On the Role of Diversity Measures for Multi-objective Test Case Selection

Andrea De Lucia1, Massimiliano Di Penta2, Rocco Oliveto3, Annibale Panichella1
1University of Salerno, via Ponte don Melillo, Fisciano (SA), 84084 Salerno, Italy
2University of Sannio, Palazzo ex Poste, Via Traiano, 82100 Benevento, Italy
3University of Molise, Contrada Fonte Lappone, 86090 Pesche (IS), Italy
Emails: adelucia@unisa.it, dipenta@unisannio.it, rocco.oliveto@unimol.it, apanichella@unisa.it

Abstract—Test case selection has been recently formulated as multi-objective optimization problem trying to satisfy conflicting goals, such as code coverage and computational cost. This paper introduces the concept of asymmetric distance preserving, useful to improve the diversity of non-dominated solutions produced by multi-objective Pareto efficient genetic algorithms, and proposes two techniques to achieve this objective. Results of an empirical study conducted over four programs from the SIR benchmark show how the proposed technique (i) obtains non-dominated solutions having a higher diversity than the previously proposed multi-objective Pareto genetic algorithms; and (ii) improves the convergence speed of the genetic algorithms.

Keywords—Search-based Software Testing; Test Case Selection; Niched Genetic Algorithms; Empirical Studies.

I. INTRODUCTION

Regression testing is the process to validate modified software for detecting whether new errors have been introduced into unchanged parts of software and to guarantee that the changed parts behave as intended. A complete re-testing of the changed system might be too expensive, especially if such a system is very large. Therefore it is crucial to perform (i) test case selection, i.e., to determine a subset of test cases that are able to satisfy a given testing adequacy criterion, and (ii) test case prioritization, i.e., to rank test cases with the purpose of first executing those having the highest likelihood of revealing faults.

During past and recent years, several techniques for test case selection and prioritization have been proposed [1], [2], [3], [4], [5], [6], [7]. Regression testing should pursue two contrasting goals: (i) re-test the unchanged parts of the software system on the basis of the test requirements denoted by a test criterion; and (ii) reduce the regression testing cost, i.e., the number of test cases to execute. Most of the existing approaches (see e.g., [1] [5]) have been developed by considering one single-objective only (e.g., test suite minimization), while fixing a constraint on the other objectives (e.g., test adequacy).

Recently, Yoo and Harman [8] treated the problem of test case selection as a Pareto-efficient multi-objective optimization problem, considering cost and coverage as two, conflicting objectives. Specifically, they applied two multi-objective search-based optimization techniques, the Non-Dominating Sorting Genetic Algorithm (NSGA-II) [9] and an island GA variant of NSGA-II, named vNSGA-II. An empirical study indicated that, in some cases, the search-based multi-objective approach was able to outperform the previously proposed greedy approach [4], and that greedy and multi-objective approaches can be combined to achieve better solutions.

This paper—building upon the work of Yoo and Harman—aims at enhancing the vNSGA-II algorithm by increasing population diversity in the obtained Pareto fronts. Indeed, when solving a multi-objective problem using GA, there is the risk that solutions are biased towards the solutions of the sub-problems, and, in the specific case of multi-objective GA, towards the creation of a limited number of groups of solutions (niches). This would likely compromise the quality (in terms of high code coverage and low testing cost) of the produced solutions. To mitigate such a problem, we propose two approaches, aimed at ensuring population diversity in vNSGA-II. The first approach is based on fitness sharing, which aims at penalizing solutions in crowded areas [10], while the second approach aims at partitioning the Pareto front and applying a density function to ensure a uniform distribution of solutions over the various partitions. The standard fitness sharing is customized for test case selection problem encouraging the diversity for only one of the objective functions (i.e., only for coverage but not for cost) while the density function on coverage space is an alternative asymmetric distance preserving mechanism introduced for the first time in this paper for reaching the same goal in a different way.

The benefits provided by the two proposed approaches for solution diversity have been evaluated on four programs from the Siemens benchmark1, namely printtokens, printtokens2, schedule and schedule2. Specifically, we have compared the vNSGA-II algorithm proposed by Yoo and Harman with the fitness sharing vNSGA-II and the density-based vNSGA-II. Results indicate that the two variants of vNSGA-II proposed in this paper outperform the original version of vNSGA-II in terms of convergence speed and optimality of the achieved solutions.

1Available at http://esquared.unl.edu/sir/.
This paper is organized as follows. Section II describes the related work, while Section III provides backgrounds on multi-objective test case selection. Section IV describes the two variants of vNSGA-II we propose to ensure population diversity in Pareto fronts. Section V reports and discusses the evaluation of the proposed approaches, while Section VI concludes the paper and outlines directions for future work.

II. RELATED WORK

Previous work aiming at controlling the size of test suite can be categorized in three main research areas: test suite minimization (or reduction), test case selection, and test case prioritization. A complete survey on these works can be found in the paper by Yoo and Harman [11]. In this section, we focus on approaches for test suite minimization and test case selection. Both test test suite minimization and test case selection consist of selecting a subset of tests from a test suite that satisfy all the test requirements. However, while test case selection chooses a subset of test cases based on the changes made to a software system, test suite minimization reduces the test suite based on external criterion such as structural coverage [11].

Harrold et al. [1] formulated test suite minimization as a minimal hitting set problem. Even if test suite minimization involves multiple test criteria, the authors treated such a multi-objective problem as a series of single-objective problems. They solved the problem by optimizing one objective first, and then using the obtained solutions as starting point to optimize the other objective. While this sequential approach is one of the classical techniques to solve multi-objective optimization problems, it may produce less optimal results compared to Pareto-efficient methods [8], [12].

Leung et al. [13] showed that the test case minimization problem is less expensive than the gain obtained (in terms of testing cost) by reducing the test suite. Moreover, test suite minimization problem may potentially reduce the fault detection ability of a test suite. Some studies have highlighted how the fault-detection ability of the reduced test suite was worsened [14], while others have shown that the fault-detection ability of the reduced test suite was preserved [15].

Several test case selection techniques have been proposed in literature, such as symbolic execution based approaches [7], dependence graph based techniques [2], and flow graph-based approaches [5]. In these single-objective formulations of test case selection, greedy algorithms have been used to maximize coverage. A greedy approach starts with an empty subset of the test suite and iteratively adds a test case provides higher coverage than the remaining test cases. Other works considered multiple criteria to decide whether to select a test case reducing the multi-objective problem to a single-objective one by using an aggregate function that condense more objectives. For example, a two-objective formulation of the problem (based on both coverage and cost) can be resolved with a greedy algorithm considering the coverage per unit time [4].

Recently, the test case selection [8], test case prioritization [16], [17] and test suite minimization [12] problems have been defined as multi-objective optimization problems without aggregate approaches. The authors of these papers showed how all these problems can be optimized using a Pareto efficient approach. As for the test case selection, Yoo and Harman presented a two-objective formulation in which code coverage and execution cost are considered when selecting test cases [8]. Three algorithms for multi-objective optimization are used: (i) NSGA-II (a multi-objective genetic algorithm) (ii) vNSGA-II (a variant of NSGA-II), and (iii) a greedy algorithm that uses a heuristic to provide a set of optimal solutions. The results of an empirical study highlighted the utility of multi-objective genetic algorithms (GAs) as alternative to the greedy algorithm.

As described in the introduction, our work is motivated by the work of Yoo and Harman i.e., we propose a modified version of vNSGA-II with the goal of increasing convergence speed and quality of the achieved solutions.

III. MULTI-OBJECTIVE TEST CASE SELECTION

The multi-objective test case selection problem can be defined as selecting a Pareto efficient subset of test cases on the basis of multiple test criteria. Formally, let \( T = \{t_1, \ldots, t_n\} \) and \( F = \{f_1, \ldots, f_m\} \) be a test suite and a set of objective functions, respectively. The multi-objective test case selection problem consists of finding a subset \( T' \subseteq T \) such that \( T' \) is the Pareto optimal set (i.e., set of non-dominated solutions\(^2\)) with respect to the objective functions in \( F \).

In this paper, we consider two objective functions (as also done by Yoo and Harman [8]): \textit{code coverage}, that measures test adequacy, and \textit{execution time}, that measures the test cost. Thus, we are interested in finding a set of non-dominated solutions that maximize coverage while minimizing the cost of the selected test cases.

The analytical identification of the best \( T' \) might be time consuming when the number of test cases composing the test suite becomes high. GAs [10] can be efficiently applied to deal with multi-objective problems that could not be solved analytically due to their complex nature. GAs tackle the problem by means of its intrinsic parallelism, i.e., having multiple solutions (individuals) evolving in parallel to explore different parts of the search space. GA is a stochastic search technique based on the mechanism of natural selection and natural genetics [10]. The search starts with a random population where each individual represents a solution to the global optimization problem. The population evolves through subsequent generations and, during each

\(^2\)A solution is called non-dominated, Pareto optimal, or Pareto efficient if it cannot be eliminated from consideration by replacing it with another solution which improves an objective without worsening another one.
generation, the individuals are evaluated using the fitness functions to be optimized. To create the next generation, new individuals are generated by either (i) recombining two individuals from the current generation using the crossover operator; or (ii) modifying individuals using the mutation operator. A new generation is made by selecting—according to the fitness values—some parents and offspring in order to keep the population size constant. Best individuals have higher probability to survive in the next generation.

While crossover, mutation, and selection operators are usually provided by frameworks supporting the definition of GAs and standard operators can be employed, the solution representation and the fitness function have to be carefully designed in order to implement an efficient multi-objective GA for test case selection. In this paper, we used the solution representation and fitness function definition that were successfully used in previous works to deal with the multi-objective test case selection and test suite minimization problems [8], [12].

Let \( m \) be the number of test goals to be covered (e.g., code statements), and \( n \) the number of test cases in the test suite. We can define a binary matrix \( A \) containing the trace data that captures the test goals achieved by each test case. A generic entry \( a_{i,j} \) of \( A \) is equal to 1 if the \( i^{th} \) test goal was covered by the \( j^{th} \) test case, 0 otherwise [8], [12].

Each individual of the population (i.e., solution) can be represented as a binary vector \( X \) of length \( n \), where the generic element \( x_{i} \) is 1 if test case \( t_{i} \) is selected by the individual \( X \), 0 otherwise. Then, we can evaluate the cost and the coverage (cov) achieved by the solution represented by \( X \) as follows [8], [12]:

\[
cost(X) = \sum_{i=1}^{n} x_{i} \cdot cost(t_{i})
\]

\[
cov(X) = \frac{1}{m} \sum_{i=1}^{m} \phi_{i} \quad \text{where} \quad \phi_{i} = \begin{cases} 1 & \text{if } g_{i} > 0 \\ 0 & \text{otherwise} \end{cases}
\]

where \( g_{i} \in G \) denotes the number of times the \( i^{th} \) test goal was achieved by the test cases selected by the individual \( X \). The vector \( G \) can be computed as follows:

\[
G = A \cdot X = \begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{bmatrix} \cdot \begin{bmatrix} x_{1} \\ \vdots \\ x_{n} \end{bmatrix}
\]

IV. DIVERSITY MEASURES AND TEST CASE SELECTION

In general, the goals of a multi-objective optimization are (i) to guide the search towards the Pareto-optimal front and (ii) to maintain population diversity in trade-off fronts [18]. However, when solving a multi-objective problem using GAs, it is generally difficult to maintain the diversity of the produced solutions. This is because the overall solutions are biased towards solutions of sub-problems. Indeed, a multi-objective GA tends to create a limited number of groups of solutions (niches), that are close to each other in the solution space, leaving the rest of the Pareto frontier empty and sparsely populated. This phenomenon is known as genetic drift (or population drift) [9]. Population that are scarcely diversified within the solution space can cause a premature convergence of the GAs towards some sub-optimal solutions. Thus, the lower the population diversity, the higher the likelihood of a GA of being trapped in some local optimum. Better diversified Pareto fronts will contain a wider range of solutions that, when combined through the crossover operator, are more likely to lead towards a better achievement of the optimization objectives, in our case low testing cost and high code coverage.

Diversity preserving techniques, such as crowding distance, fitness sharing, or tournament selection, have been proposed to mitigate the genetic drift [9]. Also the vNSGA-II algorithm proposed by Yoo and Harman [8] uses a diversity preserving technique. In particular, the authors apply the crowding schema on the phenotype space\(^3\), thus discriminating individuals in terms of both objective functions. This means that vNSGA-II tries to achieve a Pareto frontier by selecting individuals that are distant from each other in terms of both cost and coverage. However, obtaining solutions that are not too close each other in terms of cost is not a goal of the test case selection problem. Indeed, from a theoretical point of view, we are interested in finding a set of non-dominated solutions with minimum costs for different coverage values. Moreover, adding the constraint on cost could be self-defeating because a good set of non-dominated solutions should contain solutions with costs tending to zero, therefore, with solutions not so far in terms of cost.

Starting from the above observation, in this paper we define two variants of the vNSGA-II algorithm aiming at encouraging the population diversity in a reduced phenotype space, i.e., the space of coverage only. In particular, we replace the crowding schema of vNSGA-II with a variant of multi-objective niche technique [9], and a novel diversity-preserving technique based on the concept of entropy.

A. vNSGA-II with Fitness Sharing

Niche techniques (also called niche schemes or niche-formation methods) are methods for ensuring that a population does not converge to a niche, i.e., a limited number of Pareto points [9]. Intuitively, such techniques spur the development of multiple niches while limiting the growth of any single niche.

A common niche technique is fitness sharing [10]. The basic idea is to penalize the fitness of individuals in crowded areas, thus drastically reducing the probability of their survival to the next generations. Formally, let \( f_{k}(X_{i}) \) be the

\(^3\)The problem functions space is usually named phenotype space while the search space of the genetic operators is called the genotype space.
value of the $k^{th}$ objective function for the individual $X_i$. The fitness sharing ($fs$) value for $X_i$ on the $k^{th}$ objective function is defined as follows:

$$fitnessSharing_k(X_i) = \frac{f_k(X_i)}{\sum_{X_j \in P} sh(X_i, X_j)}$$

where $sh(X_i, X_j)$ is a sharing function that degrades the objective function value $f_k(X_i)$ on the basis of the presence of nearby individuals in the current population $P$. A widely used function is the triangular sharing function:

$$sh(X_i, X_j) = \begin{cases} 
1 - \frac{d(X_i, X_j)}{\sigma_{sh}} & \text{if } d(X_i, X_j) < \sigma_{sh} \\
0 & \text{otherwise}
\end{cases}$$

where $d(X_i, X_j)$ is the genotype or phenotype distance between the individuals $X_i$ and $X_j$. The parameter $\sigma_{sh}$ is called niched radius and is usually fixed, at the start of a GA, to a value estimated to be the minimum desired distance between niches.

For our purposes, we enforce the fitness sharing mechanism only on coverage objective function in order to encourage an asymmetric diversity of population and to obtain a well distributed Pareto frontier. This is because, as explained in the previous section, the goal of the test case selection problem is to find a set of non-dominated solutions with minimum costs for different coverage values. Thus, the distance between each pair of individuals $X_i$ and $X_j$ is:

$$d(X_i, X_j) = (cov(X_i) - cov(X_j))^2$$

B. vNSGA-II with Density Function

The multi-objective test case selection problem has an important peculiarity as compared to traditional multi-objective optimization problems. While in the latter problems the ranges of the objective functions for the global optimal solution are often unknown a priori, in the test case selection problem it is known that the Pareto frontier solutions range from 0% to 100% in the coverage space. It is worth noting that the upper bound of the coverage range might be lower than 100% because of unreachable branches. In this case, the coverage levels of all test cases can be used for computing the actual coverage range. Thus, a simple way to spur the next generations toward well distributed solutions in the coverage space consists of partitioning the coverage space and measuring how scattered the solutions are across the identified partitions (sub-intervals of the coverage range).

Ideally, a well-distributed Pareto frontier in terms of coverage has a uniform number of solutions for each partition. Thus, the basic idea is to encourage individuals with coverage values contained in partitions sparsely populated and to penalize individuals with coverage values contained in partitions densely populated. A simple way to achieve such a goal is to add a dummy fitness function to the test case selection problem:

$$density(X_i) = \begin{cases} 
0 & \text{if } |S_k| < \sigma_d, \ X_i \in S_k \\
|S_k| & \text{otherwise}
\end{cases}$$

where $X_i$ is an individual of the population, $S_k$ denotes the set of solutions that are in the $k^{th}$ partition (or sub-interval) of the coverage, and $\sigma_d$ (partition size) denotes the maximum desired number of individuals that each partition should contain. The partition size can be estimated as follows:

$$\sigma_d = \frac{\text{populationSize}}{K}$$

where $K$ represents the number of partitions which the coverage space should be divided into.

Having a new fitness function (density) the original two-objective formulation of the test case selection problem can be transformed into a three-objective formulation aiming at minimizing the cost, maximizing the coverage, and minimize the density. It is worth noting that the vNSGA-II can be still applied to solve such a three-objective problem.

V. Empirical Evaluation

This section describes the experiment carried out to evaluated the effects provided by the proposed diversity preserving techniques to solve the two-objective formulation of the test case selection problem.

The context of our study consists of four real world programs of the Siemens suite, namely printtokens, printtokens2, schedule and schedule2, available from the Software-artifact Infrastructure Repository (SIR) [19]. Table I reports the characteristics of the four programs. Since SIR provides a large number of test suites but with a limited number of test cases (less than 20 on average), in the context of our study we perform test selection on all the available test cases. In this way we stress as much as possible the considered GAs simulating large-scale testing environments. When using all the available test suites, the number of test cases to be considered for selection increases from few tens to several thousands (see Table I). As done in previous work [8], the two-objective functions are the statement coverage and the computational cost.

In the context of our study we formulated the following research questions:

---

Table I

<table>
<thead>
<tr>
<th>Program</th>
<th>Lines of Code</th>
<th>Avg. test suite size</th>
<th>Tot. number of test cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>printtokens</td>
<td>726</td>
<td>16</td>
<td>4,130</td>
</tr>
<tr>
<td>printtokens2</td>
<td>570</td>
<td>17</td>
<td>4,115</td>
</tr>
<tr>
<td>schedule</td>
<td>412</td>
<td>8</td>
<td>2,650</td>
</tr>
<tr>
<td>schedule2</td>
<td>374</td>
<td>8</td>
<td>2,710</td>
</tr>
</tbody>
</table>

For asymmetric diversity in the objective space we mean a mechanism that encourages the diversity for only one of the objective functions.
• **RQ1:** Do the proposed diversity preserving methods improve the optimality of vNSGA-II?

• **RQ2:** Do the proposed diversity preserving methods improve the convergence speed of vNSGA-II?

To answer our research questions, we applied three variants of the vNSGA-II algorithm on the selected programs:

- **vNSGA-II:** the original vNSGA-II proposed by Yoo and Harman [8]. This is our baseline to evaluate the benefits of the proposed diversity-preserving methods;
- **FS-vNSGA-II,** the vNSGA-II with an asymmetric fitness sharing in place of the crowding distance used in the original version of vNSGA-II;
- **DF-vNSGA-II,** the vNSGA-II with a dummy fitness function to analyze the density of the population in the $K$ partitions of the coverage space.

In the context of our study, we set $\sigma_{\text{share}} = 0.01$ (niche radius) for FS-vNSGA-II, while the number of partitions $K$ for the DF-vNSGA-II was set to 20. These two values have been empirically identified using a trial-and-error process.

All the experimented algorithms solve the test suite selection problem by selecting Pareto-optimal subsets of test cases. An initial population is pseudo-randomly generated with the aim of having the initial solutions randomly distributed in the phenotype space. All the algorithms have been executed ten times on each object program to account the inherent randomness of GAs. For all the algorithms we used the same configuration. Since the number of available test cases is really large (more than 2,000 for each program), the population size was set to 300 individuals with scattered crossover and bit-flip mutation [8]. The stopping criteria was average change in the spread of Pareto solutions less than $10^{-10}$ for 300 generations or a total of 10,000 generations.

All the algorithms were implemented using MATLAB’s Global Optimization Toolbox (release R2011b)\(^5\). The gamultiobj routine, that applies the vNSGA-II algorithm, is used to implement the two proposed diversity preserving methods.

### A. Evaluation Mechanism

By definition, it is impossible to know a priori the actual Pareto frontier of the test case selection problem. However, a simple a posteriori way to evaluate the optimality of the achieved Pareto frontiers is to construct a reference Pareto frontier. This is a hybrid frontier obtained combining the best parts of the different frontiers achieved by all approaches (in all the runs) and considering only the solutions that are not dominated by any other in the combined frontier.

Formally, let $P = \{P_1, \ldots, P_n\}$ be the set of $n$ different Pareto frontiers, the reference Pareto frontier $P_{\text{ref}}$ is defined as follows:

$$P_{\text{ref}} \subseteq \bigcup_{i=1}^n P_i, \forall p \in P_{\text{ref}} \exists q \in P_{\text{ref}} : q \succ p$$

Thus, $P_{\text{ref}}$ allows us to compare the global optimality of the different algorithms on the basis of the Pareto frontiers they produce.

We perform a comparison of the different algorithms by estimating three metrics widely used in global optimization problems:

- **Size of Pareto frontier,** that represents the number of non-dominated solutions obtained by each Pareto frontiers $P_i$.
- **Non dominated solutions,** that is the set of solutions that are not dominated by the reference Pareto frontiers $P_{\text{ref}}$. Formally, it can be defined as the cardinality of the set $P_i^* = \{ p \in P_i : \nexists q \in P_{\text{ref}} : q \succ p \}$.
- **Convergence speed,** that is the number of generations required to the convergence of each algorithm.

The first two metrics were used to answer RQ1, while the third metric is related to RQ2.

We also statistically analyze the results achieved to check whether the optimality and/or the convergence speed produced by two different algorithms are statistically significant or not. In particular, the values of the three employed metrics achieved in the ten runs by two algorithms were statistically compared using the Welch’s $t$ test [20]. Significant $p$-values suggest that the corresponding null hypothesis can be rejected in favor of the alternative hypothesis, i.e., that one of the algorithms produced a larger size of Pareto frontier. In all our statistical tests we rejected the null hypotheses for $p$-values $< 0.05$ (i.e., we accept a 5% chance of rejecting a null hypothesis when it is true [20]).

### B. Analysis of the Results

Table II reports the achieved results in terms of size of Pareto frontiers, number of non-dominated solutions, and number of generations. The reported values represent the mean of the value achieved for each metric on the 10 runs.

<table>
<thead>
<tr>
<th>Program</th>
<th>Method</th>
<th>Generation</th>
<th>Pareto size</th>
<th>Non Dominated Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Mean</td>
<td>M. Min</td>
<td>Mean</td>
</tr>
<tr>
<td>printtokens</td>
<td>vNSGA-II</td>
<td>3.699</td>
<td>313</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>FS-vNSGA-II</td>
<td>2.993</td>
<td>239</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>DF-vNSGA-II</td>
<td>2.399</td>
<td>103</td>
<td>21</td>
</tr>
<tr>
<td>printtokens2</td>
<td>vNSGA-II</td>
<td>3.699</td>
<td>296</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>FS-vNSGA-II</td>
<td>2.993</td>
<td>188</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>DF-vNSGA-II</td>
<td>2.409</td>
<td>156</td>
<td>50</td>
</tr>
<tr>
<td>schedule</td>
<td>vNSGA-II</td>
<td>1.032</td>
<td>47</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>FS-vNSGA-II</td>
<td>1.339</td>
<td>134</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>DF-vNSGA-II</td>
<td>1.084</td>
<td>73</td>
<td>41</td>
</tr>
<tr>
<td>schedule2</td>
<td>vNSGA-II</td>
<td>1.625</td>
<td>148</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>FS-vNSGA-II</td>
<td>1.262</td>
<td>227</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>DF-vNSGA-II</td>
<td>1.657</td>
<td>213</td>
<td>32</td>
</tr>
</tbody>
</table>

\(^5\)http://www.mathworks.it/products/global-optimization/
than those provided by FS-vNSGA-II and DF-vNSGA-II. These results indicated that the two proposed variants provide an improvement of the optimality as compared to vNSGA-II, as also highlighted by Figure 1. The figure provides a graphical comparison between the Pareto frontiers achieved by the three algorithms and the reference Pareto frontier for prntokens. As it can be noticed, the Pareto frontiers provided by the two variants of vNSGA-II (especially the one provided by DF-vNSGA-II) are much closer to the reference Pareto frontier than the Pareto front provided by vNSGA-II, indicating the higher optimality of FS-vNSGA-II and DF-vNSGA-II. Both FS-vNSGA-II and DF-vNSGA-II provide a wider diversity of non-dominated solutions with higher coverage and uniformity along the Pareto frontier than vNSGA-II. Similar graphs, related to the other programs show consistent results and are not shown for sake of space limitations. Thus, we can provide a positive answer to RQ1.

The proposed diversity-preserving techniques not only improve the optimality of non-dominated solutions provided by a Pareto efficient GA, but also improve its convergence speed (RQ2). In particular, FS-vNSGA-II allows to save more than 300 generations with respect to vNSGA-II. DF-vNSGA-II performs even better. In particular, it always allows to save more than 500 generations (more than 1,000 for prntokens and prntokens2) with respect to the baseline method (vNSGA-II). Since each fitness evaluation implies a matrix product between very large matrices (the number of columns is greater then 2,500), the achieved result represents an important improvement in terms of computational costs.

The above considerations are also supported by our statistical analysis (see Table III). In particular, the Welch’s t-tests confirm that diversity measures improve the optimality (RQ1) of vNSGA-II for multi-objective test case selection problem as well as the convergence speed (RQ2). Comparing DF-vNSGA-II and FS-vNSGA-II, none of the two algorithms turns out to be statistically better in terms of Pareto optimality, while DF-vNSGA-II has a statistically higher convergence speed than FS-vNSGA-II.

We also performed a qualitative analysis of the achieved results in order to try to characterize the test cases selected by DF-vNSGA-II and/or FS-vNSGA-II and discarded by vNSGA-II. We observed that the test cases that were selected only by the two variants of vNSGA-II usually test boundary conditions. In particular, the two variants of vNSGA-II selected about 50% (for prntokens and prntokens2 more than 100%) more test cases that are related to borderline input than those selected by vNSGA-II. For example, such test cases on the prntokens program aim at testing the program behavior when the input is represented by a string that is empty or contains invalid identifiers (i.e. `2variable`, `@interger`, etc.). Thus, as side-effect of the more diverse Pareto frontiers, the test cases selected by the two variants of vNSGA-II ensure a proper boundary testing.

C. Threats to Validity

This section discusses the threats to validity that can affect our results. Threats to construct validity concern the relationship between theory and observation. We used several well known metrics (number of required generations, Pareto frontier size, number of solutions not dominated by reference Pareto frontiers) for investigating whether the optimality and the convergence speed of vNSGA-II improves when using the proposed diversity preserving techniques. One potential construct validity involves the correctness of the coverage and computation cost information. To mitigate such a threat, the original code coverage information was extracted from the SIR dataset, while cost information is measured by counting the number of source code statements expected to be executed by the test cases.
Threats to internal validity involves the random nature of the meta-heuristic techniques themselves. To address this problem, we run the experimented algorithms 10 times for each subject program (as done in previous works [8], [12]), and considered the mean values of the measures used to evaluate the optimality and the convergence speed. Another threat related to internal validity is represented by the choice of the niched radius (for FS-vNSGA-II) and the number of partitions \( K \) (for DF-vNSGA-II). We experimented several values and used in our experimentation the ones providing the best performances.

A relevant threats to external validity is related to the programs used in the case study. The chosen subjects are benchmark systems available on SIR, that is a well managed software archive widely used in previous work about regression testing. However, in order to corroborate our findings, replications on a wider range of programs and optimization techniques are needed.

Finally, for what concerns conclusion validity, we support our findings by using proper statistics, i.e. the Welch’s t-test intended for use with two samples having possibly unequal variance. We performed Wilk-Shapiro normality test to make sure Welch’s t-test was applicable on our data.

VI. CONCLUSIONS AND FUTURE WORK

This paper proposed two heuristics, one based on fitness sharing and another based on a density function, to enhance diversity of Pareto fronts in the vNSGA-II multi-objective test case selection proposed by Yoo and Harman [8]. An empirical study conducted over four programs and test suites of the Siemens benchmark showed that the proposed heuristics improve both the optimality and the converge speed of vNSGA-II.

Future work aims at replicating the study over a large set of programs, as well as at deeply investigating the characteristics of the obtained sets of solutions.

REFERENCES


