Self-adaptive Teaching-learning-based Optimizer with Improved RBF and Sparse Autoencoder for Complex Optimization Problems

Jing Bi, Ziqi Wang, Haitao Yuan, Junfei Qiao, Jia Zhang, MengChu Zhou

Abstract

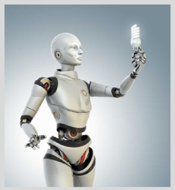
Evolutionary algorithms are commonly used to solve many complex optimization problems in such fields as robotics, industrial automation, and complex system design. Yet, their performance is limited when dealing with high dimensional complex problems because they often require enormous computational resources to yield desired solutions, and they may easily trap into local optima. To solve this problem, this work proposes a Self-adaptive Teaching-learning-based Optimizer with an improved Radial basis function model and a sparse Autoencoder (STORA). In STORA, a Self-adaptive Teaching-learning-based Optimizer is designed to dynamically adjust parameters for balancing exploration and exploitation during its solution process. Then, a sparse autoencoder (SAE) is adopted as a dimension reduction method to compress search space into lower-dimensional one for more efficiently guiding population to converge towards global optima. Besides, an Improved Radial Basis Function model (IRBF) is designed as a surrogate model to balance training time and prediction accuracy. It is adopted to save computational resources for improving overall performance. In addition, a dynamic population allocation strategy is adopted to well integrate SAE and IRBF in STORA. We evaluate it by comparing it with several stateof-the-art algorithms through six benchmark functions. We further test it by applying it to solve a real-world computational offloading problem.

Introduction

Evolutionary algorithms (EAs) have been widely applied to solve different types of benchmarks and real-world engineering problems in a variety of fields, e.g., computer vision, robots, cloud computing and manufacturing scheduling problems. Some practical problems have a large number of decision variables and can be characterized as high-dimensional problems. These problems present a exponentially growing search space with many decision variables that bring big challenges for EAs to efficiently explore the search space. In other words, they often require a large number of function evaluations (FEs) to yield satisfactory solutions. However, FEs in many real-world problems can be computationally intensive or highly costly. Moreover, some of EAs may easily trap into local optima when solving high-dimensional problems. As a result, it is important to balance exploration and exploitation abilities of EAs during their optimization process.

To solve high-dimensional and complex problems, a number of studies have been proposed, which can be divided into two types. The first type incorporates surrogate models into EAs. Surrogate-assisted EAs (SAEAs) have been considered as viable methods to deal with high-dimensional problems. A surrogate model can be employed to replace a part of a true model for evaluating individuals. It takes fewer computation resources than the true model. However, SAEAs bring additional surrogate models into the structure that also brings additional training time especially for a high dimensional training set. Moreover, the optimization process is partially guided by surrogate models whose accuracy has direct impact on the optimization direction. Inaccurate surrogate models may mislead the optimization direction and result in poor or inaccurate search results.

The second type belongs to the dimension reduction, which is widely adopted to deal with the huge amount of high-dimensional data because of the curse of dimensionality. It aims to extract useful features of data to reduce the dimension of objective functions or the search space for reducing computational stress. However, although the feature data is extracted under a specific dimension reduction method, some data including important information for the optimization process may be lost. As a result, it is highly important to choose a proper method and suitable time for dimension reduction.



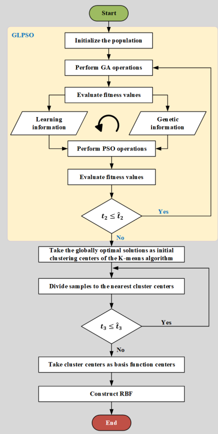
Proposed Framework

We adopt Sparse autoencoder (SAE) to compress a high dimensional space into a reduced one for facilitating the evolution. In STORA, its initial generations are conducted by STO for providing samples to train the SAE. As the population is evolving towards better regions, the trained SAE can extract some important information of promising evolution directions to compress the dimension of individuals. When the termination condition is reached, SAE is trained and used in the next stage.

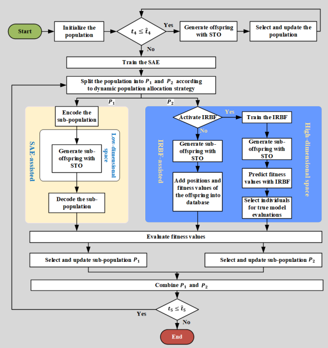
Moreover, this work proposes an IRBF as a surrogate model to solve this problem. We extract characteristics of data to construct neural networks for simplifying the network structure. To realize it, we adopt the K-means algorithm to select centers of basis function, which locate important areas of the input space after the clustering. Furthermore, Genetic learning particle swarm optimization (GLPSO) is adopted in our structure to optimize initial centers of the K-means algorithm.

At the beginning of STORA, population P is initialized randomly in the decision space by Latin hypercube sampling (LHS). Then, several generations of evolution are carried out by STO to collect data samples for the training of SAE. Once the preset condition is reached, SAE is constructed based on the accumulated data samples. After the SAE training, the population is split into two sub-populations (P1 and P2) with the dynamic population allocation strategy to be introduced next. Then, P1 and P2 coevolve in a distributed manner to ensure diversity. P1 is assisted by SAE to find promising solutions rapidly and P2 is guided by STO (possibly assisted by IRBF) in the original space. The diversity of the population helps STORA to jump out of local optima that are imperative to the optimization process.

In the SAE-assisted STO, P1 is first encoded by the trained SAE into a lower-dimensional space. Then, STO is adopted to generate offspring. In that case, individuals have higher possibility to find promising offspring in the relatively lowdimensional space to speed up the optimization process. Due to the dimensional mismatch, FEs cannot be completed in the low dimensional space. After the decoding phase of the SAE, the population is in the original and high dimensional space, its individuals can be directly evaluated by the fitness function. Finally, new P1 is updated for the next generation. Furthermore, in the IRBF-assisted STO, P2 evolves with STO before the activation of IRBF. In addition, all positions and their fitness values in previous iterations are stored in a database for later training of IRBF. Once the activation condition is met, IRBF is trained based on the collected data samples, and it is adopted to prescreen individuals in the rest of the optimization process. To be specific, after STO generates the offspring, the positions of the offspring are the input of IRBF that outputs the predicted fitness values of those individuals. Furthermore, to ensure the search accuracy, some individuals still need to be selected for the true model evaluation. In STORA, individuals are sorted based on their predicted fitness values, the top M individuals are selected for the true model evaluation because they have higher possibility to find optima quickly. Then, new P2 is updated for the next generation. New P1 and P2 are combined together to form a new population P after each iteration. The whole process continues until the termination condition is met.



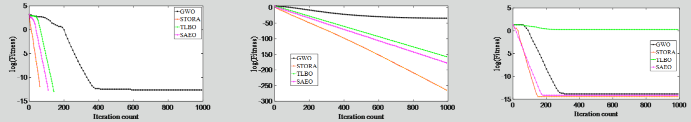
Structure of IRBF



Structure of STORA

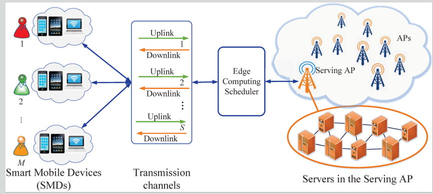
Experimental Results

We compare STORA with two metaheuristic algorithms (teaching-learning-based optimization (TLBO) and grey wolf optimizer (GWO)) and one recently proposed algorithm (SAEO), which is suitable to solve highdimensional problems. We choose six different benchmark functions including unimodal and multimodal functions. The results shows that STORA has stable performance and great robustness and STORA yields the best search result with the least time among all compared algorithms.

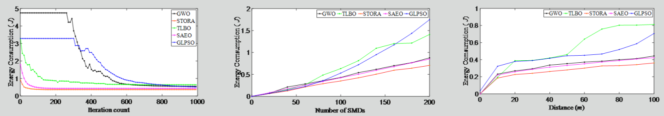


Part of results on benchmark functions

We apply STORA to solve a real-world computation task offloading problem in an edge-computing-enabled large-scale factory. This problem considers to migrate a part of the data processing of mobile applications from resource-constrained smart mobile devices (SMDs) to highperforming platforms in a network edge, which is known as computation offloading. The optimized decision variables include the computational speed of each SMD, its data transmission power, and task offloading ratio. Moreover, the constraints include maximum latency for executing applications, maximum transmission power and maximum computational speed of each SMD.



We compare STORA with GWO, TLBO, SAEO and GLPSO by applying them to solve the above problem. The results demonstrate that STORA can be well applied to real-world problems, it yields higher-quality solutions that meeting all constraints than its typical peers.



Part of results on the real-world problem

Conclusions

This work presents a Self-adaptive Teaching-learning-based Optimizer with an improved Radial basis function model and a sparse Autoencoder (STORA) for complex optimization problems. First, to trade off the exploration and exploitation abilities, a Self-adaptive Teaching-learning-based Optimizer (STO) is designed to adjust parameters in the search process. Second, a sparse autoencoder (SAE) is adopted to speed up optimization in a high-dimensional space and give more possibility to worse individuals evolving towards promising areas. Third, an Improved Radial Basis Function model (IRBF) is designed as a surrogate model to find better solutions with fewer computing resources and less training time. Then, a dynamic population allocation strategy is designed to enhance integration of SAE and IRBF for improved performance of STORA. Finally, STORA is compared against its peers on six high-dimensional benchmark functions. Experimental results demonstrate that STORA yields the best search results with the least time among all compared algorithms. Then, we apply STORA to solve a real-world computational offloading problem in an edge computing environment, and results show that STORA yields higher-quality solutions meeting all constraints than its typical peers. Our next work should extend it to solve manyobjective optimization high-dimensional problems with discrete and continuous parameters. In addition, other advanced surrogate models can be applied to better solve these problems, thus further improving performance of STORA.