# Comparison of Different Surrogate Models for the JADE Algorithm

Konrad Krawczyk[a] and Jarosław Arabas[b]
*Institute of Computer Science, Warsaw University of Technology, Warsaw, Poland*

Keywords: Evolutionary Algorithms, Surrogate Models, Differential Evolution, Ranking Procedure, JADE, CEC.

Abstract: We investigate the performance of various regression-based surrogate models integrated with a ranking procedure in the Adaptive Differential Evolution with an Optional External Archive (JADE) method. We perform regression of the fitness function by the surrogate model to reduce the number of fitness evaluations needed to achieve the optimization progress. The surrogate model training process should be relatively cheap since training is performed many times along with the optimization process. Therefore we investigate surrogate models based on k Nearest Neighbors, Random Forests, and Support Vector Machines. We test the effectiveness of JADE with and without the surrogate models using the CEC2013 benchmark set for single-criterion continuous optimization. Experimental data confirm the benefits of using the surrogate models and indicate the difference in efficiency improvement between the considered models.

## 1 INTRODUCTION

In this text, we aim to improve the efficiency of the Adaptive Differential Evolution with an Optional External Archive (JADE) algorithm (Zhang and Gao, 2013), which is a popular evolutionary algorithm and has demonstrated remarkable performance in single-criterion continuous optimization tasks. However, the computational cost associated with the fitness evaluations required by the JADE algorithm can become a bottleneck, especially when dealing with computationally expensive simulations or real-world systems. This problem can be tackled by using surrogate models, also known as metamodels, to reduce the computing time by providing an estimation of the fitness function. We introduce a methodology to handle the surrogate model adaptation and investigate several candidates for the surrogate models to be integrated with JADE. We evaluate the efficiency of surrogate-assisted JADE using the CEC2013 bound-constrained, single-criterion benchmark suite.

Surrogate models can either directly return the estimation of the fitness function value of each individual or they can approximate the result of comparison of fitness values between two or more individuals. For example, in the case of Differential Evolution (DE), the selection process is based on the comparison between each offspring and its parent. One way of using surrogate models would be to compute the surrogate fitness values and to perform selection on the basis of their comparison. Note that the result of comparison affects the selection rather than the fitness values, therefore we may be less careful about the approximation accuracy of the fitness function with the surrogate model, provided that the proper relationship of the individuals' fitness is preserved by the surrogate values. Hence we can distinguish between the regression-based and the classification-based surrogate models. Example applications of both concepts for the DE can be found, e.g., in (Zhang and Sanderson, 2007) and in (X. Lu and Yao, 2011).

When the surrogate model of the fitness function is used, several decisions need to be made:

- **Individual vs. universal metamodel:** The surrogate function can be defined for each individual separately, or it can be universal for all population members. If the variability pattern of the fitness landscape is relatively similar among individuals, a single surrogate function may suffice; otherwise, assigning each individual its own surrogate function might lead to better approximations and ultimately improve the optimization process (Ong et al., 2004).
- **Frequency of re-training the surrogate function:** Retraining the surrogate model after each generation can provide a more accurate representation of the problem space, but it is quite expensive. Updating the model after a few generations

[a] https://orcid.org/0000-0003-2012-0803
[b] https://orcid.org/0000-0002-5699-947X



Krawczyk, K. and Arabas, J.
Comparison of Different Surrogate Models for the JADE Algorithm.
DOI: 10.5220/0012166100003595
In *Proceedings of the 15th International Joint Conference on Computational Intelligence (IJCCI 2023)*, pages 186-194
ISBN: 978-989-758-674-3; ISSN: 2184-3236

can reduce the computation cost but it can result in a poorer quality of the model, so the frequency of retraining reflects the desired trade-off between computational time and solution quality.

- **Handling the true fitness function samples:** Surrogate models are built on the basis of the true fitness function samples. Initially, such a sample can be created using a limited number of evaluations when the true fitness function is computed. As the optimization continues, the sample can be expanded by evaluating only a portion of the points created by the optimization process. This dynamic approach helps maintain a balance between exploration and exploitation and may result in a more accurate surrogate model over time (He et al., 2023).
- **Metamodel adaptation style:** The surrogate model can be retrained either from scratch, or it can be adaptively, incrementally changed. This can involve updating the model's parameters or structure, which can help in reducing the computational cost while maintaining a reasonable level of accuracy (Picheny et al., 2013).
- **Surrogate model class:** There are numerous types of surrogate models that can be used in optimization, including but not limited to kriging, linear regression, support vector machines, artificial neural networks, Gaussian process regression, k-nearest neighbors, and decision trees. The choice of the surrogate model depends on the problem domain, the available data, and the desired trade-off between model complexity and computational cost (Zaborski and Mańdziuk, 2022),(Zhang et al., 2022).

We define a regression-based surrogate model to be commonly used by all population individuals (a global surrogate model). The surrogate model is adaptively changed in each iteration of JADE. The surrogate model adaptation process is performed after the offspring population is completed. We adopted the approximate ranking procedure proposed in (S. Kern and Koumoutsakos, 2006) as a procedure to control the process of computing true fitness values, when the surrogate model approach is coupled with the CMA-ES method. The main idea is to control the surrogate model quality without knowing the correct ranking of the complete population.

In this study, we compare the performance of four widely used regression-based surrogate models (k Nearest Neighbors, Random Forest, XGBoost, and Support Vector Regression) when integrated with the JADE algorithm with the aforementioned ranking-based model adaptation procedure for single-criterion continuous optimization problems. We aim to identify the most suitable surrogate model for JADE by considering their performance and efficiency improvement.

To achieve this, we use the CEC2013 benchmark set for single-criterion continuous optimization, which covers a wide range of test functions with various optimization challenges. By analyzing the performance of JADE with a surrogate model employed in the ranking procedure, we aim to provide insights into the effectiveness of different surrogate models and their potential to enhance the JADE algorithm's performance.

# 2 RELATED WORK

## 2.1 JADE Algorithm

The Adaptive Differential Evolution with an Optional External Archive (JADE) is an extended version of the Differential Evolution algorithm, as presented in (Zhang and Sanderson, 2009). A key aspect of JADE is its self-regulating control of the mutation and crossover parameters, $F$ and $CR$, which are essential for achieving a balance between exploration and exploitation during the optimization process. JADE utilizes a learning strategy that updates $F$ and $CR$ based on successful mutations, allowing the algorithm to adjust to the problem's characteristics throughout the optimization process. This adaptive approach removes the need for manual parameter tuning, making JADE more robust and effective for solving a wide range of optimization problems.

Another remarkable feature of JADE is the optional utilization of an external archive, which stores the solutions that are replaced during the evolutionary process. The external archive enhances the algorithm's ability to preserve population diversity and prevent premature convergence. By integrating the archived solutions into the mutation process, JADE can efficiently explore the search space and identify promising regions, resulting in improved optimization performance.

In JADE, the DE/current-to-pbest/k mutation scheme is used. It selects the $p\%$ best individuals from the current population, where $p\%$ is a user-defined parameter, and combines them with the current individual and a randomly selected individual from either the current population or the external archive. This mutation operation can be expressed as follows:

$$m_i = x_i + F \cdot (x_j - x_i) + F \cdot (x_{i_1} - x_{i_2}) \qquad (1)$$

where $m_i$ is the mutant, $x_i$ is the target vector, $x_j$ is a vector chosen from the fraction of $p\%$ best fit vectors from the current population, and $x_{i_1}$ and $x_{i_2}$ are randomly selected individuals from either the current population or the external archive. A more comprehensive pseudocode of JADE is provided in section 3, along with the adaptations made to include the surrogate model adaptation.

## 2.2 Approximate Ranking Procedure

In (S. Kern and Koumoutsakos, 2006) the method called approximate ranking procedure as assistance to the surrogate model approach implemented in the CMA-ES is proposed. Its main task is to help control model quality without knowing the correct ranking of the complete population. In that procedure, in the beginning, a surrogate model is prepared based on the training set $S$. With the use of the prepared surrogate model, a sorting order for $\mu$ best individuals is prepared. Next, a fraction of offspring individuals is evaluated with the original fitness function and added to $S$, the surrogate model is updated, and the sorting order is established anew. If this results in a change of the sorting order, the process of evaluating a fraction of offspring, model update, and ranking is repeated until the ranking stabilizes.

## 2.3 Considered Surrogate Models

**K Nearest Neighbours.** kNN is a non-parametric, instance-based model (Dudani, 1976) which uses an archive of points $S$. When given an input point $x$, the model defines its output $\hat{f}(x)$ as:

$$\hat{f}(x) = \frac{\sum_{i=1}^{k} w_i \cdot f(x_i)}{\sum_{i=1}^{k} w_i} \tag{2}$$

where $f(x_i)$ is the approximated function value for the $i$-th closest neighbor of the point $x$ from the set $S$, $w_i$ is its weight, and the summation runs over the $k$ nearest neighbors. The weights are assigned to the neighbors based on the inverse of their distance to the query point:

$$w_i = \frac{1}{||x_i - x||} \tag{3}$$

where $||\cdot||$ is the assumed norm (in our case, Euclidean).

kNN is a non-parametric method, which means it does not make any assumptions about the approximated function formula. This flexibility allows it to capture complex relationships between the input and output variables, making it suitable for a wide range of optimization problems (James et al., 2021). The choice of $k$ influences the model's performance, with smaller values providing a more flexible fit and larger values offering a smoother approximation.

**Random Forest Regression.** RFR is an ensemble learning method that combines multiple regression trees to produce a more accurate and robust prediction (Breiman, 2001).

Each regression tree is constructed independently using a bootstrapped sample of the training set $S$. During the construction of each tree, a random subset of features is selected at each node, which helps to decorrelate the individual trees and improve the overall performance of the ensemble. The final prediction for an input point $x$ is obtained by averaging the outputs of all the individual trees in the forest:

$$\hat{f}(x) = \frac{1}{T} \sum_{i=1}^{T} \hat{f}_i(x) \tag{4}$$

where $\hat{f}_i(x)$ is the output of the $i$-th decision tree, and $T$ is the total number of trees in the forest.

The number of trees ($T$) and the size of the random feature subset at each node are important hyperparameters that influence the performance of the RFR model.

**XGBoost.** XGBoost builds on the principles of gradient boosting, a method that sequentially combines regression trees into a weighted average that can accurately model complex relationships between input and output variables (Chen and Guestrin, 2016). The key idea behind XGBoost is to iteratively learn a regression tree that can best correct the errors made by the previously combined trees. The final prediction for an input point $x$ is defined as:

$$\hat{f}(x) = \sum_{i=1}^{T} w_i \hat{f}_i(x) \tag{5}$$

where $T$ is the number of trees, $\hat{f}_i(x)$ is the prediction of the $i$-th tree for the input point $x$, and $w_i$ is the corresponding weight. Important hyperparameters in XGBoost include the maximum tree depth, learning rate (shrinkage), the number of trees, and regularization parameters (L1 and L2).

**Support Vector Regression.** SVR (Drucker et al., 1997) is a regression version of the Support Vector Machine (Cortes and Vapnik, 1995). SVR yields its prediction as a weighted sum of a form:

$$\hat{f}(x) = \sum_{x_i \in S} w_i k(x, x_i) + b \tag{6}$$

where $w_i$ are the weights, $b$ is the bias term, and $k(x, x_i)$ is the kernel function which takes as arguments a point $x_i$ from the dataset $S$ and the input point $x$. The archive points whose corresponding weights are nonzero are called the support vectors.

A critical aspect of SVR is the proper choice of the kernel function which determines the transformation of the input data and the shape of the approximation hyperplane. Commonly used kernels include the linear, polynomial, radial basis, and sigmoid. Other important hyperparameters in SVR include the regularization parameter $C$, which controls the trade-off between model complexity and the degree of allowed error, and the $\varepsilon$ parameter, which defines the acceptable error margin around the optimal hyperplane.

# 3 JADE WITH A SURROGATE MODEL

The use of the fraction of $p\%$ best points in the JADE mutation scheme is analogous to the selection scheme used by CMA-ES. This motivated us to couple JADE with a ranking procedure originally formulated for CMA-ES in (S. Kern and Koumoutsakos, 2006). Pseudocodes of the resulting JADE version and of the approximate ranking procedure are presented as Algorithm 1 and Algorithm 2, respectively.

$P \leftarrow \{x_1, x_2, ..., x_\mu\}$
$S \leftarrow (x_k, f(x_k)), k = 1, ..., \mu$
$\mu_{CR} = 0.5, \mu_F = 0.5$
**while** *!stop* **do**
  $S_{CR} = \emptyset, S_F = \emptyset$
  $P_{pbest} \leftarrow find\_pbest(q(P))$
  **for** $i := 1$ **to** $\mu$ **do**
    $CR_i = randn_i(\mu_{CR}, 0.1)$
    $F_i = randc_i(\mu_F, 0.1)$
    $x_{i_1}, x_{i_2} \leftarrow sample(P)$
    $x_{pbest} \leftarrow sample(P_{pbest})$
    $m_i \leftarrow x_i + F_i \cdot (x_{pbest} - x_i + x_{i_1} - x_{i_2})$
    $o_i \leftarrow crossover(x_i, m_i)$
  **end**
  $\hat{f}, q(O), S \leftarrow ranking\_procedure(O, S)$
  $q(P) \leftarrow \hat{f}(P)$
  **for** $i := 1$ **to** $\mu$ **do**
    **if** $q(o_i) < q(x_i)$ **then**
      $x_i \leftarrow o_i$
      $S_{CR} \leftarrow S_{CR} \cup \{CR_i\}$
      $S_F \leftarrow S_F \cup \{F_i\}$
    **end**
  **end**
  $\mu_{CR} = (1-c) \cdot \mu_{CR} + c \cdot mean_A(S_{CR})$
  $\mu_F = (1-c) \cdot \mu_F + c \cdot mean_L(S_F)$
**end**

Algorithm 1: JADE with the surrogate model-based ranking procedure.

In the modified JADE, we distinguish between the true fitness function and the surrogate function, denoted with $f$ and $\hat{f}$, respectively. The fitness value of a particular individual $x_i$, maintained in a population, is denoted by $q(x_i)$ and is assigned, depending on the context, either by the true fitness or the surrogate function. Every point evaluated by the true fitness function is stored in the archive $S$ which is organized as a FIFO queue. This archive is used to train the surrogate models and should be distinguished from the optional archive of points, which is maintained by JADE and is used to increase the number of difference vectors.

The algorithm starts by generating the initial population $P$ and evaluating all the individuals in the population using the original fitness function $f$. As the optimization loop begins, the population of $p\%$ best individuals is identified on the basis of either the original fitness function (when available) or the surrogate model. The offspring individuals are generated using the mutation and crossover, and they are evaluated using the ranking procedure that updates the surrogate model. After that, the parents' fitness values obtained using the previous surrogate model are re-calculated using the updated surrogate model. The selection of

$\hat{f} \leftarrow build\_surrogate\_model(S)$
$q(P) \leftarrow \hat{f}(P)$
$r_0(P) \leftarrow$ sort according to $q(P)$
$S \leftarrow S \cup \{(x_i, f(x_i))\}$, i = 1, ..., $n_{init}$
**for** $i := 1$ **to** $(\mu - n_{init})/n_b$ **do**
  $\hat{f} \leftarrow build\_surrogate\_model(S)$
  $q(P) \leftarrow \hat{f}(P)$
  $r_i(P) \leftarrow$ sort according to $q(P)$
  **if** $r_{i-1}(P)[1:pbest] = r_i(P)[1:pbest]$ **then**
    **break**
  **end**
  **else**
    $S \leftarrow S \cup \{(x_i, f(x_i))\}$, x = 1, ..., $n_b$
  **end**
**end**
**if** $i > 2$ **then**
  $n_{init} \leftarrow \min(n_{init} + n_b, \mu - n_b)$
**end**
**else if** $i < 2$ **then**
  $n_{init} \leftarrow \max(n_b, n_{init} - n_b)$
**end**

Algorithm 2: Ranking procedure $(P, S)$.

the new parent population is based on the fitness values obtained with the use of the same surrogate model for both old parents and new offspring. Thus the new parent population is completed.

# 4 EXPERIMENTAL STUDY

## 4.1 Experiment Conditions

**CEC2013 Benchmark Suite.** We tested the JADE supported by the aforementioned regression-based surrogate models using the CEC2013 benchmark suite (Liang et al., 2013) which defines 28 challenging optimization functions that have a wide spectrum of landscapes and difficulties. Thus it is particularly well-suited for assessing the effectiveness of surrogate models.

The CEC2013 benchmark suite fitness functions are challenging for regression models, particularly in terms of local estimation – many of them are highly nonlinear and non-convex, making it difficult for regression models to accurately capture the underlying relationships between input and output variables, particularly in local regions. Some of them present noisy landscapes, and some are locally asymmetrical about their local minima. In this study, we focus on problems with $D = 10$ dimensions.

**Parameter Settings.** To ensure statistically meaningful results, we executed 51 independent runs for each combination of JADE with a surrogate model and for each optimization problem from the benchmark suite. In each run, we recorded the best achieved true fitness value and treated it as a result of that run. In each independent run, the algorithm was given a budget of $100,000$ fitness evaluations, as it was suggested in the CEC2013 benchmark suite. The JADE population contained $\mu = 20$ individuals, and the search range was defined as $[-100, 100]^{10}$. The optimization process was terminated either after using the admissible budget or after reaching a point whose fitness differed no more than $10^{-8}$ from the global optimum which has been defined in the benchmark suite. Bound constraints imposed by the benchmark suite were handled with the reflection method, according to the guidelines from (Biedrzycki et al., 2018).

For the kNN regression model, the value of $k$ was set to $D+2$ after a preliminary tuning. In the case of SVR, RFR, and XGBoost, we used the default parameters' settings provided in the libraries that implement these methods (Chen and Guestrin, 2016), (Pedregosa et al., 2011).

## 4.2 Results

**Efficiency of Surrogate Models and the Archive Size.** The first series of experiments was aimed at selecting the appropriate archive size for each surrogate model. We analyzed statistics of the results obtained by surrogate model-assisted JADE for archive size ranging from $2 \cdot \mu$ up to $50 \cdot \mu$, where $\mu$ stands for the population size. The results of the experiments are provided in Fig. 2, 3 in the form of boxplots. The median value is indicated by a horizontal line within the box. The width of the box corresponds with the interquartile range, and the whiskers correspond to the distance between extreme values and the first/third quartiles. Outliers are represented with bullets.

Labels on axis X represent various surrogate model settings: <model>_<n> means that it is a combination of JADE with a specific surrogate model and the size of the archive of points (expressed as the population size multiplier <n>).

In the case of kNN, RFR, and XGBoost, the performance of the surrogate-assisted JADE usually grows with the size of the archive used for creating the surrogate model, but for some optimization problems, e.g. F15, F16, F23, it appears that an optimum value of the archive size can be observed. Moreover, the relationship between the archive size and the optimization efficiency is similar between the three aforementioned model types.

In the SVR case, quite often the quality of results is worse than the results of JADE alone. Moreover, the results get worse along with the model size. Clarification of this effect needs much deeper investigation, we hypothesize that perhaps the kernel formula and/or the kernel parameter are responsible for this behavior.

**Comparison Between Surrogate Model Efficiency.** For each surrogate model, we selected the best-performing archive size using the Wilcoxon test in pairs by comparing the results yielded by JADE with and without the surrogate model across 28 different CEC2013 functions. Then we aggregated the results by counting the number of wins, draws, and losses of a surrogate model-assisted JADE with JADE without the model. The winning population size for the surrogate model was equivalent to the value for each the difference between the number of wins and the number of losses was greatest. The archive size of 20 gave the best performance within each surrogate model.

Then we applied the same methodology to the comparison between the results yielded by the surrogate models with the optimal archive size. Table 1 summarizes the results. Each row in the table

Table 1: Results of the Wilcoxon test for each surrogate model and the standard JADE version for functions from CEC2013 benchmark.

| function | kNN | RFR | XGBoost | SVR |
|---|---|---|---|---|
| F1 | * | * | - | * |
| F2 | + | * | * | - |
| F3 | - | * | * | - |
| F4 | + | * | * | - |
| F5 | * | * | - | - |
| F6 | * | - | - | - |
| F7 | + | + | + | - |
| F8 | * | * | * | * |
| F9 | + | + | + | * |
| F10 | * | - | - | - |
| F11 | * | * | - | - |
| F12 | + | + | + | - |
| F13 | + | * | + | - |
| F14 | - | - | - | - |
| F15 | + | + | + | - |
| F16 | - | - | * | - |
| F17 | * | - | * | - |
| F18 | + | + | * | - |
| F19 | * | * | * | - |
| F20 | + | + | + | - |
| F21 | + | - | - | - |
| F22 | - | - | - | - |
| F23 | + | + | + | - |
| F24 | + | * | + | - |
| F25 | + | * | + | - |
| F26 | * | + | * | - |
| F27 | + | + | + | - |
| F28 | * | - | - | - |
| | 14/10/4 | 9/11/8 | 10/9/9 | 0/3/25 |

corresponds to a specific function, and the columns represent each surrogate model with its best archive size. We indicate the results with symbols "+", "–", and "*", when the surrogate-model assisted JADE was performing better than JADE alone, worse than JADE alone, or there was no statistically significant difference between the compared JADE versions. The last row of the table summarizes the number of wins/draws/losses of the surrogate model supported JADE versus JADE alone.

Among the considered surrogate models, a clear winner is kNN. JADE coupled with kNN yields results that are superior to JADE without a surrogate model in the case of 14 optimization problems from the benchmark suite, in 10 cases the results were comparable, and only in 4 cases, pure JADE was the winner. Notably, it never happened that some other model improved and kNN deteriorated the quality of JADE. SVR consistently underperforms compared to other methods, probably due to an improper choice of kernel or its parameters.

**Analysis of Convergence Curves.** The Wilcoxon test-based comparison relates to the best results after spending the admissible budget of fitness evaluations. Here we provide the convergence curves to compare the intermediate results obtained for several selected CEC2013 problems. The convergence curves (Fig. 1) illustrate the evolution of the best-performing point fitness in each of 51 independent runs, averaged over the independent runs. The results for surrogate model-assisted JADE are obtained for the best archive size.

In certain instances, JADE coupled with a surrogate model, clearly converges faster and yields better results compared to the standard JADE. This underlines the effectiveness of the JADE with a surrogate model in swiftly navigating the search space and reaching near-optimal solutions in a shorter time frame. A good example is problem 27, where most models show behavior like that, except the SVR model. However, there are also scenarios where the surrogate model assisted JADE initially converges faster but plateaus at a certain level. This could be indicative of getting stuck in local optima, which is a well-known issue in evolutionary algorithms. These observations provide insights into potential areas of refinement for our method, such as incorporating mechanisms to escape from local optima and to reach global ones. For problem 5 we can see this effect for SVR and XGBoost models. In addition, there are also situations where the standard JADE converges faster initially but is eventually overtaken by its surrogate model-assisted versions, which finally achieve better results. This suggests that while the standard JADE may be quicker to find promising regions in the search space for some of the functions, surrogate model assistance could improve the results in the longer run. For kNN and RFR, problem 20 is a good example of such behavior.

## 5 CONCLUSION

We show the potential of using surrogate models to improve the efficiency of the JADE algorithm. Among tested methods, kNN with an archive size equal to 20 achieved the best performance on the CEC2013 benchmark suite. Note that kNN is a relatively cheap model because it needs no training phase to build the internal model structure.

Our findings highlight the important role of surrogate model selection. While certain benchmark functions delivered comparable results regardless of the surrogate model used, we have shown that most functions reacted differently to varying models, leading to significant differences in the results. This points to the necessity of understanding the specific characteristics and requirements of each function when choosing the most suitable model.

Despite the promising outcomes, we recognize that there is always potential for improving our method. Future research will thus concentrate on enhancing the performance of analyzed models on the test functions where they currently lag.

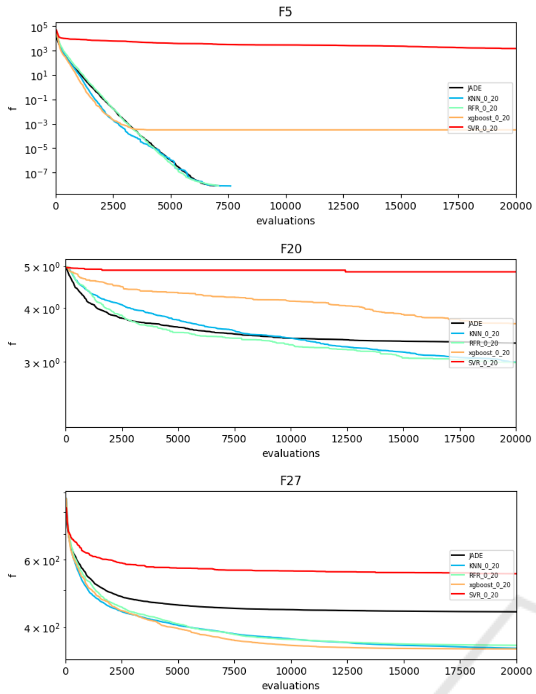


Figure 1: Convergence curves of problems 5, 20, and 27.

## REFERENCES

Biedrzycki, R., Arabas, J., and Jagodziński, D. (2018). Bound constraints handling in differential evolution: An experimental study. *Swarm and Evolutionary Computation*, 50.

Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. pages 785–794.

Cortes, C. and Vapnik, V. (1995). Support vector networks. *Machine Learning*, 20:273–297.

Drucker, H., Chris, Kaufman, B. L., Smola, A., and Vapnik, V. (1997). Support vector regression machines. In *Advances in Neural Information Processing Systems 9*, volume 9, pages 155–161.

Dudani, S. A. (1976). The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(4):325–327.

He, C., Zhang, Y., Gong, D., and Ji, X. (2023). A review of surrogate-assisted evolutionary algorithms for expensive optimization problems. *Expert Systems with Applications*, 217:119495.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2021). An introduction to statistical learning, with applications in r.

Liang, J., Qu, B., Suganthan, P., and Hernández-Díaz, A. (2013). Problem definitions and evaluation criteria for the cec 2013 special session on real-parameter optimization. *Technical Report 201212, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China*.

Ong, Y., Nair, P., Keane, A., and Wong, K. (2004). Surrogate-assisted evolutionary optimization frameworks for high-fidelity engineering design problems. *Studies in Fuzziness and Soft Computing Series*, 167.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Picheny, V., Wagner, T., and Ginsbourger, D. (2013). A benchmark of kriging-based infill criteria for noisy optimization. *Structural and Multidisciplinary Optimization*, 48.

S. Kern, N. H. and Koumoutsakos, P. (2006). Local meta-models for optimization using evolution strategies. In *Parallel Problem Solving from Nature - PPSN IX*. Springer Berlin Heidelberg.

X. Lu, K. T. and Yao, X. (2011). Classification-assisted differential evolution for computationally expensive problems. *IEEE Congress on Evolutionary Computation*.

Zaborski, M. and Mańdziuk, J. (2022). Improving lshade by means of a pre-screening mechanism. pages 884–892.

Zhang, C. and Gao, L. (2013). An effective improvement of JADE for real-parameter optimization. pages 58–63.

Zhang, J. and Sanderson, A. (2007). DE-AEC: A differential evolution algorithm based on adaptive evolution control. *IEEE Congress on Evolutionary Computation*.

Zhang, J. and Sanderson, A. (2009). JADE: Adaptive differential evolution with optional external archive. *IEEE Congress on Evolutionary Computation*.

Zhang, T., Li, F., Zhao, X., Qi, W., and Liu, T. (2022). A convolutional neural network-based surrogate model for multi-objective optimization evolutionary algorithm based on decomposition. *Swarm and Evolutionary Computation*, 72:101081.

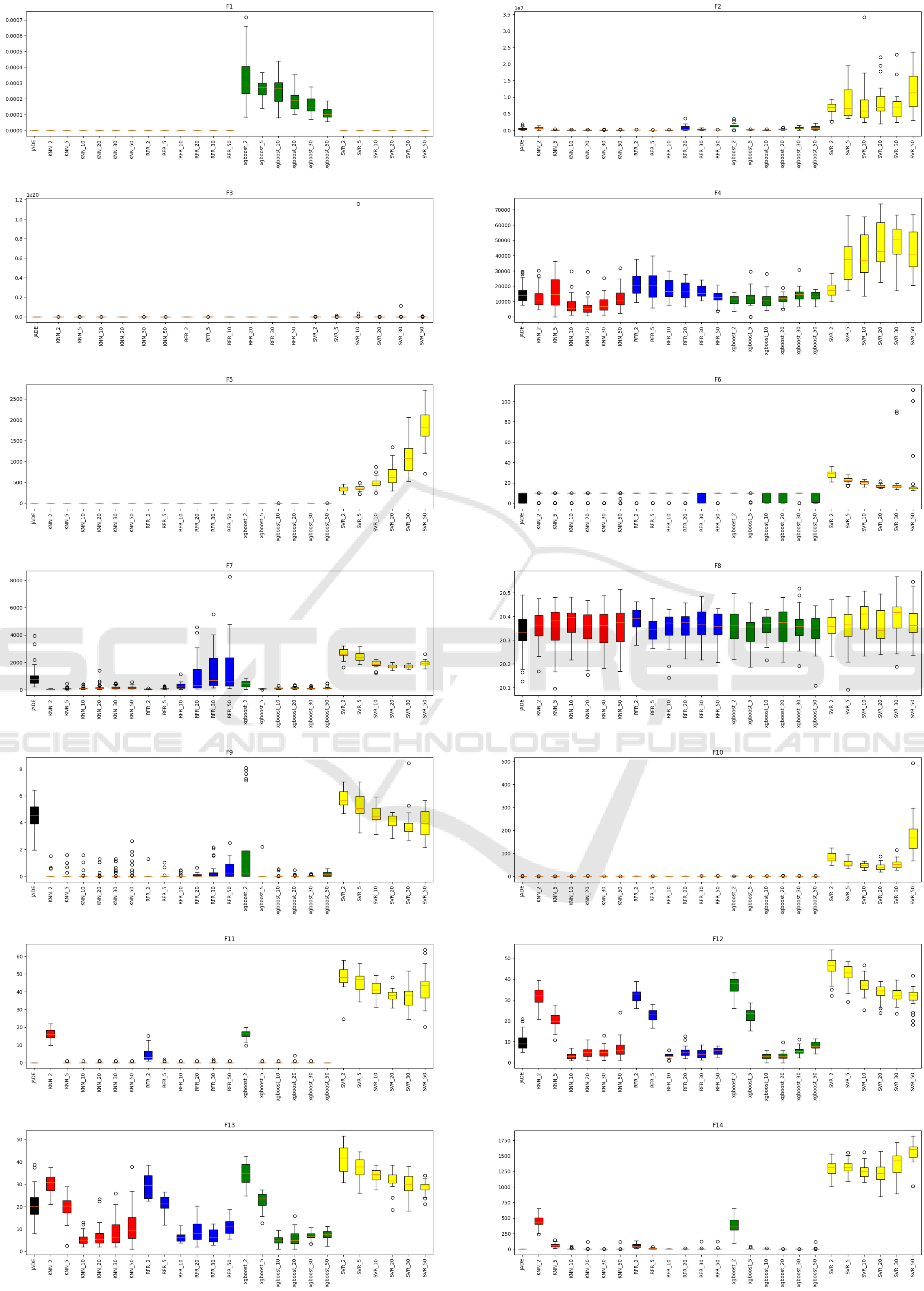


Figure 2: Boxplot results for functions 1 - 14 from CEC2013 Benchmark.

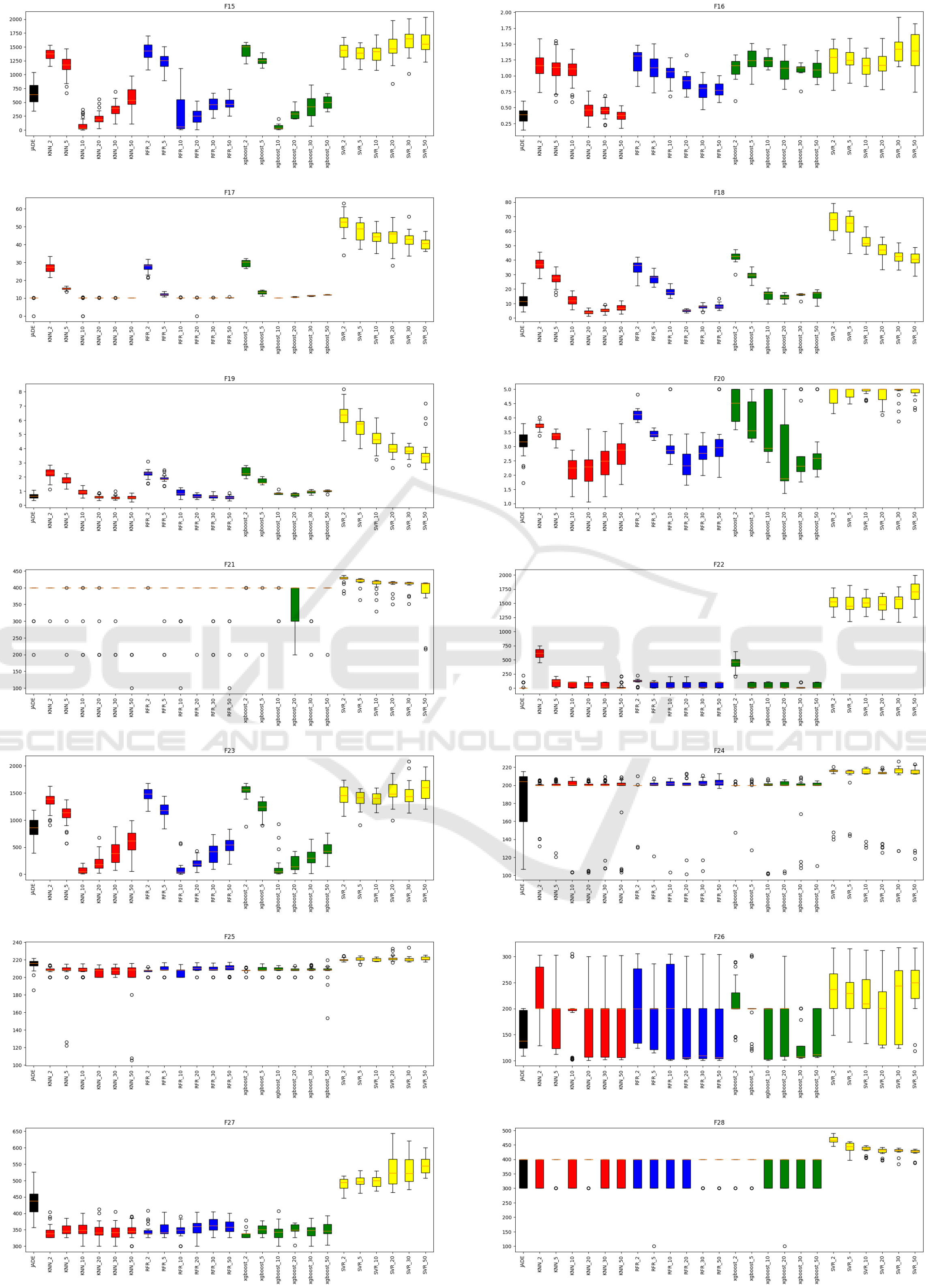


Figure 3: Boxplot results for functions 15 - 28 from CEC2013 Benchmark.