Comparison of synchronous and asynchronous parallelization of extreme surrogate-assisted multi-objective evolutionary algorithm

Tomohiro Harada¹ · Misaki Kaidan² · Ruck Thawonmas³

Accepted: 1 September 2020 / Published online: 18 September 2020
© The Author(s) 2020

Abstract

This paper investigates the integration of a surrogate-assisted multi-objective evolutionary algorithm (MOEA) and a parallel computation scheme to reduce the computing time until obtaining the optimal solutions in evolutionary algorithms (EAs). A surrogate-assisted MOEA solves multi-objective optimization problems while estimating the evaluation of solutions with a surrogate function. A surrogate function is produced by a machine learning model. This paper uses an extreme learning surrogate-assisted MOEA/D (ELMOEA/D), which utilizes one of the well-known MOEA algorithms, MOEA/D, and a machine learning technique, extreme learning machine (ELM). A parallelization of MOEA, on the other hand, evaluates solutions in parallel on multiple computing nodes to accelerate the optimization process. We consider a synchronous and an asynchronous parallel MOEA as a master-slave parallelization scheme for ELMOEA/D. We carry out an experiment with multi-objective optimization problems to compare the synchronous parallel ELMOEA/D with the asynchronous parallel ELMOEA/D. In the experiment, we simulate two settings of the evaluation time of solutions. One determines the evaluation time of solutions by the normal distribution with different variances. On the other hand, another evaluation time correlates to the objective function value. We compare the quality of solutions obtained by the parallel ELMOEA/D variants within a particular computing time. The experimental results show that the parallelization of ELMOEA/D significantly reduces the computational time. In addition, the integration of ELMOEA/D with the asynchronous parallelization scheme obtains higher quality of solutions quicker than the synchronous parallel ELMOEA/D.

Keywords Evolutionary computation · Extreme learning machine · Multi-objective optimization · Parallelization · Surrogate model

1 Introduction

Evolutionary algorithms (EAs) have been applied to many real-world applications like engineering (Obayashi et al. 2010; Oyama et al. 2017), data mining (Devos et al. 2014;

Soufan et al. 2015), electronics (Roberge et al. 2014), and nanoscience (Shayeghi et al. 2015; Davis et al. 2015) because of their high search capability without any preliminary knowledge of a target problem. Since many real-world applications include two or more objectives with conflict, i.e., a trade-off, multi-objective evolutionary algorithms (MOEAs) have attracted much attention for

---

This work was supported by Japan Society for the Promotion of Science Grant-in-Aid for Young Scientists Grant Number JP19K20362.

✉ Tomohiro Harada
harada@tmu.ac.jp
Misaki Kaidan
is0160xk@ed.ritsumeimei.ac.jp
Ruck Thawonmas
ruck@is.ritsumeimei.ac.jp

- ² Graduate School of Information Science and Engineering, Ritsumeikan University, 1-1-1 Noji-Higashi Kusatsu, Shiga, Japan
- ³ College of Information Science and Engineering, Ritsumeikan University, 1-1-1 Noji-Higashi, Kusatsu, Shiga, Japan

¹ Faculty of System Design, Tokyo Metropolitan University, 6-6 Asahigaoka Hino, Tokyo, Japan

dealing with multiple objectives simultaneously. One problem when applying MOEAs to real-world applications is a necessity of enormous computing time to obtain optimal solutions. This is because MOEAs need to evaluate many tentative solutions through the optimization process, and the evaluation of tentative solutions is computationally expensive.

Previous works have proposed surrogate-assisted EAs (SAEAs) (Jin 2011; Haftka et al. 2016) to reduce the computational time required for the EA process. SAEAs construct a surrogate model that estimates the evaluation values of solutions from already evaluated solutions using machine learning techniques. A proper optimizer, like an MOEA, is executed using the evaluation estimated by a constructed surrogate model. Finally, the surrogate-assisted optimization obtains promising solutions that can exhibit superior actual evaluation value, and a time-consuming process actually evaluates them. The computational time of the estimation with a surrogate model is extremely shorter than the actual evaluation, and the actual evaluation is applied merely to promising solutions. Thus, an SAEA can reduce the computational time to obtain optimal solutions. Previous works have proposed some surrogate-assisted MOEAs, for example, MOEA/D-RBF (Zapotecas Martínez and Coello Coello 2013) or ParEGO (Knowles 2006), which use a Gaussian process or a radial based function (RBF) as a surrogate model. ELMOEA/D (Pavelski et al. 2014, 2016), extreme learning assisted MOEA/D, is one of the state-of-the-art as a novel SAEA. This method adopts the surrogate model based on an extreme learning machine (ELM) (Huang et al. 2004), and MOEA/D Zhang and Li (2007) generates promising solutions with the ELM surrogate evaluation. Their work demonstrated that ELMOEA/D outperforms ParEGO and MOEA/D-RBF on multi-objective optimization benchmarks especially when the dimension of the search space is large.

To further reduce computational time, the parallelization of SAEAs can be considered as a potential attempt. Parallel EAs (PEAs) (Alba and Tomassini 2002; Alba et al. 2013) also have been studied in the last decades. In particular, a master-slave parallelization is a straightforward approach to implement PEAs. A master-slave PEA performs the main procedure of EAs, i.e., initialization, selection, variation, and replacement, on one master node. Many slave nodes, on the other hand, execute evaluations of newly generated solutions in parallel. Since the solution evaluation is the most time-consuming part in EAs, PEAs can reduce the computational time compared to sequentially evaluating solutions. In recent years, several works have studied the integration of SAEA with parallelization for reducing the computational time of the EA process. For example, a parallel efficient global optimization has been proposed (Wang et al. 2016) and applied to the

optimization of microwave antennas (Liu et al. 2019). The work of Akinsolu et al. (2019) proposed parallel surrogate-assisted MOEA with the Gaussian process and applied it to electromagnetic design optimization.

Master-slave PEAs can be classified into two schemes, synchronous PEAs (SPEAs) and asynchronous PEAs (APEAs). SPEAs wait for all evaluations of solutions done by slave nodes and generate a new population by utilizing all newly evaluated solutions. APEAs, on the other hand, continuously generate a new solution without waiting for evaluations of other solutions. Since APEAs do not wait for the slowest solution evaluation, it is possible to efficiently utilize the computing resource of slave nodes in a situation where the evaluation of solutions differs. The previous works that integrate SAEA and the parallelization have considered only the synchronous PEA. However, it is generally assumed that the evaluation time of solutions is enormous and generally differs. For this fact, we consider that the integration of an SAEA with the asynchronous parallelization scheme is effective. The purpose of this study is to clarify which scheme of the synchronous and the asynchronous scheme is suitable for SAEA. Specifically, considering the parallelization of ELMOEA/D, we compare a synchronous parallel ELMOEA/D (SP-ELMOEA/D) and an asynchronous parallel ELMOEA/D (AP-ELMOEA/D).

This paper conducts experiments that compare SP-ELMOEA/D with AP-ELMOEA/D to investigate which parallelization scheme is appropriate for ELMOEA/D. We test these methods on well-known multi-objective optimization benchmarks, ZDT series (Zitzler et al. 2000), WFG series (Huband et al. 2005), and DTLZ series (Deb et al. 2002). We also use a few benchmarks from real-world problems provided in the black-box optimization competition (Loshchilov and Glasmachers 2015). This paper employs a simulated master-slave parallel computing environment proposed in the work of Zăvoianu et al. (2015) to measure the computational time of the competitive methods. We simulate two settings of the evaluation time in the experiment to investigate the influence of the different evaluation time characteristics. One evaluation time is determined by the normal distribution with the different variances, while another evaluation time correlates to the objective function value. We employ a hypervolume (HV) indicator (Zitzler and Thiele 1998) to assess the quality of obtained solutions.

The remaining of this paper is organized as follows. Sect. 2 firstly presents the description of a multi-objective optimization problem and explains the detail of MOEA/D. Section 3 shows the detail of ELM and ELMOEA/D. Section 4 briefly explains PEAs and proposes integrated algorithms of SP-ELMOEA/D and AP-ELMOEA/D. Section 5 describes the experimental setup to compare the synchronous and asynchronous ELMOEA/D on multi-

objective optimization problem (MOP) benchmarks and real-world problems. Section 6 shows the results on MOP benchmarks with the normally-distributed evaluation time, while Sect. 7 shows the one with the fitness-correlated evaluation time. Section 8 shows the results on real-world problems. Finally, Sect. 9 concludes this research and shows future work.

2 Multi-objective evolutionary algorithms

This section firstly explains what multi-objective optimization problem is, and then, we show a detailed algorithm of MOEA/D (Zhang and Li 2007), which is an underlying algorithm of ELMOEA/D.

2.1 Multi-objective optimization problem

A multi-objective optimization problem (MOP) is a problem of minimizing or maximizing m mutually competing objective functions $\mathbf{f}(\mathbf{x})$ (Deb 2001) expressed as follows:

$$\min_{\mathbf{x}} \mathbf{f}(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})\}^T \quad (1)$$

$$\text{Subject to: } x_{L,d} \leq x_d \leq x_{U,d} \quad (d = 1, \dots, D), \quad (2)$$

where $\mathbf{x} = (x_1, \dots, x_D)$ is a D dimensional decision variable. $x_{L,d}$ and $x_{U,d}$ are the lower and upper bounds of the d th variable, respectively. $\mathbf{f} : \Omega \rightarrow \mathbb{R}^m$ is a vector having m objective functions and maps the decision variable space Ω to the objective space $\mathbb{R}^m$.

It is difficult to obtain a single optimal solution in MOPs because each objective function is in a trade-off relationship. Therefore, the goal of MOPs is to acquire the Pareto optimal solution set. The Pareto optimal solution set is defined with the dominance relationship of solutions in MOPs denoted as follows:

$$\mathbf{x} \prec \mathbf{y} \Leftrightarrow \begin{cases} f_j(\mathbf{x}) \leq f_j(\mathbf{y}) & \forall j \in \{1, \dots, m\} \\ f_j(\mathbf{x}) < f_j(\mathbf{y}) & \exists j \in \{1, \dots, m\} \end{cases} \quad (3)$$

where $\mathbf{x} \prec \mathbf{y}$ means $\mathbf{x}$ dominates $\mathbf{y}$. The Pareto optimal solution set consists of solutions that are not dominated by any other solutions, and it is denoted as follows:

$$P = \{\mathbf{x} \in \Omega \mid \mathbf{x}' \not\prec \mathbf{x} \quad \forall \mathbf{x}' \in \Omega\} \quad (4)$$

2.2 MOEA/D

MOEA/D (Zhang and Li 2007) is a multi-objective evolutionary algorithm based on decomposition. It decomposes multiple objectives into many single objective sub-problems based on an aggregation function associated with the weight vector uniformly distributed on the objective space. MOEA/D simultaneously performs many single-objective searches with genetic operations within a local region determined by the neighborhood of the weight vectors. This paper adopts MOEA/D-DE (Li and Zhang 2009) that introduces a differential evolution (DE) operator (Storn and Price 1997; Das and Suganthan 2011) into MOEA/D as a crossover operator.

Algorithm 1 An algorithm of MOEA/D-DE.

Step 1. Initialization

- 1: Generate weight vectors $\lambda_1, \dots, \lambda_N$ uniformly on the objective space
- 2: Generate a neighborhood set $\mathbf{B}(i)$ of a weight vector λ_i
- 3: Generate an initial population $P = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$
- 4: Calculate an ideal point $\mathbf{z}^* = \min_{\mathbf{x} \in P}(\mathbf{f}(\mathbf{x}))$

Step 2. Update solutions

- 1: Permutably select the index of an updated weight vector $i \in \{1, \dots, N\}$
- 2: Set $r_1 = i$ and randomly select two indexes r_2 and r_3 ($i \neq r_2 \neq r_3$) from a neighborhood set $\mathbf{B}(i)$ with the neighborhood probability δ , otherwise, from all indices.
- 3: Generate an offspring $\mathbf{y}'$ from $\mathbf{x}_{r_1}$ ($= \mathbf{x}_i$), $\mathbf{x}_{r_2}$, and $\mathbf{x}_{r_3}$ by the DE operator:

$$y'_j = \begin{cases} x_{r_1,j} + F \times (x_{r_2,j} - x_{r_3,j}) & \text{if } rand_j < CR \vee jrand = j \\ x_{r_1,j} & \text{otherwise} \end{cases} \quad (5)$$

where j ($j = 1, \dots, D$) is the index of the variable, while $jrand$ is a randomly selected index from $1, \dots, D$. F and CR are DE parameters, while $rand_j$ is a random real-value in $[0, 1)$ for the j th variable.

- 4: Apply the polynomial mutation (PM) (Deb and Goyal, 1996) to $\mathbf{y}'$ and generate $\mathbf{y}$.
- 5: Evaluate $\mathbf{y}$
- 6: If $z_k^* < f_k(\mathbf{y})$, update $z_k^* = f_k(\mathbf{y})$ ($k = 1, \dots, m$).
- 7: Update the population P with $\mathbf{y}$ according to an aggregation function $g(\mathbf{y}|\lambda, \mathbf{z}^*)$

Step 3. Stopping Criteria

- 1: If stopping criteria are satisfied, the search is terminated. Otherwise, return to Step 2.
-

Algorithm 1 describes the detailed algorithm of MOEA/D-DE. Firstly, the weight vector sets W are generated uniformly on the objective space. W consists of N (the population size) weight vectors λ_i ($i = 1, \dots, m$). The weight vector is used to decompose multiple objectives to a single objective with an aggregation function (detailed later). Then, the neighborhood set $\mathbf{B}(i)$ is generated, consisting of T weight vectors closest to the weight vector λ_i . Next, the initial population is generated and evaluated. The ideal point $\mathbf{z}^*$ is calculated from the objective function values of the initial population. The parent solutions for the DE operator are selected from the neighborhood set $\mathbf{B}(i)$ with the probability of δ , while selected from the entire

population with the probability of $(1 - \delta)$. After applying the DE operator and the polynomial mutation (PM) (Deb and Goyal 1996), a newly generated solution is evaluated, and the ideal point $\mathbf{z}^*$ is updated. Finally, the population is updated according to the aggregation function value of a new solution. These procedures are repeated until the stopping criteria are satisfied.

Although previous works have proposed several aggregation functions, this research uses the Penalty Boundary Intersection (PBI) aggregation function. PBI is defined as Eqs. (6, 7 and 8):

$$g_{PBI}(\mathbf{x}|\lambda, \mathbf{z}^*) = d_1 + \theta d_2 \quad (6)$$

$$d_1 = \frac{\|(\mathbf{f}(\mathbf{x}) - \mathbf{z}^*)^T \boldsymbol{\lambda}\|}{\|\boldsymbol{\lambda}\|} \quad (7)$$

$$d_2 = \|\mathbf{f}(\mathbf{x}) - (\mathbf{z}^* + d_1 \boldsymbol{\lambda})\| \quad (8)$$

where θ is the penalty factor.

3 Extreme learning surrogate-assisted MOEA/D

This section introduces the original extreme learning assisted MOEA/D, ELMOEA/D (Pavelski et al. 2016). ELMOEA/D uses an extreme learning machine (ELM) (Huang et al. 2004) as a surrogate model and generates promising solutions by using MOEA/D-DE (Li and Zhang 2009) according to objective function values estimated by a constructed ELM model. In the following subsections, we first explain the overview of ELM, and then we provide the detail of ELMOEA/D.

3.1 Extreme learning machine (ELM)

ELM is a kind of machine learning technique (Huang et al. 2004). ELM is constructed as a single layer feed-forward neural network that consists of L hidden layer neurons weighted by $\mathbf{a}_j$ ($j = 1, \dots, L$), output weights $\boldsymbol{\beta}$, and an activation function $G(\mathbf{x}, \mathbf{a}, b)$. $\mathbf{a}$ indicates weights to a hidden neuron, while b indicates a bias value. The output of ELM is calculated as $Y = \mathbf{H}\boldsymbol{\beta}$, where $\boldsymbol{\beta}$ is a learned parameter, while $\mathbf{H}$ is a matrix of the activation function as $\mathbf{H} = \{G(\mathbf{x}_i, \mathbf{a}_j, b_j)\}$.

The most notable feature of ELM is that the hidden layer weights $\mathbf{a}$ and biases b are randomly assigned and are not learned. The output weights $\boldsymbol{\beta}$ are only parameters to be learned from N distinct training samples of n -dimensional inputs $\mathbf{x}_i$ with m dimensional outputs $\mathbf{t}_i$. In particular, $\boldsymbol{\beta}$ is calculated as $\boldsymbol{\beta} = (\mathbf{I}/C + \mathbf{H}\mathbf{H}^T)^{-1} \mathbf{H}^T \mathbf{T}$, where C is a regularization parameter, $\mathbf{I}$ indicates an $N \times N$ unit matrix, and $\mathbf{T} = [\mathbf{t}_1, \dots, \mathbf{t}_N]$ is the matrix of desired outputs.

ELM can use any piece-wise nonlinear continuous activation functions in the hidden layer, i.e., $G(\mathbf{x}, \mathbf{a}, b)$. The original work of ELMOEA/D used the following three activation functions:

– Sigmoid (SIG):

$$G_{SIG}(\mathbf{x}, \mathbf{a}, b) = \frac{1}{1 + \exp(-(\mathbf{a} \cdot \mathbf{x} + b))} \quad (9)$$

– Gaussian (GAU):

$$G_{GAU}(\mathbf{x}, \mathbf{a}, b) = \exp(-b\|\mathbf{x} - \mathbf{a}\|^2) \quad (10)$$

– Multiquadric (MQ):

$$G_{MQ}(\mathbf{x}, \mathbf{a}, b) = \left(\|\mathbf{x} - \mathbf{a}\|^2 + b^2\right)^{\frac{1}{2}} \quad (11)$$

The advantages of ELM to neural networks and support vector machines (SVMs) are summarized as follows (Huang et al. 2004; Huang 2015):

- The learning speed of ELM is extremely fast. It can train single-layer feed-forward networks much faster than classical learning algorithms.
- Unlike traditional classic gradient-based learning algorithms, which intend to reach minimum training error but do not consider the magnitude of weights, ELM tends to reach not only the smallest training error but also the smallest norm of weights. Thus, ELM tends to have better generalization performance for feed-forward neural networks.
- Unlike the traditional classic gradient-based learning algorithms that only work for differentiable activation functions, the ELM learning algorithm can be used to train single-layer feed-forward networks with non-differentiable activation functions.
- Unlike traditional classic gradient-based learning algorithms facing several issues like local minimum, improper learning rate, and overfitting, etc, ELM tends to reach the solutions straightforward without such trivial issues. The ELM learning algorithm looks much simpler than most learning algorithms for feed-forward neural networks.
- Unlike SVM, which does not consider the feature representation and functioning roles of each inner hidden layer, ELM studies feature representations in each layer.

Algorithm 2 An algorithm of ELMOEA/D

Step 1. Initialization

- 1: Generate an initial solutions by LHS.
- 2: Evaluate all initial solutions with actual evaluation.
- 3: Store all evaluated solutions in the archive population P_a .
- 4: Set all evaluated solutions to an initial training set (P_t) of an ELM model.
- 5: Generate a weight vector set W of MOEA/D.
- 6: Generate a selector set W_s used to select promising solutions.
- 7: Associate a neighborhood vector set $\mathbf{B}^s(i)$ for each selector weight vector.

Step 2. Construction of the ELM surrogate model

- 1: Train ELMs with different settings by using a training set P_t .
- 2: Select one ELM model with the smallest mean square error (MSE) for the training set.

Step 3. Optimization by MOEA/D-DE

- 1: Initialize the population of MOEA/D-DE from the archive population P_a .
- 2: Optimize the population up to the maximum number of generations (G generations) with MOEA/D-DE while evaluating solutions with the ELM surrogate model.

Step 4. Selection of N_s promising solutions

- 1: Calculate aggregation function values for all solutions in the final population.
- 2: Select the best N_s solutions from each selector weight vector λ_i^s .

Step 5. Update population and training set

- 1: Apply actual evaluation to all selected promising solutions
- 2: Add all evaluated solutions to the archive population P_a .
- 3: Update the training set P_t with the evaluated solutions.

Step 6. Stopping criteria

- 1: Repeat Steps 2 to 5 until the terminal condition.
-

3.2 Algorithm of ELMOEA/D

Algorithm 2 briefly describes the algorithm of ELMOEA/D (see detail in the work of (Pavelski et al. 2016)).

ELMOEA/D uses Latin hypercube sampling (LHS) (McKay et al. 1979) as the sampling method in the initialization step. ELMOEA/D uses two weight vector sets, one is used for the evolution of MOEA/D, while another is

used to select promising solutions. A weight vector set W used for the evolution of MOEA/D-DE is generated that uniformly spread on the objective space. The weight vector set consists of N weight vectors ($W = \{\lambda_1, \dots, \lambda_N\}$). Subsequently, a selector weight vector set W_s used for the selection of N_s promising solutions is generated, which consists of N_s selector weight vectors $W^s = \{\lambda_1^s, \dots, \lambda_{N_s}^s\} \subset W$. Finally, a neighborhood vector set for the promising solution selection $\mathbf{B}^s(i) =$

$\{\lambda_{i,1}, \dots, \lambda_{i,K_s}\} (\lambda_{i,j} \in W)$ is associated with each selector weight vector $\lambda_i^s \in W_s$. Each neighborhood vector set $\mathbf{B}^s(i)$ consists of $K_s (= N/N_s)$ weight vectors in W that are closest to each selector weight vector λ_i^s .

After the initialization, ELMOEA/D constructs an ELM surrogate model by using a training set P_t . ELMs are trained with different activation functions and different normalization parameters C in this step. The best ELM model that performs the smallest mean square error (MSE) for the training set is selected and used as a surrogate model.

The constructed ELM model is used for evaluating solutions during the MOEA/D-DE procedure. The initial population is randomly selected from the non-dominated solutions in the archive population P_a if its size exceeds the population size N . Otherwise, LHS generates the remaining solutions in the bound of $[\mu - \sigma, \mu + \sigma]$, where μ and σ are the mean and the standard deviation calculated from the non-dominated solutions in P_a .

The promising solutions for the actual evaluation are selected from the final population after the optimization of MOEA/D-DE. The aggregation function values of solutions in the final population are calculated for each $\lambda_{i,j} \in \mathbf{B}^s(i)$ based on the estimated objective values by the surrogate model. Then, the best solution is selected from each neighborhood vector set $\mathbf{B}^s(i)$, and totally, N_s promising solutions are selected for the actual evaluation.

The previous work (Pavelski et al. 2016) reported that ELMOEA/D has the following advantages:

- Since ELMOEA/D uses a simpler surrogate model with less user-specific parameters, it consumes less computational resources and is faster than the conventional surrogate-assisted EA.
- ELMOEA/D showed promising results on the MOP benchmarks with high dimensional design variables compared with ParEGO and MOEA/D-RBF.

For these advantages, our research employs ELMOEA/D.

4 Parallelization of ELMOEA/D

This paper considers two schemes of ELMOEA/D parallelization, a synchronous parallel ELMOEA/D (SP-ELMOEA/D), and an asynchronous parallel ELMOEA/D (AP-ELMOEA/D). This section firstly introduces parallel EAs briefly, and then, we explain the detailed description of SP-ELMOEA/D and AP-ELMOEA/D. We finally introduce two selection mechanisms for AP-ELMOEA/D.

4.1 Parallel evolutionary algorithm

When applying EAs to real-world optimization problems, solution evaluations may take much computational time because of, for example, a requirement of physical simulation. Previous works have studied parallel EAs (PEAs) (Alba and Tomassini 2002; Alba et al. 2013) to speed up the optimization process of EAs by using multiple computing resources. In recent years, parallel MOEAs have been studied to reduce the computing time to approximate the Pareto front (Talbi 2019).

A master-slave parallelization is one of the most typical and straightforward approaches of PEAs. A master-slave PEA performs the main procedure of EAs like initialization, selection, variation, and replacement on a single master computing node. At the same time, many slave computing nodes execute fitness evaluations in parallel. Such a master-slave parallelization has been applied to solving real-world problems. For example, the work of Barbera et al. (2018) applied the massively parallel EA to the phylogenetic placement problem. The work of Strofylas et al. (2018) proposed a parallel differential evolution method for calibrating a second-order macroscopic traffic flow.

Master-slave PEAs can be classified into two schemes, a synchronous evolution and an asynchronous one. Synchronous PEAs (SPEAs) is similar to generation based EAs, where the new population is generated after waiting for all solution evaluations. On the other hand, asynchronous PEAs (APEAs) is similar to steady-state EAs. A new solution is immediately generated whenever one solution evaluation completes without waiting for other solution evaluations. In general, an SPEA has a higher search capability than an APEA because an SPEA can utilize many evaluation information simultaneously when generating a new population. On the other hand, when the evaluation time of solutions differs from each other, i.e., the variance of the evaluation time is large, an APEA shows a higher search efficiency than in terms of the computational time (Harada and Takadama 2013, 2014; Scott and De Jong 2015a, b). This is because APEAs can reduce the idling time of slave nodes that quickly complete the evaluation tasks, unlike SPEAs need to wait for the slowest solution evaluations.

4.2 Synchronous parallel ELMOEA/D

Algorithm 3 shows a master-slave synchronous parallel ELMOEA/D (SP-ELMOEA/D). SP-ELMOEA/D is an extension of ELMOEA/D, and the underlined texts denote the difference from Algorithm 2. At the initialization procedure, all generated solutions are sent to slave nodes in Step 1–2, and their evaluations are firstly waited for in Step 1–3. Then, the remaining initialization procedure is

executed. After evolving solutions on MOEA/D-DE with an ELM surrogate, N_s promising solutions are selected, and they are sent to slave nodes in Step 5–1. SP-ELMOEA/D waits for all evaluations from slave nodes in Step 5–2 and

updates the archive population and the training set. SP-ELMOEA/D repeats these procedures until the terminal condition is satisfied.

Algorithm 3 An algorithm of SP-ELMOEA/D. Modified parts from Algorithm 2 are described with underlined text.

Step 1. Initialization

- 1: Generate an initial solutions by LHS.
- 2: Send all initial solutions to slave nodes.
- 3: Receive all evaluations from slave nodes.
- 4: Store all evaluated solutions in the archive population P_a .
- 5: Set all evaluated solutions to an initial training set (P_t) of an ELM model.
- 6: Generate a weight vector set W of MOEA/D.
- 7: Generate a selector set W_s used to select promising solutions.
- 8: Associate a neighborhood vector set $\mathbf{B}^s(i)$ for each selector weight vector.

Step 2. Construction of the ELM surrogate model

- 1: Train ELMs with different settings by using a training set P_t .
- 2: Select one ELM model with the smallest mean square error (MSE) for the training set.

Step 3. Optimization by MOEA/D-DE

- 1: Initialize the population of MOEA/D-DE from the archive population P_a .
- 2: Optimize the population up to the maximum number of generations (G generations) with MOEA/D-DE while evaluating solutions with the ELM surrogate model.

Step 4. Selection of N_s promising solutions

- 1: Calculate aggregation function values for all solutions in the final population.
- 2: Select the best N_s solutions from each selector weight vector λ_i^s .

Step 5. Update population and training set

- 1: Send all selected promising solutions to slave nodes.
- 2: Receive all evaluations from slave nodes.
- 3: Add all evaluated solutions to the archive population P_a .
- 4: Update the training set P_t with the evaluated solutions.

Step 6. Stopping criteria

- 1: Repeat Steps 2 to 5 until the terminal condition.
-

4.3 Asynchronous parallel ELMOEA/D

AP-ELMOEA/D evaluates solutions in parallel as same as SP-ELMOEA/D but waits for only one solution evaluation every step. When one solution evaluation completes, it is appended to the archive population, and MOEA/D-DE with the ELM surrogate model generates a new promising solution. The differences between AP-ELMOEA/D and SP-ELMOEA/D are in Steps 4 and 5 in Algorithm 3. Concretely, these procedures are modified as Algorithm 4.

First, AP-ELMOEA/D waits for only one evaluation from a slave node in Step 5'–2, unlike SP-ELMOEA/D waits for all N_s solution evaluations. In response to this, Steps 4' and 5' of AP-ELMOEA/D are modified to address one solution. In Step 4–1 in Algorithm 3, SP-ELMOEA/D selects N_s promising solutions in each generation and actually evaluates them in parallel. On the other hand, AP-ELMOEA/D selects only one promising solution in Step 4'–3. For this selection, AP-ELMOEA/D needs to select an index k_s of the selector weight vector in Step 4'–2 instead of considering all N_s selector vectors. We take this into account in the next subsection. In addition to this, AP-ELMOEA/D addresses only one solution in the update of the archive population P_a^{g+1} and the training set P_t^{g+1} in Steps 5'–3 and 5'–4. Besides, because only one solution is selected for an actual evaluation every MOEA/D-DE optimization step, the surrogate model is constructed every N_s actual evaluations in AP-ELMOEA/D not to increase the number of the ELM training.

4.4 Promising solution selection of AP-ELMOEA/D

When applying the asynchronous evolution scheme to ELMOEA/D, we need to determine how to select one promising solution from the population optimized by MOEA/D-DE on the surrogate evaluation model. In particular, an index k_s should be chosen Step 4' in Algorithm 4 whenever one solution evaluation completes. This paper explores simple two selection mechanisms of an index k_s : the index order based selection and the random order selection.

4.4.1 Index order based selection:

A straightforward way to select an index k_s is to use the order of the index of the neighborhood vector set. This paper denotes AP-ELMOEA/D with the promising selection using the order of the index of the neighborhood vector set as AP-ELMOEA/D-IO. AP-ELMOEA/D-IO selects a promising solution depending on the order of the index of the neighborhood vector set. AP-ELMOEA/D-IO initially set an index k_s to 1, and whenever a promising solution is selected, k_s is incremented as $k_s \leftarrow k_s + 1$. If k_s exceeds K_s , which is the number of the neighborhood vector set, k_s is reset to 1.

4.4.2 Random order selection

The second way is to choose an index k_s every selection procedure randomly. This paper denotes this as AP-

---

Algorithm 4 An algorithm of AP-ELMOEA/D. Only Step 4 and Step 5 modified from Algorithm 3 are shown, and modified parts are described with underlined text.

Step 4'. Selection of one promising solution

- 1: Calculate aggregation function values for all solutions in the final population.
  - 2: Select an index k_s of the selector weight vector ($k_s \in \{1, \dots, N_s\}$) (de-tailed below).
  - 3: Select the best solution from k_s th selector weight vector $\lambda_{k_s}^s$.
-

Step 5'. Update population and training set

- 1: Send one selected promising solution to an idling slave node.
  - 2: Receive one evaluation from a slave node.
  - 3: Add the evaluated solution to the archive population P_a^{g+1} .
  - 4: Update the next training set (P_t^{g+1}) with the evaluated solution.
-

ELMOEA/D-RO. Whenever an index k_s is selected, it is uniformly chosen from 1 to N_s without considering the previous index, i.e., consecutive selection of the same index is allowed. Since an actual implementation of MOEA/D often uses a random permutation to select the aggregation vector during the optimization Fan et al. (2019); Nebro et al. (2015), the random order selection can be a valid strategy in AP-ELMOEA/D.

5 Experiment

We conduct an experiment to compare three different parallel ELMOEA/D variants, i.e., SP-ELMOEA/D, AP-ELMOEA/D-IO, and AP-ELMOEA/D-RO, to clarify which parallelization scheme is suitable for ELMOEA/D. Additionally, we compare the sequential ELMOEA/D (i.e., run on a single CPU) to confirm the effectiveness of the parallelization. The following subsections explain the benchmark problems used in the experiment, the experimental setting, and the evaluation criteria.

5.1 Problem instance

We use well-known multi-objective benchmarks, the ZDT test suite (Zitzler et al. 2000), the WFG test suite (Huband et al. 2005), and the DTLZ test suite (Deb et al. 2002). This experiment considers bi-objective minimization for all test suites. The dimension of the decision variable D is as:

- ZDT1–3: $D = 30$
- ZDT4 and 6: $D = 10$
- WFG1–9: $D = 24$
- DTLZ1–7: $D = 5$.

Additionally, we use four real-world problems provided by the black box optimization competition (Loshchilov and Glasmachers 2015). In particular, three are from the engineering design, a heat exchanger problem, a hydro-dynamics problem, and a vibrating platform problem, while one is from the operational research, a facility placement problem. All of them are bi-objective minimization, and $D = 16$ for the heat exchanger problem, $D = 6$ for the hydro-dynamics problem, $D = 5$ for the vibrating platform problem, and $D = 20$ for the facility placement problem.

5.2 Experimental setting

We conduct the experiment on the pseudo (*simulated*) master-slave parallel computing environment that measures the computational time according to the model proposed in the work of Zăvoianu et al. (2015). In the experiment, we define the unit of the computing time as “simulation time”. In particular, we define one unit

simulation time as the time to complete the sequential tasks on the master node, i.e., the ELM learning and the MOEA/D optimization with the surrogate evaluation.

We test two settings of evaluation time of solutions. One determines the evaluation time of any solutions by the normal distribution of $\mathcal{N}(t_p, c_v \times t_p)$ simulation time on the distributed slave nodes. t_p ($\gg 1$) is the mean evaluation time, while the random variance is added according to the normal distribution with the standard deviation $c_v \times t_p$, where c_v ($c_v \geq 0$) is a parameter to determine the variance of the evaluation time. In this setting, the evaluation time does not depend on the feature of evaluated solutions. We call this evaluation time *normally-distributed evaluation time*.

Another setting defines the evaluation time of solutions correlates to the objective function value. In particular, the evaluation time is calculated as:

$$t_{cor}(\mathbf{x}) = \begin{cases} t_p - c_g t_p + c_g t_p f_1(\mathbf{x}) & 0 \leq f_1(\mathbf{x}) \leq 2 \\ t_p - c_g t_p & f_1(\mathbf{x}) < 0 \\ t_p + c_g t_p & \text{otherwise} \end{cases} \quad (12)$$

where $f_1(\mathbf{x})$ indicates the first objective function value of a solution $\mathbf{x}$, t_p ($\gg 1$) is the base evaluation time when $f_1(\mathbf{x}) = 1$, while c_g denotes the parameter to decides the gradient of the evaluation time. If $c_g > 0$, the evaluation time positively correlates to the first objective function value, i.e., the greater the objective function value, the longer the evaluation time. If $c_g < 0$, on the other hand, the evaluation time negatively correlates to the first objective function value, i.e., the greater the objective function value, the shorter the evaluation time. Since the evaluation time depends on the objective function value, the search direction of the AP-ELMOEA/D variants has the possibility to *biased* toward the search space where solutions quickly complete their evaluations (Scott and De Jong 2015a).

This experiment uses the parameters shown in Table 1, most of which are the same as the original work (Pavelski et al. 2016). Notably, we use the MOEA/D-DE parameter settings that the original MOEA/D-DE work recommended (Li and Zhang 2009). Ten slave nodes are used for solution evaluations, which means $N_s = 10$ selected promising solutions can be evaluated simultaneously. To clarify how the variance of the evaluation time on the parallel computing environment affects the search performance, we consider different magnitudes of the variance $c_v = \{0.02, 0.05, 0.10, 0.20\}$ in the normally-distributed evaluation time. On the other hand, in the fitness-correlated evaluation time, we use the different gradient parameters $c_g = \{\pm 0.2, \pm 0.5\}$ to clarify the influence of the evaluation time gradient.

This paper uses the hypervolume (HV) indicator (Zitzler and Thiele 1998) to assess the quality of the achieved

Table 1 Parameter settings

MOEA/D-DE parameters	
#Generations (G)	100
Population size (N)	100
Neighborhood size (T)	10
Neighborhood probability (δ)	0.9
PBI penalty (θ)	5.0
DE scaling factor (F)	0.5
DE crossover rate (CR)	1.0
Polynomial mutation rate (p_m)	1/ D
Distribution index (η)	20
ELM parameters	
Training set size	$10 \times D$
Number of hidden neurons (N_H)	$2D + 1$
Activation Function (A)	SIG, MUL, GAU
Regularization parameter (C)	$2^{-5}, 2^0, 2^5$
ELMOEA/D parameters	
#Selector weights (N_s)	10
Initial minimum distance (ϵ) ^a	0.001
Master-slave parallel environment	
#slave nodes	10 ($= N_s$)
Processing time of master node (t_s)	1 [simulation time]
Mean evaluation time (t_p)	1000 [simulation time]
Variance parameter (c_v)	0.02, 0.05, 0.1, 0.2
Gradient parameter (c_g)	$\pm 0.2, \pm 0.5$

^aThis parameter is used when selecting a promising solution(s) to avoid selecting similar solutions already exist in the archive population

solution set by the methods. The HV indicator can evaluate both the convergence and the diversity of obtained solutions. For the benchmark problems, the scale of HV value differs in each benchmark problem. To clearly display the results in the table, we normalize the HV value by the ideal HV value calculated from the true Pareto front for each benchmark problem. The normalized HV value $\widetilde{HV}$ is defined as follows:

$$\widetilde{HV} = \frac{HV}{HV_{ideal}}, \quad (13)$$

where HV indicates the HV value obtained by each method, while HV_{ideal} indicates the ideal HV value. On the other hand, for the real-world problems, since the ideal HV value is unknown, we use raw HV values.

The reference point, which is the extreme point of the area to be covered by acquired solutions, is required to calculate the HV value. We use the following reference points, which all are the same as the original work:

- ZDT1–3, 6, WFG1–9, DTLZ2, 4–6: $\{5.0, 5.0\}$

- ZDT4, DTLZ1, 3, 7: $\{100.0, 100.0\}$

For the real-world problems, on the other hand, the reference point is set to $\{1.0, 1.0\}$, which is used in (Loshchilov and Glasmachers 2015).

We assess three methods from the viewpoint of the number of actual evaluations and the elapsed simulation time. The HV values of 30 independent trials are calculated for the maximum number of actual evaluations and the maximum elapsed simulation time. We set the maximum number of the *actual* evaluations as 2000 for all ZDT instances, 1000 for all WFG instances, and 200 for all DTLZ instances. For the real-world problems, we set the maximum number of the actual evaluations as 500 for the heat exchanger problem, 200 for the vibrating platform problem, and 1000 for the hydro-dynamics and the facility placement problems. The maximum elapsed simulation time, on the other hand, is set to 1.5×10^5 simulation time for the ZDT instances, 7.0×10^4 simulation time for the WFG instances, and 1.0×10^4 simulation time for the DTLZ instances. For the real-world problems, we set the maximum elapsed simulation time to 3.0×10^4 simulation time for the heat exchanger problem, 1.2×10^4 simulation time for the vibrating platform problem, and 7.0×10^4 simulation time for the hydro-dynamics and the facility placement problems. These maximum simulation times are chosen because all asynchronous schemes complete their executions before these computational times, though the synchronous one needs more computational time to complete its execution.

6 Results on benchmark problems with the normally-distributed evaluation time

This section shows the experimental results on the benchmark problems with the use of the normally-distributed evaluation time. First, we compare the HV values of the parallel ELMOEA/D variants at the maximum number of the actual evaluations. Then we compare them from the viewpoint of the elapsed simulation time.

6.1 Hypervolume at the maximum number of the actual evaluations

Table 2 shows the mean normalized HV values and their standard deviations of the competitive methods at the maximum number of the actual evaluations. This table does not include the sequential ELMOEA/D because its result is the same as SP-ELMOEA/D at the same number of the actual evaluations. For each benchmark and each c_v value, the maximum mean values are shown in the

Table 2 The mean normalized HV values ($\widehat{HV}$) and their standard deviation at the maximum number of the actual evaluations in 30

independent trials with the normally-distributed evaluation time

	c_v	SP-ELMOEA/D		AP-ELMOEA/D-IO		AP-ELMOEA/D-RO	
		Mean	Std.	Mean	Std.	Mean	Std.
ZDT1	0.02	0.9996	1.0×10^{-4}	0.9997 ¹	7.1×10^{-5}	0.9996	1.0×10^{-4}
	0.05	0.9996	1.1×10^{-4}	0.9996	8.6×10^{-5}	0.9996	7.6×10^{-5}
	0.1	0.9996	9.4×10^{-5}	0.9997 ^{1,3}	9.1×10^{-5}	0.9996	9.6×10^{-5}
	0.2	0.9996	1.1×10^{-4}	0.9997 ¹	10.0×10^{-5}	0.9996	9.8×10^{-5}
ZDT2	0.02	0.9997	1.2×10^{-4}	0.9998 ¹	8.9×10^{-5}	0.9997	8.6×10^{-5}
	0.05	0.9997	1.2×10^{-4}	0.9997 ³	8.2×10^{-5}	0.9997	9.5×10^{-5}
	0.1	0.9997	9.9×10^{-5}	0.9997 ¹	8.3×10^{-5}	0.9997 ¹	7.9×10^{-5}
	0.2	0.9997	9.3×10^{-5}	0.9997	6.4×10^{-5}	0.9998 ¹	8.0×10^{-5}
ZDT3	0.02	0.9787 ^{2,3}	1.1×10^{-2}	0.9727	1.2×10^{-2}	0.9689	1.4×10^{-2}
	0.05	0.9776 ^{2,3}	9.0×10^{-3}	0.9714	1.4×10^{-2}	0.9713	1.3×10^{-2}
	0.1	0.9751	1.1×10^{-2}	0.9747	1.3×10^{-2}	0.9689	1.8×10^{-2}
	0.2	0.9780 ^{2,3}	1.2×10^{-2}	0.9712	1.4×10^{-2}	0.9726	1.1×10^{-2}
ZDT4	0.02	0.6170	6.5×10^{-2}	0.6296	6.0×10^{-2}	0.6299	5.7×10^{-2}
	0.05	0.6394 ³	6.7×10^{-2}	0.6276	6.3×10^{-2}	0.6056	4.1×10^{-2}
	0.1	0.6373	7.5×10^{-2}	0.6482 ³	6.4×10^{-2}	0.6122	5.0×10^{-2}
	0.2	0.6061	4.2×10^{-2}	0.6357 ¹	6.9×10^{-2}	0.6255	5.6×10^{-2}
ZDT6	0.02	0.9701	3.4×10^{-2}	0.9544	4.0×10^{-2}	0.9765 ²	3.2×10^{-2}
	0.05	0.9637	4.0×10^{-2}	0.9735	3.3×10^{-2}	0.9652	3.7×10^{-2}
	0.1	0.9629	3.9×10^{-2}	0.9585	3.8×10^{-2}	0.9653	3.5×10^{-2}
	0.2	0.9586	3.8×10^{-2}	0.9692	3.3×10^{-2}	0.9793	2.0×10^{-2}
WFG1	0.02	0.5976	6.7×10^{-4}	0.5979	5.7×10^{-4}	0.5979	6.8×10^{-4}
	0.05	0.5978	5.5×10^{-4}	0.5978	5.8×10^{-4}	0.5980	8.5×10^{-4}
	0.1	0.5977	6.5×10^{-4}	0.5977	4.8×10^{-4}	0.5978	7.3×10^{-4}
	0.2	0.5976	7.3×10^{-4}	0.5979	6.1×10^{-4}	0.5981 ¹	5.8×10^{-4}
WFG2	0.02	0.7977	3.6×10^{-2}	0.8050	2.9×10^{-2}	0.7993	3.7×10^{-2}
	0.05	0.7992	3.3×10^{-2}	0.8165 ^{1,3}	2.2×10^{-2}	0.7936	4.0×10^{-2}
	0.1	0.7984	3.3×10^{-2}	0.8007	3.6×10^{-2}	0.8092	2.7×10^{-2}
	0.2	0.7967	3.9×10^{-2}	0.8125 ¹	2.9×10^{-2}	0.7980	4.0×10^{-2}
WFG3	0.02	0.7355	3.1×10^{-3}	0.7367	3.0×10^{-3}	0.7361	3.0×10^{-3}
	0.05	0.7351	2.5×10^{-3}	0.7370 ¹	2.2×10^{-3}	0.7362	3.2×10^{-3}
	0.1	0.7350	3.2×10^{-3}	0.7358	2.8×10^{-3}	0.7359	2.7×10^{-3}
	0.2	0.7351	2.6×10^{-3}	0.7359	2.7×10^{-3}	0.7357	2.0×10^{-3}
WFG4	0.02	0.8295	1.7×10^{-2}	0.8374 ³	1.6×10^{-2}	0.8283	1.6×10^{-2}
	0.05	0.8280	1.7×10^{-2}	0.8306	1.3×10^{-2}	0.8373 ^{1,2}	1.8×10^{-2}
	0.1	0.8279	1.4×10^{-2}	0.8342 ¹	1.7×10^{-2}	0.8375 ¹	1.6×10^{-2}
	0.2	0.8337	1.5×10^{-2}	0.8324	1.6×10^{-2}	0.8374	1.6×10^{-2}
WFG5	0.02	0.7841	2.6×10^{-2}	0.8066 ¹	2.8×10^{-2}	0.7976 ¹	2.1×10^{-2}
	0.05	0.7946	2.8×10^{-2}	0.7932	2.6×10^{-2}	0.7990	2.4×10^{-2}
	0.1	0.7896	2.7×10^{-2}	0.8030 ¹	1.8×10^{-2}	0.8042 ¹	1.9×10^{-2}
	0.2	0.7951	2.3×10^{-2}	0.7953	2.4×10^{-2}	0.7923	2.9×10^{-2}
WFG6	0.02	0.9405	1.7×10^{-3}	0.9417 ¹	2.1×10^{-3}	0.9419 ¹	1.5×10^{-3}
	0.05	0.9406	1.7×10^{-3}	0.9417 ¹	1.9×10^{-3}	0.9420 ¹	1.6×10^{-3}
	0.1	0.9405	1.6×10^{-3}	0.9424 ¹	1.3×10^{-3}	0.9422 ¹	1.2×10^{-3}

Table 2 (continued)

	c_v	SP-ELMOEA/D		AP-ELMOEA/D-IO		AP-ELMOEA/D-RO	
		Mean	Std.	Mean	Std.	Mean	Std.
WFG7	0.2	0.9412	1.4×10^{-3}	0.9421 ¹	2.0×10^{-3}	0.9420 ¹	1.1×10^{-3}
	0.02	0.8293	7.9×10^{-3}	0.8305	6.7×10^{-3}	0.8302	6.0×10^{-3}
	0.05	0.8256	8.5×10^{-3}	0.8320 ¹	7.2×10^{-3}	0.8318 ¹	7.8×10^{-3}
	0.1	0.8301	8.2×10^{-3}	0.8306	7.5×10^{-3}	0.8319	6.6×10^{-3}
WFG8	0.2	0.8257	9.9×10^{-3}	0.8300	8.5×10^{-3}	0.8284	7.8×10^{-3}
	0.02	0.7579	9.9×10^{-3}	0.7571	9.9×10^{-3}	0.7577	1.1×10^{-2}
	0.05	0.7564	10.0×10^{-3}	0.7587	9.4×10^{-3}	0.7582	8.4×10^{-3}
	0.1	0.7562	1.1×10^{-2}	0.7587	9.5×10^{-3}	0.7610	8.7×10^{-3}
WFG9	0.2	0.7554	1.0×10^{-2}	0.7604 ¹	9.6×10^{-3}	0.7594	9.1×10^{-3}
	0.02	0.7817	2.1×10^{-2}	0.7827	2.5×10^{-2}	0.7828	2.7×10^{-2}
	0.05	0.7832	2.3×10^{-2}	0.7842	3.1×10^{-2}	0.7795	2.5×10^{-2}
	0.1	0.7873 ³	2.6×10^{-2}	0.7838	1.8×10^{-2}	0.7760	2.3×10^{-2}
DTLZ1	0.2	0.7883 ^{2,3}	1.9×10^{-2}	0.7694	1.9×10^{-2}	0.7781	2.5×10^{-2}
	0.02	0.9080	1.8×10^{-2}	0.9069	1.8×10^{-2}	0.9085	1.8×10^{-2}
	0.05	0.9052	1.9×10^{-2}	0.9124 ³	2.2×10^{-2}	0.9039	1.7×10^{-2}
	0.1	0.9064	2.1×10^{-2}	0.9093	2.0×10^{-2}	0.9031	1.8×10^{-2}
DTLZ2	0.2	0.9049	1.9×10^{-2}	0.9072	1.9×10^{-2}	0.9068	1.8×10^{-2}
	0.02	0.9966	7.3×10^{-4}	0.9965	1.1×10^{-3}	0.9964	7.1×10^{-4}
	0.05	0.9963	6.8×10^{-4}	0.9966 ^{1,3}	8.9×10^{-4}	0.9961	10.0×10^{-4}
	0.1	0.9963	7.8×10^{-4}	0.9966 ³	8.5×10^{-4}	0.9962	6.9×10^{-4}
DTLZ3	0.2	0.9963	9.6×10^{-4}	0.9967	8.0×10^{-4}	0.9964	7.5×10^{-4}
	0.02	0.4916	1.0×10^{-1}	0.5028	1.1×10^{-1}	0.4698	1.2×10^{-1}
	0.05	0.4750	1.1×10^{-1}	0.4971	1.1×10^{-1}	0.4919	1.0×10^{-1}
	0.1	0.4600	1.2×10^{-1}	0.4845	1.1×10^{-1}	0.4739	1.0×10^{-1}
DTLZ4	0.2	0.4860	1.0×10^{-1}	0.4829	1.2×10^{-1}	0.4656	1.1×10^{-1}
	0.02	0.8784	6.9×10^{-2}	0.8743	6.9×10^{-2}	0.8676	6.4×10^{-2}
	0.05	0.8734	6.7×10^{-2}	0.8701	6.3×10^{-2}	0.8769	7.0×10^{-2}
	0.1	0.8770	6.8×10^{-2}	0.8717	6.5×10^{-2}	0.8686	6.6×10^{-2}
DTLZ5	0.2	0.8726	6.6×10^{-2}	0.8692	6.6×10^{-2}	0.8746	6.7×10^{-2}
	0.02	0.9980	7.3×10^{-4}	0.9979	1.1×10^{-3}	0.9977	7.1×10^{-4}
	0.05	0.9977	6.8×10^{-4}	0.9980 ^{1,3}	8.9×10^{-4}	0.9974	10.0×10^{-4}
	0.1	0.9976	7.8×10^{-4}	0.9980 ³	8.5×10^{-4}	0.9976	7.0×10^{-4}
DTLZ6	0.2	0.9976	9.6×10^{-4}	0.9980	8.0×10^{-4}	0.9978	7.5×10^{-4}
	0.02	0.9995	1.4×10^{-3}	1.0002	6.7×10^{-4}	1.0001	6.7×10^{-4}
	0.05	0.9995	1.2×10^{-3}	1.0001 ¹	1.0×10^{-3}	1.0002 ¹	6.0×10^{-4}
	0.1	0.9997	1.3×10^{-3}	1.0000	7.3×10^{-4}	1.0000	8.1×10^{-4}
DTLZ7	0.2	0.9996	1.3×10^{-3}	1.0002 ¹	6.8×10^{-4}	1.0001	6.0×10^{-4}
	0.02	1.0001	1.9×10^{-6}	1.0001	7.0×10^{-6}	1.0001	2.0×10^{-5}
	0.05	1.0001	2.1×10^{-6}	1.0001	9.8×10^{-6}	1.0001	9.0×10^{-6}
	0.1	1.0001	5.8×10^{-6}	1.0001	9.6×10^{-6}	1.0001	6.9×10^{-6}
	0.2	1.0001 ²	4.5×10^{-6}	1.0001	6.5×10^{-6}	1.0001	4.7×10^{-6}

The result that is significantly better than another is signed with a number of the method (1: SP-ELMOEA/D, 2: AP-ELMOEA/D-IO, 3: AP-ELMOEA/D-RO). The highest mean values are shown with in the **bold** style

Table 3 An aggregation of the pairwise statistical test using Mann–Whitney U test with a 5% significance level for the result at the maximum number of the actual evaluations with the normally-distributed evaluation time

c_v	Sync. vs. IO			Sync. vs. RO			IO vs. RO		
	1	≈	2	1	≈	2	1	≈	2
0.02	1	16	4	1	18	2	1	19	1
0.05	1	13	7	2	15	4	5	15	1
0.1	0	16	5	1	16	4	4	17	0
0.2	3	12	6	2	16	3	0	21	0
total	5	57	22	6	65	13	10	72	2

If the first algorithm wins against the second one, “1” is counted, while “2” is counted if the second one wins. If two are a tie, “≈” is counted. “Sync.” indicates SP-ELMOEA/D, “IO” indicates AP-ELMOEA/D-IO, and “RO” indicates AP-ELMOEA/D-RO

bold style. We conduct a statistic test with the Mann–Whitney U test with a 5% significance level. If a method is significantly better than another, it is specified as the superscript character in the “Mean” column. “1” indicates that a corresponding method is significantly better than SP-ELMOEA/D, and similarly “2” indicates AP-ELMOEA/D-IO, and “3” indicates AP-ELMOEA/D-RO. These results indicate that AP-ELMOEA/D-IO has the highest mean HV values in most cases, although the difference is not significant for many benchmarks. On the other hand, SP-ELMOEA/D significantly outperforms the AP-ELMOEA/D variants in ZDT3 and WFG9. There is no trend in the variance of the evaluation time.

Table 3 shows an aggregation of the statistic test of the pairwise comparison. Each group of three columns indicates the compared two methods, while each column shows the count of the results of the statistic test. The “1” column indicates the first method is significantly better than the second one, while the “2” column indicates the second method is significantly better than the first one. If there is no significant difference, “≈” is counted. “Sync.” indicates SP-ELMOEA/D, “IO” indicates AP-ELMOEA/D-IO, and “RO” indicates AP-ELMOEA/D-RO. This result firstly indicates that AP-ELMOEA/D-IO is significantly better than SP-ELMOEA/D in 22 experimental cases of total 84 cases, while SP-ELMOEA/D wins in 5 cases. Comparing SP-ELMOEA/D and AP-ELMOEA/D, AP-ELMOEA/D-RO is significantly better than SP-ELMOEA/D in 13 experimental cases, while SP-ELMOEA/D wins in 6 cases. Comparing AP-ELMOEA/D-IO and AP-ELMOEA/D-RO, there is no significant difference in most cases, but in 10 cases, AP-ELMOEA/D-IO is significantly

better than AP-ELMOEA/D-RO, while AP-ELMOEA/D-RO wins in only 2 cases.

These results reveal that the integration of the surrogate-assisted EA with the asynchronous evolution scheme improves the search capability, even taking no account of the computational time.

6.2 Hypervolume at the maximum elapsed simulation time

The mean normalized HV values at the maximum elapsed simulation time are shown in Table 4. The maximum mean values are shown in the bold style, and the results of the statistic test are specified as the superscript character. Both of them are the same as Table 2. Table 5 aggregates the statistic test of the pairwise comparison as same as Table 3.

First, let us compare the results of the parallel and the sequential ELMOEA/Ds. It is confirmed that the parallel ELMOEA/Ds significantly outperforms the sequential ELMOEA/D for all benchmarks and all c_v values. From this, the parallelization of ELMOEA/D contributes to reducing the computational time for the optimization.

Then, let us compare the parallelization schemes, i.e., the synchronous and the asynchronous parallel ELMOEA/Ds. We can confirm a similar tendency to Table 2 from these results. The HV value of AP-ELMOEA/D-IO is significantly better than or equal to SP-ELMOEA/D in all benchmark problems. On the other hand, AP-ELMOEA/D-RO is also significantly better than or equal to SP-ELMOEA/D especially when the variance of the evaluation time is large. In particular, AP-ELMOEA/D-IO performs the highest mean HV values in most experimental cases. The results in Table 5 indicate that AP-ELMOEA/D-IO is significantly better than SP-ELMOEA/D in 30 experimental cases of total of 84 cases. In comparison, AP-ELMOEA/D-RO is significantly better than SP-ELMOEA/D in 15 experimental cases. Comparing AP-ELMOEA/D-IO and AP-ELMOEA/D-RO, AP-ELMOEA/D-IO is significantly better in 15 experimental cases.

Next, we compare the difference in performance as the variance of the evaluation time changes. When the variance of the evaluation time is small, the difference between SP-ELMOEA/D and the AP-ELMOEA/D variants is small. In particular, SP-ELMOEA/D outperforms AP-ELMOEA/D-RO in a few benchmarks. In contrast, when the variance of the evaluation time is large, the cases of AP-ELMOEA/Ds outperforming SP-ELMOEA/D increase. This result shows that as the variance of the evaluation time increases, the AP-ELMOEA/D variants shows higher performance.

From these results, it is revealed that the asynchronous evolution scheme also improves the search efficiency in terms of computational time. However, in some benchmark problems, there is no significant difference between SP-

Table 4 The mean normalized HV values ($\widehat{HV}$) and their standard deviation at the maximum elapsed simulation time in 30 independent trials with the normally-distributed evaluation time

	c_v	SP-ELMOEA/D		AP-ELMOEA/D-IO		AP-ELMOEA/D-RO		Seq. ELMOEA/D	
		Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
ZDT1	0.02	0.9994 ⁴	2.1×10^{-4}	0.9995 ^{1,3,4}	1.6×10^{-4}	0.9994 ⁴	2.3×10^{-4}	0.8263	3.7×10^{-2}
	0.05	0.9994 ⁴	2.3×10^{-4}	0.9995 ⁴	1.7×10^{-4}	0.9995 ⁴	1.5×10^{-4}	0.8263	3.7×10^{-2}
	0.1	0.9994 ⁴	2.1×10^{-4}	0.9995 ^{1,3,4}	1.5×10^{-4}	0.9994 ⁴	2.1×10^{-4}	0.8263	3.7×10^{-2}
	0.2	0.9993 ⁴	2.4×10^{-4}	0.9995 ^{1,4}	2.2×10^{-4}	0.9995 ^{1,4}	1.8×10^{-4}	0.8263	3.7×10^{-2}
ZDT2	0.02	0.9996 ⁴	1.3×10^{-4}	0.9997 ^{1,3,4}	1.2×10^{-4}	0.9996 ⁴	1.1×10^{-4}	0.6108	4.3×10^{-2}
	0.05	0.9996 ⁴	1.7×10^{-4}	0.9997 ^{1,3,4}	9.6×10^{-5}	0.9996 ⁴	1.1×10^{-4}	0.6108	4.3×10^{-2}
	0.1	0.9996 ⁴	1.2×10^{-4}	0.9996 ^{1,4}	1.0×10^{-4}	0.9996 ^{1,4}	9.9×10^{-5}	0.6108	4.3×10^{-2}
	0.2	0.9996 ⁴	1.2×10^{-4}	0.9997 ^{1,4}	8.0×10^{-5}	0.9997 ^{1,4}	8.7×10^{-5}	0.6108	4.3×10^{-2}
ZDT3	0.02	0.9696 ⁴	1.8×10^{-2}	0.9660 ⁴	1.6×10^{-2}	0.9646 ⁴	1.6×10^{-2}	0.6905	7.2×10^{-2}
	0.05	0.9701 ⁴	1.2×10^{-2}	0.9652 ⁴	1.9×10^{-2}	0.9673 ⁴	1.3×10^{-2}	0.6905	7.2×10^{-2}
	0.1	0.9645 ⁴	1.3×10^{-2}	0.9727 ^{1,4}	1.3×10^{-2}	0.9653 ⁴	2.0×10^{-2}	0.6905	7.2×10^{-2}
	0.2	0.9694 ⁴	1.4×10^{-2}	0.9655 ⁴	1.6×10^{-2}	0.9619 ⁴	1.9×10^{-2}	0.6905	7.2×10^{-2}
ZDT4	0.02	0.5465 ⁴	6.5×10^{-2}	0.5711 ⁴	7.4×10^{-2}	0.5536 ⁴	7.6×10^{-2}	0.4259	1.2×10^{-1}
	0.05	0.5651 ⁴	9.0×10^{-2}	0.5573 ⁴	7.5×10^{-2}	0.5602 ⁴	5.1×10^{-2}	0.4156	1.2×10^{-1}
	0.1	0.5454 ⁴	9.8×10^{-2}	0.5840 ⁴	9.0×10^{-2}	0.5457 ⁴	8.0×10^{-2}	0.3979	1.1×10^{-1}
	0.2	0.5227 ⁴	8.6×10^{-2}	0.5800 ^{1,4}	8.1×10^{-2}	0.5625 ^{1,4}	7.1×10^{-2}	0.4235	1.1×10^{-1}
ZDT6	0.02	0.9676 ⁴	3.5×10^{-2}	0.9519 ⁴	4.0×10^{-2}	0.9724 ⁴	3.3×10^{-2}	0.9351	4.1×10^{-2}
	0.05	0.9622 ⁴	4.0×10^{-2}	0.9705 ⁴	3.4×10^{-2}	0.9639 ⁴	3.6×10^{-2}	0.9391	4.6×10^{-2}
	0.1	0.9573 ⁴	3.9×10^{-2}	0.9571 ⁴	3.8×10^{-2}	0.9620 ⁴	3.5×10^{-2}	0.9381	4.3×10^{-2}
	0.2	0.9553 ⁴	3.8×10^{-2}	0.9674 ⁴	3.3×10^{-2}	0.9725 ⁴	2.4×10^{-2}	0.9304	4.2×10^{-2}
WFG1	0.02	0.5965 ⁴	1.2×10^{-3}	0.5970 ^{1,4}	6.4×10^{-4}	0.5971 ^{1,4}	8.5×10^{-4}	0.5128	4.0×10^{-2}
	0.05	0.5968 ⁴	6.6×10^{-4}	0.5966 ⁴	1.2×10^{-3}	0.5969 ⁴	1.0×10^{-3}	0.5128	4.0×10^{-2}
	0.1	0.5964 ⁴	1.1×10^{-3}	0.5968 ⁴	6.2×10^{-4}	0.5969 ^{1,4}	8.8×10^{-4}	0.5128	4.0×10^{-2}
	0.2	0.5960 ⁴	1.0×10^{-3}	0.5971 ^{1,4}	7.0×10^{-4}	0.5970 ^{1,4}	8.8×10^{-4}	0.5128	4.0×10^{-2}
WFG2	0.02	0.7621 ⁴	4.0×10^{-2}	0.7742 ⁴	3.1×10^{-2}	0.7673 ⁴	3.9×10^{-2}	0.6770	2.8×10^{-2}
	0.05	0.7516 ⁴	4.1×10^{-2}	0.7712 ^{1,3,4}	3.9×10^{-2}	0.7414 ⁴	4.6×10^{-2}	0.6770	2.8×10^{-2}
	0.1	0.7584 ⁴	4.0×10^{-2}	0.7700 ⁴	3.5×10^{-2}	0.7715 ⁴	3.2×10^{-2}	0.6770	2.8×10^{-2}
	0.2	0.7400 ⁴	4.3×10^{-2}	0.7707 ^{1,4}	3.9×10^{-2}	0.7567 ^{1,4}	4.0×10^{-2}	0.6770	2.8×10^{-2}
WFG3	0.02	0.7295 ⁴	4.1×10^{-3}	0.7309 ⁴	3.2×10^{-3}	0.7298 ⁴	4.1×10^{-3}	0.6865	2.0×10^{-2}
	0.05	0.7292 ⁴	2.6×10^{-3}	0.7310 ^{1,3,4}	3.7×10^{-3}	0.7295 ⁴	3.7×10^{-3}	0.6865	2.0×10^{-2}
	0.1	0.7282 ⁴	4.3×10^{-3}	0.7303 ^{1,4}	4.1×10^{-3}	0.7293 ⁴	3.8×10^{-3}	0.6865	2.0×10^{-2}
	0.2	0.7283 ⁴	4.2×10^{-3}	0.7302 ⁴	3.6×10^{-3}	0.7295 ⁴	3.4×10^{-3}	0.6865	2.0×10^{-2}
WFG4	0.02	0.8110 ⁴	2.1×10^{-2}	0.8226 ^{1,4}	1.6×10^{-2}	0.8158 ⁴	1.6×10^{-2}	0.6568	6.8×10^{-2}
	0.05	0.8165 ⁴	2.1×10^{-2}	0.8124 ⁴	1.5×10^{-2}	0.8188 ⁴	2.0×10^{-2}	0.6568	6.8×10^{-2}
	0.1	0.8097 ⁴	1.7×10^{-2}	0.8185 ^{1,4}	1.6×10^{-2}	0.8175 ⁴	1.6×10^{-2}	0.6568	6.8×10^{-2}
	0.2	0.8119 ⁴	1.8×10^{-2}	0.8148 ⁴	1.7×10^{-2}	0.8209 ⁴	1.8×10^{-2}	0.6568	6.8×10^{-2}
WFG5	0.02	0.7627 ⁴	2.3×10^{-2}	0.7818 ^{1,4}	3.4×10^{-2}	0.7773 ^{1,4}	2.9×10^{-2}	0.7115	2.8×10^{-2}
	0.05	0.7735 ⁴	3.0×10^{-2}	0.7726 ⁴	2.5×10^{-2}	0.7729 ⁴	3.1×10^{-2}	0.7115	2.8×10^{-2}
	0.1	0.7701 ⁴	2.6×10^{-2}	0.7765 ⁴	2.4×10^{-2}	0.7797 ⁴	2.9×10^{-2}	0.7115	2.8×10^{-2}
	0.2	0.7693 ⁴	2.8×10^{-2}	0.7719 ⁴	2.6×10^{-2}	0.7802 ⁴	2.9×10^{-2}	0.7115	2.8×10^{-2}
WFG6	0.02	0.9240 ⁴	8.6×10^{-3}	0.9300 ^{1,4}	1.1×10^{-2}	0.9309 ^{1,4}	4.2×10^{-3}	0.6455	3.5×10^{-2}
	0.05	0.9225 ⁴	8.0×10^{-3}	0.9277 ^{1,4}	1.3×10^{-2}	0.9309 ^{1,4}	4.1×10^{-3}	0.6455	3.5×10^{-2}
	0.1	0.9206 ⁴	1.3×10^{-2}	0.9302 ^{1,4}	7.7×10^{-3}	0.9305 ^{1,4}	5.1×10^{-3}	0.6455	3.5×10^{-2}
	0.2	0.9170 ⁴	1.5×10^{-2}	0.9309 ^{1,4}	5.9×10^{-3}	0.9301 ^{1,4}	5.2×10^{-3}	0.6455	3.5×10^{-2}

Table 4 (continued)

	c_v	SP-ELMOEA/D		AP-ELMOEA/D-IO		AP-ELMOEA/D-RO		Seq. ELMOEA/D	
		Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
WFG7	0.02	0.8177 ⁴	1.1×10^{-2}	0.8198 ⁴	8.2×10^{-3}	0.8205 ⁴	8.7×10^{-3}	0.7524	3.3×10^{-2}
	0.05	0.8151 ⁴	1.1×10^{-2}	0.8217 ^{1,4}	8.9×10^{-3}	0.8214 ^{1,4}	9.4×10^{-3}	0.7524	3.3×10^{-2}
	0.1	0.8174 ⁴	1.1×10^{-2}	0.8203 ⁴	8.9×10^{-3}	0.8197 ⁴	1.1×10^{-2}	0.7524	3.3×10^{-2}
	0.2	0.8155 ⁴	1.1×10^{-2}	0.8179 ⁴	7.3×10^{-3}	0.8181 ⁴	9.1×10^{-3}	0.7524	3.3×10^{-2}
WFG8	0.02	0.7564 ⁴	10.0×10^{-3}	0.7546 ⁴	1.0×10^{-2}	0.7556 ⁴	1.2×10^{-2}	0.7183	3.3×10^{-2}
	0.05	0.7543 ⁴	1.1×10^{-2}	0.7560 ⁴	9.4×10^{-3}	0.7556 ⁴	8.5×10^{-3}	0.7183	3.3×10^{-2}
	0.1	0.7541 ⁴	1.1×10^{-2}	0.7562 ⁴	9.9×10^{-3}	0.7587 ⁴	9.4×10^{-3}	0.7183	3.3×10^{-2}
	0.2	0.7525 ⁴	1.1×10^{-2}	0.7577 ^{1,4}	9.5×10^{-3}	0.7568 ⁴	9.7×10^{-3}	0.7183	3.3×10^{-2}
WFG9	0.02	0.7425 ⁴	2.4×10^{-2}	0.7394 ⁴	2.9×10^{-2}	0.7467 ⁴	2.4×10^{-2}	0.6569	2.1×10^{-2}
	0.05	0.7447 ⁴	2.9×10^{-2}	0.7462 ⁴	2.2×10^{-2}	0.7500 ⁴	2.5×10^{-2}	0.6569	2.1×10^{-2}
	0.1	0.7437 ⁴	2.9×10^{-2}	0.7502 ^{1,3,4}	1.6×10^{-2}	0.7435 ⁴	2.6×10^{-2}	0.6569	2.1×10^{-2}
	0.2	0.7361 ⁴	2.6×10^{-2}	0.7405 ⁴	2.0×10^{-2}	0.7462 ^{1,4}	2.0×10^{-2}	0.6569	2.1×10^{-2}
DTLZ1	0.02	0.8867 ⁴	2.9×10^{-2}	0.8822 ⁴	2.8×10^{-2}	0.8872 ⁴	2.3×10^{-2}	0.6711	1.7×10^{-1}
	0.05	0.8835 ⁴	3.0×10^{-2}	0.8878 ⁴	2.5×10^{-2}	0.8834 ⁴	3.4×10^{-2}	0.6711	1.7×10^{-1}
	0.1	0.8813 ⁴	2.8×10^{-2}	0.8865 ⁴	2.5×10^{-2}	0.8804 ⁴	2.5×10^{-2}	0.6711	1.7×10^{-1}
	0.2	0.8799 ⁴	2.9×10^{-2}	0.8773 ⁴	4.8×10^{-2}	0.8809 ⁴	2.7×10^{-2}	0.6711	1.7×10^{-1}
DTLZ2	0.02	0.9943 ^{3,4}	2.3×10^{-3}	0.9940 ⁴	2.4×10^{-3}	0.9935 ⁴	2.0×10^{-3}	0.9877	3.5×10^{-3}
	0.05	0.9938 ⁴	2.2×10^{-3}	0.9944 ^{3,4}	2.1×10^{-3}	0.9933 ⁴	2.0×10^{-3}	0.9877	3.5×10^{-3}
	0.1	0.9937 ⁴	2.2×10^{-3}	0.9946 ^{3,4}	2.2×10^{-3}	0.9935 ⁴	2.0×10^{-3}	0.9877	3.5×10^{-3}
	0.2	0.9934 ⁴	2.3×10^{-3}	0.9944 ^{1,3,4}	2.5×10^{-3}	0.9933 ⁴	2.4×10^{-3}	0.9877	3.5×10^{-3}
DTLZ3	0.02	0.3528 ⁴	1.2×10^{-1}	0.4087 ^{1,3,4}	1.4×10^{-1}	0.3563 ⁴	1.5×10^{-1}	0.1642	1.8×10^{-1}
	0.05	0.3485 ⁴	1.2×10^{-1}	0.3830 ⁴	1.3×10^{-1}	0.3764 ⁴	1.4×10^{-1}	0.1642	1.8×10^{-1}
	0.1	0.3405 ⁴	1.3×10^{-1}	0.3690 ⁴	1.3×10^{-1}	0.3587 ⁴	1.4×10^{-1}	0.1642	1.8×10^{-1}
	0.2	0.3234 ⁴	1.4×10^{-1}	0.3760 ⁴	1.4×10^{-1}	0.3580 ⁴	1.4×10^{-1}	0.1642	1.8×10^{-1}
DTLZ4	0.02	0.8546 ⁴	4.5×10^{-2}	0.8548 ⁴	4.4×10^{-2}	0.8561 ⁴	4.8×10^{-2}	0.8500	4.4×10^{-2}
	0.05	0.8546 ⁴	4.5×10^{-2}	0.8549 ⁴	4.5×10^{-2}	0.8562 ⁴	4.8×10^{-2}	0.8500	4.4×10^{-2}
	0.1	0.8561 ⁴	4.8×10^{-2}	0.8548 ⁴	4.5×10^{-2}	0.8564 ⁴	4.9×10^{-2}	0.8500	4.4×10^{-2}
	0.2	0.8561 ⁴	4.8×10^{-2}	0.8563 ⁴	4.8×10^{-2}	0.8565 ⁴	4.8×10^{-2}	0.8500	4.4×10^{-2}
DTLZ5	0.02	0.9957 ^{3,4}	2.3×10^{-3}	0.9953 ⁴	2.4×10^{-3}	0.9949 ⁴	2.0×10^{-3}	0.9891	3.5×10^{-3}
	0.05	0.9952 ⁴	2.2×10^{-3}	0.9958 ^{3,4}	2.1×10^{-3}	0.9947 ⁴	2.0×10^{-3}	0.9891	3.5×10^{-3}
	0.1	0.9951 ⁴	2.3×10^{-3}	0.9959 ^{3,4}	2.2×10^{-3}	0.9948 ⁴	2.0×10^{-3}	0.9891	3.5×10^{-3}
	0.2	0.9948 ⁴	2.3×10^{-3}	0.9958 ^{1,3,4}	2.5×10^{-3}	0.9946 ⁴	2.4×10^{-3}	0.9891	3.5×10^{-3}
DTLZ6	0.02	0.9839 ⁴	3.7×10^{-2}	0.9934 ^{1,4}	1.9×10^{-2}	0.9910 ⁴	2.1×10^{-2}	0.8226	1.3×10^{-1}
	0.05	0.9854 ⁴	3.2×10^{-2}	0.9925 ⁴	2.0×10^{-2}	0.9919 ⁴	2.0×10^{-2}	0.8226	1.3×10^{-1}
	0.1	0.9870 ⁴	3.1×10^{-2}	0.9928 ⁴	1.9×10^{-2}	0.9861 ⁴	3.0×10^{-2}	0.8226	1.3×10^{-1}
	0.2	0.9801 ⁴	5.4×10^{-2}	0.9905 ^{1,4}	2.6×10^{-2}	0.9837 ⁴	3.4×10^{-2}	0.8226	1.3×10^{-1}
DTLZ7	0.02	1.0000 ⁴	4.9×10^{-5}	1.0000 ⁴	1.9×10^{-4}	1.0000 ⁴	2.0×10^{-4}	0.9959	4.5×10^{-3}
	0.05	1.0000 ^{3,4}	1.1×10^{-5}	1.0000 ^{3,4}	8.8×10^{-5}	1.0000 ⁴	1.2×10^{-4}	0.9959	4.5×10^{-3}
	0.1	1.0000 ⁴	2.7×10^{-5}	1.0000 ⁴	5.8×10^{-5}	1.0000 ⁴	4.2×10^{-5}	0.9959	4.5×10^{-3}
	0.2	1.0000 ⁴	2.7×10^{-5}	1.0000 ⁴	1.7×10^{-4}	1.0000 ⁴	6.0×10^{-5}	0.9959	4.5×10^{-3}

The result that is significantly better than another is signed with a number of the method (1: SP-ELMOEA/D, 2: AP-ELMOEA/D-IO, 3: AP-ELMOEA/D-RO, 4: Sequential ELMOEA/D). The highest mean values are shown in the **bold** style

Table 5 An aggregation of the pairwise statistical test using Mann–Whitney U test with a 5% significance level for the result at the maximum elapsed simulation time with the normally-distributed evaluation time

c_v	Sync. vs. IO			Sync. vs. RO			IO vs. RO		
	1	≈	2	1	≈	2	1	≈	2
0.02	0	13	8	2	16	3	3	18	0
0.05	0	16	5	1	18	2	6	15	0
0.1	0	14	7	0	18	3	4	17	0
0.2	0	11	10	0	14	7	2	19	0
Total	0	54	30	3	66	15	15	69	0

c_v	Sync. vs. Seq.			IO vs. Seq.			RO vs. Seq.		
	1	≈	2	1	≈	2	1	≈	2
0.02	21	0	0	21	0	0	21	0	0
0.05	21	0	0	21	0	0	21	0	0
0.1	21	0	0	21	0	0	21	0	0
0.2	21	0	0	21	0	0	21	0	0
Total	84	0	0	84	0	0	84	0	0

If the first algorithm wins against the second one, “1” is counted, while “2” is counted if the second one wins. If two are a tie, “≈” is counted. “Sync.” indicates SP-ELMOEA/D, “IO” indicates AP-ELMOEA/D-IO, “RO” indicates AP-ELMOEA/D-RO, and “Seq.” indicates Sequential ELMOEA/D

ELMOEA/D and the AP-ELMOEA/D variants even though the evaluation time variance is large. The next subsection discusses the details of the transition of the HV values over the elapsed simulation time. In particular, we focus on ZDT4, ZDT6, WFG8, and DTLZ4, because there is no significant difference among SP-ELMOEA/D and the AP-ELMOEA/D variants in these problems.

6.3 Transition of HV over the elapsed simulation time

Figures 1–4 show the transition of the mean normalized HV values over the elapsed simulation time. These figures include two parts: the top part shows the mean normalized HV value transition, while the bottom part shows the results of the statistic test at every simulation time. In the top part, the horizontal axis shows the elapsed simulation time. The vertical axis shows the normalized mean HV value. The solid line shows SP-ELMOEA/D, the dashed line shows AP-ELMOEA/D-IO, and the dash-dot line shows AP-ELMOEA/D-RO. In the bottom part, on the other hand, the horizontal axis is common, while the vertical axis indicates the pair of the comparison, where “1”, “2”, and “3” indicate SP-ELMOEA/D, AP-ELMOEA/D-

IO, and AP-ELMOEA/D-RO, respectively. The “+” mark indicates the first method of the pair significantly outperforms the second one at the corresponding elapsed simulation time. In contrast, the “-” mark indicates the second one outperforms the first one. For example, if the y-axis of the bottom part is “1 v 2” and the mark is “-”, AP-ELMOEA/D-IO significantly outperforms SP-ELMOEA/D at the corresponding elapsed simulation time.

There is no significant difference over time between any pairs of the parallel ELMOEA/D variants in WFG8 and DTLZ4. On the other hand, in ZDT4, the significant difference between AP-ELMOEA/D-IO and SP-ELMOEA/D, and that between AP-ELMOEA/D-RO and SP-ELMOEA/D can be found in the early and the middle term of the evolution when the variance of the evaluation time is low, though there is no significant difference in the final part of the evolution. This tendency can be found in the high variance in ZDT6, where AP-ELMOEA/D-RO significantly outperforms SP-ELMOEA/D in the middle term of the evolution. This result indicates that AP-ELMOEA/D-IO and AP-ELMOEA/D-RO reach to the final HV value quicker than SP-ELMOEA/D, even though the final HV value is almost the same.

These results reveal that the convergence speed of the AP-ELMOEA/D variants is faster than SP-ELMOEA/D even though they finally achieve a similar quality of solutions.

7 Results on benchmark problems with the fitness-correlated evaluation time

This section shows the experimental results on the benchmark problems with the fitness-correlated evaluation time. As same as the previous section, we first compare the HV values of the parallel ELMOEA/D variants at the maximum number of the actual evaluations, then we compare them from the viewpoint of the elapsed simulation time.

Note that in this experiment, we exclude the results on the benchmarks of DTLZ1 and DTLZ3. This is because the range of the first objective function value of these benchmarks is vast (the maximum value is about 100.0), and the evaluation time calculated by Eq. (12) is not appropriate.

7.1 Hypervolume at the maximum number of the actual evaluations

Tables 6 and 7 show the mean normalized HV values and their standard deviations of the competitive methods at the maximum number of the actual evaluations with the fitness-correlated evaluation time. Table 6 shows the result of the positively correlated evaluation time. In contrast,

Fig. 1 ZDT4: The transition of the HV value over the elapsed simulation time. The solid red line, the dashed green line, and the dash-dot blue line show the results of SP-ELMOEA/D, AP-ELMOEA/D-IO, and AP-ELMOEA/D-RO, respectively. (Color figure online)

Table 7 shows the result of the negatively correlated evaluation time. These tables do not include the sequential ELMOEA/D for the same reason as Section 6.1. Tables 8 and 9 show aggregations of the statistic test of the pairwise comparison of the results in Tables 6 and 7. The notations in these tables are the same as in Sect. 6.1.

The experimental results at the maximum number of the actual evaluations show that SP-ELMOEA/D and the AP-ELMOEA/D variants are less significantly different. In particular, compared with the previous section, the number of cases in which the AP-ELMOEA/D variants are significantly better than SP-ELMOEA/D decreases. At the same time, SP-ELMOEA/D significantly outperforms the AP-ELMOEA/D variants in some benchmarks. For example, SP-ELMOEA/D outperforms in the benchmarks of ZDT6, wFG9, and DTLZ3 when the evaluation time positively correlates to the first objective function value, while in ZDT3 when the evaluation time negatively correlates. However, AP-ELMOEA/D-IO comprehensively shows the

highest average HV values in many of the benchmarks, and it is statistically equivalent to or better than the other two methods. These results indicate that although AP-ELMOEA/D-IO is affected by the evaluation time bias, it still performs high search capability.

7.2 Hypervolume at the maximum elapsed simulation time

Tables 10 and 11 show the mean normalized HV values and their standard deviations of the competitive methods at the maximum elapsed simulation time with the fitness-correlated evaluation time. Table 10 shows the result of the positively correlated evaluation time. In contrast, Table 11 shows the result of the negatively correlated evaluation time. Tables 12 and 13 show aggregations of the statistic test of the pairwise comparison of the results in Tables 10 and 11. The notations in these tables are the same as in Sect. 6.2.

Fig. 2 ZDT6: The transition of the HV value over the elapsed simulation time. The solid red line, the dashed green line, and the dash-dot blue line show the results of SP-ELMOEA/D, AP-ELMOEA/D-IO, and AP-ELMOEA/D-RO, respectively. (Color figure online)

First, comparing the sequential ELMOEA/D and the parallel ELMOEA/Ds, it is shown that the parallel variants have significantly better performance than the sequential method in all benchmarks and all evaluation time setting. This result shows that the parallelization of ELMOEA/D contributes to shortening the computational time of the optimization even when the evaluation time is biased.

Next, comparing the three parallel ELMOEA/Ds, the AP-ELMOEA/D variants are significantly better than SP-ELMOEA/D in several benchmarks. However, the number of them is reduced compared to the case of the normally-distributed evaluation time. This is because the search direction of AP-ELMOEA/D is biased to the solution region with a short evaluation time if the evaluation time correlates to the objective function value. This causes to decrease the search capability of AP-ELMOEA/D. However, in terms of the computation time, the waiting time of SP-ELMOEA/D increases because the fitness-correlated evaluation time induces the evaluation time variance depending on the objective function value, thus its computational efficiency decreases. Therefore, although SP-

ELMOEA/D is not affected by the evaluation time bias in search, it cannot outperform the AP-ELMOEA/D variants at the maximum elapsed evaluation time.

In this paper, we use ten slave nodes for the evaluation. However, APEAs are less affected by the evaluation time bias when the number of the slave nodes is small. Therefore, the performance deterioration of AP-ELMOEA/Ds was suppressed in our experiment. However, if we use more slave nodes, it may deteriorate the performance of the AP-ELMOEA/D variants.

8 Results on real-world problems

This section shows the experimental results on real-world problems. As same as the previous section, we first compare the HV values of the parallel ELMOEA/D variants at the maximum number of the actual evaluations, then we compare them from the viewpoint of the elapsed simulation time.

Fig. 3 WFG8: The transition of the HV value over the elapsed simulation time. The solid red line, the dashed green line, and the dash-dot blue line show the results of SP-ELMOEA/D, AP-ELMOEA/D-IO, and AP-ELMOEA/D-RO, respectively. (Color figure online)

8.1 Hypervolume at the maximum number of the actual evaluations

Table 14 shows the mean HV values and their standard deviation of the competitive methods at the maximum number of the actual evaluations with the normally-distributed evaluation time. We represent each problem as an abbreviation. “HX” denotes the heat exchanger problem, “HD” denotes the hydro-dynamics problem, “VP” denotes the vibrating platform problem, and “FP” denotes the facility placement problem. The best results are shown in the bold style. As same as the results on the benchmark problems, we conduct a statistic test with the Mann–Whitney U test with a 5% significance level, and its result is shown in this table as the same style in Table 2.

These results show that, in the heat exchanger problem and the facility placement problem, the AP-ELMOEA/D variants show better performance than SP-ELMOEA/D. Especially in some cases, the asynchronous variants significantly outperform the synchronous one. On the other

hand, in the hydro-dynamics problem and the vibrating platform problem, AP-ELMOEA/D-RO shows worse performance than the other two methods. In particular, SP-ELMOEA/D shows the best mean HV values in the hydro-dynamics problem for all evaluation time variances.

Table 15 shows the mean HV values and their standard deviation of the competitive methods at the maximum number of the actual evaluations with the fitness-correlated evaluation time. The notation in this table is same as Table 14. Even when the evaluation time correlates to the objective function value, the tendency is almost the same as the results with the normally-distributed evaluation time. In particular, in the heat exchanger problem and the facility placement problem, the AP-ELMOEA/D variants show better performance than SP-ELMOEA/D. In the hydro-dynamics problem and the vibrating platform problem, on the other hand, AP-ELMOEA/D-RO is worse than the other two methods.

These results indicate that no clear difference is found in the real-world problems at the maximum number of the

Fig. 4 DTLZ4: The transition of the HV value over the elapsed simulation time. The solid red line, the dashed green line, and the dash-dot blue line show the results of SP-ELMOEA/D, AP-ELMOEA/D-IO, and AP-ELMOEA/D-RO, respectively. (Color figure online)

actual evaluations. However, AP-ELMOEA/D-IO shows averagely better performance than the other two methods, i.e., AP-ELMOEA/D-IO is never significantly worse.

8.2 Hypervolume at the maximum elapsed simulation time

Table 16 shows the mean HV values and their standard deviation of the competitive methods at the maximum elapsed simulation time with the normally-distributed evaluation time. On the other hand, Table 17 shows the result with the fitness-correlated evaluation time. The notations of these tables are the same as Tables 14 and 15. These results include the sequential ELMOEA/D.

First, from the result of the sequential ELMOEA/D, it is shown that the parallel variants all significantly outperform the sequential ELMOEA/D in all problems and both of the evaluation time settings. This result indicates that the parallelization of ELMOEA/D is even useful in real-world problems.

From the comparison of the parallel variants of ELMOEA/D in the normally-distributed evaluation time, no significant difference between three parallel ELMOEA/Ds is found in the heat exchanger and the vibrating platform problems. Note that only in $c_v = 0.2$ in the vibrating platform problem, AP-ELMOEA/D-IO significantly outperforms SP-ELMOEA/D, where the variance of the evaluation time is enormous. On the other hand, in the facility placement problem, the asynchronous variants significantly outperform SP-ELMOEA/D regardless of the variance of the evaluation time. In the hydro-dynamics problem, SP-ELMOEA/D significantly outperforms the AP-ELMOEA/D variants when the variance of the evaluation time is low, i.e., $c_v = \{0.02, 0.05\}$. Meanwhile, SP-ELMOEA/D decreases its performance as the variance of the evaluation time increases even in the hydro-dynamics problem. On the other hand, the AP-ELMOEA/D variants keep their performance as the variance increases. Especially, AP-ELMOEA/D-IO shows the highest mean HV

Table 6 The mean normalized HV values ($\widetilde{HV}$) and their standard deviation at the maximum number of evaluations in 30 independent trials with the fitness-correlated evaluation time ($c_g < 0$)

	c_g	SP-ELMOEA/D		AP-ELMOEA/D-IO		AP-ELMOEA/D-RO	
		Mean	Std.	Mean	Std.	Mean	Std.
ZDT1	0.2	0.9997	1.1×10^{-4}	0.9997	8.3×10^{-5}	0.9996	1.1×10^{-4}
	0.5	0.9997	9.5×10^{-5}	0.9997	9.2×10^{-5}	0.9996	7.3×10^{-5}
ZDT2	0.2	0.9997	1.3×10^{-4}	0.9998 ¹	7.9×10^{-5}	0.9997 ¹	1.1×10^{-4}
	0.5	0.9997	1.3×10^{-4}	0.9997 ¹	9.1×10^{-5}	0.9997	7.6×10^{-5}
ZDT3	0.2	0.9719	1.2×10^{-2}	0.9703	1.4×10^{-2}	0.9712	1.1×10^{-2}
	0.5	0.9751	10.0×10^{-3}	0.9708	1.3×10^{-2}	0.9752	1.3×10^{-2}
ZDT4	0.2	0.6102	5.6×10^{-2}	0.6496 ¹	8.3×10^{-2}	0.6169	6.5×10^{-2}
	0.5	0.6272	7.5×10^{-2}	0.6454	6.9×10^{-2}	0.6313	6.5×10^{-2}
ZDT6	0.2	0.9724	3.3×10^{-2}	0.9706	3.4×10^{-2}	0.9676	4.0×10^{-2}
	0.5	0.9741 ³	3.5×10^{-2}	0.9640	4.3×10^{-2}	0.9614	3.8×10^{-2}
WFG1	0.2	0.5977	6.4×10^{-4}	0.5980 ¹	6.5×10^{-4}	0.5981 ¹	6.6×10^{-4}
	0.5	0.5977	7.6×10^{-4}	0.5980	6.9×10^{-4}	0.5980	6.7×10^{-4}
WFG2	0.2	0.7908	3.6×10^{-2}	0.8049	3.2×10^{-2}	0.8031	3.0×10^{-2}
	0.5	0.7915	3.9×10^{-2}	0.8039	2.9×10^{-2}	0.8122 ¹	2.6×10^{-2}
WFG3	0.2	0.7347	2.7×10^{-3}	0.7363 ^{1,3}	2.4×10^{-3}	0.7345	3.0×10^{-3}
	0.5	0.7349	3.0×10^{-3}	0.7366 ¹	3.3×10^{-3}	0.7354	3.6×10^{-3}
WFG4	0.2	0.8290	1.7×10^{-2}	0.8348	1.8×10^{-2}	0.8299	1.3×10^{-2}
	0.5	0.8282	1.7×10^{-2}	0.8302	1.6×10^{-2}	0.8330	1.7×10^{-2}
WFG5	0.2	0.7897	2.7×10^{-2}	0.8020	2.9×10^{-2}	0.7928	2.7×10^{-2}
	0.5	0.7936	2.7×10^{-2}	0.7968	2.9×10^{-2}	0.7987	2.1×10^{-2}
WFG6	0.2	0.9397	2.1×10^{-3}	0.9423 ¹	1.2×10^{-3}	0.9421 ¹	1.6×10^{-3}
	0.5	0.9401	2.2×10^{-3}	0.9422 ^{1,3}	1.5×10^{-3}	0.9416 ¹	1.6×10^{-3}
WFG7	0.2	0.8289	9.0×10^{-3}	0.8316	8.7×10^{-3}	0.8326	9.2×10^{-3}
	0.5	0.8291	7.4×10^{-3}	0.8319	8.0×10^{-3}	0.8308	7.6×10^{-3}
WFG8	0.2	0.7569	1.1×10^{-2}	0.7606	9.7×10^{-3}	0.7605	1.0×10^{-2}
	0.5	0.7567	1.2×10^{-2}	0.7590	1.0×10^{-2}	0.7616	1.0×10^{-2}
WFG9	0.2	0.7846	2.6×10^{-2}	0.7799	2.7×10^{-2}	0.7888	2.5×10^{-2}
	0.5	0.7887 ²	2.6×10^{-2}	0.7753	2.1×10^{-2}	0.7769	2.2×10^{-2}
DTLZ2	0.2	0.9967	5.6×10^{-4}	0.9967 ³	8.5×10^{-4}	0.9963	7.9×10^{-4}
	0.5	0.9967 ³	6.6×10^{-4}	0.9968 ³	6.7×10^{-4}	0.9963	9.0×10^{-4}
DTLZ4	0.2	0.8731	6.7×10^{-2}	0.8674	6.1×10^{-2}	0.8771	7.0×10^{-2}
	0.5	0.8823	7.5×10^{-2}	0.8702	6.3×10^{-2}	0.8683	6.4×10^{-2}
DTLZ5	0.2	0.9981	5.6×10^{-4}	0.9980 ³	8.5×10^{-4}	0.9977	7.9×10^{-4}
	0.5	0.9981 ³	6.6×10^{-4}	0.9982 ³	6.7×10^{-4}	0.9976	9.0×10^{-4}
DTLZ6	0.2	0.9995	1.2×10^{-3}	1.0002 ¹	6.4×10^{-4}	0.9999	8.8×10^{-4}
	0.5	0.9994	1.4×10^{-3}	1.0001 ¹	7.6×10^{-4}	1.0002 ¹	6.5×10^{-4}
DTLZ7	0.2	1.0001 ²	1.0×10^{-5}	1.0001	3.3×10^{-6}	1.0000 ²	2.9×10^{-5}
	0.5	1.0001	1.5×10^{-5}	1.0001	6.1×10^{-6}	1.0001	3.7×10^{-6}

The evaluation time of solutions *increases* as the 1st objective function value *increases*. The result that is significantly better than another is signed with a number of the method (1: SP-ELMOEA/D, 2: AP-ELMOEA/D-IO, 3: AP-ELMOEA/D-RO). The highest mean values are shown in the **bold** style

Table 7 The mean normalized HV values ($\widetilde{HV}$) and their standard deviation at the maximum number of evaluations in 30 independent trials with the fitness-correlated evaluation time ($c_g < 0$)

	c_g	SP-ELMOEA/D		AP-ELMOEA/D-IO		AP-ELMOEA/D-RO	
		Mean	Std.	Mean	Std.	Mean	Std.
ZDT1	-0.2	0.9996	7.9×10^{-5}	0.9996 ^{1,3}	7.6×10^{-5}	0.9996	1.0×10^{-4}
	-0.5	0.9996	1.2×10^{-4}	0.9996	8.3×10^{-5}	0.9996	8.9×10^{-5}
ZDT2	-0.2	0.9997	9.5×10^{-5}	0.9997	8.8×10^{-5}	0.9997	9.5×10^{-5}
	-0.5	0.9997	1.2×10^{-4}	0.9998 ¹	9.5×10^{-5}	0.9997	1.1×10^{-4}
ZDT3	-0.2	0.9797 ³	1.0×10^{-2}	0.9739 ³	1.4×10^{-2}	0.9679	1.3×10^{-2}
	-0.5	0.9781 ²	8.9×10^{-3}	0.9729	1.1×10^{-2}	0.9753	1.3×10^{-2}
ZDT4	-0.2	0.6216	5.4×10^{-2}	0.6260	6.7×10^{-2}	0.6571 ¹	7.4×10^{-2}
	-0.5	0.6269	6.8×10^{-2}	0.6247	7.2×10^{-2}	0.6367	6.8×10^{-2}
ZDT6	-0.2	0.9644	3.9×10^{-2}	0.9677	3.6×10^{-2}	0.9617	4.3×10^{-2}
	-0.5	0.9559	3.4×10^{-2}	0.9664	3.6×10^{-2}	0.9611	3.8×10^{-2}
WFG1	-0.2	0.5978	6.6×10^{-4}	0.5980	5.1×10^{-4}	0.5980	4.6×10^{-4}
	-0.5	0.5976	8.4×10^{-4}	0.5978	6.4×10^{-4}	0.5981 ¹	5.7×10^{-4}
WFG2	-0.2	0.7877	4.3×10^{-2}	0.8106 ¹	2.9×10^{-2}	0.8059	3.1×10^{-2}
	-0.5	0.7937	3.8×10^{-2}	0.8115 ¹	3.0×10^{-2}	0.8069	3.1×10^{-2}
WFG3	-0.2	0.7353	3.0×10^{-3}	0.7371 ¹	2.5×10^{-3}	0.7356	3.7×10^{-3}
	-0.5	0.7361	3.2×10^{-3}	0.7374 ³	2.5×10^{-3}	0.7355	2.5×10^{-3}
WFG4	-0.2	0.8319	1.9×10^{-2}	0.8361	1.2×10^{-2}	0.8386	1.7×10^{-2}
	-0.5	0.8290	1.8×10^{-2}	0.8367	1.7×10^{-2}	0.8347	1.7×10^{-2}
WFG5	-0.2	0.7921	3.1×10^{-2}	0.7964	2.6×10^{-2}	0.7956	2.1×10^{-2}
	-0.5	0.7916	3.0×10^{-2}	0.7978	2.7×10^{-2}	0.7901	2.7×10^{-2}
WFG6	-0.2	0.9406	1.6×10^{-3}	0.9418 ¹	1.8×10^{-3}	0.9421 ¹	1.4×10^{-3}
	-0.5	0.9401	1.6×10^{-3}	0.9422 ¹	1.3×10^{-3}	0.9421 ¹	1.4×10^{-3}
WFG7	-0.2	0.8273	8.1×10^{-3}	0.8315	8.9×10^{-3}	0.8310	8.1×10^{-3}
	-0.5	0.8292	8.9×10^{-3}	0.8307	8.3×10^{-3}	0.8313	8.3×10^{-3}
WFG8	-0.2	0.7559	9.4×10^{-3}	0.7599	9.1×10^{-3}	0.7581	8.6×10^{-3}
	-0.5	0.7567	8.4×10^{-3}	0.7603	8.4×10^{-3}	0.7581	9.5×10^{-3}
WFG9	-0.2	0.7771	2.1×10^{-2}	0.7738	2.0×10^{-2}	0.7831 ²	2.5×10^{-2}
	-0.5	0.7801	2.1×10^{-2}	0.7788	2.7×10^{-2}	0.7814	2.4×10^{-2}
DTLZ2	-0.2	0.9960	1.1×10^{-3}	0.9966 ¹	7.1×10^{-4}	0.9963	8.1×10^{-4}
	-0.5	0.9960	1.3×10^{-3}	0.9964	8.5×10^{-4}	0.9963	7.6×10^{-4}
DTLZ4	-0.2	0.8802	6.9×10^{-2}	0.8624	5.8×10^{-2}	0.8674	6.4×10^{-2}
	-0.5	0.8801	7.1×10^{-2}	0.8768	7.1×10^{-2}	0.8686	6.2×10^{-2}
DTLZ5	-0.2	0.9974	1.1×10^{-3}	0.9979 ¹	7.1×10^{-4}	0.9977	8.1×10^{-4}
	-0.5	0.9973	1.3×10^{-3}	0.9978	8.5×10^{-4}	0.9977	7.7×10^{-4}
DTLZ6	-0.2	0.9995	1.1×10^{-3}	1.0002 ¹	5.2×10^{-4}	1.0003 ¹	4.8×10^{-4}
	-0.5	0.9993	1.5×10^{-3}	0.9999	7.5×10^{-4}	1.0002 ¹	6.7×10^{-4}
DTLZ7	-0.2	1.0001	1.4×10^{-5}	1.0001	3.9×10^{-6}	1.0001	3.4×10^{-6}
	-0.5	1.0001	6.4×10^{-6}	1.0001	3.8×10^{-6}	1.0000	1.7×10^{-5}

The evaluation time of solutions *increases* as the 1st objective function value *increases*. The result that is significantly better than another is signed with a number of the method (1: SP-ELMOEA/D, 2: AP-ELMOEA/D-IO, 3: AP-ELMOEA/D-RO). The highest mean values are shown in the **bold** style

Table 8 An aggregation of the pairwise statistical test by using Mann–Whitney U test with 5% of significance level for the result at the maximum number of the actual evaluations with the fitness-correlated evaluation time ($c_g > 0$)

c_g	Sync. vs. IO			Sync. vs. RO			IO vs. RO		
	1	≈	2	1	≈	2	1	≈	2
0.2	1	12	6	0	16	3	3	15	1
0.5	0	15	4	3	13	3	3	16	0
Total	1	27	10	3	29	6	6	31	1

If the first algorithm wins against the second one, “1” is counted, while “2” is counted if the second one wins. If two are a tie, “≈” is counted. “Sync.” indicates SP-ELMOEA/D, “IO” indicates AP-ELMOEA/D-IO, and “RO” indicates AP-ELMOEA/D-RO

Table 9 An aggregation of the pairwise statistical test using Mann–Whitney U test with a 5% significance level for the result at the maximum number of the actual evaluations with the fitness-correlated evaluation time ($c_g < 0$)

c_g	Sync. vs. IO			Sync. vs. RO			IO vs. RO		
	1	≈	2	1	≈	2	1	≈	2
−0.2	0	12	7	1	15	3	2	16	1
−0.5	1	15	3	0	16	3	1	18	0
Total	1	27	10	1	31	6	3	34	1

If the first algorithm wins against the second one, “1” is counted, while “2” is counted if the second one wins. If two are a tie, “≈” is counted. “Sync.” indicates SP-ELMOEA/D, “IO” indicates AP-ELMOEA/D-IO, and “RO” indicates AP-ELMOEA/D-RO

values in most cases and significantly outperforms the other two methods in some of the cases.

Finally, we compare the parallel variants of ELMOEA/D in the fitness-correlated evaluation. First, there is no significant difference between SP-ELMOEA/D and AP-ELMOEA/D-IO in all cases except for the facility placement problem with $c_g = 0.5$. On the other hand, although AP-ELMOEA/D-RO shows better performance only in the heat exchanger problem, it is inferior to AP-ELMOEA/D-IO in the other problems. In total, AP-ELMOEA/D-IO shows the highest mean HV values in most cases and significantly outperforms the other two methods in some of the cases.

These results reveal that AP-ELMOEA/D-IO shows high search performance in all real-world problems and all evaluation time settings. Although SP-ELMOEA/D shows the high performance when the variance of the evaluation time is low, its performance decreases as the variance increases.

9 Conclusion

This paper introduced a parallel evaluation scheme into a surrogate-assisted multi-objective evolutionary algorithm (MOEA) for further reducing the optimization time. As a surrogate-assisted MOEA, this paper used extreme learning assisted MOEA/D (ELMOEA/D) that uses an extreme learning machine (ELM) as a surrogate model and MOEA/D-DE as an optimizer. On the other hand, we considered two different schemes of the parallelization, i.e., a synchronous evolution scheme and an asynchronous evolution

Table 10 The mean normalized HV values ($\widetilde{HV}$) and their standard deviation at the maximum elapsed simulation time in 30 independent trials with the fitness-correlated evaluation time ($c_g > 0$)

	c_g	SP-ELMOEA/D		AP-ELMOEA/D-IO		AP-ELMOEA/D-RO		Seq. ELMOEA/D	
		Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
ZDT1	0.2	0.9995 ⁴	1.7×10^{-4}	0.9996 ⁴	1.5×10^{-4}	0.9995 ⁴	1.8×10^{-4}	0.8263	3.7×10^{-2}
	0.5	0.9995 ⁴	1.9×10^{-4}	0.9996 ^{1,4}	1.4×10^{-4}	0.9995 ⁴	1.1×10^{-4}	0.8263	3.7×10^{-2}
ZDT2	0.2	0.9996 ⁴	1.5×10^{-4}	0.9997 ^{1,3,4}	8.8×10^{-5}	0.9996 ⁴	1.3×10^{-4}	0.6108	4.3×10^{-2}
	0.5	0.9996 ⁴	1.3×10^{-4}	0.9997 ^{1,4}	9.3×10^{-5}	0.9997 ^{1,4}	8.4×10^{-5}	0.6108	4.3×10^{-2}
ZDT3	0.2	0.9684 ⁴	1.3×10^{-2}	0.9666 ⁴	1.4×10^{-2}	0.9650 ⁴	1.7×10^{-2}	0.6905	7.2×10^{-2}
	0.5	0.9680 ⁴	1.3×10^{-2}	0.9671 ⁴	1.2×10^{-2}	0.9690 ⁴	1.8×10^{-2}	0.6905	7.2×10^{-2}
ZDT4	0.2	0.5547 ⁴	6.5×10^{-2}	0.5937 ^{1,4}	8.2×10^{-2}	0.5677 ⁴	7.1×10^{-2}	0.3957	1.2×10^{-1}
	0.5	0.5445 ⁴	6.3×10^{-2}	0.5921 ^{1,4}	7.6×10^{-2}	0.6043 ^{1,4}	7.6×10^{-2}	0.4106	1.1×10^{-1}
ZDT6	0.2	0.9687 ⁴	3.5×10^{-2}	0.9693 ⁴	3.5×10^{-2}	0.9610 ⁴	4.1×10^{-2}	0.9430	4.3×10^{-2}
	0.5	0.9712 ⁴	3.6×10^{-2}	0.9629 ⁴	4.3×10^{-2}	0.9591	3.8×10^{-2}	0.9420	4.5×10^{-2}
WFG1	0.2	0.5964 ⁴	1.0×10^{-3}	0.5966 ⁴	9.3×10^{-4}	0.5968 ^{1,4}	7.1×10^{-4}	0.5128	4.0×10^{-2}
	0.5	0.5960 ⁴	1.1×10^{-3}	0.5965 ^{1,4}	8.6×10^{-4}	0.5965 ⁴	7.9×10^{-4}	0.5128	4.0×10^{-2}

Table 10 (continued)

	c_g	SP-ELMOEA/D		AP-ELMOEA/D-IO		AP-ELMOEA/D-RO		Seq. ELMOEA/D	
		Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
WFG2	0.2	0.7474 ⁴	4.3×10^{-2}	0.7657 ^{1,4}	3.9×10^{-2}	0.7727 ^{1,4}	3.3×10^{-2}	0.6770	2.8×10^{-2}
	0.5	0.7409 ⁴	4.3×10^{-2}	0.7688 ^{1,4}	3.7×10^{-2}	0.7674 ^{1,4}	3.8×10^{-2}	0.6770	2.8×10^{-2}
WFG3	0.2	0.7279 ⁴	3.5×10^{-3}	0.7300 ^{1,3,4}	3.6×10^{-3}	0.7280 ⁴	3.6×10^{-3}	0.6865	2.0×10^{-2}
	0.5	0.7273 ⁴	3.8×10^{-3}	0.7298 ^{1,4}	4.4×10^{-3}	0.7290 ⁴	4.4×10^{-3}	0.6865	2.0×10^{-2}
WFG4	0.2	0.8110 ⁴	2.1×10^{-2}	0.8184 ⁴	2.0×10^{-2}	0.8101 ⁴	1.7×10^{-2}	0.6568	6.8×10^{-2}
	0.5	0.8038 ⁴	2.4×10^{-2}	0.8122 ⁴	1.9×10^{-2}	0.8104 ⁴	1.9×10^{-2}	0.6568	6.8×10^{-2}
WFG5	0.2	0.7691 ⁴	3.1×10^{-2}	0.7802 ⁴	3.2×10^{-2}	0.7718 ⁴	2.8×10^{-2}	0.7115	2.8×10^{-2}
	0.5	0.7652 ⁴	3.1×10^{-2}	0.7712 ⁴	2.9×10^{-2}	0.7783 ^{1,4}	2.7×10^{-2}	0.7115	2.8×10^{-2}
WFG6	0.2	0.9129 ⁴	1.7×10^{-2}	0.9283 ^{1,4}	6.9×10^{-3}	0.9274 ^{1,4}	8.5×10^{-3}	0.6455	3.5×10^{-2}
	0.5	0.9037 ⁴	2.7×10^{-2}	0.9212 ^{1,4}	1.4×10^{-2}	0.9203 ^{1,4}	1.2×10^{-2}	0.6455	3.5×10^{-2}
WFG7	0.2	0.8150 ⁴	1.0×10^{-2}	0.8196 ^{1,4}	9.2×10^{-3}	0.8193 ^{1,4}	9.6×10^{-3}	0.7524	3.3×10^{-2}
	0.5	0.8130 ⁴	1.1×10^{-2}	0.8209 ^{1,4}	7.0×10^{-3}	0.8192 ^{1,4}	9.0×10^{-3}	0.7524	3.3×10^{-2}
WFG8	0.2	0.7541 ⁴	1.2×10^{-2}	0.7582 ⁴	1.1×10^{-2}	0.7583 ⁴	1.1×10^{-2}	0.7183	3.3×10^{-2}
	0.5	0.7537 ⁴	1.2×10^{-2}	0.7564 ⁴	9.6×10^{-3}	0.7592 ⁴	1.1×10^{-2}	0.7183	3.3×10^{-2}
WFG9	0.2	0.7323 ⁴	2.1×10^{-2}	0.7387 ⁴	1.8×10^{-2}	0.7460 ^{1,4}	2.7×10^{-2}	0.6569	2.1×10^{-2}
	0.5	0.7267 ⁴	2.1×10^{-2}	0.7331 ⁴	2.0×10^{-2}	0.7375 ^{1,4}	2.5×10^{-2}	0.6569	2.1×10^{-2}
DTLZ2	0.2	0.9944 ⁴	1.9×10^{-3}	0.9942 ⁴	2.6×10^{-3}	0.9939 ⁴	2.2×10^{-3}	0.9877	3.5×10^{-3}
	0.5	0.9943 ⁴	1.9×10^{-3}	0.9948 ⁴	2.0×10^{-3}	0.9942 ⁴	2.0×10^{-3}	0.9877	3.5×10^{-3}
DTLZ4	0.2	0.8563 ⁴	4.8×10^{-2}	0.8549 ⁴	4.5×10^{-2}	0.8566 ⁴	4.8×10^{-2}	0.8500	4.4×10^{-2}
	0.5	0.8563 ⁴	4.8×10^{-2}	0.8549 ⁴	4.5×10^{-2}	0.8575 ⁴	4.9×10^{-2}	0.8500	4.4×10^{-2}
DTLZ5	0.2	0.9958 ⁴	1.9×10^{-3}	0.9955 ⁴	2.6×10^{-3}	0.9952 ⁴	2.2×10^{-3}	0.9891	3.5×10^{-3}
	0.5	0.9956 ⁴	1.9×10^{-3}	0.9962 ⁴	2.0×10^{-3}	0.9956 ⁴	2.0×10^{-3}	0.9891	3.5×10^{-3}
DTLZ6	0.2	0.9901 ⁴	2.0×10^{-2}	0.9922 ⁴	1.9×10^{-2}	0.9875 ⁴	2.5×10^{-2}	0.8226	1.3×10^{-1}
	0.5	0.9755 ⁴	5.2×10^{-2}	0.9919 ^{1,4}	1.9×10^{-2}	0.9892 ^{1,4}	2.7×10^{-2}	0.8226	1.3×10^{-1}
DTLZ7	0.2	1.0000 ^{2,4}	6.1×10^{-5}	1.0000 ⁴	6.0×10^{-5}	1.0000 ⁴	6.7×10^{-5}	0.9959	4.5×10^{-3}
	0.5	1.0000 ⁴	3.5×10^{-5}	1.0000 ⁴	3.2×10^{-5}	1.0000 ⁴	7.2×10^{-5}	0.9959	4.5×10^{-3}

The evaluation time of solutions *increases* as the 1st objective function value *increases*. The result that is significantly better than another is signed with a number of the method (1: SP-ELMOEA/D, 2: AP-ELMOEA/D-IO, 3: AP-ELMOEA/D-RO, 4: Sequential ELMOEA/D). The highest mean values are shown in the **bold** style

Table 11 The mean normalized HV values ($\widehat{HV}$) and their standard deviation at the maximum elapsed simulation time in 30 independent trials with the fitness-correlated evaluation time ($c_g < 0$)

	c_g	SP-ELMOEA/D		AP-ELMOEA/D-IO		AP-ELMOEA/D-RO		Seq. ELMOEA/D	
		Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
ZDT1	-0.2	0.9994 ⁴	1.6×10^{-4}	0.9995 ^{1,4}	1.5×10^{-4}	0.9994 ⁴	2.2×10^{-4}	0.8263	3.7×10^{-2}
	-0.5	0.9993 ⁴	3.3×10^{-4}	0.9994 ⁴	1.8×10^{-4}	0.9995 ⁴	1.9×10^{-4}	0.8263	3.7×10^{-2}
ZDT2	-0.2	0.9996 ⁴	1.3×10^{-4}	0.9997 ^{1,3,4}	9.9×10^{-5}	0.9996 ⁴	1.3×10^{-4}	0.6108	4.3×10^{-2}
	-0.5	0.9995 ⁴	1.7×10^{-4}	0.9996 ^{1,3,4}	1.1×10^{-4}	0.9996 ^{1,4}	1.4×10^{-4}	0.6108	4.3×10^{-2}
ZDT3	-0.2	0.9722 ^{3,4}	1.4×10^{-2}	0.9677 ^{3,4}	1.5×10^{-2}	0.9602 ⁴	1.9×10^{-2}	0.6905	7.2×10^{-2}
	-0.5	0.9673 ⁴	1.9×10^{-2}	0.9667 ⁴	1.8×10^{-2}	0.9659 ⁴	1.8×10^{-2}	0.6905	7.2×10^{-2}
ZDT4	-0.2	0.5546 ⁴	7.1×10^{-2}	0.5466 ⁴	7.6×10^{-2}	0.5579 ⁴	8.1×10^{-2}	0.4417	1.2×10^{-1}
	-0.5	0.5460 ⁴	9.9×10^{-2}	0.5654 ⁴	8.0×10^{-2}	0.5790 ⁴	8.6×10^{-2}	0.4055	1.2×10^{-1}
ZDT6	-0.2	0.9603 ⁴	4.0×10^{-2}	0.9652 ⁴	3.6×10^{-2}	0.9569 ⁴	4.4×10^{-2}	0.9235	3.7×10^{-2}
	-0.5	0.9505 ⁴	3.5×10^{-2}	0.9655 ⁴	3.6×10^{-2}	0.9560 ⁴	4.0×10^{-2}	0.9244	3.7×10^{-2}
WFG1	-0.2	0.5966 ⁴	1.1×10^{-3}	0.5974 ^{1,4}	5.4×10^{-4}	0.5974 ^{1,4}	5.4×10^{-4}	0.5128	4.0×10^{-2}
	-0.5	0.5969 ⁴	9.1×10^{-4}	0.5975 ^{1,4}	6.3×10^{-4}	0.5978 ^{1,4}	6.2×10^{-4}	0.5128	4.0×10^{-2}
WFG2	-0.2	0.7439 ⁴	4.7×10^{-2}	0.7706 ^{1,4}	3.8×10^{-2}	0.7658 ^{1,4}	4.3×10^{-2}	0.6770	2.8×10^{-2}
	-0.5	0.7439 ⁴	4.4×10^{-2}	0.7625 ^{1,4}	3.9×10^{-2}	0.7707 ^{1,4}	4.0×10^{-2}	0.6770	2.8×10^{-2}
WFG3	-0.2	0.7290 ⁴	3.8×10^{-3}	0.7310 ^{1,4}	3.3×10^{-3}	0.7300 ⁴	4.2×10^{-3}	0.6865	2.0×10^{-2}
	-0.5	0.7284 ⁴	3.7×10^{-3}	0.7324 ^{1,3,4}	3.3×10^{-3}	0.7306 ^{1,4}	3.5×10^{-3}	0.6865	2.0×10^{-2}
WFG4	-0.2	0.8121 ⁴	2.0×10^{-2}	0.8226 ^{1,4}	1.6×10^{-2}	0.8155 ⁴	1.8×10^{-2}	0.6568	6.8×10^{-2}
	-0.5	0.8098 ⁴	1.8×10^{-2}	0.8271 ^{1,4}	1.9×10^{-2}	0.8214 ^{1,4}	1.8×10^{-2}	0.6568	6.8×10^{-2}
WFG5	-0.2	0.7634 ⁴	2.5×10^{-2}	0.7758 ^{1,4}	2.6×10^{-2}	0.7827 ^{1,4}	2.1×10^{-2}	0.7115	2.8×10^{-2}
	-0.5	0.7690 ⁴	2.8×10^{-2}	0.7805 ^{1,4}	2.8×10^{-2}	0.7725 ⁴	2.5×10^{-2}	0.7115	2.8×10^{-2}
WFG6	-0.2	0.9216 ⁴	8.0×10^{-3}	0.9332 ^{1,4}	5.1×10^{-3}	0.9328 ^{1,4}	5.6×10^{-3}	0.6455	3.5×10^{-2}
	-0.5	0.9186 ⁴	1.2×10^{-2}	0.9372 ^{1,4}	2.6×10^{-3}	0.9364 ^{1,4}	3.1×10^{-3}	0.6455	3.5×10^{-2}
WFG7	-0.2	0.8139 ⁴	9.5×10^{-3}	0.8229 ^{1,3,4}	9.6×10^{-3}	0.8177 ⁴	8.0×10^{-3}	0.7524	3.3×10^{-2}
	-0.5	0.8084 ⁴	1.1×10^{-2}	0.8188 ^{1,4}	1.0×10^{-2}	0.8175 ^{1,4}	9.4×10^{-3}	0.7524	3.3×10^{-2}
WFG8	-0.2	0.7538 ⁴	9.8×10^{-3}	0.7578 ⁴	9.4×10^{-3}	0.7559 ⁴	9.3×10^{-3}	0.7183	3.3×10^{-2}
	-0.5	0.7531 ⁴	9.9×10^{-3}	0.7584 ^{1,4}	9.0×10^{-3}	0.7559 ⁴	9.5×10^{-3}	0.7183	3.3×10^{-2}
WFG9	-0.2	0.7435 ⁴	2.4×10^{-2}	0.7457 ⁴	1.7×10^{-2}	0.7468 ⁴	2.3×10^{-2}	0.6569	2.1×10^{-2}
	-0.5	0.7333 ⁴	2.8×10^{-2}	0.7500 ^{1,4}	2.2×10^{-2}	0.7557 ^{1,4}	2.7×10^{-2}	0.6569	2.1×10^{-2}
DTLZ2	-0.2	0.9934 ⁴	3.0×10^{-3}	0.9931 ⁴	2.8×10^{-3}	0.9933 ⁴	1.9×10^{-3}	0.9877	3.5×10^{-3}
	-0.5	0.9921 ⁴	3.1×10^{-3}	0.9932 ⁴	2.5×10^{-3}	0.9930 ⁴	2.1×10^{-3}	0.9877	3.5×10^{-3}
DTLZ4	-0.2	0.8547 ⁴	4.5×10^{-2}	0.8548 ⁴	4.5×10^{-2}	0.8562 ⁴	4.7×10^{-2}	0.8500	4.4×10^{-2}
	-0.5	0.8547 ⁴	4.5×10^{-2}	0.8599 ⁴	5.4×10^{-2}	0.8561 ⁴	4.8×10^{-2}	0.8500	4.4×10^{-2}
DTLZ5	-0.2	0.9947 ⁴	3.0×10^{-3}	0.9944 ⁴	2.8×10^{-3}	0.9947 ⁴	1.9×10^{-3}	0.9891	3.5×10^{-3}
	-0.5	0.9935 ⁴	3.1×10^{-3}	0.9945 ⁴	2.5×10^{-3}	0.9944 ⁴	2.1×10^{-3}	0.9891	3.5×10^{-3}
DTLZ6	-0.2	0.9839 ⁴	3.8×10^{-2}	0.9972 ^{1,4}	4.9×10^{-3}	0.9861 ⁴	2.9×10^{-2}	0.8226	1.3×10^{-1}
	-0.5	0.9804 ⁴	3.9×10^{-2}	0.9936 ^{1,4}	1.5×10^{-2}	0.9932 ^{1,4}	1.8×10^{-2}	0.8226	1.3×10^{-1}
DTLZ7	-0.2	1.0000 ⁴	1.4×10^{-4}	1.0000 ^{3,4}	6.4×10^{-5}	1.0000 ⁴	7.8×10^{-5}	0.9959	4.5×10^{-3}
	-0.5	1.0000 ^{3,4}	1.2×10^{-4}	0.9996 ⁴	1.9×10^{-3}	0.9999 ⁴	2.1×10^{-4}	0.9959	4.5×10^{-3}

The evaluation time of solutions *increases* as the 1st objective function value *decreases*. The result that is significantly better than another is signed with a number of the method (1: SP-ELMOEA/D, 2: AP-ELMOEA/D-IO, 3: AP-ELMOEA/D-RO, 4: Sequential ELMOEA/D). The highest mean values are shown in the **bold** style

Table 12 An aggregation of the pairwise statistical test by using Mann–Whitney U test with 5% of significance level for the result at the maximum elapsed simulation time with the fitness-correlated evaluation time ($c_g > 0$)

c_g	Sync. vs. IO			Sync. vs. RO			IO vs. RO		
	1	≈	2	1	≈	2	1	≈	2
0.2	1	12	6	0	14	5	2	17	0
0.5	0	10	9	0	11	8	0	19	0
Total	1	22	15	0	25	13	2	36	0

c_g	Sync. vs. Seq.			IO vs. Seq.			RO vs. Seq.		
	1	≈	2	1	≈	2	1	≈	2
0.2	19	0	0	19	0	0	19	0	0
0.5	19	0	0	19	0	0	18	1	0
Total	38	0	0	38	0	0	37	1	0

If the first algorithm wins against the second one, “1” is counted, while “2” is counted if the second one wins. If two are a tie, “≈” is counted. “Sync.” indicates SP-ELMOEA/D, “IO” indicates AP-ELMOEA/D-IO, “RO” indicates AP-ELMOEA/D-RO, and “Seq.” indicates Sequential ELMOEA/D

Table 13 An aggregation of the pairwise statistical test using Mann–Whitney U test with a 5% significance level for the result at the maximum elapsed simulation time with the fitness-correlated evaluation time ($c_g < 0$)

c_g	Sync. vs. IO			Sync. vs. RO			IO vs. RO		
	1	≈	2	1	≈	2	1	≈	2
−0.2	0	9	10	1	14	4	4	15	0
−0.5	0	8	11	1	9	9	2	17	0
Total	0	17	21	2	23	13	6	32	0

c_v	Sync. vs. Seq.			IO vs. Seq.			RO vs. Seq.		
	1	≈	2	1	≈	2	1	≈	2
−0.2	19	0	0	19	0	0	19	0	0
−0.5	19	0	0	19	0	0	19	0	0
Total	38	0	0	38	0	0	38	0	0

If the first algorithm wins against the second one, “1” is counted, while “2” is counted if the second one wins. If two are a tie, “≈” is counted. “Sync.” indicates SP-ELMOEA/D, “IO” indicates AP-ELMOEA/D-IO, “RO” indicates AP-ELMOEA/D-RO, and “Seq.” indicates Sequential ELMOEA/D

scheme, and We proposed the parallelization methods of ELMOEA/D. One is a synchronous parallel ELMOEA/D (SP-ELMOEA/D) that generates new promising solutions after waiting all solution evaluations. On the other hand, another is an asynchronous parallel ELMOEA/D (AP-ELMOEA/D) that generates a new solution immediately after waiting for one solution evaluation. For AP-ELMOEA/D, we proposed two selection scheme of a promising solution. One is the index order selection that selects a promising solution depending on the order of the

index of the neighborhood vector set. Another is the random order selection that randomly select the index of the neighborhood vector set.

We conducted the experiment on multi-objective optimization problems to compare the sequential ELMOEA/D, SP-ELMOEA/D, and the AP-ELMOEA/D variants. The experiment used multi-objective benchmarks and real-world problems. The experiment was conducted on the pseudo (simulated) parallel computing environment and considered two settings of evaluation time of solutions; the

Table 14 The mean HV values and their standard deviation at the maximum number of the actual evaluations with the normally-distributed evaluation time in 30 independent trials on the real-world problems

	c_v	SP-ELMOEA/D		AP-ELMOEA/D-IO		AP-ELMOEA/D-RO	
		Mean	Std.	Mean	Std.	Mean	Std.
HX	0.02	0.3100	1.8×10^{-2}	0.3169 ¹	1.7×10^{-2}	0.3250 ¹	2.0×10^{-2}
	0.05	0.3140	1.9×10^{-2}	0.3146	2.1×10^{-2}	0.3154	2.2×10^{-2}
	0.1	0.3121	2.3×10^{-2}	0.3197	2.2×10^{-2}	0.3235 ¹	2.1×10^{-2}
	0.2	0.3131	2.0×10^{-2}	0.3162	2.2×10^{-2}	0.3117	1.8×10^{-2}
HD	0.02	0.6401 ³	8.0×10^{-3}	0.6367 ³	1.2×10^{-2}	0.6266	1.8×10^{-2}
	0.05	0.6387 ³	1.4×10^{-2}	0.6311	2.1×10^{-2}	0.6293	1.7×10^{-2}
	0.1	0.6389 ³	1.2×10^{-2}	0.6373 ³	1.2×10^{-2}	0.6296	2.0×10^{-2}
	0.2	0.6335	1.7×10^{-2}	0.6282	2.1×10^{-2}	0.6302	1.8×10^{-2}
FP	0.02	0.3547	1.2×10^{-2}	0.3592	1.1×10^{-2}	0.3561	1.1×10^{-2}
	0.05	0.3535	1.0×10^{-2}	0.3608 ¹	1.3×10^{-2}	0.3566	1.1×10^{-2}
	0.1	0.3585	1.4×10^{-2}	0.3585	1.1×10^{-2}	0.3591	1.2×10^{-2}
	0.2	0.3561	9.0×10^{-3}	0.3558	1.3×10^{-2}	0.3575	1.0×10^{-2}
VP	0.02	0.8545	1.9×10^{-2}	0.8533	2.2×10^{-2}	0.8489	2.0×10^{-2}
	0.05	0.8528	1.7×10^{-2}	0.8547	2.1×10^{-2}	0.8499	1.9×10^{-2}
	0.1	0.8537	1.8×10^{-2}	0.8568 ³	1.4×10^{-2}	0.8480	1.7×10^{-2}
	0.2	0.8545	1.9×10^{-2}	0.8540	1.9×10^{-2}	0.8506	1.8×10^{-2}

The result that is significantly better than another is signed with a number of the method (1: SP-ELMOEA/D, 2: AP-ELMOEA/D-IO, 3: AP-ELMOEA/D-RO). The highest mean values are shown in the **bold** style. *HX* Heat Exchanger, *HD* Hydro-Dynamics, *VP* Vibrating Platform, *FP* Facility Placement

Table 15 The mean HV values and their standard deviation at the maximum number of the actual evaluations with the fitness-correlated evaluation time in 30 independent trials on the real-world problems

	c_g	SP-ELMOEA/D		AP-ELMOEA/D-IO		AP-ELMOEA/D-RO	
		Mean	Std.	Mean	Std.	Mean	Std.
HX	0.2	0.3146	2.2×10^{-2}	0.3167	2.1×10^{-2}	0.3142	2.3×10^{-2}
	0.5	0.3076	2.0×10^{-2}	0.3140	2.2×10^{-2}	0.3226 ¹	2.3×10^{-2}
	−0.2	0.3080	1.8×10^{-2}	0.3079	2.4×10^{-2}	0.3165 ^{1,2}	1.8×10^{-2}
	−0.5	0.3127	2.4×10^{-2}	0.3237 ^{1,3}	1.8×10^{-2}	0.3123	2.0×10^{-2}
HD	0.2	0.6362 ³	1.8×10^{-2}	0.6385 ³	8.7×10^{-3}	0.6307	2.0×10^{-2}
	0.5	0.6292 ³	2.4×10^{-2}	0.6369 ³	1.3×10^{-2}	0.6287	1.9×10^{-2}
	−0.2	0.6335	1.8×10^{-2}	0.6365	1.2×10^{-2}	0.6309	1.8×10^{-2}
	−0.5	0.6337 ³	2.0×10^{-2}	0.6370 ³	1.3×10^{-2}	0.6275	2.2×10^{-2}
FP	0.2	0.3569	1.2×10^{-2}	0.3576	1.5×10^{-2}	0.3569	1.3×10^{-2}
	0.5	0.3544	1.3×10^{-2}	0.3597 ¹	1.2×10^{-2}	0.3591 ¹	1.4×10^{-2}
	−0.2	0.3542	1.3×10^{-2}	0.3595	1.4×10^{-2}	0.3590	1.2×10^{-2}
	−0.5	0.3593	1.1×10^{-2}	0.3566	1.0×10^{-2}	0.3585	1.3×10^{-2}
VP	0.2	0.8536	1.5×10^{-2}	0.8587 ³	1.5×10^{-2}	0.8482	1.9×10^{-2}
	0.5	0.8533	1.7×10^{-2}	0.8577 ³	1.6×10^{-2}	0.8492	1.5×10^{-2}
	−0.2	0.8538	1.5×10^{-2}	0.8604 ³	1.3×10^{-2}	0.8516	1.9×10^{-2}
	−0.5	0.8572 ³	1.7×10^{-2}	0.8511	1.6×10^{-2}	0.8450	1.8×10^{-2}

The result that is significantly better than another is signed with a number of the method (1: SP-ELMOEA/D, 2: AP-ELMOEA/D-IO, 3: AP-ELMOEA/D-RO). The highest mean values are shown in the **bold** style. *HX* Heat Exchanger, *HD* Hydro-Dynamics, *VP* Vibrating Platform, *FP* Facility Placement

Table 16 The mean HV values and their standard deviation at the maximum elapsed simulation time in 30 independent trials on the real-world problems

c_g		SP-ELMOEA/D		AP-ELMOEA/D-IO		AP-ELMOEA/D-RO		Seq. ELMOEA/D	
		Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
HX	0.02	0.2920 ⁴	1.8×10^{-2}	0.2905 ⁴	2.1×10^{-2}	0.2945 ⁴	2.3×10^{-2}	0.2697	1.9×10^{-2}
	0.05	0.2887 ⁴	1.9×10^{-2}	0.2919 ⁴	1.5×10^{-2}	0.2916 ⁴	2.4×10^{-2}	0.2697	1.9×10^{-2}
	0.1	0.2875 ⁴	2.4×10^{-2}	0.2928 ⁴	2.0×10^{-2}	0.2947 ⁴	2.4×10^{-2}	0.2697	1.9×10^{-2}
	0.2	0.2886 ⁴	2.0×10^{-2}	0.2922 ⁴	1.6×10^{-2}	0.2887 ⁴	1.7×10^{-2}	0.2697	1.9×10^{-2}
HD	0.02	0.6189 ^{2,3,4}	2.5×10^{-2}	0.6080 ⁴	2.6×10^{-2}	0.5976 ⁴	2.6×10^{-2}	0.4549	3.4×10^{-2}
	0.05	0.6186 ^{3,4}	2.5×10^{-2}	0.6087 ^{3,4}	3.1×10^{-2}	0.5970 ⁴	2.5×10^{-2}	0.4548	3.4×10^{-2}
	0.1	0.6129 ⁴	2.6×10^{-2}	0.6098 ⁴	2.7×10^{-2}	0.6019 ⁴	3.0×10^{-2}	0.4547	3.4×10^{-2}
	0.2	0.6044 ⁴	2.7×10^{-2}	0.6064 ⁴	2.6×10^{-2}	0.6010 ⁴	3.2×10^{-2}	0.4547	3.4×10^{-2}
FP	0.02	0.3271 ⁴	1.4×10^{-2}	0.3356 ^{1,4}	1.5×10^{-2}	0.3325 ⁴	1.5×10^{-2}	0.2359	1.9×10^{-2}
	0.05	0.3269 ⁴	1.5×10^{-2}	0.3377 ^{1,4}	1.4×10^{-2}	0.3344 ^{1,4}	1.3×10^{-2}	0.2359	1.9×10^{-2}
	0.1	0.3281 ⁴	1.6×10^{-2}	0.3393 ^{1,4}	1.1×10^{-2}	0.3373 ^{1,4}	1.4×10^{-2}	0.2359	1.9×10^{-2}
	0.2	0.3253 ⁴	1.5×10^{-2}	0.3347 ^{1,4}	1.6×10^{-2}	0.3373 ^{1,4}	1.4×10^{-2}	0.2359	1.9×10^{-2}
VP	0.02	0.8316 ⁴	2.0×10^{-2}	0.8262 ⁴	2.1×10^{-2}	0.8247 ⁴	1.8×10^{-2}	0.7296	4.9×10^{-2}
	0.05	0.8311 ⁴	2.1×10^{-2}	0.8322 ⁴	2.3×10^{-2}	0.8300 ⁴	1.9×10^{-2}	0.7296	4.9×10^{-2}
	0.1	0.8252 ⁴	2.2×10^{-2}	0.8326 ⁴	2.0×10^{-2}	0.8260 ⁴	2.2×10^{-2}	0.7296	4.9×10^{-2}
	0.2	0.8223 ⁴	2.1×10^{-2}	0.8328 ^{1,4}	2.1×10^{-2}	0.8254 ⁴	2.2×10^{-2}	0.7296	4.9×10^{-2}

The result that is significantly better than another is signed with a number of the method (1: SP-ELMOEA/D, 2: AP-ELMOEA/D-IO, 3: AP-ELMOEA/D-RO). The highest mean values are shown in the **bold** style. HX Heat Exchanger, HD Hydro-Dynamics, VP Vibrating Platform, FP Facility Placement

Table 17 The mean HV values and their standard deviation at the maximum elapsed simulation time in 30 independent trials on the real-world problems

c_g		SP-ELMOEA/D		AP-ELMOEA/D-IO		AP-ELMOEA/D-RO		Seq. ELMOEA/D	
		Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
HX	0.2	0.2922 ⁴	1.8×10^{-2}	0.2949 ⁴	1.6×10^{-2}	0.2952 ⁴	2.0×10^{-2}	0.2697	1.9×10^{-2}
	0.5	0.2988 ⁴	1.9×10^{-2}	0.3057 ⁴	2.2×10^{-2}	0.3078 ⁴	2.0×10^{-2}	0.2697	1.9×10^{-2}
	-0.2	0.2865 ⁴	1.9×10^{-2}	0.2873 ⁴	1.9×10^{-2}	0.2961 ⁴	2.3×10^{-2}	0.2697	1.9×10^{-2}
	-0.5	0.2852 ⁴	2.1×10^{-2}	0.2927 ⁴	2.4×10^{-2}	0.2868 ⁴	2.0×10^{-2}	0.2697	1.9×10^{-2}
HD	0.2	0.6196 ^{3,4}	2.7×10^{-2}	0.6191 ⁴	2.3×10^{-2}	0.6112 ⁴	2.4×10^{-2}	0.4546	3.4×10^{-2}
	0.5	0.6173 ⁴	2.6×10^{-2}	0.6255 ^{3,4}	2.0×10^{-2}	0.6165 ⁴	2.3×10^{-2}	0.4546	3.4×10^{-2}
	-0.2	0.6003 ⁴	3.1×10^{-2}	0.6077 ⁴	2.5×10^{-2}	0.6037 ⁴	2.8×10^{-2}	0.4545	3.5×10^{-2}
	-0.5	0.6045 ^{3,4}	3.1×10^{-2}	0.5957 ⁴	3.2×10^{-2}	0.5860 ⁴	2.5×10^{-2}	0.4393	2.8×10^{-2}
FP	0.2	0.3329 ⁴	1.4×10^{-2}	0.3400 ⁴	1.8×10^{-2}	0.3391 ^{1,4}	1.3×10^{-2}	0.2359	1.9×10^{-2}
	0.5	0.3308 ⁴	1.3×10^{-2}	0.3429 ^{1,4}	1.2×10^{-2}	0.3407 ^{1,4}	1.7×10^{-2}	0.2359	1.9×10^{-2}
	-0.2	0.3299 ⁴	1.5×10^{-2}	0.3353 ⁴	1.5×10^{-2}	0.3348 ⁴	1.5×10^{-2}	0.2359	1.9×10^{-2}
	-0.5	0.3263 ⁴	1.8×10^{-2}	0.3314 ⁴	1.3×10^{-2}	0.3286 ⁴	1.8×10^{-2}	0.2359	1.9×10^{-2}
VP	0.2	0.8320 ⁴	1.5×10^{-2}	0.8327 ⁴	1.5×10^{-2}	0.8314 ⁴	2.1×10^{-2}	0.7296	4.9×10^{-2}
	0.5	0.8400 ⁴	2.1×10^{-2}	0.8445 ^{3,4}	1.8×10^{-2}	0.8350 ⁴	2.1×10^{-2}	0.7296	4.9×10^{-2}
	-0.2	0.8272 ⁴	2.0×10^{-2}	0.8278 ⁴	1.7×10^{-2}	0.8236 ⁴	2.1×10^{-2}	0.7296	4.9×10^{-2}
	-0.5	0.8225 ⁴	1.7×10^{-2}	0.8232 ⁴	1.7×10^{-2}	0.8211 ⁴	1.8×10^{-2}	0.7296	4.9×10^{-2}

The result that is significantly better than another is signed with a number of the method (1: SP-ELMOEA/D, 2: AP-ELMOEA/D-IO, 3: AP-ELMOEA/D-RO). The highest mean values are shown in the **bold** style. HX Heat Exchanger, HD Hydro-Dynamics, VP Vibrating Platform, FP Facility Placement

normally-distributed evaluation time and the fitness-correlated evaluation time.

The experimental results first revealed that the parallelization of ELMOEA/D significantly reduces the computational time compared with the sequential one. From the comparison of the synchronous and the asynchronous parallelization schemes, the asynchronous evolution scheme outperforms the synchronous one, in particular, the asynchronous one converges faster even if the variance of the evaluation time is enormous. Even though the evaluation time is biased, the asynchronous evolution scheme has an advantage to the synchronous one. This is because the synchronous evolution scheme still decreases its computational efficiency due to the difference of the evaluation time. Moreover, although two promising solution selection strategies are not significantly different, the promising solution selection based on the index order showed the best performance in most test problems.

In the future, we will conduct experiments using not only ELMOEA/D but also other surrogate-assisted EAs. For example, the method proposed in (Habib et al. 2019) proposes a novel surrogate-assisted MOEA/D that uses multiple surrogate models. We can apply the parallelization scheme we have proposed to such MOEA/D-based approaches. For further improvement of the proposed method, we will introduce a preference-based EA like RVEA (Cheng et al. 2016) to AP-ELMOEA/D instead of MOEA/D. Since AP-ELMOEA/D selects only a single promising solution, we can improve the quality of the promising solution by concentrating the search only around the target weight vector.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

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

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Akinsolu MO, Liu B, Grout V, Lazaridis PI, Mognaschi ME, Barba PD (2019) A parallel surrogate model assisted evolutionary algorithm for electromagnetic design optimization. *IEEE Trans Emerg Top Comput Intell* 3(2):93–105
- Alba E, Tomassini M (2002) Parallelism and evolutionary algorithms. *IEEE Trans Evol Comput* 6(5):443–462. <https://doi.org/10.1109/TEVC.2002.800880>
- Alba E, Luque G, Nesmachnow S (2013) Parallel metaheuristics: recent advances and new trends. *Int Trans Oper Res* 20(1):1–48. <https://doi.org/10.1111/j.1475-3995.2012.00862.x>
- Barbera P, Kozlov AM, Czech L, Morel B, Darriba D, Flouri T, Stamatakis A (2018) EPA-ng: massively parallel evolutionary placement of genetic sequences. *Syst Biol* 68(2):365–369. <https://doi.org/10.1093/sysbio/syy054>
- Cheng R, Jin Y, Olhofer M, Sendhoff B (2016) A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Trans Evol Comput* 20(5):773–791. <https://doi.org/10.1109/TEVC.2016.2519378>
- Das S, Suganthan PN (2011) Differential evolution: A survey of the state-of-the-art. *IEEE Trans Evol Comput* 15(1):4–31. <https://doi.org/10.1109/TEVC.2010.2059031>
- Davis JBA, Shayeghi A, Horswell SL, Johnston RL (2015) The Birmingham parallel genetic algorithm and its application to the direct dft global optimisation of irn (n = 10–20) clusters. *Nanoscale* 7:14032–14038. <https://doi.org/10.1039/C5NR03774C>
- Deb K (2001) Multi-objective optimization using evolutionary algorithms. Wiley, New York
- Deb K, Goyal M (1996) A combined genetic adaptive search (GeneAS) for engineering design. *Comput Sci Inform* 26:30–45
- Deb K, Thiele L, Laumanns M, Zitzler E (2002) Scalable multi-objective optimization test problems. In: Proceedings of the 2002 congress on evolutionary computation. CEC'02 (Cat. No.02TH8600), 1:825–830 vol.1. <https://doi.org/10.1109/CEC.2002.1007032>
- Devos O, Downey G, Duponchel L (2014) Simultaneous data pre-processing and svm classification model selection based on a parallel genetic algorithm applied to spectroscopic data of olive oils. *Food Chem* 148:124–130. <https://doi.org/10.1016/j.foodchem.2013.10.020>
- Fan Z, Li W, Cai X, Huang H, Fang Y, You Y, Mo J, Wei C, Goodman E (2019) An improved epsilon constraint-handling method in moea/d for cmops with large infeasible regions. *Soft Comput* 23(23):12491–12510. <https://doi.org/10.1007/s00500-019-03794-x>
- Habib A, Singh HK, Chugh T, Ray T, Miettinen K (2019) A multiple surrogate assisted decomposition-based evolutionary algorithm for expensive multi/many-objective optimization. *IEEE Trans Evol Comput* 23(6):1000–1014. <https://doi.org/10.1109/TEVC.2019.2899030>
- Haftka RT, Villanueva D, Chaudhuri A (2016) Parallel surrogate-assisted global optimization with expensive functions—a survey. *Struct Multidiscip Optim* 54(1):3–13. <https://doi.org/10.1007/s00158-016-1432-3>
- Harada T, Takadama K (2013) Asynchronous evaluation based genetic programming: comparison of asynchronous and synchronous evaluation and its analysis. In: Krawiec K, Moraglio A, Hu T, Etaner-Uyar A, Hu B (eds) Genetic programming, vol 7831. Lecture notes in computer science. Springer, Berlin, pp 241–252. https://doi.org/10.1007/978-3-642-37207-0_21
- Harada T, Takadama K (2014) Asynchronously evolving solutions with excessively different evaluation time by reference-based evaluation. In: Proceedings of the 2014 annual conference on

- genetic and evolutionary computation, ACM, New York, USA, GECCO '14, pp 911–918, <https://doi.org/10.1145/2576768.2598330>
- Huang GB (2015) What are extreme learning machines? filling the gap between Frank Rosenblatt's dream and John von Neumann's puzzle. *Cogn Comput* 7(3):263–278
- Huang GB, Zhu QY, Siew CK (2004) Extreme learning machine: a new learning scheme of feedforward neural networks. In: 2004 IEEE International joint conference on neural networks (IEEE Cat. No.04CH37541), vol 2, pp 985–990 vol.2, <https://doi.org/10.1109/IJCNN.2004.1380068>
- Huband S, Barone L, While L, Hingston P (2005) A scalable multi-objective test problem toolkit. In: Coello Coello CA, Hernández Aguirre A, Zitzler E (eds) *Evolutionary multi-criterion optimization*. Springer, Berlin, pp 280–295
- Jin Y (2011) Surrogate-assisted evolutionary computation: recent advances and future challenges. *Swarm Evolut Comput* 1(2):61–70. <https://doi.org/10.1016/j.swevo.2011.05.001>
- Knowles J (2006) ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Trans Evol Comput* 10(1):50–66. <https://doi.org/10.1109/TEVC.2005.851274>
- Li H, Zhang Q (2009) Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Trans Evol Comput* 13(2):284–302. <https://doi.org/10.1109/TEVC.2008.925798>
- Liu B, Akinsolu MO, Ali N, Abd-Alhameed R (2019) Efficient global optimisation of microwave antennas based on a parallel surrogate model-assisted evolutionary algorithm. *IET Microw Antennas Propag* 13(2):149–155
- Loshchilov I, Glasmachers T (2015) Black box optimization competition. <https://www.ini.rub.de/PEOPLE/glasmtbl/projects/bbcomp/index.html>, Online
- McKay MD, Beckman RJ, Conover WJ (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21(2):239–245
- Nebro AJ, Durillo JJ, Vergne M (2015) Redesigning the jMetal multi-objective optimization framework. In: Proceedings of the companion publication of the 2015 annual conference on genetic and evolutionary computation, Association for computing machinery, New York, NY, USA, GECCO Companion '15, p 1093–1100, <https://doi.org/10.1145/2739482.2768462>
- Obayashi S, Jeong S, Shimoyama K, Chiba K, Morino H (2010) Multi-objective design exploration and its applications. *Int J Aeronaut Space Sci* 4(4):247–265. <https://doi.org/10.5139/IJASS.2010.11.4.247>
- Oyama A, Kohira T, Kemmotsu H, Tatsukawa T, Watanabe T (2017) Simultaneous structure design optimization of multiple car models using the k computer. In: 2017 IEEE symposium series on computational intelligence (SSCI), pp 1–4, <https://doi.org/10.1109/SSCI.2017.8285350>
- Pavelski LM, Delgado MR, Almeida CP, Gonçalves RA, Venske SM (2014) ELMOEAD-DE: Extreme learning surrogate models in multi-objective optimization based on decomposition and differential evolution. In: 2014 Brazilian conference on intelligent systems, pp 318–323, <https://doi.org/10.1109/BRACIS.2014.64>
- Pavelski LM, Delgado MR, Almeida CP, Gonçalves RA, Venske SM (2016) Extreme learning surrogate models in multi-objective optimization based on decomposition. *Neurocomputing* 180:55–67. <https://doi.org/10.1016/j.neucom.2015.09.111>
- Roberge V, Tarbouchi M, Okou F (2014) Strategies to accelerate harmonic minimization in multilevel inverters using a parallel genetic algorithm on graphical processing unit. *IEEE Trans Power Electron* 29(10):5087–5090. <https://doi.org/10.1109/TPEL.2014.2311737>
- Scott EO, De Jong KA (2015a) Evaluation-time bias in asynchronous evolutionary algorithms. In: Proceedings of the companion publication of the 2015 annual conference on genetic and evolutionary computation, ACM, New York, NY, USA, GECCO Companion '15, pp 1209–1212, <https://doi.org/10.1145/2739482.2768482>
- Scott EO, De Jong KA (2015b) Understanding simple asynchronous evolutionary algorithms. In: Proceedings of the 2015 ACM conference on foundations of genetic algorithms XIII, ACM, New York, NY, USA, FOGA '15, pp 85–98, <https://doi.org/10.1145/2725494.2725509>
- Shayeghi A, Gotz D, Davis JBA, Schafer R, Johnston RL (2015) Pool-bcga: a parallelised generation-free genetic algorithm for the ab initio global optimisation of nanoalloy clusters. *Phys Chem Chem Phys* 17:2104–2112. <https://doi.org/10.1039/C4CP04323E>
- Soufan O, Kleptogiannis D, Kalnis P, Bajic VB (2015) Dwfs: a wrapper feature selection tool based on a parallel genetic algorithm. *PLoS ONE* 10(2):1–23. <https://doi.org/10.1371/journal.pone.0117988>
- Storn R, Price K (1997) Differential evolution—a simple and efficient Heuristic for global optimization over continuous spaces. *J Global Optim* 11(4):341–359. <https://doi.org/10.1023/A:1008202821328>
- Strofiyas G, Porfyri K, Nikolos I, Delis A, Papageorgiou M (2018) Using synchronous and asynchronous parallel differential evolution for calibrating a second-order traffic flow model. *Adv Eng Softw* 125:1–18. <https://doi.org/10.1016/j.advengsoft.2018.08.011>
- Talbi EG (2019) A unified view of parallel multi-objective evolutionary algorithms. *J Parallel Distrib Comput* 133:349–358. <https://doi.org/10.1016/j.jpdc.2018.04.012>
- Wang J, Clark SC, Liu E, Frazier PI (2016) Parallel Bayesian global optimization of expensive functions. [arXiv:1602.05149](https://arxiv.org/abs/1602.05149)
- Zapotecas Martínez S, Coello Coello CA (2013) MOEA/D assisted by RBF networks for expensive multi-objective optimization problems. In: Proceedings of the 15th annual conference on genetic and evolutionary computation, ACM, New York, NY, USA, GECCO '13, pp 1405–1412, <https://doi.org/10.1145/2463372.2465805>
- Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput* 11(6):712–731. <https://doi.org/10.1109/TEVC.2007.892759>
- Zitzler E, Thiele L (1998) Multiobjective optimization using evolutionary algorithms—a comparative case study. In: Eiben AE, Bäck T, Schoenauer M, Schwefel HP (eds) *Parallel problem solving from nature—PPSN V*. Springer, Berlin, pp 292–301
- Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. *Evol Comput* 8(2):173–195
- Zăvoianu AC, Lughofer E, Koppelstätter W, Weidenholzer G, Amrhein W, Klement EP (2015) Performance comparison of generational and steady-state asynchronous multi-objective evolutionary algorithms for computationally-intensive problems. *Knowl-Based Syst* 87(C):47–60. <https://doi.org/10.1016/j.knsys.2015.05.029>