Pareto-Optimal Search-Based Software Engineering (POSBSE): A Literature Survey

Abdel Salam Sayyad  Hany Ammar
Lane Department of Computer Science and Electrical Engineering
West Virginia University
Morgantown, WV, USA
asayyad@mix.wvu.edu  hany.ammar@mail.wvu.edu

Abstract—The Search-Based Software Engineering (SBSE) community is increasingly recognizing the inherit “multiobjectiveness” in Software Engineering problems. The old ways of aggregating all objectives into one may very well be behind us. We perform a well-deserved literature survey of SBSE papers that used multiobjective search to find Pareto-optimal solutions, and we pay special attention to the chosen algorithms, tools, and quality indicators, if any. We conclude that the SBSE field has seen a trend of adopting the Multiobjective Evolutionary Optimization Algorithms (MEOAs) that are widely used in other fields (such as NSGA-II and SPEA2) without much scrutiny into the reason why one algorithm should be preferred over the others. We also find that the majority of published work only tackled two-objective problems (or formulations of problems), leaving much to be desired in terms of exploiting the power of MEOAs to discover solutions to intractable problems characterized by many trade-offs and complex constraints.

Index Terms—Multiobjective Optimization, Pareto-Optimal Solutions, Search-Based Software Engineering.

I. INTRODUCTION

Many real-life challenges involve trading off multiple evaluation criteria. Rather than aggregating all the criteria in one equation that leads to a unique optimal solution, it is more pragmatic to present the end-user with a bouquet of candidate solutions that are reasonably distributed along the possible objective values. These are formally known as Pareto-optimal solutions or the Pareto front.

Historically, the field of Search-Based Software Engineering (SBSE) has seen a slow adoption of Pareto optimization techniques, generally known as multiobjective optimization techniques1. Back in 2001, when Harman and Jones coined the term SBSE [24], all surveyed and suggested techniques were based on single-valued fitness functions. In 2007, Harman commented on the current state and future of SBSE [23], and in the “Road-map for Future Work” section he suggested using multiobjective optimization. Then in 2009, Harman et al. [26] were able to cite several works in which multiobjective optimization techniques were deployed. Still, most of the work reviewed therein, as well as work done thereafter, optimized two objectives only, while higher numbers appeared only occasionally. Also, the typical tendency was to use algorithms that are popular in other domains, such as NSGA-II and SPEA2. In this survey, we attempt to systematically identify these trends and provide some recommendations to the growing SBSE community.

We were motivated to undertake this survey by our recent finding [44] that IBEA, an algorithm never used before in software engineering, outperformed several others (including NSGA-II and SPEA2) when it was applied to complex models and many objectives, hence our questions regarding suitability of optimizers to software engineering problems.

Several surveys and review papers were published in the field of SBSE [23], [26], [27], [11]. Some surveys focused on subfields, such as search-based software design [40], and search-based test data generation [37]. This paper is the first of its kind, in which we survey SBSE research work that employed Pareto optimization techniques. Such work spans the application areas of software requirements, design, testing, and management.

The rest of the paper is organized as follows: section II provides brief background material on Pareto optimality. Section III describes the survey’s inclusion/exclusion criteria. Section IV presents the results of the survey. In sections V, we analyze the survey data, and in section VI we present our conclusions and recommendations.

II. BACKGROUND

A. Pareto Optimization

Pareto optimization is the process of finding a set of non-dominated solutions from a pool of candidate solutions. Each non-dominated solution can be viewed as an optimal trade-off in all the objectives together.

Formally, a vector \( u = \{u_1, \ldots, u_k \} \) is said to dominate a vector \( v = \{v_1, \ldots, v_k \} \) if and only if \( u \) is partially less than \( v \), i.e.

\[
\forall i \in \{1, \ldots, k\}, u_i \leq v_i \text{ and } \exists i \in \{1, \ldots, k\} : u_i < v_i \quad (1)
\]

The set of all points in the objective space that are not dominated by any other points is called the Pareto Front.

B. Multiobjective Evolutionary Optimization Algorithms (MEOAs)

MEOAs are the most popular class of Pareto optimizers. Among them, the most popular in SBSE are NSGA-II [13],

---

1 Some techniques (e.g. awGA, HATS, Max-Min) are labeled as multiobjective but they actually aggregate the objectives and provide a single optimal solution. Such techniques are outside the scope of this survey.
C. Tools/Frameworks

Tools are available in which MEOAs are coded and ready to use. The most comprehensive tool we know is jMetal [14], which implements 17 different MEOAs in addition to Random Search, and provides a framework for experimentation with standard and user-defined problems.

D. Quality Indicators

Quality indicators can be calculated to assess the performance of MEOAs by computing aggregate values from the objective values in the Pareto front. For example, jMetal [14] allows the computation of six quality indicators, namely: Hypervolume (HV), Spread, Generational Distance (GD), Inverted Generational Distance (IGD), Epsilon, and Generalized Speed.

Other quality indicators were found in the SBSE literature that we surveyed. They were: coverage, convergence, error ratio, and attainment surfaces.

III. INCLUSION/EXCLUSION CRITERIA

We seek to include all published research works, up to date, in the field of search-based software engineering (SBSE) that provide Pareto-optimal solutions using multiobjective optimization algorithms. In each paper, we look for the number of objectives, the algorithms and how they were chosen, whether a framework for metaheuristic algorithms was used, and whether quality indicators were used to judge the quality of the Pareto front.

When more than one paper is published by the same author(s) that apply the same technique(s) to the same class of problems, we list only one of those papers, unless the extra paper adds algorithms for comparison, or changes the number of objectives, or adds quality indicators.

We utilized the CREST center SBSE repository\(^2\), which is the primary listing of SBSE papers. The “area of application” classification that we follow here is, generally, the same one used in that repository.

IV. RESULTS

The total number of surveyed papers was 51. Figure 1 breaks them down by area of application, while Figure 2 classifies them by year of publication.

Table 1 is a list of all the surveyed papers, along with the multiobjective algorithms, the number of objectives, tools and quality indicators.

V. ANALYSIS

In this section, we analyze and provide commentary about the data collected from the surveyed papers with regard to algorithms, number of objectives, tools, and quality indicators.

\(^2\) http://crestweb.cs.ucl.ac.uk/resources/sbse_repository/
### Table 1: List of All Surveyed Works

<table>
<thead>
<tr>
<th>Ref</th>
<th>Author(s)</th>
<th>Year</th>
<th>Title</th>
<th>Number of Objectives</th>
<th>Algorithm(s)</th>
<th>Tool</th>
<th>Quality Indicators</th>
</tr>
</thead>
<tbody>
<tr>
<td>[35]</td>
<td>Z. Liu, H. Guo, D. Li, T. Han, J. Zhang</td>
<td>2007</td>
<td>Solving Multi-objective and Fuzzy Multi-attributive Integrated Technique for QoS-Aware Web Service Selection</td>
<td>5</td>
<td>MOGA</td>
<td>Frontier</td>
<td>--</td>
</tr>
<tr>
<td>Ref</td>
<td>Author(s)</td>
<td>Year</td>
<td>Title</td>
<td>Number of Objectives</td>
<td>Algorithm(s)</td>
<td>Tool</td>
<td>Quality Indicators</td>
</tr>
<tr>
<td>------</td>
<td>----------------------------------</td>
<td>------</td>
<td>---------------------------------------------------------</td>
<td>----------------------</td>
<td>----------------------</td>
<td>---------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>[56]</td>
<td>S. Yoo, M. Harman</td>
<td>2007</td>
<td>Pareto Efficient Multi-Objective Test Case Selection</td>
<td>2, 3</td>
<td>NSGA-II, vNSGA-II</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>[38]</td>
<td>G. H. L. Pinto, S. R. Vergilio</td>
<td>2010</td>
<td>A Multi-Objective Genetic Algorithm to Test Data Generation</td>
<td>3</td>
<td>NSGA-II</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>[34]</td>
<td>W. B. Langdon, M. Harman, Y. Jia</td>
<td>2010</td>
<td>Efficient Multi Objective Higher Order Mutation Testing with Genetic Programming</td>
<td>2</td>
<td>NSGA-II</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>[57]</td>
<td>S. Yoo, M. Harman</td>
<td>2010</td>
<td>Using Hybrid Algorithm For Pareto Efficient Multi-Objective Test Suite Minimisation</td>
<td>2, 3</td>
<td>HNSGA-II</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

### Application Area: Software Project Management

<table>
<thead>
<tr>
<th>Ref</th>
<th>Author(s)</th>
<th>Year</th>
<th>Title</th>
<th>Number of Objectives</th>
<th>Algorithm(s)</th>
<th>Tool</th>
<th>Quality Indicators</th>
</tr>
</thead>
<tbody>
<tr>
<td>[22]</td>
<td>S. Gueorguiev, M. Harman, G. Antoniol</td>
<td>2009</td>
<td>Software project planning for robustness and completion time</td>
<td>2</td>
<td>SPEA2</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

### Application Area: Various Applications (Next Release Problem & Test Case Selection)

<table>
<thead>
<tr>
<th>Ref</th>
<th>Author(s)</th>
<th>Year</th>
<th>Title</th>
<th>Number of Objectives</th>
<th>Algorithm(s)</th>
<th>Tool</th>
<th>Quality Indicators</th>
</tr>
</thead>
</table>
Most researchers didn’t state any reason for adopting a certain MEOA (42%) or stated popularity of the algorithm (particularly NSGA-II) as the sole reason (25%). This is usually justified by the fact that many of those papers introduced Pareto-optimal solutions to their problems for the first time, which in itself is considered a contribution. A sizeable portion of these papers (10, 28%) introduced new MEOAs to solve a problem, but only compared their outcome with that of single-objective algorithms. Looking at the 15 papers that actually compared MEOAs to one another, 5 of them introduced new MEOAs and compared their outcomes with popular MEOAs.

In the single-MEOA papers, only one stated that the MEOA was chosen because it was more suitable for the particular problem, and one paper stated that the chosen algorithm had been reported to have better performance in higher-dimension objective spaces.

**B. Number of Objectives**

The first step in Pareto optimization is to re-formulate the problem to bring out the competing objectives. Although the objectives were known all along, legacy research combined them into one fitness function. Pareto optimization researchers then had to decide which objectives to represent as separate, which to combine into one, and which to ignore; thus a different number of objectives may have been defined for the same problem. Figure 6 shows the variety of “number of objectives in the surveyed papers. There were 7 papers that presented different formulations for each problem, with a different number of objectives in each formulation.

**C. Tools/Frameworks**

17 papers (33%) reported using implementations of the algorithms that are available in tools such as jMetal (13 papers) and Matlab (2 papers). Two thirds of the time, the researchers had to code their own implementations, which is the assumption we made when tools weren’t mentioned.

**D. Quality Indicators**

15 papers (30%) used quality indicators to assess the quality of the Pareto fronts and, most of the time (12 papers), to compare the performance of various MEOAs against one another. Hypervolume (HV) was the most widely used indicator (12 papers), while there was lesser agreement on the use of other indicators.

Quality indicators are useful as aggregate measures for large sets of solutions, especially with higher dimensions in the objective space. Skipping the computation of quality indicators is understandable when it’s possible to directly assess the objective values, which is the case when the Pareto front can be charted in 2-D or 3-D.

**VI. Conclusions and Recommendations**

In this paper, we surveyed 51 research papers that applied multiobjective search-based optimization methods to software engineering problems. This being a relatively young trend in SBSE, we have observed certain shortcomings in many papers:

1. Lack of clarity regarding the reasons why an algorithm is chosen for a problem (Figure 5).
2. Tendency to simplify problems by specifying fewer objectives to evaluate (Figure 6).
3. Heavy reliance on personal implementations of widely-used algorithms (Table 1, the “Tool” column).
4. Lack of agreement on whether to utilize quality indicators, and which indicators to use (Table 1, the “Quality Indicators” column).

But we also noticed some promising directions:

1. Researchers are comparing algorithms against one another to discover better performance, and to reason about suitability of the algorithms to the problems at hand (15 papers out of 51).
2. Some papers are exploring different formulations of their problems wherein the complexity is increased and more objectives are evaluated (7 papers out of 51).
3. Increasing use of the open-source jMetal framework with its rich set of MEOAs and quality indicators (13 papers out of 51).

Finally, we offer some recommendations to the SBSE research community:

1- Single-valued fitness functions are a thing of the past. Software engineering problems are multiobjective by nature, and Pareto optimization is the best way to find all the possible trade-offs among the objectives such that the stakeholders can make enlightened decisions.
2- More attention needs to be paid to the suitability of an algorithm to the type of problem at hand. It is true that the right optimizer for a specific problem remains an open question [54], but there has to be a thought process about the structure of the problem and the suitability of the metaheuristic.
3- More comparisons regarding the performance of various algorithms when applied to specific problems.
4- Reformulating two- and three-objective problems to bring out objectives that might have been aggregated or ignored. In addition to being closer to the business reality of many competing objectives, this should put MEOAs under increased stress, and enable testing out reported results about certain MEOAs performing better than others at higher dimensions (e.g. [44], [4]).

5- Utilizing and contributing to the MEOAs available in jMetal, as well as its quality indicator offerings.

ACKNOWLEDGMENT

This research work was funded by the Qatar National Research Fund (QNRF) under the National Priorities Research Program (NPRP) Grant No.: 09-1205-2-470.

REFERENCES


