A Graph Neural Network Assisted Evolutionary Algorithm for Expensive Multi-objective Optimization

Xiangyu Wang†
Faculty of Technology, Bielefeld University
33619 Bielefeld, Germany
Email: xiangyu.wang@uni-bielefeld.de

Xilu Wang†
Faculty of Technology, Bielefeld University
33619 Bielefeld, Germany
Email: xilu.wang@uni-bielefeld.de

Yaochu Jin*
School of Engineering, Westlake University
Hangzhou 310030, China
Email: jinyaochu@westlake.edu.cn

Ulrich Rückert
Faculty of Technology, Bielefeld University
33619 Bielefeld, Germany
Email: rueckert@techfak.uni-bielefeld.de

***Abstract*—Surrogate-assisted evolutionary algorithms (SAEAs) have emerged as a promising approach to addressing expensive and black-box problems. Most existing SAEAs leverage regression models to predict the objective values, reducing the use of true objective functions. However, these methods focus on learning the mapping from the decision space to the objective space and may fail to reveal the relationship between solutions in the decision space. Recently, graph neural networks (GNNs) have attracted increased attention due to their powerful ability to expose sample interaction. In this paper, we propose employing a graph neural network for learning embeddings of solutions in the decision space, followed by a classification task aimed at predicting dominance relationships between solutions in the objective space and a regression task for obtaining the estimated fitness values. To this end, we generate a graph at each generation to represent the topology relationship between solutions in the decision space, where nodes represent solutions, and edges are added depending on the Euclidean distances between nodes. In addition, a new acquisition function that adaptively weights the predictions on objective values and dominance relationships is proposed to effectively identify new samples. The performance of the proposed method is examined by extensive empirical studies on a widely used test suite in comparison to its peer algorithms, and the results confirm the effectiveness of the proposed method.**

***Index Terms*—Expensive optimization, Surrogate model, Graph neural network, Acquisition function.**

I. Introduction

Expensive multi-objective optimization problems can be seen in many real-world applications [1], [2], such as feature selection [3], robust optimization of large-scale network [4], and neural architecture search [5]. Take neural architecture search as an example, the evaluation of an architecture requires training and testing on datasets, which is very computationally expensive [6]. Therefore, optimizing a neural architecture is challenging for optimization algorithms, as the number of affordable evaluations is limited. Thus, specifically designing effective methods for expensive optimization problems is in high demand, which can use less real fitness evaluation time to obtain better results.

Multi-objective evolutionary algorithms (MOEAs) have been widely used in solving multi-objective optimization problems, as they can obtain a set of Pareto optimal solutions in a single run. In general, MOEAs generate an offspring population from the previous population and select better solutions based on their fitness values to survive to the next generation. However, MOEAs generally require a large number of objective evaluations to find a set of non-dominated solutions, which is inapplicable to expensive multi-objective optimization problems. Therefore, surrogate-assisted evolutionary algorithms (SAEAs) have been proposed to construct surrogate models to approximate the objective functions or learn the dominance relationship. Based on the predictions provided by the surrogates, promising candidate solutions can be selected for evaluation using the true objective functions, thereby reducing the number of true objective evaluations and improving search efficiency [6].

Many SAEAs have been proposed based on different types of cheap surrogate models, such as Kriging models [7], [8], random forests [9], radial basis functions (RBFs) [10], and neural networks [11]. These models often take the decision vector as the input and its corresponding objective functions as the output, and generally learn a regression model to efficiently approximate expensive objective functions. Alternatively, classification models have been adopted in multi-objective SAEAs to identify whether a candidate solution is non-dominated. Besides, classification models are also often used in the constrained problems [9] to determine whether a solution is feasible or not, using k-nearest neighbors algorithms [12], or support vector machine [13], [14], to name

†These authors contributed equally to this work.
*Corresponding authors.

a few. Moreover, ensemble models are studied to take advantage of different surrogate models with the help of model management strategies [15], [16]. After constructing surrogate models, the next step is called model management, i.e., the way of collaboratively using the surrogate models and the true objective functions, and efficiently updating the surrogates and effectively guiding the search. For example, acquisition functions (AFs) in Bayesian optimization are designed to strike a balance between exploration and exploitation. As a result, candidate solutions that are promising to improve the optimization performance are obtained by optimizing the acquisition function. Commonly used AFs include the expected improvement [17], lower confidence bound (LCB) [2].

Most surrogate models, both for regression and classification, do not pay much attention to the relationship between the solutions in the decision space. However, exploiting the topology of decision variables in the decision space is crucial for further enhancing prediction accuracy. Because the relative position information between solutions in the decision space can reflect their corresponding positions in the objective space, particularly in continuous optimization. The definition of continuous functions f at x_0 is that for every $\epsilon > 0$, there exists a $\delta > 0$ such that for all x in the domain:

$$|x - x_0| < \delta \quad \text{implies} \quad |f(x) - f(x_0)| < \epsilon. \tag{1}$$

Furthermore, when the function exhibits Lipschitz continuity, its rate of change is bounded by a real value. In other words, the difference in objective function values is expected to be relatively small if the two solutions are neighbors in the decision space. Therefore, exploring the topology structure in the decision space should hopefully help better approximate the functional relationship between the decision variables and the objectives. Fortunately, the realization of this concept is attainable through the application of graph neural networks, given their adeptness at capturing relationships among nodes within a graph [18].

Different from other machine learning methods that can only learn specific data structures, such as multilayer perceptrons (MLPs) [19] for vector data, convolution neural networks [20] for image data, and recurrent neural networks [21] for sequence data, GNNs are capable of dealing with irregular data, for example, graphs with different topology structures, and have been applied in many real-world applications [22], including chemical reaction prediction [23], traffic state prediction [24], and social recommendation [25]. The distinct characteristic of GNNs is that they update nodes by aggregating and combining information (i.e., the embeddings) from their neighboring nodes through edges by taking graphs as the input [26], demonstrating remarkable power and flexibility in revealing topology within graphs.

In this paper, we propose a GNN-assisted evolutionary algorithm, called GNNAEA, for solving expensive multi-objective optimization problems (EMOPs). Inspired by the Lipschitz continuity in mathematics, a graph is constructed depending on the Euclidean distance of solutions, which is expected to contain the implicit information in the objective space. A GNN model is employed to generate embeddings for the downstream classification and regression tasks. Based on the prediction of the classification and regression models, a new acquisition function is proposed to effectively identify new samples. The main contributions of this work are summarized as follows.

- To better represent the topology relationship between solutions in the decision space, a graph is constructed wherein each node represents a solution. Edges are added if the Euclidean distance between two nodes is below a predefined threshold among all Euclidean distances.
- Taking the generated graph as its input, a graph neural network is applied to learn the representation of the solutions in the decision space, followed by two MLPs designed to predict the approximate fitness values (treated as a regression task) and the domination relationship (a classification task), respectively. By taking advantage of GNNs' ability of information reveal in irregular structured data, the proposed method can extract relationships among nodes in a more accurate way, thus achieving better results in regression and classification tasks.
- A new acquisition function (AF) is proposed based on the prediction of the objective values and the dominance relationship. First, an MOEA optimizes the predicted objective values obtained from the trained GNN-based model for a fixed number of generations. Afterward, the final optimized population is evaluated by the AF that adaptively weights prediction of the objective values and dominance relationship probability, from which the new samples are selected. The motivation of this method is to balance the exploitation and exploration by minimizing the objective values and maximizing the uncertainty of the dominance relationship.

The remainder of this paper is organized as follows. Section II briefly introduces the definition of Pareto fronts, surrogate models, and graph neural networks. Section III describes the proposed method GNNAEA in detail. In Section IV, we present the results of comparative experiments on benchmark problems, as well as the sensitivity analysis and ablation studies. Section V summarizes this paper and suggests promising future research directions.

II. Related Work

A. Pareto Optimal Fronts

Multi-objective optimization problems (MOPs) aim to simultaneously optimize multiple objective functions that are often conflicting with each other. Therefore, there is no single solution that could be optimal for all functions. Instead, the target is to obtain a set of solutions forming the Pareto optimal set, where every element $\mathbf{x}^*$ in the set satisfies:

$$\forall i : f_i(\mathbf{x}^*) \leq f_i(\mathbf{x}), \exists j : f_j(\mathbf{x}^*) < f_j(\mathbf{x}), \tag{2}$$

where $\mathbf{x}$ represents other solutions not included in the Pareto optimal set, i and $j = 1, 2, \ldots, M$, and M is the number of optimization functions. The objective values of elements in the Pareto optimal set consist of the Pareto optimal front.

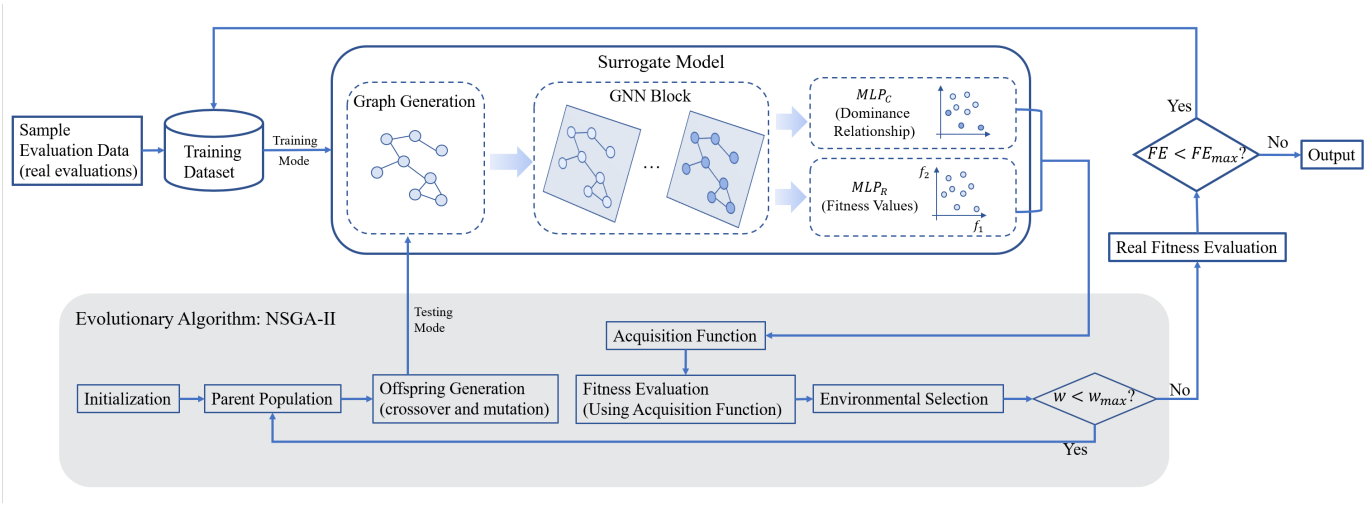


Fig. 1: The Framework of the proposed method GNNAEA.

B. Surrogate Models

Surrogate models are often used in evolutionary algorithms to reduce the time or cost for real fitness evaluations [27]. In general, surrogate models are trained before the evolutionary process, using the collected data, and are used to evaluate fitness values during the evolutionary process. In online data-driven optimization, models can be updated and fine-tuned by leveraging the real fitness evaluations obtained from newly generated solutions.

The accuracy of surrogate models is essential to the overall performance of MOEAs, and some simple yet efficient surrogate models have been studied in the past decades. Kriging model is a statistical method that gives the best linear unbiased prediction as well as its uncertainty at unsampled locations [28]. However, as a machine-learning based model, GNN is more flexible in representation learning and generalization to unseen data. Radial basis function networks [29] can be considered as machine learning-based surrogate models, which contain several weighted basis functions. The basic idea of using RBFs is that the predicted objective values should be closely aligned with those that share similar values in the decision variables. However, the fundamental concept differs in the utilization of the Euclidean distance between RBFs and GNNs; that is, the usage of RBFs emphasizes the Euclidean distance between samples and basis function centers, while GNNs focus on capturing the relationships between samples.

C. Graph Neural Networks

Over the past decade, graph and node embedding methods have gained popularity for solving graph-based problems, with numerous promising approaches proposed to explore and reveal structural information within graphs, such as DeepWalk [30] and Node2Vec [31]. These methods often employ random walks around nodes to get insights within a local area. However, they face limitations in capturing generalized patterns within a single graph or across multiple graphs. In recent years, graph neural networks have attracted increasing attention with their learning ability in graphs. These networks are typically categorized into spectral-based and spatial-based approaches. The former aims to design new graph filters based on graph signal processing theory [26], while the latter focuses on introducing novel message-passing approaches [24].

Here, we give a brief introduction to the basic idea of spectral-based GNNs that we utilize in this work. In general, with a graph filter $\mathbf{g}$, the input $\mathbf{x}$ is expressed as

$$\mathbf{x} *_G \mathbf{g} = \mathcal{F}^{-1}(\mathcal{F}(\mathbf{x}) \odot \mathcal{F}(\mathbf{g})), \tag{3}$$

where $\odot$ represents the element-wise product. $\mathcal{F}(\mathbf{x}) = \mathbf{U}^T\mathbf{x}$ and $\mathcal{F}^{-1}(\hat{\mathbf{x}}) = \mathbf{U}^T\hat{\mathbf{x}}$ are the graph Fourier transformation and the inverse graph Fourier transformation, respectively. $\mathbf{U}$ can be found in $\mathbf{L} = \mathbf{I}_n - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{A}$ is the adjacency matrix of a graph, $\mathbf{D}$ is the diagonal matrix representing the node degrees, $\mathbf{\Lambda}$ and $\mathbf{U}$ are the diagonal matrix of eigenvalues and the corresponding matrix of eigenvectors of $\mathbf{L}$.

III. The Proposed Method

A. The Overall Framework

The overall framework of the proposed GNNAEA is shown in Fig. 1. At first, Latin hypercube sampling is applied to sample $11D - 1$ points to be evaluated by real fitness evaluations in the decision space, where D is the dimension of the decision space. The training dataset comprises node pairs, each associated with a solution, its real fitness values and dominance relationships. Within the surrogate model, the initial step involves generating a graph based on the Euclidean distance among training data. This graph is then used as the input for the GNN block to obtain a richer expression by revealing the graph structure and the relationships between nodes. The embedding of solutions obtained from the GNN block is subsequently utilized in two MLPs, denoted as MLP_C for learning the dominance relationship and MLP_R for fitting the fitness values of the solutions, respectively.

On the other hand, the acquisition function is used to evaluate solutions generated from the evolutionary algorithm NSGA-II [32], and choose u solutions for real evaluations after several iterations. In the evolutionary process, the population is first initialized and is considered as a parent population to generate offspring by crossover and mutation. The generated offspring population is then processed by the surrogate model to get the approximate fitness values. The new parent population is obtained after environmental selection if $w < w_{max}$; otherwise, the population is evaluated by the AF and the promising u solutions are output for real evaluations.

After getting solutions through real evaluations, the surrogate model is updated. Finally, all solutions in the training dataset are output as the final result once the number of real fitness evaluations FE reaches the predefined value FE_{max}.

B. A GNN-assisted Model

1) Graph Generation: The graph is constructed depending on the Euclidean distance among solutions. The Euclidean distance matrix $\mathbf{E} \in \mathbb{R}^{N\times N}$ is first calculated, where $\mathbf{E}_{ij}$ represents the Euclidean distance between solution i and j, with N being the number of solutions. Then, the adjacent matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ is derived by setting elements in $\mathbf{E}$ whose values are greater than the threshold to zero and the rest to one. In other words, two nodes (solutions) are connected if the Euclidean distance is less than the specified threshold. The threshold is determined as the value below which the connection ratio r_c of distance values falls, for example, the threshold is the lower quantile when $r_c = 0.25$. This concept is motivated by the idea that solutions positioned closely together in the decision space tend to exhibit similarities and can thus benefit from learning from each other.

2) GNN Block: Here, we apply the classic spectral-based GNN model, GCN [26], to process the generated graph and obtain the representation containing the relationship between nodes.

We assume that each row of solution matrix $\mathbf{X} \in \mathbb{R}^{N\times D}$ represents one solution, where D is the dimensionality of the decision space. The k-th layer of GCN can be expressed as

$$\mathbf{H}^{(k)} = \text{ReLU}(\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}\mathbf{H}^{(k-1)}\mathbf{W}^{(k)}), \tag{4}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_n$ and $\tilde{\mathbf{D}}$ is the diagonal degree matrix of $\tilde{\mathbf{A}}$. $\mathbf{W}^{(k)} \in \mathbb{R}^{D'\times D'}$ is the learnable weight matrix in the k-th layer ($k \in \mathbb{Z}^+$), where D' is the number of dimensions of embeddings, $\mathbf{H}^{(0)} = \mathbf{X}$, and $\mathbf{W}^{(1)} \in \mathbb{R}^{D\times D'}$. The GCN model has a good ability to generalize to unseen graphs [33], therefore, it is suitable to use as a surrogate model in this task, since input graphs during training and testing have different topologies and different numbers of nodes. In this paper, we set $D' = D$, and $K = 2$ (The GCN model in this paper is constructed by a python package DGL www.dgl.ai). This is because a higher dimension of node embeddings and complex structure of GCN may cause overfitting and oversmoothing problems in GNNs.

3) MLPs for Regression and Classification: The final node embedding matrix $\mathbf{H}^{(K)}$ is obtained from K hidden layers, which contains the information extracted from each node and its neighbor nodes. The downstream tasks are regression for fitting the real fitness values and classification for distinguishing the non-dominated solutions from the dominated ones, facilitated by two MLPs.

Applying MLP_R, the estimated fitness values are obtained through the equation:

$$\mathbf{F}_e = \sigma(\mathbf{H}^{(K)} \times \mathbf{W}_e), \tag{5}$$

where σ represents the sigmoid function, and $\mathbf{W}_e \in \mathbb{R}^{D'\times M}$.

On the other hand, the classification target is conducted by MLP_C as follows,

$$\mathbf{L}_e = \delta(\mathbf{H}^{(K)} \times \mathbf{W}_c), \tag{6}$$

where δ is the *softmax* function, and $\mathbf{W}_e \in \mathbb{R}^{D'\times 2}$. Since the softmax function is applied as the activated function, two dimensions of $\mathbf{L}_e$ represent the probability of solutions being grouped into 0 (dominated) and 1 (non-dominated), respectively.

4) Training Method: For the regression task, the loss function f_r is the Mean Squared Error (MSE) function, which aims to minimize the distance between the estimated fitness values and the normalized real fitness values. On the other hand, the cross-entropy function f_c is applied in the classification task. Therefore, the final loss function f is obtained as follows,

$$f = f_c + 5 \times \min\{f_c, f_r\}. \tag{7}$$

The AdamW optimizer with the weight decay being 0.01 is applied to optimize and update weights in the GNN model. The training rate is set to be 0.005, and 20 epochs are used for one-round training. To avoid overfitting in training surrogate models, only part of the training data (r_s of the number of training data) is sampled for generating the graph, where r_s is a percentage.

C. New Sample Selection

After training the GNN model and MLPs to predict objective values and classify the candidate solutions into non-dominated and dominated ones, the next step is to identify promising new samples by balancing exploration and exploitation. To achieve this, we propose a new selection strategy based on the information provided by the surrogate models.

First, a multi-objective evolutionary algorithm (NSGA-II in this work) is adopted to optimize the predicted values for a fixed number of generations, resulting in the optimized population P^*. Then an AF is proposed to select u promising candidate solutions to be evaluated by the true objective functions. Based on the way of constructing the surrogate model, an insight into the predictions of MLP_R and MLP_C can be provided: minimizing the predictions of the objective values $\hat{\mathbf{f}}$ given by MLP_R enhances the exploitation, while maximizing the probability of being dominated p_0 for candidate solutions prioritizes exploration. Hence, we carefully design a new

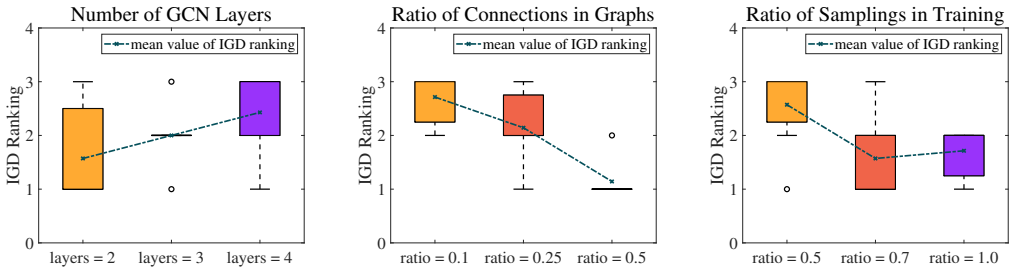


Fig. 2: The sensitivity analysis of hyperparameters, all experiments are conducted over 10 independent runs.

acquisition function based on the predictions of the regression model and the classifier as follows:

$$\boldsymbol{AF}_{RC}(\mathbf{x}) = \alpha \cdot \hat{\mathbf{f}} - (1-\alpha) \cdot p_0, \tag{8}$$

where α is a trade-off parameter. Generally, learning the functional relationship between decision variables and each objective value is more difficult than that between the decision valuables and the domination relationship. Accordingly, α is adapted to the process of the optimization. In the beginning, the selection relies more on the prediction provided by the classifier, as the regression model may fail to effectively estimate the candidates. This leads to a smaller value assigned to α. With the optimization processes, more data are acquired, further enhancing the quality of the regression model. As a result, the fitness predictions can play a more important role in the selection of new samples. That is, the value of α should be adapted as follows:

$$\alpha = FE/FE_{max}, \tag{9}$$

where FE is the current fitness evaluation number and FE_{max} denotes the maximum number of available fitness evaluations.

IV. Experiments

A. Experimental Settings

1) *Test Problems*: To test the effectiveness of the proposed GNN-assisted evolutionary algorithm, we have selected a widely used test suite of scalable multi-objective test problems, i.e., the DTLZ [34] test suite. For all the test instances used in the experimental studies, the number of decision variables D is set to 10, and the number of objectives is set to $M = 3$ and 5, respectively.

2) *Performance Indicators*: The inverted generational distance (IGD) [35], is adopted as the performance indicator to evaluate the quality of the non-dominated solutions obtained by each algorithm. We denote a set of uniformly distributed solutions sampled from the objective space along the true Pareto front as P^*, and the Parteo front approximation as $\hat{P}$ obtained approximation to the PF. IGD is calculated as follows:

$$IGD(P^*, \hat{P}) = \frac{\sum_{v \in P^*} d(v, \hat{P})}{|P^*|} \tag{10}$$

where $d(v, \hat{P})$ is the minimum Euclidean distance between v and all points in $\hat{P}$. The smaller IGD value, the better the achieved solution set is.

3) *Parameter Settings*: All algorithms use evolutionary algorithms as an optimizer, and the population size is set to 100. After 20 generations of optimization based on surrogate models, three promising candidate solutions are evaluated by the true objective functions. The maximum number of real fitness evaluations during the whole evolutionary process is set to 300.

In our experiments, we perform each algorithm 10 times on each test instance and record the corresponding mean and standard deviations (std) of the IGD results. Furthermore, we also use the Wilcoxon rank sum test at a significance level of 0.05 to estimate whether there is a significant difference between the proposed method and other algorithms under comparison. The symbol "(–)" indicates that a significantly better performance is achieved by the proposed algorithm, while the symbol "(+)" indicates the compared algorithm achieves a significantly better performance. Additionally, the symbol "($\approx$)" indicates that the compared and proposed algorithms show similar performance in terms of IGD values.

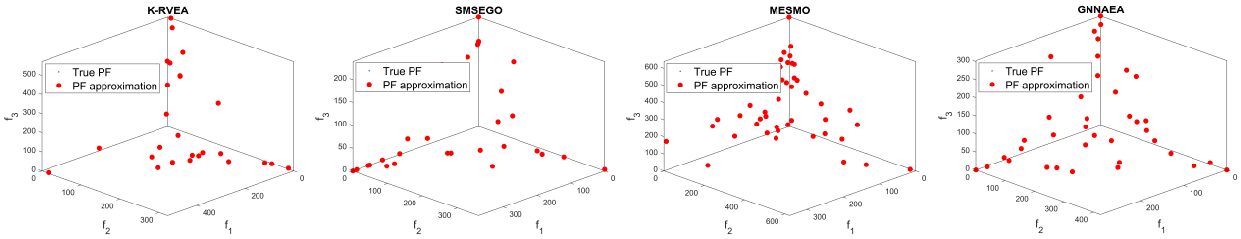
B. Sensitivity Analysis

We conduct a sensitivity analysis on three hyperparameters: the number of GCN layers in the GNN block, the ratio of connections in the generated graphs r_c, and the ratio of sampling data to generate the graph r_s. The experimental results are obtained over 10 independent runs on three-objective DTLZ1-DTLZ7, where the ranking of IGD values are presented in Fig. 2.

As we can see in Fig. 2 (a), the performance of the algorithm degrades with an increase in the number of GCN layers. This can be attributed to over-smoothing [36], wherein the performance of GNNs diminishes with deeper structures as nodes tend to exhibit similar or indistinguishable representations. The IGD ranking decreases with the ratio of connections r_c in the generated graph increasing from 0.1 to 0.5 in Fig. 2 (b), which is reasonable as nodes can gather more information in a dense graph. Based on the mean value of the IGD ranking in Fig. 2 (c), 70% of the data yields the optimal performance to balance between the overfitting and sample size. Therefore, we adopt

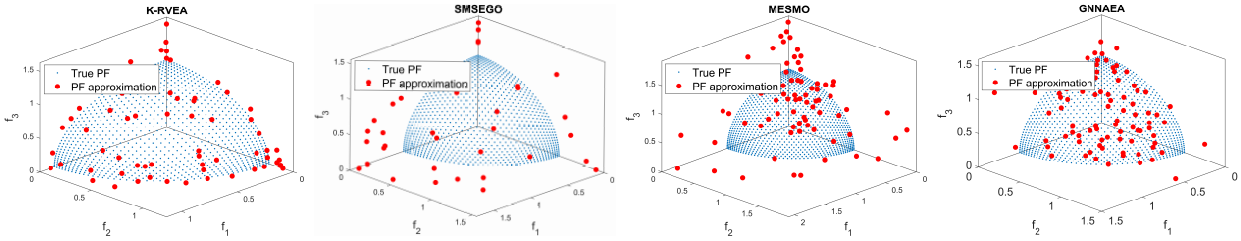
TABLE I: Mean (Standard Deviation) IGD values obtained by K-RVEA, SMSEGO, MESMO and GNNAEA for MOPs with $M = 3$ and 5.

Problem	M	K-RVEA	SMSEGO	MESMO	GNNAEA
DTLZ1	3	1.04e+2 (1.53e+1) –	1.05e+2 (3.22e+1) –	8.91e+1 (1.61e+1) –	6.72e+1 (2.34e+1)
	5	5.31e+1 (1.40e+1) –	5.58e+1 (1.27e+1) –	/	4.57e+1 (1.04e+1)
DTLZ2	3	1.87e–1 (5.19e–2) –	3.18e–1 (4.11e–2) –	1.43e–1 (2.41e–2) –	1.13e–1 (3.23e–3)
	5	2.84e–1 (2.15e–2) –	3.94e–1 (1.28e–2) –	/	2.09e–1 (1.11e–2)
DTLZ3	3	2.49e+2 (5.18e+1) –	2.47e+2 (6.25e+1) –	2.18e+2 (1.86e+0) ≈	2.14e+2 (4.32e+1)
	5	1.45e+2 (3.76e+1) ≈	1.64e+2 (4.00e+1) –	/	1.40e+1 (4.41e+1)
DTLZ4	3	4.40e–1 (9.82e–2) +	7.21e–1 (1.51e–1) –	4.32e–1 (1.69e–1) +	6.13e–1 (9.60e–2)
	5	4.40e–1 (9.82e–2) +	7.63e–1 (6.51e–2) ≈	/	8.00e–1 (5.41e–2)
DTLZ5	3	1.05e–1 (3.29e–2) –	1.85e–1 (4.58e–2) –	1.70e–1 (1.34e–2) –	6.46e–2 (4.23e–3)
	5	4.56e–2 (1.41e–2) +	1.28e–1 (2.52e–2) –	/	6.07e–2 (8.80e–3)
DTLZ6	3	3.00e+0 (4.81e–1) +	4.99e+0 (3.96e–1) +	4.53e+0 (1.67e+0) +	6.66e+0 (2.69e–2)
	5	1.94e+0 (2.06e–1) +	4.11e+0 (4.03e–1) ≈	/	4.83e+0 (4.01e–2)
DTLZ7	3	1.29e–1 (7.67e–3) +	2.58e+0 (1.10e+0) –	1.17e+0 (6.14e–1) –	6.42e+0 (8.23e–2)
	5	4.95e–1 (4.42e–2) +	1.15e+0 (2.77e–1) +	/	6.73e+0 (1.62e+0)
+/–/ ≈		6/6/2	2/10/2	2/4/1	



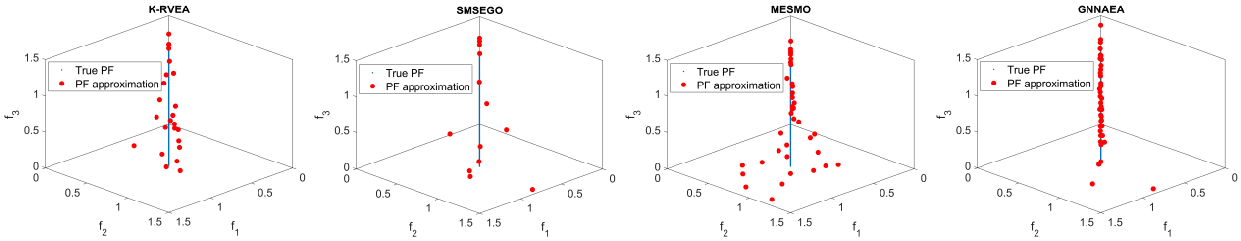
(a) (b) (c) (d)

Fig. 3: The final solution set with the median IGD values found by K-RVEA, SMSEGO, MESMO and GNNAEA on DTLZ1 with $M = 3$.



(a) (b) (c) (d)

Fig. 4: The final solution set with the median IGD values found by K-RVEA, SMSEGO, MESMO and GNNAEA on DTLZ2 with $M = 3$.



(a) (b) (c) (d)

Fig. 5: The final solution set with the median IGD values found by K-RVEA, SMSEGO, MESMO and GNNAEA on DTLZ5 with $M = 3$.

TABLE II: Mean (Standard Deviation) IGD values obtained by GNNAEA with its three variants MLP1-AEA, MLP2-AEA, and GNN2-AEA for MOPs with $M = 3$.

Problem	M	MLP1-AEA	MLP2-AEA	GNN2-AEA	GNNAEA
DTLZ1	3	9.36e+1 (1.38e+1) –	9.89e+1 (1.30e+1) –	1.05e+2 (1.39e+1) –	6.72e+1 (2.34e+1)
DTLZ2	3	3.57e–1 (2.53e–2) –	3.08e–1 (3.82e–2) –	2.05e–1 (1.88e–2) –	1.13e–1 (3.23e–3)
DTLZ3	3	3.50e+2 (6.84e+1) –	2.83e+2 (6.01e+1) –	3.03e+2 (6.14e+1) –	2.14e+2 (4.32e+1)
DTLZ4	3	7.19e–1 (6.53e–2) –	6.50e–1 (9.26e–2) –	7.43e–1 (6.07e–2) –	6.13e–1 (9.60e–2)
DTLZ5	3	2.70e–1 (3.55e–2) –	2.37e–1 (2.23e–2) –	1.05e–1 (1.67e–2) –	6.46e–2 (4.23e–3)
DTLZ6	3	6.16e+0 (1.28e–1) +	6.15e+0 (8.40e–1) +	6.75e+0 (9.24e–2) –	6.66e+0 (2.69e–2)
DTLZ7	3	1.17e+0 (4.42e–1) +	1.63e+0 (2.42e–1) +	5.73e+0 (1.54e+0) +	6.42e+0 (8.23e–2)
+/–/ ≈		2/5/0	2/5/0	1/6/0	

the number of GCN layers being 2, $r_c = 0.5$, and $r_s = 0.7$ in the following experiments.

C. Comparative Experimental Results

We compare the proposed algorithm with several representative and state-of-the-art surrogate-assisted evolution algorithms, i.e., K-RVEA [7], SMSEGO [37] and MESMO [38]. We first test all algorithms on the DTLZ test suite with $M = 3$ and 5 and the results of IGD values are presented in Table I. Note that the IGD results for $M = 5$ achieved by MESMO are not given in Table I due to its unaffordable computation cost. Accordingly, we can observe that the proposed GNNAEA achieves the best overall performance on the DTLZ test suite, compared with K-RVEA, SMSEGO and MESMO. Specifically, GNNAEA can find better Pareto front approximations on DTLZ1, DTLZ2, DTLZ3 and DTLZ5. This indicates that the proposed surrogate model can effectively guide the optimization of the expensive multi-objective optimization. The proposed algorithm fails to address DTLZ6 and DTLZ7, where K-RVEA shows promising performance. A possible explanation is that DTLZ6 has a degenerate Pareto front, which is always a curve in the hyper-space, regardless of the number of objectives; DTLZ7 has a disconnected Pareto front, where the number of segments can be as large as 2^{M-1}, where M is the number of objectives. While learning the topology of the Pareto front of DTLZ6 and DTLZ7 is challenging for GNNAEA, K-RVEA has shown its ability to address problems with degenerate or disconnected Pareto fronts [39].

To take a closer look at the results achieved by each algorithm, Figs. 3-5 illustrates the Pareto front approximations on DTLZ1, DTLZ2 and DTLZ5 with $M = 3$, respectively. Specifically, Fig. 3 shows the superiority of GNNAEA as the Pareto front approximation found by GNNAEA is better than that found by all other algorithms under comparison. Despite that K-RVEA also converges to the true Pareto front, it fails to maintain the diversity of the population. By contrast, MESMO is able to achieve a set of non-dominated solutions with good diversity, which is, however, far from the true Pareto front. Similar observations can be made from the results of DTLZ2 and DTLZ5. Therefore, the proposed GNNAEA shows competitive performance on MOPs, indicating the efficiency of the GNN-based surrogate model and the designed new acquisition function.

D. Ablation Study

In this subsection, we analyze the effectiveness of the proposed surrogate model by comparing it with three variants, namely, MLP1-AEA, MLP2-AEA, and GNN2-AEA. To be specific, MLP1-AEA and MLP2-AEA replace the GNN block with one-layer MLP and two-layer MLP with sigmoid activating function, respectively. On the other hand, in GNN2-AEA, two GNN models are employed and updated independently to generate embeddings for the classification and regression tasks, respectively. The results are shown in Table II.

In general, the proposed method GNNAEA performs the best in five instances out of seven instances. Two GNN-based methods outperform the MLP-based ones in most cases, indicating the information processing and extraction capabilities of GNNs. Additionally, the superior performance of GNNAEA compared to GNN2-AEA indicates that a single GNN, concurrently driven by updates for both classification and regression tasks, can generate higher-quality representations. In other words, the classification and regression tasks mutually enhance each other's performance.

V. Conclusion

In this paper, we represent solutions as a graph, based on which we propose a new surrogate based on the GNN model to learn the graph structure. Afterward, regression models to predict the objective values and a classification model to provide the domination relationship are constructed. Finally, a new acquisition function is proposed to carefully trade off between exploration and exploitation, resulting in effective new sample selection. The performance of the proposed GNNAEA and the effectiveness of the proposed surrogate models and new acquisition function are validated on the DTLZ test suite under the comparison with three representative surrogate-assisted multi-objective evolutionary algorithms.

Despite that GNNAEA provides insights into the use of GNN models to leverage the graph structure information from the search space, it fails to address some multi-objective optimization problems with degenerate and discontinuous PFs. Therefore, further investigation on how to effectively learn the topology of irregular PFs is an interesting research direction. Moreover, the generation of graphs based on the solutions in search space impacts the quality of the GNN models, which deserves more exploration.

REFERENCES

[1] Y. Jin and B. Sendhoff, "A systems approach to evolutionary multiobjective structural optimization and beyond," *IEEE Computational Intelligence Magazine*, vol. 4, no. 3, pp. 62–76, 2009.

[2] B. Liu, Q. Zhang, and G. G. Gielen, "A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 180–192, 2013.

[3] S. Liu, H. Wang, W. Peng, and W. Yao, "A surrogate-assisted evolutionary feature selection algorithm with parallel random grouping for high-dimensional classification," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 5, pp. 1087–1101, 2022.

[4] S. Wang, J. Liu, and Y. Jin, "Surrogate-assisted robust optimization of large-scale networks based on graph embedding," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 4, pp. 735–749, 2019.

[5] R. Shi, J. Luo, and Q. Liu, "Fast evolutionary neural architecture search based on bayesian surrogate model," in *2021 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2021, pp. 1217–1224.

[6] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen, "Data-driven evolutionary optimization: An overview and case studies," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 442–458, 2018.

[7] 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 Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 129–142, 2016.

[8] L. Willmes, T. Back, Y. Jin, and B. Sendhoff, "Comparing neural networks and kriging for fitness approximation in evolutionary optimization," in *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.*, vol. 1. IEEE, 2003, pp. 663–670.

[9] H. Wang and Y. Jin, "A random forest-assisted evolutionary algorithm for data-driven constrained multiobjective combinatorial optimization of trauma systems," *IEEE Transactions on cybernetics*, vol. 50, no. 2, pp. 536–549, 2018.

[10] R. G. Regis, "Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 326–347, 2013.

[11] Y. Jin, M. Olhofer, and B. Sendhoff, "A framework for evolutionary optimization with approximate fitness functions," *IEEE Transactions on evolutionary computation*, vol. 6, no. 5, pp. 481–494, 2002.

[12] Y. Tenne, K. Izui, and S. Nishiwaki, "Handling undefined vectors in expensive optimization problems," in *European Conference on the Applications of Evolutionary Computation*. Springer, 2010, pp. 582–591.

[13] S. D. Handoko, C. K. Kwoh, and Y.-S. Ong, "Feasibility structure modeling: An effective chaperone for constrained memetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 5, pp. 740–758, 2010.

[14] J. Poloczek and O. Kramer, "Local svm constraint surrogate models for self-adaptive evolution strategies," in *KI 2013: Advances in Artificial Intelligence: 36th Annual German Conference on AI, Koblenz, Germany, September 16-20, 2013. Proceedings 36*. Springer, 2013, pp. 164–175.

[15] D. Lim, Y. Jin, Y.-S. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 329–355, 2009.

[16] H. Wang, Y. Jin, and J. Doherty, "Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems," *IEEE Transactions on cybernetics*, vol. 47, no. 9, pp. 2664–2677, 2017.

[17] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global optimization*, vol. 13, pp. 455–492, 1998.

[18] J. Kakkad, J. Jannu, K. Sharma, C. Aggarwal, and S. Medya, "A survey on explainability of graph neural networks," *arXiv preprint arXiv:2306.01958*, 2023.

[19] X. Wang, J. Wang, K. Zhang, F. Lin, and Q. Chang, "Convergence and objective functions of noise-injected multilayer perceptrons with hidden multipliers," *Neurocomputing*, vol. 452, pp. 796–812, 2021.

[20] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

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

[22] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI open*, vol. 1, pp. 57–81, 2020.

[23] K. Do, T. Tran, and S. Venkatesh, "Graph transformation policy network for chemical reaction prediction," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 750–760.

[24] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 922–929.

[25] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *The world wide web conference*, 2019, pp. 417–426.

[26] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[27] C. He, Y. Zhang, D. Gong, and X. Ji, "A review of surrogate-assisted evolutionary algorithms for expensive optimization problems," *Expert Systems with Applications*, p. 119495, 2023.

[28] Y. He, J. Sun, P. Song, and X. Wang, "Dual kriging assisted efficient global optimization of expensive problems with evaluation failures," *Aerospace Science and Technology*, vol. 105, p. 106006, 2020.

[29] J. Yi, L. Gao, X. Li, C. A. Shoemaker, and C. Lu, "An on-line variable-fidelity surrogate-assisted harmony search algorithm with multi-level screening strategy for expensive engineering design optimization," *Knowledge-based systems*, vol. 170, pp. 1–19, 2019.

[30] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.

[31] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[32] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[33] H. Li, M. Wang, S. Liu, P.-Y. Chen, and J. Xiong, "Generalization guarantee of training graph convolutional networks with graph topology sampling," in *International Conference on Machine Learning*. PMLR, 2022, pp. 13 014–13 051.

[34] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, vol. 1. IEEE, 2002, pp. 825–830.

[35] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.

[36] C. Yang, R. Wang, S. Yao, S. Liu, and T. Abdelzaher, "Revisiting over-smoothing in deep gcns," *arXiv preprint arXiv:2003.13663*, 2020.

[37] T. Wagner, M. Emmerich, A. Deutz, and W. Ponweiser, "On expected-improvement criteria for model-based multi-objective optimization," in *Parallel Problem Solving from Nature, PPSN XI: 11th International Conference, Kraków, Poland, September 11-15, 2010, Proceedings, Part I 11*. Springer, 2010, pp. 718–727.

[38] S. Belakaria, A. Deshwal, and J. R. Doppa, "Max-value entropy search for multi-objective Bayesian optimization," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[39] K. Wan, C. He, A. Camacho, K. Shang, R. Cheng, and H. Ishibuchi, "A hybrid surrogate-assisted evolutionary algorithm for computationally expensive many-objective optimization," in *2019 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2019, pp. 2018–2025.