Optimization of multi-objective Clustering SWAY using KMeans, Boolean Domination and Decision Tree for Explanation

Amisha Bipin Waghela
Email: awaghel@ncsu.edu
Unity ID: awaghel

Sanket Tangade
Email: sstangad@ncsu.edu
Unity ID: sstangad

Tilak Satra
Email: trsatra@ncsu.edu
Unity ID: trsatra

***Abstract*—Search-based optimization techniques have become increasingly popular in Software Engineering (SE) to solve problems with multiple conflicting objectives. his paper focuses on improving the accuracy of the Sampling Way (SWAY) algorithm, which is a divide-and-conquer optimizer used in search-based software engineering (SBSE). Although SWAY is known for its speed and ability to act as a baseline model in comparison to other models, the accuracy of the model can be improved. Specifically, this paper proposes alternatives to the HALF and BETTER functions used in SWAY by implementing the K-Means algorithm and Binary Domination approaches. The experimental results demonstrate that using K-Means instead of HALF and Binary Domination instead of Zitzler predicate can improve the accuracy of SWAY in achieving multi-objective criteria. The proposed approaches are simpler, less computationally expensive, and more robust to the presence of outliers and noise in the data. Thus, this paper recommends using K-Means and Binary Domination as more sophisticated and nuanced approaches to clustering the data and selecting the candidates from the data that better represent the system, respectively.**

I. Introduction

Search-based software engineering or SBSE is a methodology used to solve problems where engineers need to provide solutions by evaluating the trade-off between various aspects of the system. The Sampling Way (SWAY) is a baseline optimizer that provides the base performance values which can be used by engineers to olve such problems. SWAY is a divide-and-conquer algorithm that can find promising data points among a large set of candidates using a very small number of model evaluations. The algorithm does not evaluate all of the random candidates. Instead, SWAY employs a top-down bi-cluster procedure that finds two distant points X, and Y, then labels all points according to which of X, and Y they are closest to using the "SPLIT / HALF" function. The points are then evaluated using a "BETTER" function, and all points near the worst one are culled. Hence, SWAY only evaluates the log(N) of N candidates, which makes it a relatively very fast algorithm. Since the number of required model evaluations is much less than that of common evolutionary algorithms, SWAY terminates very early. This reduction in runtime is an important feature of SWAY since some optimization studies can be very CPU intensive. Thus, being not so resource expensive process, SWAY serves as an ideal baseline method to explore optimization problems, particularly multiple-objective optimization problems in Software Engineering. SWAY assumes a close association between genotype and phenotype spaces in SE problems, which is supported by examples such as cloud environment configuration and POM3, XOMO, and software product line models.

SWAY possesses the ability to act as a baseline model in comparison to other models in terms of being less computationally expensive. However, just speed of evaluation cannot be the topmost criteria always. Additionally, we would also be concerned about the accuracy of the model under use. Moreover, SWAY depends on random samples based on which it performs separation and the best-rest match for those data points. How are those initial random samples selected plays a crucial role in the overall functioning and accuracy of the model. Hence, through this paper we focus on how we can increase the accuracy to make SWAY more than just a Baseline model.

Thus, based on the problems identified, we would propose alternatives to HALF and BETTER functions by implementing K-Means algorithm and Binary Domination approaches to tackle accuracy.

The SWAY algorithm is composed of two main components, HALF and BETTER. The HALF component splits the data into two parts, namely the best and the rest, while the BETTER component is used to determine the importance of data and discard the unimportant data. However, a more robust approach for partitioning the data is to use the K-Means algorithm instead of HALF. K-Means allows for a more nuanced and fine-grained understanding of the data, and can identify more subtle patterns in the data that may not have been immediately apparent with just two partitions. Our experimental results demonstrate that using K-Means instead of HALF led to more accurate results in achieving the multi-objective criteria. Therefore, we recommend using K-Means as a more sophisticated and nuanced approach to clustering the data, which can provide a more comprehensive understanding of the underlying patterns in the data. SWAY also uses Zitzler predicate to select the candidates from data that better represent the system. However, we propose to use Binary Domination. Firstly, binary domination is a simpler approach that is easier to understand and implement. It requires less computational resources than Zitzler, making it more suitable for large-scale optimization problems. Additionally, binary domination is more robust to the presence of outliers and noise in the data. Unlike Zitzler, which considers the entire objective space, binary domination only looks at the relative performance of solutions, making it less sensitive to fluctuations in the objective values. Moreover, binary domination is a more intuitive approach, as it directly compares solutions in terms of their objectives. This can help users gain a better understanding of the trade-offs involved in multi-objective optimization problems.

The project paper follows a structured approach divided into six main sections. The structure goes as starting with the ”Related Work” section which provides an overview of existing research in the area, related reseach findings as well as highlighting gaps that the project aims to address. Following, the ”Methods” section is divided into sub-sections that describe the algorithms, dataset, performance measures, and summarization methods used in the project. The ”Results” section presents the findings using visual aids such as tables and graphs. The ”Threats to Validity” section outlines potential limitations and suggests areas for future research. The ”Conclusion” section summarizes the key findings and provides recommendations, while the ”Future Work” section outlines potential avenues for future research and development. The paper's clear and logical structure makes it easy to follow the research and understand its implications.

II. Related Work

The task of multi-objective semi-supervised explanation is to provide explanations for a machine learning model's decisions while considering multiple objectives for the use case and making use of the provided data. There are several approaches to addressing this problem, including active learning, where the model selects the most informative samples and optimization-based methods that balance multiple objectives. Additionally, several techniques exist for generating explanations for machine learning models, such as decision trees, and rule-based methods.

Random projection is a technique used in machine learning and data analysis to reduce the dimensionality of high-dimensional data. The idea behind random projection is to project high-dimensional data onto a lower-dimensional space using a random matrix, while still preserving the structure of the data to some extent. The random projection technique works on the principle that, for many high-dimensional datasets, the structure of the data can be maintained even when the data is projected onto a lower-dimensional space. The key is to select a random matrix that satisfies certain properties, such as being orthogonal or having low coherence. The advantage of random projection is that it is computationally efficient since it does not require the computation of eigenvectors and eigenvalues, as is the case with other dimensionality reduction techniques such as principal component analysis (PCA). Random projection can be particularly useful for large datasets, where the computational cost of other techniques becomes prohibitive. However, it is important to note that random projection is not guaranteed to preserve all of the structure of the original data, and the degree to which the structure is preserved depends on the properties of the random matrix used. As a result, random projection is often used in combination with other techniques, such as clustering or classification algorithms, to analyze the reduced-dimensional data.

Semi-supervised learning is a type of machine learning that combines both labeled and unlabeled data to train a model. In contrast to supervised learning, where the algorithm is trained on labeled data only, and unsupervised learning, where the algorithm is trained on unlabeled data only, semi-supervised learning uses both labeled and unlabeled data to improve the accuracy of the model. The basic idea behind semi-supervised learning is to use the labeled data to learn the structure of the problem and the unlabeled data to improve the generalization of the model. The labeled data is used to train the model to identify the patterns and features in the data, while the unlabeled data is used to fine-tune the model and improve its performance. A Semi-Supervised algorithm assumes the following about the data:

Continuity Assumption: The algorithm assumes that the points which are closer to each other are more likely to have the same output label. Cluster Assumption: The data can be divided into discrete clusters and points in the same cluster are more likely to share an output label.

Manifold Assumption: The data lies approximately on a manifold of a much lower dimension than the input space. This assumption allows the use of distances and densities which are defined on a manifold. Semi-supervised learning is particularly useful in cases where labeling data can be expensive, time-consuming, or difficult. By using unlabeled data, the model can learn more about the underlying structure of the data, and better generalize to new, unseen examples. There are different techniques for semi-supervised learning, including self-training, co-training, and multi-view learning. Self-training involves training a model on labeled data and then using it to label unlabeled data, which is then added to the labeled dataset for re-training. Co-training involves training multiple models on different subsets of the data and then exchanging information between the models to improve performance. Multi-view learning involves training multiple models on different views of the data, such as different feature sets, and combining the results to improve accuracy. However, the major drawback involves hand-labeling the data which is a very costly process, especially when dealing with large volumes of data.

Heuristics are generally rule-of-thumb strategies that user experience and knowledge to make quick and often intuitive decisions. Heuristics work by leveraging our cognitive biases and pattern recognition abilities to simplify complex decision-making processes. They are often based on past experiences or known patterns in the data and are optimized to quickly arrive at a good solution without necessarily guaranteeing that it is the best or optimal solution. Heuristics are useful in scenarios where there is a large amount of data to process or where there is uncertainty or ambiguity in the problem. In these cases, heuristics can help to reduce the cognitive load on the decision-maker and provide a faster and more efficient way to arrive at a solution. Thus, a good heuristic is achieved by trading optimality, completeness, accuracy and precision for speed. Although the optimization models performed better at data fitting (adjusting their parameters to the data of the past 10 years) than the simple heuristic did, they performed worse at predicting the future. Thus, they usually overfit the past data which provides great deal of robustness to the heuristics. In contrast, the heuristics, which do not estimate any parameter, usually do not overfit. However, it is important to note that heuristics are not foolproof and can sometimes lead to errors or biases in decision-making. They may also be less accurate or optimal than more complex algorithms or methods, particularly in situations where there is a lot of noise or randomness in the data.

Several techniques have been proposed to address the challenging problem of optimizing multiple objectives in the presence of limited labeled data, also known as the multi-objective semi-supervised problem. These techniques include multi-objective optimization, active learning, and transfer learning. However, in many real-world scenarios, the speed of these solutions can be a critical factor. To address this issue, we propose a tweaked version of SWAY, which can find promising individuals among a large set of candidates using a very small number of model evaluations. As a baseline method, SWAY can terminate early, as the number of required model evaluations is much less than that of other methods in the literature. Our proposed approach can be particularly useful when model evaluation is computationally expensive, making SWAY a promising solution for the multi-objective semi-supervised problem in such cases.

III. Methods

In this study, we have attempted to optimize and find trade-offs for the implement the methods listed below. In the implementation, we run the sway method repeatedly for twenty iterations with different seed values, where sway clusters candidates in order to identify a superior cluster. It selects a small superior set of candidates among a group of candidates using the *BETTER* function. The algorithm splits candidates into two parts using the *HALF* function, compares representatives of each side and generates clusters, namely: BEST and REST, prunes REST based on the comparison, and recursively applies SWAY to each remaining part. In the result section, we have displayed aggregated values over all 20 runs.

A. Algorithms in SWAY

1) HALF vs K-Means Clustering: **K-Means** is a clustering algorithm that can be used to group similar data points together in a dataset. It is a popular unsupervised learning algorithm that partitions a dataset into K distinct clusters based on the similarity of the data points. The algorithm works by iteratively assigning data points to the nearest cluster center (centroid) and then recomputing the centroid of each cluster.

For our implementation, we optimize the HALF function responsible for splitting the dataset recursively into the best candidates and the rest, by using K-Means clustering to form these two clusters. We propose that it improves the performance of the algorithm in several ways:

- Improved clustering: KMeans can provide more accurate clustering than the HALF function, which only considers the distance between two reference points. KMeans takes into account the entire dataset and partitions it into clusters based on the similarity of the data points. This can help identify more distinct clusters and improve the accuracy of the clustering process.
- Robustness to outliers: KMeans is more robust to outliers than the HALF function. Outliers are data points that are significantly different from the majority of the dataset, and they can affect the clustering process. KMeans minimizes the sum of squared distances between data points and the centroid of their assigned cluster, which makes it less sensitive to outliers.
- Scalability: KMeans can handle larger datasets more efficiently than the HALF function. The HALF function requires calculating the distance between each data point and two farthest points, which can become computationally expensive as the dataset grows. KMeans, on the other hand, can handle larger datasets by using more efficient algorithms to assign data points to clusters.

2) Binary Domination vs Zitzler's Predicate: **Binary domination** is a concept that considers an individual better than another if it performs better on at least one goal and worse on none. Binary domination is more robust to the presence of outliers and noise in the data. Unlike Zitzler's predicate, which considers the entire objective space, binary domination only looks at the relative performance of solutions, making it less sensitive to fluctuations in the objective values. Moreover, binary domination is a more intuitive approach, as it directly compares solutions in terms of their objectives. This can help users gain a better understanding of the trade-offs involved in multi-objective optimization problems. In this paper, we use Boolean domination for the *better* method, which will be used to recursively split the dataset into two clusters: best and rest.

3) EXPLN vs Decision Tree: The EXPLN method is a data summarization technique that involves a multi-step process. Firstly, it creates several bins and generates rules for determining certain values for different columns of the dataset. Secondly, it employs the BETTER method, which uses heuristics to select the top best rows of the data based on the rules generated in the previous step. The BETTER method evaluates the quality of the generated rules by identifying the rows that best represent the underlying patterns in the data. Finally, the SWAY algorithm aggregates the values generated by the rules, and the resulting summary is evaluated.

Decision Tree Classifier: To optimize the explanation algorithm for the clustering model, we have utilized decision trees. Decision trees can provide insights into how the clustering algorithm is making its predictions. Decision trees are constructed by recursively splitting the data into smaller subsets based on the values of different features, resulting in a tree-like structure that represents the decision-making process of the algorithm. Specifically, decision trees can be used to identify the most important features that the clustering model is using to group the data points which is not the case for the current explain method.

B. Data

With the advent of large datasets, there has been a significant increase in the use of data structures for efficient storage and retrieval of data. However, when dealing with datasets containing multiple data types, such as numeric and symbolic values, designing an effective data structure becomes a challenging task. This report of the project works on the dataset with the data structure containing mainly two types of values, Num and Sym, and proposes an approach for organizing and manipulating the data.

The dataset under consideration comprises of various attributes such as Clndrs, Volume, HpX, Lbs-, Acc+, Model, origin, and Mpg+. The paper aims to analyze the properties of the dataset and suggest a data structure that can handle the unique characteristics of the dataset. The paper further investigates the effectiveness of the proposed data structure in terms of retrieval and manipulation of the data.

Our project involved working with a set of 11 datasets that contained a variety of data types, including numerical, symbolic, and textual data. The datasets were obtained from different sources and covered various domains, including healthcare, software engineering, and transportation. To provide a comprehensive analysis of these datasets, we employed data analysis techniques such as descriptive statistics In this paper, we will present an in-depth analysis of a couple of the datasets to illustrate the type of data they contain and the methods we used to analyze them. Using these datasets we were able to train the new variant of the SWAY algorithm. Below is description of a few datasets we worked on.

TABLE I
AUTO93.CSV

	Min	Mean	Median	Max	IQR
Clndrs	3	5.4	4	8	4
Volume	68	193.5	148.5	455	157.8
Lbs-	1613	2970.4	2803.5	5140	1384.25
Acc	8	15.6	15.5	24.8	3.34
origin	1	1.6	1	3	1
Mpg+	10	23.8	20	50	10

The table above displays summary statistics for a dataset related to car design, which includes variables such as Clndrs, Volume, Lbs-, Acc, origin, and Mpg+. The dataset provides information about car specifications such as the number of cylinders, engine volume, weight, acceleration, origin, and miles per gallon. The dataset has the numeric values for the attributes mentioned of different types of cars present and has around 400 entries. The table reports the minimum, mean, median, maximum, and interquartile range (IQR) of each variable, which can provide important insights into the distribution of the data and any potential outliers. We have used this data for the algorithm optimization as well as for testing purpose.

TABLE II
HEALTHCLOSEISSES-EASY.CSV

	Min	Mean	Median	Max	IQR
MRE-	0	92.3	119.3	140.7	19.7
ACC+	-17.2	-8.5	-12.24	0	11
PRED40+	0	17.8	0	83.3	0
N_estimators	10	106.1	1110	200	100
Min_sample_leaves	1	10.47	11	20	10
Max_depth	1	10.5	11	20	10

Similarly, the table shown above summarizes The ”healthCloseIsses-easy.csv” dataset aims to provide insights into issue close time by presenting different statistics on a set of variables. The table consists of six columns: MRE-, ACC+, PRED40+, N_estimators, Min_sample_leaves, Min_impurity_decrease, and Max_depth. The variables are grouped based on their relevance to the domain of issue close time. The MRE- (Mean Relative Error) column provides information on the average error in percentage between the predicted and actual close times. The ACC+ (Accuracy) column shows the accuracy percentage of the predictions, with negative values indicating under-prediction. PRED40+ (Precision at 40% indicates the percentage of predictions that were within 40% of the actual close time. The last three columns, N_estimators, Min_sample_leaves, and Max_depth, are parameters used in the random forest algorithm, which was used to make predictions in this dataset. The table shows the minimum, maximum, median, mean, and interquartile range (IQR) for each variable. The IQR shows the range between the 25th and 75th percentiles of the data. These insights could helped us to guide decisions related to optimizing the algorithm as well as selecting the model for the clustering part.

TABLE III
CHINA.CSV

	Min	Mean	Median	Max	IQR
IDX	1	250	250	499	249
Input	0	167.1	63	9404	125.5
Output	0	113.6	42	2455	99
Interface 0 d	24.3	0.0	1572	20	
Duration	1	8.7	7.0	84	7
N_effort-	31	4277.6	2098	54670	3416

Table mentioned above contains statistical information that was calculated from a dataset file ”china.csv” related to software project estimation. The dataset contains various columns including IDX, AFP, Input, Output, Enquiry, File, Interface, Added, Changed, Deleted, PDR_AFP, PDR_UFP, NPDR_AFP, NPDU_UFP, Resource, DevType, Duration, N_effort-, and EffortX. The table is grouped by each column, and displays the minimum, mean, median, maximum, and interquartile range (IQR) values for each column. The IDX column represents the project index, while the Input and

Output columns represent the number of inputs and outputs for each project respectively. The Interface 0 d column represents the number of files with no interface for each project. The Duration column represents the time duration of each project, and the N_effort- and EffortX columns represent the estimated and actual effort required for each project respectively. All the datasets used in the project were similar to the datasets explained above. These statistical results as well as the dataset details helped to finalize the hyperparameters, the model for the algorithm, what all changes can be done in the code and the methods to improvise the current SWAY structure and the model.

C. Performance Measures

The following plots display the distibution of values of each column belonging to the Auto2 dataset, when we modify the values of the hyperparameter 'min'. The hyperparameter 'min' regulates the size of the smaller cluster best, and thus, also regulates the number of evaluations done by sway. From the plots below, we infer the following:

- As the min value increases and the number of evaluation decreases, the average value provided by optimized sway2 grows greater than sway1 for objectives that we want to maximize
- As the min value decreases and the number of evaluation increases, the average value provided by optimized sway2 scales similarly to sway1 for objectives that we want to minimize

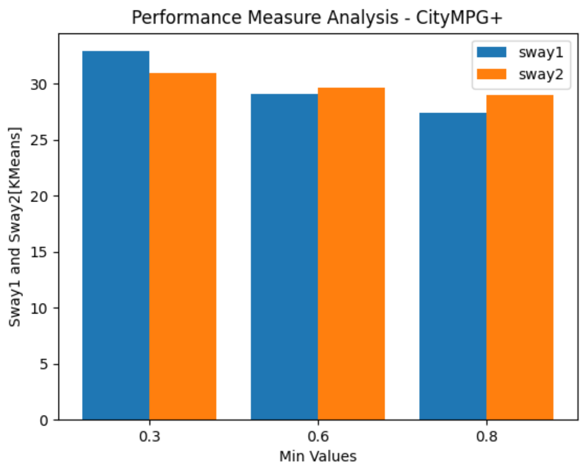

Fig. 1. CityMPG+ vs Sway values.

D. Summarization Methods

Effect size refers to the magnitude or strength of the relationship between two variables. Significance testing, on the other hand, is a statistical method used to determine whether the observed relationship between two variables is statistically significant or simply due to chance.
In order to assess the statistical significance of the data,

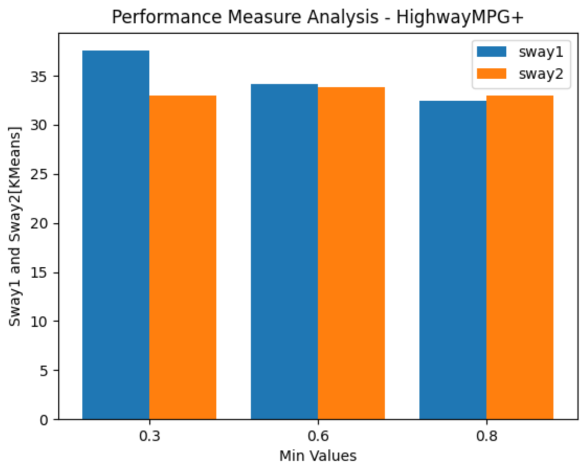

Fig. 2. HighwayMPG+ vs Sway values.

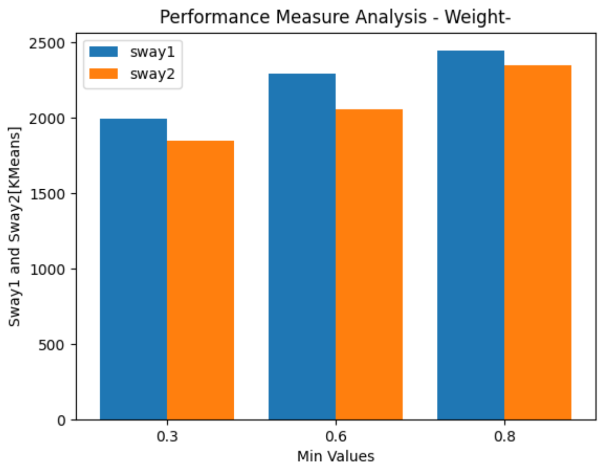

Fig. 3. Weight- vs Sway values.

we used Kruskal-Wallis and Mann-Whitney. These tests are non-parametric and designed to compare independent samples from two or more groups. The decision to use these tests was based on the fact that they do not require normal distribution within groups and are more robust in the presence of outliers and because these tests do not assume that the data follows a specific distribution.

1) Kruskal-Wallis: The Kruskal-Wallis test is a non-parametric statistical test used to compare three or more independent groups of samples. Specifically, the test compares the sum of the ranks for each group, with adjustments made for the different sample sizes in each group.

2) Mann-Whitney: The Mann-Whitney U test is a non-parametric test used to compare two independent groups of samples. The test ranks the data from both groups and then compares the ranks between the two groups to determine if there are significant differences between them.

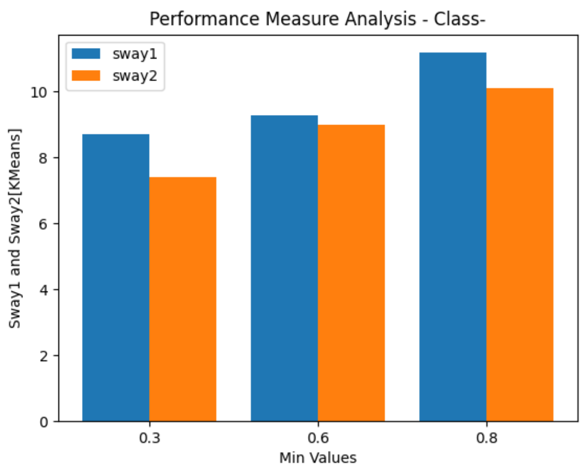

Fig. 4. Class- vs Sway values.

TABLE V
RESULT TABLE 2: AUTO2

	CityMPG+	HighwayMPG+	Weight-	Class-
all to all	=	=	=	=
all to sway1	≠	≠	≠	≠
all to sway2	≠	≠	≠	≠
sway1 to sway2	≠	≠	≠	≠
sway1 to xpln1	≠	≠	≠	≠
sway2 to xpln2	≠	≠	≠	≠
sway1 to top	≠	≠	≠	≠

TABLE VI
RESULT TABLE 3: AUTO93

	Lbs-	Acc+	Mpg+
all	2800.0	15.5	20.0
sway1	2092.7	16.255	32.5
sway2	2290.0	15.5	30.0
xpln1	2274.9	16.235	29.5
xpln2	2282.6	15.93	30.0
top	1985.0	18.78	40.0

IV. RESULTS

In this paper, by implementing the optimizations as described in Section 3, we have answered the following research questions and obtained the following results.

A. RQ1: Does the optimised SWAY2 provide a better result than SWAY1?

We have compared the performance of SWAY2 with that of SWAY1 and we have observed the following:

- As we can observe from Tables 4, 6 and 8 for datasets Auto2 and Auto96 and Nasa93dem, we can infer that the values for each column for sway2 are better than those of sway1.
- Further, by running the optimized SWAY on the various datasets in the scope of this project, we discovered similar patterns for most of the datasets reccorded here: https://github.com/amisha-w/CSC591-Project-Group-1/tree/main/etc/out.
- By running conjunction of an effect size test, we observe in Tables 5,7 and 9 that each algorithm is unequal to the other, except for 'all to all' is equal according to prudence test.

TABLE IV
RESULT TABLE 1: AUTO2

	CityMPG+	HighwayMPG+	Weight-	Class-
all	21.0	28.0	3040.0	17.7
sway1	29.55	34.65	2175.75	9.265
sway2(kmeans)	32.0	36.5	2045.0	9.0
xpln1	29.15	33.85	2319.0	9.61
xpln2	30.9	36.55	2091.25	9.06
top	33.0	39.4	2051.0	8.84

B. RQ2: Does the optimised EXPLN2 provide better results than EXPLN1?

We have compared the performance of EXPLN2 with that of EXPLN1 and we have observed the following:

- As we can observe from Tables 4, 6 and 8 for datasets Auto2 and Auto96 and Nasa93dem, we can infer that the values for each column for expln2 are slightly better than those of sway1.

C. Sampling & Explanation Tax

The **"sampling tax"** refers to the potential loss that can occur when only looking at a subset of data instead of the entire dataset. The speaker believes that this loss is inevitable and necessary, as taking a quick glance at data can still be helpful despite potentially missing some information.

The **"explanation tax"** refers to the loss of information between a complex concept (sway) and a simplified explanation of it (xplan). The speaker expects this loss to occur, as simplifying complex ideas often requires ignoring some details. This is the cost of generating simpler explanations, which may be more accessible but less precise than the original concept. But that's the price of taking heuristic peeks at things

V. THREATS TO VALIDITY

A. Dataset Bias

In this research paper, we have used a k-means clustering model for the sway algorithm and have trained and tested it on 11 specific datasets. However, it is important to note that the validity and generalizability of the results may not be the same for other types of datasets. While our approach has shown promising results on the datasets used in this study, it cannot be assumed that it will perform optimally on all types of datasets. Additionally, there are other clustering models available in the field of software engineering that may potentially provide even more optimized results. Therefore, future research should continue to explore and compare various clustering models to determine the most effective approach for different types of datasets and software engineering applications.

TABLE VII
RESULT TABLE 4: AUTO93

	Lbs-	Acc+	Mpg+	
all to all	=	=	=	
all to sway1	≠	≠	≠	
all to sway2	≠	≠	≠	≠
sway1 to sway2	≠	≠	≠	
sway1 to xpln1	≠	≠	≠	
sway2 to xpln2	≠	≠	≠	
sway1 to top	≠	≠	≠	

TABLE VIII
RESULT TABLE 5: NASA93DEM

	Kloc+	Effort-	Defects-	Months-
all	41.0	240.0	2004.0	21.3
sway1	19.31	89.94	814.15	14.505
sway2	3.5	18.0	172.0	9.1
xpln1	23.17	114.65	977.1	15.235
xpln2	8.11	47.7	337.2	11.99
top	3.5	10.8	109.0	7.8

B. Internal Bias

We should also consider the possibility of internal bias introduced by the algorithm itself. As SWAY uses random samples to perform data separation and selection, the initial set of random samples may impact the overall accuracy of the results. For the optimized SWAY, we expect that the results obtained from different runs may vary, leading to potential inaccuracies. We propose that this issue is mitigated by running the SWAY algorithm over 20 iterations, with each run having a different random seed. By aggregating the best row column values from each run, we aim to obtain a more robust and accurate set of candidate solutions. This approach will help reduce the impact of internal bias and provide us with a more reliable outcome for further analysis and decision-making.

TABLE IX
RESULT TABLE 6: NASA93DEM

	Kloc+	Effort-	Defects-	Months-
all to all	=	=	=	=
all to sway1	≠	≠	≠	≠
all to sway2	≠	≠	≠	≠
sway1 to sway2	≠	≠	≠	≠
sway1 to xpln1	≠	≠	≠	≠
sway2 to xpln2	≠	≠	≠	≠
sway1 to top	≠	≠	≠	≠

TABLE X
SAMPLING TAX AND EXPLAINATION TAX: AUTO2

	CityMPG+	HighwayMPG+	Weight-	Class-
samp tax1	-8.55	-6.65	864.25	8.43
samp tax2	-11	-8.5	995	8.7
xpln tax1	0.4	0.8	-143.25	-0.34
xpln tax2	1.1	0.05	-46.25	-0.06

VI. CONCLUSION & FUTURE WORK

A. Conlcusion

In this research paper, we have presented various optimizations for the SWAY algorithm to achieve solutions with a lower cost for multiple objective problems. Our aim was to obtain accurate and informative results, and therefore, we explored different approaches and areas to enhance the traditional SWAY method. We have proposed the use of the KMeans algorithm and Boolean Domination to replace the HALF and BETTER functions, which are inherent in SWAY. Additionally, we introduced the Decision Tree Classifier to optimize the EXPLN method. Our experiments showed that these optimizations led to improved accuracy, and we observed that SWAY2 and EXPLN2 outperformed SWAY1 and EXPLN1 for most of the datasets studied. The results demonstrate the potential of our proposed optimizations to increase the accuracy or maximize the multi-objective value of the SWAY algorithm.

B. Future Work

In the project, we have focused on optimizing the SWAY algorithm, which currently consists of three methods. The first method is called HALF, which divides the data into two clusters based on a given criterion. However, we found that using a k-means clustering algorithm instead of the HALF method resulted in better clustering performance. Therefore, in future work, we could explore other clustering algorithms that might improve the performance of the SWAY algorithm even further. The second method in the SWAY algorithm is called BETTER, which uses Zitzler's principle to identify the better solution between two given solutions. However, there might be other principles or methods that could be used to optimize the complete SWAY algorithm. Therefore, we could investigate other principles or methods that might improve the performance of the SWAY algorithm, particularly in cases where the Zitzler's principle does not yield satisfactory results. Finally, we could also modify the hyperparameters of the SWAY algorithm to further optimize its performance. For example, we could explore different values for the number of clusters or the stopping criterion for the algorithm. By tuning these hyperparameters, we might be able to achieve better clustering performance or more accurate identification of the better solution. Overall, future work in this area could focus on exploring different clustering algorithms, principles or methods, and hyperparameter tuning to further optimize the SWAY algorithm for different types of data and applications.

REFERENCES

[1] Chen, J., Nair, V., Krishna, R. and Menzies, T. (2018). *"Sampling" as a Baseline Optimizer for Search-based Software Engineering* IEEE Transactions on Software Engineering, 44(11), 1048-1061. doi: 10.1109/TSE.2017.2774259

[2] J. Krall, T. Menzies, and M. Davies. Gale: Geometric active learning for search-based software engineering. IEEE Transactions on Software Engineering, 41(10):1001–1018, Oct 2015.

[3] Nikolaos Mittas and Lefteris Angelis. Ranking and Clustering Software Cost Estimation Models through a Multiple Comparisons Algorithm. IEEE Transactions of Software Engineering, 2013.

[4] Mark Harman, Stephen Swift, and Kiarash Mahdavi. An empirical study of the robustness of two module clustering fitness functions. In Proceedings of the 7th annual conference on Genetic and evolutionary computation, pages 1029–1036. ACM, 2005.

[5] R C Holte. Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. Machine Learning, 11:63, 1993.

[6] David Owen, Bojan Cukic, and Tim Menzies. An alternative to model checking: Verification by random search of and-or graphs representing finite-state models. In High Assurance Systems Engineering, 2002. Proceedings. 7th IEEE International Symposium on, pages 119–126. IEEE, 2002.

[7] Varsha Veerappa and Emmanuel Letier. Understanding clusters of optimal solutions in multi-objective decision problems. In Proceedings of the 2011 IEEE 19th International Requirements Engineering Conference, RE 2011, pages 89–98, 2011.

[8] Andrea Arcuri and Gordon Fraser. Parameter tuning or default values? an empirical investigation in search-based software engineering. Empirical Software Engineering, 18(3):594–623, 2013.

[9] Marcela Zuluaga, Andreas Krause, Guillaume Sergent, and Markus Püschel. Active learning for multi-objective optimization. In International Conference on Machine Learning (ICML), 2013.

[10] Ali Ouni, Raula Gaikovina Kula, Marouane Kessentini, Takashi Ishio, Daniel M German, and Katsuro Inoue. Search-based software library recommendation using multi-objective optimization. Information and Software Technology, 83:55–75, 2017.