Arabian Journal for Science and Engineering (2022) 47:10147–10162
https://doi.org/10.1007/s13369-021-06523-5

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

A Customized PSO Model for Large-Scale Many-Objective Software Package Restructuring Problem

Amarjeet Prajapati[1]

Received: 31 August 2021 / Accepted: 16 December 2021 / Published online: 6 January 2022

Abstract

Recently, a variety of large-scale many-objective optimization algorithms (LSMaOAs) have been designed and proposed to address different classes of large-scale many-objective optimization problems (LSMaOPs). Even after tremendous progress in the development of LSMaOAs for the various types of synthetic LSMaOPs, the real-world LSMaOPs such as large-scale many-objective software package restructuring (LSMaOSPR) gained little attention. This work proposes a particle swarm optimization (PSO) based LSMaOA for the LSMaOSPR problem. To this contribution, different components of PSO framework such as selection of inertia weight, selection of cognitive and social constant, updating velocity and position of particles, and determination of personal best and global best are customized based on the suitability of the LSMaOSPR characteristics. To evaluate the supremacy of the proposed approach, we tested it over five LSMaOSPR problems. The optimization results indicate that the proposed LSMaOA approach has enough capability for generating an evenly distributed and well-converged approximation of the Pareto front for the large and complex LSMaOSPR problems.

Keywords Software package restructuring · Many-objective optimization · Large-scale optimization · PSO algorithm

1 Introduction

The size of multi-featured modern software systems is mostly very large that generally makes program complex to understand and difficult to maintain. To reduce the complexity, developers usually employ object-oriented concepts to design and implement the software systems. In the object-oriented system, the low-level software entities are distributed into the high-level software entities based on the various object-oriented design principles and guidelines. Even though, the applications of object-oriented concepts help in reducing the complexity, still improper use of its design guidelines can increase the software complexity [1]. For example, in Java programming based object-oriented software systems, if the organization of classes in the packages do not comply the different package design principles, it will make the software system complex. Many times, it has been observed that due to short delivery time and increase product cost during software development or maintenance, developers usually do not strict with the design principles that often makes software product complex.

In the Java based object-oriented software systems, to produce a good quality software product, the classes of the source code need to be distributed among the packages according to multiple software design criteria. The quality of software products often deteriorates if the systems are developed or maintained without following the package design criteria. In other word, the poor software package design has the several negative consequences on the system's understandability and maintainability. To keep the important software products relevant and useful, the package design of the deteriorated systems often needs to be overhauled by improving the package structure. However, improving the package structure of a software system is a challenging task because there requires optimization of multiple design criteria simultaneously.

To improve the various aspects of package design of a Java programming based object-oriented systems, a variety of software package restructuring (SPR) approaches based on different metaheuristic algorithms have been proposed (e.g. [2–4]). In these approaches, the SPR problem is modeled as a single, multi or many-objective optimization (MaOO) problem and solved using appropriate metaheuristic algo-

✉ Amarjeet Prajapati
amarjeetnitkkr@gmail.com

[1] Computer Engineering& IT Department, JIIT, Noida, India

rithms. Even the existing optimization approaches have the huge potential to tackle the vast spectrum of SPR problems, still there are many forms of SPR problems exist that poses several performance challenges to these approaches. The optimization model of SPR problems, consisting large number decision variables (i.e. more than 100) and objectives (i.e. more than 3) deteriorate the search capability of traditional optimization approaches and thus degrade the overall results. Overall, the scalability of the metaheuristic optimization algorithms with respect to the number of objectives and decision variables of the optimization problem is remained as a challenging issue for different science and engineering optimization problems [5–7].

To enhance the scalability of optimization algorithms in terms of objective functions, a number of optimization approaches have been proposed, e.g. IBEA [8], MOEA/D [9], KnEA [10], NSGA-III [11], and Tk-MaOEA [12]. Similarly, to improve the scalability of optimization algorithms in terms of decision variable, a variety of optimization approaches have also been proposed [13, 14]. To improve the scalability of optimization algorithms in terms of both objective functions and decision variables, in the previous few years some optimization have been proposed, e.g. CCGDE3 [15], MOEA/DVA [16], MOEA/D-RDG [17], LMEA [18], information feedback model [19], MOEA/D with information feedback models [5], NSGA-III algorithms with information feedback models [20], and weighted optimization framework (WOF) [21]).

In summary, the optimization problems containing more than three objective functions and more than hundred decision variables are the special category of optimization problems and such optimization problems are commonly regarded as LSMaOPs [22]. Most of the optimization algorithms typically focus on the optimization problems consisting 1, 2 or 3 objectives and small number (i.e. less than 100) decision variables. Additionally, these algorithms are designed by keeping the characteristics of synthetic optimization problems and there are very little works are carried out in designing the optimization algorithms for real-world optimization problems belong to the category of LSMaOPs. In software engineering, there are many problems that exhibit the characteristics of LSMaOPs. In object-oriented systems, SPR problems often contain properties of LSMaOP and can be defined as large-scale many-objective software package restructuring (LSMaOSPR) problem.

To solve the different aspects of the SPR problem apart from the LSMaOSPR perspective, in previous one decade, many search-based optimization approaches have been proposed. These search-based optimization approaches designed for the SPR problems can be categorized into following three major categories: single-objective SPR approach (e.g. [2]), multi-objective SPR approach (e.g. [3]), and many-objective SPR approach [1]. These approaches have been successfully applied and found to be as effective approach within a certain constraint. These approaches work well with a certain formulation of SPR problem and demonstrate poor performance if applied over large and complex definition of SPR problems (e.g. LSMaOSPR problem formulation). Even there has been huge progress in the development of a variety of optimization approaches to address the various aspects of SPR problems, the LSMaOSPR aspect of SPR problem gained little attention. To address the LSMaOSPR aspect of SPR problem, we design a customized particle swarm optimization algorithm (PSO) by exploiting and adapting the several existing strategies related to the selection of inertia weight, selection of cognitive and social constant, updating velocity and position of particles, and selection of personal best and global best position of the particles. The key contributions of this work are given below.

- To address the LSMaOSPR problem, a PSO based large-scale many-objective optimization approach, namely large-scale many-objective algorithm (LSMaOA) has been proposed.
- To this contribution, different components of PSO framework such as selection of inertia weight, selection of cognitive and social constant, updating velocity and position of particles, and determination of personal best and global best are customized.
- To lead the optimization process towards the evenly distributed and well-converged approximation of Pareto optimal front, different information feedback models (IFMs) strategies for the personal best selection have been exploited.
- To guide the optimization process towards a well-converged approximation of Pareto optimal front, grid-based, fuzzy-Pareto dominance-based, and Indicator based global best selection strategies have been adapted.

The subsequent part of the paper is presented as follows. Section 2 covers the literature mainly focusing on different aspects of search-based optimization and SPR approaches. Section 3 discusses SPR problem formulation as the LSMaOP. Section 4 presents the working process of the proposed approach along with the formation of different variants. Section 5 discusses the experimental setup and test problem selection. Section 6 presents comparative results of the different variants of the proposed approach. Section 7 and 8 present the discussion and threats to the validity of the proposed approach, respectively. Section 9 concludes the paper with future suggestions.

2 Related Work

This section covers the literature focusing on different aspects of search-based optimization approaches, especially large-scale single-objective, large-scale multi-objective, large-scale many-objective, and search-based SPR approaches.

2.1 Large-Scale Single-Objective Optimization

The class of optimization problems with a minimum of 100 decision variables and a maximum of 1 objective function is commonly referred to as large-scale single-objective optimization problems (LSSoOPs) and corresponding algorithms are known as large-scale single-objective optimization algorithms (LSSoOAs). In the survey of the LSSoOAs [14], a comprehensive discussion on the different aspects of LSSoOPs has been provided. The interested reader can read the paper to build up a strong understanding of the concepts of the LSSoOPs. In this survey paper, a detailed discussion related to the characteristics of the LSSoOPs has been provided. Further, a general framework to address the different types of LSSoOPs has also been provided.

To understand the effectiveness of the various LSSoOAs over the different types of benchmark optimization problems the authors LaTorre et al. [23] conducted an extensive comparative study. In this study, they evaluated the different LSSoOAs on several benchmark LSSoOPs. This work also studied the generality of the LSSoOAs, i.e. the capability to adapt performances to different LSSoOPs without tuning their parameters. Yang et al. [24] examined the general divide and conquer concept on LSSoOPs. They concluded that the major challenges of the divide and conquer strategy in the optimization process lie in the dimensionality mismatch. To tackle the dimensionality mismatch, they suggested evaluating the partial solutions of each sub problem separately by involving a computationally cheap meta-model. Andranik et al. [25] proposed an LSSoOAs, namely parallel multi-agent genetic algorithm to solve the large-scale black-box single-objective optimization problems.

2.2 Large-Scale Multi-Objective Optimization

The class of optimization problems having a minimum of 100 decision variables and minimum 2 objective functions are generally known as the large-scale multi-objective optimization problems (LSMoOPs) and the algorithms designed to solve them are known as large-scale multi-objective optimization algorithms (LSMoOAs). In the past decade, there has been tremendous progress in the development of LSMoOAs addressing the different aspects of LSMoOPs. Recently, Authors Hong et al. [6] conducted a review study on the progress of the evolutionary computation for the LSMoOAs. In this review paper, a comprehensive study on the LSMoOAs has been provided. The main focus of the study was to provide a summarized view of the scalability analysis and challenges of traditional LSMoOAs when applied to LSMoOPs. Based on the scalability, they categorized the LSMoOAs into the following three groups: enhanced search-based LSMoOAs, dimension reduction based LSMoOAs, and divide-and-conquer based LSMoOAs.

Recently other authors Tian et al. [7] presented a comprehensive survey of state-of-the-art LSMoOAs for addressing LSMoOPs. The survey is focused on the assessment methods, methodologies, and future directions of the LSMoOAs. To provide a comprehensive view of the literature, they categorized the LSMoOAs into the following three categories: novel search strategy based LSMoOAs, decision space reduction based LSMoOAs, and decision variable grouping based LSMoOAs. The detailed elaboration of these three categories along with their advantages and disadvantages has also been provided. These two survey papers provide a piece of detailed information related to the LSMoOAs, so the interested readers can read the papers to enhance their understandability. Even there are a huge number of LSMoOAs have been proposed, these two papers have covered most of the state-of-the-art LSMoOAs.

2.3 Large-Scale Many-Objective Optimization

The LSMoOPs having more than three objectives are commonly known as the LSMaOPs [26]. To solve the LSMaOPs, LSMaOAs are employed [27]. Designing of LSMaOAs for the different types of synthetic and real-world LSMaOPs is a challenging task compared to the designing of LSSaOAs to LSSaOPs and LSMoOAs for the LSMoOPs. In the last decades, a variety of LSMaOAs has been proposed addressing the different types of synthetic and real-world LSMaOPs (e.g. [19], 26, 27).

Cheng et al. [28] proposed a new approach to designing benchmark test problems for the LSMoOAs and LSMaOPs. Using the principles of the proposed approach, nine test problems, namely LSMOP1–LSMOP4 having linear Pareto front, LSMOP5–LSMOP8 have a nonlinear Pareto front and LSMOP9 having a disconnected Pareto front are instantiated. Zhang et al. [22] proposed a decision variable clustering-based LSMaOA, named as large-scale many-objective evolutionary algorithm (LMEA) for solving LSMaOPs. To divide the decision variables into different groups (e.g. diversity and convergence related decision variables) a clustering-based decision variable partitioning method are developed. Gu et al. [20] proposed an LSMaOA, by exploiting the information feedback model into the framework of the many-objective evolutionary algorithms NSGA-III. In this approach, the historical information of individual candidate solutions is used to guide the optimization process. Zhang et al. [5] have applied a similar information feedback approach in the framework of the MOEA/D to design the LSMaOA. Both of the information feedback based LSMaOA performed best on the different LSMaOPs.

2.4 Search-Based Package Restructuring

SPR is a common and challenging optimization problem of the software engineering field. To address the different optimization forms of the SPR problem, a variety of search-based metaheuristic optimization approaches have been proposed (e.g. [2], 329). These search-based metaheuristic optimization approaches addressed the different optimization forms of the SPR problems. Abdeen et al. [2] addressed the single-objective optimization aspect of the SPR problem using the Simulated Annealing (SA) metaheuristic optimization algorithm (Kirkpatrick et al. [30]. They combined the package coupling and cohesion into a single objective function and optimized it with the SA metaheuristic optimization algorithm. In this continuation, Abdeen et al. [3] treated the different quality criteria (e.g. coupling and cohesion) independently and optimized them simultaneously using NSGA-II, a multi-objective metaheuristic optimization algorithm.

Amarjeet and Chhabra [4] also treated the different quality criteria (e.g. coupling and cohesion) as independent objective functions and optimized them simultaneously using NSGA-II. However, they exploited more artefact information such as various dimensions of structural and lexical relationships in defining the package coupling and cohesion for the SPR problem. Mkaouer et al. [1] used a large number (more than 3) of SPR criteria as the objective functions and to restructure the software systems these objectives are optimized using NSGA-III a customized MaOA.

3 Problem Description

The formulation and encoding of any real-world problem into a search-based optimization problem is an important and challenging task of the metaheuristic optimization field. This section covers the SPR problem description and its formulation and encoding as the LSMaOP.

3.1 SPR Problem

The SPR is an important activity of the software reengineering process. In the SPR problem, a set of classes (i.e. Java programming based object-oriented software system) is reorganized into a disjoint set of packages based on the distinguish software qualities. The SPR problem can be formally defined as follows:

- For any object-oriented software system, consider the $C = \{c_1, c_2, \ldots, c_N\}$, where ($|C| = N$) is the set of N classes and $R = \{r_1, r_2, \ldots, r_T\}$, where ($|R| = T$) is the set of relationships existing among the classes of the system to be restructured.
- In the SPR problem, the set of classes $C = \{c_1, c_2, \ldots, c_N\}$ of the system need to be redistributed into a set of packages $P = \{p_1, p_2, \ldots, p_K\}$, ($|P| = K$) based on a set of quality criteria, i.e. the set of objective functions, $F = \{f_1, f_2, \ldots, f_M\}$, ($|F| = M$).

The software system can be represented as the graph, where the set of the classes $C = \{c_1, c_2, \ldots, c_N\}$ and set of relationships $R = \{r_1, r_2, \ldots, r_T\}$ can be the viewed as the graph's node and edges, respectively. The set of the classes $C = \{c_1, c_2, \ldots, c_N\}$ of the defined graph can be partitioned into the different partition represented with the set of packages $P = \{p_1, p_2, \ldots, p_K\}$. Overall, we can say that the SPR problem is basically a graph partitioning problem. It has already been proved that the graph partitioning problem is essentially a NP-hard problem which makes the exact/deterministic algorithms difficult to produce an optimal solution within a reasonable amount of time Mancoridis et al. [31]. This observation motivates to tackle the SPR problem as an evolutionary computing where the near-optimal solution can be generated within a reasonable amount of time.

3.2 SPR Problem as a LSMaOP

A LSMaOP is a special form of multi-objective optimization problem (MuOP), where the number of decision variables n and objectives m are restricted to the lower bound $n > 100$ and $m > 3$, respectively. The LSMaOPs can be either minimization or maximization optimization problem. The minimization LSMaOPs can be mathematically defined as follows:

$$\begin{array}{l} minimize F(x) = (f_1(x), f_2(x), \ldots, f_m(x)) \\ subjectto : x \in \Omega \end{array} \tag{1}$$

where $x = \{x_1, x_2, \ldots, x_n\} \in \Omega \subset R^n$ is a set of n decision variables. $F(x) = (f_1(x), f_2(x), \ldots, f_m(x)) \in Z \subset R^m$ is a set of m objective functions where a particular value of each objective forms an objective vector of having m-dimension located in objective space Z.

To map SPR problem on an LSMaOP, the decision variables and objective functions need to be defined. In the SPR problem, the set of classes $C = \{c_1, c_2, \ldots, c_N\}$ are redistributed among the set of packages $P = \{p_1, p_2, \ldots, p_K\}$ based on the set of objective functions $F = \{f_1, f_2, \ldots, f_M\}$. So, here the decision variables and their values can be defined in terms of the set of classes $C = \{c_1, c_2, \ldots, c_N\}$ and the set of packages $P = \{p_1, p_2, \ldots, p_K\}$. As a

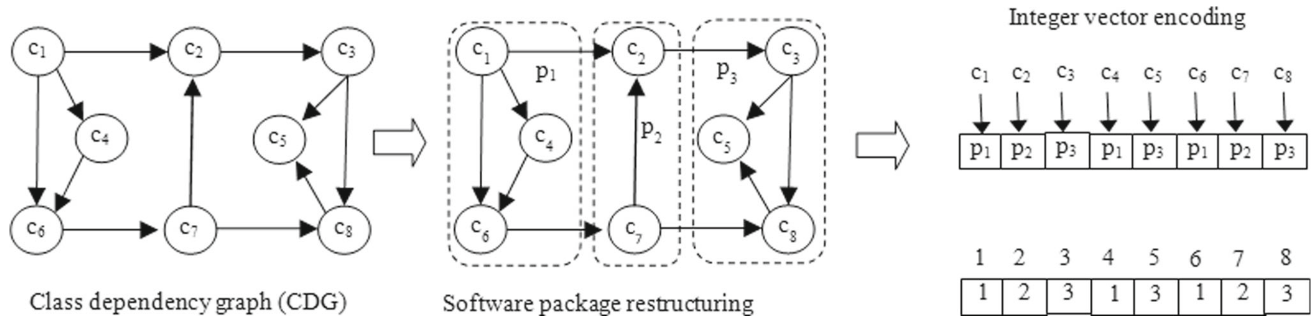

Fig. 1 SPR solution representation terms of decision variables

particular class c_i can be placed in any of the packages from the set $P = \{p_1, p_2, \ldots, p_K\}$, hence, a class c_i can be mapped with a decision variable name and the range of package with the decision variable values (i.e. 1 to K). To represent the solution space of SPR problem, there requires an appropriate encoding scheme. The integer vector encoding scheme ([29], [32], [33]) is an effective method to generate the solution space for the similar software engineering problems. The demonstration of integer vector encoding for a hypothetical SPR problem is provided in Fig. 1.

Since each of the classes in the SPR problem is mapped with a different decision variable. Hence, the number of decision variables increases with the increase in the number of classes of the SPR problem. It has been found that most of the multi-featured software systems are very large in size and generally involve more than 100 classes. So, the number of decision variables for such multi-objective formulation of the SPR problem will contain more than 100 decision variables. Apart from the formulation of the SPR problem in terms of decision variables, the problem objectives need to be defined in terms of different aspects of software quality. The different dimensions of software qualities are generally considered as the objective functions of the SPR problem. Hence, the number of objectives in the SPR problem gets increases with the increase in the number of involved software quality criteria. Overall, the formulation of the SPR problem as an optimization problem generally consists of multiple objective functions and a large number of decision variables. Therefore, the SPR problem consisting of more than 100 decision variables and more than 3 objectives can be characterized as the LSMaOP and can be referred to as the large-scale many-objective software package restructuring (LSMaOSPR).

To formulate the objective functions for the LSMaOSPR problem, we used the quality criteria suggested in studies [1, 4, 29]. These quality criteria are as follows: (1) structural software package coupling (to minimize), (2) structural software package cohesion (to maximize), (3) lexical software package coupling (to minimize), (4) lexical software package cohesion (to maximize), (5) changed-history software package coupling (to minimize), and (6) changed-history software package cohesion (to maximize), (7) the number of packages (to maximize), (8) difference between the minimum and the maximum number of classes in the packages (to minimize), and (9) Modularization quality (MQ). Based on the nature and number of quality criteria, we have designed the following five sets of problem objectives:

- Objective Set-4: (1) structural software package coupling (to minimize), (2) structural software package cohesion (to maximize), (3) the number of packages (to maximize), and (4) difference between the minimum and the maximum number of classes in the packages (to minimize).
- Objective Set-5: (1) structural software package coupling (to minimize), (2) structural software package cohesion (to maximize), (3) the number of packages (to maximize), (4) difference between the minimum and the maximum number of classes in the packages (to minimize), and Modularization quality (MQ).
- Objective Set-6: (1) structural software package coupling (to minimize), (2) structural software package cohesion (to maximize), (3) lexical software package coupling (to minimize), (4) lexical software package cohesion (to maximize), (5) the number of packages (to maximize), and (6) difference between the minimum and the maximum number of classes in the packages (to minimize).
- Objective Set-7: (1) structural software package coupling (to minimize), (2) structural software package cohesion (to maximize), (3) lexical software package coupling (to minimize), (4) lexical software package cohesion (to maximize), (5) the number of packages (to maximize), (6) difference between the minimum and the maximum number of classes in the packages (to minimize), and Modularization quality (MQ).
- Objective Set-8: (1) structural software package coupling (to minimize), (2) structural software package cohesion (to maximize), (3) lexical software package coupling (to minimize), (4) lexical software package cohesion (to maximize), (5) changed-history software package coupling (to minimize), and (6) changed-history software package

cohesion (to maximize), (7) the number of packages (to maximize), and (8) difference between the minimum and the maximum number of classes in the packages (to minimize).

In the objective set-5 and objective set-7, the computation MQ metrics involved a different set of information. In objective set-5, the MQ is defined in terms of the structural dependency information only, whereas in objective set-7, the MQ is defined in terms of both structural and lexical dependency information. A detailed description of the above objective functions defined in the different objective sets can be found in the literature [1], [4], [29]. Each set of objectives defined above consist of a different number and types of software quality criteria as the objective function. In the SPR problem, according to the requirements, the developers can consider different types of software quality criteria as the objective functions. To test the scalability of our proposed approach in terms of the number of objective functions and number of decision variables, we have considered the different sets of objectives with varying numbers of objectives functions.

4 Proposed Approach

Generally, a metaheuristic optimization framework contains several components and these components need to be defined to solve a particular class of optimization problems. The LSMaOSPR is a class of discrete combinatorial optimization problems. Due to inherent complications in such real-world discrete combinatorial optimization problems, designing a metaheuristic optimization algorithm becomes a very difficult task. It becomes more difficult if the optimization problem is a class of large-scale many-objective optimization. For the LSMaOSPR optimization problem, we utilized the PSO framework and redefined its components according to the requirements and suitability of the problem characteristics. The details of the redefined components and their strategies of the proposed approach are provided in the following sub-sections.

4.1 Velocity and Position of Particles

In the optimization model of the LSMaOSPR problem, the decision variables are discrete in nature, whereas most of the PSO-based optimization approaches are designed to work with the optimization problem having the continuous decision variables. Therefore, for the LSMaOSPR problem the definition of operators used in the updating velocity and position need to be redefined. In this work, we derived the velocity and position updating rules from the work presented in [29]. The mathematical equation updating the velocity and position is given as follows:

$$\begin{aligned} v_i^{\text{new}} = &\Omega\big(\omega \times v_i + c_1 r_1 \times \left(\text{pbest}_i \oplus p_i\right) \\ &+ c_2 r_2 \times (\text{gbest} \oplus p_i)\big) \end{aligned} \quad (2)$$

$$p_i^{new} = p_i \Theta v_i^{new} \quad (3)$$

where ω represents inertia weight. $c_1 r_1$ and $c_2 r_2$ both are combination of learning factor and random value (between 0 to1) used to control the level of cognitive and social components, respectively. The pbest_i, gbest,v_i, and p_i, represent personal best position, global best position, current velocity, and current position of the ith particle of the swarm. The operators ×, +, and ⊕ are defined as the simple multiplication, addition, and XOR operator.

The symbol ω represents the inertia weight which controls the impact of current velocity in the new velocity of a particle of the swarm. It has major influence on exploration and exploitation process of the optimization algorithm. Specially, its low values enforce the algorithm towards exploitation and high value towards exploration. To set the value of ω, a variety of approaches have been suggested. However, customized approaches have been found more appropriate. In this work, we use the nonlinear decreasing strategy of the inertia weight presented by the authors Ting et al. [34]. According to this approach the value of the ω is computed as follows:

$$\omega(t) = \omega_{max} - \left(\frac{t-1}{t_{max}-1}\right)^{\alpha} (\omega_{max} - \omega_{min}) \quad (4)$$

The symbols t and t_{max} are the current iteration and the maximum iterations, respectively, of the optimization process. The ω_{min} and ω_{max} are the minimum and maximum inertia weight value, respectively. The parameter α denotes the decline exponent. The value of cognitive coefficient (c_1) and the social coefficient (c_2) also has an important effect on the exploitation and exploration of the search space. It has been observed that the linear variation in cognitive coefficient (decreasing) and social coefficient (increasing) over algorithm iteration helps the optimization process in smooth transition from exploration to exploitation. In this work, we adopt the similar approach in the definition of linear variations of the cognitive and social coefficients. The linear variations in the cognitive coefficient (c_1) and the social coefficient (c_2) are defined as follows:

$$c_1(itr) = c_1^i + \left(c_1^f - c_1^i\right)\frac{t}{t_{max}} \quad (5)$$

$$c_2(itr) = c_2^i + \left(c_2^f - c_2^i\right)\frac{t}{t_{max}} \quad (6)$$

where c_1^i, c_1^f and c_2^i, c_2^f are the initial and final value of the cognitive and social coefficients. The t and t_{max} are the current iteration and the maximum iterations, respectively, of the optimization process.

The function $\Omega(.)$ returns a velocity vector (i.e. if the received vector index value is > 0 then return 1 otherwise return 0). To define the Θ operator used in $p_i \Theta v_i^{\text{new}}$, we exploit the strategy proposed in the study [35]. According to this strategy, the $p_i \Theta v_i^{\text{new}}$ is defined as follows:

$$\begin{aligned}\sigma_p =& \frac{\exp(fit(p_i(t)))}{exp\left(\frac{1}{N}\sum_{i=1}^{N} fit(p_i(t))\right)}, \sigma_a \\ =& \frac{\exp\left(fit\left(a_j(t)\right)\right)}{exp\left(\frac{1}{|CA+DA|}\sum_{j=1}^{|CA+DA|} fit\left(a_j(t)\right)\right)}\end{aligned} \tag{7}$$

$$\begin{aligned}p_i^{new} =& p_i \Theta v_i^{new} \\ =& \left\{\begin{array}{ll} p_{i,d}^{new} = p_{i,d}^{old}, & if\, v_{i,d} = 0 \\ p_{i,d}^{new} = RAND\left(p_{1,d}, p_{2,d}, \right. & if\ v_{i,d} = 1 \\ \quad \left. \ldots, p_{N,d}\right), & \end{array}\right\} \quad if\, \sigma_a < \sigma_p\end{aligned} \tag{8}$$

$$\begin{aligned}p_i^{new} =& p_i \Theta v_i^{new} \\ =& \left\{\begin{array}{ll} p_{i,d}^{new} = p_{i,d}^{old}, & if\ v_{i,d} = 0 \\ p_{i,d}^{new} = RAND\left(a_{1,d}, a_{2,d}, \right. & if\ v_{i,d} = 1 \\ \quad \left. \ldots, a_{|CA+DA|,d}\right), & \end{array}\right\} \quad if\, \sigma_a \geq \sigma_p\end{aligned} \tag{9}$$

In the above equations, $fit(p_i(t))$ denotes the fitness of candidate solution, i.e. position p_i. The $CAandDA$ are the external archives, and $RAND\left(p_{1,d}, p_{2,d}, \ldots, p_{N,d}\right)$ is used to randomly select a value from the range of decision variable.

4.2 Managing External Archives

Preserving the elitist candidate solutions in multi-objective optimization algorithm is an important activity [36]. In the literature of multi-objective optimization, a variety of mechanisms to preserve the elitist candidate solution during optimization process have been presented [2–4]. The external archive-based elitist preservation mechanism has been found as the most effective approach in preserving the elitist candidate solution in multi-objective optimization algorithms [37]. In the external archive-based elitist preservation mechanism the non-dominated solutions found during the optimization process are stored in a fixed size external archive. As the optimization process proceeds from one generation to another generation, the non-dominated solutions of the current external archive may not be non-dominated with the produced non-dominated solutions, therefore the external archive requires continuous updating mechanism. Due to the limited size of the external archives and huge numbers of non-dominated solutions, it also requires an effective truncation mechanism so that a well converge and diverse non-dominated solution set can be achieved at the end of algorithm termination.

In this work, we adopted the two-archive2 external storing mechanism presented in [37]. In the two-arch2, two archives, namely convergence archive (CA) and diverse archive (DA) of the same size are used to store non-dominated solutions. To store the non-dominated solutions following rules are used: (1) if both archives CA and DA are empty then the new non-dominated solution is placed in the DA archive; (2) If the non-dominated solution of the swarm is non-dominated with the CA and DA candidate solution then it is placed in the DA archives; (3) If the new non-dominated solution dominates one or more non-dominated solutions of CA and DA archives then dominated solutions of CA and DA are deleted and placed into the CA archive. If the non-dominated solution is dominated by one or more candidate solutions of CA and DA, then the non-dominated solution is simply discarded. However, because of computational efficiency and storage space, the size of CA and DA is not infinite. When the CA and DA archives reach its maximum size, to remove the extra non-dominated solutions from both the archives, in this work, we adopted the same strategies as suggested in [37].

4.3 Selection of Personal Best

In the PSO, the optimization process is usually driven by two things: the particle's personal best and the swarm's global best. The selection of the personal and global best in the case of LSMaOPs is not straightforward. The selection of these two guides can make a significant influence on the searching capability of the algorithm. In LSMaOP, if the new position of a particle dominates the current personal best, then it is straightforward that the current personal best will be replaced with the new position. However, it becomes challenging if both the new position and current personal best are non-dominated with each other. In this case, the simplest and easiest strategy can be the selection of either a new position or a current personal best as a personal best. Many of the existing approaches have used a similar idea in the selection of personal best. To make a balance impact on the convergence and diversity in the approximation set of Pareto front, it is necessary to use some more effective approach for the selection of personal best.

In this work, we adopt the concept of information feedback model to select the personal best [38]. In this model the information of the current personal best and previous personal best positions is used to determine the personal best position. In this work, we only consider the information feedback of previous three personal best positions. For the ease of understanding, let's consider for each i^{th} individual of the swarm the temporary personal best is p_{temp}^i, and the previous three personal best positions are:p^{i}_{prev-1}, p^{i}_{prev-2}, and p^{i}_{prev-3}. The fitness of these three previous best personal best positions is: f^{i}_{prev-1}, f^{i}_{prev-2}, and f^{i}_{prev-3}. To determine the final personal best positron p^{i}_{final} for each of the particle i, we have the following three models. The temporary personal best is p^{i}_{temp} is selected based on the traditional personal best selection approach.

Model P1: In this model, to generate the final personal best position, we consider the temporary personal best and the first previous personal best position.

$$p^{i}_{final} = \left[\left(\alpha \odot p^{i}_{temp}\right) \oplus \left(\beta \odot p^{i}_{prev-1}\right)\right], where\, \alpha = \frac{f^{j}_{temp}}{f^{1}_{temp} + f^{j}_{prev-1}}, \beta = \frac{f^{i}_{prev-1}}{f^{1}_{temp} + f^{j}_{prev-1}} \quad (10)$$

The α and β are the parameters satisfying $\alpha + \beta = 1$. The $\odot$ operator selects the number of the decision variables from the individual candidate solution according to proportional of α and β values. The $\oplus$ operator combines the selected decision variables values into a single individual. For example, if $\alpha = 0.7$ and $= 0.3$, then they select the 70% and 30% decision variables from the beginning and end of the individual candidate solutions, respectively.

Model P2: In this model, to generate the final personal best position, we consider the temporary personal best and the first and second previous personal best position.

$$p^{i}_{final} = \left[\left(\alpha \odot p^{i}_{temp}\right) \oplus \left(\beta 1 \odot p^{i}_{prev-1}\right) \oplus \left(\beta 2 \odot p^{i}_{prev-2}\right)\right] \quad (11)$$

The α, $\beta 1$, and $\beta 2$ are the parameters satisfying the $\alpha + \beta 1 + \beta 2 = 1$. The values of these parameters are determined using the following equation.

$$\begin{aligned} \alpha &= \frac{1}{2} * \frac{f^{i}_{temp} + f^{i}_{prev-1}}{f^{i}_{temp} + f^{i}_{prev-1} + f^{i}_{prev-2}}, \\ \beta 1 &= \frac{1}{2} * \frac{f^{i}_{prev-1} + f^{i}_{prev-2}}{f^{i}_{temp} + f^{i}_{prev-1} + f^{i}_{prev-2}}, \\ \beta 2 &= \frac{1}{2} * \frac{f^{i}_{prev-2} + f^{i}_{temp}}{f^{i}_{temp} + f^{i}_{prev-1} + f^{i}_{prev-2}} \end{aligned} \quad (12)$$

Model P3:In this model, to generate the final personal best position, we consider the temporary personal best position and all three previous personal best position.

$$p^{i}_{final} = \left[\left(\alpha \odot p^{i}_{temp}\right) \oplus \left(\beta 1 \odot p^{i}_{prev-1}\right) \oplus \left(\beta 2 \odot p^{i}_{prev-2}\right) \oplus \left(\beta 3 \odot p^{i}_{prev-3}\right)\right] \quad (13)$$

The α, $\beta 1$, $\beta 2$ and $\beta 3$ are the parameters satisfying the $\alpha + \beta 1 + \beta 2 + \beta 3 = 1$. The values of these parameters are determined using the following equion.

$$\begin{aligned} \alpha &= \frac{1}{3} * \frac{f^{i}_{temp} + f^{i}_{prev-1} + f^{i}_{prev-2}}{f^{i}_{temp} + f^{i}_{prev-1} + f^{i}_{prev-2} + f^{i}_{prev-3}}, \\ \beta 1 &= \frac{1}{3} * \frac{f^{i}_{prev-1} + f^{i}_{prev-2} + f^{i}_{prev-3}}{f^{i}_{temp} + f^{i}_{prev-1} + f^{i}_{prev-2} + f^{i}_{prev-3}} \end{aligned} \quad (14)$$

$$\begin{aligned} \beta 2 &= \frac{1}{3} * \frac{f^{i}_{prev-2} + f^{i}_{prev-3} + f^{i}_{temp}}{f^{i}_{temp} + f^{i}_{prev-1} + f^{i}_{prev-2} + f^{i}_{prev-3}}, \\ \beta 2 &= \frac{1}{3} * \frac{f^{i}_{prev-3} + f^{i}_{temp} + f^{i}_{prev-1}}{f^{i}_{temp} + f^{i}_{prev-1} + f^{i}_{prev-2} + f^{i}_{prev-3}} \end{aligned} \quad (15)$$

To compute the fitness of each of the personal best positions, first we combine each of the four personal best position, then compute the fitness using the binary additive epsilon ($I_{\in+}$) quality indicator [8].

4.4 Selection of Global Best

The main role of the global best in the PSO is to lead the swarm towards the Pareto front. However, due to a large number of the non-dominated solutions in the case of the LSMaOPs, it is inevitably difficult to select a global best position. In the global best selection mechanism, it is generally suggested to choose those candidate solutions from a set of non-dominated solutions that can guide the swarm towards a well converged and well-distributed approximation of the Pareto front. In this work, the elitist non-dominated solutions are maintained in the CA and DA archives. Therefore, we select a non-dominated solution from these solutions as the global best position. To rank the non-dominated solutions, a variety of methods can be used. In this work, we considered the following three methods.

Model G1: In this model, we use the Grid-based strategy to select the global best from the external archives [39]. The grid-based selection strategy influences both the convergence and diversity of the optimization because it considers both properties while determining the fitness value for the selection of a candidate solution.

Model G2: In this model of the global best selection approach, we use the fuzzy-Pareto dominance strategy [33], [40]. In this global best selection strategy, fuzzy-Pareto dominance-based fitness values for the non-dominated candidate solutions are computed and then based on the fitness value each candidate solution is ranked.

Model G3: In this model of the global best selection approach, we use an indicator-based approach [8] a popular approach to distinguish the non-dominated solutions. In this strategy, a quality indicator-based technique is used to compute the degree of dominance of the candidate solutions and based on the degree of dominance the non-dominated solutions are ranked and selected.

5 Experimental Setup

To test the proposed approach, we use the five object-oriented software projects: (1) Xerces-J v2.7.0 (#classes 991, #packages 55), (2) JDI-Ford v5.8 (#classes 638, #packages 37), (3) JHotDraw v6.0.b.1 (#classes 398, #packages 17), (4) DOM 4 J v1.5.2 (#classes 195, #packages 16), and JUnit v3.81(#classes 100, #packages (6). These software projects have varying characteristics and can be easily found on the web. These software projects have widely been considered by the academicians and researchers to test the similar types of metaheuristic approaches designed for the SPR problems. The modelling of each of the software projects as an optimization problem consists of 100, 195, 398, 638, and 991 decision variables, respectively. The variation in size and complexity has made the selected software more appropriate for testing the proposed approach.

Based on the different strategies used for the selection of personal best and global best, we categorized the proposed approach into nine variants. These nine variants of the proposed approach are P1G1, P1G2, P1G3, P2G1, P2G2, P2G3, P3G1, P2G2, and P3G3. Each of these variants is executed over all the five problem instances with a varying number of decision variables and objective functions. Due to the stochastic nature, the final output of these variants may not be the same on the different runs over the same problem instance. To draw the most appropriate conclusion about the performance of these variants, we conducted the statistical test using the Mann–Whitney Wilcoxon rank-sum test with a 5% confidence level and 95% level of significance. The selection of quality indicators for the evaluation of obtained Pareto front is another challenging task. In this approach, we use the following three most widely quality indicators for the evaluation of the final results: hypervolume (HV) [41], modularization quality (MQ) [32], and inverted generational distance (IGD) [42].

6 Results

The Pareto fronts of package restructuring solutions obtained through the different variants, i.e. P1G1, PP1G2, P1G3, P2G1, P2G2, P2G3, P3G1, P2G2, and P3G3 are assessed in terms of the IGD, Hypervolume, and MQ measures. The statistical Wilcoxon rank-sum test results applied over all variants of the proposed work are in the following form: some variants may be significantly better, significantly worst, or no significant difference to the other variants of the proposed approach. To make the comparison among all variants more comprehensive, we use the concept of difference [ranks]. In the difference [ranks] comparison approach, the difference for a particular variant denotes that the difference of significantly better and significantly worse compared to other variants. The rank value of a particular variant is computed based on the ordering of the difference values. Tables 1, 2, and 3 present the achieved difference [ranks] values of the different variants corresponding to the IGD, Hypervolume, and MQ quality measures, respectively.

In Table 2, the difference [ranks] values corresponding to the IGD measure demonstrate that the variant P3G1 has achieved rank1 in most of the cases. Now if we see the ranking results of P3G1, its value do not degrades with the increase in the decision variables and objective functions. In other words, the P3G1 has maintained its performance in the case of an increased number of decision variables and objective functions. This result reflects that the P3G1 is more scalable corresponding to the decision variables and objective functions. Now if we see the results of other variants, the P2G1 is producing more competitive results to the P3G1. The P2G3 and P1G3 are performing very poor compared to the other variants, as they have very poor ranking values. Overall, the performance ordering of all the variants based on the ranking are: P3G1 > P2G1 > P1G1 > P3G2 > P2G2 > P1G2 > P3G3 > P2G3 > P1G3. Even these variants have a different ranking; the performance of these variants is stable with the increase in the number of decision variables and objective functions. In other words, each variant of the proposed approach has good scalability to the number of decision variables and objective functions. Figure 2 provides the summarized view of different variants in terms of IGD.

The difference [ranks] values achieved by the different variants corresponding to the Hypervolume quality measure are presented in Table 2. The difference [ranks] values of the different variants presented in Table 2 show that the P3G1 has gained rank 1 in most of the cases. Moreover, the P3G1 has also preserved its performance with problem instance having large number of decision variables and objective functions. This observation provides sufficient support that the P3G1 has the enough capability to produce a well-distributed approximation of Pareto front. Now if we see the ranking values of the other variants, their ranking are in the following decreasing order: P2G1 > P1G1 > P3G2 > P2G2 > P1G2 > P3G3 > P2G3 > P1G3. If we assess the scalability of these variants, all the proposed variants are scalable in terms of the number of decision variables and the objective functions. Because the ranking values of these variants are not influenced by the number of decision variables and objective functions. Figure 3 provides the summarized view of different variants in terms of HV.

Table 1 Comparative differences [ranks] results of different variants of proposed work in terms of IGD

System	Objectives	P3G1	P2G1	P1G1	P3G2	P2G2	P1G2	P3G3	P2G3	P1G3
JUnit (100)	4	+ 8[1]	+ 6[2]	+ 1[3]	+ 1[3]	− 2[6]	+ 0[5]	− 2[6]	− 4[7]	− 5[8]
	5	+ 6[2]	+ 8[1]	+ 4[3]	+ 2[4]	− 1[5]	− 6[7]	− 1[5]	− 6[7]	− 6[7]
	6	+ 8[1]	+ 6[2]	+ 4[3]	+ 2[4]	+ 0[5]	− 4[6]	− 4[6]	− 4[6]	− 4[6]
	7	+ 8[1]	+ 6[2]	+ 3[3]	+ 3[3]	+ 0[5]	− 5[7]	− 2[6]	− 5[7]	− 5[7]
	8	+ 8[1]	+ 6[2]	+ 4[3]	− 2[5]	− 2[5]	− 2[5]	+ 2[4]	− 2[5]	− 6[8]
DOM 4 J (195)	4	+ 7[1]	+ 7[1]	+ 2[3]	+ 1[4]	+ 0[6]	+ 2[5]	− 5[7]	− 5[7]	− 8[9]
	5	+ 8[1]	+ 4[2]	+ 4[2]	+ 3[4]	− 4[6]	− 4[6]	+ 1[5]	− 4[6]	− 8[9]
	6	+ 8[1]	+ 6[2]	+ 3[3]	− 3[6]	+ 3[3]	− 3[6]	− 7[8]	+ 0[5]	− 7[8]
	7	+ 5[2]	+ 8[1]	+ 5[2]	− 5[7]	− 2[6]	− 5[7]	+ 2[4]	+ 0[5]	− 8[9]
	8	+ 7[1]	+ 7[1]	+ 4[3]	− 6[3]	+ 0[5]	− 3[6]	+ 2[4]	− 3[6]	− 8[9]
JHotDraw (398)	4	+ 8[1]	+ 2[2]	+ 2[2]	+ 1[4]	+ 1[4]	+ 1[4]	− 7[8]	− 7[8]	− 1[7]
	5	+ 5[2]	+ 8[1]	+ 2[4]	+ 3[3]	+ 1[5]	− 3[7]	− 2[6]	− 6[8]	− 8[9]
	6	+ 8[1]	+ 5[2]	+ 5[2]	+ 2[4]	− 3[6]	+ 0[5]	− 7[8]	− 3[6]	− 7[8]
	7	+ 6[2]	+ 8[1]	+ 2[3]	+ 2[3]	+ 2[3]	− 4[7]	− 2[6]	− 8[9]	− 6[8]
	8	+ 8[1]	+ 6[2]	+ 2[4]	+ 3[3]	+ 1[5]	− 7[8]	− 3[6]	− 3[6]	− 7[8]
JFreeChart (638)	4	+ 8[1]	+ 2[3]	+ 2[3]	+ 5[2]	− 2[5]	− 2[5]	− 4[7]	− 5[9]	− 4[7]
	5	+ 5[2]	+ 8[1]	+ 3[3]	+ 1[4]	− 1[5]	− 1[5]	− 1[5]	− 6[8]	− 8[9]
	6	+ 8[1]	+ 4[3]	+ 6[2]	+ 1[4]	− 1[5]	− 3[7]	− 2[6]	− 5[8]	− 8[9]
	7	+ 8[1]	+ 6[2]	+ 3[3]	+ 0[5]	+ 2[4]	− 3[6]	− 3[6]	− 6[8]	− 7[9]
	8	+ 8[1]	+ 4[2]	+ 4[2]	+ 0[5]	− 1[6]	+ 3[4]	− 4[7]	− 8[9]	− 6[8]
Xerces-J (991)	4	+ 1[1]	+ 1[1]	+ 1[1]	+ 0[6]	+ 0[6]	+ 0[6]	+ 0[6]	+ 0[6]	− 4[9]
	5	+ 8[1]	+ 5[2]	+ 2[4]	+ 4[3]	− 5[7]	− 2[6]	+ 1[5]	− 8[9]	− 5[7]
	6	+ 6[2]	+ 8[1]	+ 2[4]	+ 4[3]	− 1[5]	− 4[7]	− 1[5]	− 6[8]	− 8[9]
	7	+ 8[1]	+ 6[2]	+ 2[4]	+ 0[5]	+ 4[3]	− 2[6]	− 4[7]	− 6[8]	− 8[9]
	8	+ 8[1]	+ 5[2]	+ 2[4]	+ 4[3]	− 1[5]	4[7]	− 1[5]	− 6[8]	− 8[9]

The difference [ranks] values of each variant computed in terms of the MQ measure are presented in Table 3. Similar to the difference [ranks] values of the different variants corresponding to the IGD and Hypervolume measure, the P3G1 again has been able to secure rank 1 in most of the cases. In this case, the ranking values of the other variants are in the following order: P2G1 > P1G1 > P3G2 > P2G2 > P1G2 > P3G3 > P2G3 > P1G3. Now if we see the ranking values of each variant, these values are consistent over the different problem instances. It demonstrates that the performance of each variant is not affected with the increase in decision variables and objective functions. From the IGD, Hypervolume, and MQ results it can be concluded that the proposed variants have enough capability to maintain scalability in terms of decision variables and objective functions. Figure 4 provides the summarized view of different variants in terms of MQ.

In summary, the IGD, Hypervolume, and MQ results presented in Tables 1, 2 and 3 indicate that all the proposed variants are highly scalable with the number of decision variables and objective functions. The comparative results of all the variants showed that the P3G1 is more effective compared to the rest of the variants. As the IGD and Hypervolume measure evaluate both convergence and diversity of the Pareto front, hence the presented results indicate that the P3G1 can generate solutions that approximate the Pareto front well in terms of diversity and convergence. Now if we analyse the reason for such an effective result of P3G1, it could be the better strategies involved in the optimization framework. In any multi/many-objective PSO, the personal best and global best selection strategies are the two main important components that highly affect the performance of the algorithm. In P3G1, we have used the best feedback model and grid-based ranking schemes which guide the optimization process towards the generation solutions that approximate the Pareto front well in terms of diversity and convergence.

7 Discussion

The main goal of this work is to design a metaheuristic optimization algorithm that can produce a well-converged and well-distributed Pareto front for the LSMaOSPR problem. To achieve the goal multiple strategies have been exploited and incorporated into the framework of the PSO algorithm. Based on the various methods of personal best selection and global

Table 2 Comparative differences [ranks] results of different variants of proposed work in terms of Hypervolume

System	Objectives	P3G1	P2G1	P1G1	P3G2	P2G2	P1G2	P3G3	P2G3	P1G3
JUnit (100)	4	+ 8[1]	+ 1[2]	+ 1[2]	+ 1[2]	− 1[5]	− 1[5]	− 1[5]	− 4[8]	− 4[8]
	5	+ 6[2]	+ 8[1]	+ 2[3]	+ 2[3]	− 1[6]	+ 1[5]	− 5[7]	− 5[7]	− 8[9]
	6	+ 8[1]	+ 6[2]	+ 4[3]	+ 2[4]	+ 0[5]	− 4[6]	− 4[6]	− 4[6]	− 8[9]
	7	+ 7[1]	+ 7[1]	+ 4[3]	+ 1[4]	+ 1[4]	− 3[6]	− 7[8]	− 3[6]	− 7[8]
	8	+ 7[1]	+ 6[2]	− 1[5]	+ 5[3]	− 1[5]	+ 2[4]	− 6[8]	− 4[7]	− 8[9]
DOM 4 J (195)	4	+ 6[2]	+ 8[1]	+ 3[3]	+ 3[3]	− 3[6]	+ 0[5]	− 3[6]	− 7[8]	− 7[8]
	5	+ 8[1]	+ 6[2]	+ 3[3]	+ 0[5]	+ 3[3]	− 3[6]	− 6[8]	− 3[6]	− 8[9]
	6	+ 8[1]	+ 6[2]	+ 3[3]	+ 3[3]	− 4[7]	+ 0[5]	− 2[6]	− 6[8]	− 8[9]
	7	+ 7[1]	+ 7[1]	+ 0[5]	+ 4[3]	+ 3[4]	− 6[8]	− 2[6]	− 4[7]	− 8[9]
	8	+ 8[1]	+ 6[2]	+ 4[3]	+ 2[4]	− 2[6]	+ 0[5]	− 4[7]	− 6[8]	− 8[9]
JHotDraw (398)	4	+ 8[1]	+ 8[1]	+ 4[3]	+ 1[4]	+ 6[3]	+ 1[4]	− 6[8]	− 3[7]	− 8[9]
	5	+ 8[1]	+ 6[2]	+ 2[3]	+ 2[3]	+ 2[3]	+ 2[3]	− 6[8]	− 4[7]	− 6[8]
	6	+ 8[1]	+ 6[2]	+ 0[4]	+ 0[4]	+ 4[3]	− 1[5]	− 7[8]	− 2[7]	− 7[8]
	7	+ 8[1]	+ 6[2]	+ 6[2]	− 2[5]	+ 2[4]	− 3[6]	− 4[8]	− 8[9]	− 4[8]
	8	+ 8[1]	+ 6[2]	+ 6[2]	+ 4[3]	+ 2[4]	− 1[5]	− 6[8]	− 4[7]	− 8[9]
JFreeChart(638)	4	+ 7[1]	+ 7[1]	+ 4[3]	+ 0[5]	+ 2[4]	− 5[7]	− 5[7]	− 3[6]	− 7[9]
	5	+ 8[1]	+ 6[2]	+ 6[2]	+ 2[4]	+ 4[3]	− 2[5]	− 6[8]	− 3[7]	− 8[9]
	6	+ 8[1]	+ 6[2]	+ 4[3]	+ 0[5]	− 3[6]	+ 2[4]	− 4[7]	− 5[8]	− 8[9]
	7	+ 8[1]	+ 6[2]	+ 3[3]	+ 0[5]	+ 3[3]	− 3[6]	− 3[6]	+ 6[8]	− 8[9]
	8	+ 6[2]	+ 8[1]	+ 6[2]	+ 0[5]	+ 2[4]	+ 4[3]	− 2[6]	− 5[7]	− 8[9]
Xerces − J (991)	4	+ 8[1]	+ 6[2]	+ 2[3]	+ 2[3]	− 2[6]	− 6[8]	− 4[7]	− 8[9]	− 6[8]
	5	+ 8[1]	+ 6[2]	+ 1[4]	+ 4[3]	− 4[7]	− 2[6]	− 8[9]	− 6[8]	− 8[9]
	6	+ 8[1]	+ 6[2]	+ 2[3]	+ 2[3]	− 4[7]	− 2[5]	− 2[6]	− 6[8]	− 8[9]
	7	+ 6[1]	+ 6[1]	+ 3[3]	+ 1[4]	+ 0[5]	− 3[7]	− 3[7]	− 6[8]	− 7[9]
	8	+ 7[1]	+ 6[2]	+ 1[4]	+ 1[4]	+ 5[3]	− 5[7]	− 5[7]	− 5[7]	− 7[9]

best selection, nine variants of the proposed approach were created. These nine variants of the proposed approach are P1G1, P1G2, P1G3, P2G1, P2G2, P2G3, P3G1, P2G2, and P3G3. The results show that the nine variants of the proposed work generate different Pareto front because the different combinations of the personal best selection model and global best selection model contribute differently in maintaining convergence and diversity.

The IGD, HV, and MQ results demonstrate that the P3G1 variant is performing better compared to the rest of the variants. This indicates that the P3G1 variant has enough potential to solve the optimization problems having a large number of decision variables and objective functions. Results also demonstrate that the performance of the P3G1 variant does not deteriorate with the increase in the decision variables and objective functions. It means the P3G1 has good scalability potential with respect to the number of decision variables and objective functions of the optimization problems. The overall performance order of the different variants of the proposed approach corresponding to the IGD, HV, and MQ quality indicator measures is: P3G1 > P2G1 > P1G1 > P3G2 > P2G2 > P1G2 > P3G3 > P2G3 > P1G3. Even though the P2G1, P1G1, P3G2, P2G2, P1G2, P3G3, P2G3, P1G3 are performing poorly compared to the P3G1, still they have similar scalability potential.

The main reason of producing such good results with the P3G1 is that its personal best selection and global best selection strategy have the better capacity in guiding the optimization process towards a well-distributed approximation of the Pareto front. The personal best selection strategy in the P3G1 uses previous personal best feedback information for determining the next personal best of the particle. It indicates that more feedback information helps in guiding the optimization process towards better search space points. Moreover, the P3G1 uses a grid-based selection strategy to determine the global best from the external archives. The grid-based selection strategy (i.e. G1) involves both the divergence and convergence properties in guiding the optimization process. This could be another reason of generating good results.

The major limitation of this work is that the proposed approach is solving the LSMaOSPR problem as the LSMaOP, where the main goal is to produce a well-distributed approximation of the Pareto front rather than

Table 3 Comparative differences [ranks] results of different variants of proposed work in terms of MQ

System	Objectives	P3G1	P2G1	P1G1	P3G2	P2G2	P1G2	P3G3	P2G3	P1G3
JUnit (100)	4	+ 8[1]	+ 2[3]	+ 6[2]	+ 2[3]	− 1[5]	− 3[7]	− 1[5]	− 5[8]	− 8[9]
	5	+ 8[1]	+ 6[2]	+ 4[3]	+ 2[4]	+ 0[5]	− 4[7]	− 2[6]	− 6[8]	− 8[9]
	6	+ 6[1]	+ 6[1]	+ 5[3]	− 1[5]	− 1[5]	+ 3[4]	− 6[8]	− 4[7]	− 8[9]
	7	+ 6[1]	+ 5[2]	+ 5[2]	+ 4[4]	− 4[7]	− 1[5]	− 2[6]	− 5[8]	− 8[9]
	8	+ 8[1]	+ 5[2]	+ 5[2]	+ 0[5]	+ 2[4]	− 2[6]	− 5[7]	− 5[7]	− 8[9]
DOM 4 J (195)	4	+ 7[1]	+ 7[1]	+ 3[3]	+ 3[3]	+ 0[5]	− 2[6]	− 7[8]	− 4[7]	− 7[8]
	5	+ 8[1]	+ 6[2]	+ 4[3]	+ 0[5]	+ 2[4]	− 4[7]	− 2[6]	− 7[8]	− 7[8]
	6	+ 8[1]	+ 4[2]	+ 4[2]	+ 3[4]	+ 0[5]	− 4[7]	− 6[8]	− 1[6]	− 8[9]
	7	+ 7[1]	+ 7[1]	+ 3[3]	+ 1[4]	+ 1[4]	− 4[7]	− 8[9]	− 1[6]	− 6[8]
	8	+ 8[1]	+ 6[2]	+ 4[3]	+ 2[4]	− 2[6]	+ 0[5]	− 4[7]	− 7[8]	− 7[8]
JHotDraw (398)	4	+ 5[2]	+ 8[1]	+ 5[2]	+ 2[4]	+ 0[5]	+ 5[7]	− 2[6]	− 5[7]	− 8[9]
	5	+ 7[1]	+ 7[1]	+ 2[4]	+ 4[3]	+ 0[5]	− 3[6]	− 3[6]	− 6[8]	− 8[9]
	6	+ 4[2]	+ 8[1]	+ 1[3]	+ 1[3]	− 2[6]	+ 0[5]	− 2[6]	− 6[9]	− 4[8]
	7	+ 8[1]	+ 5[2]	+ 5[2]	+ 1[4]	+ 1[4]	− 4[7]	− 3[6]	− 6[8]	− 7[9]
	8	+ 8[1]	+ 6[2]	+ 3[3]	+ 2[4]	+ 0[5]	− 3[7]	− 2[6]	− 6[8]	− 8[9]
JFreeChart(638)	4	+ 8[1]	+ 6[2]	+ 2[3]	+ 2[3]	− 2[5]	− 2[5]	− 2[5]	− 6[8]	− 6[8]
	5	+ 8[1]	+ 6[2]	+ 4[3]	+ 1[4]	+ 1[4]	− 3[6]	− 6[8]	− 3[6]	− 8[9]
	6	+ 8[1]	+ 6[2]	+ 4[3]	+ 1[4]	− 1[6]	+ 0[5]	− 7[8]	− 4[7]	− 7[8]
	7	+ 8[1]	+ 6[2]	+ 3[3]	+ 3[3]	+ 0[5]	− 2[6]	− 6[8]	− 4[7]	− 8[9]
	8	+ 8[1]	+ 6[2]	+ 4[3]	− 4[6]	+ 2[4]	+ 0[5]	− 4[6]	− 4[6]	− 8[9]
Xerces-J (991)	4	+ 6[2]	+ 8[1]	+ 3[3]	+ 0[5]	− 2[6]	− 5[7]	+ 3[3]	− 8[9]	− 5[7]
	5	+ 8[1]	+ 6[2]	+ 2[4]	+ 4[3]	− 2[5]	− 2[5]	− 2[5]	− 6[8]	− 8[9]
	6	+ 8[1]	+ 6[2]	+ 4[3]	+ 2[4]	− 2[6]	+ 0[5]	− 4[7]	− 6[8]	− 8[9]
	7	+ 6[2]	+ 8[1]	+ 4[3]	+ 2[4]	+ 0[5]	− 4[7]	− 2[6]	− 7[8]	− 7[8]
	8	+ 8[1]	+ 6[2]	+ 4[3]	+ 2[4]	+ 0[5]	− 2[6]	− 5[7]	− 5[7]	− 8[9]

to produce a package restructuring solution which can be better from the developer's perspective. Even the proposed approach is designed to solve the LSMaOSPR problem, it can be easily extended to the other real-world LSMaOPs. As many science and engineering problems are exhibiting the nature of LSMaOPs, so the proposed approach can be beneficial to address these problems effectively. This work has exploited only a few strategies for the selection of personal best and global best, other more suitable strategies can also be used by researchers and practitioners to design the more effective metaheuristic optimizers for the more complex LSMaOPs.

8 Threats to Validity

The paper has utilized the various strategies of search-based optimization techniques for implementing the proposed approach; hence, there can be many factors that can affect the different aspects of validities related to the proposed approach. We discuss various types of threats to validity and their mitigation for our proposed approach based on the taxonomy presented by Wohlin et al. [43].

8.1 Threats to Internal Validity

The internal validity of an empirical study is concerned with the causal relationship between outcome and treatment. In particular, the outcome of experimentation must not be determined by the factors which are not under the control of researchers. In this study, the proposed approach uses various parameters that have a major influence on the generation of results. If the values of the parameters are not appropriate, the approach can generate poor results. To mitigate this threat to internal validity, we determined the parameters' values based on the trial-and-error approach which is considered an effective method in case of designing novel metaheuristic algorithms.

8.2 Threats to External Validity

The external validity is referred to the degree of generalization of the proposed approach over a larger population of

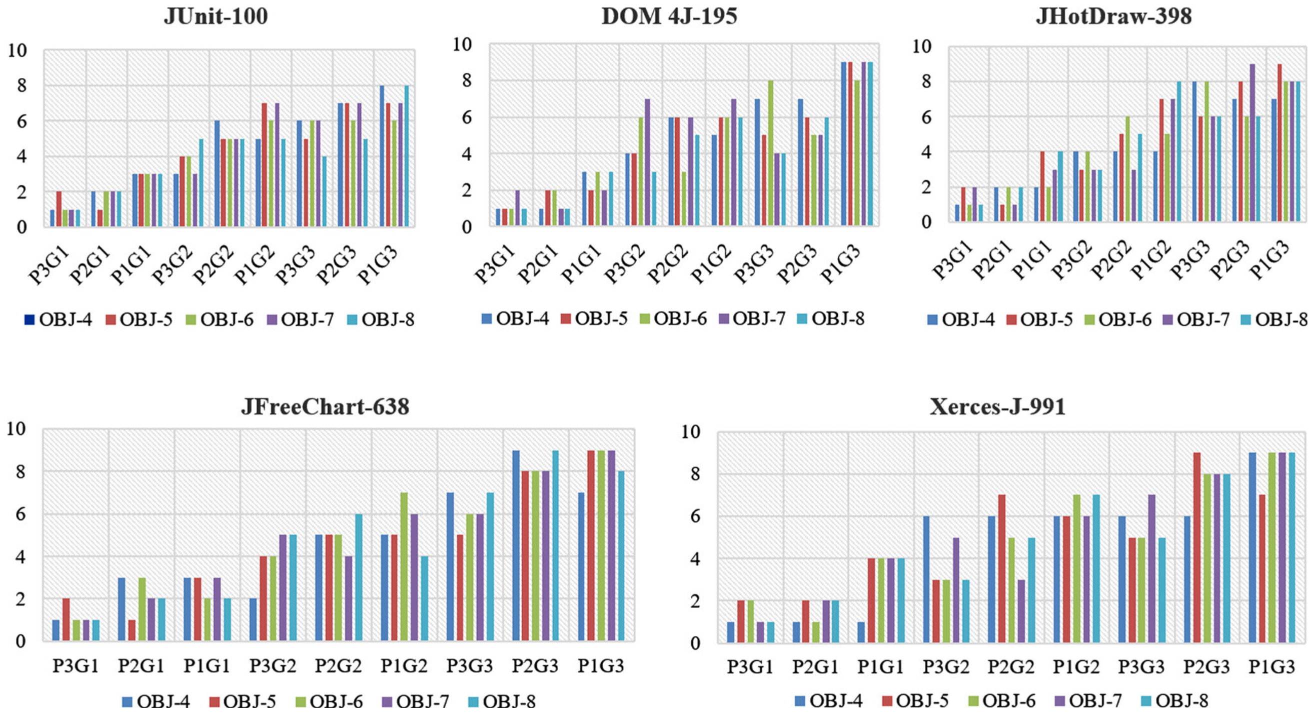

Fig. 2 Comparison of different variants in terms of IGD

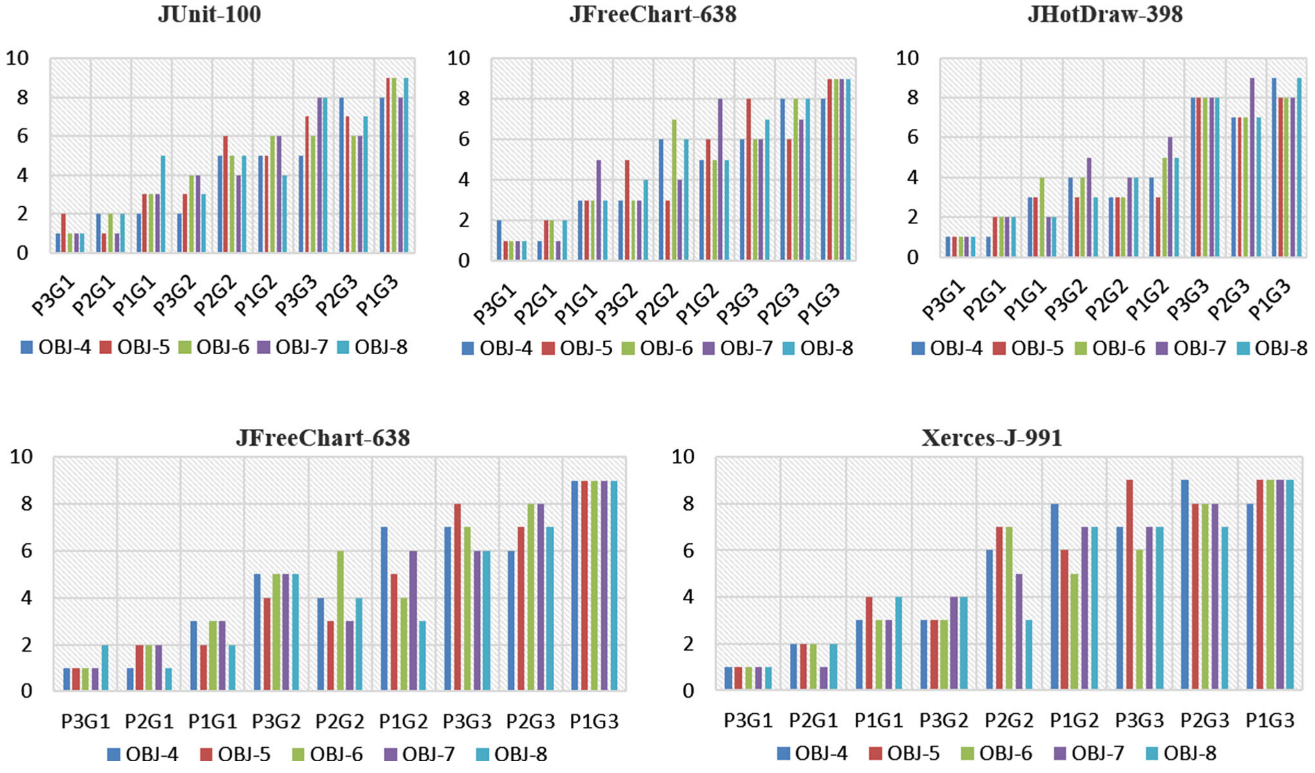

Fig. 3 Comparison of different variants in terms of HV

problem instances. The proposed approach can perform differently if applied over a synthetic problem instance having the special characteristics and selected randomly. To generalize the results over a larger population of problem instances,

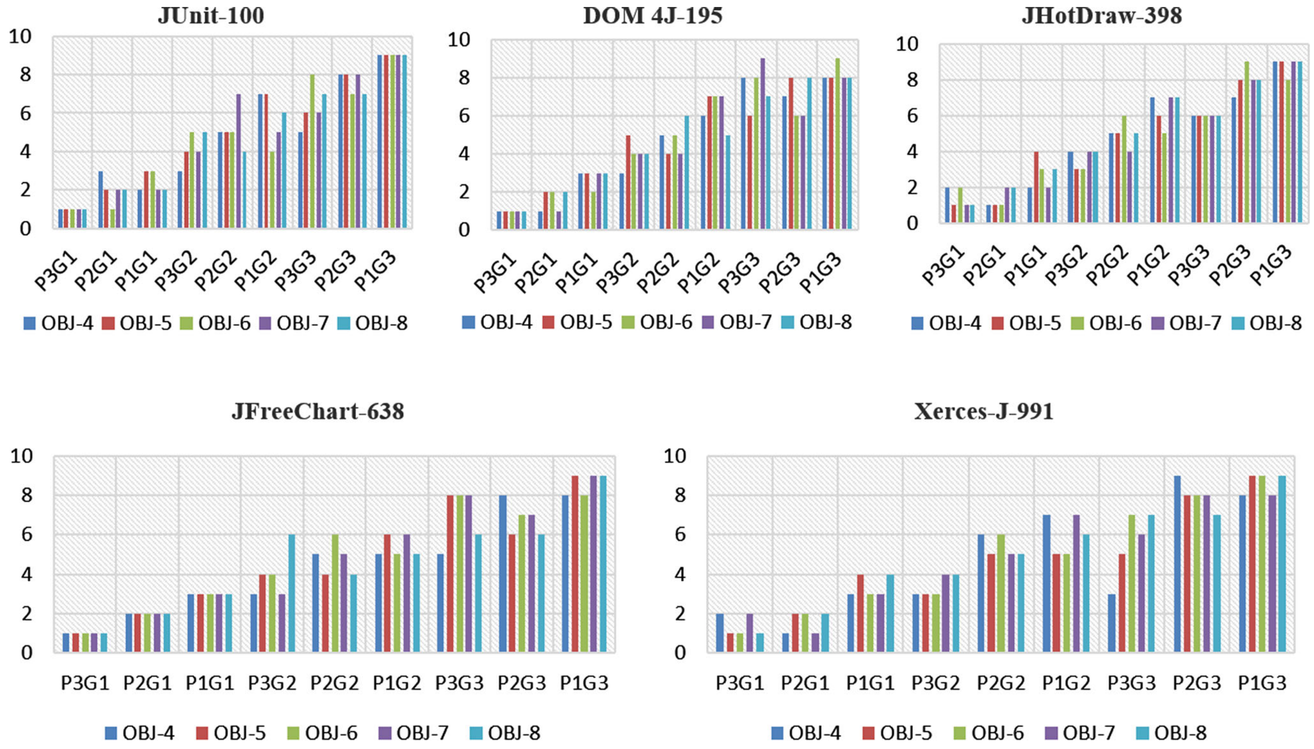

Fig. 4 Comparison of different variants in terms of MQ

it is required to select the sample of problem instances that can be representative of the larger population of problem instances. To mitigate this threat, we selected a variety of real-world problem instances covering major characteristics of the larger population of problem instances. These problem instances are designed for different classes of applications and have different complexity levels. The size of the problem instances also varied from smaller to larger.

8.3 Threats to Construct Validity

In the empirical study of any proposed approach, the theory behind the method and observations of the experimentation should be highly correlated. In particular, treatment of the study should be properly imitating the construct of the cause and the outcome of the experimentation should be properly reflecting the construct of effect. In this study, the definition of different objective functions can be considered as a threat to construct validity. To mitigate this threat, we have adopted the software quality criteria defined in the existing approaches as the objective functions.

8.4 Threats to Conclusion Validity

The conclusion validity of an empirical study is concerned with the statistical relationship between the parts involved in the experimentation. In our experimentation, the results are collected by executing each variant of the proposed approach 31 times on the same problem instance. Therefore, to compare the results of different variants, an appropriate statistical test should be used. To draw the most appropriate conclusion about the performance of the variants, we used the Mann–Whitney Wilcoxon rank-sum test. Because the Mann–Whitney Wilcoxon rank-sum test is considered to be most appropriate if the characteristics of the underlying data of evaluation are not normally distributed.

9 Conclusion and Future Work

In this work, we introduced a PSO based LSMaOA to solve the LSMaOSPR. To this contribution, we exploited a variety of strategies in customizing the various components (e.g. inertia weight, cognitive and social constants, velocity and position updating, management of non-dominated solutions in external archives, and selection of personal and global best position) of the PSO framework. Based on the different combinations of personal best and global best selection strategies, nine variants, namely P1G1, P1G2, P1G3, P2G1, P2G2, P2G3, P3G1, P2G2, and P3G3 have been designed. These variants of the proposed approach have been tested over five Java-based object-oriented software systems and the produced Pareto front of the package restructuring solutions are evaluated in terms of IGD, Hypervolume, and MQ measures. The obtained results showed that each of the variants is highly scalable to the number of objective functions and decision variables. However, the variant P3G1 is found as the best performer and P1G3 is the worst performer in most cases. Overall performance ordering of these variants is as follows: P3G1 > P2G1 > P1G1 > P3G2 > P2G2 > P1G2 > P3G3 > P2G3 > P1G3. Future works include the designing of more advanced LSMaOAs that can address many other real-world LSMaOPs.

References

1. Mkaouer, W.; Kessentini, M.; Shaout, A.; Koligheu, P.; Bechikh, S.; Deb, K.: Ouni, A: Many-objective software remodularization using NSGA-III. ACM Trans Software Eng. Methodol. **24**(3), 1–45 (2015)
2. Abdeen, H.; Ducasse, S.; Sahraoui, H.; Alloui, I: Automatic Package Coupling and Cycle Minimization. 16th Working Conference on Reverse Engineering (2009), https://doi.org/10.1109/WCRE.2009.13
3. Abdeen, H; Sahraoui, H.; Shata, O.; Anquetil, N.; Ducasse, S.:Towards automatically improving package structure while respecting original design decisions,2013 20th Working Conference on Reverse Engineering (WCRE), 212–221 (2013)
4. Chhabra, J.K.: Improving package structure of object-oriented software using multi-objective optimization and weighted class connections. J King Saud University Comput Infor Sci **29**(3), 349–364 (2017)
5. Zhang, Y.; Wang, G.G.; Li, K.; Yeh, W.C.; Jian, M.; Dong, J.: Enhancing MOEA/D with information feedback models for large-scale many-objective optimization. Inf. Sci. **522**, 1–16 (2020)
6. Hong, W.J.; Yang, P.; Tang, K.: Evolutionary computation for large-scale multi-objective optimization: a decade of progresses. Int. J. Autom. Comput. **18**, 155–169 (2021)
7. Tian, Y.; Si, L.; Zhang, X.; Cheng, R.; He, C.; Tan, K.C.; Jin, Y.: Evolutionary Large-Scale Multi-Objective Optimization: A Survey. J. ACM **54**(8), 1–34 (2021)
8. Zitzler, E.; Kunzli, S.: Indicator-based selection in multiobjective search. in Parallel Problem Solving. In: Yao, X., et al. (Eds.) Nature—*PPSN VIII* (LNCS 3242), pp. 832–842. Springer, Heidelberg (2004)
9. Zhang, Q.; Hui, L.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2008)
10. Zhang, X.; Tian, Y.; Jin, Y.: A knee point driven evolutionary algorithm for many-objective optimization. IEEE Trans. Evol. Comput. **19**(6), 761–776 (2014)
11. Deb, K.; Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. IEEE Trans. Evol. Comput. **18**(4), 577–601 (2014)
12. Ma, L., et al.: A novel many-objective evolutionary algorithm based on transfer matrix with Kriging model. Inf. Sci. **509**, 437–456 (2020)
13. Tang, K.; Li, X.; Suganthan, P.; Yang, Z.; Weise, T.: Benchmark functions for the CEC 2008 special session and competition on large scale global optimization, December (2009)
14. Mahdavi, S.; Shiri, M.E.; Rahnamayan, S.: Metaheuristics in large-scale global continues optimization: a survey. Inf. Sci. **295**, 407–428 (2015)
15. Antonio, L.M.; Coello, C.A.C.: Use of cooperative coevolution for solving large scale multi-objective optimization problems. In: 2013 IEEE Congress on Evolutionary Computation, pp. 2758–2765 (2013)
16. Ma, X., et al.: A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables. IEEE Trans. Evol. Comput. **20**(2), 275–298 (2016)
17. Song, A.; Yang, Q.; Chen, W.; Zhang, J.: A random-based dynamic grouping strategy for large scale multi-objective optimization. In: 2016 IEEE congress on evolutionary computation (CEC), pp. 468–475 (2016)
18. Zhang, X.; Tian, Y.; Cheng, R.; Jin, Y.: A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization. IEEE Trans. Evol. Comput. **22**, 99 (2016)
19. Wang, Q.; Zhang, L.; Wei, S.; Li, B.: Tensor decomposition-based alternate sub-population evolution for large-scale many-objective optimization. Inf. Sci. **569**, 376–399 (2021)
20. Gu, Z.M.; Wang, G.G.: Improving NSGA-III algorithms with information feedback models for large-scale many-objective optimization. Futur. Gener. Comput. Syst. **107**, 49–69 (2020)
21. Zille, H.; Ishibuchi, H.; Mostaghim, S.; Nojima, Y.: Framework for large-scale multiobjective optimization based on problem transformation. IEEE Trans. Evol. Comput. **22**(2), 260–275 (2018)
22. Zhang, X.; Tian, Y.; Cheng, R.; Jin, Y.: A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization. IEEE Trans. Evol. Comput. **22**(1), 97–112 (2018)
23. LaTorre, A.; Muelas, S.; Peña, J.M.: A comprehensive comparison of large scale global optimizers. Inf. Sci. **316**, 517–549 (2015)
24. Yang, P.; Tang, K.; Yao, X.: Turning high-dimensional optimization into computationally expensive optimization. IEEE Trans. Evol. Comput. **22**(1), 143–156 (2018)
25. Akopov, S.A.; Beklaryan, L.A.; Thakur, M.; Verma, B.D.: Parallel multi-agent real-coded genetic algorithm for large-scale black-box single-objective optimisation. Knowledge-Based Sys **174**, 103–122 (2019)
26. Ma, L.; Huang, M.; Yang, S.; Wang, R.; Wang, X.: An adaptive localized decision variable analysis approach to large-scale multiobjective and many-objective optimization. IEEE Trans Cybernet (2021). https://doi.org/10.1109/TCYB.2020.3041212
27. Cao, B.; Zhang, Y.; Zhao, J.; Liu, X.; Skonieczny, L.; Lv, Z.: Recommendation based on large-scale many-objective optimization for the intelligent internet of things system. IEEE Internet Things J (2021). https://doi.org/10.1109/JIOT.2021.3104661
28. Cheng, R.; Jin, Y.; Olhofer, M.; Sendhoff, B.: Test problems for large-scale multiobjective and many-objective optimization. IEEE Trans Cybernet **47**(12), 4108–4121 (2017)
29. Prajapat, A.; Kumar, S.: PSO-MoSR: a PSO-based multi-objective software remodularization. Int J Bio-Inspired Computat **15**(4), 254–263 (2020)
30. Kirkpatrick, S., Jr.; Gelatt, C.D.; Vecchi, M.P.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983)
31. Mancoridis, S.; Mitchell, B.S.; Rorres, C.; Chen, Y.-F.; Gansner, E.R.: Using automatic clustering to produce high-level system organizations of source code. Proc. Int’l Workshop program comprehension, pp. 45–53 (1998)
32. Praditwong, K.; Harman, M.; Yao, X.: Software module clustering as a multi-objective search problem. IEEE Trans Software Eng **37**(2), 264–282 (2011)
33. Amarjeet; Chhabra, J.K: FP-ABC: Fuzzy-Pareto dominance driven artificial bee colony algorithm for many-objective software module clustering. Computer Languages, Systems & Structures, 15:1–21 (2018)

34. Ting, T.; Shi, Y.; Cheng, S.; Lee, S.:Exponential inertia weight for particle swarm optimization, In: Advances in swarm intelligence, Springer, (2012)
35. Liu, H.; Zhang, X.W.; Tu, L.P.: A modified particle swarm optimization using adaptive strategy. Expert Sys App **152**, 113353 (2020)
36. Zitzler, E.; Thiele, L.: Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. IEEE Trans. Evol. Comput. **3**(4), 257–271 (1999)
37. Wang, H.; Jiao, L.; Yao, X.: Two_Arch2: an improved two-archive algorithm for many-objective optimization. IEEE Trans. Evol. Comput. **19**(4), 524–541 (2015)
38. Wang, G.; Tan, Y.: Improving metaheuristic algorithms with information feedback models. IEEE Trans. Cybern. **49**(2), 542–555 (2019)
39. Yang, S.; Li, M.; Liu, X.; Zheng, J.: A grid-based evolutionary algorithm for many-objective optimization. IEEE Trans. Evol. Comput. **17**(5), 721–736 (2013)
40. Köppen, M.; Vicente-Garcia, R.: A fuzzy scheme for the ranking of multivariate data and its application. In: Proceedings of annual meeting of the north american fuzzy information processing society; 140–155 (2004)
41. Zitzler, E.; Thiele, L.: Multi-objective evolutionary algorithms: A comparative case study and the strength Pareto approach. IEEE Trans. Evol. Comput. **3**, 257–271 (1999)
42. Goh, C.K.; Tan, K.C.: Evolving the Tradeoffs between Pareto-Optimality and Robustness in Multi-Objective Evolutionary Algorithms. In: Yang, S.; Ong, Y.S.; Jin, Y. (Eds.) Evolutionary Computation in Dynamic and Uncertain Environments Studies in Computational Intelligence. Springer, Berlin (2007)
43. Wohlin, C.; Runeson, P.; Höst, M.; Ohlsson, M.C.; Regnell, B.; Wesslén, A.: Experimentation in software engineering. Springer, Berlin (2012)