

# A Novel Aggregation-based Dominance for Pareto-based Evolutionary Algorithms to Configure Software Product Lines

Yani Xue<sup>a,\*</sup>, Miqing Li<sup>b</sup>, Martin Shepperd<sup>a</sup>, Stasha Lauria<sup>a</sup>, Xiaohui Liu<sup>a</sup>

<sup>a</sup>*Department of Computer Science, Brunel University London, Uxbridge, Middlesex, UB8 3PH, United Kingdom*

<sup>b</sup>*CERCIA, School of Computer Science, University of Birmingham, Edgbaston, Birmingham, B15 2TT, United Kingdom*

---

## Abstract

In software engineering, optimal feature selection for software product lines (SPLs) is an important and complicated task, involving simultaneous optimization of multiple competing objectives in large but highly constrained search spaces. A feature model is the standard representation of features of all possible products as well as the relationships among them for an SPL. Recently, various multi-objective evolutionary algorithms have been used to search for valid product configurations. However, the issue of the balance between correctness and diversity of solutions obtained in a reasonable time has been found very challenging for these algorithms. To tackle this problem, this paper proposes a novel aggregation-based dominance (ADO) for Pareto-based evolutionary algorithms to direct the search for high-quality solutions. Our method was tested on two widely used Pareto-based evolutionary algorithms: NSGA-II and SPEA2+SDE and validated on nine different SPLs with up to 10,000 features and two real-world SPLs with up to 7 objectives. Our experiments have shown the effectiveness and efficiency of both ADO-based NSGA-II and SPEA2+SDE: (1) Both algorithms could generate 100% valid solutions for all feature models. (2) The performance of both algorithms was improved as measured by the hypervolume metric in 7/9 and 8/9 feature models. (3) Even for the largest tested feature model with 10,000 features, it required under 40 seconds on a standard desktop to find 100% valid solutions in a single run of both algorithms.

*Keywords:*

Optimal feature selection, Software product line, Evolutionary algorithm, Multi-objective optimization

---

## 1. Introduction

With the development of mobile and service-based applications, companies need to reconfigure their applications in order to retain and extend their market share. To meet the demand of different customers, companies often develop and maintain many variations of software products [1]. Recently, there is an increasing trend to adopt software product lines (SPLs) to reduce development costs, shorten development

---

\*Corresponding author

*Email address:* [yani.xue@brunel.ac.uk](mailto:yani.xue@brunel.ac.uk) (Yani Xue)

cycles, and improve software reusability and flexibility [2]. An SPL is a class of similar software products, all of which share some core functionalities. Each product configuration is different with different features selected that aim to satisfy the specific requirements of a particular market segment [3].

A feature model [1] is a tree structure that provides representations of an SPL for configuring all possible software products. The key task of an SPL is to select a set of desired features from its feature model in order to fulfill multiple functional requirements (e.g., minimize the product cost, maximize users' preferences) and satisfy the constraints related to various features. In practice, real-world SPLs often contain hundreds or even thousands of features and complex constraints. For example, the Linux X86 kernel feature model from LVAT (Linux Variability Analysis Tools) [4] repository contains 6,888 features and 343,944 constraints. It is extremely difficult to manually select optimal features for products configuration in such a large and constrained search space. This is called the *optimal feature selection* problem [5].

The optimal feature selection problem can be seen as a *multi-objective optimization problem* (MOP) involving simultaneously optimization of two or more competing objectives [6]. Evolution computation techniques, such as particle swarm optimization (PSO) and evolutionary algorithms (EA) have been successfully applied in many real-world optimization problems due to their population-based metaheuristic that allows to search for a set of optimal solutions in a single run [7–12]. In the past decade, there have been many studies that adopt different *multi-objective evolutionary algorithms* (MOEAs) as automatic configuration approaches to solve the optimal feature selection problem [6, 13–19].

A representative MOEA in this area is the indicator-based evolutionary algorithm (IBEA) [20], which is a popular MOEA using a performance metric hypervolume to guide the search towards optimal solutions. Sayyad et al. [6], Sayyad et al. [13] and Tan et al. [15] have demonstrated its advantage over some popular MOEAs, such as the non-dominated sorting genetic algorithm II (NSGA-II) [21] and strength Pareto evolutionary algorithm 2 (SPEA2) [22] in searching for valid solutions on both small and large feature models from the online feature model repositories SPLOT[23] and LVAT, respectively. However, the experiments in [13] shown that IBEA was not able to achieve 100% valid solutions by using a proposed statistic analysis to detect prunable features for reducing search space and a “seeding” technique to find more valid solutions for some large feature models.

To address the challenge, Tan et al. [15] proposed a feedback-directed IBEA that use the number of constraint violations as feedback for two evolutionary operators to improve the ability of algorithm to search for more valid solutions. Henard et al. [14] introduced two smart evolutionary operators into IBEA, called SATIBEA where an invalid solution (a software product configuration with constraint violations) is identified and fixed by using the SAT solver. Hierons et al. [16] presented a *Shrink Prioritize* (SIP) framework for different type of MOEAs. In particular, the novel encoding method is used to identify prunable features and the  $(1 + n)$  approach is used to first optimize the number of constraint violations and then other objectives at the same time to guide the search to feasible space.

From the discussion of the state-of-the-art algorithms (e.g., SATIBEA and SIP-based MOEAs) above, we note that there are few studies focusing on developing new MOEAs to deal with the optimal feature selection problem in configuring SPLs. Therefore, two research questions arise:

1. Are traditional MOEAs really worse than the state-of-the-art algorithms in solving optimal feature selection problem for SPLs?
2. Are there methods that could improve the quality of solutions obtained by using traditional MOEAs, such as NSGA-II without using SAT solvers?

In this paper, we investigate along these lines and propose an *aggregation-based dominance* (ADO) for Pareto-based evolutionary algorithms to direct the search for high-quality solutions. ADO is in part inspired by recent studies that have demonstrated the effectiveness of aggregation methods to drive populations to different parts of the Pareto front through weighted vectors, which were originally designed in decomposition-based MOEAs (e.g., MOEA/D [24]).

ADO can be incorporated into three genetic operators fitness assignment, mating selection, and environmental selection in different Pareto-based evolutionary algorithms to search valid optimal solutions for a SPL. Our method has been tested on two widely used MOEAs: NSGA-II and SPEA2+SDE and validated on nine different SPLs with up to 10,000 features and two real-world SPLs. Experiments demonstrate that ADO-based NSGA-II and SPEA2+SDE can significantly improve the quality of solutions in the final population as measured by the hypervolume metric without decreasing the correctness of solutions when compared with the state-of-the-art algorithms.

Our major contributions can be summarized as follows:

- 1) We introduce a new concept, called aggregation-based dominance (ADO) into Pareto-based evolutionary algorithms to distinguish between individuals during the optimization process. ADO is used as a secondary criterion when Pareto dominance-based criterion fails to distinguish between individuals by applying Pareto-based evolutionary algorithms.
- 2) We implement the idea of ADO in NSGA-II and SPEA2+SDE respectively, to improve their performance on optimal feature selection for SPLs, resulting in two enhanced algorithms, i.e., NSGA-II-ADO and SPEA2+SDE-ADO.
- 3) Our experiments show that compared to the original SIP framework-based algorithms used in [16], the performance of both NSGA-II-ADO and SPEA2+SDE-ADO were improved as measured by hypervolume for 7/9 and 8/9 SPLs, respectively. Furthermore, SPEA2+SDE-ADO outperforms the other 4 state-of-the-art MOEAs in terms of hypervolume metric for 8 out of 9 SPLs. In addition both algorithms generated 100% valid solutions in less than 40 seconds on a standard desktop even for large SPLs with 10,000 features.

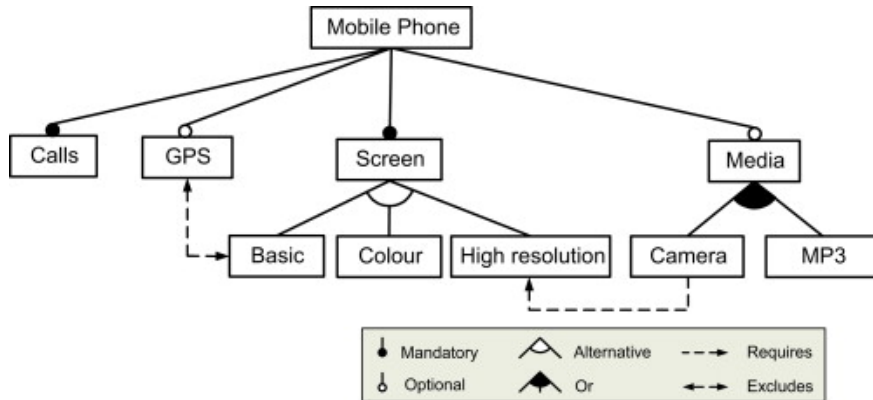
- 4) We evaluate the sensitivity of the algorithms to hyper-parameter settings and show that high quality solutions can be obtained with parameter settings within a range of 0.5 to 0.6.

This paper is organized as follows. Section 2 gives some background on software product lines and evolutionary multi-objective algorithms, and reviews the related work about MOEAs for optimal feature selection problems in configuring SPLs. In Section 3, we present our new aggregation-based dominance for Pareto-based evolutionary algorithms and its incorporation with NSGA-II and SPEA2+SDE. The experimental results are detailed in Section 4. Finally, the conclusions and future work are set out in Section 5.

## 2. Background

### 2.1. Optimal feature selection problem in SPLs

In software engineering, feature models are a standard representation of features for all possible products of an SPL and the relationships between them [25]. A feature model describes a product as valid feature combinations [26] by the expressions of constraints between them. A feature model is represented as a tree-like structure composed of a set of nodes representing features and connections between them. For example, in Fig. 1 illustrates a simplified feature model for mobile phone SPL.



**Fig. 1.** A simple feature model for mobile phone SPL (adapted from Benavides et al. [27])

Relationships between a parent feature and its child features (or sub-features) including:

- **Mandatory.** A mandatory feature must be included if its parent feature is included in a product, such as the “call” feature in the example.
- **Optional.** An optional feature can be optionally included in a product, so in Fig. 1, “Media” feature can be optionally included in products that contain its parent feature (“Mobile Phone” feature).

- **Alternative.** If the parent feature is included in a product, exactly one feature should be selected among a group of sub-features. For instance, a mobile phone must provide support for either a “Basic”, or a “Color”, or a “High resolution” screen in the same product.
- **Or.** If the parent feature is included in a product, one or more of child features should be selected. In Fig. 1, a mobile phone can provide support for a “Camera”, an “MP3”, or both of them when their parent feature “Media” is included in the products.

Apart from the above parental relationships between features, feature models also adopt cross-tree constraints (CTCs) to represent the mutual relationship for features. Typically, there are two types of CTCs:

- **Requires.** This relationship allows some features to co-occur, namely, If feature  $F_a$  requires feature  $F_b$ , the inclusion of  $F_a$  implies the inclusion of  $F_b$  in this product. In Fig. 1, the mobile phone with “Camera” feature requires the “high resolution” feature.
- **Excludes.** This relationship indicates that some feature cannot exist simultaneously in the same product, namely if a feature  $F_a$  excludes a feature  $F_b$ , the inclusion of  $F_a$  implies the exclusion of feature  $F_b$  in this product, and vice versa. In Fig. 1, the mobile phone with “GPS” feature excludes the “Basic” feature.

## 2.2. Multi-objective optimization

When dealing with optimization problems in the real world, it often involves two or more performance criteria in order to determine how “good” a certain solution is. These criteria, termed as objectives (e.g., cost, safety, efficiency) need to be optimized simultaneously, but usually conflict with each other. This type of problem is called multi-objective optimization problem (MOP). To model such problems, an MOP can be mathematically defined as follows [28]:

$$\begin{aligned}
& \text{minimize/maximize} && z = F(x) = (f_1(x), f_2(x), \dots, f_n(x)), && z \in Z \\
& \text{subject to} && g_j(x) \leq 0, && j = 1, 2, \dots, J \\
& && h_k(x) = 0, && k = 1, 2, \dots, K \\
& && x = (x_1, x_2, \dots, x_m), && x \in X
\end{aligned} \tag{1}$$

where  $x$  denotes an  $m$ -dimensional decision variable vector from the feasible region in the decision space  $X$ .  $z$  represents an  $n$ -dimensional objective vector,  $f_i(x)$  is the  $i$ -th objective to be minimized. Objective functions  $f_1, f_2, \dots, f_n$  constitute a multi-dimensional space called the objective space  $Z$ .  $J$  and  $K$  are the numbers of inequality and equality constraints, respectively.

*Definition 1 (Pareto Dominance).* Given two decision vectors  $p, q \in X$  of a minimization problem.  $p$  is said to *dominate*  $q$  (denoted as  $p \prec q$ ), or equivalently  $q$  is dominated by  $p$ , if and only if [29]

$$\forall i \in (1, 2, \dots, m) : f_i(p) \leq f_i(q) \wedge \exists i \in (1, 2, \dots, m) : f_i(p) < f_i(q) \tag{2}$$

Namely, given two solutions, one solution is said to Pareto dominate, or simply dominates the other solution if it is at least as good as the other solution in any objective and is strictly better in at least one objective.

*Definition 2 (Pareto Optimality).* A decision vector  $p \in X$  is said to be Pareto optimal if and only if there is no  $q \in X$  that dominates it.

A decision vector that is not dominated by any other vector is called a Pareto-optimal solution (or non-dominated solution) of an MOP.

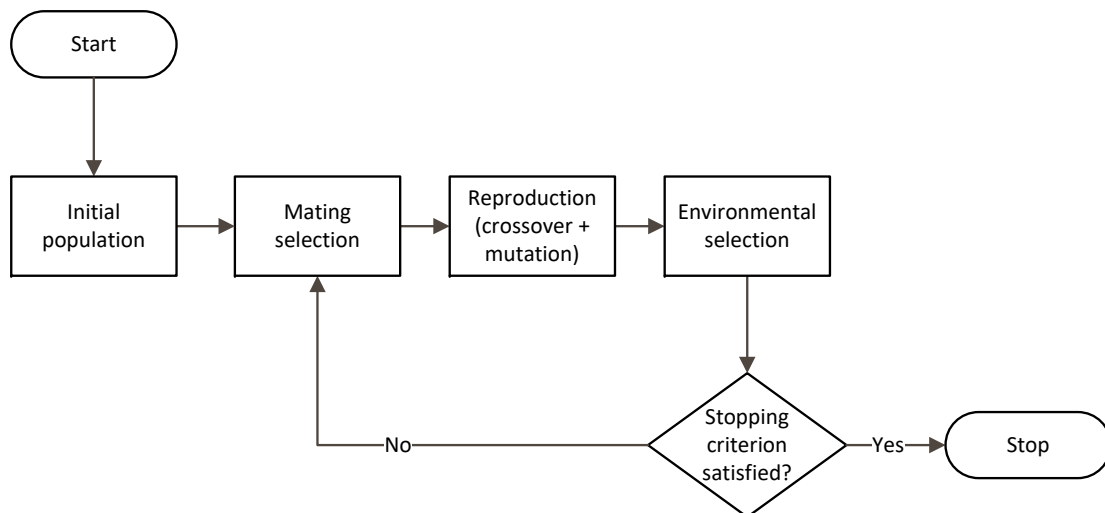
*Definition 3 (Pareto Set).* For a given MOP, the Pareto set (PS) is referred to a set of all Pareto-optimal (or non-dominated) solutions in the decision space. There is often no single optimal solution for an MOP, but rather a set of optimal trade-off solutions, namely Pareto-optimal solutions.

*Definition 4 (Pareto Front).* For a given MOP, the set of corresponding objective vectors to a Pareto set is called the Pareto front (PF).

We note here that identifying those Pareto optimal solutions is the key for a decision maker to select the trade-off solutions, which satisfy objectives as much as possible.

### 2.3. Multi-objective evolutionary algorithms

Evolutionary algorithms, inspired by natural selection, have shown high practicability in tackling MOPs especially if the search space is large and complex. They are referred to as multi-objective evolutionary algorithms (MOEAs). Their population-based property allows them to simultaneously deal with a number of possible solutions. Therefore, they could obtain several solutions from the Pareto set in a single run of the algorithm. A general framework of MOEA is illustrated in Fig. 2.



**Fig. 2.** A general MOEA framework

In an MOEA, first, an initial population of  $m$  parents is randomly generated. Second, in the mating selection, individuals that have better quality tend to become parents of the next generation to push for quality improvement. By doing this, the combination of these parents is more likely to generate good offspring. Third, evolutionary operators (e.g., crossover and mutation) are applied to these parents to produce  $n$  offspring. Fourth, environmental selection is applied to reduce the expanded  $m + n$  population of parents and offspring to  $m$  individuals as the parent population of the next generation. Each individual in the union population is ranked according to their fitness values which are determined by dominance criteria of the evolutionary algorithm. An individual with a higher rank will have a higher chance of survival in the next generation. This evolutionary process continues until a terminating condition (e.g., the number of generations exceeds a predefined upper bound) is reached.

The goal of MOEAs is to achieve a population with both good convergence (minimizing the distance of the resulting non-dominated solutions to the Pareto front) and diversity (maximizing the distribution of the obtained non-dominated solutions over the Pareto front). However, compared with low-dimensional multi-objective optimization problem (with two or three objectives), well-known Pareto-based evolutionary algorithms, such as non-dominated sorting genetic algorithm II (NSGA-II) [21], Pareto envelope-based selection algorithm II (PESA-II) [30], strength Pareto evolutionary algorithm 2 (SPEA2) [22] lose their efficiency in solving MOPs. In Pareto-based evolutionary algorithms, individuals are compared through two criteria: Pareto dominance relation and diversity. However, in high dimensional search space, most individuals in a population become non-dominated (i.e., equally good solutions). Since the Pareto dominance-based primary selection criterion is not effective to facilitate the convergence of the population, the second criterion – density – becomes the main criterion to guide the search, which leads to a substantial reduction of the selection pressure towards the Pareto front and the slowdown of the evolution process. This is termed the *active diversity promotion* (ADP) phenomenon in [31].

Some studies [32, 33] observed that ADP phenomenon could lead to the failure of Pareto-based algorithms in finding a good approximation of the Pareto front, because of the preference of dominance resistant solutions [34]. These are solutions which are extremely inferior to others in at least one objective, but nearly as good in other objectives. They have worse performance in terms of convergence, but they are treated as non-dominated solutions. As a result of the ADP phenomenon in Pareto-based evolutionary algorithms, the final set of solutions could be widely covered but far away from the true Pareto front.

To address this difficulty one approach is to reduce the number of objectives. The use of aggregation methods, such as the weighted sum function, weighted min-max function, and Tchebycheff function can be integrated into Pareto-based algorithms, which has shown good convergence performance and computational efficiency when integrated with EAs (e.g., repeated single objective (RSO) [35], MSOPS [36], MOEA/D [24], MODELS [37]). However, the diversity performance of these aggregation-based algorithms, especially for problems with irregular Pareto fronts, is subject to the distribution of weight vectors. In addition, to

adapt to higher dimensional search space, several studies focus on the modification of the primary criterion (Pareto dominance relation), such as  $\epsilon$ -domination based multi-objective evolutionary algorithm ( $\epsilon$ -MOEA) [38], fuzzy Pareto dominance [39],  $k$ -optimality [40], preference order ranking [41], and grid-dominance [42]. These enhancements of the primary criterion have been found to perform well by increasing the selection pressure toward the Pareto front during the evolutionary process. The main weakness of these modified Pareto dominance relations is that they often involve extra parameters and the performance depends on the setting of these parameters.

Another challenge lies in the design of evolutionary operators. As the search space becomes extremely large in MOPs with more than 3 objectives, solutions in one generation are likely to be very distant from each other. This can undermine the effectiveness of evolutionary operators (e.g., recombination). In the studies [43, 44], if two distant parent solutions are selected, the offspring solutions are likely to be distant from their parents. Therefore, it is necessary to develop particular recombination operators, such as mating restriction.

In order to deal with these issues above in applying Pareto-based evolutionary algorithms into optimal feature selection problem for SPLs, this paper presents aggregation-based dominance in terms of three evolutionary operators: fitness assignment, mating selection, and environmental selection. The goal of aggregation-based dominance is to rescue the negative influence due to dominance resistant solutions and continuously differentiate individuals during the searching process. These operators could work with other evolutionary operators, such as encoding, crossover, and mutation in Pareto-based evolutionary algorithms.

#### 2.4. Related work

Sayyad et al. [13] first formulated five optimization objectives and investigated seven MOEAs for optimal feature selection problem. The authors highlighted the importance to adopt correctness (the number of valid product to satisfy the requirement constraints) as an objective that should be minimized because feature models are often highly constrained. Their experiments demonstrated strong performance of IBEA among tested MOEAs. However, IBEA cannot deal with constraints exactly since the correctness of obtained solution set did not achieve 100% for two feature models with up to 290 features.

Subsequently, many approaches have been suggested for IBEA to deal with the difficulties caused by constraints in optimal feature selection for SPLs. For example, Henard et al. [14] proposed a hybrid SAT-IBEA algorithm that used the satisfiability (SAT) solver as the constraint solving technique implemented into “smart” mutation and replacement operators of IBEA. Their empirical study indicated that SATIBEA performs significantly better than the original IBEA on five large real-world feature models with features in the range 1,244 – 6,888. However, the two operators are more time-consuming since they are much more complicated than traditional ones. In addition, Tan et al. [15] proposed the feedback-directed mechanism that uses the number of violated constraints as feedback for crossover and mutation operators. The authors



demonstrated that IBEA outperformed three tested Pareto-based evolutionary algorithms (NSGA-II [21], ssNSGA-II [45], and MOCell [46]) in terms of the correctness on six tested feature models.

Moreover, some authors attempted to improve the performance of other MOEAs for handling optimal feature selection problem. In particular, Hierons et al. [16] addressed the problem using the SIP framework that provides a novel encoding to shrink the representation, and the  $(1 + n)$  approach to optimize violated constraints as an objective first and then the other objectives simultaneously. Their experiments showed that SIP is an effective in obtaining 100% valid products for three different types of MOEAs (e.g., NSGA-II [21], SPEA2+SDE [31], IBEA [20], and three variants of MOEA/D [24]), but there is no one MOEA superior to others.

Recently, Xiang et al. [17] highlighted the importance to maintain the diversity of obtained solutions. They proposed SATVaEA that combine SAT solvers (which used for the repair of an invalid product configuration and diversity promotion) and a new MOEA, called VaEA [47]. The results demonstrated that SATVaEA outperformed SATIBEA in terms of correctness of obtained solutions and two SIP-based MOEAs (i.e., NSGA-II and SPEA2+SDE) in terms of diversity. In addition, SATVaEA was shown to be an effective method since it only took few minutes to find 100% valid solutions for feature models with more than 10,000 features.

### 3. Aggregation-based dominance for optimal feature selection

In this section, we illustrate an aggregation-based dominance as a secondary criterion when Pareto-based criterion fails to distinguish between individuals in Pareto-based evolutionary algorithms. In the proposed aggregation-based dominance relation, an aggregation-based fitness function is used to calculate the estimated convergence performance of those non-dominated individuals. We aim to avoid the negative influence of dominance resistant solutions in the optimization by giving those solutions lower fitness values. Then, based on aggregation-based dominance, we present three improved evolutionary operators (fitness assignment, mating selection and environmental selection) to select and keep the promising individuals in each generation and guide a Pareto-based evolutionary algorithm to search for optimal solutions for SPLs. The proposed aggregation-based dominance is incorporated into NSGA-II and SPEA2+SDE, which are two representative Pareto-based evolutionary algorithms to handle optimal feature selection problems for SPLs, called NSGA-II-ADO and SPEA2+SDE-ADO, respectively.

#### 3.1. Aggregation-based dominance

##### 3.1.1. Normalization

We first need to normalize a MOP where objective values may be disparately scaled in different SPLs. This is to enhance the robustness of the algorithm when the scale of the objective values are different [24, 43].

Algorithm 1 shows the pseudocode for the objective normalization procedure. In normalization, every solution in a population is normalized according to the ideal point and nadir point of each objective. Formally, in a minimizing MOP and a population  $P$  with  $M$  individuals and  $N$  objectives, the ideal point  $z^{min} = (z_1^{min}, z_2^{min}, \dots, z_N^{min})^T$  is determined by searching the minimum value of each objective for all individuals in  $P$ , which is denoted as  $z_j^{min} = \min_{i=1}^M f_j(x_i)$ ,  $x_i \in P$ , and  $i = 1, 2, \dots, M$  (line 1). Similarly, the nadir point  $z^{max} = (z_1^{max}, z_2^{max}, \dots, z_N^{max})^T$  is calculated by  $z_j^{max} = \max_{i=1}^M f_j(x_i)$ , where  $z_j^{max}$  is the maximum of  $f_j(x_i)$ ,  $x_i \in P$ , and  $i = 1, 2, \dots, M$  (line 2). For each solution  $\mathbf{x} \in P$ , its objective vector  $f_j(\mathbf{x})$ ,  $j = 1, 2, \dots, N$  is normalized to  $\tilde{f}(\mathbf{x}) = (\tilde{f}_1(\mathbf{x}), \tilde{f}_2(\mathbf{x}), \dots, \tilde{f}_N(\mathbf{x}))^T$  which is calculated by the following equation (lines 3, 5-6 and 8-9):

$$\tilde{f}(\mathbf{x}) = \frac{f_j(\mathbf{x}) - z_j^{min}}{z_j^{max} - z_j^{min}} \quad (3)$$

---

**Algorithm 1** Normalization( $P$ )

---

**Require:**  $P$  (the current population),  $M$  (population size),  $N$  (the number of objective function)

- 1: find  $z_j^{min} = \min_{i=1}^M f_j(x_i)$ ,  $x_i \in P$ ,  $i = 1, 2, \dots, M$
  - 2: find  $z_j^{max} = \max_{i=1}^M f_j(x_i)$ ,  $x_i \in P$ ,  $i = 1, 2, \dots, M$
  - 3: **for**  $i = 0 \rightarrow M$  **do**
  - 4:      $f^{con}(x_i) = 0$
  - 5:     **for**  $j = 0 \rightarrow N$  **do**
  - 6:          $\tilde{f}_j(x_i) = (f_j(x_i) - z_j^{min}) / (z_j^{max} - z_j^{min})$
  - 7:          $f^{con}(x_i) = f^{con}(x_i) + \tilde{f}_j(x_i)$
  - 8:     **end for**
  - 9: **end for**
- 

During the normalization procedure, the aggregation value of individual  $x_i$ , denoted as  $f^{con}(x_i)$  is also calculated, which is described in the following Section 3.1.2.

### 3.1.2. Aggregation function

Aggregation functions are basic techniques in recent decomposition-based MOEAs and often used to decompose a MOP into a number of sub-problems and optimize these problems in a collaborative way. However, we attempt to use aggregation functions in a different manner and enhance the selection pressure in traditional Pareto-based evolutionary algorithms. Here, an aggregation function is used to estimate the convergence performance of an individual, which is based on aggregated individual's information (by taking the all normalized objective value into consideration).

Our aggregation function seeks to estimate the convergence performance for each individual  $x_i$ ,  $i = 1, 2, \dots, M$ , in population  $P$ , denoted as  $f^{con}(x_i)$ . This aggregation value considers the sum of each normalized

objective value in the range  $[0, 1]$  (lines 3-5 and 7-9 in Algorithm 1), formulated as [48]

$$f^{con}(x_i) = \sum_{j=1}^N \tilde{f}_j(x_i) \quad (4)$$

where  $\tilde{f}_j(x)$  denotes the normalized objective value of individual  $x$  in the  $j$ -th objective, and  $N$  is the number of objectives. This aggregation function is determined by two factors: first the number of objectives, and second the performance in each objective. In minimizing multi-objective optimization problems, an individual with good convergence (that has slightly worse value in at least one objective but has significantly better value in most of the other objectives) is more likely to obtain a lower (better) aggregation value. A smaller aggregation value of an individual could indicate a good convergence performance. If an individual/solution is a dominance resistant solution, it is more likely to obtain an extremely large aggregation value compared to other individuals in a population.

The aggregation method will be involved later in the definition of aggregation-based dominance relation, where the aggregated information of an individual is compared with others.

### 3.1.3. Aggregation-based dominance

In a minimization MOP with  $N$  objectives, the proposed aggregation-based dominance is defined on population  $P$ , where each individual  $\mathbf{x}$  is assigned an aggregation value representing its estimated convergence performance, denoted as  $f^{con}(\mathbf{x})$ .

*Definition 5 (Convergence Difference).* Let  $\mathbf{x}, \mathbf{y} \in P$  where  $\mathbf{x}$  and  $\mathbf{y}$  are non-dominated solutions with respect to Pareto dominance, the convergence difference between them is denoted as:

$$CD(\mathbf{x}, \mathbf{y}) = f^{con}(\mathbf{x}) - \alpha \cdot f^{con}(\mathbf{y}) \quad (5)$$

$\alpha$  is a predefined parameter  $[0 < \alpha < 1]$ . With the definition of  $CD(\cdot)$ , aggregation-based dominance is defined as follows:

*Definition 6 (Aggregation-based dominance).* Let  $\mathbf{x}, \mathbf{y} \in P$  where  $\mathbf{x}$  and  $\mathbf{y}$  are non-dominated solutions with respect to Pareto dominance.  $\mathbf{x}$  is said to ADO-dominate  $\mathbf{y}$ , or equivalently  $\mathbf{y}$  is ADO-dominated by  $\mathbf{x}$ , denoted by  $\mathbf{x} \prec_{ADO} \mathbf{y}$ , if

$$CD(\mathbf{x}, \mathbf{y}) < 0 \quad \text{and} \quad CD(\mathbf{y}, \mathbf{x}) > 0 \quad (6)$$

Similarly,  $\mathbf{y}$  ADO-dominates  $\mathbf{x}$  when  $CD(\mathbf{y}, \mathbf{x}) < 0$  and  $CD(\mathbf{x}, \mathbf{y}) > 0$ . Otherwise, both individual  $\mathbf{x}$  and  $\mathbf{y}$  are ADO-non-dominated solutions. It is clear that there is no possibility of both  $CD(\mathbf{x}, \mathbf{y}) < 0$  and  $CD(\mathbf{y}, \mathbf{x}) < 0$  at the same time.

Aggregation-based dominance is partly inspired by the idea of using additional convergence-related criterion in addition to traditional Pareto dominance based criterion, such as knee points in KnEA [49], a grid-dominance-based criterion defined in [50] and so on. The proposed aggregation-based dominance is

used as a selection criterion, which is denoted as aggregation-based dominance (called ADO) in this paper. ADO is activated when the Pareto dominance-based selection criterion fails to distinguish between individuals in the evolutionary process. For ADO integrated with traditional Pareto-based evolutionary algorithms, such as NSGA-II, we use an ADO-nondominated sorting in selection procedure to classify  $P$  into different ADO-nondomination levels.

Similar to  $\epsilon$ -dominance [51] in the  $\epsilon$ -dominance-based algorithm [38], the ADO also modifies the traditional Pareto dominance to enhance the selection pressure towards the Pareto front. Both of them can be categorized into a relax form of Pareto dominance, which are used to determine the survival of individuals in the evolutionary process. However, the difference is that  $\epsilon$ -dominance enlarges the dominating space of each individual in population, while the proposed ADO is cooperated with Pareto dominance and designed for those non-dominated individuals.

In the following sections, ADO is integrated into three evolutionary operators – fitness assignment, mating selection, and environmental selection in which individuals are compared with each other in terms of convergence and diversity. In our algorithm ADO is used as an additional criterion to Pareto dominance. By applying ADO, we aim to enhance the selection pressure on those non-dominated solutions in the population of each generation and avoid the negative influence of dominance resistant solutions in the optimization process.

### 3.2. Fitness assignment

In Pareto-based evolutionary algorithms, the fitness of individuals should cover the performance of each individual in terms of *both* convergence and diversity. To deal with ADP phenomenon of Pareto-based evolutionary algorithms, our fitness assignment strategy focuses on the calculation of convergence by incorporating the ADO into Pareto domination-based fitness assignment strategies. In our fitness assignment strategy, Pareto dominance and ADO are used together to evaluate the performance of individuals in terms of convergence, which works by applying ADO when Pareto dominance fails to distinguish individuals in a population. There are different fitness assignment strategies based on Pareto domination to calculate the performance of individuals. In the experimental study, our strategy was tested on the fitness assignment strategies of two popular Pareto-based evolutionary algorithms: NSGA-II and SPEA2+SDE.

First strategy tested is introduced by Deb et al. in NSGA-II [21], called non-dominated sorting-based fitness assignment. It suggested using a Pareto dominance relation to distinguish individuals in a mixed set of population. In this strategy, those solutions that are non-dominated in terms of Pareto optimal become the first front. After removing these solutions temporarily, the remaining non-dominated solutions constitute the second front, and so on. Finally, each individual in the same front is assigned a nondomination rank. Lower rank values represent better degrees of convergence for individuals. The final fitness of an individual depends on its nondomination rank (represents convergence information) and the crowding distance in (represents

diversity information). Compared to our fitness assignment, the difference is on the non-domination rank calculation of individuals in NSGA-II. Those individuals in each front depend on both Pareto dominance and aggregation-based dominance.

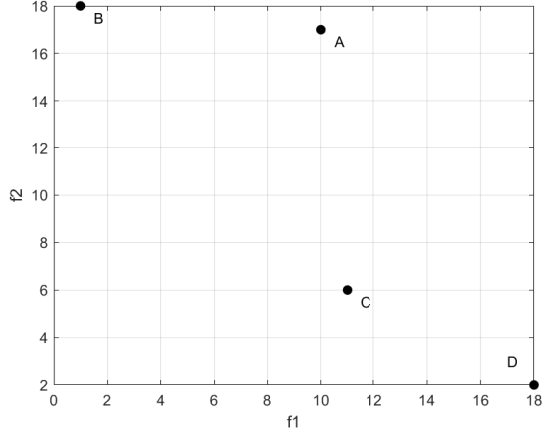
Second strategy tested is a fine-grained fitness assignment that is proposed in SPEA2 [22]. Compared with non-dominated sorting, this strategy only uses total fitness to include both convergence and density information of an individual in a population. In the fitness assignment process, an individual  $\mathbf{x}$  is assigned a “strength” which reflects its domination degree. The strength is calculated based on Pareto dominance counts: the number of solutions it dominates. Then, individual  $\mathbf{x}$  is assigned a raw fitness  $R(\mathbf{x})$  which is determined by summing strength of the individuals that dominate it. A low raw fitness value means that an individual dominates many individuals which in turn is dominated by many individuals. In particular, non-dominated individuals are assigned value zero as raw fitness. This is followed by the density estimation  $D(\mathbf{x})$  of individual  $\mathbf{x}$  that takes  $k$ -th nearest neighbor of individual  $\mathbf{x}$  into consideration. Finally, the total fitness  $F(\mathbf{x})$  of individual  $\mathbf{x}$  is calculated by the equation:  $F(\mathbf{x}) = R(\mathbf{x}) + D(\mathbf{x})$ . For SPEA2+SDE, the only difference is the density estimation of an individual  $\mathbf{x}$ , where  $D(\mathbf{x})$  is calculated based on shifted position of other individuals in a population according to the convergence comparison between individual  $\mathbf{x}$  and other individuals on each objective. In contrast to our fitness assignment, strength value of individuals in SPEA2+SDE is calculated according to both Pareto dominance and aggregation-based dominance among individuals in a population.

In the fitness assignment process, the diversity estimation also plays an important role. It aims to obtain a set of well-distributed solutions along the entire Pareto front. In our fitness assignment, density estimation techniques remain the same as in original Pareto-based evolutionary algorithms (e.g., clustering technique in SPEA [52], crowding distance in NSGA-II [21],  $k$ -th nearest neighbor in SPEA2 [22], and Shift-Based Density Estimation (SDE) in SPEA2+SDE [31]).

In summary, the main difference in the fitness assignment process by incorporating the ADO is the evaluation of convergence for an individual. It leads to some changes of calculating non-domination rank of individuals in NSGA-II and strength value of individuals in SPEA2+SDE. The rest is applied in the usual way as proposed in original NSGA-II and SPEA2+SDE. In the following, we will evaluate the impact of ADO in fitness assignment process by considering a sorting strategy in NSGA-II-ADO as an example.

To illustrate ADO in the fitness assignment, let’s consider a population of 4 non-dominated individuals in a particular generation, including A (10, 17), B (1, 18), C (11, 6) and D (18, 2) in a bi-objective minimization scenario, as shown in Fig. 3.

Table 1 shows ranking assignment results of an individual A in a typical bi-objective minimization scenario. In particular, the column  $f_1$  and  $f_2$  represents two objective values of each individual, while column  $\tilde{f}_1$  and  $\tilde{f}_2$  are the corresponding normalized objective value of each individual respectively. The column  $f^{con}(\cdot)$  is the aggregation-based fitness value of an individual, which is calculated by simply summing



**Fig. 3.** Four non-dominated individuals in a bi-objective minimization scenario

up all normalized objective value of that individual. Column Comparison lists the ADO-based comparison results for two individuals. If there is an individual  $\mathbf{x}$  ADO-dominates individual  $\mathbf{y}$ , namely  $\mathbf{x} \prec_{ADO} \mathbf{y}$ ,  $\mathbf{y}$  will be assigned with a rank equals to the rank of  $\mathbf{x}$  adding 1, which will be recorded in Column Rank. By doing so, an individual with better convergence will be assigned a lower value, and it will be more likely to survive into the next generation.

**Table 1**

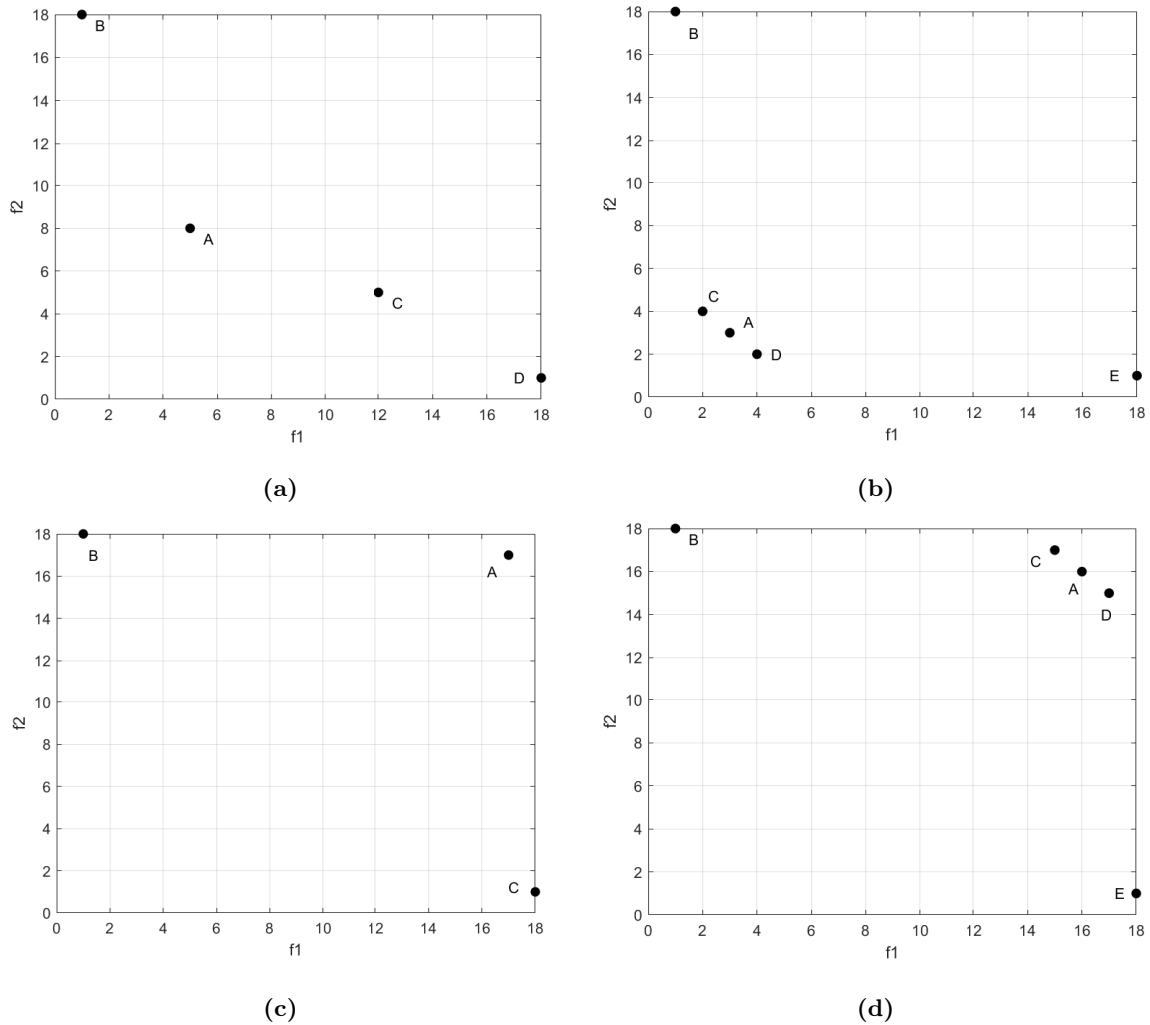
The ranking assignment results of an individual in a typical bi-objective minimization scenario by using the ADO

Individuals	(f1, f2)	$\tilde{f}_1$	$\tilde{f}_2$	$f^{con}(\cdot)$	Comparison	Rank
A	(10, 17)	0.53	0.94	1.47	—	2
B	(1, 18)	0	1	1	—	1
C	(11, 6)	0.59	0.25	0.84	$C \prec_{ADO} A$	1
D	(18, 2)	1	0	1	—	1

In the fitness assignment process of NSGA-II-ADO, a lower rank of an individual implies better fitness degree. Considering two non-dominated individuals A and C in Table 1, A, which is ADO-dominated by C ( $CD(C, A) = 0.84 - \alpha \times 1.47 < 0$ ,  $CD(A, C) = 1.47 - \alpha \times 0.84 > 0$  assuming  $\alpha = 0.6$ ), will have a larger rank value than C (the rank value of A (2) equals to the rank value of C (1) adding 1 [see Table 1]). Namely, A has worse fitness degree compared with C and it is more likely to be removed from the population in the following evolutionary process. In this scenario, Individual A is a dominance resistant solution since there are two individuals B and C performing significantly better than individual A in one objective but slightly worse than A in the other objective. Therefore, this implies ADO could distinguish between dominance resistant solutions.

In order to explore the performance of aggregation-based dominance, we then consider four typical situations of individual A in a population for minimizing a MOP, namely, (a) good convergence and diversity,

(b) good convergence and poor diversity, (c) poor convergence and good diversity, and (d) poor convergence and diversity), as summarized in Fig. 4. Notice that these individuals in the population set are non-dominated with each other.



**Fig. 4.** Four situations of individual A in a population for minimizing a MOP. (a) Performing well in both convergence and diversity. (b) Performing well in convergence but poorly in diversity. (c) Performing poorly in convergence but well in diversity. (d) Performing poorly in both convergence and diversity.

Table 2 shows the ranking assignment results of individual A of four typical bi-objective minimization scenarios, which correspond to four situations described in Fig. 4, namely: (a) good convergence and diversity, (b) good convergence and poor diversity, (c) poor convergence and good diversity, and (d) poor convergence and diversity.

As shown in Table 2 (a) and Table 2 (b), an individual with either both good convergence and diversity, or good convergence but poor diversity will be assigned a low-rank value. Although the two types of individuals

**Table 2**

The ranking assignment results of an individual A in four typical bi-objective minimization scenarios by using the ADO.

Individuals	(f1, f2)	$\tilde{f}_1$	$\tilde{f}_2$	$f^{con}(\cdot)$	Comparison	Rank
A	(5, 8)	0.24	0.41	0.65	–	1
B	(1, 18)	0	1	1	–	1
C	(12, 5)	0.65	0.24	0.89	–	1
D	(18, 1)	1	0	1	–	1

**(a)** good convergence and diversity ( $k = 0.5$ )

Individuals	(f1, f2)	$\tilde{f}_1$	$\tilde{f}_2$	$f^{con}(\cdot)$	Comparison	Rank
A	(3, 3)	0.12	0.12	0.24	–	1
B	(1, 18)	0	1	1	$B \prec_{ADO} A$	2
C	(2, 4)	0.06	0.18	0.24	–	1
D	(4, 2)	0.18	0.06	0.24	–	1
E	(18, 1)	1	0	1	$E \prec_{ADO} A$	2

**(b)** good convergence and poor diversity ( $k = 0.5$ )

Individuals	(f1, f2)	$\tilde{f}_1$	$\tilde{f}_2$	$f^{con}(\cdot)$	Comparison	Rank
A	(17, 17)	0.94	0.94	1.88	–	2
B	(1, 18)	0	1	1	$B \prec_{ADO} A$	1
C	(18, 1)	1	0	1	$C \prec_{ADO} A$	1

**(c)** poor convergence and good diversity ( $k = 0.6$ )

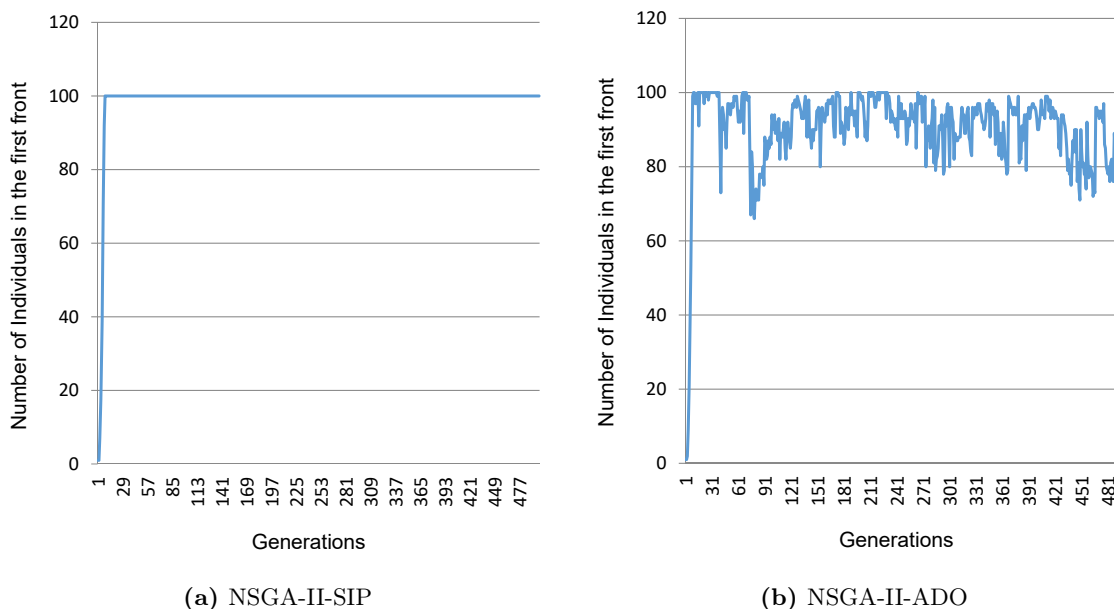
Individuals	(f1, f2)	$\tilde{f}_1$	$\tilde{f}_2$	$f^{con}(\cdot)$	Comparison	Rank
A	(16, 16)	0.88	0.88	1.76	–	2
B	(1, 18)	0	1	1	$B \prec_{ADO} A$	1
C	(15, 17)	0.82	0.94	1.76	$B \prec_{ADO} C$	2
D	(17, 15)	0.94	0.82	1.76	$B \prec_{ADO} D$	2
E	(18, 1)	1	0	1	$E \prec_{ADO} A$	1

**(d)** poor convergence and diversity ( $k = 0.6$ )



have high possibility of existing in the same front, it could be solved by the diversity maintenance techniques of different Pareto-based evolutionary algorithms. In addition, the individuals with poor convergence will be assigned a high-rank value no matter how well the performance of diversity is [see Table 2 (c) and Table 2 (d)] as a result of the searching process of an MOEA, so these individuals are more likely to be removed in the next generation. Therefore, by incorporating aggregation-based dominance, more selection pressure could be provided among non-dominated individuals in Pareto-based evolutionary algorithms, and enables individuals with both good convergence and diversity [see Fig. 4 (a)] to have the highest chance to survive in the next generation.

Finally, we examine the ability of ADO in dealing with ADP phenomenon caused by dominance resistant solutions. Fig. 5 gives the comparative results of the NSGA-II-SIP (NSGA-II based on SIP framework [16]) and NSGA-II-ADO (NSGA-II embedded with ADO) by plotting the changes of the number of individuals in the first front in each generation in one run. Compared to NSGA-II-SIP that all individuals of the population belong to the first front after few generations, the individuals belong to the first front in NSGA-II-ADO continuously change in the evolutionary process. This could be fully attributed to the incorporation of the proposed fitness assignment strategy, which can provide more selection pressure during the evolutionary process. Therefore, it indicates that our strategy could overcome ADP phenomenon of Pareto-based evolutionary algorithm to some extent.



**Fig. 5.** The number of individuals in the first front in each generation in one run of (a) NSGA-II-SIP (b) NSGA-II-ADO on the feature model - Drupal, respectively. Population size is 100.

### 3.3. Mating selection with constraint handling

In mating selection, a binary tournament selection strategy based on the dominance relation and density information is often used to select promising individuals from the current population for reproduction. In NSGA-II-ADO, we enhance selection operators by considering four criteria, namely the number of constraint violations (NCV), Pareto dominance, aggregation-based dominance, and crowding distance. Because the search space for SPLs includes a large number of constraints, our constraint-handling strategy considers optimal feature selection for SPLs as a constrained MOP, we introduce a constraint handling strategy by taking into account NCV as the first criterion to check. In addition, we consider the convergence (evaluated by Pareto dominance and aggregation-based dominance) and diversity information (evaluated by crowding distance) of individuals. Algorithm 2 gives a detailed procedure of this binary tournament selection in NSGA-II-ADO.

In the binary tournament mating pool, two individuals are randomly selected from current population  $P$  (lines 1–3). If an individual has fewer constraint violations than the other, then the former is chosen (lines 4–7). For individuals that have the same number of constraints, then the one that dominates the other is preferred (lines 8–12). If the two individuals are non-dominated with respect to each other, the one that ADO-dominates the other wins the tournament (lines 13–17). If these both individuals are ADO-nondominated to each other, then crowding distance is used for comparison (lines 18–22). The individual that with the larger crowding distance is chosen, because a high distance to the closest neighbors indicates the individual is located in a sparse area. When crowding distance still cannot distinguish between the two individuals, one of them will be randomly chosen for reproduction.

### 3.4. Environmental selection with constraint handling

Environmental selection preserves a group of best solutions in the population found so far as the parent population of the next generation. The proposed environmental selection process in NSGA-II-ADO performs a three-level-sorting method which is similar to non-dominated sorting in NSGA-II. But here the number of constraint violations becomes the primary criterion, followed by Pareto-dominance and proposed ADO in Section 3.1.3.

Algorithm 3 illustrates the framework of environmental selection procedure in NSGA-II-ADO for optimal feature selection in SPL problem. First, the combined population set  $S$  is normalized and the aggregation value of each individual in  $S$  is calculated by the function normalization (line 2). The normalization process is described in Section 3.1.1. Then three-level-sorting is performed to divide set  $S$  into  $l$  fronts,  $F_i$ ,  $1 \leq i \leq l$  (line 3) according to three criteria: the number of constraint violations, Pareto dominance, and aggregation-based dominance.

---

**Algorithm 2** Binary\_tournament\_selection( $P$ )

---

**Require:**  $P$  (the current population),  $M$  (population size)

```
1:  $Q \leftarrow \emptyset$  /* new population set for crossover and mutation*/
2: while  $|Q| < M$  do
3:   randomly choose  $a$  and  $b$  from  $P$ 
4:   if  $NCV(a) < NCV(b)$  then
5:      $Q \leftarrow Q \cup \{a\}$ 
6:   else if  $NCV(b) < NCV(a)$  then
7:      $Q \leftarrow Q \cup \{b\}$ 
8:   else
9:     if  $a \prec b$  then
10:       $Q \leftarrow Q \cup \{a\}$ 
11:     else if  $b \prec a$  then
12:        $Q \leftarrow Q \cup \{b\}$ 
13:     else
14:       if  $a \prec_{ADO} b$  then
15:          $Q \leftarrow Q \cup \{a\}$ 
16:       else if  $b \prec_{ADO} a$  then
17:          $Q \leftarrow Q \cup \{b\}$ 
18:       else
19:         if  $crowd\_distance(a) > crowd\_distance(b)$  then
20:            $Q \leftarrow Q \cup \{a\}$ 
21:         else if  $crowd\_distance(b) > crowd\_distance(a)$  then
22:            $Q \leftarrow Q \cup \{b\}$ 
23:         else
24:           if  $random(0, 1) < 0.5$  then
25:              $Q \leftarrow Q \cup \{a\}$ 
26:           else
27:              $Q \leftarrow Q \cup \{b\}$ 
28:           end if
29:         end if
30:       end if
31:     end if
32:   end if
33: end while
34: return  $Q$ 
```

---

---

**Algorithm 3** Environmental\_selection( $S, P, Q$ )

---

**Require:**  $S$ , the union of  $P$  (parent population) and  $Q$  (offspring population),  $M$  (population size)

```
1:  $Q \leftarrow \emptyset$  /* parent population for next generation */
2: Normalization( $S$ )
3:  $(F_1, F_2, \dots, F_l) = \text{three-level-sorting}(S)$ 
4: while  $|Q| + |F_i| \leq M$  do
5:    $Q = Q \cup F_i$  /* add  $i$ -th front in the parent population  $Q$  */
6:    $i = i + 1$  /* check the next front for inclusion */
7: end while
8: if  $|Q| = M$  then
9:   return  $Q$ 
10:   last front to be included  $F_c = F_i$ 
11: else if  $(|Q| + |F_i| > M)$  then
12:    $\text{sort}(F_c, \prec_n)$  /* sort front  $F_c$  in descending order using  $\prec_n$  */
13:    $Q = Q \cup F_i[1 : (M - |Q|)]$  /* add first  $(M - |Q|)$  individuals from  $F_c$  */
14: end if
15: return  $Q$ 
```

---

The three-level-sorting could be seen as a complex non-dominated sorting proposed in NSGA-II. It works as follows: (1) the population set  $S$  is sorted according to the number of constraint violations, and  $S$  is divided into different fronts, where each front contains the same number of constraint violations. Thus, the one with fewer constraint violations belongs to the first layer. Those with the second fewest number of violated constraints belong to the second layer, and so on. The purpose of this is to quickly drive the population to feasible regions, namely finding valid feature configurations (or valid solutions) and keeping those valid solutions in the next generation. (2) For those individuals with the same number of violated constraints in front  $F_a$ , Pareto-dominance criterion is activated to further classify front  $F_a$  into  $n$  non-dominated fronts ( $F'_a, F'_{a+1}, \dots, F'_{a+n}$ ). (3) For those non-dominated individuals in new front  $F'_a$ , aggregation-based dominance is applied to further distinguish them and divide front  $F'_a$  into  $m$  ADO-nondominated fronts ( $F_a'', F_{a+1}'', \dots, F_{a+m}''$ ). This step could help to prevent potential worse solutions (dominance resistant solutions) stay for longer generations, while for those individuals with better convergence will have more chances to survive. By performing the three-level-sorting,  $l$  fronts ( $F_1, F_2, \dots, F_l$ ) are formed for population set  $S$ . This strategy is important in evolutionary process, since it aims to generate more pressure towards convergence of a population, while handling the constraints included in SPLs at the same time.

After sorting, individuals belong to the first front ( $F_1$ ) are the most promising individuals in  $S$ . If the size of  $F_1$ , denoted as  $|F_1|$  is equal to population size  $M$ , all individuals in  $F_1$  are chosen to form parent population  $Q$  (lines 4-5 and 7-9). In case that  $|F_1|$  is smaller than population size  $M$ , the remaining individuals are chosen from the subsequent fronts according to the front rank. That is, NSGA-II-ADO turns to the second front ( $F_2$ ) for choosing the remaining  $(M - |F_1|)$  parent individuals in  $Q$ , followed by adding

the individuals from the third front ( $F_3$ ),  $C$ -th front ( $F_c$ ) and so on (lines 4-9). By doing this, individuals with the lower number of constraint violations would always be included first to form the new population. Thus, we could quickly search valid individuals (product configurations in a SPL).

If the number of individuals in the last included front  $F_c$  denoted as  $|F_c|$  is larger than  $M - |Q|$ , then a crowded-comparison operator denoted as  $\prec_n$  (the same in NSGA-II) is applied to sort individuals in  $|F_c|$  in descending order (lines 10-12). After that, the first  $(M - |Q|)$  individuals in sorted  $F_c$  are chosen to fill up the parent population set (lines 13-15). Then, the parent population set will be used for mating selection, crossover and mutation to create new generation.

## 4. Experimental results

In this section, we test the improvement for Pareto-based evolutionary algorithms by using three enhanced evolution operators. Therefore the proposed NSGA-II-ADO and SPEA2+SDE-ADO are compared with four state-of-the-art ShrInk Prioritize (SIP) framework [16] based MOEAs for optimal feature selection problems in SPLs. We first introduce the experimental design, which includes the description of competitor algorithms, subject models and optimization objectives, the basic parameter settings and system description, and performance metrics. Next, we evaluate the performance of ADO in Pareto-based evolutionary algorithms by comparing NSGA-II-ADO and SPEA2+SDE-ADO with four different types of MOEAs based on SIP framework. Finally, we discuss our findings on time comparison, hypervolume comparison, and parameter sensitivity.

### 4.1. Experimental Design

#### 4.1.1. SPL feature model

The characteristics of nine SPL feature models used in this empirical study are summarized in Table 3. For each feature model, it presents the number of features (Total features), the number of fixed features (Fixed features), namely core and parents features with the mandatory or alternative relationship, the number of Cross-Tree Constraints (CTC), and the number of optimization objectives (Objectives).

These SPL feature models were obtained from the recently published SPL literature. The *BerkeleyDB* model describes the variability of a database system. *ERS* is a feature model for an Emergency Response System. Two feature models *Web Portal* and *E-Shop* are from online SPLOT repository. It is widely used by researchers as a benchmark on optimal product selection for SPLs. Web portal is the feature model for a web portal product line. *E-Shop* is the largest feature model in SPLOT with 290 features. *Drupal* and *Amazon EC2* are two recently published feature models, which represent the variability of open-source web content management framework and Amazon Elastic Computing Service, respectively. The *Random-10000* model is a randomly generated feature model with 10,000 features, which is the largest model in our experiment

**Table 3**

SPL feature models summary.

Feature models	# Total features	#Fixed features	#CTC	#Objectives
BerkeleyDB [53]	13	3	0	4
ERS [54]	36	11	0	4
Web Portal [55]	43	15	6	4
E-Shop [56]	290	88	21	4
Drupal [57]	48	9	21	4
Amazon EC2 [58]	79	20	0	4
Random-10000 [23]	10,000	4,078	0	4
DrupalReal [57]	48	9	21	7
AmazonEC2Real [58]	79	20	0	7

to evaluate the scalability of proposed ADO in Pareto-based evolutionary algorithms. Two feature models *DrupalReal* and *AmazonEC2Real* are derived versions of *Drupal* and *Amazon EC2* with realistic attribute values, respectively.

*Feature models with four objectives.* In Table 3, for those feature models with 4 optimization objectives, the following 2 optimization objectives are considered to be maximized:

- *Richness of features.* How many features that are selected in a configuration.
- *Features that were used before.* How many features that are not used before in a configuration.

In addition, 2 objectives should be minimized:

- *Known defects.* How many known defects in a configuration.
- *Cost.* The total cost of a configuration.

*DrupalReal with seven objectives.* The following 3 optimization objectives should be maximized:

- *The richness of features.* How many features that are selected in a configuration.
- *Test assertions.* How many test assertions of each feature.
- *The number of reported installations.* How many times a feature has been installed as reported by Drupal users.

In addition, 4 objectives should be minimized:

- *The number of lines of code.*
- *Cyclomatic complexity.* How many independent logic paths used in a program.

- *The number of developers.* How many developers is involved in the development of each DrupalReal feature.
- *The number of changes.* How many changes are made in each feature.

*AmazonEC2Real with seven objectives.* 5 optimization objectives should be maximized:

- *The richness of features.* How many features that are selected in a configuration.
- *Instance.cores.* How many cores of the instance. Randomly generated value between 1 and 32.
- *Instance.ecu.* The Amazon EC2 Compute Unites. Randomly generated value between 0 and 108.
- *Instance.ram.* The memory of the instance. Randomly generated value between 0 and 250.
- *Instance.ssBacked.* If the instance storage is SSD backed. Boolean.

In addition, 2 objectives should be minimized:

- *EC2.costMonth.* Randomly generated value between 0 and 20,000.
- *Instance.costHour.* Randomly generated value between 0 and 18.

Finally, many recent studies also consider correctness (the number of violated constraints in a configuration) as an optimization objective that should be minimized. However, in our experiment, correctness is used to handle constrained search space of feature models rather than an objective (see more explanations in Sections 3.3 and 3.4).

#### 4.1.2. Competitor algorithms and parameter settings

Since no clear winner of MOEAs using the SIP framework was reported in [16], we choose four typical MOEAs using the SIP framework, denoted as NSGA-II-SIP, SPEA2+SDE-SIP, IBEA-SIP, MOEA/D-TCH-SIP. In particular, SPEA2+SDE [31] applies the Shift-based Density Estimation (SDE) scheme into the fitness assignment and archive truncation in the environmental selection of SPEA2. As a general modification of the diversity maintenance method, the SDE scheme aims to make Pareto-based algorithms suitable for MOPs. IBEA [20] is a representative indicator-based MOEA that employs a performance metric for measuring the overall quality of a population set, rather than Pareto-dominance criterion and diversity estimation criterion used in Pareto-based evolutionary algorithms. MOEA/D-TCH [24] is an aggregation-based algorithm, which decomposes a MOP into a number of single-objective optimization subproblems through a Tchebycheff aggregation function and optimizes these subproblems in parallel.

To facilitate a fair comparison with the state-of-the-art for solving optimal feature selection problem in SPLs, we kept the same settings as Hierons et al. [16]. Settings for all the MOEAs are:

- 30 runs for each algorithms per feature model to decrease the impact of their stochastic nature.
- The termination criterion is a predefined maximum of 50,000 evaluations.
- For crossover and mutation operators, uniform crossover and bit-flip mutation were used, with crossover and mutation probability set to 1.0 and  $1/n$  (where  $n$  represents the number of decision variables) respectively.

There are some special configurations specific to certain MOEAs.

- The size of population for NSGA-II, SPEA2+SDE, and IBEA, is set to 100, while the size of population for MOEA/D-TCH is set to 126 and 120 for 4- and 7-objective optimization problems, respectively.
- We set the scaling factor  $k$  in IBEA to 0.05 as suggested in the original paper [20], and the neighborhood size to 10 percent of the population size in MOEA/D-TCH as suggested in the original paper [24].
- The parameter used in our proposed aggregation-based dominance is set to 0.5 for both NSGA-II-ADO and SPEA2+SDE-ADO.

All algorithms were implemented in *C* and all the experiments were conducted on an Intel(R) Core(TM) i5-2500 CPU @ 3.30GHz with 4GB RAM, running on Windows 7.

#### 4.1.3. Performance assessment

Algorithms performance is assessed by four performance metrics.

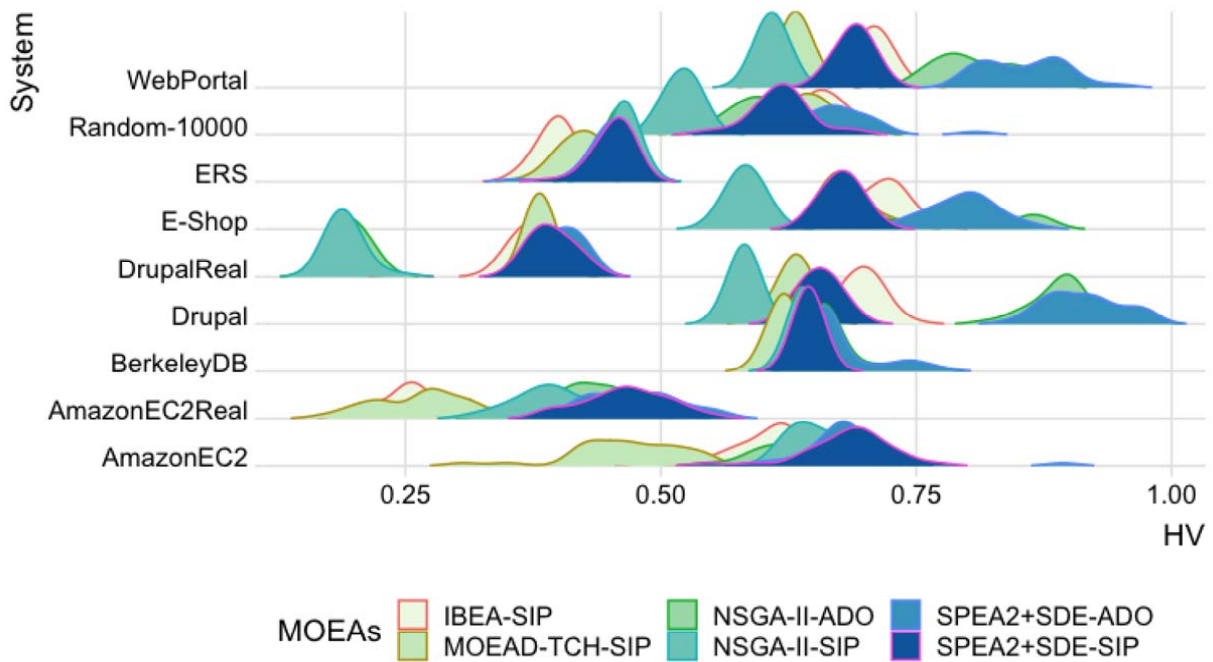
- (1) VN - the number of runs for each algorithm that returns at least one valid solution (i.e., without violating any constraints) in the final solution set [16]. This is used to evaluate the ability of an algorithm to find valid products from a feature model. Runs with invalid solutions are discarded from our study, since invalid product configuration is useless in practice.
- (2) VP - the proportion of valid distinct solutions in the final population [16]. A high value of VP is preferred, as more options are provided to software engineers.
- (3) TT100% - the time to reach TT100% valid solutions in a population. This quality indicator was first introduced by Sayyad et al. [13]. It measures the speed of convergence to a large number of points within the Pareto front that have no violations. It will be calculated when the VP values for different algorithms are the same.
- (4) Hypervolume (HV) is a popular performance metric in evolutionary multi-objective optimization area. HV evaluates how well a Pareto front fulfils the optimization objectives in terms of convergence and diversity, which is commonly used in the EMO field. Let  $R = (r_1, r_2, \dots, r_n)^T$  be a reference point in the objective space that is dominated by all the solutions on the Pareto front. As defined in [52], HV



metric measures the volume of the objective space that is dominated by solutions in the Pareto front  $A$  and bounded by  $R$ , mathematically  $HV(A) = V(U_{X \in A} [f_1(x), r_1] \times [f_2(x), r_2] \times \dots \times [f_n(x), r_n])$  where  $n$  is the number of objectives,  $V(S)$  is the Lebesgue measure [59] of a set  $S$ ,  $F = (f_1(x), f_2(x), \dots, f_n(x))$  is an objective vector in  $A$ . In our experiments, all the objective values of solutions were normalized to  $[0, 1]$  according to different ranges of a particular objective. In addition, the reference point  $R$  is set to  $(1.0, 1.0, \dots, 1.0)^T$ , which is constructed with the worst value on each objective, so-called Nadir point. A larger HV value is preferable because it reflects better quality of  $A$  in terms of convergence, diversity, and uniformity. Finally, only those final solution sets with VP of 100% are used to calculate HV.

#### 4.2. Performance comparison

To provide an overall picture of the relative performance of the six different MOEAs across the nine different feature models (systems) we represent the hyper-volume (HV) results as joy plots<sup>1</sup> in Fig. 6. Recall that each algorithm is applied 30 times to a particular system model and the distribution of results is represented by a kernel density approximation. Thus for each row or system there are six distributions representing the performance of each MOEA.



**Fig. 6.** Kernel Density Plots of MOEA performances in terms of Hypervolume by System Feature Model

<sup>1</sup>Joy plots are superimposed kernel density plots and are so-called in homage to the iconic 1979 album cover for "Unknown Pleasures" by the British band Joy Division. We used the R package `ggridges` courtesy of Claus Wilke.

The first observation is there is a good deal more variability between system models than there is between algorithms. In other words some models are fundamentally more difficult to optimise than others. Second, the performance of individual algorithms are not well represented by a single point value e.g., mean or median given the dispersion of performance results. As an extreme example consider MOEAD-TCH-SIP for AmazonEC2. From a practical point of view this means the stochastic nature of the algorithms leads to considerable variability in performance. Third, there is no one algorithm that is always best and there is some interaction between algorithm and system<sup>2</sup>. However, we can see that in general SPEA2+SDE-ADO provides competitive performance.

Table 4 shows the same results quantitatively, in terms of VN ( $\setminus 30$ ), VP, TT100%, and the mean and standard deviation of HV metric, respectively. For each test feature model, among different algorithms, the algorithm that has the best result based on the specific metric is shown in bold.

As shown in Table 4, almost all algorithms could obtain valid solutions (VN of 30 and VP of 100%) for all feature models (both with 4 and 7 objectives). One exception is IBEA-SIP, which only returned valid solutions with a VP value of 100% in 12/30 runs for AmazonEC2Real. In addition, none of the algorithms could return valid solutions with a VN value of 30 for Random-10000. SPEA2+SDE-ADO returned the highest number of successful runs (VN = 29), followed by NSGA-II-ADO and SPEA2+SDE-SIP (VN = 27), IBEA-SIP (VN = 25), NSGA-II-SIP (VN = 19). MOEA/D-TCH-SIP returned the lowest number of successful runs (VN = 14).

Because of the similar good performance in terms of VN and VP, we further compare the execution time of these algorithms using the TT100% metric. MOEA/D-TCH-SIP is shown to be the fastest algorithm for 7 of the 9 feature models. NSGA-II-SIP is the second fastest algorithm for 8 of the 9 feature models and is the fastest algorithm for AmazonEC2Real. Note that NSGA-II-ADO is slower than NSGA-II-SIP for almost all feature models but is the fastest algorithm for the largest feature model Random-10000, which only takes under 29 seconds. In addition, although the performance of SPEA2+SDE-ADO in terms of TT100% metric is poorer than that of its competitors, except for IBEA-SIP, the longest execution time of SPEA2+SDE-ADO is under 37 seconds for the largest feature model in our study. This indicates the proposed ADO is an efficient method in searching valid product configurations in SPLs.

Next, we investigate the quality of obtained solutions by comparing the HV metric values of algorithms. As shown in Table 4, the proposed SPEA2+SDE-ADO is the most effective tested algorithm with regard to the best results in 8 test feature models out of 9 (apart from ERS). The proposed NSGA-II-ADO performs slightly worse than SPEA2+SDE-ADO on feature models with four objectives (BerkeleyDB, WebPortal, E-Shop, Drupal), but significantly better than the four SIP-based MOEAs. SPEA2+SDE-SIP achieves the second best result for three feature models: Amazon EC2 (with four objectives), DrupalReal (with seven

---

<sup>2</sup>This is confirmed by a robust 2-way ANOVA that finds a non-trivial interaction term.

**Table 4**

Mean of VN ( $\setminus 30$ ), VR, TT100%, and mean and standard deviation of HV metric on nine test feature models. The best result for each feature model is highlighted in boldface.

Feature model	Metric	NSGA-II-ADO	SPEA2+SDE-ADO	NSGA-II-SIP	SPEA2+SDE-SIP	IBEA-SIP	MOEA/D-TCH-SIP
BerkeleyDB	VN( $\setminus 30$ )	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>
	TT100% (ms)	2.2	23.7	1.8	20.9	40.4	<b>0.8</b>
	HV	0.6730	<b>0.6773</b>	0.6394	0.6445	0.6381	0.6203
	Std. Dev.	0.0297	0.0344	0.0021	0	0.0012	0.0072
ERS	VN( $\setminus 30$ )	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>
	TT100% (ms)	48.1	566.8	44.0	501.8	1018.4	<b>6.0</b>
	HV	0.4596	0.4488	<b>0.4643</b>	0.4531	0.3964	0.4220
	Std. Dev.	0.0095	0.0223	0.0056	0.0173	0.0143	0.0211
WebPortal	VN( $\setminus 30$ )	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>
	TT100% (ms)	9.5	107.0	9.0	101.9	197.9	<b>2.4</b>
	HV	0.8067	<b>0.8518</b>	0.6087	0.6894	0.7067	0.6326
	Std. Dev.	0.0369	0.0469	0.0085	0.0144	0.016	0.0089
E-Shop	VN( $\setminus 30$ )	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>
	TT100% (ms)	52.4	257.4	45.1	245.5	485.4	<b>22.8</b>
	HV	0.7657	<b>0.7882</b>	0.5828	0.6770	0.7180	0.6812
	Std. Dev.	0.0695	0.0445	0.0144	0.0169	0.0225	0.0218
Drupal	VN( $\setminus 30$ )	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>
	TT100% (ms)	12.6	108.7	10.3	100.8	204.3	<b>2.6</b>
	HV	0.8952	<b>0.9136</b>	0.5815	0.6554	0.6976	0.6310
	Std. Dev.	0.0332	0.0367	0.0063	0.0175	0.0179	0.0119
AmazonEC2	VN( $\setminus 30$ )	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>
	TT100% (ms)	76.2	671.9	72.6	628.3	1169.6	<b>24.2</b>
	HV	0.6451	<b>0.6862</b>	0.6518	0.6829	0.6082	0.4657
	Std. Dev.	0.0404	0.0498	0.0252	0.0439	0.0401	0.0548
Random-10000	VN( $\setminus 30$ )	27	<b>29</b>	19	27	25	14
	TT100% (ms)	<b>28653.2</b>	36081.4	29438.3	35982.3	39675.2	37225.3
	HV	0.6068	<b>0.6546</b>	0.5211	0.6172	0.6485	0.6247
	Std. Dev.	0.0391	0.0513	0.0122	0.0291	0.0324	0.0418
DrupalReal	VN( $\setminus 30$ )	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>
	TT100% (ms)	13.4	122.1	11.8	113.7	1080.1	<b>2.9</b>
	HV	0.1943	<b>0.4008</b>	0.1912	0.3931	0.3688	0.3812
	Std. Dev.	0.0159	0.0198	0.016	0.02	0.0283	0.0037
AmazonEC2Real	VN( $\setminus 30$ )	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>	12	<b>30</b>
	TT100% (ms)	109.6	1305.6	<b>91.4</b>	841.8	11998.2	213.8
	HV	0.4233	<b>0.4730</b>	0.3992	0.4654	0.2820	0.2694
	Std. Dev.	0.0353	0.0451	0.0448	0.042	0.0632	0.0512

Note: The value of VP metric is 100% for all algorithms in 30 independent runs.

objectives), and AmazonEC2Real (with seven objectives). Although the performance of NSGA-II-SIP and IBEA-SIP are worse than that of their competitors, they perform the best on ERS and the second best on the largest feature model Random-10000, respectively. For MOEA/D-TCH-SIP, there is no clear advantage over its competitors.

Table 5 summarizes the HV differences between NSGA-II-ADO and NSGA-II-SIP, and the differences between SPEA2+SDE-ADO and SPEA2+SDE-SIP on nine feature models. Positive values of HV indicate improvements, while negative values of HV indicate the opposite. As seen in Table 4 and 5, the proposed NSGA-II-ADO performs better than NSGA-II-SIP in 7 out of 9 tested feature models regarding the HV metric with. The best performance is on Drupal with an increase of 53.96%, followed by WebPortal (32.52%), E-Shop (31.39%), and Random-10000 (16.44%). For feature models with seven objectives, NSGA-II-ADO could also obtain better results with 1.62% and 6.03% improvement for DrupalReal and AmazonEC2Real, respectively. Although there is a decrease of 0.99% on ERS and 1.03% on Amazon EC2, it could be seen as similar results due to the randomness. In addition, the proposed SPEA2+SDE-ADO performs better than SPEA2+SDE-SIP in 8 out of 9 tested feature models regarding the HV metric. The largest percent of HV increase is 39.38% and only a very small deterioration (0.94%) on ERS.

**Table 5**

HV differences between proposed ADO based algorithm and SIP framework based algorithm on nine feature models (i.e.,  $(HV(NSGA-II-ADO)/HV(NSGA-II-SIP) \times 100\% - 1)$ )

Feature models	NSGA-II (%)	SPEA2+SDE (%)
BerkeleyDB (4)	5.25	5.10
ERS (4)	-0.99	-0.94
WebPortal (4)	32.52	23.55
E-Shop (4)	31.39	16.43
Drupal (4)	53.96	39.38
Amazon EC2 (4)	-1.03	0.48
Random-10000 (4)	16.44	6.06
DrupalReal (7)	1.62	1.96
AmazonEC2Real (7)	6.03	1.63

In summary, compared to SIP, the ADO can help NSGA-II and SPEA2+SDE to generally improve the quality of obtained solutions for 7 and 8 out of 9 feature models in this experiment, respectively.

In Fig. ?? (a) – Fig. 7 (i), the distribution of the HV values on the 30 runs of all algorithms (NSGA-II-SIP, NSGA-II-ADO, SPEA2+SDE-SIP, SPEA2+SDE-ADO, IBEA-SIP, and MOEA/D-TCH-SIP) is plotted. Our results indicate that the ADO has clear advantages over SIP for the nine experimental subjects, ADO can largely improve the performance of Pareto-based algorithms to find high-quality solutions. Compared to NSGA-II and SPEA2+SDE based on SIP framework, using ADO could find high-quality solutions, which is evident from the HV values shown in Table 4. In particular, SPEA2+SDE-ADO is the most effective MOEA for three feature models that proved to be most challenging for search, including those with a larger number

of objectives (Drupal and Amazon with realistic attribute values) and the larger randomly generated model (Random-10000).



**Fig. 7.** Hypervolume value comparison of six algorithms on the nine feature models

### 4.3. Parameter sensitivity analysis

To evaluate the sensitivity of the parameter  $\alpha$  in the proposed aggregation-based Pareto dominance, we investigate how much the HV value fluctuates with the increase of the value of parameter  $\alpha$  in the pre-defined intervals. We perform the experiments with NSGA-II-ADO on six feature models (BerkeleyDB, ERS, WebPortal, E-Shop, Drupal, and Amazon EC2) with four objectives. HV is chosen as a performance metric since it reflects a good balance between convergence and diversity.

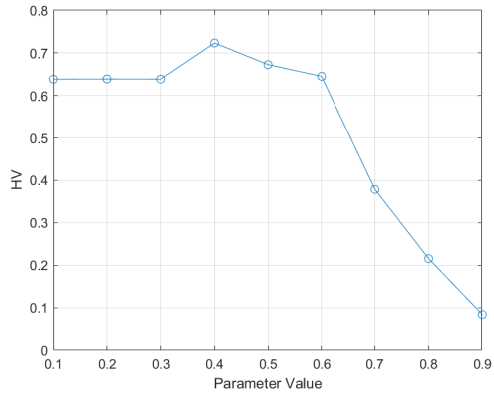
In Fig. 8, relative changes of the HV values corresponding to the increases of  $\alpha$  from 0.1 to 0.9 with a step of 0.1 for 30 independent runs and 50,000 evaluations are plotted.

In Fig. 8 (a) – Fig. 8 (f), almost all of the feature models reach roughly highest HV values when the parameter  $\alpha$  value is between 0.5 – 0.6, except for BerkeleyDB in Fig. 8 (a) with the value of  $\alpha$  is 0.4, and E-Shop in Fig. 8 (d) with the value of  $\alpha$  is 0.8. In addition, for most of the feature models, the average of HV values is sharply reduced when the value of  $\alpha$  exceeds 0.6. However, there is one exception E-Shop, as shown in Fig. 8 (d), with its average HV value rising significantly with the  $\alpha$  value grows from 0.3 to 0.8, followed by a significant decrease. WebPortal, as shown in Fig. 8 (c) and Drupal, as shown in Fig. 8 (e) have similar performance in that their average HV values change little with the increase of  $\alpha$  from 0.1 to 0.3, and then a significant improvement of HV values with the increase of  $\alpha$  from 0.3 to 0.5, which followed by a little fluctuate between range  $\alpha$  from 0.5 to 0.6, and finally a significant decrease between range  $\alpha$  from 0.6 to 0.9.

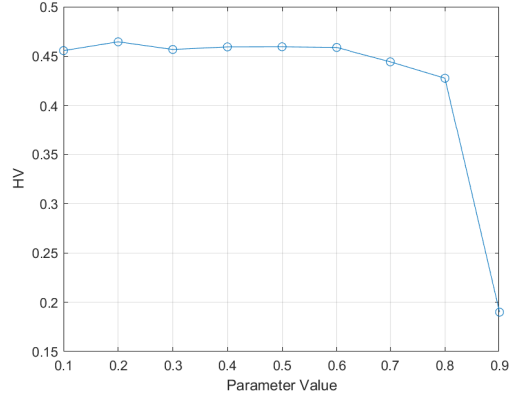
To sum up, the setting of parameter  $\alpha$  in ADO is determined based on the experiment results with two factors - the performance metrics and SPLs taken into account. In particular, the performance of the NSGA-II-ADO algorithm are measured by hypervolume metric on six different feature models with up to 290 features in 30 independent runs. In general, we advise that  $\alpha$  should be set within a range of 0.5 to 0.6. This is because small values have little influence on the final solutions set, while large value may lead to a subpopulation set that is limited in local optimal regions of the Pareto front. Although the HV value fluctuates a little between 0.5 and 0.6, the changes are small and the sensitivities are at acceptable levels. NSGA-II-ADO seems robust on the parameter  $\alpha$  in the suggested ranges. High-quality solutions with regard to HV metric obtained by NSGA-II-ADO do not fluctuate much with the value of parameter  $\alpha$  changes in the suggested range.

## 5. Conclusion

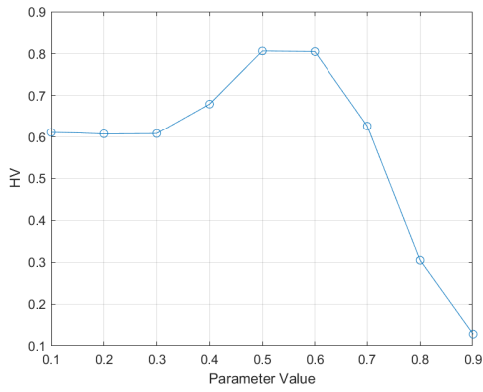
In this paper, we address the two main difficulties of using Pareto-based evolutionary algorithms to solve optimal feature selection in software product lines, which can be seen as a multi-objective optimization problem with constrained search space. The two difficulties are the ineffectiveness of Pareto dominance relations and weakness of evolutionary operators.



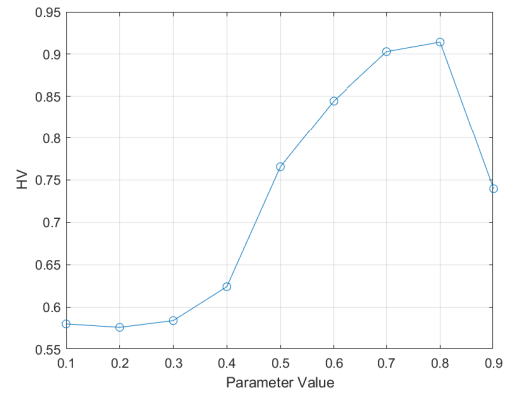
(a) BerkeleyDB



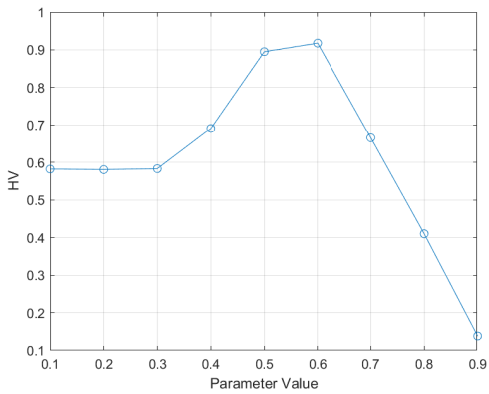
(b) ERS



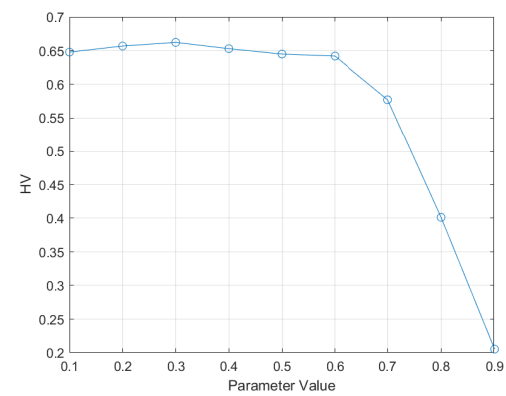
(c) WebPortal



(d) E-Shop



(e) Drupal



(f) Amazon EC2

**Fig. 8.** The curves of HV with regard to  $\alpha$  values varying from 0.1 to 0.9 with a step of 0.1. The value of HV for each feature model is an average value of 30 independent runs generated by proposed NSGA-II-ADO.

Some studies have observed that *active diversity promotion* phenomenon is the main reason for the first challenge in traditional Pareto-based evolutionary algorithms for multi-objective optimization problems. The active diversity promotion phenomenon is caused by the preference of dominance resistant solutions in a population. The basic idea of our *aggregation-based dominance* (ADO) is to avoid the active diversity promotion phenomenon by removing dominance resistant solutions from populations in a run of Pareto-based evolutionary algorithms. ADO is inspired by the effectiveness of aggregation methods and the idea of using additional convergence-related criteria in addition to traditional Pareto dominance based criterion in recent studies. In ADO, a smaller aggregation value of an individual indicates a good convergence performance. If an individual is a dominance resistant solution, it is more likely to obtain an extremely large aggregation value compared to other individuals in a population and to be ADO-dominated by other solutions. In this way, these dominance resistant solutions can be distinguished.

For the second challenge, we enhance the effectiveness of three evolutionary operators – fitness assignment, mating selection, and environmental selection by incorporating ADO. In these evolutionary operators, dominance resistant solutions will have less chance to survive and therefore provide additional selective pressure to Pareto front. For both mating and environmental selection, a constraint handling strategy is introduced to quickly find valid products in constrained search space of software product lines by continuously reducing the infeasible search space in the evolutionary process.

Our experiments are conducted on two widely used Pareto-based evolutionary algorithms, NSGA-II and SPEA2+SDE for nine feature models. The two algorithms are integrated with ADO, or simply NSGA-II-ADO and SPEA2+SDE-ADO and their performances are compared with four multi-objective evolutionary algorithms based on state-of-the-art SIP framework for SPLs. Our experiments indicate that both NSGA-II-ADO and SPEA2+SDE-ADO could generate 100% valid solutions in no more than a second and under 40 seconds for the largest tested feature model (Random-10000) in the worst case.

Furthermore, the quality of found solutions (in terms of hypervolume metric) is improved by 53.96% and 39.38% in the best case by integrating ADO into NSGA-II and SPEA2+SDE on nine tested feature models compared to SIP, respectively. In particular, SPEA2+SDE-ADO is the most effective algorithm that outperforms other tested MOEAs for 8 out of 9 test feature models (apart from ERS).

In summary, we have inherited strengths from existing Pareto-based evolutionary algorithms while managing to address the weakness in dealing with complex SPL selection optimization problems. In particular, ADO is an effective method that is capable of accelerating the convergence information in order to enhance the selection pressure towards the Pareto front by reducing the detrimental impact caused by the active diversity promotion phenomenon.

In the future, we plan to investigate the influence of parameter ( $\alpha$ ) setting in ADO in more Pareto-based evolutionary algorithms. Furthermore, we will address the scalability of our method in terms of the size of feature models. In this empirical study, there is only one large feature model out of nine feature models,



but more complex and large feature models could be investigated, e.g., LVAT repositories with real-world feature models including the aforementioned Linux X86 kernel model with 6,888 features. Moreover, we plan to investigate more Pareto-based evolutionary algorithms by using the proposed ADO and understand its characteristics in more depth. Finally, we attempt to apply ADO to some other interesting areas related to optimal feature selection, such as data classification in medicine or business [60, 61].

## References

- [1] D. Batory, Feature Models, Grammars, and Propositional Formulas, in: International Conference on Software Product Lines, Springer, Berlin, Heidelberg, 2005, pp. 7–20. doi:10.1007/11554844\_3.
- [2] M. Griss, Software reuse architecture, process, and organization for business success, in: Proceedings of the Eighth Israeli Conference on Computer Systems and Software Engineering, IEEE, 2002, pp. 86–89. doi:10.1109/ICCSSE.1997.599879.
- [3] P. Clements, L. Northrop, Software Product Lines: Practices and Patterns, Addison Wesley, 2001.
- [4] Linux Variability Analysis Tools (LVAT) Repository, <https://code.google.com/archive/p/linux-variability-analysis-tools/>, n.d. Accessed: Feb. 15, 2019.
- [5] J. Guo, J. White, G. Wang, J. Li, Y. Wang, A genetic algorithm for optimized feature selection with resource constraints in software product lines, *Journal of Systems and Software* 84 (2011) 2208–2221. doi:10.1016/j.jss.2011.06.026.
- [6] A. S. Sayyad, T. Menzies, H. Ammar, On the value of user preferences in search-based software engineering: A case study in software product lines, in: 2013 35th International Conference on Software Engineering (ICSE), IEEE, 2013, pp. 492–501. doi:10.1109/ICSE.2013.6606595.
- [7] N. Zeng, Z. Wang, Y. Li, M. Du, X. Liu, A Hybrid EKF and Switching PSO Algorithm for Joint State and Parameter Estimation of Lateral Flow Immunoassay Models, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 9 (2012) 321–329. doi:10.1109/TCBB.2011.140.
- [8] N. Zeng, Z. Wang, B. Zineddin, Y. Li, M. Du, L. Xiao, X. Liu, T. Young, Image-Based Quantitative Analysis of Gold Immunochromatographic Strip via Cellular Neural Network Approach, *IEEE Transactions on Medical Imaging* 33 (2014) 1129–1136. doi:10.1109/TMI.2014.2305394.
- [9] Y.-J. Zheng, H.-F. Ling, J.-Y. Xue, S.-Y. Chen, Population Classification in Fire Evacuation: A Multiobjective Particle Swarm Optimization Approach, *IEEE Transactions on Evolutionary Computation* 18 (2014) 70–81. doi:10.1109/TEVC.2013.2281396.
- [10] Z. Zhu, G. Zhang, M. Li, X. Liu, Evolutionary Multi-Objective Workflow Scheduling in Cloud, *IEEE Transactions on Parallel and Distributed Systems* 27 (2016) 1344–1357. doi:10.1109/TPDS.2015.2446459.
- [11] W. Liu, Z. Wang, X. Liu, N. Zeng, D. Bell, A Novel Particle Swarm Optimization Approach for Patient Clustering from Emergency Departments, *IEEE Transactions on Evolutionary Computation* (2018). doi:10.1109/TEVC.2018.2878536.
- [12] Y. Wu, Y. Zhang, X. Liu, Z. Cai, Y. Cai, A multiobjective optimization-based sparse extreme learning machine algorithm, *Neurocomputing* 317 (2018) 88–100. doi:10.1016/J.NEUCOM.2018.07.060.
- [13] A. S. Sayyad, J. Ingram, T. Menzies, H. Ammar, Scalable product line configuration: A straw to break the camel’s back, in: 2013 28th IEEE/ACM International Conference on Automated Software Engineering (ASE), IEEE, 2013, pp. 465–474. doi:10.1109/ASE.2013.6693104.
- [14] C. Henard, M. Papadakis, M. Harman, Y. Le Traon, Combining Multi-Objective Search and Constraint Solving for Configuring Large Software Product Lines, in: 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, IEEE, 2015, pp. 517–528. doi:10.1109/ICSE.2015.69.
- [15] T. H. Tan, Y. Xue, M. Chen, J. Sun, Y. Liu, J. S. Dong, Optimizing selection of competing features via feedback-directed

- evolutionary algorithms, in: *Proceedings of the 2015 International Symposium on Software Testing and Analysis - ISSTA 2015*, ACM Press, New York, New York, USA, 2015, pp. 246–256. doi:10.1145/2771783.2771808.
- [16] R. M. Hierons, M. Li, X. Liu, W. Zheng, S. Segura, SIP: Optimal Product Selection from Feature Models Using Many-Objective Evolutionary Optimization, *ACM Trans. Softw. Eng. Methodol* 25 (2016) 17:1–17:39. doi:10.1145/2897760.
- [17] Y. Xiang, Y. Zhou, Z. Zheng, M. Li, Configuring Software Product Lines by Combining Many-Objective Optimization and SAT Solvers, *ACM Transactions on Software Engineering and Methodology (TOSEM)* 26 (2018) 14:1–14:46. doi:10.1145/3176644.
- [18] R. Olacoea, D. Rayside, J. Guo, K. Czarnecki, Comparison of exact and approximate multi-objective optimization for software product lines, in: *Proceedings of the 18th International Software Product Line Conference - Volume 1*, New York, New York, USA, 2014, pp. 92–101. doi:10.1145/2648511.2648521.
- [19] Y. Xue, J. Zhong, T. H. Tan, Y. Liu, W. Cai, M. Chen, J. Sun, IBED: Combining IBEA and DE for optimal feature selection in software product line engineering, *Applied Soft Computing* 49 (2016) 1215–1231. doi:10.1016/J.ASOC.2016.07.040.
- [20] E. Zitzler, S. Künzli, Indicator-Based Selection in Multiobjective Search, in: *International Conference on Parallel Problem Solving from Nature*, Springer, Berlin, Heidelberg, 2004, pp. 832–842. doi:10.1007/978-3-540-30217-9\_84.
- [21] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation* 6 (2002) 182–197. doi:10.1109/4235.996017.
- [22] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength pareto evolutionary algorithm, *TIK-Report* 103 (2001). doi:10.3929/ethz-a-004284029.
- [23] M. Mendonca, M. Branco, D. Cowan, S.P.L.O.T.: Software Product Lines Online Tools, in: *Proceeding of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications - OOPSLA '09*, ACM, 2009, pp. 761–762. doi:10.1145/1639950.1640002.
- [24] Q. Zhang, H. Li, MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition, *IEEE Transactions on Evolutionary Computation* 11 (2007) 712–731. doi:10.1109/TEVC.2007.892759.
- [25] K. Kang, S. Cohen, J. Hess, W. Novak, A. Peterson, Feature-oriented domain analysis (FODA) feasibility study, *Technical Report*, Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst, 1990.
- [26] T. Thum, D. Batory, C. Kastner, Reasoning about edits to feature models, in: *2009 IEEE 31st International Conference on Software Engineering, IEEE, 2009*, pp. 254–264. doi:10.1109/ICSE.2009.5070526.
- [27] D. Benavides, S. Segura, A. Ruiz-Cortés, Automated analysis of feature models 20 years later: A literature review, *Information Systems* 35 (2010) 615–636. doi:10.1016/J.IS.2010.01.001.
- [28] N. Srinivas, K. Deb, Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms, *Evolutionary Computation* 2 (1994) 221–248. doi:10.1162/evco.1994.2.3.221.
- [29] C. M. Fonseca, P. J. Fleming, An Overview of Evolutionary Algorithms in Multiobjective Optimization, *Evolutionary Computation* 3 (1995) 1–16. doi:10.1162/evco.1995.3.1.1.
- [30] D. W. Corne, N. R. Jerram, J. D. Knowles, M. J. Oates, PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization, in: *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, 2001, pp. 283–290.
- [31] M. Li, S. Yang, X. Liu, Shift-Based Density Estimation for Pareto-Based Algorithms in Many-Objective Optimization, *IEEE Transactions on Evolutionary Computation* 18 (2014) 348–365. doi:10.1109/TEVC.2013.2262178.
- [32] H. Ishibuchi, N. Tsukamoto, Y. Nojima, Evolutionary many-objective optimization: A short review, in: *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, IEEE, 2008, pp. 2419–2426. doi:10.1109/CEC.2008.4631121.
- [33] T. Wagner, N. Beume, B. Naujoks, Pareto-, Aggregation-, and Indicator-Based Methods in Many-Objective Optimization, in: *Evolutionary Multi-Criterion Optimization*, Springer, Berlin, Heidelberg, 2007, pp. 742–756. doi:10.1007/978-3-540-70928-2\_56.

- [34] K. Ikeda, H. Kita, S. Kobayashi, Failure of Pareto-based MOEAs: does non-dominated really mean near to optimal?, in: Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), volume 2, IEEE, 2001, pp. 957–962. doi:10.1109/CEC.2001.934293.
- [35] E. Hughes, Evolutionary many-objective optimisation: many once or one many?, in: 2005 IEEE Congress on Evolutionary Computation, volume 1, IEEE, 2005, pp. 222–227. doi:10.1109/CEC.2005.1554688.
- [36] E. Hughes, Multiple single objective Pareto sampling, in: The 2003 Congress on Evolutionary Computation, 2003. CEC '03., volume 4, IEEE, 2003, pp. 2678–2684. doi:10.1109/CEC.2003.1299427.
- [37] E. J. Hughes, E. James, Many-objective directed evolutionary line search, in: Proceedings of the 13th annual conference on Genetic and evolutionary computation, ACM, 2011, pp. 761–768. doi:10.1145/2001576.2001680.
- [38] K. Deb, M. Mohan, S. Mishra, Evaluating the  $\epsilon$ -Domination Based Multi-Objective Evolutionary Algorithm for a Quick Computation of Pareto-Optimal Solutions, Evolutionary Computation 13 (2005) 501–525. doi:10.1162/106365605774666895.
- [39] Z. He, G. G. Yen, J. Zhang, Fuzzy-Based Pareto Optimality for Many-Objective Evolutionary Algorithms, IEEE Transactions on Evolutionary Computation 18 (2014) 269–285. doi:10.1109/TEVC.2013.2258025.
- [40] M. Farina, P. Amato, A fuzzy definition of "optimality" for many-criteria optimization problems, IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 34 (2004) 315–326. doi:10.1109/TSMCA.2004.824873.
- [41] F. di Pierro, S.-T. Khu, D. A. Savi, An Investigation on Preference Order Ranking Scheme for Multiobjective Evolutionary Optimization, IEEE Transactions on Evolutionary Computation 11 (2007) 17–45. doi:10.1109/TEVC.2006.876362.
- [42] M. Li, J. Zheng, R. Shen, K. Li, Q. Yuan, A grid-based fitness strategy for evolutionary many-objective optimization, in: Proceedings of the 12th annual conference on Genetic and evolutionary computation, ACM, 2010, pp. 463–470. doi:10.1145/1830483.1830570.
- [43] K. Deb, H. Jain, An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints, IEEE Transactions on Evolutionary Computation 18 (2014) 577–601. doi:10.1109/TEVC.2013.2281535.
- [44] H. Sato, H. Aguirre, K. Tanaka, Variable space diversity, crossover and mutation in MOEA solving many-objective knapsack problems, Annals of Mathematics and Artificial Intelligence 68 (2013) 197–224. doi:10.1007/s10472-012-9293-y.
- [45] J. J. Durillo, A. J. Nebro, F. Luna, E. Alba, On the Effect of the Steady-State Selection Scheme in Multi-Objective Genetic Algorithms, in: M. Ehrgott, C. M. Fonseca, X. Gandibleux, J.-K. Hao, M. Sevaux (Eds.), Evolutionary Multi-Criterion Optimization, Springer, Berlin, Heidelberg, 2009, pp. 183–197. doi:10.1007/978-3-642-01020-0\_18.
- [46] A. J. Nebro, J. J. Durillo, F. Luna, B. Dorronsoro, E. Alba, MOCcell: A cellular genetic algorithm for multiobjective optimization, International Journal of Intelligent Systems 24 (2009) 726–746. doi:10.1002/int.20358.
- [47] Y. Xiang, Y. Zhou, M. Li, Z. Chen, A Vector Angle-Based Evolutionary Algorithm for Unconstrained Many-Objective Optimization, IEEE Transactions on Evolutionary Computation 21 (2017) 131–152. doi:10.1109/TEVC.2016.2587808.
- [48] M. Li, S. Yang, X. Liu, Bi-goal evolution for many-objective optimization problems, Artificial Intelligence 228 (2015) 45–65. doi:10.1016/J.ARTINT.2015.06.007.
- [49] X. Zhang, Y. Tian, Y. Jin, A Knee Point-Driven Evolutionary Algorithm for Many-Objective Optimization, IEEE Transactions on Evolutionary Computation 19 (2015) 761–776. doi:10.1109/TEVC.2014.2378512.
- [50] S. Yang, M. Li, X. Liu, J. Zheng, A Grid-Based Evolutionary Algorithm for Many-Objective Optimization, IEEE Transactions on Evolutionary Computation 17 (2013) 721–736. doi:10.1109/TEVC.2012.2227145.
- [51] M. Laumanns, L. Thiele, K. Deb, E. Zitzler, Combining Convergence and Diversity in Evolutionary Multiobjective Optimization, Evolutionary Computation 10 (2002) 263–282. doi:10.1162/106365602760234108.
- [52] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, IEEE Transactions on Evolutionary Computation 3 (1999) 257–271. doi:10.1109/4235.797969.

- [53] N. Siegmund, M. Rosenmüller, M. Kuhlemann, C. Kästner, S. Apel, G. Saake, SPL Conqueror: Toward optimization of non-functional properties in software product lines, *Software Quality Journal* 20 (2012) 487–517. doi:10.1007/s11219-011-9152-9.
- [54] N. Esfahani, S. Malek, K. Razavi, GuideArch: Guiding the Exploration of Architectural Solution Space under Uncertainty, in: 2013 35th International Conference on Software Engineering (ICSE), IEEE Press, 2013, pp. 43–52.
- [55] M. Mendonça, T. T. Bartolomei, D. Cowan, Decision-making coordination in collaborative product configuration, in: Proceedings of the 2008 ACM symposium on Applied computing, New York, New York, USA, 2008, pp. 108–113. doi:10.1145/1363686.1363715.
- [56] S. Q. Lau, Domain Analysis of E-Commerce Systems Using Feature-Based Model Templates, Ph.D. thesis, Dept. Electrical and Computer Engineering, University of Waterloo, 2006.
- [57] A. B. Sánchez, S. Segura, J. A. Parejo, A. Ruiz-Cortés, Variability testing in the wild: the Drupal case study, *Software & Systems Modeling* 16 (2017) 173–194. doi:10.1007/s10270-015-0459-z.
- [58] J. García-Galán, P. Trinidad, O. F. Rana, A. Ruiz-Cortés, Automated configuration support for infrastructure migration to the cloud, *Future Generation Computer Systems* 55 (2016) 200–212. doi:10.1016/J.FUTURE.2015.03.006.
- [59] M. Fleischer, The Measure of Pareto Optima Applications to Multi-objective Metaheuristics, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer, Berlin, Heidelberg, 2003, pp. 519–533. doi:10.1007/3-540-36970-8\_37.
- [60] F. Jimenez, C. Martinez, E. Marzano, J. T. Palma, G. Sanchez, G. Sciavicco, Multiobjective Evolutionary Feature Selection for Fuzzy Classification, *IEEE Transactions on Fuzzy Systems* 27 (2019) 1085–1099. doi:10.1109/TFUZZ.2019.2892363.
- [61] J. González, J. Ortega, M. Damas, P. Martín-Smith, J. Q. Gan, A new multi-objective wrapper method for feature selection – Accuracy and stability analysis for BCI, *Neurocomputing* 333 (2019) 407–418. doi:10.1016/J.NEUCOM.2019.01.017.