2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Long Short-Term Memory Network Assisted Evolutionary Algorithm for Computationally Expensive Multiobjective Optimization

Cheng He[1], Hongbin Li[1], Jianqing Lin[2], and Zhichao Lu[3]

Abstract— **Computationally expensive multiobjective optimization problems (EMOPs) that require significant computational resources are commonly encountered in real-world applications. To address the challenges associated with such problems, using computationally inexpensive surrogate models to approximate objectives has emerged as an effective approach to handle EMOPs. However, the current collaboration between evolutionary algorithms (EAs) and surrogate models is limited, relying on static regression or classification methods that do not fully capture the dynamic evolution process of EAs. This study aims to advance the integration of surrogate-assisted multiobjective optimization by incorporating time-series prediction models. The target is to track the evolutionary trajectory of an EA and enhance its search capability. Specifically, long short-term memory (LSTM) networks are embedded into an EA for surrogate-assisted optimization (SAO). The role of LSTM networks in SAO is thoroughly investigated through ablation studies. Experimental results on six EMOPs demonstrate the potential of using LSTM networks in SAO. The results are compared with those obtained from four representative surrogate-assisted EAs, providing insights into the effectiveness of LSTM-based approaches in addressing EMOPs.**

I. Introduction

The proliferation of simulation techniques has revolutionized the modelling of intricate systems that yield multiple outputs [1]. Such techniques have been applied to diverse systems, ranging from computational fluid dynamics employed in airfoil design [2] to electromagnetic simulation for fault line selection [3]. The use of multiobjective evolutionary algorithms (MOEAs) has become ubiquitous for addressing complicated optimization challenges across multiple systems. MOEAs are attractive due to their global search capacity, universality, flexibility, and ability to obtain multiple trade-off solutions within a single run [4]. However, high-fidelity simulations necessitate substantial computation resources, posing a barrier to MOEAs for solving computationally expensive multiobjective optimization problems. Specifically, Evolutionary Algorithms (EAs) necessitate a large number of real function evaluations (FEs) to attain satisfactory performance, whereas limited FEs are available [5]. To surmount this conundrum, cost-effective surrogate models such as neural networks, Kriging models (Gaussian process models, GP models), radial basis function (RBF) networks, and support vector machine (SVM) models are integrated with EAs to enable surrogate-assisted optimization (SAO) [6].

Over the past two decades, surrogate-assisted evolutionary algorithms (SAEAs) have been developed to address computationally expensive single-objective optimization problems [7]. However, these SAEAs are unsuitable for computationally expensive multiobjective optimization problems (EMOPs) due to the conflicting nature of multiple objectives. Recently, specialized SAEAs have been proposed to address EMOPs using mono/multi-surrogate models to approximate different targets [8]. For example, the classification-based SAEA (CSEA) used a single classification-based neural network to learn the dominance relationship between a solution and a solution set [9]. The efficient global optimization for Pareto optimization (ParEGO) employed a regression-based GP model to approximate a randomly selected single-objective subproblem [10]. The reference vector-guided EA (K-RVEA) adopted multiple regression-based GP models to approximate the angle penalized distance [11]. The Kriging model was also used to approximate the single-objective S-metric for saving FEs in the S-metric-based SAEA (SMS-EGO) [12]. Some researchers attempted to leverage surrogate models [13] by designing infill criteria to balance exploration and exploitation [14], adopting multiple surrogate models for embedding learning [15], or refining datasets for better surrogate model training. Modern SAEAs can achieve acceptable performance in solving EMOPs with hundreds of real FEs, thanks to improved surrogate model accuracy, generalization, and uncertainty information [16].

Despite the cooperation among surrogate models, training datasets, and MOEA in existing SAEAs, the mapping relationships extracted by the surrogate model(s) from the EMOP are regarded as static and independent. The surrogate model(s) are trained without considering the order of the solutions evaluated by the real FEs. As a result, the historical evolution information is ignored, and the evolution tendency is only influenced by the machine learning components, i.e., the training dataset and surrogate models.

To investigate the effectiveness of the historical information during the evolution process of SAEAs, a representative time-series prediction model, i.e., the long short-term memory (LSTM) network, is embedded in an MOEA for SAO. Unlike conventional MOEAs that applied the historical information for parameter adjustment [17] or operator selec-

This work was supported by the National Natural Science Foundation of China No. U20A20306.

[1] Cheng He and Hongbin Li are with the School of Electrical and Electronic Engineering, Huazhong University of Science and Technology, Wuhan, China chenghe_seee@hust.edu.cn, lihongbin@hust.edu.cn

[2] Jianqing Lin is with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan, China jqlin@hust.edu.cn

[3] Zhichao Lu is with the Department of Computer Science, City University of Hong Kong, Hong Kong luzhichaocn@gmail.com

978-0-7381-4408-5/23/$31.00 ©2023 IEEE

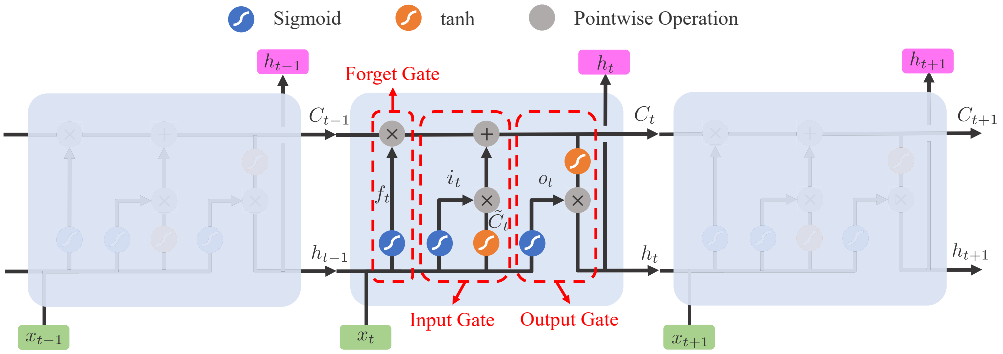

Fig. 1. An illustrative example of the LSTM network, where x_t and h_t denote the input and hidden state of the network at tth generation, respectively. Note that activation functions "sigmoid" and "tanh" are adopted in the network.

tion [18] via tracking the short-term evolution, the long-term evolution of an MOEA is expected to be learned to assist the MOEA more naturally. As a preliminary study, we first investigate the role of the LSTM network in predicting (i) the decision vectors in the next generation, (2) the objective vector of a candidate solution, and (3) the decision vector and its objective vector in the future generation. Then, the best-performed variant is compared with four representative SAEAs to show the potential of time-series prediction in solving EMOPs.

Two main observations of this work could be used to improve the performance of SAEAs in solving real-world EMOPs. First, the mapping from the decision space to the objective space should be learned in LSTM-based surrogate model for pushing the population toward the Pareto optimal front. If only the tendency of the time-series decision vectors is learned, the selection pressure of the population will be lost. Second, the time-series prediction model is promising in steadily obtaining well-converged solutions. Due to the consecutiveness of time-series prediction models, the SAEA tends to select similar but promising solutions to survive.

The rest of this paper is organized as follows. In Section II, we briefly introduce some backgrounds of EMOPs, and then the adopted multiobjective EA and the LSTM network are elaborated. The framework of our proposed LSTM-based SAEA is described in Section III, and the parameter settings of the algorithms and test problems are given in Section IV. In Section V, different variants of the proposed SAEA are compared to analyse the effect of the time-series prediction model in surrogate-assisted multiobjective optimization. Empirical results compared on six benchmark EMOPs are presented in Section V-A, and the conclusions are drawn in Section VI.

II. Background & Motivation

Here, we illustrate the principles of the long short-term memory (LSTM) network [19] and the decomposition-based MOEA based on differential evolution (MOEA/D-DE) [20]. The reasons for using the model and the MOEA are elaborated in what follows.

A. LSTM network

As a typical recurrent neural network (RNN) architecture, the LSTM network was designed to model long-range dependencies precisely [21]. Attributed to the advantage of LSTM networks in addressing the vanishing gradient problem, they have been successfully applied to various complex sequence prediction tasks [22].

Unlike conventional RNNs, an LSTM unit consists of three gates, i.e., forget gate, input gate, and output gate. An illustrative example of the LSTM unit is shown in Figure 1, where the "sigmoid" and "tanh" functions are used for activation. Generally, the LSTM unit consists of three inputs h_{t-1}, C_{t-1}, x_t and two outputs h_t, C_t. For a given time t, h_t is the hidden state, C_t is the unit state/memory, and x_t is the current data point or input. With the assistance of different activation functions and pointwise operations, three functional gates are formed for different purposes.

The forget gate decides what information from the previous state C_{t-1} should be forgotten or kept, which can be given by

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \tag{1}$$

where $\sigma(\cdot)$ represents sigmoid function, W_f is weights, h_{t-1} is the output from the previous unit, x_t is the input, and b_f is bias. The input gate consists of two parts, including

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), & (2)\\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C), & (3) \end{aligned}$$

where W_i, W_C are weights, b_i, b_C are biases, and $\tanh(\cdot)$ represents tanh function. Then, the state is updated by

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t. \tag{4}$$

Finally, the output gate determines the value of the next hidden state h_t by

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{5}$$

$$h_t = o_t \times \tanh(C_t), \tag{6}$$

where W_o is weights and b_o is bias of the output gate [23].

Algorithm 1: Pseudo code of MOEA/D-DE

```
Input: $n$ (population size), $t$ (length of time series), $T$ (number
    of neighbors), $n_r$ (number of replaced solutions).
Output: $P$ (final population).
 1: $\Lambda \leftarrow \{\boldsymbol{\lambda}_1, \cdots, \boldsymbol{\lambda}_n\}$ // generate $n$ weight vectors
 2: $P \leftarrow \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ // initialize $n$ solutions
 3: $\mathbf{z} \leftarrow \min(\mathbf{f}(\mathbf{x}_1), \cdots, \mathbf{f}(\mathbf{x}_n))$ // ideal point
 4: for $i \leftarrow 1 : n$ do
 5:     $B_i \leftarrow \{i_1, i_2 \cdots, i_T\}$ // distinguish the $T$
        closest neighbor weight vectors of $\boldsymbol{\lambda}_i$
 6: end
 7: for $i \leftarrow 1 : t$ do
 8:     for $j \leftarrow 1 : n$ do
 9:         if $rand() < 0.9$ then
10:             $E \leftarrow B_i$
11:         else
12:             $E \leftarrow \{1, 2 \cdots, n\}$
13:         end
14:         $k_1, k_2 \leftarrow$ Randomly select two indices from $E$
15:         $\mathbf{y} \leftarrow$ DE$(\mathbf{x}_1, \mathbf{x}_{k_1}, \mathbf{x}_{k_2})$ // $k_1, k_2 \neq i$
16:         $\mathbf{z} \leftarrow \min(\mathbf{z}, \mathbf{f}(\mathbf{y}))$ // Tchebycheff approach
17:         for $l \leftarrow 1 : n_r$ do
18:             if $\langle |\mathbf{f}(\mathbf{x_k}) - \mathbf{z}|, \boldsymbol{\lambda}_{k_l} \rangle \geq \langle |\mathbf{f}(\mathbf{y}) - \mathbf{z}|, \boldsymbol{\lambda}_{k_l} \rangle$ then
19:                 $\mathbf{x}_j \leftarrow \mathbf{y}$
20:             end
21:         end
22:     end
23: end
```

B. MOEA/D-DE

The pseudo-code of MOEA/D-DE is given in Algorithm 1, which follows the idea of decomposing the MOP into a series of single-objective optimization subproblems and optimizing them simultaneously. Generally, a set of n uniformly distributed weight vectors Λ are first generated, followed by the random initialization of population P of size n (Steps 1-2). Next, T neighborhood weight vectors of each weight vector are distinguished according to the Euclidean distance between pairwise weight vectors (Steps 4-6). Then the four-step main loop of MOEA/D-DE begins, where each subproblem is optimized for t generations (Steps 7-23). First, the parents for each solution are selected according to probability δ (Steps 9-14). Second, an offspring solution $\mathbf{y}$ is generated using the DE operator (Step 15). Third, the Tchebycheff approach is adopted to calculate the aggregation function values of the offspring solution and its n_r neighborhood solutions. Fourthly, the neighborhood solution with the worse aggregation function value will be replaced by the offspring solution (Steps 17-19) [20]. Eventually, the optimized population is output.

C. Motivations

We aim to explore the potential of the time-series prediction model in computationally expensive multiobjective optimization. Before elaborating on the detailed algorithm, we first explain why LSTM and MOEA/D-DE are used.

In conventional SAO, surrogate models are employed to approximate the objective-based fitness function [9]. However, the surrogate models cannot extract additional knowledge from the MOEA, as they do not learn directly from the algorithm [24]. To address this limitation, efforts have been made to enhance the collaboration between the surrogate model and the MOEA. For instance, in [14], the infill criteria of the Kriging models have been improved to better utilize the uncertainty information, thereby enhancing the exploration capability of the MOEA. In K-RVEA [11], training samples are associated with different reference vectors. In MOEA/D-EGO, training samples are clustered into different groups to better capture the evolution process of the EA, which can help to improve the fitting capability of the surrogate model.

The SAEAs mentioned above regard the training samples as time-independent, whereas the training samples are obtained sequentially with a certain ranking. Also, the evolution of the population is controlled by the iteration number, i.e., time, making the ordered sequential dataset a natural time-series one. Hence, the time-series prediction model LSTM is expected to learn the evolution capability of the MOEA.

Unlike single-objective optimization problems, the conflicting nature of EMOPs leads to multiple Pareto non-dominated optima. The evolution of the population is essentially the convergence process over different undetermined search paths, making the characterization of the evolution challenging. Taking the fast non-dominated sorting-based MOEA (NSGA-II) [25] as an example, the population is updated according to the non-dominated sorting and crowding distance. Once a set of offspring solutions are generated, the current and offspring populations are merged and refined to form the population for the next generation. Though the population size is constantly maintained, the evolution of each solution from generation to generation is hard to track.

The application of decomposition-based MOEA (MOEA/D) has demonstrated a notable advantage in providing an intuitive correlation between the optimization of each subproblem and its corresponding weight vector [26]. Such a correlation ensures that the evolution of the solutions associated with a weight vector can be tracked precisely. Specifically, the solutions associated with a weight vector form a time-series dataset that can be leveraged to train an LSTM network. In addition, the Differential Evolution (DE) operator is an effective tool for generating high-quality solutions in local regions, thereby smoothing the evolution of the subpopulation linked to the weight vector. MOEA/D-DE has also shown its capability in handling complex problems with diverse Pareto sets, indicating its proficiency in capturing optima-related information in the decision space. As a result, MOEA/D-DE is an ideal candidate for implementing the MOEA framework in time-series prediction model-assisted optimization.

III. Proposed Framework

The pseudocodes of the LSTM-assisted MOEA framework, named MOEA/D-T, are given in Algorithm 2. To begin with, MOEA/D-DE is adopted to optimize the EMOP for t generations (Step 1). Notably, the solutions associated with each weight vector are merged to form n solution sets $\mathbf{S} = \{S_1, \cdots, S_n\}$, and all the real FE evaluated solutions

Algorithm 2: Framework of MOEA/D-T

```
Input: $FE_{max}$ (maximum number of real FEs), $n$ (population
    size), $t$ (length of time series).
Output: $P$ (final population).
    /* initialize population and LSTM models
    */
 1: $P, \mathbf{S}, A \leftarrow$ MOEA/D-DE($n, t, T, nr$) // $\mathbf{S} = \{S_1, \cdots, S_n\}$
 2: $FEs \leftarrow t \times n$
 3: for $i \leftarrow 1 : n$ do
 4:     $\mathbf{net}(i) \leftarrow LSTM(S_i)$ // train LSTM network
 5: end
 6: while $FEs \leq FE_{max}$ do
 7:     $Y \leftarrow$ SA-MOEA/D($P$,$\mathbf{net}$,$T$,$n_r$) // $Y = \{\mathbf{y}_1, \cdots, \mathbf{y}_n\}$
 8:     $P, A, \mathbf{S} \leftarrow$ Update($P, A, \mathbf{S}, Y$)
 9:     for $i \leftarrow 1 : n$ do
10:         $\mathbf{net}(i) \leftarrow LSTM(S_i)$ // train LSTM network
11:     end
12:     $FEs \leftarrow FEs + n$
13: end
```

are merged in archive A. Then, n LSTM networks are trained, where the input of the network is the decision vector of each solution while the output is the target of the SAEA (Steps 3-5). Three MOEA/D-T variants are mentioned in this study. The first variant predicts the decision vector of the next generation for offspring generation in MOEA/D-DE; the second variant predicts the objective vector of the current decision vector; the last variant predicts both the decision and objective vectors of the next generation. Next, the iterative SAO is conducted until the maximum number of real FEs are used (Steps 6-13). Specifically, LSTM networks are embedded in MOEA/D-DE for SAO (Step 7), followed by the update of population, training dataset, and LSTM networks (Steps 7-11).

A. SA-MOEA/D

Algorithm 3: SA-MOEA/D-D

```
Input: $P$ (population of size $n$), $gen$ (iteration number of SAO),
    $\mathbf{net}$ (LSTM network set).
Output: $Y = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ (candidate solution set).
for $i \leftarrow 1 : gen$ do
    for $j \leftarrow 1 : n$ do
        $\mathbf{y} \leftarrow \mathbf{net}_i(\mathbf{x}_j)$ // LSTM-assisted generation
        $\mathbf{x}_j \leftarrow \mathbf{y}$
    end
end
```

Three LSTM-assisted MOEA/D-DE variants are given, including LSTM-assisted offspring generation (named SA-MOEA/D-D, Algorithm 3), LSTM-based objective prediction (named SA-MOEA/D-O), and LSTM-assisted offspring generation and objective prediction (named SA-MOEA/D-A). The output of all these three algorithms is the candidate solution set Y, which will be evaluated using real FEs.

SA-MOEA/D-D generates candidate solutions directly by using the LSTM network, and then the corresponding solution in the current population is replaced by the generated candidate solution. SA-MOEA/D-O follows the framework of conventional SAEAs, where the objective vectors of the DE-generated candidate solutions are evaluated by the LSTM network. In other words, the objective vector $\mathbf{f}(\mathbf{y})$ in Step 16 of Algorithm 1 is predicted by the trained LSTM network. In SA-MOEA/D-A, the LSTM network is used to predict the candidate offspring solution and its objective vector simultaneously. Specifically, the offspring solution $\mathbf{y}$ in Step 15 and its objective vector $\mathbf{f}(\mathbf{y})$ in Step 16 of Algorithm 1 are predicted by the trained LSTM network.

Once the three variants are embedded in Algorithm 2, three LSTM-assisted SAEAs, namely MOEA/D-TD, MOEA/D-TO, and MOEA/D-TA, can be obtained, respectively.

IV. Experimental Settings

PlatEMO v4.0 is adopted as the experiment platform, and four representative SAEAs are compared with MOEA/D-T. Notably, we select SMS-EGO [12], K-RVEA [11], KTA2 [16], and HeEMOEA [27] for comparison as they represent SAEAs with performance indicator-assisted fitness transformation, training dataset processing, embedding learning, and heterogeneous ensemble based infill criterion, respectively. All these compared algorithms are set as recommended in the literature. Some details about surrogate models and experimental comparisons are presented in what follows.

A. Settings of test Problems

Six DTLZ problems with 10 decision variables ($D = 10$) and two to three objectives ($M \in \{2, 3\}$) are used [28]. Notably, DTLZ4 is not tested as it could cause the model collapse in the DACE toolbox as the number of available unique samples is less than the number of decision variables. The maximum number of real FEs is set to 300 in all experiments as recommended in [16].

B. Settings of surrogate models

Kriging: the DACE toolbox is adopted for SMS-EGO, KTA2, and K-RVEA; the zero-order polynomial regression function is adopted as the regression model in KTA2, and the one-order polynomial regression function is adopted in SMS-EGO and K-RVEA; the Gaussian correlation function is used in both algorithms; the initial θ is set to 5, the lower boundary is set to 1, and the upper boundary is set t to 100 as recommended in [11], [16].

RBF: two RBF networks are involved. The first RBF network uses the least square method to determine its weights, and the second one uses the back-propagation algorithm for model training. The number of hidden neurons for each RBF network is set to $\lceil \sqrt{M + D} + 3 \rceil$, where M and D are the numbers of objectives and decision variables, respectively.

SVM: the L1 soft margin approach was used to train the model, and the three-order polynomial kernel is used with the scale being set to 3 as recommended in [27].

LSTM: the maximum number of epochs is set to 800, the initial learning rate is set to 0.035, and the RMSProp optimizer is adopted for training the model. The structure of the designed LSTM network includes four layers: a sequence input layer with input size D, an LSTM unit with 100 hidden units, a batch normalization layer with mean decay of 0.1 and a variance decay of 0.1, and a fully connected layer with output size being D, M, and $D + M$ in the three variants, respectively.

C. *Performance indicator*

Since the Pareto optimal fronts of the test problems are known, the inverted generalized distance (IGD) indicator is adopted for performance assessment [29]. A number of roughly 10,000 uniformly distributed reference points are used as the reference point set for IGD calculation.

Each SAEA has been executed on each test instance over 25 independent runs for Friedman's test at a significant level of 0.05 [30], where “$+$”, “$-$”, and “$\approx$” indicates the algorithm is significantly better than, significantly worse than, and statistically equal to the compared algorithm, respectively.

V. Ablation study

Statistics of IGD results achieved by three MOEA/D-T variants on six DTLZ problems are given in Table I, where the best result in each row is highlighted. As can be observed, MOEA/D-TO has achieved the most best mean IGD results, followed by MOEA/D-TA. Generally, MOEA/D-TO mainly performs the best on DTLZ1 to DTLZ3, while MOEA/D-TA mainly performs the best on DTLZ5 to DTLZ7. One conclusion can be drawn that learning the objective space is essential, and thus supervised learning is capable of providing the selection pressure for pushing the population toward the PF. Regarding the performance differences between MOEA/D-TO and MOEA/D-TA, it seems that MOEA/D-TO is good at convergence maintenance as only the objective space is considered, while MOEA/D-TA shows its strength in diversity maintenance due to the simultaneous consideration of both decision and objective spaces.

TABLE I

Statistics of IGD Results Achieved by Three MOEA/D-T Variants on DTLZ Problems with 10 Decision Variables.

Problem	M	MOEA/D-TD	MOEA/D-TO	MOEA/D-TA
DTLZ1	2	8.9831e+1 (1.58e+1) $\approx$	7.4019e+1 (2.69e+1) $\approx$	8.2005e+1 (1.93e+1)
	3	7.4722e+1 (1.93e+1) $-$	4.4162e+1 (2.10e+1) $+$	6.3370e+1 (1.51e+1)
DTLZ2	2	2.4741e-1 (6.76e-2) $\approx$	2.0346e-1 (6.81e-2) $\approx$	2.2067e-1 (4.59e-2)
	3	3.7490e-1 (4.59e-2) $\approx$	3.2401e-1 (5.72e-2) $+$	3.5686e-1 (4.39e-2)
DTLZ3	2	2.1183e+2 (1.89e+1) $-$	1.5456e+2 (7.05e+1) $+$	2.0011e+2 (1.06e+1)
	3	1.7358e+2 (3.99e+1) $-$	9.4854e+1 (6.38e+1) $+$	1.6217e+2 (3.36e+1)
DTLZ5	2	2.3118e-1 (6.58e-2) $\approx$	1.7960e-1 (5.78e-2) $+$	2.3450e-1 (6.97e-2)
	3	2.3612e-1 (6.17e-2) $-$	2.1955e-1 (4.30e-2) $\approx$	2.0669e-1 (8.85e-2)
DTLZ6	2	3.1507e+0 (6.01e-1) $-$	3.0631e+0 (8.92e-1) $\approx$	2.7116e+0 (8.40e-1)
	3	2.6585e+0 (6.95e-1) $\approx$	2.3877e+0 (8.89e-1) $\approx$	2.1593e+0 (7.78e-1)
DTLZ7	2	3.2035e+0 (1.37e+0) $\approx$	2.9457e+0 (1.16e+0) $\approx$	3.0074e+0 (1.38e+0)
	3	5.1114e+0 (2.22e+0) $\approx$	4.5718e+0 (1.35e+0) $\approx$	4.3073e+0 (1.62e+0)
$+/-/\approx$		0/5/7	5/0/7	

Due to the multi-modality of the DTLZ1 problem, conventional SAEAs can hardly obtain well-converged solutions on this problem. Figure 2 shows the real function evaluated solutions obtained by the three MOEA/D-T variants on bi-objective DTLZ1 with ten decision variables. As can be observed, MOEA/D-TD and MOEA/D-TA achieved clustered solution sets, while MOEA/D-TO can obtain solutions evenly distributed around the weight vectors. This phenomenon may be attributed to the fact that the output sizes of the LSTM networks in the three variants are D, M, and $D+M$, respectively. In other words, the LSTM networks in MOEA/D-TD and MOEA/D-TA involve more learnable parameters than in MOEA/D-TM, causing the problem of overfitting as only a limited number of training samples are available.

Unlike the DTLZ problem, the PF of the tri-objective DTLZ6 is a degenerated curve, and thus brings great challenges to weight/reference vector-based MOEAs [31]. The real function evaluated solutions obtained by the three MOEA/D-T variants are given in Figure 3. Since only weight vector $[0, 0, 1]$ interacts with the PF, most real function evaluated solutions are closed to f_3. The ineffectiveness of MOEA/D-T variants on this problem could be addressed by selecting weight vectors that interact with the PF.

Since the main challenge in solving EMOPs lies in convergence enhancement, decision-makers could prefer SAEAs with better convergence enhancement instead of capability in diversity maintenance. Thus, MOEA/D-TO is used as the best variant for expensive multiobjective optimization.

A. *Comparisons with Existing SAEAs*

The effectiveness of MOEA/D-TO is tested by comparing it with four representative SAEAs, i.e., K-RVEA, KTA2, HeEMOEA, and SMS-EGO. Statistics of IGD results achieved by the five compared algorithms on six DTLZ problems are given in Table II. It is obvious that MOEA/D-TO has achieved the most best mean IGD results, followed by KTA2 and K-RVEA. It is also worth noting that MOEA/D-TO mainly performs the best on DTLZ1, DTLZ3, and DTLZ6, which are hard to obtain well-converged solutions; on the contrary, KTA2 has performed the best on DTLZ2 and DTLZ5; K-RVEA has performed the best on DTLZ7. The convergence of the proposed algorithm is competitive compared with the four representative SAEAs attributed to the assistance of the LSTM network. Nevertheless, MOEA/D-T performs poorly in diversity maintenance, which may be attributed to the maintenance of a small population.

To show the strength of MOEA/D-TO in convergence enhancement, the non-dominated solutions obtained by the five compared algorithms on tri-objective DTLZ3 are given in Figure 4. The non-dominated solutions obtained by HeEMOEA and K-RVEA are far from the PF, while only MOEA/D-TO can obtain well-converged solutions, as shown in the subplot.

The results in Table II demonstrate that MOEA/D-TO achieves acceptable variances in multiple runs, where the variance of the IGD results is smaller than the mean IGD result. However, K-RVEA and KTA2 have both achieved statistical results with large variances. Therefore, MOEA/D-TO's superior robustness, compared to the other algorithms evaluated, is promising for solving real-world EMOPs.

VI. Conclusions

This study explored the use of time-series prediction models in learning the evolution process for computationally expensive multiobjective optimization. Attributing to

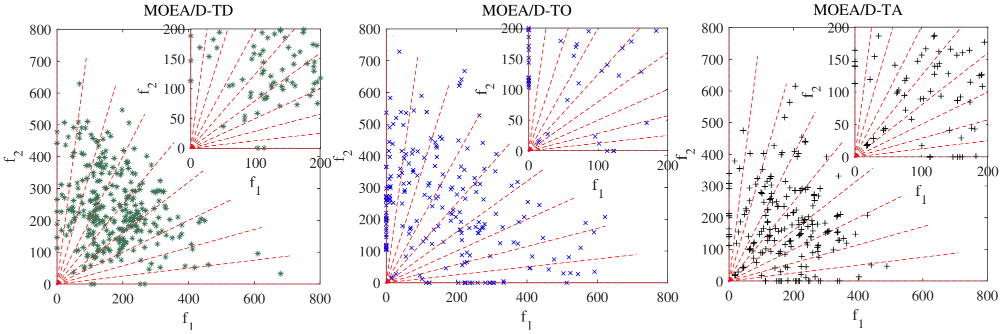

Fig. 2. All real function evaluated solutions obtained by three variants of MOEA/D-T on bi-objective DTLZ1 with 10 decision variables. The red dash lines are the adopted weight vectors.

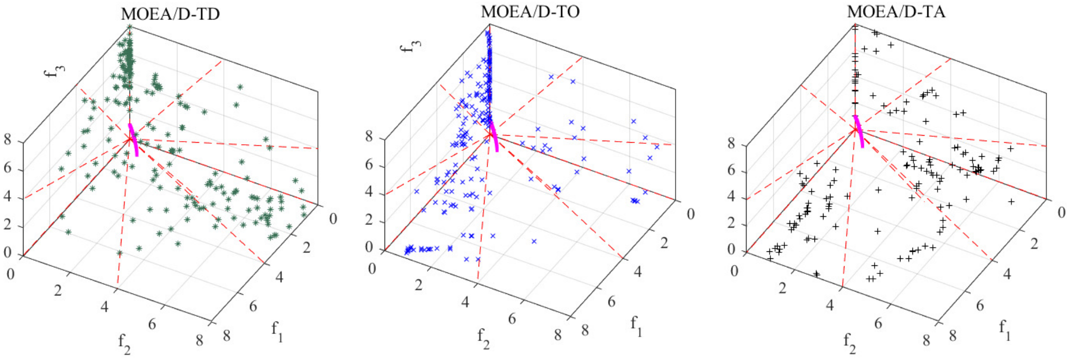

Fig. 3. All real function evaluated solutions obtained by three variants of MOEA/D-T on tri-objective DTLZ6 with 10 decision variables. The red dash lines are the adopted weight vectors, and the magenta dots are the Pareto optimal solutions.

TABLE II

Statistics of IGD Results Achieved by K-RVEA, KTA2, HeEMOEA, SMS-EGO, and MOEA/D-TO on Six DTLZ Problems with 10 Decision Variables.

Problem	M	K-RVEA	KTA2	HeEMOEA	SMS-EGO	MOEA/D-TO
DTLZ1	2	8.9762e+1 (4.59e+1) ≈	1.1751e+2 (4.76e+1) ≈	1.4948e+2 (3.24e+1) −	8.9204e+1 (4.03e+1) ≈	7.4019e+1 (2.69e+1)
	3	9.5041e+1 (2.33e+1) −	6.4702e+1 (3.57e+1) ≈	1.0705e+2 (2.88e+1) −	7.4849e+1 (2.87e+1) ≈	4.4162e+1 (2.10e+1)
DTLZ2	2	9.1752e-2 (1.29e-1) ≈	8.0855e-2 (1.52e-1) +	1.4309e-1 (2.38e-2) ≈	2.7762e-1 (6.67e-2) ≈	2.0346e-1 (6.81e-2)
	3	2.1620e-1 (1.06e-1) +	1.3100e-1 (1.41e-1) +	1.7829e-1 (1.07e-2) +	2.7641e-1 (5.09e-2) ≈	3.2401e-1 (5.72e-2)
DTLZ3	2	2.4072e+2 (1.39e+2) ≈	2.4648e+2 (1.10e+2) ≈	3.6847e+2 (6.84e+1) −	2.4662e+2 (1.09e+2) ≈	1.5456e+2 (7.05e+1)
	3	2.7030e+2 (7.55e+1) −	1.9695e+2 (9.79e+1) ≈	2.9209e+2 (8.77e+1) −	2.2145e+2 (1.10e+2) −	9.4854e+1 (6.38e+1)
DTLZ5	2	8.7780e-2 (1.32e-1) ≈	7.1991e-2 (1.35e-1) +	1.3463e-1 (2.15e-2) ≈	2.6166e-1 (7.78e-2) ≈	1.7960e-1 (5.78e-2)
	3	1.4614e-1 (8.75e-2) +	7.5257e-2 (1.30e-1) +	1.2281e-1 (1.64e-2) +	1.3889e-1 (9.10e-2) +	2.1955e-1 (4.30e-2)
DTLZ6	2	3.8490e+0 (1.18e+0) ≈	3.3361e+0 (2.19e+0) ≈	7.5148e+0 (1.13e-1) −	3.2245e+0 (2.21e+0) ≈	3.0631e+0 (8.92e-1)
	3	3.9423e+0 (1.48e+0) −	2.6012e+0 (1.85e+0) ≈	6.6418e+0 (1.20e-1) −	4.0937e+0 (1.49e+0) −	2.3877e+0 (8.89e-1)
DTLZ7	2	8.0909e-1 (1.84e+0) +	9.7421e-1 (1.95e+0) +	3.3084e+0 (4.75e-1) ≈	1.5234e+0 (1.62e+0) ≈	2.9457e+0 (1.16e+0)
	3	1.2792e+0 (2.70e+0) +	1.3246e+0 (2.56e+0) +	5.5028e+0 (4.93e-1) ≈	2.3972e+0 (2.25e+0) ≈	4.5718e+0 (1.35e+0)
+/ − / ≈		4/3/5	6/0/6	2/6/4	1/2/9	

the decomposition-based MOEA framework, optimizing a subproblem in an EMOP over several generations can be regarded as a time-series evolutionary optimization process. Therefore, the LSTM network can approximate the time-series data regarding the objective or the decision vectors. Subsequently, the trained network can predict the candidate offspring solution or the fitness value of a candidate solution in an SAEA. Three variants that adopted the LSTM network for surrogate-assisted multiobjective optimization are proposed and compared. The ablation study indicates the effectiveness of the LSTM network in learning the objective functions in terms of time-series prediction. Empirical comparisons with SOTA SAEAs have validated the effectiveness and robustness of time-series prediction models in computationally expensive multiobjective optimization.

Despite the effectiveness of the proposed time-series prediction model based SAEA, some problems should be addressed in future work. (1) Due to the limitation in the number of FEs, the length of the time-series samples for training the LSTM network is short, which could suffer from the loss of model accuracy. Innovative ideas for data augmentation could significantly improve the accuracy of the

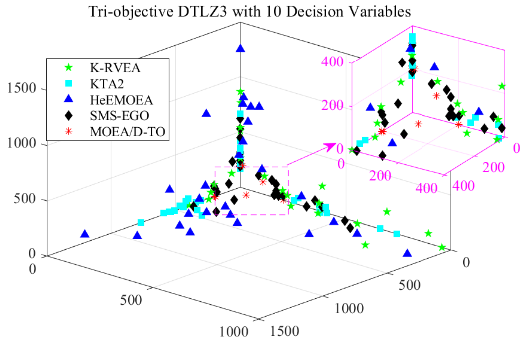

Fig. 4. Non-dominated solutions obtained by five compared algorithms on tri-objective DTLZ3 with 10 decision variables in the run associated with the medium IGD results.

surrogate models. (2) The decomposition-based framework is sensitive to the shape of the PF, and thus its performance could degenerate on EMOPs with irregular PFs. The employment of weight adaptation strategies or adaptive MOEA/D is promising for improving the versatility of the proposed SAEA. (3) Only the LSTM network was used as the time-series prediction model in this work. The potential of other statistical, machine learning, or deep learning time-series models should be investigated to demonstrate the advantages of time-series prediction models in SAO. (4) The applications of time-series prediction model-based SAEAs in real-world EMOPs are also highly desirable.

References

[1] D. Mourtzis, "Simulation in the design and operation of manufacturing systems: state of the art and new trends," *Int. J. Prod. Res.*, vol. 58, no. 7, pp. 1927–1949, 2020.

[2] J. Wang, C. He, R. Li, H. Chen, C. Zhai, and M. Zhang, "Flow field prediction of supercritical airfoils via variational autoencoder based deep learning framework," *Phys. Fluids*, vol. 33, no. 8, p. 086108, 2021.

[3] Q. Chen, J. Peng, H. Li, S. Yang, L. Zhou, D. Pang, and H. Deng, "Extremely low frequency-based faulty line selection of low-resistance grounding system," *IET Gener. Transm. Dis.*, vol. 15, no. 10, pp. 1565–1577, 2021.

[4] A. Slowik and H. Kwasnicka, "Evolutionary algorithms and their applications to engineering problems," *Neural. Comput. Appl.*, vol. 32, no. 16, pp. 12 363–12 379, 2020.

[5] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm Evol. Comput.*, vol. 1, no. 2, pp. 61–70, 2011.

[6] C. Sun, Y. Jin, R. Cheng, J. Ding, and J. Zeng, "Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 644–660, 2017.

[7] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive multiobjective optimization by moea/d with gaussian process model," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 456–474, 2010.

[8] J. Stork, M. Friese, M. Zaefferer, T. Bartz-Beielstein, A. Fischbach, B. Breiderhoff, B. Naujoks, and T. Tušar, "Open issues in surrogate-assisted optimization," in *High-performance simulation-based optimization*. Springer, 2020, pp. 225–244.

[9] L. Pan, C. He, Y. Tian, H. Wang, X. Zhang, and Y. Jin, "A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 1, pp. 74–88, 2019.

[10] J. Knowles, "Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 10, no. 1, pp. 50–66, 2006.

[11] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, and K. Sindhya, "A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 129–142, 2018.

[12] W. Ponweiser, T. Wagner, D. Biermann, and M. Vincze, "Multiobjective optimization on a limited budget of evaluations using model-assisted $\mathcal{S}$-metric selection," in *Proc. Int. Conf. Parallel Probl. Solving Nat.*, G. Rudolph, T. Jansen, N. Beume, S. Lucas, and C. Poloni, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 784–794.

[13] R. Alizadeh, J. K. Allen, and F. Mistree, "Managing computational complexity using surrogate models: a critical review," *Res. Eng. Des.*, vol. 31, no. 3, pp. 275–298, 2020.

[14] M. Wu, L. Wang, J. Xu, P. Hu, and P. Xu, "Adaptive surrogate-assisted multi-objective evolutionary algorithm using an efficient infill technique," *Swarm Evol. Comput.*, vol. 75, p. 101170, 2022.

[15] T. Sonoda and M. Nakata, "Multiple classifiers-assisted evolutionary algorithm based on decomposition for high-dimensional multi-objective problems," *IEEE Trans. Evol. Comput.*, vol. 26, no. 6, pp. 1581 – 1595, 2022.

[16] Z. Song, H. Wang, C. He, and Y. Jin, "A Kriging-assisted two-archive evolutionary algorithm for expensive many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 25, no. 6, pp. 1013–1027, 2021.

[17] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *IEEE Congr. Evol. Comput.*, 2013, pp. 71–78.

[18] J. Zhang, Z.-h. Zhan, Y. Lin, N. Chen, Y.-j. Gong, J.-h. Zhong, H. S. Chung, Y. Li, and Y.-h. Shi, "Evolutionary computation meets machine learning: A survey," *IEEE Comput. Intell. Mag.*, vol. 6, no. 4, pp. 68–75, 2011.

[19] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5929–5955, 2020.

[20] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, 2008.

[21] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, 2000.

[22] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D*, vol. 404, p. 132306, 2020.

[23] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, 2019.

[24] J. Li, J. Cai, and K. Qu, "Surrogate-based aerodynamic shape optimization with the active subspace method," *Struct. Multidiscipl. Optim.*, vol. 59, no. 2, pp. 403–419, 2019.

[25] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[26] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, 2007.

[27] D. Guo, Y. Jin, J. Ding, and T. Chai, "Heterogeneous ensemble-based infill criterion for evolutionary multiobjective optimization of expensive problems," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 1012–1025, 2018.

[28] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *IEEE Congr. Evol. Comput.*, vol. 1, 2002, pp. 825–830 vol.1.

[29] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima, "Reference point specification in inverted generational distance for triangular linear pareto front," *IEEE Trans. Evol. Comput.*, vol. 22, no. 6, pp. 961–975, 2018.

[30] D. G. Pereira, A. Afonso, and F. M. Medeiros, "Overview of Friedman's test and post-hoc analysis," *Communications in Statistics-Simulation and Computation*, vol. 44, no. 10, pp. 2636–2653, 2015.

[31] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 169–190, 2016.