**Chapter 4: Validation**

* 1. **Introduction**

To evaluate our work for adapting systems’ interfaces using MOEA/D, we conducted a set of experiments based on one large system (Give a name to the system). Each experiment is repeated 31 times, and the obtained results are subsequently statistically analyzed with the aim to measure the performance of our MOEA/D proposal with various given criteria. In this section, we first present our research questions and then describe and discuss the obtained results.

* 1. **Research Questions**

We assess the performance of our approach by finding out whether it could generate meaningful sequences of rules that improve the adaptivity of interfaces while reducing the number of rules needed. Our validation is conducted by addressing the following research questions outlined below. We also explain how our experiments are designed to address these questions:

**RQ1**: To what extent can the MOEA/D help in improving the adaptivity of interfaces in the system?

**RQ2**: To what extent can the use of multiple metrics improve the adaptivity of interfaces in the system?

**RQ3**: To what extent can the proposed approach minimize the number of needed rules?

**RQ4**: How does the proposed MOEA/D perform compared to a mono-objective approach?

To answer RQ1, we (…).

To answer RQ2, we (…).

To answer RQ3, we (…).

To answer RQ4, we (…).

* 1. **Studied Project**

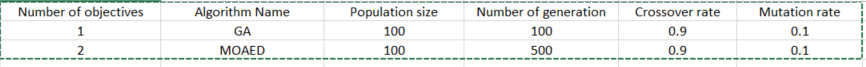
(…)

* 1. **Experimental Setting**

Parameter setting influences significantly the performance of a search algorithms on a given search problem. It is usually difficult to preemptively set the best tuning setting. For this reason, we perform a set of experiments using several population sizes: 10, 20, 40, 80, 160, and 320 for our 2 objectives. The maximum number of generations used is 100, 200, 400, 8000 and 1600. For each algorithm, to generate an initial population, we start by defining the maximum vector length (maximum number of rules per solution). As a higher number of operations in a solution do not necessarily mean that the results will be better, we empirically determine the best set of setting through the above mentioned trials. Ideally, a small number of rules should be sufficient to provide a good trade-off between the fitness functions. This parameter can be also specified by the user or derived randomly from the sizes of the program and the used operations list. During the creation, the solutions have random sizes inside the allowed range. We use the trial and error method [reference needed] in order to obtain a good parameter configuration. Since we are comparing different search algorithms, we classify parameters into common parameters and specific parameters. Table X depicts the important common parameters. For MOEA/D, the neighborhood size is set to 20

Statistical Tests

Since metaheuristic algorithms are stochastic optimizers, they can provide different results for the same problem instance from one run to another. For this reason, our experimental study is performed based on 31 independent simulation runs for each problem instance and the obtained results are statistically analyzed by using the Wilcoxon rank sum test [18][19] with a 99% confidence level (α = 1%). The latter verifies the null hypothesis H0 that the obtained results of two algorithms are samples from continuous distributions with equal medians, against the alternative that they are not H1. The p-value of the Wilcoxon test corresponds to the probability of rejecting the null hypothesis H0 while it is true (type I error). A p-value that is less than or equal to α (≤ 0.01) means that we accept H1 and we reject H0. However, a p-value that is strictly greater than α (> 0.01) means the opposite. In fact, for each problem instance, we compute the p-value obtained by comparing NSGA-II, IBEA, MOEA/D and mono-objective search results with NSGA-III ones. In this way, we determine whether the performance difference between NSGA-III and one of the other approaches is statistically significant or just a random result.



* 1. **Conclusion**

In this chapter, we present our two contributions of our research which are the proposed metrics of adaptive user interface evaluation and the meta-heuristic used to generate a set of evaluation rules that evaluate a several adaptive interfaces.

In the next chapter, we will discuss the experimentation of our evolutionary algorithm. Moreover, we will present a comparison study of our used multi-objective evolutionary algorithm with another mono-objective evolutionary algorithm.