set commonly in all cases to the inertia weight. When setting $c_{1s} = 2.0, 2.5, 3.0, 3.5, 4.0$, $c_{1e} = 0.0$, $c_{2s} = c_{1e}$, $c_{2e} = c_{1s}$, their search results are tabulated in Table 11 and Table 12, respectively, where the bold entries represent the mean global best costs blow 4700. Note here that the crossing value is given by $c_{cross} = 0.5(c_{1e} + c_{2e})$ because of adopting a symmetric crossing strategy, and the convergence rate denotes $(N_{suc}/20) \times 100$, where N_{suc} is defined as the number in which the global best cost (Gbest) became less than 4700.

It is found from Table 11 that about the basic function of Chatterjee and Siarry [11], in crossing value $c_{cross} = 1.25$, the good result is obtained from $t_{cross} = 75$ to $t_{cross} = 59$ in the first half of the search time. Moreover, in crossing value $c_{cross} = 1.5$, it turns out that the good result is mostly obtained in search time all over the districts from $t_{cross} = 150$ to $t_{cross} = 59$. However, as for the thing c_{cross} is near 2.0 or 1.0, the convergence accuracy has deteriorated. Although tabulating the data was omitted, even when crossing value c_{cross} is the same, it is the tendency for a convergence accuracy not to be so good in $c_{1e} = c_{2s} \neq 0$. Similarly, the good result is obtained also about the basic function of Wang and Liu [86] in crossing value $c_{cross} = 1.25$ and $c_{cross} = 1.5$,

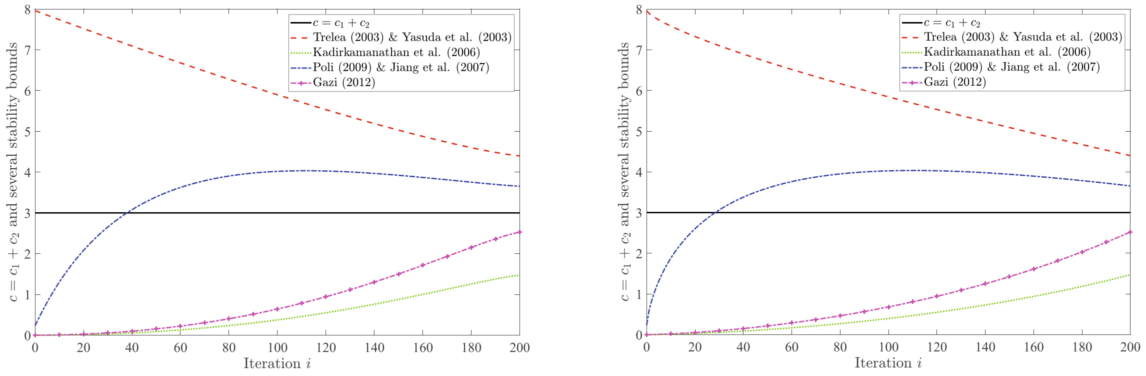


(a) 3D view

(b) Side view

(c) Top view

Fig. 17 One sample path optimized by using a basic function due to Chatterjee and Siarry [11]



(a) Bounds due to Chatterjee and Siarry [11] with $n = 1.25$

(b) Bounds due to Wang and Liu [86] with $n = 0.75$

Fig. 16 Stability bounds with different basic functions when using $c_{1s} = 3.0$, $c_{1e} = 0.0$

Table 13 Optimized results of using basic functions $e^{-(t_{\max}-t)/t_{\max}}$ (Method A) and $e^{-t_{\max}/(t_{\max}-t)}$ (Method B), where $c_{1ini} = c_{2e}, c_{2ini} = c_{1e} = 0.0$ (a symmetric case)

	c_{1ini}				
	2.0	2.5	3.0	3.5	4.0
(a) Method A: $e^{-(t_{\max}-t)/t_{\max}}$, where $t_{cross} = 125$					
c_{cross}	1.0	1.25	1.5	1.75	2.0
Mean	**4688**	**4678**	4701	4726	4967
Std.	11.4	9.63	106	44.0	185
Conv. rate (%)	85	95	95	30	0
(b) Method B: $e^{-t_{\max}/(t_{\max}-t)}$, where $t_{cross} = 82$					
c_{cross}	1.0	1.25	1.5	1.75	2.0
Mean	4707	4721	4720	**4683**	4850
Std.	103	143	143	20.0	195
Conv. rate (%)	90	90	90	100	5

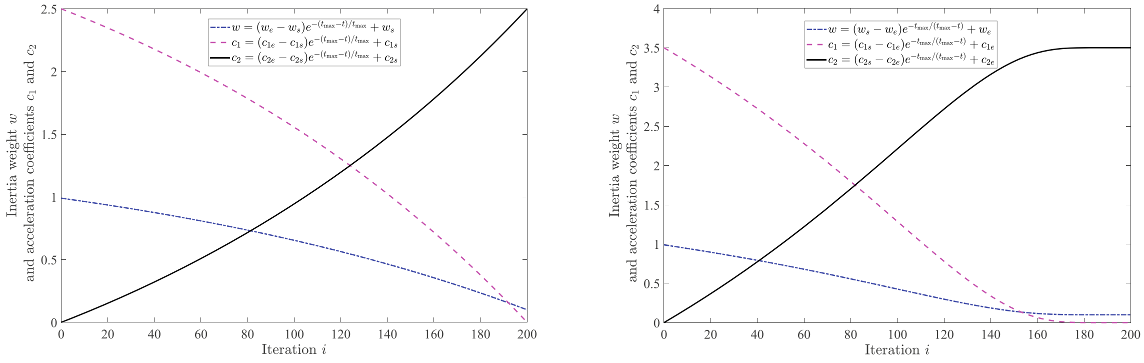
but it is not as uniformly good over the time as obtained by Chatterjee and Siarry [11].

Figure 15 shows that, for the case of $c_{1s} = 3.0$ and $c_{1e} = 0.0$, the time history of the global best cost due to Chatterjee and Siarry [11] with $n = 1.25$ is compared to one due to Wang and Liu [86] with $n = 0.75$, Fig. 16 shows the time history of stability bounds in the corresponding case (refer to Appendix on each stability bound equation), and Fig. 17 gives the 3D view, side view and top view that depict the path search result in one sample when using the method of Chatterjee and Siarry [11] with $n = 1.25$ at the same case. As can be seen from Fig. 16 that in the stability bound due to Poli [59] and Jiang et al. [35], the method due to Wang and Liu [86] gives the start of stability bound slightly faster than that due to Chatterjee and Siarry [11], it is also slightly faster in the transient search speed of Fig. 15 than Chatterjee and Siarry [11], but there is no difference at all in the search ability between both methods after 100 generations.

In the proposed search method based on a crossing strategy in which nonlinear time-varying weight and coefficients, $w(t)$, $c_1(t)$ and $c_2(t)$, are available, the criteria due to Kadirkamanathan et al. [37] and Gazi [25] do not satisfy the bound condition of stability over the search duration as found from the later examples, while those due to Trelea [83] and Yasuda et al. [95] (or Order-1 stability criterion) and Poli [59] and Jiang et al. [35] (or Order-2 stability criterion) satisfy the bound condition of stability as shown in Fig. 16. Thus, note that, although it needs to check the Order-1 stability criterion and the Order-2 stability criterion after that, the Order-1 stability criterion is relaxed too much, so that it concentrates on only the Order-2 criterion in fact.

8.3 Using of basic functions with ill-posed initial boundary in symmetric crossing strategy

The monotonically increasing $e^{-(t_{\max}-t)/t_{\max}}$ (Method A) and decreasing $e^{-t_{\max}/(t_{\max}-t)}$ (Method B) are here adopted as



(a) Use of a basic function $e^{-(t_{\max}-t)/t_{\max}}$ (Method A)

(b) Use of a basic function $e^{-t_{\max}/(t_{\max}-t)}$ (Method B)

Fig. 18 Time history of acceleration coefficients and inertia weight due to Methods A and B, where $c_{1ini} = 2.5$, $c_{2ini} = 0.0$ for the former, whereas $c_{1ini} = 3.5$, $c_{2ini} = 0.0$ for the latter

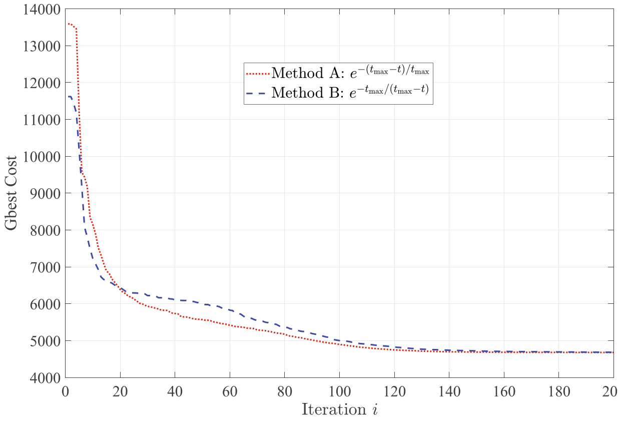


Fig. 19 Time history of gbest cost with basic functions due to Method C with $\alpha = 1.0$ and Method D with $\alpha = 0.6$, when using $c_{1s} = 3.0, c_{1ter} = 0.0$

basic functions to constitute $w(t)$ or $c_i(t)$. Since these do not satisfy the initial boundary values well, the fictitious boundary values should be obtained. Note that the latter function is a basic function that was used to form a deterministic weight $w(t)$, random adaptive coefficients $c_1(t)$ and $c_2(t)$, for improving a PSO-PF (particle swarm optimization particle filter) in Wu et al. [89], whereas the former function is of using the reciprocal value of the exponent $t_{\max}/(t_{\max}-t)$ in the latter function. Applying the fictitious initial boundary values when the initial boundary condition is ill-posed in the previous section, the inertia weight and acceleration coefficients for Method A and Method B are constituted respectively by using the following equations:

$$a_s = \left(\frac{e}{e-1}\right) a_{ini} - \frac{a_e}{e-1}$$
$$a(t) = (a_e - a_s)e^{-(t_{\max}-t)/t_{\max}} + a_s \quad \text{for Method A}$$

Table 14 Crossing-time of $c_1(t)$ and $c_2(t)$ for a symmetric crossing strategy, when using $e^{-\alpha t/t_{\max}}$ (Method C) as a basic function

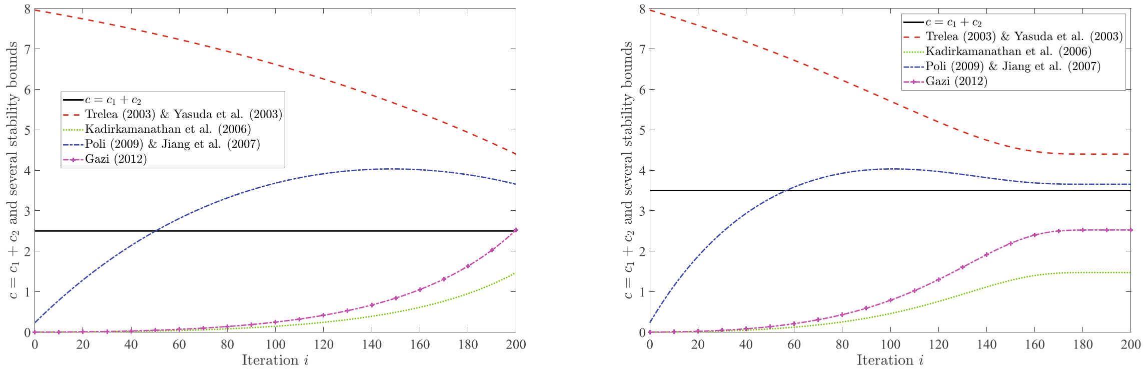
	Decay rate α						
	0.1	0.5	1.0	2.0	3.0	4.0	5.0
$t_{cross} = -\frac{t_{\max}}{\alpha}(\ln \frac{1+1/e^{\alpha}}{2})$	96	88	76	57	43	34	28

and

$$a_s = e a_{ini} + a_e(1-e)$$
$$a(t) = (a_s - a_e)e^{-t_{\max}/(t_{\max}-t)} + a_e \quad \text{for Method B}$$

Table 13a shows the result in which a symmetric crossing search was conducted with the former basic function, under the condition of $c_{2ini} = c_{1e} = 0.0$. Fig. 18a shows the time histories of the inertia weight and acceleration coefficients when setting $c_{1ini} = 2.5$. In this method, since the crossing time is given by $t_{cross} = t_{\max} - t_{\max}[1 - \ln(\frac{e+1}{2})]$, it results in $t_{cross} = 125$ uniquely, that is, the global search strategy is taken later than the mid crossing time. Thus, after achieving the local search to some extent, it is transferred to a global search so that the good results are obtained even in the crossing values of 1.0 and 1.25.

Table 13b shows the result in which a symmetric crossing search was conducted with the latter basic function, under the same condition as used in the former function. Fig. 18b shows the time histories of the inertia weight and acceleration coefficients when setting $c_{1ini} = 3.5$. In this method, since the crossing time is given by $t_{cross} = t_{\max} - \frac{t_{\max}}{1-\ln(0.5)}$, it results in $t_{cross} = 82$ uniquely, that is, the global search strategy is taken earlier than the mid crossing time. Thus, since the local search is not achieved sufficiently, the search ability is inferior at pin points (such as $c_{ini} = 2.5$ and 3.0) to the former, but the convergence rate results better than 90%



(a) Bounds due to Method A with $c_{1ini} = 2.5, c_{2ini} = 0.0$

(b) Bounds due to Method B with $c_{1ini} = 3.5, c_{2ini} = 0.0$

Fig. 20 Stability bounds due to Methods A and B, where $c_{1ini} = 2.5, c_{2ini} = 0.0$ for the former, whereas $c_{1ini} = 3.5, c_{2ini} = 0.0$ for the latter

Table 15 Crossing-time of $c_1(t)$ and $c_2(t)$ for a symmetric crossing strategy, when using $e^{-\alpha t_{\max}/t}$ (Method D) as a basic function

	Decay rate α						
	0.2	0.4	0.6	0.7	0.8	1.0	1.2
$t_{cross} = \frac{t_{\max}}{1-\frac{\ln(0.5)}{\alpha}}$	45	74	93	101	108	119	127

are obtained at the initial setting regions ($c_{ini} = 2.0 \rightarrow 3.5$), which are wider than the former.

Figure 19 shows the time history comparision of the global best cost in both methods at the same conditon, and Fig. 20 shows the time history of stability bounds in the corresponding case. As can be seen from Fig. 19, it is found that the convergence speed of the Method B is fast transiently up to $t = 20$, but rather, that of the Method A is fast at from $t = 20$ to around $t = 120$.

8.4 Using of basic functions with ill-posed terminal boundary in symmetric crossing strategy

Here, a monotonically decreasing $e^{-\alpha t/t_{\max}}$ (Method C) due to Lu et al. [50] and a monotonically increasing $e^{-\alpha t_{\max}/t}$ (Method D) are adopted as the basic functions that make up $w(t)$ or $c_i(t)$. Since these do not satisfy the terminal boundary conditions well, a fictitious terminal value has to be found. However, it should be noted that Method D uses the reciprocal of $t/t_{\max}$ in the basic function. As a result, Method C and Method D use respectively the following coefficient formulas:

$$a_e = \left(\frac{e^{\alpha}}{e^{\alpha}-1}\right) a_{ter} - \frac{a_s}{e^{\alpha}-1}$$

$$a(t) = (a_s - a_e)e^{-\alpha t/t_{\max}} + a_e \quad \text{for Method C}$$

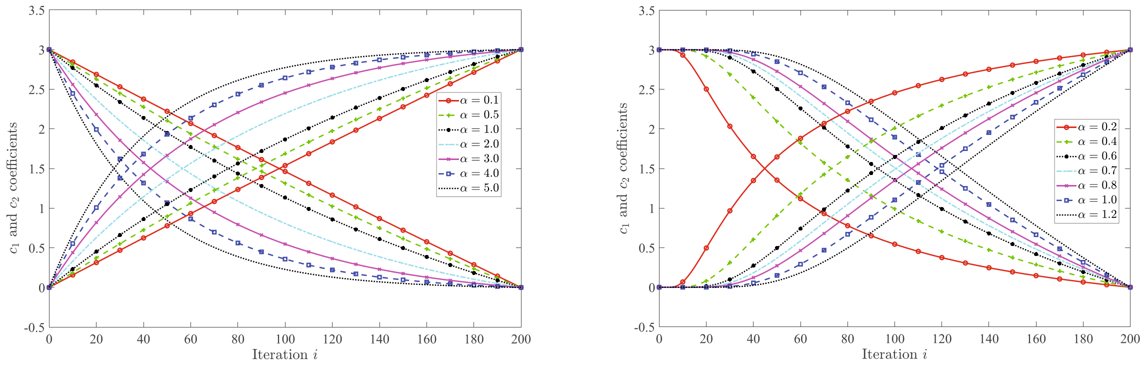
$$a_e = e^{\alpha} a_{ter} + a_s(1 - e^{\alpha})$$

$$a(t) = (a_e - a_s)e^{-\alpha t_{\max}/t} + a_s \quad \text{for Method D}$$

In these examples as well, the crossing strategy for $c_1(t)$ and $c_2(t)$ is to be symmetrical crossing search, and the search using the basic function of Method C was conducted at the interval of $t = 96(\alpha = 0.1)$ to $t = 28(\alpha = 5.0)$, as shown in Table 14. On the other hand, it searched for Method D the interval of $t = 45(\alpha = 0.2)$ to $t = 127(\alpha = 1.2)$ (see Table 15). It can be seen that t_{cross} in Method C varies inversely with the value of α, while t_{cross} in Method D varies in direct proportion to the value of α. Note, however, that Method C's t_{cross} can only handle the restricted crossing time region of $t_{cross} \in [0, \frac{1}{2}t_{\max}]$ for $\alpha \in [0, \infty]$, as will be shown in the following discussion. Note also that the original study in Lu et al. [50] did not use $e^{-\alpha t/t_{\max}}$ to make the acceleration coefficients, just only used it to make a time-varying weight. Figure 21 shows the acceleration coefficients used at this time.

Table 16 shows the research results for the case of $c_{1s} = 2.0, 2.5, 3.0, 3.5, 4.0$, $c_{1ter} = 0.0$, $c_{2s} = c_{1ter}$, $c_{2ter} = c_{1s}$. In Method C (using the basic function due to Lu et al. [50], when the decay rate α is $\alpha \rightarrow \infty$, the crossing time is $t_{cross} \rightarrow 0$ and $t_{cross} \rightarrow \frac{1}{2}t_{\max}$ if $\alpha \rightarrow 0$. Therefore, only the global convergence strategy before the mid crossing time can be taken. Looking at the crossing value c_{cross} around 1.5, most of them are less than $\alpha = 2.0$ (that is, after $t_{cross} = 57$) to obtain good global convergence values. Also, as can be seen from the value of the convergence rate, the number of times that a good global convergence value is obtained by the mid crossing time is very small.

On the other hand, it is found from Table 17 that, for Method D (using the basic function with the reciprocal of $t/t_{\max}$), the good global convergence values are obtained over the wide range of the decay rate from $\alpha = 0.4(t_{cross} = 75)$ to $\alpha = 1.2(t_{cross} = 127)$, at the crossing value $c_{cross} = 1.25$ or around 1.5 Therefore, it seems that convergence to the



(a) Use of a basic function by Method C

(b) Use of a basic function by Method D

Fig. 21 Time history of acceleration coefficients with different decay rates α

Table 16 Optimized results of Method C when using a symmetric crossing strategy, where $c_{1ter} = c_{2s} = 0.0$

	Decay rate α						
	0.1	0.5	1.0	2.0	3.0	4.0	5.0
(a) $c_{1s} = c_{2ter} = 2.0$, $c_{cross} = 1.0$							
Mean	**4687**	4731	4708	4713	4829	4756	4802
Std.	8.40	141	106	106	221	172	207
Conv. rate (%)	95	90	95	75	65	85	70
(b) $c_{1s} = c_{2ter} = 2.5$, $c_{cross} = 1.25$							
Mean	4726	4724	4726	**4677**	4701	4746	4744
Std.	141	142	151	2.58	102	169	169
Conv. rate (%)	90	90	90	100	95	85	85
(c) $c_{1s} = c_{2ter} = 3.0$, $c_{cross} = 1.5$							
Mean	4747	4788	**4676**	**4696**	4764	4764	4834
Std.	173	206	4.61	103	192	191	229
Conv. rate (%)	80	75	100	95	80	80	65
(d) $c_{1s} = c_{2ter} = 3.5$, $c_{cross} = 1.75$							
Mean	4744	4767	4731	**4674**	4770	4791	4767
Std.	143	180	147	6.43	195	208	193
Conv. rate (%)	55	75	90	100	80	75	80
(e) $c_{1s} = c_{2ter} = 4.0$, $c_{cross} = 2.0$							
Mean	4883	4838	4830	4818	4891	4872	4818
Std.	183	169	165	192	245	227	224
Conv. rate (%)	0	0	0	0	10	15	35

Table 17 Optimized results of Method D when using a symmetric crossing strategy, where $c_{1ter} = c_{2s} = 0.0$

	Decay rate α						
	0.2	0.4	0.6	0.7	0.8	1.0	1.2
(a) $c_{1s} = c_{2ter} = 2.0$, $c_{cross} = 1.0$							
Mean	4707	4757	4709	**4686**	4733	**4689**	4709
Std.	103	143	104	7.33	144	10.5	105
Conv. rate (%)	95	75	90	95	90	90	85
(b) $c_{1s} = c_{2ter} = 2.5$, $c_{cross} = 1.25$							
Mean	4861	4700	4724	4794	4701	4701	**4680**
Std.	233	102	141	205	102	102	8.37
Conv. rate (%)	60	95	85	75	95	95	95
(c) $c_{1s} = c_{2ter} = 3.0$, $c_{cross} = 1.5$							
Mean	**4693**	4718	**4673**	**4677**	4745	4724	4700
Std.	104	143	4.92	5.81	172	144	103
Conv. rate (%)	95	90	100	100	85	90	95
(d) $c_{1s} = c_{2ter} = 3.5$, $c_{cross} = 1.75$							
Mean	4767	**4678**	4736	**4689**	4716	4735	4740
Std.	195	7.72	152	14.2	106	112	107
Conv. rate (%)	80	100	70	85	70	30	25
(e) $c_{1s} = c_{2ter} = 4.0$, $c_{cross} = 2.0$							
Mean	4797	4833	4819	4838	4884	4944	4995
Std.	172	181	60.5	74.5	144	174	196
Conv. rate (%)	5	5	0	0	0	0	0

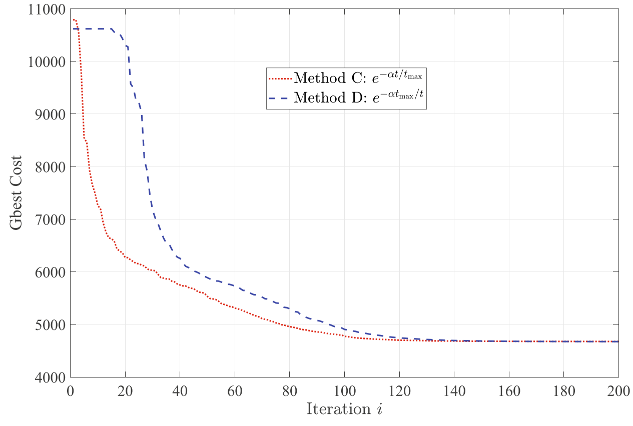


Fig. 22 Time history of gbest cost with basic functions due to Method C with α = 1:0 and Method D with α = 0:6, when using c_{1s} = 3:0; c_{1ter} = 0:0

global best value is more likely to be obtained by adjusting the decay rate than Method C (using the basis function due to Lu et al. [50]). Figure 22 shows a comparison of time histories in Gbest by Method C with $\alpha = 1.0$ and Method D with $\alpha = 0.6$ when $c_{1s} = 3.0$ and $c_{1ter} = 0.0$, and a diagram of each stability bound is shown in Fig. 23. These figures also confirm that Method C is an early search type, while Method D is a wide area search type with respect to time.

8.5 Using of basic functions with ill-posed initial boundary in asymmetric crossing strategy

In the symmetric search in which the basic functions had ill-posed initial values, as had been discussed already, the crossing time was determined uniquely, so that the time transferred to a global search was not controlled in search strategy, because their basic functions did not have any modulation index or decay rate parameter. Therefore, setting here that c_{1ini} and c_{2e} are not equal, or c_{2ini} and c_{1e} are not equal, the crossing time of $c_1(t)$ and $c_2(t)$ is moved back and forth from the one at the symmetry crossing strategy, and as a result, one can handle an asymmetric crossing strategy, in which the crossing value c_{cross} can be placed higher and down than the one at the symmetry crossing strategy. However, in such an asymmetric crossing strategy, t_{cross} and c_{cross} are respectively as follows:

$$\begin{aligned} t_{cross} &= t_{\max} - t_{\max}\{1 - \ln[\beta e + (1-\beta)]\} \\ c_{cross} &= (c_{1e} - c_{1ini})\beta + c_{1ini} \quad \text{for Method A} \end{aligned}$$

$$\begin{aligned} t_{cross} &= t_{\max} - \frac{t_{\max}}{1 - \ln(1-\beta)} \\ c_{cross} &= (c_{1e} - c_{1ini})\beta + c_{1ini} \quad \text{for Method B} \end{aligned}$$

where β is given by

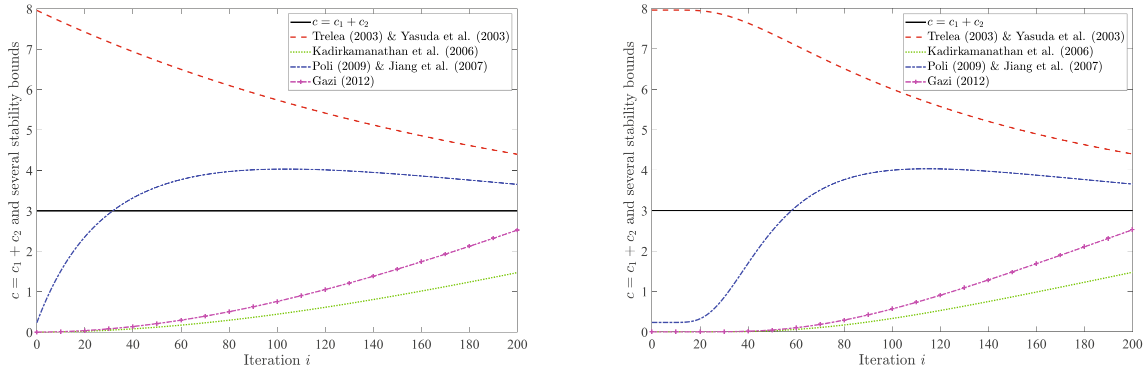
$$\beta = \frac{c_{12ini}}{c_{21e} + c_{12ini}}, \; c_{12ini} = c_{1ini} - c_{2ini}, \; c_{21e} = c_{2e} - c_{1e}$$

In what follows, for the nonlinear time-varying acceleration coefficients as used in Method A and Method B under a symmetric crossing strategy, the following three cases are considered:

Case 1: $c_{2ini} = c_{1e} = 0$
Case 2: $c_{2ini} \neq 0$, $c_{1e} = 0$
Case 3: $c_{2ini} = 0$, $c_{1e} \neq 0$

8.5.1 Case 1 results

It is found from Table 18 that good results are obtained for Method A, when c_{cross} is at $1.27 \sim 1.62$ (t_{cross} is at $98 \sim 153$). Note, however, that with this asymmetric crossing strategy, the same t_{cross} has a different c_{cross}, and the same c_{cross} has a different t_{cross}. On the other hand, it is confirmed from Table 19 that in Method B, the search tendency seen in the range of c_{cross} or t_{cross} where good results



(a) Bounds due to Method C with $\alpha = 1.0$

(b) Bounds due to Method D with $\alpha = 0.6$

Fig. 23 Stability bounds due to Methods C and D, when using $c_{1s} = 3.0$, $c_{1ter} = 0.0$

Table 18 Optimized results of using a basic function $e^{-(t_{\max}-t)/t_{\max}}$ (Method A), when using an asymmetric crossing strategy, where $c_{2ini} = c_{1e} = 0.0$

(a) A case of $c_{1ini} = 2.0$

	Terminal social accel. coefficient, c_{2e}				
	2.5	3.0	3.5	3.75	4.0
t_{cross}	114	105	98	94	91
c_{cross}	1.11	1.20	1.27	1.30	1.33
Mean	4749	4701	**4698**	4721	4821
Std.	168	102	104	145	223
Conv. rate (%)	85	95	95	90	70

(b) A case of $c_{1ini} = 2.5$

	Terminal social accel. coefficient, c_{2e}				
	2.0	3.0	3.5	3.75	4.0
t_{cross}	135	116	108	105	102
c_{cross}	1.11	1.36	1.46	1.50	1.54
Mean	4757	**4699**	4702	**4684**	4813
Std.	168	104	102	8.60	203
Conv. rate (%)	80	95	95	100	40

(c) A case of $c_{1ini} = 3.0$

	Terminal social accel. coefficient, c_{2e}				
	2.0	2.5	3.5	3.75	4.0
t_{cross}	142	133	117	114	111
c_{cross}	1.20	1.36	1.62	1.67	1.71
Mean	4709	**4684**	4759	4742	4776
Std.	101	15.1	169	115	118
Conv. rate (%)	90	90	70	30	15

(d) A case of $c_{1ini} = 3.5$

	Terminal social accel. coefficient, c_{2e}				
	2.0	2.5	3.0	3.75	4.0
t_{cross}	148	139	132	121	118
c_{cross}	1.27	1.46	1.62	1.81	1.87
Mean	4739	4706	**4683**	4794	4859
Std.	143	101	8.08	143	129
Conv. rate (%)	75	90	95	10	0

(e) A case of $c_{1ini} = 3.75$

	Terminal social accel. coefficient, c_{2e}				
	2.0	2.5	3.0	3.5	4.0
t_{cross}	151	142	135	128	122
c_{cross}	1.30	1.50	1.67	1.81	1.94
Mean	4711	**4682**	4716	4755	4897
Std.	102	8.30	104	117	163
Conv. rate (%)	95	100	75	15	0

(f) A case of $c_{1ini} = 4.0$

	Terminal social accel. coefficient, c_{2e}				
	2.0	2.5	3.0	3.5	3.75
t_{cross}	153	145	137	131	127
c_{cross}	1.33	1.54	1.71	1.87	1.94
Mean	**4688**	4728	4719	4764	4821
Std.	9.14	146	107	106	64.6
Conv. rate (%)	85	90	70	5	0

Table 19 Optimized results of using a basic function $e^{-t_{\max}/(t_{\max}-t)}$ (Method B), when using an asymmetric crossing strategy, where $c_{2ini} = c_{1e} = 0.0$

(a) A case of $c_{1ini} = 2.0$

	Terminal social accel. coefficient, c_{2e}				
	2.5	3.0	3.5	3.75	4.0
t_{cross}	75	68	63	60	58
c_{cross}	1.11	1.20	1.27	1.30	1.33
Mean	4770	**4698**	4718	**4696**	4772
Std.	190	105	145	106	197
Conv. rate (%)	80	95	90	95	80

(b) A case of $c_{1ini} = 2.5$

	Terminal social accel. coefficient, c_{2e}				
	2.0	3.0	3.5	3.75	4.0
t_{cross}	90	76	71	68	66
c_{cross}	1.11	1.36	1.46	1.50	1.54
Mean	4708	4812	4719	**4685**	4789
Std.	105	218	145	16.8	195
Conv. rate (%)	95	70	90	85	50

(c) A case of $c_{1ini} = 3.0$

	Terminal social accel. coefficient, c_{2e}				
	2.0	2.5	3.5	3.75	4.0
t_{cross}	96	89	77	75	72
c_{cross}	1.20	1.36	1.62	1.67	1.71
Mean	4732	**4674**	**4699**	**4685**	4738
Std.	140	3.78	105	8.13	105
Conv. rate (%)	80	100	95	95	0

(d) A case of $c_{1ini} = 3.5$

	Terminal social accel. coefficient, c_{2e}				
	2.0	2.5	3.0	3.75	4.0
t_{cross}	101	94	88	80	78
c_{cross}	1.27	1.46	1.62	1.81	1.87
Mean	4730	**4697**	4771	4754	4764
Std.	142	103	198	156	109
Conv. rate (%)	85	95	80	55	20

(e) A case of $c_{1ini} = 3.75$

	Terminal social accel. coefficient, c_{2e}				
	2.0	2.5	3.0	3.5	4.0
t_{cross}	103	96	90	85	80
c_{cross}	1.30	1.50	1.67	1.81	1.94
Mean	4727	**4676**	**4695**	4707	4849
Std.	142	5.32	103	115	207
Conv. rate (%)	90	100	95	95	5

(f) A case of $c_{1ini} = 4.0$

	Terminal social accel. coefficient, c_{2e}				
	2.0	2.5	3.0	3.5	3.75
t_{cross}	105	98	92	87	85
c_{cross}	1.33	1.54	1.71	1.87	1.94
Mean	**4684**	**4699**	**4673**	4715	4778
Std.	9.52	103	4.40	112	161
Conv. rate (%)	95	95	100	75	20

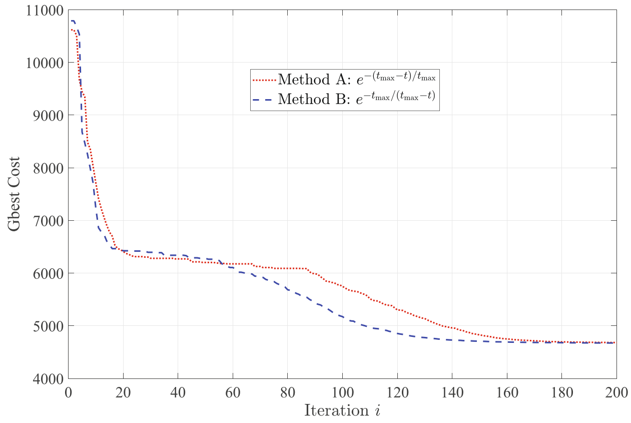


Fig. 24 Time history of gbest cost due to Methods A and B in an asymmetric crossing strategy, where $c_{1ini} = 3.75, c_{2e} = 2.5$ for Method A, and $c_{1ini} = 4.0$, $c_{2e} = 3.0$ for Method B, but $c_{2ini} = c_{1e} = 0$ for both cases

are obtained is roughly similar to Method A. Note, however, that the range of c_{cross} where better results are obtained with Method B is somewhat wider, i.e., c_{cross} is 1.2 $\sim$ 1.71 (or 60 $\sim$ 105 in t_{cross}). In particular, it can be recognized that the number of good results in the case of c_{cross} greater than or equal to 1.5 is much higher than Method A.

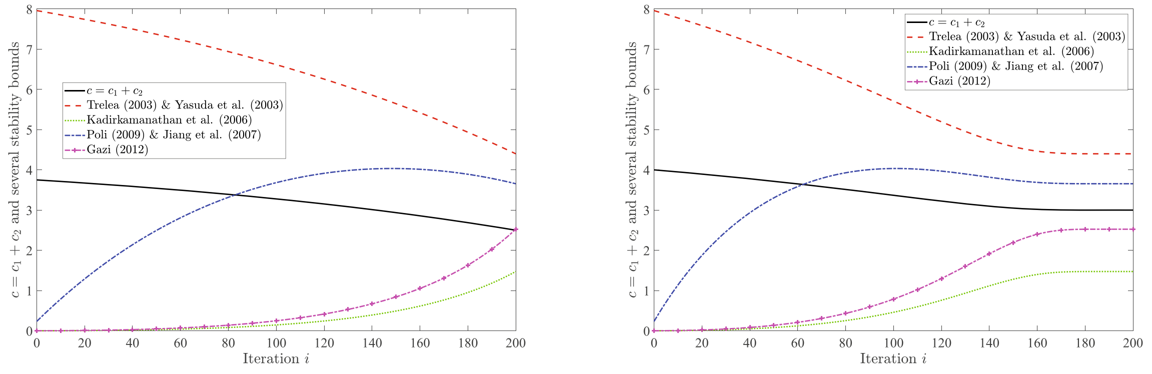
Figures 24 and 25 show a comparison of the Gbest time histories in Method A and Method B, and the stability bound diagrams of both methods. However, it shows the case of $c_{1ini} = 3.75$ and $c_{2e} = 2.5$ for Method A, whereas $c_{1ini} = 4.0$ and $c_{2e} = 3.0$ for Method B. As can be seen from these figures, in Method B, convergence to the global value starts around $t = 60$, whereas in Method A, the stability bound condition of the Order-2 stability criterion is not satisfied until after about $t = 80$, so that convergence to the global value begins about 20 generations later than Method B.

8.5.2 Case 2 results

As shown in Tables 20 and 21, the value of c_{cross} is higher overall when $c_{2ini} = 0.5 \neq 0.0$, resulting in a larger acceleration coefficient to the Gbest search in the exploitation process before the crossing time than when $c_{2ini} = 0.0$, and the results are not as good as those in Case 1. In fact, the Gbest evaluation for the 4600s is not obtained at all except for $c_{1ini} = 3.0, c_{2e} = 3.0$ in Method B. Figures 26 and 27 show a comparison of the time histories of Gbest for Methods A and B and stability bound diagrams for both methods. For Method A, the time histories are shown for $c_{1ini} = 2.75, c_{2e} = 2.5$, whereas for Method B, the time histories are shown for $c_{1ini} = 3.0, c_{2e} = 3.0$. As these figures show, Method B begins to converge to global values around $t = 50$, while Method A does not satisfy the stability bound condition of the Order-2 stability criteria until after about $t = 70$, which is about 20 generations later than Method B, convergence to the global value begins about 20 generations later than Method B.

8.5.3 Case 3 results

As can be seen from Table 22, more good results are obtained for Method A than for Case 1 when c_{cross} is relatively high, due to the effect of $c_{1e} = 0.5$. On the other hand, for Method B, the number of good results for relatively high c_{cross} is not much different from that in Case 1, but the number of good results for c_{cross} less than 1.50 is about half of that in Case 1, as shown in Table 23. Figures 28 and 29 show a comparison of the time histories of Gbest for methods A and B, and the stability bound diagrams for both methods, respectively, where for Method A, the case of $c_{1ini} = 2.0, c_{2e} = 3.0$ is shown, whereas for Method B, the case of $c_{1ini} = 3.5, c_{2e} = 2.75$ is shown. In the stability bound comparison diagram in Fig. 29, unlike the previous two comparisons, the stability bound con-



(a) Bounds due to Method A with $c_{1ini} = 3.75, c_{2e} = 2.5$

(b) Bounds due to Method B with $c_{1ini} = 4.0, c_{2e} = 3.0$

Fig. 25 Stability bounds due to Methods A and B in an asymmetric crossing strategy, when using $c_{2ini} = c_{1e} = 0.0$

Table 20 Optimized results of using a basic function $e^{-(t_{\max}-t)/t_{\max}}$ (Method A), when using an asymmetric crossing strategy, where $c_{2ini} = 0.5, c_{1e} = 0.0$

	Terminal social accel. coefficient, c_{2e}					
	2.0	2.5	3.0	3.5	3.75	4.0
(a) A case of $c_{1ini} = 2.5$						
t_{cross}	125	114	105	98	94	91
c_{cross}	1.25	1.39	1.50	1.59	1.63	1.67
Mean	4753	4768	4747	4768	4904	4912
Std.	170	189	172	175	244	243
Conv. rate (%)	80	80	85	55	20	5
(b) A case of $c_{1ini} = 2.75$						
t_{cross}	130	120	111	103	100	97
c_{cross}	1.29	1.45	1.57	1.67	1.72	1.76
Mean	4798	4725	4798	4794	4855	4913
Std.	204	146	208	190	217	228
Conv. rate (%)	70	90	75	35	10	0
(c) A case of $c_{1ini} = 3.0$						
t_{cross}	135	125	116	108	105	102
c_{cross}	1.33	1.50	1.64	1.75	1.80	1.85
Mean	4734	4771	4737	4876	4797	4929
Std.	149	191	151	225	115	243
Conv. rate (%)	85	80	85	10	0	0
(d) A case of $c_{1ini} = 3.5$						
t_{cross}	142	133	125	117	114	111
c_{cross}	1.40	1.59	1.75	1.88	1.94	2.00
Mean	4734	4733	4762	4872	4850	4918
Std.	149	138	171	226	165	198
Conv. rate (%)	95	75	60	10	0	0

Table 21 Optimized results of using a basic function $e^{-t_{\max}/(t_{\max}-t)}$ (Method B), when using an asymmetric crossing strategy, where $c_{2ini} = 0.5, c_{1e} = 0.0$

	Terminal social accel. coefficient, c_{2e}					
	2.0	2.5	3.0	3.5	3.75	4.0
(a) A case of $c_{1ini} = 2.5$						
t_{cross}	82	75	68	63	60	58
c_{cross}	1.25	1.39	1.50	1.59	1.63	1.67
Mean	4728	4721	4717	4798	4838	4938
Std.	142	143	143	211	237	315
Conv. rate (%)	90	90	90	70	65	15
(b) A case of $c_{1ini} = 2.75$						
t_{cross}	86	79	72	67	64	62
c_{cross}	1.29	1.45	1.57	1.67	1.72	1.76
Mean	4799	4768	4789	4753	4818	4782
Std.	203	193	209	175	213	178
Conv. rate (%)	70	80	75	80	40	20
(c) A casae of $c_{1ini} = 3.0$						
t_{cross}	90	82	76	71	68	66
c_{cross}	1.33	1.50	1.64	1.75	1.80	1.85
Mean	4748	4835	**4673**	4804	4821	4922
Std.	169	226	7.25	213	231	278
Conv. rate (%)	85	65	100	65	65	10
(d) A case of $c_{1ini} = 3.5$						
t_{cross}	96	89	82	77	75	72
c_{cross}	1.40	1.59	1.75	1.88	1.94	2.00
Mean	4799	4765	4791	4833	4803	4865
Std.	203	190	207	230	263	179
Conv. rate (%)	75	80	75	65	25	0

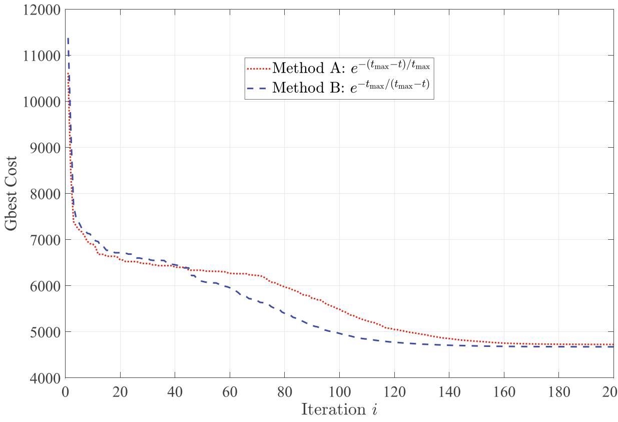


Fig. 26 Time history of gbest cost due to Methods A and B in an asymmetric crossing strategy, where $c_{1ini} = 2.75, c_{2e} = 2.5$ for Method A, and $c_{1ini} = 3.0, c_{2e} = 3.0$ for Method B, but $c_{2ini} = 0.5 \neq 0, c_{1e} = 0$ for both cases

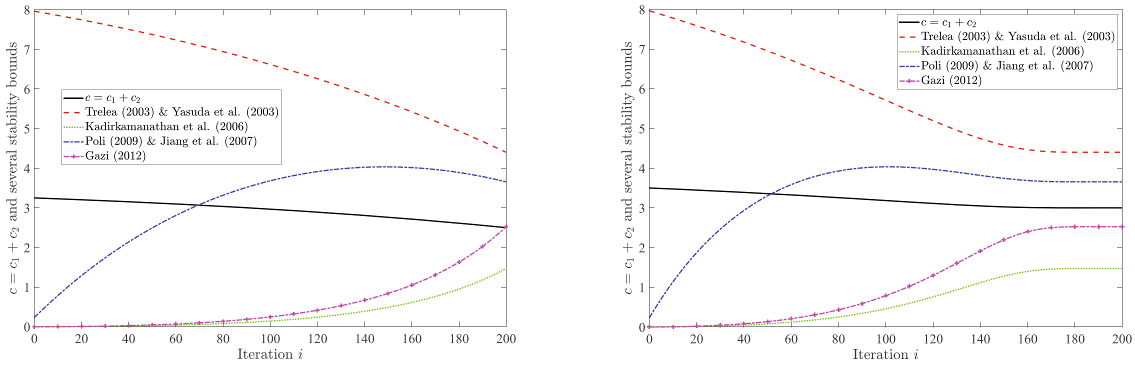
dition for the Order-2 stability criteria begins to be satisfied at around $t = 40$ for Method A, while the same stability bound condition is not satisfied until after about $t = 50$ for Method B. It can be seen from Fig. 28 that the actual Gbest values of Method B approach those of Method A in the neighborhood of 40 generations, but only after 80 generations, about 40 generations later than those of Method A.

8.6 Time cost experiments and ablation studies

In this section, we conduct experiments to compare the time cost of the examples used to compare the time histories of the Gbest cost handled in Section 8.2 through Section 8.5. We also perform so-called ablation studies in which the time-varying acceleration coefficients are not used and the cross-strategy is removed.

Table 22 Optimized results of using a basic function $e^{-(t_{\max}-t)/t_{\max}}$ (Method A), when using an asymmetric crossing strategy, where $c_{2ini} = 0.0, c_{1e} = 0.5$

	Terminal value, c_{2e}			
	2.5	2.75	3.0	3.5
(a) A case of $c_{1ini} = 2.0$				
t_{cross}	125	119	114	105
c_{cross}	1.25	1.29	1.33	1.40
Mean	**4680**	4701	**4680**	4702
Std.	6.97	103	4.76	103
Conv. rate (%)	100	95	100	95
(b) A case of $c_{1ini} = 2.5$				
t_{cross}	135	129	125	116
c_{cross}	1.39	1.45	1.50	1.59
Mean	4725	4702	4772	4721
Std.	145	104	191	119
Conv. rate (%)	90	95	80	70
(c) A case of $c_{1ini} = 3.0$				
t_{cross}	142	137	133	125
c_{cross}	1.50	1.57	1.64	1.75
Mean	4702	**4684**	4712	4771
Std.	104	12.4	104	102
Conv. rate (%)	90	95	80	0
(d) A case of $c_{1ini} = 3.5$				
t_{cross}	148	144	139	132
c_{cross}	1.59	1.67	1.75	1.88
Mean	**4685**	4719	4734	4852
Std.	8.85	99.1	108	128
Conv. rate (%)	90	65	25	0
(e) A case of $c_{1ini} = 3.75$				
t_{cross}	151	146	142	135
c_{cross}	1.63	1.72	1.80	1.94
Mean	**4692**	4724	4726	4875
Std.	16.8	103	37.3	121
Conv. rate (%)	65	50	35	0
(f) A case of $c_{1ini} = 4.0$				
t_{cross}	153	149	145	137
c_{cross}	1.67	1.76	1.85	2.0
Mean	**4691**	4726	4762	4928
Std.	12.0	42.1	49.2	185
Conv. rate (%)	80	25	10	0



(a) Bounds due to Method A with $c_{1ini} = 2.75, c_{2e} = 2.5$

(b) Bounds due to Method B with $c_{1ini} = 3.0, c_{2e} = 3.0$

Fig. 27 Stability bounds due to Methods A and B in an asymmetric crossing strategy, when using $c_{2ini} = 0.5 \neq 0, c_{1e} = 0.0$

Table 23 Optimized results of using a basic function $e^{-t_{\max}/(t_{\max}-t)}$ (Method B), when using an asymmetric crossing strategy, where $c_{2ini} = 0.0$, $c_{1e} = 0.5$

	Terminal value, c_{2e}			
	2.5	2.75	3.0	3.5
(a) A case of $c_{1ini} = 2.0$				
t_{cross}	82	78	75	68
c_{cross}	1.25	1.29	1.33	1.40
Mean	4768	4767	4742	**4697**
Std.	190	191	172	104
Conv. rate (%)	80	80	85	95
(b) A case of $c_{1ini} = 2.5$				
t_{cross}	90	86	82	76
c_{cross}	1.39	1.45	1.50	1.59
Mean	**4697**	4720	**4698**	4763
Std.	104	141	105	181
Conv. rate (%)	95	90	95	75
(c) A case of $c_{1ini} = 3.0$				
t_{cross}	96	92	89	82
c_{cross}	1.50	1.57	1.64	1.75
Mean	**4698**	4720	4727	4715
Std.	103	143	149	27.2
Conv. rate (%)	95	90	85	40
(d) A case of $c_{1ini} = 3.5$				
t_{cross}	101	97	94	88
c_{cross}	1.59	1.67	1.75	1.88
Mean	**4699**	**4673**	**4676**	4777
Std.	105	5.87	5.08	118
Conv. rate (%)	95	100	100	10
(e) A case of $c_{1ini} = 3.75$				
t_{cross}	103	100	96	90
c_{cross}	1.63	1.72	1.80	1.94
Mean	4721	4721	4712	4819
Std.	143	143	109	149
Conv. rate (%)	90	90	85	0
(f) A case of $c_{1ini} = 4.0$				
t_{cross}	105	102	98	92
c_{cross}	1.67	1.76	1.85	2.0
Mean	**4675**	4721	4711	4794
Std.	5.52	144	103	114
Conv. rate (%)	100	90	85	0

8.6.1 Comparison of time cost

First, Table 24 shows the time required for 20 trials and the average time per trial for each crossing strategy that incorporates the six basic functions discussed in Section 8.2 through Section 8.5. Note that the design parameters selected are those used in the previous section for the Gbest cost time history comparison. The counted time is the elapsed time when the Gbest cost is displayed at each iteration time. On the other hand, Table 25 shows similar results for the asymmetric crossing strategy for methods A and B. As can be seen from these tables, the time elapsed for roughly 20 trials ranges from 715 to 724 s, with an average time per trial of about 36 s. There is no significant difference in time cost between the methods.

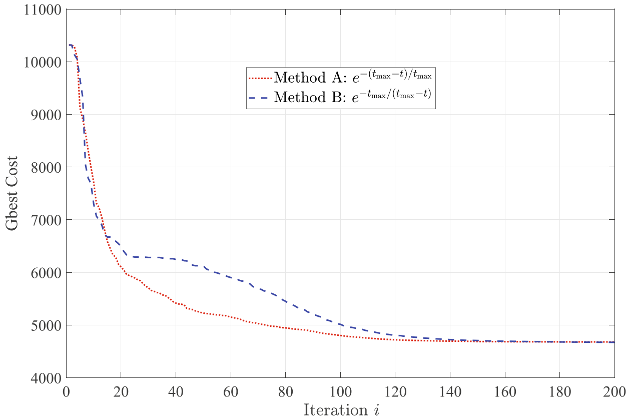


Fig. 28 Time history of gbest cost due to Methods A and B in an asymmetric crossing strategy, where $c_{1ini} = 2.0$, $c_{2e} = 3.0$ for Method A, and $c_{1ini} = 3.5$, $c_{2e} = 2.75$ for Method B, but $c_{2ini} = 0$, $c_{1e} = 0.5 \neq 0$ for both cases

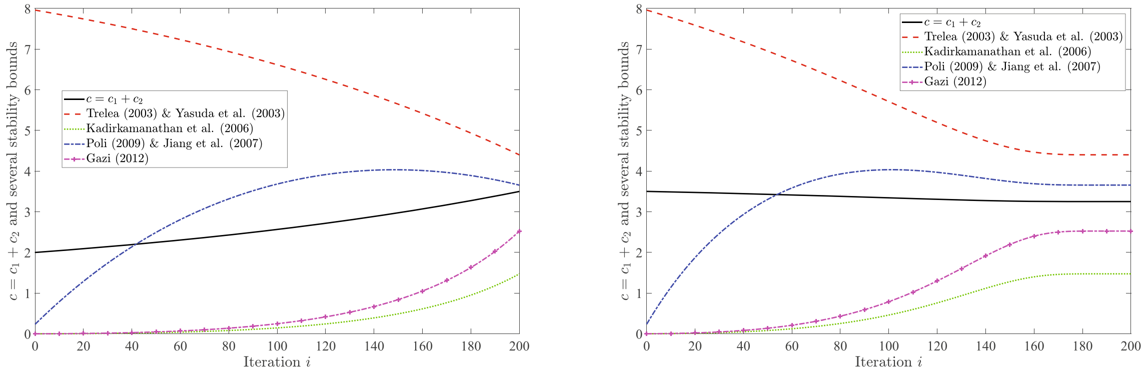
8.6.2 Ablation studies

In this ablation study, we investigate the search performance and computational cost of five PSO variants that originally utilize the different basic functions focused on in Section 8.2 through Section 8.5, as shown in Table 26. Ablation conditions for each PSO variant are described below.

First, in the PSO variant of Chatterjee and Siarry [11], the inertia weight was defined as $w(t) = (w_s - w_e)(1 - t/t_{\max})^n + w_e$ using a monotonically decreasing function with the basic function of $g(t, t_{\max}) = (1 - t/t_{\max})^n$. In this paper, $w_s = 0.6$, $w_e = -0.4$ and the recommended value of n is $n = 1.2$, while the acceleration coefficients are fixed values with no crossing, $c_1 = c_2 = 2.0$ or $c_1 = c_2 = 1.5$.

On the other hand, the PSO variant in the original paper by Wang and Liu [86] used a monotonically increasing basic function, $f(t, t_{\max}) = (t/t_{\max})^n$ for all $w(t)$, $c_1(t)$ and $c_2(t)$, but $w_s = 0.9$, $w_e = 0.4$, $c_{1s} = 2.5$, $c_{1e} = 0.5$. However, it was assumed that $n = 1/3, 1/2, 1, 2, 3$ can be taken. They finally concluded that $n = 1/3$ for $c_1(t)$ and $n = 1$ for $c_2(t)$, in other words, cross-realization before the intermediate iteration time. However, $w(t)$ always dealt only with $n = 1$. Therefore, for comparison objectives, we fix $n = 1$ for $w(t)$ and $c_2(t)$, and $n = 1/3$ for $c_1(t)$.

The mixed type of these two basic functions at $n = 1$ is the PSO variant of Ratnaweera et al. [65]. That is, $w(t)$ with $g(t, t_{\max}) = 1 - t/t_{\max}$ is linearly monotonically decreasing time-varying weight with $w_s = 0.9$ and $w_e = 0.4$, while $c_1(t)$ and $c_2(t)$ have $f(t, t_{\max}) = t/t_{\max}$ as a basic function, and they are linearly monotonically decreasing and monotonically increasing time-varying coefficients, respectively, in which the acceleration coefficients intersect at exactly $0.5t_{\max}$. However, $c_{1s} = 2.5$ and $c_{1e} = 0.5$ are used for comparison.



(a) Bounds due to Method A with $c_{1ini} = 2.0, c_{2e} = 3.0$

(b) Bounds due to Method B with $c_{1ini} = 3.5, c_{2e} = 2.75$

Fig. 29 Stability bounds due to Methods A and B in an asymmetric crossing strategy, when using $c_{2ini} = 0, c_{1e} = 0.5 \neq 0.0$

Table 24 Time cost comparisons among methods when using a symmetric crossing strategy

Referred paper or method's name	Type of BF	Parameters	Time cost [s]	
			20 trials	Mean
Chatterjee and Siarry [11]	$g(t, t_{\max}) = (1 - t/t_{\max})^n$	$n = 1.25$ $c_{1s} = 3.0, c_{1e} = 0.0$	715.60	35.78
Wang and Liu [86]	$f(t, t_{\max}) = (t/t_{\max})^n$	$n = 0.75$ $c_{1s} = 3.0, c_{1e} = 0.0$	724.14	36.21
Method A	$f(t, t_{\max}) = e^{-(t_{\max}-t)/t_{\max}}$	$c_{1ini} = 2.5,$ $c_{2ini} = 0.0$	720.36	36.02
Method B or Wu et al. [89]	$g(t, t_{\max}) = e^{-t_{\max}/(t_{\max}-t)}$	$c_{1ini} = 3.5,$ $c_{2ini} = 0.0$	718.19	35.91
Method C or Lu et al. [50]	$g(t, t_{\max}) = e^{-\alpha t/t_{\max}}$	$\alpha = 1.0$ $c_{1s} = 3.0, c_{1ter} = 0.0$	721.54	36.08
Method D	$f(t, t_{\max}) = e^{-\alpha t_{\max}/t}$	$\alpha = 0.6$ $c_{1s} = 3.0, c_{1ter} = 0.0$	723.79	36.19

Table 25 Time cost comparisons between methods A and B when using an asymmetric crossing strategy

Referred paper or method's name	Type of BF	Parameters	Time cost [s]	
			20 trials	Mean
Method A	$f(t, t_{\max}) = e^{-(t_{\max}-t)/t_{\max}}$	$c_{1ini} = 3.75, c_{2e} = 2.5$ $c_{2ini} = c_{1e} = 0.0$	717.07	35.85
Method B or Wu et al. [89]	$g(t, t_{\max}) = e^{-t_{\max}/(t_{\max}-t)}$	$c_{1ini} = 4.0, c_{2e} = 3.0$ $c_{2ini} = c_{1e} = 0.0$	716.61	35.83
Method A	$f(t, t_{\max}) = e^{-(t_{\max}-t)/t_{\max}}$	$c_{1ini} = 2.75, c_{2e} = 2.5$ $c_{2ini} = 0.5 \neq 0.0, c_{1e} = 0.0$	722.28	36.11
Method B or Wu et al. [89]	$g(t, t_{\max}) = e^{-t_{\max}/(t_{\max}-t)}$	$c_{1ini} = 3.0, c_{2e} = 3.0$ $c_{2ini} = 0.5 \neq 0.0, c_{1e} = 0.0$	717.17	35.86
Method A	$f(t, t_{\max}) = e^{-(t_{\max}-t)/t_{\max}}$	$c_{1ini} = 2.0, c_{2e} = 3.0$ $c_{2ini} = 0.0, c_{1e} = 0.5 \neq 0.0$	724.25	36.21
Method B or Wu et al. [89]	$g(t, t_{\max}) = e^{-t_{\max}/(t_{\max}-t)}$	$c_{1ini} = 3.5, c_{2e} = 2.75$ $c_{2ini} = 0.0, c_{1e} = 0.5 \neq 0.0$	721.74	36.09

Table 26 Ablation comparisons, where in those of Chatterjee and Siarry [11], Lu et al. [50], and Phung and Ha [58], the time-varying function of $c_1(t), c_2(t)$ is removed, while in Wang and Liu [86] the modulation index n is $n = 1$ for $w(t)$ and $c_2(t)$, and it is set to $n = 1$ (i.e., linear) for all $w(t)$, $c_1(t)$ and $c_2(t)$ in Ratnaweera et al. [65]

PSO variants	Gbest cost		Conv.	Time cost [s]		Remarks
	Mean	Std.	rate [%]	20 trials	Mean	
Chatterjee and Siarry [11] ($c_1 = c_2 = 2.0$)	5073.1	272.2	5	710.04	35.50	Time-varying only for $w(t)$ with $g(t, t_{\max}) = (1 - t/t_{\max})^n$, $n = 1.2$, but with $c_1 = c_2 = 2.0$
Chatterjee and Siarry [11] ($c_1 = c_2 = 1.5$)	4956.8	258.4	35	717.24	35.86	Time-varying only for $w(t)$ with $g(t, t_{\max}) = (1 - t/t_{\max})^n$, $n = 1.2$, but with $c_1 = c_2 = 1.5$
Wang and Liu [86]	4838.4	226.8	65	720.79	36.04	Time-varying for all $w(t)$, $c_1(t)$ and $c_2(t)$ with $f(t, t_{\max}) = (t/t_{\max})^n$, but $n = 1$ for $w(t)$ and $c_2(t)$, whereas $n = 1/3$ for $c_1(t)$
Ratnaweera et al. [65]	4845.3	229.9	60	713.39	35.67	$g(t, t_{\max}) = 1 - t/t_{\max}$ for $w(t)$, $f(t, t_{\max}) = t/t_{\max}$ for $c_1(t), c_2(t)$
Lu et al. [50]	4976.5	251.0	25	708.88	35.44	Time-varying only for $w(t)$ with $g(t, t_{\max}) = e^{-\alpha t/t_{\max}}$, $\alpha = 7.0$, but with $c_1 = c_2 = 2.0$
Phung and Ha [58]	4874.6	232.6	50	721.00	36.05	Time-varying only for $w(t)$, i.e., $w(t) = 0.98w(t-1)$, but with $c_1 = c_2 = 1.5$

In the PSO variant of Lu et al. [50], only the inertia weight $w(t)$ is constructed using a monotonically decreasing basic function, $g(t, t_{\max}) = e^{-\alpha t/t_{\max}}$. However, $w_s = 0.9$, $w_e = 0.2$ and $\alpha \in [6, 8]$ were recommended, so that $\alpha = 7.0$ is used for the comparison here, as well as setting $c_1(t) = c_2(t) = 2$.

The PSO variant of Phung and Ha [58] also has fixed acceleration coefficients, but only the inertia weight was time-varying. However, the condition used in the original 3D path planning is that $w(t)$ is in a form that incorporates the damping coefficient, i.e., $w(t) = w_{damp}w(t-1)$, $w_{damp} = 0.98$, $w(0) = 1.0$, and the acceleration coefficients are $c_1(t) = c_2(t) = 1.5$.

These five PSO variants can be broadly divided into cases where the time-varying function of the acceleration coefficients is removed and consequently no crossing strategy is used, and cases where the basic function characteristics are partially linearly restricted. As can be seen from Table 26, the ablation-type PSO variants are clearly computationally less expensive, with a time of 709 to 721 s for 20 trials and an average time of about 35 to 36 s per trial. However, as can be seen from Tables 11, 12, 16, and 26, even with an increase in computational cost of about one second per trial, the search performance obtained by the proposed PSO, which incorporates a crossing strategy with nonlinear time-varying acceleration coefficients, shows significant improvement in terms of average Gbest cost and convergence rate.

8.7 How do we choose the basis functions in the sequel? And how do different basis functions affect search performance?

It is easier to obtain crossing times and crossing values on the crossing strategy if the basic functions are well-posed. In this sense, it is easy to use those of Chatterjee and Siarry [11] and Wang and Liu [86], where the crossing time and crossing value can be freely changed with the modulation parameter n. However, even when using ill-posed Napier numbers or trigonometric functions as basic functions, the crossing time and crossing value can be adjusted by correcting the acceleration coefficient at the beginning or end of the search with the fictitious boundary value and changing the decay rate α. Therefore, there is no problem in terms of crossing strategy, but there is some algorithmic complexity.

Although the effect of differences in basis functions on search performance cannot be discussed in general terms, a good convergence region can be predicted in advance from the pattern of crossing times t_{cross} for different modulation indices of acceleration coefficients and decay rates. For example, the basic functions of Chatterjee and Siarry [11] and Wang and Liu [86] are originally linear functions $1 - t/t_{\max}$ and $t/t_{\max}$ in their components. Figure 14 shows that the pattern is distributed around $\frac{1}{2}t_{\max}$ even when the modulation index n changes. The densely distributed crossing time patterns indicate the formation of the first half-dependent crossing pattern before $\frac{1}{2}t_{\max}$ and the second half-dependent crossing pattern after $\frac{1}{2}t_{\max}$, respectively, those of which are also supported by the search results in Tables 11 and 12, which include the convergence rate.

Similarly, Fig. 21 shows that the symmetric crossing patterns of the acceleration coefficients of Methods C and D, which use Napier numbers with decay rate α as basic functions, are dense around $t_{cross} = 30$ and 90 for the former and around $t_{cross} = 110$ for the latter. The former is found in $\alpha = 2.0$ ($t_{cross} = 59$) from Table 16, and the latter is found in $\alpha = 0.2 \sim 1.2$ ($t_{cross} = 45 \sim 127$) from Table 17, showing good results in a wide range of crossing times.

On the other hand, when the basic functions do not include modulation index n or decay rate α, improved search performance can be expected by adopting asymmetric crossing, but one key is how to design the asymmetry of the two acceleration coefficients. Here, c_{1ini} and c_{2e} were selected and adjusted experimentally by looking at the values of t_{cross} and c_{cross}. In the future, there is room for more in-depth research on asymmetric crossing strategies, including more systematic selection and adjustment methods.

8.8 Limitations of the proposed method

Finally, to demonstrate the limitations of the proposed method, we compare its advantages with its disadvantages and constraints.

8.8.1 Necessity of prior information

When selecting basic functions (such as trigonometric functions or the Napier's constant), it is essential to check whether the chosen function is continuous or contains discontinuities and non-differentiable points. Additionally, we need to verify whether the initial and terminal values are well-posed (i.e., lie within the range of $(0, 1)$ or $(1, 0)$). If ill-posed, determining fictitious initial or terminal values becomes necessary.

8.8.2 Choice of crossing method

Depending on the complexity of the search problem (e.g., unimodal or multimodal), different crossing methods should be considered. For simpler problems, a linear crossing method may suffice. However, for complex problems or cases where complexity is unknown, nonlinear symmetric crossing methods that allow adjustments between local and global exploration are recommended. Asymmetric crossing methods should be chosen when basic functions do not include a modulation parameter or decay rate.

8.8.3 Computational cost and complexity

In general, adjusting crossing values and timing in the proposed method allows for efficient global solution discovery compared to conventional methods with fixed acceleration coefficients. Implementation complexity is minimal, involving only setting fictitious initial or terminal values and adding one or two lines for time-varying acceleration coefficients. Although the computational cost increases by approximately one second per trial compared to conventional methods (as found in Section 8.6), significant improvements in evaluation function and convergence rate justify this cost.

8.8.4 Applicability to constraint conditions

We believe that the proposed method can be applied to various engineering problems, similar to conventional PSO variants. For instance, applying NCS-PSO to the design of flexible hinge displacement amplification mechanisms (as proposed in [33]) could lead to improved machine performance and efficiency by finding optimal magnification values. Additionally, hybridizing the “Taguchi" method with NCS-PSO for optimizing process parameters in electrical discharge machining (EDM) (as explored in [82]) may yield better results. However, it is crucial to recognize that, unlike other evolutionary computations (EC) or genetic algorithms (GA), PSO generally does not employ “arbitrary particle-to-particle crossovers" or “mutation operations," which means there is always a risk of converging to local solutions in non-convex problems.

9 Conclusions

In this paper, we have introduced a unified methodology for designing nonlinear time-varying inertia weight and acceleration coefficients, resulting in the development of Nonlinear Crossing Strategy-based Particle Swarm Optimizations (NCS-PSOs) for global optimization. We started by explaining the process of designing monotonically decreasing time-varying parameters, ensuring they meet specified initial and terminal boundary values by using and amplifying well-posed monotonic functions. Furthermore, we proposed a method to address ill-posed basic functions by introducing fictitious boundary values when extending this concept. Then, our exploration of the crossing strategy method, wherein acceleration coefficients are intersected during the PSO search, revealed novel insights into optimizing the search process. The introduction of crossing values and times as new information during the search demonstrated promising avenues for enhancing optimization outcomes. Comparative evaluations against conventional PSO variants and contemporary metaheuristics, using the CEC2014 benchmark functions, showcased the competitive performance of our proposed NCS-PSOs.

While our study predominantly focused on the integration of both cognitive and social acceleration terms, with their balance dynamically shifting throughout the search process, we recognize the potential for further refinement. Inspired by previous works, such as Kennedy's cognition-only and social-only models [39], and Duan's exploration of switching between these models [18], future research directions could involve the development of new search algorithms encompassing such variations [19]. Additionally, extending our investigation to encompass periodic and multi-point crossing methods to handle time intersections at multiple points presents intriguing avenues for future exploration. Finally, we acknowledge the importance of convergence analysis in ensuring the efficacy of optimization algorithms. Although not extensively covered in this paper, the work of Clerc and Kennedy [15] serves as a foundational reference in this domain, and future endeavors could delve deeper into convergence analysis to provide robust guarantees of convergence to global solutions. In summary, our contributions pave the way for more sophisticated and effective PSO optimization techniques, with potential applications spanning various domains, including but not limited to 3D path planning for drones.

Appendix

A Stability checks

When performing a PSO search using a specific crossing strategy with a time-varying inertial weight $w(t)$ and time-varying acceleration coefficients $c_1(t)$ and $c_2(t)$, some stability analysis methods or evaluation costs for PSO that have been derived are here introduced to check if the stability bounds are satisfied over the iteration interval.

A.1 Condition of Trelea and Yasuda et al.

This analysis method is due to Trelea [83] and Yasuda et al. [95] and is given as follows:

$$\bar{c} < 2(w+1) \tag{107}$$

where w is $|w| < 1$ and $\bar{c} = (c_1 + c_2)/2 > 0$. This stability is also called "Order-1 stability."

A.2 Condition of Kadirkamanathan et al.

The result of this analysis is given by Kadirkamanathan et al. [37] as follows:

$$c < \frac{2(1-2|w|+w^2)}{1+w} = \frac{2(1-|w|)^2}{1+w} \tag{108}$$

where assume that $|w| < 1$, $w \neq 0$, and $c \triangleq c_1 + c_2$. This original analysis uses a deterministic Lyapunov function and deals with the Rur'e stabilization problem.

A.3 Condition of Poli and Jiang et al.

The result of this analysis is given below from Poli [59, 60] or Jiang et al. [35].

$$c < \frac{24(1-w^2)}{7-5w} \tag{109}$$

where let $|w| < 1$ and $c \triangleq c_1 + c_2$. This stability, also called "Order-2 stability," was analyzed by stochastic process theory in the original analysis.

A.4 Gazi's condition

The result of this analysis is given by Gazi [25] as follows:

$$\bar{c} < \frac{12(1-2|w|+w^2)}{7(1+w)} = \frac{12(1-|w|)^2}{7(1+w)} \tag{110}$$

where assume that $\bar{c} \triangleq (c_1 + c_2)/2 > 0$ and $|w| < 1$. This original analysis uses a stochastic Lyapunov method to discuss the positive realness of absolute stability based on Typkin's result.

Author Contributions Keigo Watanabe: Conceptualization, Methodology, Software, Writing-original draft preparation, review and editing. Xiongshi Xu: Investigation, Software, Validation, Writing-review and editing.

Funding Open Access funding provided by Okayama University.

Data Availability The authors declare that the data supporting the findings of this study are available within the paper.

Declarations

Competing interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethical and informed consent for data used This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Abd Latiff I, Tokhi MO (2009) Fast convergence strategy for particle swarm optimization using spread factor. In: Proceeding of 2009 IEEE congress on evolutionary computation (CEC 2009), pp 2693–2700, https://doi.org/10.1109/CEC.2009.4983280
2. Alvarez-Alvarado MS, Rengifo J, Gallegos-Núñez RM et al (2022) Particle swarm optimization for optimal frequency response with high penetration of photovoltaic and wind generation. Energies 15:8565. https://doi.org/10.3390/en15228565
3. Askarzadeh A (2016) A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. Comput Struct 169:1–12. https://doi.org/10.1016/j.compstruc.2016.03.001
4. Awad NH, Ali MZ, Liang JJ, et al (2016) Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective real-parameter numerical optimization. Tech. rep., Nanyang Technological University, Jordan University of Science and Technology, Zhengzhou University, https://github.com/P-N-Suganthan/CEC2017-BoundContrained
5. Awla HQ, Kareem SW, Mohammed AS (2023) A comparative evaluation of bayesian networks structure learning using falcon optimization algorithm. Int J Interact Multimed Artif Intell 8(2):81–87. https://doi.org/10.9781/ijimai.2023.01.004
6. Basavanna M, Shivakumar M (2019) An overview of path planning and obstacle avoidance algorithms in mobile robots. Int J Eng Technol 8(12):478–482. https://doi.org/10.17577/IJERTV8IS120252
7. Ben Khoud K, Bouallègue S, Ayadi M (2018) Design and co-simulation of a fuzzy gain-scheduled pid controller based on particle swarm optimization algorithms for a quad tilt wing unmanned aerial vehicle. Trans Inst Meas Control 40(14):3933–3952. https://doi.org/10.1177/0142331217740947
8. Caraffini F, Iacca G (2020) The sos platform: designing, tuning and statistically benchmarking optimization algorithms. Mathematics 8(5):785. https://doi.org/10.3390/math8050785
9. Carstensen S, Lin JCW (2024) Tku-pso: an efficient particle swarm optimization model for top-k high-utility itemset mining. Int J Interact Multimed Artif Intell. https://doi.org/10.9781/ijimai.2024.01.002
10. Chai WS, bin Romli MIF, Yaakob SB et al (2022) Regenerative braking optimization using particle swarm algorithm for electric vehicle. J Adv Comput Intell Intell Inform 26(6):1022–1030. https://doi.org/10.20965/jaciii.2022.p1022
11. Chatterjee A, Siarry P (2006) Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. Comput Oper Res 33(3):859–871. https://doi.org/10.1016/j.cor.2004.08.012
12. Chen G, Jia J (2006) Han Q (2006 (in Chinese)) Study on the strategy of decreasing inertia weight in particle swarm optimization algorithm. J Xi'an Jiaotong Univ 40(1):53–56. 0253-987X, 01-0053-04
13. Chen S, Cai G, Guo W et al (2007) (in Chinese)) Study on the nonlinear strategy of acceleration coefficient in particle swarm optimization (pso) algorithm. J of Yangtze University Sci & Eng (Nat Sci Ed) 4(4):1–4. https://doi.org/10.16772/j.cnki.1673-1409.2007.04.047
14. Chong X (2021) Hybrid pso-svm for financial early-warning model of small and medium-sized enterprises. In: Proceedings of the 6th International Conference on Financial Innovation and Economic Development (ICFIED 2021), pp 107–114, https://doi.org/10.2991/aebmr.k.210319.020
15. Clerc M, Kennedy J (2002) The particle swarm–explosion, stability, and convergence in a multidimensional complex space. IEEE Trans Evol Comput 6(1):58–73. https://doi.org/10.1109/4235.985692
16. Derrouaoui SH, Bouzid Y, Guiatni M (2021) Pso based optimal gain scheduling backstepping flight controller design for a transformable quadrotor. J Intell Robot Syst 102(3):1–25. https://doi.org/10.1007/s10846-021-01422-1
17. Du Y, Xu F (2020) A hybrid multi-step probability selection particle swarm optimization with dynamic chaotic inertial weight and acceleration coefficients for numerical function optimization. Symmetry 12(922):1–25. https://doi.org/10.3390/sym12060922
18. Duan H, Qiao P (2014) Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning. Int J Intell Comput Cybern 7(1):24–37. https://doi.org/10.1108/IJICC-02-2014-0005
19. Engelbrecht AP (2007) Computational Intelligence: An Introduction, 2nd edn. John Wiley and Sons, West Sussex, UK
20. Faria J, Marques C, Pombo J et al (2023) Optimal sizing of renewable energy communities: a multiple swarms multi-objective particle swarm optimization approach. Energies 16:7227. https://doi.org/10.3390/en16217227
21. Fernandes Junior FE, Yen GG (2019) Particle swarm optimization of deep neural networks architectures for image classification. Swarm Evol Comput 49:62–74. https://doi.org/10.1016/j.swevo.2019.05.010
22. Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Am Stat Assoc 32(200):675–701. http://www.jstor.org/stable/2279372
23. Fu Y, Ding M, Zhou C et al (2009) Path planning for uav based on quantum-behaved particle swarm optimization. In: Proceeding of medical imaging, parallel processing of images, and optimization techniques (MIPPR 2009), https://doi.org/10.1117/12.832476
24. Fu Y, Ding M, Zhou C (2012) Phase angle-encoded and quantum-behaved particle swarm optimization applied to three-dimensional route planning for uav. IEEE Trans Syst Man Cybern Part A Syst Hum 42(2):511–526. https://doi.org/10.1109/TSMCA.2011.2159586
25. Gazi V (2012) Stochastic stability analysis of the particle dynamics in the pso algorithm. In: Proceeding of 2012 IEEE int. symp. on intelligent control (ISIC), Part of 2012 IEEE multi-conference on systems and control, Dubrovnik, Croatia, pp 708–713, https://doi.org/10.1109/ISIC.2012.6398264
26. Geng N, Chen Z, Nguyen QA et al (2021) Particle swarm optimization algorithm for the optimization of rescue task allocation with uncertain time constraints. Complex Intell Syst 7:873–890. https://doi.org/10.1007/s40747-020-00252-2
27. Ghasemi M, Akbari E, Rahimnejad A et al (2019) Phasor particle swarm optimization: a simple and efficient variant of pso. Soft Comput 23:9701–9718. https://doi.org/10.1007/s00500-018-3536-8
28. He X, Chen Y, Hu K et al (2022) Application of pso-optimized twin support vector machine in medium and long-term load fore-

casting under the background of new normal economy. Adv Multimed 2022:9. https://doi.org/10.1155/2022/2015538
29. Hu M, Wu T, Weir JD (2013) An adaptive particle swarm optimization with multiple adaptive methods. IEEE Trans Evol Comput 17(5):705–720. https://doi.org/10.1109/TEVC.2012.2211025
30. Hu S, Li K (2023) Bayesian network demand-forecasting model based on modified particle swarm optimization. Appl Sci 13:10088. https://doi.org/10.3390/app131810088
31. Huang Y, Wang X, Chen H (2022) Location selection for regional logistics center based on particle swarm optimization. Sustainability 14:16409. https://doi.org/10.3390/su142416409
32. Hung CW, Mao WL, Huang HY (2019) Modified pso algorithm on recurrent fuzzy neural network for system identification. Intell Autom Soft Comput 25(2):329–341. https://doi.org/10.31209/2019.100000093
33. Huynh NT, Nguyen TVT, Tam NT et al (2021) Optimizing magnification ratio for the flexible hinge displacement amplifier mechanism design. In: Long B, Kim Y, Ishizaki K, et al (eds) Proceedings of the 2nd annual international conference on Material, Machines and Methods for Sustainable development (MMMS2020), p 102, https://doi.org/10.1007/978-3-030-69610-8_102
34. Innocente MS, Sienz J (2010) Coefficients' settings in particle swarm optimization: insight and guidelines. In: Proceeding of IX argentinean congress on computational mechanics, II South American Congress on Computational Mechanics, and XXXI Iberian-Latin-American Congress on Computational Methods in Engineering, Buenos Aires, Argentina, pp 9253–9269, https://cimec.org.ar/ojs/index.php/mc/article/view/3666
35. Jiang M, Luo Y, Yang S (2007) Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. Inf Process Lett 102(1):8–16. https://doi.org/10.1016/j.ipl.2006.10.005
36. Jiang S, Jiang J, Zheng C et al (2019) An improved pso algorithm with migration behavior and asynchronous varying acceleration coefficient. In: Proceeding of 15th int. conference (ICIC 2019), Nanchang, China, pp 651–659, https://doi.org/10.1007/978-3-030-26766-7_59
37. Kadirkamanathan V, Selvarajah K, Fleming PJ (2006) Stability analysis of the particle dynamics in particle swarm optimizer. IEEE Trans Evol Comput 10(3):245–255. https://doi.org/10.1109/TEVC.2005.857077
38. Kaushal C, Islam MK, Althubiti SA et al (2022) A framework for interactive medical image segmentation using optimized swarm intelligence with convolutional neural networks. Comput Intell Neurosci 2022:1–21. https://doi.org/10.1155/2022/7935346
39. Kennedy J (1997) The particle swarm: social adaptation of knowledge. In: Proceeding of the IEEE int. conf. on evolutionary computation, pp 303–308, https://doi.org/10.1109/ICEC.1997.592326
40. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceeding of int. conference neural network (ICNN), pp 1942–1948, https://doi.org/10.1109/ICNN.1995.488968
41. Kerboua A, Boukli-Hacene F, Mourad KA (2020) Particle swarm optimization for micro-grid power management and load scheduling. Int J Energy Econ Pol 10(2):71–80. https://doi.org/10.32479/ijeep.8568
42. Kothandaraman N, Kaliaperumal V (2021) Combined particle swarm optimization and modified bilinear model (pso-mbm) algorithm for nonlinearity detection and spectral unmixing of satellite imageries. Int J Remote Sens 42(13):5190–5209. https://doi.org/10.1080/01431161.2021.1910369
43. Kumar L, Singh KU, Kumar I et al (2023) Robust medical image watermarking scheme using pso, lwt, and hessenberg decomposition. Appl Sci 13:7673. https://doi.org/10.3390/app13137673
44. Lee KH (2005) First Course on Fuzzy Theory and Applications. Springer, Heidelberg, https://link.springer.com/book/10.1007/3-540-32366-X
45. Lei K, Qiu Y, He Y (2006) A new adaptive well-chosen inertia weight strategy to automatically harmonize global and local search ability in particle swarm optimization. In: Proceeding of IEEE 2006 1st int. symp. on systems and control in aerospace and astronautics, Harbin, China, pp 977–980, https://doi.org/10.1109/ISSCAA.2006.1627487
46. Li Z, Han X, Yang M et al (2020) Multi-stage power source and grid coordination planning method considering grid uniformity. Global Energy Interconnection 3(4):303–312. https://doi.org/10.1016/j.gloei.2020.10.001
47. Liang JJ, Qu BY, Suganthan PN (2013) Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization. In: Technical report 201311, computational intelligence laboratory, Zhengzhou University, Zhengzhou, China and Technical Report, Nanyang Technological University, Singapore, https://www.researchgate.net/publication/271646935
48. Liu K, Cui Y, Ren J et al (2021) An improved particle swarm optimization algorithm for bayesian network structure learning via local information constraint. IEEE Access 9:40963–40971. https://doi.org/10.1109/ACCESS.2021.3065532
49. Liu X, Hou G, Yang L (2023) Combination optimization of green energy supply in data center based on simulated annealing particle swarm optimization algorithm. Front Earth Sci 11. https://doi.org/10.3389/feart.2023.1134523
50. Lu J, Hu H, Bai Y (2014) Radial basis function neural network based on an improved exponential decreasing inertia weight-particle swarm optimization algorithm for aqi prediction. Abstr Appl Anal 2014(SI11):1–9. https://doi.org/10.1155/2014/178313
51. Marouani H (2021) Optimization for the redundancy allocation problem of reliability using an improved particle swarm optimization algorithm. J Optim 2021:9. https://doi.org/10.1155/2021/6385713
52. Melo AG, Andrade FAA, Guedes IP et al (2022) Fuzzy gain-scheduling pid for uav position and altitude controllers. Sensors 22(2173):1–21. https://doi.org/10.3390/s22062173
53. Menos-Aikateriniadis C, Lamprinos I, Georgilakis PS (2022) Particle swarm optimization in residential demand-side management: a review on scheduling and control algorithms for demand response provision. Energies 15:2211. https://doi.org/10.3390/en15062211
54. Mourtzis D, Angelopoulos J (2023) Reactive power optimization based on the application of an improved particle swarm optimization algorithm. Machines 11:724. https://doi.org/10.3390/machines11070724
55. Mquqwana MA, Krishnamurthy S (2024) Particle swarm optimization for an optimal hybrid renewable energy microgrid system under uncertainty. Energies 17:422. https://doi.org/10.3390/en17020422
56. Nayak J, Swapnarekha H, Naik B et al (2023) 25 years of particle swarm optimization: flourishing voyage of two decades. Arch Comput Methods Eng 30:1663–1725. https://doi.org/10.1007/s11831-022-09849-x
57. Nguyen HD, Van CP, Nguyen TG et al (2023) Soil salinity prediction using hybrid machine learning and remote sensing in ben tre province on vietnam's mekong river delta. Environ Sci Pollut Res 30:74340–74357. https://doi.org/10.1007/s11356-023-27516-x
58. Phung MD, Ha QP (2021) Safety-enhanced uav path planning with spherical vector-based particle swarm optimization. Appl Soft Comput 107(107376):1–15. https://doi.org/10.1016/j.asoc.2021.107376
59. Poli R (2009) Mean and variance of the sampling distribution of particle swarm optimizers during stagnation. IEEE Trans

Evol Comput 13(4):712–721. https://doi.org/10.1109/TEVC.2008.2011744

60. Poli R, Broomhead D (2007) Exact analysis of the sampling distribution for the canonical particle swarm optimiser and its convergence during stagnation. In: Proceeding IEEE int. conf. on genetic and evolutionary computation conference (GECCO'07), London, England, pp 134–141, https://doi.org/10.1145/1276958.1276977
61. Priya BK, Reddy DA, Soliman WG et al (2022) Hybrid stepper motor: model, open-loop test, traditional pi, optimized pi, and optimized gain scheduled pi controllers. Int J Control Autom Syst 20(12):3915–922. https://doi.org/10.1007/s12555-021-0371-y
62. Qamar MS, Ali F, Armghan A et al (2021) Improvement of traveling salesman problem solution using hybrid algorithm based on best-worst ant system and particle swarm optimization. Appl Sci 11:4780. https://doi.org/10.3390/app11114780
63. Qu S, He T, Zhu G (2023) Model-assisted online optimization of gain-scheduled pid control using nsga-ii iterative genetic algorithm. Appl Sci 13:6444. https://doi.org/10.3390/app13116444
64. Rajania DK, Kumarc V (2020) Impact of controlling parameters on the performance of mopso algorithm. In: International Conference on Computational Intelligence and Data Science (ICCIDS 2019), pp 2132–2139, https://doi.org/10.1016/j.procs.2020.03.272
65. Ratnaweera A, Halgamuge SK, Watson HC (2004) Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. IEEE Trans Evol Comput 8(3):240–255. https://doi.org/10.1109/TEVC.2004.826071
66. del Rio A, Barambones O, Uralde J et al (2023) Particle swarm optimization-based control for maximum power point tracking implemented in a real time photovoltaic system. Information 14:556. https://doi.org/10.3390/info14100556
67. Roy C, Das DK (2021) A hybrid genetic algorithm (ga)—particle swarm optimization (pso) algorithm for demand side management in smart grid considering wind power for cost optimization. Sādhanā 46. https://doi.org/10.1007/s12046-021-01626-z
68. Saifullah S, Drezewski R (2024) Advanced medical image segmentation enhancement: a particle-swarm-optimization-based histogram equalization approach. Appl Sci 14:923. https://doi.org/10.3390/app14020923
69. Sedghizadeh S, Beheshti S (2018) Particle swarm optimization based fuzzy gain scheduled subspace predictive control. Eng Appl Artif Intell 67:331–344. https://doi.org/10.1016/j.engappai.2017.10.009
70. Shami TM, El-Saleh AA, Alswaitti M et al (2022) Particle swarm optimization: a comprehensive survey. IEEE Access 10:10031–10061. https://doi.org/10.1109/ACCESS.2022.3142859
71. Sheela MS, Arun CA (2022) Hybrid pso-svm algorithm for covid-19 screening and quantification. Int J Inf Technol 14(4):2049–2056. https://doi.org/10.1007/s41870-021-00856-y
72. Shi K, Yuan X, Liu L (2018) Model predictive controller-based multi-model control system for longitudinal stability of distributed drive electric vehicle. ISA Trans 72:44–55. https://doi.org/10.1016/j.isatra.2017.10.013
73. Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In: Proceeding of IEEE international conference on evolutionary computation, Anchorage, pp 69–73, https://doi.org/10.1109/ICEC.1998.699146
74. Shi Y, Eberhart RC (1999) Empirical study of particle swarm optimization. In: Proceeding of the 1999 Congress on Evolutionary Computation (CEC99), pp 1945–1950, https://doi.org/10.1109/CEC.1999.785511
75. Shu J, Li J (2009) An improved self-adaptive particle swarm optimization algorithm with simulated annealing. In: Proceeding of the 3rd international conference on intelligent information technology application (IITA '09), Nanchang, China, pp 396–399, https://doi.org/10.1109/IITA.2009.476
76. Song M, Liu S, Li W et al (2021) A continuous space location model and a particle swarm optimization-based heuristic algorithm for maximizing the allocation of ocean-moored buoys. IEEE Access 9:32249–32262. https://doi.org/10.1109/ACCESS.2021.3060464
77. Song X, Wang C (2022) Hyperspectral remote sensing image classification based on spectral-spatial feature fusion and pso algorithm. J Phys Conf Series 2189:012010. https://doi.org/10.1088/1742-6596/2189/1/012010
78. Su T, Xu H, Zhou X (2019) Particle swarm optimization-based association rule mining in big data environment. IEEE Access 7:161008–161016. https://doi.org/10.1109/ACCESS.2019.2951195
79. Sugeno M (1993) Fuzzy measures and fuzzy integrals— a survey. In: Dubois D, Prade H, Yager RR (eds) Readings in fuzzy sets for intelligent systems. Morgan Kaufmann, p 251–257, https://doi.org/10.1016/B978-1-4832-1450-4.50027-4
80. Susanto ADNW, Suparwito H (2023) Svm-pso algorithm for tweet sentiment analysis #besoksenin. Indonesian Journal of Information Systems 6(1):36–47 https://ojs.uajy.ac.id/index.php/IJIS/issue/view/437
81. Tosoni D, Galli C, Hanne T et al (2022) Benchmarking metaheuristic optimization algorithms on travelling salesman problems. In: Proceedings of the international conference on e-society, e-learning and e-technologies (ICSLT '22), pp 20–25, https://doi.org/10.1145/3545922.3545926
82. Tran VT, Le MH, Vo MT et al (2023) Optimization design for die-sinking edm process parameters employing effective intelligent method. Cogent Eng 10(2). https://doi.org/10.1080/23311916.2023.2264060
83. Trelea IC (2003) The particle swarm optimization algorithm: convergence analysis and parameter selection. Inf Process Lett 85:317–325. https://doi.org/10.1016/S0020-0190(02)00447-7
84. Trojovsky P, Dehghani M (2022) Pelican optimization algorithm: a novel nature-inspired algorithm for engineering applications. Sensors 22(855):1–34. https://doi.org/10.3390/s22030855
85. Waluyo A, Jatnika H, Permatasari MRS, et al (2020) Data mining optimization uses c4.5 classification and particle swarm optimization (pso) in the location selection of student boardinghouses. In: IOP Conference Series: Materials Science and Engineering, https://doi.org/10.1088/1757-899X/874/1/012024
86. Wang G, Liu Z (2012) An analysis of nonlinear acceleration coefficients adjustment for pso. In: Proceeding of 4th international conference on Artificial Intelligence and Computational Intelligence (AICI 2012), Chengdu, China, pp 698–705, https://doi.org/10.1007/978-3-642-33478-8_86
87. Wang L, Guo N, Yue P et al (2022) Regulation of evapotranspiration in different precipitation zones and its application in high-temperature and drought monitoring. Remote Sensing 14:6190. https://doi.org/10.3390/rs14246190
88. Wilcoxon F (1945) Individual comparisons by ranking methods. Biometrics Bull 1(6):80–83. https://doi.org/10.2307/3001968
89. Wu Y, Wu C, Wang L et al (2021) Radar target tracking algorithm based on new particle swarm optimization particle filter. In: Proceeding of the 10th International Conference on Networks, Communication and Computing (ICNCC 2021), Beijing, China, pp 91–96, https://doi.org/10.1145/3510513.3510528
90. Wu Z, Zhou J (2007) A self-adaptive particle swarm optimization algorithm with individual coefficients adjustment. In: Proceeding of 2007 international conference on Computational Intelligence and Security (CIS 2007), Harbin, China, pp 133–136, https://doi.org/10.1109/CIS.2007.95
91. Xiao Y, Zhang L (2023) Smart grid energy storage capacity planning and scheduling optimization through pso-gru and

multihead-attention. Front Energy Res 11. https://doi.org/10.3389/fenrg.2023.1254371

92. Xin J, Chen G, Hai Y (2009) A particle swarm optimizer with multi-stage linearly-decreasing inertia weight. In: Proceeding of the second international joint conference on Computational Sciences and Optimization (CSO 2009), Sanya, Hainan, China, pp 505–508, https://doi.org/10.1109/CSO.2009.420
93. Xin J, Li S, Sheng J et al (2019) Application of improved particle swarm optimization for navigation of unmanned surface vehicles. Sensors 19(14):1–21. https://doi.org/10.3390/s19143096
94. Yang W, Zhou X (2021) Luo Y (2021) Simultaneously optimizing inertia weight and acceleration coefficients via introducing new functions into pso algorithm. J Phys Conf Ser 1754(1):012195. https://doi.org/10.1088/1742-6596/1754/1/012195
95. Yasuda K, Ide A, Iwasaki N (2003) Adaptive particle swarm optimization. In: Proceeding of 2003 IEEE int. conference on SMC, pp 1554–1559, https://doi.org/10.1109/ICSMC.2003.1244633
96. Yazdani D, Yazdani D, Yazdani D et al (2023) A species-based particle swarm optimization with adaptive population size and deactivation of species for dynamic optimization problems. ACM Trans Evol Learn Optim 3(4):1–25. https://doi.org/10.1145/3604812
97. Yin S, Jin M, Lu H et al (2023) Reinforcement-learning-based parameter adaptation methods. Complex Intell Syst 9:5585–5609. https://doi.org/10.1007/s40747-023-01012-8
98. Yue CT, Price KV, Suganthan PN et al (2019) Problem definitions and evaluation criteria for the cec 2020 special session and competition on single objective bound constrained numerical optimization. Tech. rep., Computational Intelligence Laboratory, Zhengzhou University, https://github.com/P-N-Suganthan/2020-Bound-Constrained-Opt-Benchmark
99. Zaini FA, Sulaima MF, Razak IAWA et al (2023) A review on the applications of pso-based algorithm in demand side management: challenges and opportunities. IEEE Access 11:53373–53400. https://doi.org/10.1109/ACCESS.2023.3278261
100. Zhang S, Tong F, Li M et al (2021) Research on multi-dimensional optimal location selection of maintenance station based on big data of vehicle trajectory. Entropy 23:495. https://doi.org/10.3390/e23050495
101. Zhang X, Du Y, Qin G et al (2005) Adaptive particle swarm algorithm with dynamically changing inertia weight. J Xi'an Jiaotong Univ 39(19):1039–1042. 0253-987X, 10-1039-04
102. Zhang X, Ren Y, Zhen G et al (2023) A color image contrast enhancement method based on improved pso. PLoS One 18(2):e0274054. https://doi.org/10.1371/journal.pone.0274054
103. Zhao S, Zhang T, Ma S et al (2023) Sea-horse optimizer: a novel nature-inspired meta-heuristic for global optimization problems. Appl Intell 53:11833–11860. https://doi.org/10.1007/s10489-022-03994-3
104. Zheng R, Zhang Y, Yang K (2022) A transfer learning-based particle swarm optimization algorithm for the traveling salesman problem. J Comput Design Eng 9(3):933–948. https://doi.org/10.1093/jcde/qwac039
105. Zhong J, Feng Y, Tang S et al (2023) A collaborative neurodynamic optimization algorithm for the traveling salesman problem. Complex Intell Syst 9:1809–1821. https://doi.org/10.1007/s40747-022-00884-6
106. Zhou R, Zhang L, Fu C et al (2022) Fuzzy neural network pid strategy based on pso optimization for ph control of water and fertilizer integration. Appl Sci 12(7383):1–19. https://doi.org/10.3390/app12157383
107. Zhu H, Tanabe Y, Baba T (2008) A random time-varying particle swarm optimization for the real time location systems. IEEJ Trans Electr Electron Eng 128(12):1747–1760. https://doi.org/10.1541/ieejeiss.128.1747
108. Zhu T, Zheng H, Ma Z (2019) A chaotic particle swarm optimization algorithm for solving optimal power system problem of electric vehicle. Adv Mech Eng 11(3):1–9. https://doi.org/10.1177/1687814019833500