Adaptive Multi-Objective Particle Swarm Optimization Using Three-Stage Strategy

Weimin Huang

Henan Polytechnic University

Wei Zhang (✉ zwei1563@hpu.edu.cn)

Henan Polytechnic University <https://orcid.org/0000-0001-9156-6779>

Research Article

Keywords: multi-objective optimization, particle swarm optimization, adaptive, three-stages strategy, decomposition

Posted Date: April 15th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-420231/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Adaptive multi-objective particle swarm optimization using three-stage strategy

Weimin Huang · Wei Zhang

Received: date / Accepted: date

Abstract It is one of the crucial problems in solving multi-objective problems (MOPs) that balance the convergence and diversity of the algorithm to obtain an outstanding Pareto optimal solution set. In order to elevate the performance further and improve the optimization efficiency of multi-objective particle swarm optimization (MOPSO), a novel adaptive MOPSO using a three-stage strategy (tssAMOPSO) is proposed in this paper, which can effectively balance the exploration and exploitation of the population and facilitate the convergence and diversity of MOPSO. Firstly, an adaptive flight parameter adjustment, formulated by the convergence contribution of nondominated solutions, can ameliorate the convergence and diversity of the algorithm enormously. Secondly, the population carries out the three-stage strategy of optimization in each iteration, namely adaptive optimization, decomposition, and Gaussian attenuation mutation. The three-stage strategy remarkably promotes the diversity and efficiency of the optimization process. Moreover, the convergence of three-stage optimization strategy is analyzed. Then, memory interval is equipped with particles to record the recent positions, which vastly improves the reliability of personal best selection. In the maintenance of external archive, the proposed fusion index can enhance the quality of nondominated solutions directly. Finally, comparative experiments are designed by a series of benchmark instances to verify the performance of tssAMOPSO. Experimental results show that the proposed

algorithm achieves admirable performance compared with other contrast algorithms.

Keywords multi-objective optimization · particle swarm optimization · adaptive · three-stages strategy · decomposition

1 Introduction

Multi-objective optimization problems (MOPs) are the most common form of decision-making and practical engineering problems (Zhang et al. 2015). Two main characteristics of MOPs can be described as follows. One is that there are multiple conflicts among objectives, which means that the improvement of one objective will lead to a degeneration in one or even more remaining objectives (Brockhoff and Zitzler 2009). The other is that the optimal solution is a set of non-inferior solutions that cannot be compared with each other directly, which is named the Pareto optimal solution set (Jara 2014).

In solving MOPs, it is an important measure standard of the performance that the obtained nondominated solutions can converge to the Pareto optimal solution set quickly and effectively, and can simultaneously keep the nondominated solutions distributed uniformly in Pareto front, that is, the convergence and diversity of the algorithms (Fonseca and Fleming 1998; Li and Zhang 2009). Nevertheless, overemphasizing the convergence of the algorithm will result in the aggregation of nondominated solutions and falling into a local optimum. Conversely, excessive emphasis on the diversity will slow down the convergence speed and affect the final convergence accuracy of the nondominated solution set (Fakhouri et al. 2020). Therefore, effectively balancing the convergence and diversity of the algorithm is an important topic in MOPs (Chakraborty et al. 2011).

Weimin Huang
School of Electrical Engineering and Automation, Henan Polytechnic University, Jiaozuo 454003, China.
E-mail: weiminh97@gmail.com

Wei Zhang
School of Electrical Engineering and Automation, Henan Polytechnic University, Jiaozuo 454003, China.
E-mail: zwei1563@hpu.edu.cn

In the development of multi-objective algorithms (MOAs), multi-objective evolutionary algorithms (MOEAs) are considered to be the most influential MOAs (Lee and Kim 2013), and the successful application of MOEAs shines brilliantly. Since the evolutionary algorithms can work out multiple objective functions in parallel based on population search and has ideal results in solving MOPs, MOEAs have attracted extensive attention from the academic community (Yen and Lu 2003). Therefore, MOEAs have developed rapidly and many successful proposals have been published one after another. The most representative algorithms include as follows, nondominant sorting genetic algorithm (NSGA) (Srinivas and Deb 1994) and NSGA-II (Deb et al. 2002), strength Pareto evolutionary algorithm (SPEA) (Zitzler and Thiele 1999) and SPEA2 (Zitzler et al. 2001), multi-objective evolutionary algorithm based on decomposition (MOEA/D) (Zhang and Li 2007), and other algorithms (Zhang et al. 2016; Tan et al. 2001.).

Stimulated by the successful application of MOEAs, many other swarm intelligence optimization algorithms have been tried to solve MOPs as well, including multi-objective ant colony algorithms (MOACAs) (Liu and Liu 2019), multi-objective evolutionary immune algorithms (MOIAs) (Qiao et al. 2020), multi-objective artificial bee colony algorithms (MOABCs) (Martin-Moreno and VegaRodriguez 2018), and multi-objective particle swarm optimizations (MOPSOs). Among them, particle swarm optimization (PSO) is one of the most popular swarm intelligence algorithms (Wang et al. 2018). Owing to its simple implementation and highly computational efficiency in single-objective optimization problems (SOPs), it is naturally considered that the excellent performance can be also achieved in solving MOPs.

MOPSO is a population-based intelligent calculation algorithm and each particle represents a potential solution to MOPs. The algorithm seeks the final solution by iteratively searching for the global optimal solution in the population (Coello and Lechuga 2002). In the last two decades, scholars have conducted a lot of studies on MOPSO, and have achieved a series of achievements. However, some basic problems of MOPSO have not been determined in common (Meza et al. 2017). For example, when PSO is applied to MOPs, the components are necessary to be modified to ensure that the algorithm is suitable for MOPs. Since there is no real optimal solution in MOPs, the global best solution (*gbest*) and personal best solution (*pbest*) of MOPSO must be selected by specific strategies (Lin et al. 2015). Moreover, when a single-objective optimization algorithm is extended to a multi-objective optimization algorithm, an additional method need to be adopted to store the nondominated solutions.

For MOPSO, except for the problems of insufficient convergence and diversity, the difficulty also lies in how to bal-

ance the global and the local search ability, that is, to strike a balance between the two processes of exploration and exploitation (Nebro et al. 2009). In terms of the exploration, the rapid convergence of MOPSO easily leads to the premature convergence and local optimum (Zhang et al. 2018). From the aspect of the exploitation, the single search strategy of PSO will bring about the problem of insufficient convergence in solving complex MOPs (Coello et al. 2004). In addition, the global best particle of MOPSO changes continually, which can make the balance of exploration and exploitation more difficult to achieve (Moubayed et al. 2014).

In summary, MOPSO has insufficient convergence and diversity in solving complex MOPs, and the exploration and exploitation cannot be effectively balanced (Maghawry et al. 2020; Wei et al. 2020). Therefore, the improvement of the MOPSO algorithm still has adequate research value. To further enhance the performance of MOPSO, an adaptive MOPSO using the three-stages strategy (tssAMOPSO) is proposed in this paper. The main contributions are presented as follows.

- (1) The optimization mechanism of the proposed tssAMOPSO algorithm is designed by using the idea of hierarchical learning, and the three-stage strategy embedded with three learning modes is formed eventually. The implementation of the three-stage strategy can make the algorithm explore larger scope and obtain more potential solutions in each iteration so that the optimization efficiency of the algorithm is improved. The three-stage strategy and the three learning modes are combined to enrich the search strategies of particles, which makes the possess of each stage have obvious different characteristics. Therefore, the performance of the proposed algorithm can be promoted in multiple aspects simultaneously.
- (2) The formulating of three-stage strategy and the settings of three learning modes are clarified to illustrate the rationality of optimization mechanism, and the basic principles and implementation procedures of each stage are introduced explicitly. Furthermore, the convergence analysis of tssAMOPSO has been provided. The optimization process of the three-stage strategy goes forward one by one, which means that the stage II and stage III are premised on the stage I. Therefore, the convergence of three stages is proved from the perspective of progressiveness in this paper.
- (3) An adaptive strategy is established to strike a balance between exploration and exploitation of the proposed algorithm further. The convergence contribution of the population is defined, and a mechanism is developed to detect the variations of evolutionary environment in optimization process, which can strengthen the ability of exploration and exploitation. Moreover, the memory interval, constituted by the positions of particles in the re-

cent iterations, is equipped to the particles for selecting the reliable personal best solution by convergence index, and a more reasonable maintenance of the external archive is proposed to improve the quality of nondominated solutions.

This paper is organized as follows. In Section 2, we briefly review some related literature of MOPSO. Section 3 introduces the framework and all components of the proposed tssAMOPSO in detail, including adaptive parameters adjustment, *gbest* selection, three-stage strategy, *pbest* selection, and maintenance of external archive. The convergence of tssAMOPSO is analyzed in Section 4, and the experimental studies are presented in Section 5. The conclusion is summarized in the last Section.

2 Literature review

Since PSO was firstly proposed by Kennedy and Eberhart in 1995, numerous scholars have devoted themselves into the study of PSO. In the early development period of PSO algorithm, the main work is to solve SOPs, and that is also the original intention of PSO. During this period, many scholars put forward a variety of schemes to upgrade the performance of PSO algorithm, which also has a profound impact on the follow-up research work on MOPs (Huang et al. 2019; Zhu et al. 2020).

With the increasing demand for solving MOPs, scholars began to apply PSO algorithm to solve MOPs. Among the abundant improvement schemes, the most representative is the MOPSO based on the adaptive grid (agMOPSO), which was proposed by Coello et al. (2004). The basic framework of MOPSO is developed completely by agMOPSO based on Pareto dominance. The algorithm employs an archive to store the discovered nondominated solutions, and the adaptive grid technology is used to maintain the external archive and guide the update of particles. In addition, the mutation is added to the algorithm in order to avoid the problems caused by the fast convergence of MOPSO. Although agMOPSO shows certainly advantages in solving MOPs compared with traditional MOEAs, it is difficult to solve the problems with multiple local Pareto front, and the diversity of nondominated solutions is poorly performed. Correspondingly, Nebro et al. (2009) proposed a PSO-based metaheuristic for multi-objective optimization (SMPSO), where the algorithm uses the speed constrained to generate new effective positions of the particles even in the cases of the velocity being too high. Another characteristic of SMPSO is that the polynomial mutation is employed as a turbulence factor. Similarly, SMPSO also adopts an external archive to store the nondominated solutions found during the optimization process. Compared with other MOAs, SMPSO shows considerable convergence, but the diversity is obviously insufficient.

In order to ameliorate the diversity of MOPSO, Raquel and Naval (2005) introduced a MOPSO based on crowding distance (cdMOPSO). The crowding distance is applied to evaluate the density of particles and solutions in the objective space, which is used to select the global best particle and maintain the distribution of nondominated solutions in the external archive. Although cdMOPSO has adopted effective measures to strengthen the diversity, there are not any strategies in terms of the convergence improvement. Therefore, cdMOPSO leads to another extreme situation, that is, the algorithm has excellent diversity but embarrassing convergence.

After the establishment of the basic framework of MOPSO, the research on MOPSO has entered a period of all flowers blooming together. Some scholars apply the idea of decomposition to MOPSO and propose various enhancements (Peng and Zhang 2008; Gong et al. 2014; Dai et al. 2015). Different from the above-mentioned algorithms that determine the search mechanism by the Pareto dominance relationship, Peng and Zhang (2008) proposed a MOPSO based on the decomposition framework of MOEA/D algorithm (MOPSO/D), which can decompose MOPs into multiple SOPs. The genetic operators of evolutionary algorithm are replaced by the search mechanism of PSO, and a brand-new MOPSO algorithm is established finally. Although MOPSO/D maintains an external archive to store the global best particles of each SOP, it does not optimize the search mechanism of PSO, and the overall performance of the algorithm is not ideal. Moreover, since the Pareto dominance is replaced by decomposition, most of the algorithm based on decomposition cannot cover the entire Pareto front in solving some complex MOPs with multiple local Pareto front. Some novel perspectives on the improvement of MOPSO have also been discovered and proposed, such as the population adjustment. Some scholars have studied the effect of population size on the performance of MOPSO. For example, Tan et al. (2001) analyzed the relationship between the population size and the performance of MOPSO algorithm, and the results showed that a large population size could obtain a better global optimal solution. Bolufé-Röhler and Chen (2013) were suggested that the premise of obtaining the optimal solution was to select the appropriate population size. Small population size can accelerate the convergence speed of the algorithm and upgrade the efficiency of function evaluation, but it may lead to premature convergence. Instead, large population size can reduce the risk, but it may reduce the efficiency of the algorithm. Therefore, some MOPSOs based on population size adjustment are emerging gradually. Leong and Yen (2008) described a MOPSO algorithm based on the dynamic population size and adaptive local archives (DMOPSO), where the algorithm designs a population size assessment rule based on the evaluation of cell hierarchical density. According to this

Fig. 1 Framework of the proposed tssMOPSO.

rule, the dynamic population size adjustment strategy is formulated to realize the adaptive growth and deletion of population size. Simulation results show that DMOPSO has made some progress in the diversity and convergence of solutions. However, the impact of improvement on MOPSO includes many aspects, such as flight parameters, optimal particle selection, and search strategy. It is not enough to enhance the performance of the algorithm only by dynamically adjusting the population size.

For accurately switching the exploration and exploitation process of MOPSO, some adaptive strategies have been proposed based on the evolutionary environment of the algorithm. Han et al. (2018) proposed an adaptive gradient MOPSO (AGMOPSO), where the algorithm mainly adopts the stocktickerMOG method and designs a self-adaptive flight parameter adjustment mechanism. The density of particles, the distance between particles and the global best particle, are used to establish the adaptive parameter adjustment of AGMOPSO. However, it is obviously short of accuracy in just using the diversity of particles as the reference standard to evaluate the evolutionary environment of the algorithm. An ingenious method for monitoring the evolutionary environment was proposed by Hu and Yen (2015), where the entropy of population is used as the feedback to obtain the current state of the algorithm. Then, the flight parameter adjustment can be formulated. Furthermore, the objective space of the algorithm is converted into a two-dimensional grid, named parallel cell coordinate, to simplify the calculation and determine other necessary parameters. This algorithm has achieved a satisfactory result in balancing exploration and exploitation and shows decent convergence and diversity. Other algorithms that adopt population information as feedback to determine the parameter adjustment schemes are also proposed in Han et al. 2017 and Zhang et al. 2018.

There are other efficient schemes proposed in the development process of MOPSO. Different from the above

MOPSO with adaptive strategy, a new optimization mechanism, multiple learning modes, has been developed to enrich the optimization strategy and enhance the optimization ability of MOPSO. Four learning modes were designed in Li et al. (2012), but it was only employed to solve SOPs. On this basis, Kumar et al. (2015) expanded further and introduced a self-learning MOPSO for MOPs (MOSLPSO). According to the influence of the velocity and position update of particles on the optimization, four learning modes are applied to promote the performance of a certain aspect of MOSLPSO. Based on the special advantages of different learning modes, the particles in the population adopt different learning modes, which means that the algorithm can boost the performance simultaneously from multiple perspectives. There are also some other enhancements of MOPSO in Qu et al. (2020) and Tao et al. (2020).

Although the improved algorithms mentioned above can upgrade the optimization ability and balance the diversity and convergence of MOPSO to some extent, the presently proposed schemes are more or less lack of comprehensiveness (Wei et al. 2020). Moreover, The applications of multiple learning modes, simply changing the velocity and position update formula, are basically unreasonable enough. Most of MOPSOs just rely on one search strategy to update the velocity and position of particles without considering the optimization efficiency of the algorithm. Therefore, there are still many gaps in the improvement of MOPSO algorithm.

3 Proposed tssAMOPSO

The whole framework of the proposed tssAMOPSO algorithm based on the three-stage strategy is presented in Fig. 1.

The proposed tssAMOPSO is mainly composed of three parts from left to right in the diagram, including obtaining the basic information of MOPSO, optimization strategies, and storing the nondominated solutions. In the first part, the

evolutionary environment of tssAMOPSO is detected by the convergence contribution of nondominated solutions, and the adaptive flight parameter adjustment is formulated according to the changes of algorithm status. To balance the convergence and diversity of the algorithm, the convergence global best solution (*c-gbest*) and the diversity global best solution (*d-gbest*) are adaptively selected to guide the optimization respectively. The memory interval is constituted by the particle's positions in recent iterations, and the personal best solution (*pbest*) is selected by the convergence of positions in the memory interval. For the second part, the population implements the three-stage optimization strategy in each iteration. Firstly, the particles conduct the adaptive strategy as the first-learning mode. Then, some particles are decomposed into two new subparticles, and two new learning modes are employed to update the velocities and positions. Finally, some particles were randomly selected to carry out the Gaussian attenuation mutation. The searched potential solutions of the algorithm are dramatically increased by using the three-stage strategy in each iteration, that is, the optimization efficiency can be enhanced. The maintenance of the external archive is important for MOAs because it can directly control the final performance of the nondominated solution set. The external archive is maintained by the fusion index in tssAMOPSO, which can reflect the synthesis performance of convergence and diversity of the nondominated solutions. The implementation details of the proposed tssAMOPSO are stated as follows.

3.1 Adaptive flight parameter adjustment

For MOPSO, an excellent adaptive parameters adjustment can commendably strengthen the ability of exploration and exploitation. The inertia weight (ω) and learning factors (c_1 and c_2) are adjusted according to the changes of evolutionary environment in order to achieve the accurate regulation of flight parameters.

When the searched nondominated solutions need to be stored in the external archive, the Pareto dominance degree between the new solutions and the previously found solutions can be calculated, and the convergence contribution is obtained as follows,

$$Ap(t) = \max_{i=1,2,\dots,n_p} Cv(s_i, Ea) \quad (1)$$

where n_p is the number of non-inferior solutions searched in iteration t , s_i is the i -th non-inferior solution, and Ea is the external archive. $Cv(s_i, Ea)$ represents the maximum dominance of s_i for all nondominated solutions in the external archive, which can be calculated by

$$Cv(s_i, Ea) = \max_{s_n \in Ea} \rho(s_i, s_n) \quad (2)$$

where s_n is a nondominated solution in the external archive and $\rho(s_i, s_n)$ denotes the dominance degree between s_i and s_n . $\rho(s_i, s_n)$ is given as follows,

$$\rho(s_i, s_n) = \begin{cases} \frac{1}{M} \sum_{m=1}^M \frac{|f_m(s_i) - f_m(s_n)|}{(f_{m,max} - f_{m,min})}, & s_i \prec s_n \\ 0, & \text{Otherwise} \end{cases} \quad (3)$$

where M is the number of objectives, $f_m(\cdot)$ is the fitness value on m -th objective, and $f_{m,max}$ and $f_{m,min}$ are the maximum and minimum fitness value on m -th objective, respectively.

From the above definition of convergence contribution, we could know that Ap represents the current convergence status of the algorithm. When a large Ap is obtained, it means that the population is in the exploratory status, and the global search ability should be maintained. With the decrease of Ap , the population gradually reaches convergence, and the local search should be strengthened. When Ap approaches to 0, it indicates that the population has achieved a stable state of convergence. Take ZDT4 problem as an example, the variations of convergence contribution and ΔAp (the incremental of convergence contribution, which is calculated by $\Delta Ap(t) = Ap(t) - Ap(t-1)$) are shown in Fig. 2.

Fig. 2 Curve of convergence contribution and detected on ZDT4.

The related research on the flight parameters of MOPSO has been introduced in Rajani et al. (2020). Generally, the larger ω and c_1 , and the smaller c_2 can boost the global exploration. On the contrary, the smaller ω and c_1 , and the larger c_2 are conducive to enhance the local exploitation. In order to avoid the influence of parameter adjustment on search continuity, the incremental adjustment of parameters is adopted as follows,

$$\xi(t) = \begin{cases} \xi(t-1), & \Delta Ap(t) = 0 \\ \xi(t-1) + \gamma |\Delta Ap(t)|, & \Delta Ap(t) > 0 \\ \xi(t-1) - \gamma |\Delta Ap(t)|, & \Delta Ap(t) < 0 \end{cases} \quad (4)$$

where $\xi(t) = [\omega(t), c_1(t), c_2(t)]^T$ is the parameters vector, and $\gamma = [\lambda_\omega, \lambda_{c_1}, \lambda_{c_2}]^T$ is the vector constituted by the incremental step factors of ω , c_1 and c_2 . Usually, the variation range of inertia weight is $[0.4, 0.9]$ and the variation range of learning factors is $[0, 2.5]$ (Hu and Yen 2015.). The initial value of ω is set to 0.9 and the initial value of c_1 and c_2 are both set to 1.5.

Remark 1: The convergence contribution is closely related to the current evolutionary status of the algorithm. Therefore, there is an interpretable theoretical basis for using the convergence contribution as the feedback to detect the evolutionary environment of the algorithm. The increment of convergence contribution represents the variation of Ap , and the positive value or the negative value of ΔAp can reflect the changes in the direction of evolutionary environment for the proposed tssAMOPSO. Moreover, the value of ΔAp indicates the variation rate of convergence state, which means that the incremental of parameters in the adaptive strategy can be determined by the absolute value of ΔAp .

3.2 Global best selection

To balance the convergence and diversity of MOPSO, the nondominated solution in the external archive with optimal convergence and the nondominated solution with optimal diversity are selected as the convergence *gbest* (*c-gbest*) and diversity *gbest* (*d-gbest*) to guide the population optimization respectively. By using the dual *gbest*, the optimization direction of the algorithm can be adjusted flexibly, which tremendously reduces the difficulty of balancing the exploration and exploitation process.

3.2.1 Convergence global best

The average ranking of particles is obtained by sorting the fitness values, which is presented as

$$AR(x_i) = \sum_{m=1}^M R_m(x_i) \quad (5)$$

where $R_m(x_i)$ is the rank of i -th particle on m -th objective. The other convergence index, named global detriment, can be described as

$$GD(x_i) = \sum_{x_i \neq x_j} \sum_{m=1}^M \max(f_m(x_i) - f_m(x_j), 0) \quad (6)$$

The two indexes (AR and GD) have their own characteristics. When AR is employed to select the optimal solutions, the particles will be clustered at the edge of Pareto front. Contrarily, GD will ignore the particles which lie on the edge (Li et al. 2016). Using the complementation of the AR

and GD , the above two indicators are combined to ameliorate the comprehensiveness of evaluation, the fusion ranking index of convergence is given as

$$FR(x_i) = \eta_1 AR(x_i) + \eta_2 GD(x_i) \quad (7)$$

where η_1 and η_2 are the weight of AR and GD , and they are usually set to 0.4 and 0.6 respectively. Then, *c-gbest* of the proposed algorithm can be determined by

$$c\text{-gbest} = \min(FR(s_n)) , \quad s_n \in Ea \quad (8)$$

3.2.2 Diversity global best

Crowding distance (Raquel and Naval 2005) is one of the most common methods to evaluate the distribution of non-dominated solutions, and the Euclidean distance between two particles is calculated by

$$dist(x_i, x_j) = \sum_{m=1}^M \sqrt{(f_m(x_i) - f_m(x_j))^2} \quad (9)$$

To avoid the influence of individual with poor performance on the evaluation, the local crowding distance is proposed to assess the distribution of solutions, which can be described as

$$LCD(x_i) = \frac{1}{n} \sum_{j=1, i \neq j}^n R(dist(x_i, x_j)) \quad (10)$$

where n is a positive integer. The value of n is an artificially determined parameter to indicate the number of particles included in a local area, that is, $LCD(x_i)$ means the average distance between the nearest n particles and the i -th particle. Therefore, *d-gbest* of the proposed algorithm can be determined by

$$d\text{-gbest} = \min(LCD(s_n)) , \quad s_n \in Ea \quad (11)$$

3.3 Three-stage strategy

In this paper, three kinds of learning modes are provided for particles, and a three-stage strategy is finally formed by combining Gaussian attenuation mutation. The three learning modes are formulated to improve the convergence and diversity of MOPSO, respectively. The three-stage strategy of MOPSO is shown in Fig. 3, and the details of each stage are presented as follows.

Fig. 3 Three-stage strategy of the i -th particle.

3.3.1 Stage I: adaptive optimization

The target of first stage is to ensure the basic convergence and diversity of the proposed algorithm. The velocity and position of particles in population need to be updated by the first learning mode, which is similar to the optimization of original MOPSO. The first learning mode is described as

$$\begin{cases} v_{i,d}^1(t+1) = \omega_i(t)v_{i,d}^1(t) + c_1(t)r_1(pbest_{i,d}(t) - x_{i,d}^1(t)) \\ \quad + c_2(t)r_2(gbest_d(t) - x_{i,d}^1(t)) \\ x_{i,d}^1(t+1) = x_{i,d}^1(t) + v_{i,d}^1(t+1) \end{cases} \quad (12)$$

The velocity of particles is updated by an adaptive strategy, that is, the flight parameters of the algorithm are updated adaptively. Moreover, the particles can be guided by different $gbest$ to promote the convergence and diversity. The simple and effective measure is that $gbest$ can be selected from the c - $gbest$ and d - $gbest$, and the selected probability of two $gbest$ is determined according to the evolutionary environment. The $gbest$ can be selected by

$$gbest(t) = \begin{cases} d\text{-}gbest, & r \leq Ap(t) \\ c\text{-}gbest, & \text{Otherwise} \end{cases} \quad (13)$$

where r is a random number in the interval of $[0,1]$.

Remark 2: The selected method of $gbest$ enables the particles to switch the status of exploration and exploitation adaptively according to the current environment of the algorithm. When the algorithm is in global exploration status (the value of Ap is larger), the particles should probably select d - $gbest$ as the flight guidance to promote the diversity of the algorithm. Contrarily, if the algorithm is in local exploitation (the value of Ap is smaller), the particles could select c - $gbest$ to ensure the convergence of the algorithm with

a relatively high probability. Therefore, the adaptive strategy of parameters and $gbest$ can effectively elevate the performance of the algorithm.

3.3.2 Stage II: decomposition

The target of second stage is to improve the optimization efficiency of the algorithm by adding the potential solutions in each iteration. According to the mechanism of decomposition, the selected particle will be split into two subparticles, and each subparticle employs a distinctive learning mode, which is specialized to enhance the diversity and convergence respectively.

The idea of vector decomposition is adopted to divide the particles, and the update method of position and velocity for subparticles are named the additional learning modes, that is, the second-learning mode and the third-learning mode. The decomposition process of particles is shown in Fig. 4.

Fig. 4 Decomposition of particle.

It can be seen from Fig. 4 that the parallelogram rule of vector operation is applied to design the learning modes for particles. The velocity obtained by the first-learning mode in Stage I is equal to the sum of the velocities obtained by the second-learning mode and the third-learning mode. it can be given as

$$v_{i,d}^1(t+1) = v_{i,d}^2(t+1) + v_{i,d}^3(t+1) \quad (14)$$

Replace velocity with position, and (14) can be denoted by

$$x_i^1(t+1) - x_i^1(t) = (x_i^2(t+1) - x_i^1(t)) + (x_i^3(t+1) - x_i^1(t)) \quad (15)$$

In each iteration, stochastic particles are selected to implement the decomposition, and the additional velocities and positions of subparticles are generated by another two learning modes.

To highlight the diversity of the algorithm, the second-learning mode just remains the items that are conducive to the global exploration of particles in original MOPSO algorithm. The formula of second-learning mode is expressed as

$$\begin{cases} v_{i,d}^2(t+1) = \omega_i(t)v_{i,d}^1(t) + c_1(t)r_1(pbest_{i,d}(t) - x_{i,d}^1(t)) \\ x_{i,d}^2(t+1) = x_{i,d}^1(t) + v_{i,d}^2(t+1) \end{cases} \quad (16)$$

In order to accelerate the convergence and satisfy the parallelogram rule, the third-learning mode is established and the particles are only guided by $gbest$. It can be described as

$$\begin{cases} v_{i,d}^3(t+1) = c_2(t)r_2(gbest_d(t) - x_{i,d}^1(t)) \\ x_{i,d}^3(t+1) = x_{i,d}^1(t) + v_{i,d}^3(t+1) \end{cases} \quad (17)$$

Contrasting (12), (16) and (17), the decomposition of particles not only follows the parallelogram rule, but also can promote the ability of exploration and exploitation by using two additional learning modes. Therefore, the proposed algorithm can achieve a considerable optimization procedure in Stage II. The implementation process of stage II is presented in Table 1.

Remark 3: The decomposition of particles follows the parallelogram rule, which means that the optimization of stage II is based on the stage I, that is, the original MOPSO with adaptive strategy. The particles are split into two parts, corresponding to two learning modes. The decomposed particles are randomly selected, and the additional positions and velocities just remain in this iteration, which can avoid the computation increasing dramatically. The implementation of Stage II makes the algorithm find more nondominated solutions in each iteration, and the added solutions can remarkably improve the convergence and diversity of the proposed algorithm.

3.3.3 Stage III: attenuation mutation

The target of third stage is to expand the exploring scope and avoid the premature convergence and local optimum, which is mainly caused by the fast convergence of MOPSO, and the mutation is considered one of the most effective methods in the current study (Tao et al. 2020). Therefore, the adaptive Gaussian attenuation mutation is rightfully carried out in the third-stage strategy of the proposed tssAMOPSO, which can be expressed as follows,

$$x_{i,d}(t) = \begin{cases} x_{i,d}(t) + rQ_m(t) \exp\left[-\left(\frac{gbest_d(t) - x_{i,d}(t)}{\sigma(t)}\right)^2\right], & r \leq Ap(t) \\ x_{i,d}(t), & \text{Otherwise} \end{cases} \quad (18)$$

where $x_{i,d}$ is the position obtained by stage I or stage II, $gbest$ is decided by (13), and σ is the variance of Gaussian function. The value of σ is determined by $\sigma(t) = \sigma_0 \exp(-(t/T)^2)$, where σ_0 is the initial value of variance, t is the current number of iterations, and T is the maximum number of iterations. $Q_m(t)$ is the maximum allowable range of mutation, which can be given as

$$Q_m(t) = \max_{j \in [1,D]} |x_{i,d}(t) - x_{j,d}(t)| \quad (19)$$

Remark 4: According to the relationship between Ap index and the evolutionary environment described in Section 3.1, the larger value of Ap indicates that the algorithm is in global exploration. Otherwise, the status of local exploitation can be manifested by the small Ap value. Based on the above conclusion and (18), the advantages of the proposed mutation operator can be expressed as follows. 1) The mutation can expand the searching scope in the status of global exploration, and the mutation probability of particles can be determined according to the evolutionary environment of the algorithm, which can save the computing resources occupied by the mutation operator. 2) The variance of Gaussian function (σ) can control the mutation range. With the increase of iterations, the maximum mutation range of particles will be gradually contracted to ensure the exploitation ability of particles in the later iterations, and the influence of mutation on convergence can be decreased further in the status of local exploitation.

3.4 Personal best selection

The selection of $pbest$ is considered as an essential step in MOPSO, which stems from the importance of $pbest$ in the process of optimization (Lee and Kim 2013). In this paper, each particle has a personal repository to store its positions in recent iterations, which forms a memory interval. The memory interval of particles can be described as

$$MI_i(t) = [x_i(t-u), x_i(t-u+1), \dots, x_i(t)] \quad (20)$$

Table 1 Implementation process of Stage II.

$Rst = \text{round}(S * \text{rand});$	// Randomly determine the number of particles that conduct the stage II
$Rp = \text{round}(0.5 * S * (\text{rand}(1, Rst) + 1));$	// Randomly determine the serial number of particles
<i>for</i> $i = 1 : Rp$	
Substitute $v(Rp(i))$ and $x(Rp(i))$ into (16), and Obtain the x_i^2 ;	
Substitute $v(Rp(i))$ and $x(Rp(i))$ into (17), and Obtain the x_i^3 ;	
<i>end</i>	
Calculate the dominated relationship between all searched solutions and exist nondominated solutions in the external archive (Ea);	
Add or replace the nondominated solutions to (in) Ea ;	

where MI_i is the memory interval of i -th particle, and u is the maximum capacity of memory interval.

According to the fusion ranking index FR described in (7), the relationship among the nondominated solutions can be quantified, and the convergence degree of each position in the memory interval can be calculated. The position with the smallest FR in memory interval is selected as the $pbest$,

$$pbest_i(t+1) = \min_{i=1,2,\dots,S} (FR(MI_i(t))) \quad (21)$$

When the memory interval is saturated, it is necessary to maintain the memory interval. Take two particles as an example, the update process of the memory interval is shown in Fig. 5.

Fig. 5 Update process of particle memory interval.

It can be seen from Fig. 5 that there are two situations in updating the memory interval. 1) If the storage location reaches the maximum capacity, the position of the earliest iteration in memory will be forgotten, i.e. $x_i(t-u)$. 2) If $x_i(t-u)$ is the best-ranked position, the adjacent position will be forgotten, i.e. $x_i(t-u+1)$. The mathematical process of the update can be described as follows,

$$MI_i(t+1) = \begin{cases} [x_i(t-u), x_i(t-u+2), \dots, x_i(t+1)], & \text{if } pbest_i(t+1) = x_i(t-u) \\ [x_i(t-u+1), x_i(t-u+2), \dots, x_i(t+1)], & \text{otherwise} \end{cases} \quad (22)$$

Remark 5: Based on the application of memory interval, the reliability of $pbest$ selection can be improved remarkably. Compared with the scheme that just uses Pareto dominance as the standard for $pbest$ selection, which may bring about the algorithm stagnates and fall into the local optimum, the proposed $pbest$ selection can effectively guide the flight direction of particles even if there is no dominant relationship between the particles and $pbest$. Moreover, the use of memory interval enables particles to keep the current optimal position, and the convergence efficiency of the algorithm can be promoted by the reliably selected $pbest$.

3.5 External archive maintenance

Most of MOPSO algorithms use the density index as the criterion to judge the performance of nondominated solutions, and the solutions with the highest density will be deleted when the external archive achieves the maximum capacity. Although this method can maintain the diversity of nondominated solution set, it may delete the solutions with excellent convergence, which would affect the exploitation ability and lead to the population degradation (Lin et al. 2015). Therefore, the density should not be regarded as the only performance index for maintaining the external archive, and the convergence of nondominated solutions should be considered as well.

According to the convergence index in (7) and distribution index in (10), the comprehensive performance of solutions in the external archive is judged by the fusion evaluation index in this paper, which can be presented as

$$CR(s_n) = \frac{FR(s_n)}{LCD(s_n)} \quad (23)$$

The smaller CR index means the better comprehensive performance. Therefore, if the number of nondominated solutions searched by the algorithm exceeds the maximum capacity of the external archive, the nondominated solutions with the largest CR index will be deleted.

Remark 6: Since the maintenance of external archive is directly related to the performance of final nondominated

solution set obtained by the algorithm, it is significant that the proposed *CR* index can greatly improve the performance of MOPSO. Compared with the scheme of the external archive maintenance only decided by the density index, the fusion index *CR* can avoid the population degradation and the false deletion of nondominated solutions with good convergence.

4 Convergence analysis

The convergence analysis is one of the difficulties for MOPSO algorithm. Several pieces of literature (Han et al. 2018; Clerc and Kennedy 2002; Chakraborty et al. 2011) claim that the convergence analysis of MOPSO has been provided, but the relevant process is lack of comprehensiveness evidently. In this Section, the convergence of the proposed tssAMOPSO is analyzed in a progressive way.

4.1 Convergence of stage I

Theorem 1: The positions of particles obtained by the first-learning mode according to (12) will converge to the Pareto optimal solution set (Λ^*). The mathematical expression can be described as follows,

$$\lim_{t \rightarrow \infty} \phi^1(t) = \hat{P}^* \quad (24)$$

where ϕ^1 is the average position of population obtained by the first-learning mode and $\hat{P}^*$ is the average of the Pareto optimal solution set.

Proof. To simplify the calculation, denote

$$\begin{cases} pbest_i(t) = P^* + Yp_i(t) \\ gbest(t) = P^* + Yg_i(t) \end{cases} \quad (25)$$

where P^* is the solution in Λ and $Yp_i(t)$ and $Yg_i(t)$ are the difference vector that represents the distance of *pbest* and *gbest* from P^* respectively.

Based on (25), the velocity of first-learning mode can be presented as

$$\begin{aligned} v_i^1(t+1) &= \omega v_i(t) + c_1 r_1 (P^* + Yp_i(t) - x_i(t)) \\ &\quad + c_2 r_2 (P^* + Yg_i(t) - x_i(t)) \\ \Rightarrow v_i^1(t+1) &= \omega v_i(t) + (c_1 r_1 + c_2 r_2) (P^* - x_i(t)) \\ &\quad + c_1 r_1 Yp_i(t) + c_2 r_2 Yg_i(t) \end{aligned} \quad (26)$$

The relationship between velocity and position can be easily obtained from (14) and (15), and (26) can be rewritten as

$$\begin{aligned} x_i^1(t+1) &= (1 + \omega - (c_1 r_1 + c_2 r_2)) x_i(t) - \omega x_i(t-1) \\ &\quad + (c_1 r_1 + c_2 r_2) P^* + c_1 r_1 Yp_i(t) + c_2 r_2 Yg_i(t) \end{aligned} \quad (27)$$

Then, take the expectations on both sides of (27),

$$\begin{aligned} E(x_i^1(t+1)) &= (1 + \omega - (c_1 E(r_1) + c_2 E(r_2))) E(x_i(t)) \\ &\quad - \omega E(x_i(t-1)) + (c_1 E(r_1) + c_2 E(r_2)) E(P^*) \\ &\quad + c_1 E(r_1) E(Yp_i(t)) + c_2 E(r_2) E(Yg_i(t)) \end{aligned} \quad (28)$$

The expectation of the random number in $[0,1]$ is 0.5. Denote $E(Yp_i(t)) \equiv \hat{Y}p_i(t)$, $E(Yg_i(t)) \equiv \hat{Y}g_i(t)$, and $E(P^*) \equiv \hat{P}^*$, we can obtain

$$\begin{aligned} \phi^1(t+1) &= (1 + \omega - \frac{c_1 + c_2}{2}) \phi(t) - \omega \phi(t-1) \\ &\quad + \frac{c_1 + c_2}{2} \hat{P}^* + \frac{c_1}{2} \hat{Y}p(t) + \frac{c_2}{2} \hat{Y}g(t) \\ \Rightarrow \phi^1(t) &= (1 + \omega - \frac{c_1 + c_2}{2}) \phi(t-1) - \omega \phi(t-2) \\ &\quad + \frac{c_1 + c_2}{2} \hat{P}^* + \frac{c_1}{2} \hat{Y}p(t-1) + \frac{c_2}{2} \hat{Y}g(t-1) \end{aligned} \quad (29)$$

Take the Z transform on both sides of (29),

$$\begin{aligned} Z(\phi^1(t)) &= (1 + \omega - \frac{c_1 + c_2}{2}) Z(\phi(t-1)) \\ &\quad - \omega Z(\phi(t-2)) + Z(\frac{c_1 + c_2}{2} \hat{P}^*) \\ &\quad + \frac{c_1}{2} Z(\hat{Y}p(t-1)) + \frac{c_2}{2} Z(\hat{Y}g(t-1)) \\ \Rightarrow \phi^1(z) &= (1 + \omega - \frac{c_1 + c_2}{2}) z^{-1} \phi(z) - \omega z^{-2} \phi(z) \\ &\quad + \frac{c_1 + c_2}{2} \hat{P}^* \frac{z}{z-1} + \frac{c_1}{2} z^{-1} \hat{Y}p(z) \\ &\quad + \frac{c_2}{2} z^{-1} \hat{Y}g(z) \end{aligned} \quad (30)$$

Since the first-learning mode is similar to the original MOPSO, we can obtain $x^1(t) = x(t)$, i.e. $\phi^1(z) = \phi(z)$, and let $\delta = (1 + \omega - \frac{c_1 + c_2}{2})$. (30) can be rewritten as

$$\begin{aligned} (1 - \delta z^{-1} + \omega z^{-2}) \phi(z) &= \frac{c_1 + c_2}{2} \hat{P}^* \frac{z}{z-1} \\ &\quad + \frac{c_1}{2} z^{-1} \hat{Y}p(z) + \frac{c_2}{2} z^{-1} \hat{Y}g(z) \\ \Rightarrow \phi(z) &= \frac{c_1 + c_2}{2} \frac{1}{(1 - \delta z^{-1} + \omega z^{-2})} \frac{z}{z-1} \hat{P}^* \\ &\quad + \frac{c_1}{2} \frac{z^{-1}}{(1 - \delta z^{-1} + \omega z^{-2})} \hat{Y}p(z) \\ &\quad + \frac{c_2}{2} \frac{z^{-1}}{(1 - \delta z^{-1} + \omega z^{-2})} \hat{Y}g(z) \\ \Rightarrow \phi(z) &= \frac{c_1 + c_2}{2} \frac{z^3}{(z^2 - \delta z + \omega)(z-1)} \hat{P}^* \\ &\quad + \frac{c_1}{2} \frac{z}{(z^2 - \delta z + \omega)} \hat{Y}p(z) \\ &\quad + \frac{c_2}{2} \frac{z}{(z^2 - \delta z + \omega)} \hat{Y}g(z) \end{aligned} \quad (31)$$

According to the final value theorem (Ogata 1995), the conversion of limits between the two domains is $\lim_{t \rightarrow \infty} \phi(t) =$

$\lim_{z \rightarrow 1} (z-1)\phi(z)$. Then,

$$\begin{aligned} \lim_{t \rightarrow \infty} \phi(t) &= \lim_{z \rightarrow 1} \frac{c_1 + c_2}{2} \frac{z^3}{(z^2 - \delta z + \omega)} \hat{P}^* \\ &+ \lim_{z \rightarrow 1} \frac{c_1}{2} \frac{z(z-1)}{(z^2 - \delta z + \omega)} \hat{Y}p(z) \\ &+ \frac{c_2}{2} \frac{z(z-1)}{(z^2 - \delta z + \omega)} \hat{Y}g(z) \end{aligned} \quad (32)$$

Consider the first term of (32), we can get

$$\lim_{z \rightarrow 1} \frac{z^3}{(z^2 - \delta z + \omega)} = \frac{1}{1 - \delta + \omega} = \frac{1}{\frac{c_1 + c_2}{2}} \quad (33)$$

Thus,

$$\lim_{z \rightarrow 1} \frac{c_1 + c_2}{2} \frac{z^3}{(z^2 - \delta z + \omega)} \hat{P}^* = \frac{c_1 + c_2}{2} \hat{P}^* \frac{1}{\frac{c_1 + c_2}{2}} = \hat{P}^* \quad (34)$$

The characteristic equation is denoted by

$$H(z) = z^2 - \delta z + \omega \quad (35)$$

The solutions of (35) are within the unit circle in Z-domain. The required condition can be investigated by *Jury's Test* (Ogata 1995) as follows. *Jury's Test* is a well-known technique in digital control theory, which is used to determine whether the roots of a characteristic equation in Z-plane are in the unit circle, without solving the equation explicitly. Condition for poles within the unit circle from *Jury's Test* can be presented as

- (1) $H(1) > 1$: i.e. $1 - \delta + \omega > 0 \Rightarrow \frac{c_1 + c_2}{2} > 0$
- (2) $(-1)^k H(-1) > 0$: i.e. $1 + \delta + \omega > 0 \Rightarrow \frac{c_1 + c_2}{2} < 2(1 + \omega)$

where k is the maximum degree of $H(z)$. This condition automatically holds for the second degree polynomial (Chakraborty et al. 2011), which means that (35) also satisfies.

- (3) $a_k > |a_0|$: i.e. $\omega < 1$
where a_k and a_0 are the coefficients of the corresponding term in the characteristic equation.

For the other two components, the limiting value depends on the nature of $\hat{Y}p(z)$ and $\hat{Y}g(z)$, and the function to be optimized. If the poles of these terms are within the unit circle, the difference vectors are bounded and not oscillatory in nature with constant magnitude.

Then, these difference vectors can be expanded as

$$\hat{Y}(z) = \sum_{z_j \in \text{Poles}} \frac{C_j z}{z - z_j} \quad (36)$$

Therefore, if $|z_j| < 1$ and condition set of *Jury's Test* is satisfied, this limit exists and the limiting value can be given as

$$\lim_{z \rightarrow 1} \frac{z(z-1)\hat{Y}(z)}{(z^2 - \delta z + \omega)} = \lim_{z \rightarrow 1} \sum_j \frac{C_j z^2(z-1)}{(z^2 - \delta z + \omega)(z - z_j)} = 0 \quad (37)$$

Use the results obtained in (34) and (37), (32) can be simplified as

$$\lim_{t \rightarrow \infty} \phi^1(t) = \hat{P}^* + 0 + 0 = \hat{P}^* \quad (38)$$

By now, the proof of **Theorem 1** has been completed.

4.2 Convergence of stage II

Theorem 2: The positions obtained from the second-learning mode and the third-learning mode will converge to the Pareto optimal solution set. That is,

$$\lim_{t \rightarrow \infty} \phi^2(t) = \lim_{t \rightarrow \infty} \phi^3(t) = \hat{P}^* \quad (39)$$

where ϕ^1 and ϕ^2 are the average position of population obtained by the second-learning mode and the third-learning mode, respectively.

Proof. Based on the convergence analysis of stage I, the position of particles converges to the Pareto optimal set finally. Obviously, $pbest$ and $gbest$ also converge to the Pareto optimal solution set.

$$\lim_{t \rightarrow \infty} pbest(t) = \lim_{t \rightarrow \infty} gbest(t) = \hat{P}^* \quad (40)$$

Thus,

$$\lim_{t \rightarrow \infty} c_1 r_1 (pbest(t) - x(t)) = \lim_{t \rightarrow \infty} c_2 r_2 (gbest(t) - x(t)) = 0 \quad (41)$$

By substituting (41) into (16) and (17), we can obtain

$$\lim_{t \rightarrow \infty} v^3(t+1) = \lim_{t \rightarrow \infty} c_2 r_2 (gbest(t) - x(t)) = 0 \quad (42)$$

$$\lim_{t \rightarrow \infty} v^2(t+1) = \lim_{t \rightarrow \infty} \omega v(t) + c_1 r_1 (pbest(t) - x(t)) = \lim_{t \rightarrow \infty} \omega v(t) \quad (43)$$

Duo to $\omega \in [0, 1]$, when t tends to infinity, we can get

$$\lim_{t \rightarrow \infty} v^2(t+1) = 0 \quad (44)$$

Consider (14) and (15), the relationship between velocity and position of particles obtained by the second-learning mode and the third-learning mode is expressed as follows.

$$\begin{cases} x^2(t+1) = x(t) + v^2(t+1) \\ x^3(t+1) = x(t) + v^3(t+1) \end{cases} \quad (45)$$

Take the limit on both sides of (43),

$$\begin{cases} \lim_{t \rightarrow \infty} \phi^2(t) = \lim_{t \rightarrow \infty} \phi(t) + \lim_{t \rightarrow \infty} v^2(t) = \hat{P}^* \\ \lim_{t \rightarrow \infty} \phi^3(t) = \lim_{t \rightarrow \infty} \phi(t) + \lim_{t \rightarrow \infty} v^3(t) = \hat{P}^* \end{cases} \quad (46)$$

Thus,

$$\lim_{t \rightarrow \infty} \phi^2(t) = \lim_{t \rightarrow \infty} \phi^3(t) = \hat{P}^* \quad (47)$$

The above process completes the proof of **Theorem 2**.

4.3 Convergence of stage III

Theorem 3: The positions obtained from the proposed Gaussian attenuation mutation will converge to the Pareto optimal solution set.

Proof. From (38), we can get that g_{best} would converge to Pareto optimal solution set. Thus,

$$\lim_{t \rightarrow \infty} \exp\left[-\left(\frac{g_{best}(t) - x(t)}{\sigma}\right)^2\right] = 1 \quad (48)$$

Substitute (48) into (18), the position of the mutated particles can be rewritten as

$$\begin{aligned} \lim_{t \rightarrow \infty} x_{i,d}(t) + rQ_m(t) \exp\left[-\left(\frac{g_{best_d}(t) - x_{i,d}(t)}{\sigma}\right)^2\right] \\ = \lim_{t \rightarrow \infty} x_{i,d}(t) + \lim_{t \rightarrow \infty} rQ_m(t) \end{aligned} \quad (49)$$

It can be seen from (??) that the variation of mutated particles mainly depends on Q_m . Since the positions obtained in stage I and stage II converge to the center of Pareto optimal solution set from the (38) and (47), the main effect of the mutation on particles can be presented as

$$\begin{aligned} \lim_{t \rightarrow \infty} Q_m(t) = \lim_{t \rightarrow \infty} \max_{j \in [1,D]} |x_{i,d}(t) - x_{j,d}(t)| \\ \leq \varepsilon \leq (X_{\max,d}^* - X_{\min,d}^*) \end{aligned} \quad (50)$$

where ε is a positive number, which can reflect the maximum difference of d -th dimension decision variables in Pareto optimal solution set.

Finally, the limiting value of positions obtained in stage III is expressed as

$$\lim_{t \rightarrow \infty} x_{i,d}(t) + rQ_m(t) \exp\left[-\left(\frac{g_{best_d}(t) - x_{i,d}(t)}{\sigma}\right)^2\right] = \Lambda^* \quad (51)$$

The proof of **Theorem 3** has been completed.

Remark 7: The convergence of the original MOPSO algorithm is the premise of ensuring the effective optimization in the three stages. It can be seen from (47) that the additional positions generated by the decomposition in stage II are the same as the positions obtained by the first-learning mode in stage I when the algorithm reaches convergence. The convergence analysis of stage III shows that although the mutation will prevent the algorithm from converging to a certain position in the decision space, the Gaussian attenuation mutation proposed in this paper still can make the algorithm converge to the Pareto optimal set eventually.

5 Experimental studies

The experimental studies of the proposed tssAMOPSO employ ZDT and DTLZ benchmark functions to test the performance, and several MOPSO competitors are introduced,

including MOPSO based on metaheuristic (SMPSO) (Nebro et al. 2009), MOPSO based on crowding distance (cd-MOPSO) (Raquel and Naval 2005), adaptive MOPSO based on parallel cell coordinate (pccsAMOPSO) (Hu and Yen 2015), and adaptive MOPSO based on multiple adaptive methods (AMOPSO) (Han et al. 2017). Moreover, for a more comprehensive evaluation, tssAMOPSO is also contrasted with three MOEAs, including NSGA-II (Deb et al. 2002), SPEA2 (Zitzler et al. 2001), and MOEA/D (Zhang and Li 2007). All experiments run on Windows 10 system, CPU frequency is 2.7 GHz, 8GB RAM, and MATLAB 2016b.

To compare with MOPSOs and MOGAs fairly, the same common parameters of all the different algorithms are presented as follows. Both the population size (S) and the maximum capability of the external archive (N) are set to 100, and the maximum number of iterations (T) is fixed at 300. The remaining parameters of tssAMOPSO are set as follows. The parameter n in local crowding distance is set to 4, the parameter σ_0 (the initial value of the variance of Gaussian function in mutation) is set to 0.8, and the parameter u (the particle memory interval length) is set to 5.

5.1 Performance metrics

To accurately measure the convergence, diversity, and computational complexity of the proposed algorithms, the following quantitative performance metrics are employed for demonstrating the different aspects of performance, respectively.

(1) *Inverted Generational Distance (IGD)*: IGD is generally used to measure the distance between the real Pareto front and the nondominant solution set obtained by the algorithm. The smaller the IGD value means that the nondominant solution set is closer to the real Pareto front. The calculation can be expressed as

$$IGD(PF, PF^*) = \frac{1}{N} \sum_{i=1}^N \min(\text{dist}(PF_i, PF^*)) \quad (52)$$

where PF is the approximate Pareto front obtained by the algorithms, PF^* is a set of sampling points from the true Pareto front, and N is the number of nondominant solutions.

(2) *Spacing (SP)*: SP is usually employed to reflect the range difference between neighboring solutions. The smaller SP value represents the more uniform distribution of the nondominant solution set. It can be given as

$$SP = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (\bar{d} - d_i)^2} \quad (53)$$

where d_i is the minimum Euclidean distance between the i -th nondominant solution and others, and $\bar{d}$ is the mean of all d_i .

Fig. 6 Nondominated solutions obtained by different algorithms on ZDT4.

Fig. 7 Nondominated solutions obtained by different algorithms on ZDT4.

(3) *Computational time*: Computational time can evaluate the computational complexity of the algorithms. The real-time performance of the algorithms can be indirectly explained by the computational time under the same environmental platform.

5.2 Experimental results

5.2.1 ZDT test problems

(1) *Comparison of IGD index*: The proposed tssAMOPSO and other five contrasted algorithms for solving ZDT3, ZDT4, and ZDT6 problems are shown in Fig. 6 - Fig. 8 re-

spectively. It can be seen that all algorithms can approach to the Pareto front on ZDT3 series. However, some algorithms perform poorly on the problems that are difficult to converge (ZDT4 and ZDT6), such as SPEA2 on ZDT4 problem and NSGA-II on ZDT6 problem.

The experimental results of the proposed algorithm and the competitors are listed in Table 2. The mean value of IGD metric obtained by SMPSO, pccsAMOPSO, AMOPSO, MOEA/D, and the proposed tssAMOPSO acquire lower than 10^{-3} on all ZDT problems, which means that those algorithms perform signally convergence on ZDT problems. The other three algorithms (cdMOPSO, NSGA-II, and SPEA2) also are good at some problems. cdMOPSO rep-

Fig. 8 Nondominated solutions obtained by different algorithms on ZDT6.

Table 2 IGD comparison results on ZDT test problems (all results were averaged on 30 independent runs).

Function	Index	MOPSOs					MOEAs		
		tssAMOPSO	SMPSO	cdMOPSO	pccsAMOPSO	AMOPSO	NSGA-II	SPEA2	MOEA/D
ZDT1	Mean	5.696E-04	3.68E-03	4.241E-03	4.013E-03	4.009E-03	5.755E-03	4.148E-03	4.032E-03
	std	4.553E-05	2.61E-05	2.580E-04	6.281E-05	5.700E-05	3.390E-04	1.770E-04	5.589E-05
	Rank	1	2-	7-	4-	3-	8-	6-	5-
ZDT2	Mean	4.176E-04	3.81E-03	4.280E-03	4.091E-03	1.983E-03	5.355E-03	4.100E-03	3.845E-03
	std	1.624E-05	3.21E-05	1.140E-04	4.815E-05	5.200E-05	2.020E-04	2.56E-04	4.434E-05
	Rank	1	3-	7-	5-	2-	8-	6-	4-
ZDT3	Mean	1.102E-03	4.48E-03	3.863E-03	3.323E-03	3.982E-03	5.834E-03	3.161E-03	8.424E-03
	std	8.585E-05	2.53E-04	7.130E-05	9.951E-05	3.623E-03	2.020E-04	0.000E+0	7.019E-03
	Rank	1	6-	4-	3-	5-	7-	2-	8-
ZDT4	Mean	2.601E-03	3.77E-03	5.912E+0	7.971E-03	4.147E-03	1.655E-02	2.495E+1	4.865E-03
	std	2.992E-04	4.44E-05	4.518E+0	1.470E-03	2.980E-04	3.174E-02	7.252E+0	8.412E-03
	Rank	1	2=	7-	5-	3-	6-	8-	4-
ZDT6	Mean	2.886E-04	3.03E-03	2.988E-03	3.406E-03	1.754E-04	1.286E-02	5.323E-03	3.992E-03
	std	3.313E-05	1.93E-04	4.543E-04	2.287E-04	8.600E-05	1.004E-03	2.656E-03	6.016E-05
	Rank	2	4-	3-	5-	1=	8-	7-	6-
Rank Sum		6	17	28	22	14	37	29	27
Final Rank		1	3	6	4	2	8	7	5
better/worse/similar		—	0/4/1	0/5/0	0/5/0	0/4/1	0/5/0	0/5/0	0/5/0

"+", "-", and "=" demonstrate that the results of other algorithms are superior, inferior, and similar to the ones obtained by tssAMOPSO respectively, which is judged by Wilcoxon's rank-sum test with the p-value=0.05.

resents satisfactory convergence on ZDT3 and ZDT6, and SPEA2 shows pleasing performance on ZDT3. The proposed tssAMOPSO outperforms other algorithms on ZDT1 and ZDT2. For ZDT3 problems, tssAMOPSO gains the optimal mean value, while cdMOPSO gets the optimal std value. The proposed tssAMOPSO performs similar performance to SMPSO on ZDT4, and one of the two algorithms obtains the optimal IGD mean value and the other obtains the

optimal IGD std value. AMOPSO achieves the best performance on ZDT6 (AMOPSO obtains the optimal IGD mean value, and tssAMOPSO obtains the optimal IGD std value). The third indicator (Rank) is calculated by Wilcoxon's rank-sum test, which is used to express the degree of performance intuitively. The 'Rank Sum' and 'Final Rank' indicate the overall performance of the algorithms on ZDT problems. We can easily find that tssAMOPSO gains the best

Fig. 9 Box-plots of the IGD results obtained by different algorithms on ZDT test functions.

Fig. 10 Box-plots of the SP results obtained by different algorithms on ZDT test functions.

ranking with an obvious advantage. AMOPSO and SMPSO occupy the second and third ranking respectively, and the two algorithms perform similarly on ZDT test problems (Rank Sum of AMOPSO and SMPSO is 14 and 17 respectively). Only NSGA-II performs poorly on ZDT problems and it is also the only algorithm that achieves over 30 on the Rank Sum. The last row in Table 2 represents the competitor better than, worse than, and similar to the proposed tssAMOPSO algorithm. From the last row, we can clearly see that tssAMOPSO is superior to all contrasted algorithms.

To further illustrate the convergence of different algorithms on ZDT problems, the box-plots of repetitive experimental results on the IGD performances are presented in Fig. 9. Compared with other algorithms, tssAMOPSO shows remarkably convergence accuracy in solving all the ZDT test problems. Comprehensively, the box-plots of the IGD metric indicate that the tssAMOPSO algorithm scarcely appears significant skewness, and the maximum deviation of IGD surpasses other contenders. The experiment results demonstrate that the proposed tssAMOPSO has visible advantages over all compared algorithms in terms of the convergence accuracy and stability.

(2) Comparison of SP index: As can be seen from Fig. 6 - Fig. 8 that the nondominated solutions obtained by some compared algorithms cannot cover entire Pareto fronts. In particular, MOEA/D solves ZDT3 and NSGA-II solves ZDT4, which clearly show that the covered region does not include the lower part of the Pareto front. SP metric can directly represent the distribution of nondominated solutions. Table 3 shows the detailed SP comparison of the proposed tssAMOPSO and seven contenders. The mean SP value of tssAMOPSO, pccsAMOPSO, AMOPSO, NSGA-II, and MOEA/D are lower than 10^{-2} on all ZDT problems. Moreover, all the mean SP values of tssAMOPSO are all lower than 10^{-3} . It is obvious from the experimental results that tssAMOPSO outperforms other algorithms on all ZDT problems, and tssAMOPSO basically achieves the optimal mean and std value (except that pccsAMOPSO gains the optimal std value on ZDT1 and AMOPSO obtains the optimal std value on ZDT6). Furthermore, from the ranking of each algorithm, tssAMOPSO obtains the first place on all five ZDT problems. MOEA/D has obtained excellent SP performance on ZDT1 and ZDT2, which is similar to tssAMOPSO. In addition, some algorithms also achieve the first-class experimental results on some problems. MOEA/D

Table 3 SP comparison results on ZDT test problems (all results were averaged on 30 independent runs).

Function	Index	MOPSOs					MOEAs		
		tssAMOPSO	SMPSO	cdMOPSO	pccsAMOPSO	AMOPSO	NSGA-II	SPEA2	MOEA/D
ZDT1	Mean	3.402E-03	7.50E-02	8.561E-02	2.574E-02	2.554E-02	5.830E-02	3.734E-02	4.851E-03
	std	3.215E-04	1.08E-02	1.425E-02	3.164E-04	6.710E-04	9.385E-03	2.674E-03	7.159E-04
	Rank	1	7-	8-	4-	3-	6-	5-	2=
ZDT2	Mean	3.200E-03	7.33E-02	1.907E-02	2.322E-02	2.257E-02	7.241E-03	1.093E-02	4.366E-03
	std	1.870E-04	1.34E-02	6.420E-04	1.987E-03	3.245E-03	7.410E-04	1.040E-03	7.451E-04
	Rank	1	8-	4-	6-	5-	7-	3-	2=
ZDT3	Mean	3.804E-03	7.182E-01	3.568E-01	7.159E-02	7.025E-02	9.222E-02	6.072E-01	8.424E-03
	std	1.834E-04	3.69E-02	2.247E-01	4.054E-03	7.141E-03	8.415E-03	1.026E+0	9.33E-03
	Rank	1	8-	6-	4-	3-	5-	7-	2-
ZDT4	Mean	3.401E-03	8.88E-02	2.046E-01	2.881E-02	2.645E-02	3.838E-02	1.390E-01	7.527E-02
	std	2.748E-04	1.03E-02	9.556E-02	5.132E-04	4.998E-04	3.837E-03	4.951E-02	6.943E-03
	Rank	1	6-	8-	3-	2-	4-	7-	5-
ZDT6	Mean	2.601E-03	3.033E-01	3.457E-02	4.024E-02	3.651E-02	8.266E-03	1.448E-02	1.884E-02
	std	3.023E-04	4.503E-01	3.884E-03	2.523E-03	1.363E-04	9.180E-04	5.168E-03	5.522E-03
	Rank	1	8-	5-	7-	6-	2-	3-	4-
Rank Sum		5	37	31	24	19	24	25	15
Final Rank		1	8	6	5	3	4	7	2
better/worse/similar		—	0/5/0	0/5/0	0/5/0	0/5/0	0/5/0	0/5/0	0/3/2

"+", "-", and "=" demonstrate that the results of other algorithms are superior, inferior, and similar to the ones obtained by tssAMOPSO respectively, which is judged by Wilcoxon's rank-sum test with the p-value=0.05.

obtains the second rank on ZDT3, cdMOPSO gains the second rank on ZDT4, and NSGA-II ranks the second on ZDT6. 'Rank Sum' shows that tssAMOPSO, MOEA/D, and AMOPSO perform the best, second, and third ranking diversity on ZDT problems, respectively. The analysis of 'Final Rank' and 'better/worse/similar' shows that only MOEA/D achieves two similar results to tssAMOPSO (on ZDT1 and ZDT2), and other algorithms are distinctly worse than the proposed tssAMOPSO algorithm.

To intuitively exhibit the diversity of various algorithms, the box-plots of SP results are plotted in Fig. 10. The box-plots show that tssAMOPSO is evidently superior to other algorithms on ZDT2 and ZDT6. The performance of tssAMOPSO and MOEA/D in solving ZDT1 and ZDT3 problems are in the same level, and tssAMOPSO represents similar diversity to pccsAMOPSO on ZDT4. The experimental results in Fig. 10 indicate that tssAMOPSO has the smallest SP metric, and the box distances are also the least, which means that the nondominated solution set obtained by the proposed tssAMOPSO algorithm has excellent distributivity and stable performance.

5.2.2 DTLZ test problems

(1) Comparison of IGD index: Based on the experimental studies above, the proposed tssAMOPSO has exhibited outstanding performance on ZDT problems. However, ZDT instances are just two-objective problems. For the fur-

ther experimental research of tssAMOPSO, DTLZ benchmark problems are employed to test the performance of tssAMOPSO in solving three-objective problems. Fig. 11 and Fig. 12 show the experimental results of several algorithms on DTLZ2 and DTLZ7 respectively. DTLZ2 and DTLZ7 problems possess obvious characteristics that DTLZ2 has many local Pareto fronts and DTLZ7 is discontinuous. As shown in Fig. 11 and Fig. 12, the nondominated solutions obtained by tssAMOPSO can perfectly converge to the Pareto front, which is undoubtedly superior to other algorithms.

The comparison of IGD metric between the proposed tssAMOPSO and several competitors on DTLZ problems is listed in Table 4. According to the 'Rank' index of the experimental results, tssAMOPSO obtains the first place on five test problems (DTLZ2, DTLZ3, DTLZ4, DTLZ5, and DTLZ7). SMPSO slightly surpasses tssAMOPSO on DTLZ1, and it performs similar to tssAMOPSO on DTLZ3 and DTLZ6 problems. In particular, tssAMOPSO, cdMOPSO, and pccsAMOPSO have similar excellent performance on DTLZ5 problem. The 'Final Rank' index indicates that SMPSO obtains the second-class performance only after tssAMOPSO, and AMOPSO and pccsAMOPSO also acquire outstanding performance on DTLZ problems. The box-plots of 30 independent experimental results on IGD metric are presented in Fig. 13. Compared with other algorithms, the proposed tssAMOPSO performs remarkably performance on DTLZ2, DTLZ4, and DTLZ7 problems.

Fig. 11 Nondominated solutions obtained by different algorithms on DTLZ2.

Fig. 12 Nondominated solutions obtained by different algorithms on DTLZ7.

Comprehensively, the box-plots show that tssAMOPSO algorithm is superior to other contenders, and scarcely appears significant skewness. This means that compared with other algorithms, the proposed tssAMOPSO algorithm has obvious advantages in convergence accuracy and stability in terms of experimental results on DTLZ problems.

(2) Comparison of SP index: The distribution of non-dominated solutions can directly reflect the diversity of the algorithms. Therefore, from the experimental results of DTLZ2 and DTLZ7, we can clearly evaluate the distribu-

tion of the nondominated solutions obtained by various algorithms. It can be observed from Fig. 11 and Fig. 12 that the nondominated solutions obtained by tssAMOPSO algorithm have the admirable distribution, and all the contenders show insufficient diversity to some extent.

The comparison of SP metric between the proposed tssAMOPSO and several competitors is shown in Table 5. For the three-objective optimization problems, it is difficult to balance the convergence and diversity of the algorithms. Although several algorithms can achieve acceptable

Table 4 IGD comparison results on DTLZ test problems (all results were averaged on 30 independent runs).

Function	Index	MOPSOs					MOEAs		
		tssAMOPSO	SMPSO	cdMOPSO	pccsAMOPSO	AMOPSO	NSGA-II	SPEA2	MOEA/D
DTLZ1	Mean	2.056E-02	1.28E-02	2.754E+1	1.360E-01	1.905E-01	1.417E-01	3.773E+1	6.040E-01
	std	1.124E-04	3.82E-04	9.342E+0	1.116E-01	2.386E-03	6.656E+0	1.450E-04	2.895E-01
	Rank	2	1+	7-	3-	5-	4-	8-	6-
DTLZ2	Mean	5.152E-03	3.12E-02	1.015E-01	6.143E-02	6.595E-02	1.059E-01	8.219E-02	6.243E-01
	std	5.440E-04	7.21E-04	1.343E-03	1.898E-03	7.241E-04	8.383E-03	2.833E-03	6.439E-05
	Rank	1	2-	6-	3-	4-	7-	5-	8-
DTLZ3	Mean	2.357E-02	3.10E-02	4.455E+1	2.190E-01	4.025E-02	1.643E+1	4.876E+1	6.517E-01
	std	2.462E-03	5.73E-04	1.017E+1	2.856E-01	7.126E-03	7.565E+0	0.000E+0	2.954E-01
	Rank	1	2=	7-	4-	3-	6-	8-	5-
DTLZ4	Mean	5.364E-03	2.60E-02	1.023E-01	4.405E-02	1.821E-02	7.306E-02	7.290E-02	2.702E-01
	std	4.094E-04	6.46E-03	3.660E-02	3.374E-03	1.012E-02	5.093E-02	1.416E-03	6.835E-03
	Rank	1	3-	7-	4-	2-	6-	5-	8-
DTLZ5	Mean	2.467E-03	8.16E-03	6.053E-03	6.618E-03	7.592E-03	8.052E-03	1.412E-02	5.942E-01
	std	8.113E-05	2.97E-04	7.293E-04	5.564E-04	3.125E-04	1.632E-03	3.541E-03	8.427E-05
	Rank	1	6-	2=	3=	4-	5-	7-	8-
DTLZ6	Mean	1.949E-03	7.87E-04	5.228E-03	5.041E-03	3.262E-03	1.472E+0	2.495E-01	6.165E-01
	std	5.714E-04	4.16E-05	3.904E-04	2.277E-04	1.905E-03	6.091E-01	5.666E-02	1.006E-04
	Rank	2	1=	5-	4-	3-	8-	6-	7-
DTLZ7	Mean	8.521E-03	3.35E-02	2.988E-02	4.277E-02	1.215E-02	6.136E-01	6.214E-02	6.569E-01
	std	8.373E-04	1.53E-03	1.651E-03	9.506E-04	8.600E-04	1.157E-01	0.000E+0	9.869E-04
	Rank	1	4-	3-	5-	2-	7-	6-	8-
Rank Sum		9	19	37	26	23	43	45	50
Final Rank		1	2	5	4	3	6	7	8
better/worse/similar		—	1/4/2	0/6/1	0/6/1	0/7/0	0/7/0	0/7/0	0/7/0

“+”, “-” and “=” demonstrate that the results of other algorithms are superior, inferior, and similar to the ones obtained by tssAMOPSO respectively, which is judged by Wilcoxon’s rank-sum test with the p-value=0.05.

Fig. 13 Box-plots of the IGD results obtained by different algorithms on DTLZ test functions.

convergence from Table 4, they are evidently insufficient in terms of the diversity. The experimental results show that tssAMOPSO algorithm basically outperforms all the contenders (except cdMOPSO on DTLZ5), and tssAMOPSO performs the same as cdMOPSO and SPEA2 on DTLZ3 problem. Only the proposed tssAMOPSO algorithm can achieve the degree of 10^{-2} in all 7 test instances. The

‘better/worse/similar’ index indicates that cdMOPSO has achieved the quite decent results on DTLZ series problem. The main reason is that cdMOPSO uses crowding distance to maintain the external archive, which can effectively filter out the high-density nondominated solutions and facilitate the performances of diversity.

Table 5 SP comparison results on DTLZ test problems (all results were averaged on 30 independent runs).

Function	Index	MOPSOs					MOEAs		
		tssAMOPSO	SMPSO	cdMOPSO	pccsAMOPSO	AMOPSO	NSGA-II	SPEA2	MOEA/D
DTLZ1	Mean	1.004E-02	7.095E-01	4.12E-02	5.791E+0	6.731E-01	5.140E-01	4.910E-01	9.832E-01
	std	1.686E-03	3.84E-02	5.64E-03	4.83E-01	5.142E-02	3.908E-02	3.410E-02	2.332E-01
	Rank	1	6-	2-	8-	5-	4-	3-	7-
DTLZ2	Mean	3.462E-02	6.210E-01	3.562E-01	3.411E-01	2.399E-01	4.162E-01	5.015E-01	8.141E-02
	std	3.821E-03	3.30E-02	1.772E-02	1.242E-02	9.732E-03	3.655E-02	2.216E-02	8.494E-03
	Rank	1	8-	6-	4-	3-	7-	5-	2-
DTLZ3	Mean	5.954E-02	6.320E-01	5.885E-02	6.094E-01	9.049E-02	8.261E-02	6.345E-02	1.796E-01
	std	7.502E-03	3.08E-02	3.926E-03	7.451E-04	1.725E-02	2.095E-03	5.122E-03	3.364E-02
	Rank	2	8-	1=	7-	5-	4-	3=	6-
DTLZ4	Mean	3.872E-02	6.376E-01	4.765E-02	5.977E-01	6.140E-01	4.014E-01	3.421E-01	9.252E-02
	std	5.621E-03	3.08E-02	2.842E-03	6.332E-02	4.01E-02	0.341E-02	2.44E-02	5.615E-03
	Rank	1	8-	2=	6-	7-	5-	4-	3-
DTLZ5	Mean	4.104E-02	1.398E-01	2.422E-02	1.021E-01	3.372E-02	6.379E-02	3.152E-02	8.045E-02
	std	9.002E-03	3.64E-02	6.223E-03	3.744E-02	6.525E-03	1.747E-03	5.317E-03	4.391E-03
	Rank	4	8-	1+	7-	3=	5=	2+	6=
DTLZ6	Mean	1.827E-02	1.042E-01	4.656E-02	5.862E-02	6.097E-02	7.023E-02	5.114E-02	3.935E-02
	std	6.631E-03	2.22E-02	2.738E-03	1.265E-04	7.742E-03	1.567E-02	4.892E-03	3.315E-03
	Rank	1	8-	3=	5-	6-	7-	4-	2=
DTLZ7	Mean	4.029E-02	7.390E-01	5.972E-01	3.988E-01	3.758E-01	4.191E-01	2.964E-01	1.854E-01
	std	5.161E-03	9.56E-02	2.133E-01	8.417E-03	7.593E-03	7.961E-03	5.29E-03	7.935E-02
	Rank	1	8-	7-	5-	4-	6-	3-	2-
Rank Sum		10	54	22	42	33	38	24	28
Final Rank		1	8	2	7	5	6	3	4
better/worse/similar		—	0/7/0	1/3/3	0/7/0	0/6/1	0/6/1	1/5/1	0/5/2

“+”, “-” and “=” demonstrate that the results of other algorithms are superior, inferior, and similar to the ones obtained by tssAMOPSO respectively, which is judged by Wilcoxon’s rank-sum test with the p-value=0.05.

Fig. 14 Box-plots of the SP results obtained by different algorithms on DTLZ test functions.

To further verify the differences of SP values obtained by different algorithms, the box-plots of repetitive experimental results on SP metric are presented in Fig. 14. The box-plots of SP metric show that tssAMOPSO has the smallest SP value and the box distances are also the least. It is obvious that the proposed tssAMOPSO surpasses all contenders on DTLZ2, DTLZ6, and DTLZ7, and also shows the

excellent SP value and stability on other problems. Therefore, tssAMOPSO is superior to the other algorithms on the performance of diversity. The experimental results indicate that the nondominated solution set obtained by the proposed tssAMOPSO algorithm shows distinguished distributivity.

Finally, the ‘Rank’ of various algorithms on all test instances are presented in Table 6. The ‘Rank Sum on

Table 6 Final rank of all the algorithms on ZDTs and DTLZs.

Problems	Algorithms							
	tssAMOPSO	SMPSO	cdMOPSO	pccsAMOPSO	AMOPSO	NSGA-II	SPEA2	MOEA/D
Rank sum on ZDTs	11	54	59	46	33	61	54	42
Rank sum on DTLZs	19	73	59	68	56	81	69	78
Total Rank sum on all the problems	30	127	118	114	89	142	123	120
Final Rank on all the problems	1	7	4	3	2	8	6	5

Since the performance of AMOPSO on some test problems is unknown, the Final Rank is obtained by the average of Rank.

ZDTs' index means the optimization ability in solving two-objective problems of the algorithms. Similarly, the 'Rank Sum on DTLZs' index reflects the optimization ability in solving three-objective problems of the algorithms. The 'Total Rank' and 'Final Rank' indicate that the comprehensive performance of the proposed tssAMOPSO and seven competitors. As can be seen from Table 6 that tssAMOPSO obtains the first place and it is much better than the second-ranked AMOPSO. cdMOPSO and pccsAMOPSO achieved the third and fourth positions with the acceptable results respectively, and the remaining algorithms perform the undistinguished comprehensive performance on all test instances.

5.2.3 Comparison of computational time

It is usually difficult to design a reasonable experiment to explain the computational complexity or computational time of the algorithms. Many methods have been proposed in Qiao et al. (2020), Hu and Yen (2015), and Rajani et al. (2020), and a general and simple one is adopted in this paper, that is, the running time of the algorithms under the same environmental platform and the same maximum number of iterations. On the basis of comparing the convergence and diversity, the running time of each algorithm is measured simultaneously to fairly demonstrate the computational complexity of the proposed tssAMOPSO algorithm. The comparison of different algorithms for the computational time is presented in Table 7.

Obviously, the experimental results show that tssAMOPSO is superior to other algorithms, and tssAMOPSO performs much shorter computational time than other algorithms on 8 problems in all 12 test instances. On ZDT series, tssAMOPSO is only slightly inferior to cdMOPSO on ZDT1 problem. And tssAMOPSO is equally excellent at DTLZ test functions and obtains the optimal performance on four problems (DTLZ2, DTLZ5, DTLZ6, and DTLZ7), and inferior results just occur on DTLZ1, DTLZ3, and DTLZ4. The primary reason for the experimental results is that the three-stage strategy adopted by tssAMOPSO greatly improves the search efficiency of the algorithm. Moreover, the addition of memory interval makes the selected *pbest* more reliable, and the exploration

and exploitation process of the algorithm has been arranged appropriately by the adaptive strategy, which can effectively save the optimization time.

6 Conclusion

A novel adaptive MOPSO using the three-stage strategy, named tssAMOPSO, is proposed in this paper, which has made a dramatic advance in balancing the exploration and exploitation process and promoting the convergence and diversity of MOPSO. By using the dual *gbest* (*c-gbest* and *d-gbest*), the optimization direction of the algorithm can be adjusted flexibly, thereby reducing the difficulty of balancing exploration and exploitation. Furthermore, the adaptive flight parameter adjustment is worked out by the convergence contribution of nondominated solutions to promote the convergence and diversity. The population will conduct the three-stage strategy of optimization in each iteration. Firstly, the adaptive strategy is employed by the particles. Then, some particles are decomposed into two new subparticles, which are adopted two new learning modes to update the velocity and position. Finally, some particles are randomly selected for the Gaussian attenuation mutation. The searched potential solutions of the algorithm in each iteration can be improved vastly by the three-stage strategy. Moreover, each particle has a memory interval to record the recent positions, and *pbest* is selected by the fusion convergence index, which greatly elevates the reliability of *pbest* selection. The fusion index, combined with the convergence and diversity of solutions, is used to maintain the external archive, which can enhance the quality of nondominated solutions.

Based on the improvements above, the convergence and diversity of tssAMOPSO can be balanced perfectly, and the ability of the exploration and exploitation is strengthened signally. The performance of tssAMOPSO is verified by comprehensive experiments and compared with seven kinds of MOPSOs and MOGAs. Experimental results demonstrate that the proposed tssAMOPSO algorithm performs superiorly in contrast to several competitors on the convergence, diversity, and computational complexity.

Table 7 Comparison of different algorithms for the computational time (all results were averaged on 30 independent runs).

Index	Functions	MOPSOs				MOEAs		
		tssAMOPSO	cdMOPSO	pccsAMOPSO	AMOPSO	NSGA-II	SPEA2	MOEA/D
Computational times(s)	ZDT1	102.14	101.72	121.95	110.64	129.17	135.23	152.45
	ZDT2	103.64	112.74	108.34	124.58	133.15	126.65	139.13
	ZDT3	108.52	135.81	124.57	134.37	132.40	137.98	135.02
	ZDT4	112.89	129.69	122.19	132.92	125.67	130.61	147.61
	ZDT6	120.20	124.23	126.62	128.11	133.52	128.74	149.55
	DTLZ1	210.72	226.47	219.83	208.83	248.13	232.47	280.73
	DTLZ2	211.36	212.73	215.63	233.02	250.26	238.69	289.36
	DTLZ3	218.33	217.51	218.26	246.27	250.98	242.81	281.75
	DTLZ4	220.65	220.98	216.44	235.15	247.37	241.33	276.84
	DTLZ5	218.28	228.21	221.92	231.62	255.91	250.62	286.17
	DTLZ6	214.39	225.40	225.73	230.83	245.69	247.97	288.33
	DTLZ7	211.54	224.64	232.41	237.97	246.38	238.11	295.42

Acknowledgements This research was supported by National Natural Science Foundation of China under Grants [61703145]; Scientific and technological innovation team of colleges and universities in Henan Province under Grants [20IRTSTHN019]. In addition, we are grateful to the anonymous reviewers and editors for their valuable suggestions and comments on the initial version of the manuscript.

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this work.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Author Contributions

W. M. Huang and W. Zhang jointly put forward the conception and design of this study. W. M. Huang designed and completed the experiment. Both W. M. Huang and W. Zhang participated in the analysis and interpretation of data. W. M. Huang drafted the first edition of the manuscript. W. Zhang revised the manuscript critically for important intellectual content. W. M. Huang and W. Zhang approved the final version of the manuscript together.

References

- Bolufé-Röhler A, Chen S (2013) Minimum population search - lessons from building a heuristic technique with two population members. In: 2013 IEEE Congress on Evolutionary Computation, pp 2061–2068, DOI 10.1109/CEC.2013.6557812
- Brockhoff D, Zitzler E (2009) Objective reduction in evolutionary multiobjective optimization: Theory and applications. *Evolutionary Computation* 17(2):135–166, PMID: 19413486

- Chakraborty P, Das S, Roy GG, Abraham A (2011) On convergence of the multi-objective particle swarm optimizers. *Information Sciences* 181(8):1411–1425
- Clerc M, Kennedy J (2002) The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation* 6(1):58–73, DOI 10.1109/4235.985692
- Coello CAC, Lechuga MS (2002) Mopso: a proposal for multiple objective particle swarm optimization. In: *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, vol 2, pp 1051–1056 vol.2, DOI 10.1109/CEC.2002.1004388
- Coello CAC, Pulido GT, Lechuga MS (2004) Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation* 8(3):256–279, DOI 10.1109/TEVC.2004.826067
- Dai C, Wang Y, Ye M (2015) A new multi-objective particle swarm optimization algorithm based on decomposition. *Information Sciences* 325:541–557
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation* 6(2):182–197
- Fakhouri HN, Hudaib A, Sleit A (2020) Multivector particle swarm optimization algorithm. *Soft Computing* 24(15):11695–11713
- Fonseca CM, Fleming PJ (1998) Multiobjective optimization and multiple constraint handling with evolutionary algorithms. i. a unified formulation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 28(1):26–37
- Gong M, Cai Q, Chen X, Ma L (2014) Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition. *IEEE Transactions on Evolutionary Computation* 18(1):82–97, DOI 10.1109/TEVC.2013.2260862
- Han H, Lu W, Qiao J (2017) An adaptive multiobjective particle swarm optimization based on multiple adaptive methods. *IEEE Transactions on Cybernetics* 47(9):2754–2767, DOI 10.1109/TCYB.2017.2692385
- Han H, Lu W, Zhang L, Qiao J (2018) Adaptive gradient multiobjective particle swarm optimization. *IEEE Transactions on Cybernetics* 48(11):3067–3079, DOI 10.1109/TCYB.2017.2756874
- Hu W, Yen GG (2015) Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system. *IEEE Transactions on Evolutionary Computation* 19(1):1–18, DOI 10.1109/TEVC.2013.2296151

- Huang H, Lv L, Ye S, Hao Z (2019) Particle swarm optimization with convergence speed controller for large-scale numerical optimization. *Soft Computing - A Fusion of Foundations, Methodologies and Applications* 23:4421–4437
- Jara EC (2014) Multi-objective optimization by using evolutionary algorithms: The p -optimality criteria. *IEEE Transactions on Evolutionary Computation* 18(2):167–179
- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol 4, pp 1942–1948 vol.4, DOI 10.1109/ICNN.1995.488968
- Kumar RS, Kondapaneni K, Dixit V, Goswami A, Thakur LS, Tiwari MK (2015) Multi-objective modeling of production and pollution routing problem with time window: A self-learning particle swarm optimization approach. *Computers and Industrial Engineering* p S0360835215002879, DOI 10.1016/j.cie.2015.07.003
- Lee K, Kim J (2013) Multiobjective particle swarm optimization with preference-based sort and its application to path following footstep optimization for humanoid robots. *IEEE Transactions on Evolutionary Computation* 17(6):755–766
- Leong W, Yen GG (2008) Pso-based multiobjective optimization with dynamic population size and adaptive local archives. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 38(5):1270–1293, DOI 10.1109/TSMCB.2008.925757
- Li C, Yang S, Nguyen TT (2012) A self-learning particle swarm optimizer for global optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 42(3):627–646, DOI 10.1109/TSMCB.2011.2171946
- Li H, Zhang Q (2009) Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii. *IEEE Transactions on Evolutionary Computation* 13(2):284–302, DOI 10.1109/TEVC.2008.925798
- Li L, Wang W, Li W, Xu X, Zhao Y (2016) A novel ranking-based optimal guides selection strategy in mopso. *Procedia Computer Science* 91:1001 – 1010, promoting Business Analytics and Quantitative Management of Technology: 4th International Conference on Information Technology and Quantitative Management (ITQM 2016)
- Lin Q, Li J, Du Z, Chen J, Ming Z (2015) A novel multi-objective particle swarm optimization with multiple search strategies. *European Journal of Operational Research* 247(3):732 – 744
- Liu J, Liu J (2019) Applying multi-objective ant colony optimization algorithm for solving the unequal area facility layout problems. *Applied Soft Computing* 74:167 – 189
- Maghawry A, Hodhod R, Omar Y, Kholief M (2020) An approach for optimizing multi-objective problems using hybrid genetic algorithms. *Soft Computing* DOI 10.1007/s00500-020-05149-3
- Martín-Moreno R, Vega-Rodríguez MA (2018) Multi-objective artificial bee colony algorithm applied to the bi-objective orienteering problem. *Knowledge-Based Systems* 154:93 – 101
- Meza J, Espitia H, Montenegro C, Giménez E, González-Crespo R (2017) Movpso: Vortex multi-objective particle swarm optimization. *Applied Soft Computing* 52:1042 – 1057
- Moubayed NA, Petrovski A, Mccall J (2014) D2mopso: Mopso based on decomposition and dominance with archiving using crowding distance in objective and solution spaces. *Evolutionary Computation* 22(1):47–77, DOI 10.1162/EVCO.a.00104
- Nebro AJ, Durillo JJ, Garcia-Nieto J, Coello Coello CA, Luna F, Alba E (2009) Smpso: A new pso-based metaheuristic for multi-objective optimization. In: *2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making(MCDM)*, pp 66–73, DOI 10.1109/MCDM.2009.4938830
- Ogata K (1995) *Discrete-Time Control Systems*, Second Edition. Prentice-Hall
- Peng W, Zhang Q (2008) A decomposition-based multi-objective particle swarm optimization algorithm for continuous optimization problems. In: *2008 IEEE International Conference on Granular Computing*, pp 534–537, DOI 10.1109/GRC.2008.4664724
- Qiao J, Li F, Yang S, Yang C, Li W, Gu K (2020) An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection. *Information Sciences* 512:446 – 470
- Qu B, Li C, Liang J, Yan L, Yu K, Zhu Y (2020) A self-organized speciation based multi-objective particle swarm optimizer for multimodal multi-objective problems. *Applied Soft Computing* 86:105886
- Rajani, Kumar D, Kumar V (2020) Impact of controlling parameters on the performance of mopso algorithm. *Procedia Computer Science* 167:2132 – 2139, DOI <https://doi.org/10.1016/j.procs.2020.03.261>, international Conference on Computational Intelligence and Data Science
- Raquel CR, Naval PC (2005) An effective use of crowding distance in multiobjective particle swarm optimization. In: *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, Association for Computing Machinery, New York, NY, USA, GECCO '05, p 257–264, DOI 10.1145/1068009.1068047
- Srinivas N, Deb K (1994) Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation* 2(3):221–248
- Tan KC, Lee TH, Khor EF (2001) Evolutionary algorithms with dynamic population size and local exploration for multiobjective optimization. *IEEE Transactions on Evolutionary Computation* 5(6):565–588
- Tao X, Guo W, Li Q, Ren C, Liu R (2020) Multiple scale self-adaptive cooperation mutation strategy-based particle swarm optimization. *Applied Soft Computing* 89:106124
- Wang D, Tan D, Liu L (2018) Particle swarm optimization algorithm: an overview. *Soft Computing* 22(2):387–408
- Wei B, Xia X, Yu F, Zhang Y, Xu X, Wu H, Gui L, He G (2020) Multiple adaptive strategies based particle swarm optimization algorithm. *Swarm and Evolutionary Computation* 57:100731
- Yen GG, Haiming Lu (2003) Dynamic multiobjective evolutionary algorithm: adaptive cell-based rank and density estimation. *IEEE Transactions on Evolutionary Computation* 7(3):253–274
- Zhang H, Zhou A, Song S, Zhang Q, Gao X, Zhang J (2016) A self-organizing multiobjective evolutionary algorithm. *IEEE Transactions on Evolutionary Computation* 20(5):792–806
- Zhang Q, Li H (2007) Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation* 11(6):712–731
- Zhang X, Tian Y, Cheng R, Jin Y (2015) An efficient approach to nondominated sorting for evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation* 19(2):201–213
- Zhang X, Zheng X, Cheng R, Qiu J, Jin Y (2018) A competitive mechanism based multi-objective particle swarm optimizer with fast convergence. *Information Sciences* 427:63 – 76
- Zhu F, Chen D, Zou F (2020) A novel hybrid dynamic fireworks algorithm with particle swarm optimization. *Soft Computing* DOI 10.1007/s00500-020-05308-6
- Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation* 3(4):257–271
- Zitzler E, Laumanns M, Thiele L (2001) *Spea2: Improving the strength pareto evolutionary algorithm*. TIK-Report 103

Figures

Figure 1

Framework of the proposed tssMOPSO.

Figure 2

Curve of convergence contribution and detected on ZDT4.

Figure 3

Three-stage strategy of the i -th particle.

Figure 4

Decomposition of particle.

Figure 5

Update process of particle memory interval.

Figure 6

Nondominated solutions obtained by different algorithms on ZDT4.

Figure 7

Nondominated solutions obtained by different algorithms on ZDT4.

Figure 8

Nondominated solutions obtained by different algorithms on ZDT6.

Figure 9

Box-plots of the IGD results obtained by different algorithms on ZDT test functions.

Figure 10

Box-plots of the SP results obtained by different algorithms on ZDT test functions.

Figure 11

Nondominated solutions obtained by different algorithms on DTLZ2.

Figure 12

Nondominated solutions obtained by different algorithms on DTLZ7.

Figure 13

Box-plots of the IGD results obtained by different algorithms on DTLZ test functions.

Figure 14

Box-plots of the SP results obtained by different algorithms on DTLZ test functions.