Hindawi
Complexity
Volume 2021, Article ID 3974635, 11 pages
https://doi.org/10.1155/2021/3974635

Research Article

Entropy-Driven Global Best Selection in Particle Swarm Optimization for Many-Objective Software Package Restructuring

Amarjeet Prajapati,[1] **Anshu Parashar**,[2] **Sunita**,[3] **and Alok Mishra**[4]

[1] *Computer Engineering & IT Department, JIIT, Noida, India*
[2] *Department of Computer Science and Engineering, TIET, Patiala, Punjab, India*
[3] *Computer Science & Engineering Department, CCET, Chandigarh, India*
[4] *Informatics and Digitalization, Molde University College -Specialized University in Logistics, Molde, Norway*

Correspondence should be addressed to Alok Mishra; alok.mishra@himolde.no

Received 12 October 2021; Revised 18 November 2021; Accepted 22 November 2021; Published 7 December 2021

Academic Editor: Zeljko Stevic

Many real-world optimization problems usually require a large number of conflicting objectives to be optimized simultaneously to obtain solution. It has been observed that these kinds of many-objective optimization problems (MaOPs) often pose several performance challenges to the traditional multi-objective optimization algorithms. To address the performance issue caused by the different types of MaOPs, recently, a variety of many-objective particle swarm optimization (MaOPSO) has been proposed. However, external archive maintenance and selection of leaders for designing the MaOPSO to real-world MaOPs are still challenging issues. This work presents a MaOPSO based on entropy-driven global best selection strategy (called EMPSO) to solve the many-objective software package restructuring (MaOSPR) problem. EMPSO makes use of the entropy and quality indicator for the selection of global best particle. To evaluate the performance of the proposed approach, we applied it over the five MaOSPR problems. We compared it with eight variants of MaOPSO, which are based on eight different global best selection strategies. The results indicate that the proposed EMPSO is competitive with respect to the existing global best selection strategies based on variants of MaOPSO approaches.

1. Introduction

The transformation of science and engineering problem into a search-based optimization problem provides a great opportunity to utilize the potential of different metaheuristic algorithms [1]. In the last three decades, a large variety of metaheuristic algorithms addressing the different classes of optimization problems have been proposed (e.g., [2–6]). As the different classes of optimization problems possess a different nature and complexity, they pose different challenges in designing the optimization algorithms. Based on the challenges posed by the different classes of optimization problems, the different types of metaheuristic algorithms have been designed [7]. Furthermore, the formulation of large and complex real-world problems as the optimization problems encourages researchers and practitioners to design the development of more variety of optimization approaches [8].

The different optimization problems can treat different goals as single objective or different independent objective functions for the optimization. Hence, based on the consideration of number of objective functions as a single or independent objectives, the optimization problems are generally classified as the single-objective optimization (i.e., single objective to be optimized), multi-objective optimization (i.e., more than objectives to be optimized), and many-objective optimization (i.e., more than three objectives to be optimized, a special case of multi-objective optimization) [9]. Similarly, the optimization approaches designed to solve these optimization problem classes can be classified as single-objective optimization approach, multi-objective optimization approach, and many-objective optimization approach, respectively.

It has been found that most of the real-world optimization problems usually require more than three objectives to be optimized independently to generate the solution. For example, industrial task scheduling [10], designing engineering models for different purposes [11], clustering of the software system to recover the software architecture [12], optimization of hybrid car controller [13], calibration optimization of the automotive [14], and improving existing software package structure [2] require more than three objectives to be optimized simultaneously to produce the solution. Such real-world many-objective optimization problems (MaOPs) pose several challenges in the designing the metaheuristic optimization approach that can address them effectively and efficiently.

To address the different types of synthetic MaOPs, a variety of many-objective optimization approaches based on the different frameworks of metaheuristic algorithms have been proposed (e.g., MOEA/D [15] and NSGA-III [3]). To solve the real-world MaOPs (e.g., many-objective software remodularization), several researchers and practitioners have customized the existing metaheuristic optimization algorithms (e.g., many-objective software remodularization using NSGA-III by [8]). Recently, particle swarm optimization (PSO) based on many-objective optimization algorithms has been widely explored to address the different science and engineering MaOPs (e.g., [16–20]). In the PSO-based many-objective optimization, the selection of leaders (i.e., personal best or global best) has the major influence on the generation of well-distributed and converged approximation of the Pareto front. Therefore, the major challenge of the PSO-based many-objective optimization algorithm is designing the effective leader selection strategies corresponding to the large and complex real-world MaOPs.

The different strategies designed to select the leaders in many-objective PSO algorithms have different influences on the convergence and diversity of the final approximation of the Pareto front. The PSO-based many-objective optimization algorithms exploited the different leader selection strategies (e.g., distance ranking scheme [21], virtual inverted generational distance indicator [22], grid-based selection [18], vector angle selection [20], and fuzzy Pareto dominance [23]). Even the leader selection strategies adapted in the different existing PSO-based many-objective optimization algorithms are effective. Still, there is huge scope in their improvement corresponding to the different types of real-world MaOPs.

Recently, the software remodularization problem of software engineering field has been widely treated as the MaOP and solved its different aspects using the different types of many-objective optimization algorithms (e.g., [2, 8, 9]). However, there are many other aspects of software remodularization problems in software engineering and such software package restructuring problems have not been sufficiently explored. In particular, the many-objective optimization aspect of software package restructuring problem (called MaOSPR problem) gained little attention. This work presents entropy-driven many-objective particle swarm optimization (MaOPSO) (called EMPSO) to solve the MaOSPR problem. Towards this contribution, EMPSO exploits the synergy of the indicator-based [24] and entropy-based [25] ranking approaches to select the leader (especially global best particle). The use of both ranking schemes together in evaluating the global best helps in maintaining diversity and convergence simultaneously. Particularly, the entropy-based ranking scheme helps in maintaining the diversity, whereas the indicator-based ranking scheme promotes convergence.

To justify the potential of proposed EMPSO, we compared it with other variants of many-objective PSO (MPSO) that uses other global best selection strategies. These are, i.e., grid-based strategy (GMPSO) [26], fuzzy Pareto dominance strategy (FMPSO) [27, 28], indicator-based approach (IMPSO) [24], distance-based ranking strategy (DMPSO) [21], b-dominance strategy (BMPSO-6) [29], ε-dominance strategy (SMPSO) [30], α-dominance strategy (AMPSO) [31], and weighted average ranking strategy (WMPSO) [32]. These variants of the MPSO have been using the same configuration and parameter setting as of the EMPSO.

The rest part of the study is presented as follows. Section 2 covers materials and methods including the related work mainly focusing on search-based software package restructuring and many-objective approaches, problem modelling, and a detailed description of the working of the proposed approach along with its variants. Section 3 discusses the experimental setup and problem selection and presents results obtained through the different variants of the proposed approach including internal and external threats to validity and their mitigations. Section 4 provides the conclusion of the work and suggests some future works.

2. Materials and Methods

The past few decades have witnessed the rapid development of PSO [33]-based metaheuristic optimization approaches due to wide application in solving the various optimization problems (e.g., [4, 34–36]). The easy implementation and effective optimization potential of the PSO algorithm make it more attractive in research and practice. In this regard, the PSO algorithm has also gained wide attention in solving the real-world optimization problems of different fields (e.g., [37–39]). The broad applicability of the PSO algorithm has encouraged researchers and practitioners to design its multiple variants to address the different aspects of the optimization problems. The variants of the PSO algorithms can be classified into four groups: 1) hybridization with other metaheuristic algorithms (e.g., Ding et al. 2021); 2) modifying and adaptive parameters (e.g., Hop et al. 2021); 3) varying topologies for the velocity updation (e.g., [40]); and 4) different learning strategies [41].

2.1. Many-Objective PSO. It has been commonly observed that the traditional multi-objective optimization algorithms face several performance challenges if applied over the many-objective optimization problems. Similar to the performance challenges faced by conventional multi-objective optimization over the many-objective optimization problems, the traditional multi-objective PSO approaches also face similar performance challenges. To overcome the performance challenges of the conventional multi-objective PSO caused by the many-objective optimization problems, a variety of many-objective PSO approaches have been proposed [16, 17, 20, 42, 43].

The major challenge of the traditional multi-objective PSO over the many-objective optimization problem is the selection of personal best and global best [22]. The existing many-objective PSO approaches have adopted different strategies to select the individual's best and global best. Mostaghim et al. [21] used the newest method [44] for the personal best selection and designed a distance-based ranking strategy to determine the global best. Hu et al. [42] exploited the concept of Pareto dominance for the selection of personal best and parallel cell coordinate approach with density estimation of non-dominated solution for the selection of the global best.

Wu et al. [22] adopted the concept of reference points to compute the contribution degree of each candidate solution of the population. Further, based on the highest contribution degree, the personal best and global best are selected. Li et al. [45] used grid-based ranking approach to improve the discriminability of the particles in many-objective PSO to choose personal best and global best. Leung et al. [19] proposed a Hybrid Global Leader Selection Scheme (HGLSS) using the concepts of space expanding strategy (SES) and Euclidean distance strategy (EDS). Yang et al. [20] designed a vector angle and decomposition-based method to select the personal best and global best for the many-objective PSO approach.

Even the personal best and global best selection strategies adapted in the existing PSO-based many-objective optimization algorithms have been satisfactory in guiding the optimization process. Still, there is massive scope in their improvement. The major limitation of the abovementioned PSO-based many-objective optimization algorithms is that most approaches use either convergence or divergence properties of the search space in designing the global best selection strategy. Such global best selection strategy types may not guide the optimization process towards the well-distributed and converged approximation of the Pareto front. Moreover, such kinds of PSO-based many-objective optimization algorithms require more specific strategies at the different other selection points such as personal best selection and archive maintenance to curb the issue caused by the global best selection strategy.

2.2. Problem Modelling. SPR problem is a significant optimization problem of software reengineering that can be viewed and modelled from different perspectives. This study models the SPR problem as a MaOO problem where multiple (generally more than 3) package restructuring objectives are optimized to produce restructuring solutions. Further, the MaOO model of SPR problem assumes that the software system needed a complete package restructuring, i.e., creating software package structure from scratch where the number and size of the package in the produced SPR can be different from the original package structure. To formulate the MaOO model of the SPR problem, the associated decision variables and the objective functions need to be appropriately encoded and defined.

The encoding of the SPR problem in terms of the decision variables is an important task. In this work, we adopted the integer-based representation scheme to encode the SPR problem as suggested by the previous researchers (e.g., [46, 47]). In this encoding scheme, the software system is first transformed into a weighted class dependency graph (WCDG) and then mapped with the integer vector. Particularly in this encoding scheme, an integer vector of size n is created (n is the number of classes of the software system), and its indices are mapped with the different classes. The packages in which the classes are to be placed are mapped with the integer values (i.e., 1,2,. . ., n). The integer vector-based encoding scheme used for the SPR problem is demonstrated in Figure 1.

After defining the decision variables, we need to define the objective functions for the SPR problem. The primary goal of the objective functions in MaOO model of SPR problem is to evaluate the quality of package restructuring solutions and guide the optimization process towards the optimal SPR solutions. Software coupling and cohesion are the two essential measures widely used to evaluate the quality of SPR solutions. The definition of coupling and cohesion can vary based on the information (e.g., structural, lexical, and changed history) used in the computation. It has been observed that all three types of source code information, i.e., structural, lexical, and changed history, play an important role to define the objective functions of the SPR problems [46, 48].

Therefore, this work considers all coupling and cohesion designed in terms of structural, lexical, and changed history information as objective functions. The definition of these objectives is based on the studies presented in [8, 46, 48]. The objectives 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). To avoid the generation of very small packages or very large packages, we have included two more objectives: 7) number of packages (to maximize) and 8) difference between the minimum and maximum numbers of classes in the packages (to minimize).

2.3. Proposed Approach. The basic framework of the proposed many-objective PSO approach is presented in Figure 2. It mainly contains five major components: 1) initialization of position and velocity of particles in the swarm, 2) updation of the external archive, 3) updation of personal best position, 4) updation of global best position, and 5) updation of current velocity and position of the particles in the swarm. The details of each component are provided in the subsequent subsections.

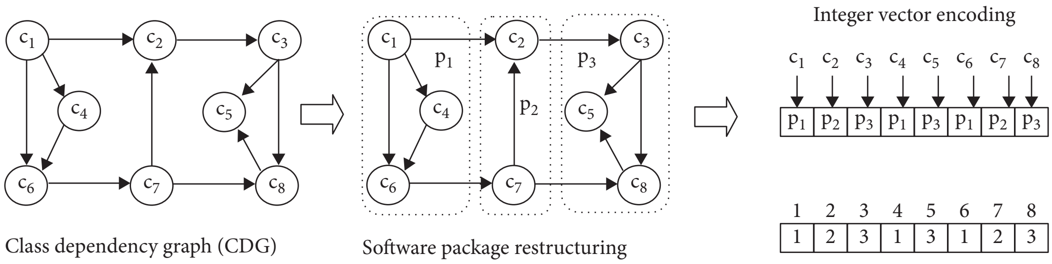


Figure 1: Integer vector-based encoding of SPR problem.

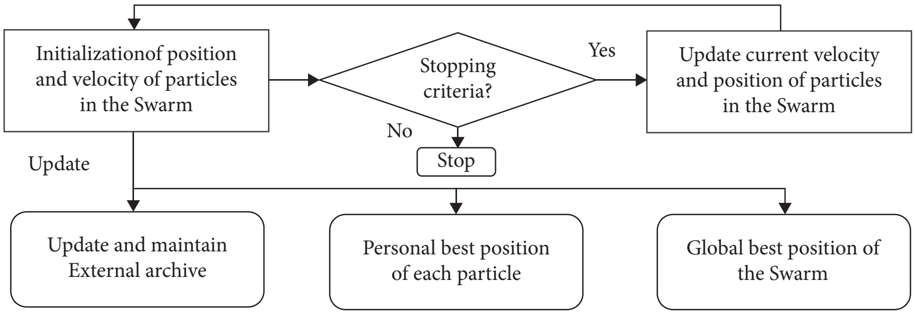

Initializationof position and velocity of particles in the Swarm
Stopping criteria?
Yes
Update current velocity and position of particles in the Swarm
No
Stop
Update
Update and maintain External archive
Personal best position of each particle
Global best position of the Swarm

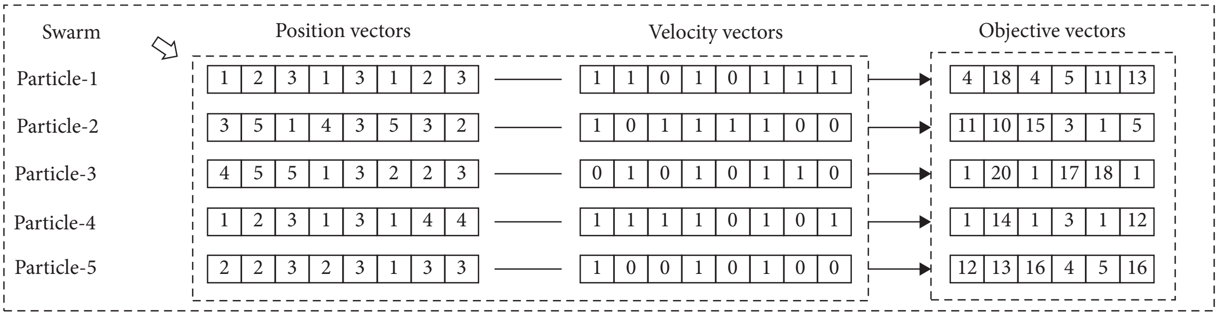
Figure 2: General framework of the proposed approach.

2.3.1. Swarm Initialization. In the initialization phase of the proposed work, the position vectors and the velocity vectors associated with each particle of the swarm are initialized. The position vector of the particle in the optimization problem is viewed as the candidate solution. In our optimization problem, the integer vector encoding approach is used to represent the candidate solution; hence, the position vector of each particle is denoted with the integer vector encoding. The velocity vector is encoded as a binary vector with the same size as the position vector. The initialization of the position vector and velocity vector of all the swarm particles is initialized using the randomization method. This is the simple and more effective approach used to initialize the candidate solutions of the population of evolutionary and swarm-based heuristic algorithms. The demonstration of position and velocity vector initialization is given in Figure 3. Just for demonstration, we have taken six objectives (i.e., assumed obj1, obj2, obj3, obj4, obj5, and obj6 are to be maximized) and their values are randomly assigned.

2.3.2. Maintaining External Archive. The concept of external archive is widely used to collect the useful elitist non-dominated solutions found in every generation of proposed many-objective PSO. To maintain the non-dominated solutions in the external archive, there are two strategies: insertion and removal. In this approach, the non-dominated candidates from the swarm are first collected and stored in the external archive. Next, the removal strategy is applied if the external archive overflows. The overall process of both strategies is as follows: 1) the non-dominated candidates from the swarm are collected, one by one; 2) the new non-dominated candidate can be part of the external archive if no candidate solution of external archive dominates it; 3) if new non-dominated candidate dominates some candidate solutions of the external archive, it should place in the external archive, and the dominated candidates of the external archives are deleted; 4) if the external archive overflows, the candidate removal strategy is applied; and 5) the candidate solutions of the external archive having the minimum Euclidean distance are deleted one by one until the size of the external archive.

2.3.3. Selection of Personal. Most PSO algorithm variants typically use personal best experience (Pbest) and the swarm best experience (Gbest) to update the particle's velocity and position. Such variants of the PSO face some potential challenges, such as premature convergence and loss of diversity. Therefore, it becomes crucial to design an effective learning strategy that can help in improving diversity and convergence. For our Java package restructuring problem, a class of discrete combinatorial optimization problems, we have designed an effective approach for selecting personal and global best positions.

The following method is used to determine the personal best position (Pbest) of a particular particle. At the beginning (i.e., initialization phase), the current position of each particle is assumed as their personal best position. In the further iterations, if the newly updated position of the particle dominates the personal best position, then the current personal best position will be replaced with the new position; otherwise, the current personal best position will remain unchanged. If both the newly generated position and the current best position are non-dominated with each other, then a solution from both solutions as a personal best is randomly selected.



Swarm	Position vectors	Velocity vectors	Objective vectors
Particle-1	1 2 3 1 3 1 2 3	1 1 0 1 0 1 1 1	4 18 4 5 11 13
Particle-2	3 5 1 4 3 5 3 2	1 0 1 1 1 1 0 0	11 10 15 3 1 5
Particle-3	4 5 5 1 3 2 2 3	0 1 0 1 0 1 1 0	1 20 1 17 18 1
Particle-4	1 2 3 1 3 1 4 4	1 1 1 1 0 1 0 1	1 14 1 3 1 12
Particle-5	2 2 3 2 3 1 3 3	1 0 0 1 0 1 0 0	12 13 16 4 5 16

Figure 3: Demonstration of swarm initialization.

2.3.4. Selection of Global Best.

The selection of global best in the PSO algorithm is an important task. It has become more crucial in case of the multi-/many-objective PSO algorithm. In many-objective PSO, it becomes more challenging to select a global best particle that can lead the optimization process towards well-distributed approximation of Pareto front. To incorporate the convergence and diversity behavior in the global best selection, we adapted the indicator-based [24] and entropy-based [25] ranking approaches together in the selection process of the global best. For this, we utilize the candidate solutions of the external archive because its candidate solutions are the best non-dominated solutions. To compute the fitness of the candidates of the external archive, we use both quality indicator [24] and entropy [25] collectively. To compute indicator$_{\text{rank}}$, which is based on the $I_{\in +}$, an indicator is defined as follows:

$$
\begin{aligned}
I_{\varepsilon+}(p,q) &= \operatorname*{arg\,min}_{\varepsilon}\left(f_i(p) - \varepsilon \le f_i(q),\ i \in \{1,2,\ldots,m\}\right),\\
\text{indicator}_{\text{rank}}(p) &= \sum_{q\in S/\{p\}} -e^{-I_{\varepsilon+}(q,p)/\alpha},
\end{aligned}
\tag{1}
$$

where $I_{\varepsilon+}(p,q)$ denotes the minimum value required so that candidate solution p dominated candidate solution q in objective space. To compute the entropy-based ranking entropy$_{\text{rank}}$ of the candidate solution p of the external archive, we define it in terms of the Shannon entropy.

$$H = -\sum_{i=1}^{m} d_i \times \log(d_i), \text{where } d_i = \frac{k}{m}. \tag{2}$$

The symbol m denotes the number of candidate solutions of the external archive and k is the number of fitness values in the ith segment. To compute the entropy$_{\text{rank}}$ of particular candidate of the external archive, the segment rank where the candidate resides is used. Now, based on the indicator$_{\text{rank}}$ and the entropy$_{\text{rank}}$, we quantify the importance of each candidate solution of the external archive as follows:

$$R(i) = \alpha \times \text{indicator}_{\text{rank}}(i) + \beta \times \text{entropy}_{\text{rank}}(i). \tag{3}$$

The small value of $R(i)$ indicates the good performance of particle i, and therefore, the particle in the external archive having the smallest $R(i)$ is considered as the global best particle. The value of α and β is used to control the contribution of indicator$_{\text{rank}}$ and the entropy$_{\text{rank}}$, respectively.

2.3.5. Updation of Velocity.

The evolution of the swarm from one iteration to iteration requires the updation of velocity and position of each particle of the swarm. The updation rule of the velocity and position depends on the nature and encoding of the velocity and position vectors. In this approach, we use the following method to update each swarm particle's velocity vector and position vector.

$$
\begin{aligned}
v_i^{\text{new}} &= \Omega\left(\omega \times v_i + c_1 r_1 \times \left(\text{pbest}_i \oplus p_i\right) + c_2 r_2 \times \left(\text{gbest} \oplus p_i\right)\right),\\
p_i^{\text{new}} &= p_i \Theta\, v_i^{\text{new}},
\end{aligned}
\tag{4}
$$

where ω represents inertia weight. The $\{c_{1,} c_{2,}\}$ and $\{r_1, r_2\}$ are the learning factors and random numbers, respectively. The v_i, p_i, pbest$_{\text{i}}$, and gbest represent the current velocity, current position, personal best position, and global best position of the ith particle of the swarm—the operators ×, +, and ⊕ present multiplication, addition, and XOR. The function $\Omega(.)$ returns a velocity vector. Figure 4 depicts the new velocity computation.

The value of inertia weight ω plays an important role in exploring and exploiting the search space during the optimization process. The high value of ω enhances the exploration (i.e., global search) capability of the optimization algorithm, and the low value of ω boosts the exploitation (i.e., local search) capability [49]. In literature, various strategies have been suggested to balance the exploration and exploitation capabilities of the optimization algorithms. In our approach, we use the nonlinear decreasing strategy of the inertia weight [49]. It is the most appropriate inertia weight assignment for our approach, because in the beginning, there require more exploration and later more exploitation. The following equation is used to update the value of inertia weight:

$$\omega(t) = \omega_{\max} - \left(\frac{t-1}{t_{\max}-1}\right)^{\alpha}\left(\omega_{\max} - \omega_{\min}\right), \tag{5}$$

where t represents the current iteration number, $t_{\max}$ is the maximum number of iterations decided for the algorithm, $\omega_{\min}$ and $\omega_{\max}$ are the minimum and maximum inertia weight values, respectively, and α denotes the decline exponent. The value of the cognitive coefficient (c_1) and the social coefficient (c_2) also has an essential effect on the exploitation and exploration of the search space. It has been observed that the linear variation in cognitive coefficient

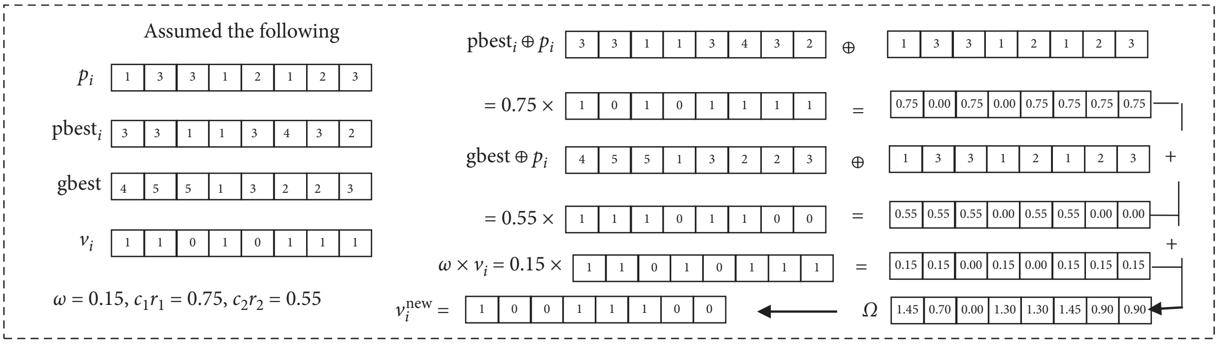

Assumed the following
p_i 1 3 3 1 2 1 2 3
pbest_i 3 3 1 1 3 4 3 2
gbest 4 5 5 1 3 2 2 3
v_i 1 1 0 1 0 1 1 1
$\omega = 0.15, c_1r_1 = 0.75, c_2r_2 = 0.55$
$\text{pbest}_i \oplus p_i$ 3 3 1 1 3 4 3 2 ⊕ 1 3 3 1 2 1 2 3
= 0.75 × 1 0 1 0 1 1 1 1 = 0.75 0.00 0.75 0.00 0.75 0.75 0.75 0.75
$\text{gbest} \oplus p_i$ 4 5 5 1 3 2 2 3 ⊕ 1 3 3 1 2 1 2 3
+
= 0.55 × 1 1 1 0 1 1 0 0 = 0.55 0.55 0.55 0.00 0.55 0.55 0.00 0.00
+
$\omega \times v_i = 0.15 \times$ 1 1 0 1 0 1 1 1 = 0.15 0.15 0.00 0.15 0.00 0.15 0.15 0.15
$v_i^{\text{new}} =$ 1 0 0 1 1 1 0 0
Ω 1.45 0.70 0.00 1.30 1.30 1.45 0.90 0.90

FIGURE 4: Demonstration of new velocity computation.

(decreasing) and social coefficient (increasing) over algorithm iteration helps the optimization process in the smooth transition from exploration to exploitation [50].

2.3.6. Updation of Position. It is commonly known that different strategies used in updating the current position of a particle have a different influence on the exploration and exploitation capabilities of the swarm optimization algorithms [51]. The strategies have been found more appropriate to balance the global search (i.e., exploration) and the local search (i.e., exploitation). In this proposed approach, we use an approach where the new position is determined based on the individuals of the current swarm, individuals of the external archives, and the new velocity of the particle. In this work, to update the particle's position, we use the approach suggested by Liu et al. [51].

2.3.7. Variants of the MPSO. To validate the effectiveness of the entropy-based global best selection method of our proposed EMPSO, we compare it with other possible global best selection strategies. In the literature of many-objective optimization, a variety of selection methods have been proposed. These selection methods are used to select a candidate solution from the pool of non-dominated candidate solutions based on the convergence and diversity criteria. In this work, we have considered the eight existing selection strategies and created eight variants of many-objective PSO. A brief description of created variants of the many-objective PSO is given as follows:

(1) GMPSO: In this variant of the proposed approach, we use the grid-based strategy [26] to select the global best position from the external archives. The grid-based selection strategy influences both convergence and diversity of the algorithm because it considers both while determining the fitness value for selecting a candidate solution.

(2) FMPSO: In this variant of the proposed approach, we use the fuzzy Pareto dominance strategy [27, 28] to select global best position from the external archives. In this selection strategy, fuzzy Pareto dominance-based fitness values for the non-dominated candidate solutions are computed, and based on the fitness value, each candidate solution is ranked.

(3) IMPSO: In this variant of the proposed approach, we use an indicator-based approach [24], 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. Furthermore, based on the degree of dominance the fitness of the nondominated solution is computed. The nondominated solutions are ranked and selected.

(4) DMPSO: In this variant of the proposed approach, we use a distance-based ranking strategy [21]. In this ranking strategy, the candidate solutions are ranked by considering the distances among the solutions based on their objective space.

(5) BMPSO: In this variant of the proposed approach, we use the b-dominance strategy [29]. This strategy is a generalized form of the Pareto domination-based box vectors. To compute its value, the whole objective space is divided into hyper-boxes based on the predefined parameter ε [30].

(6) SMPSO: In this variant of the proposed approach, we use the ε-dominance strategy [30]. This strategy makes the consideration of convergence and divergence properties simultaneously. Here, we use the additive ε-dominance, which is used to extend the Pareto dominance.

(7) AMPSO: In this variant of the proposed approach, we use the α-dominance strategy [31]. This strategy is a relaxed form of the strict Pareto domination with the weak Pareto front consideration. The main idea behind the α-dominance strategy is to set lower bounds of trade-off rates between two objectives.

(8) WMPSO: In this variant of the proposed approach, we use a weighted average ranking strategy [32], a simplistic approach to distinguish the non-dominated solutions. In this strategy, a non-dominated solution is provided with a rank in terms of the number of better objectives than the other non-dominated solutions.

3. Results and Discussions

These problem instances are complex open-source software systems and developed for different applications. An empirical study over the five Java-based software projects is conducted to validate the different variants of the many-objective PSO approach. The selected software projects as the test problems are as follows: JUnit, DOM 4J, JHotDraw, JDI-Ford, and Xerces-J.

The brief information regarding these systems is provided in Table 1.

Each variant of the many-objective PSO approach is applied over each selected software project, and SPR results are collected. In particular, each of the variants is applied 31 times on each of the software projects. The performance of each variant is evaluated in terms of inverted generational distance (IGD) [52], modularization quality (MQ) [47], and hypervolume (HV) [53] quality metrics. The performance of each variant is also compared with each other. As these variants have the nature of randomness (i.e., stochastic optimization), we need some statistical tests to compare these approaches effectively.

The Mann–Whitney Wilcoxon rank-sum test is the most widely used approach for comparing the results of metaheuristic search optimizers. Therefore, we use this statistical method to compare the different approaches. For this approach, we have taken the value of confidence level and level of significance 5% and 95%, respectively. Since the PSO-based optimization approaches are stochastic, collecting results and performing an effective assessment for such a metaheuristic search optimizer are challenging. For the proposed approach, first, we execute each of the proposed variants 31 times over each of the selected problem instances and maintain 31 Pareto fronts (i.e., best non-dominated solutions so far). Next, the IGD, hypervolume, and MQ values of each of the Pareto front obtained from the different variants of the proposed approach are computed. Then, we applied the Mann–Whitney Wilcoxon rank-sum test over the results. Finally, comparative results of each variant are provided in the form of Mann–Whitney differences [ranks] corresponding quality measures, i.e., IGD, hypervolume, and MQ. The difference value in the Mann–Whitney differences [ranks] for a particular approach represents how many times the approach is performing significantly better and how many times it performs significantly worse than the rest of the approaches. The rank values signify the ordering position of each approach computed in terms of differences. The variants of the proposed approach having the smaller rank values are better than those having larger rank values. The differences and rank values of the variants EMPSO, GMPSO, FMPSO, IMPSO, DMPSO, BMPSO, SMPSO, AMPSO, and WMPSO computed in terms of IGD, hypervolume, and MQ measure are demonstrated in Tables 2–4, respectively.

The difference [rank] values computed in terms of IGD presented in Table 2 shows that only some variants can achieve rank 1, and many have higher rank values. Now, if we see the results of EMPSO, it has achieved rank 1 in 3 cases of five cases. In the rest of the cases, the EMPSO has achieved rank 2. Such good results of EMPSO indicate that the proposed strategy used to select global best is more effective than the rest of the strategies. The results of GMPSO, FMPSO, and IMPSO show that they are performing nearly equivalent, as each of them has rank 1 in only one case and the rest of the cases have either rank 2, rank 3, rank 4, rank 5, or rank 6. This indicates that the grid-based, fuzzy Pareto dominance, and indicator-based strategies are the second best performing strategy. The results of the variants DMPSO, BMPSO, SMPSO, AMPSO, and WMPSO show that no variant could achieve rank 1 and rank 2, and has rank 3, rank 4, rank 5, or rank 6. This indicates that the rest of the selection strategies, i.e., distance-based selection, b-dominance, ε-dominance, α-dominance, and weighted average rank strategies, have the poor selection approach for the global best.

Table 3 presents the difference [rank] values of each variant computed in terms of hypervolume over all five problem instances. The difference [rank] values of variants show that the EMPSO and GMPSO perform better than the rest of the variants. Among EMPSO and GMPSO, the EMPSO is having better results than the GMPSO. If we see the ranking values of these two variants, the EMPSO has achieved rank 1 in 3 cases of 5 cases, whereas the GMPSO has achieved rank 1 in 2 cases of five cases. Similar to the IGD results, the hypervolume results of GMPSO, FMPSO, and IMPSO show that they are performing nearly equivalent, as each of them has rank 1 in only one case and in the rest of the cases they have either rank 2, rank 3, rank 4, rank 5, or rank 6. The variants DMPSO, BMPSO, SMPSO, AMPSO, and WMPSO show that they could achieve rank 1 and rank 2, and has either rank 3, rank 4, rank 5, rank 6, rank 7, rank 8, or rank 9.

Table 4 presents difference [rank] values of each variant computed in terms of MQ measure. The MQ values of the candidate solution measure the quality of the candidate solution concerning the strength of intra- and inter-package dependencies, whereas IGD and hypervolume measure the candidate solutions concerning the quality of Pareto front. The difference [rank] values presented in Table 4 show that EMPSO and GMPSO have achieved better ranking values than the other variants. The ranking values of EMPSO and GMPSO are rank 1 or rank 2 in most cases. In particular, the ranking values of these two variants are as follows: the EMPSO has achieved rank 1 in 3 cases of 5 cases, whereas the GMPSO has achieved rank 1 in 2 cases of five cases. Similar to the IGD and hypervolume results of FMPSO, IMPSO, and DMPSO, they are performing nearly equivalent. The results of the variants DMPSO, BMPSO, SMPSO, AMPSO, and WMPSO show that they are performing poorer.

The summary of the ranking results of all the variants computed in IGD, hypervolume, and MQ measures is provided in Figure 5. The proposed global best selection approach employed in the EMPSO variant has achieved better ranks (i.e., rank 1) than the other global best selection-based approaches over most problem instances. The grid dominance-based (GMPSO) approach has achieved rank 2 in most of the cases. The rest of the strategies, i.e., fuzzy Pareto dominance (FMPSO), indicator-based ranking (IMPSO), distance-based ranking (DMPSO), b-dominance strategy (BMPSO), ε-dominance strategy (SMPSO), α-dominance strategy (AMPSO), and weighted average ranking strategy (WMPSO), are performing in increasing

Table 1: Java-based software projects.

System	#Classes	Brief description
JUnit (v3.81)	100	A Java-based unit testing framework
DOM (4J v1.5.2)	195	A Java-based library to parse XML documents
JHotDraw (v6.0. b.1)	398	A Java-based graphic framework
JDI-Ford (v5.8)	638	A Java-based open-source software system
Xerces-J (v2.7.0)	991	A Java-based library to parse XML documents

Table 2: Difference [rank] values of each variant in terms of the IGD.

Problems	EMPSO	GMPSO	FMPSO	IMPSO	DMPSO	BMPSO	SMPSO	AMPSO	WMPSO
JUnit	+8[1]	+6[2]	+4[3]	−2[5]	+2[4]	−2[5]	−2[5]	−6[8]	−2[5]
DOM 4J	+7[1]	−6[3]	+4[3]	+7[1]	+0[5]	−3[6]	+2[4]	−3[6]	−8[9]
JHotDraw	+5[2]	+8[1]	+2[4]	+5[2]	+0[5]	−3[6]	−7[8]	−3[6]	−7[8]
JDI-Ford	+8[1]	+3[3]	+6[2]	+2[4]	+1[5]	−7[8]	−3[6]	−3[6]	−7[8]
Xerces-J	+6[2]	+4[3]	+8[1]	+2[4]	−1[5]	−6[7]	−6[7]	−1[5]	−6[7]

Table 3: Difference [rank] values of each variant in terms of the hypervolume.

Problems	EMPSO	GMPSO	FMPSO	IMPSO	DMPSO	BMPSO	SMPSO	AMPSO	WMPSO
JUnit	+6[2]	+4[3]	+2[4]	+8[1]	+0[5]	−4[6]	−4[6]	−8[9]	−4[6]
DOM 4J	+8[1]	+1[2]	+1[2]	+1[2]	−1[5]	−1[5]	−4[8]	−1[5]	−4[8]
JHotDraw	+7[1]	+7[1]	+3[4]	+4[3]	+0[5]	−6[8]	−2[6]	−4[7]	−8[9]
JDI-Ford	+6[2]	+6[2]	+8[1]	+2[4]	+4[3]	−1[5]	−6[8]	−8[9]	−4[7]
Xerces-J	+7[1]	+7[1]	+3[4]	+4[3]	−4[7]	+0[5]	−2[6]	−6[8]	−8[9]

Table 4: Difference [rank] values of each variant in terms of the MQ.

Problems	EMPSO	GMPSO	FMPSO	IMPSO	DMPSO	BMPSO	SMPSO	AMPSO	WMPSO
JUnit	+6[1]	+5[2]	+5[2]	+4[4]	−2[6]	−1[5]	−5[8]	−4[7]	−8[9]
DOM 4J	+4[2]	+8[1]	+1[3]	+1[3]	−2[6]	−2[6]	+0[5]	−6[9]	−4[8]
JHotDraw	+8[1]	+6[2]	+3[3]	+2[4]	+0[5]	−3[7]	−2[6]	−6[8]	−8[9]
JD-Ford	+8[1]	+6[2]	+3[3]	+3[3]	−2[6]	+0[5]	−8[9]	−4[7]	−6[8]
Xerces-J	+5[2]	+8[1]	+2[4]	+5[2]	−2[6]	−5[7]	−5[7]	+0[5]	−8[9]

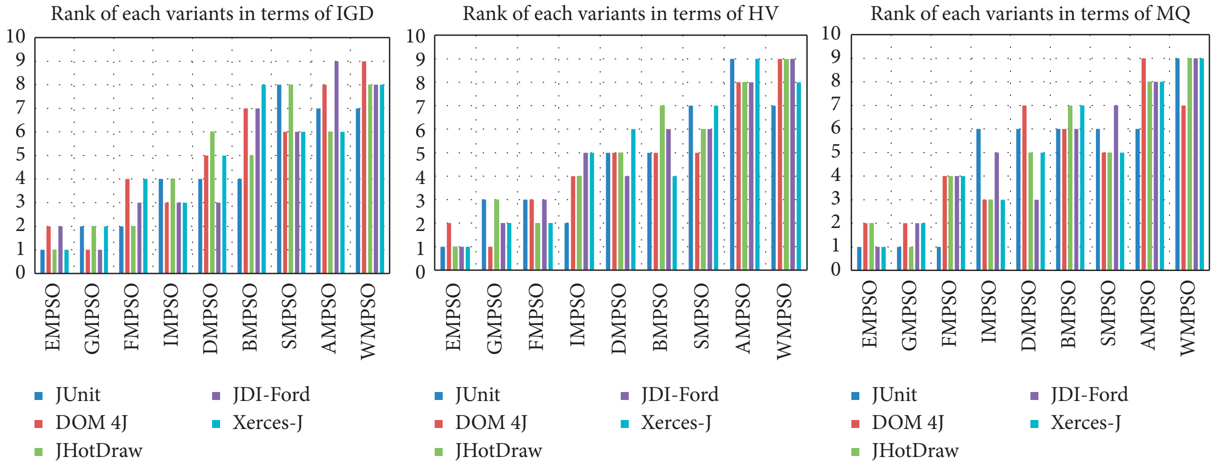


Figure 5: Comparison of variant's rank in terms of IGD, hypervolume, and MQ. (a) Rank of each variant in terms of IGD. (b) Rank of each variant in terms of HV. (c) Rank of each variant in terms of MQ.

order. Overall, the selection strategies that influence the convergence and divergence more appropriately in the optimization algorithm have generated a better Pareto front.

3.1. Discussion. The different global best selection techniques used in the proposed many-objective PSO algorithms had a distinct influence on the generation of Pareto optimal front. The better results obtained through the proposed indicator and entropy-based global best selection technique implied that it has enough capability to lead the optimization process towards a well-distributed approximation of Pareto front. In the proposed global best selection, the indicator-based fitness evaluation helps guide the optimization process towards Pareto front; i.e., it increases the convergence speed, while on the other hand entropy-based fitness evaluation helps maintain the diversity of the approximation set. This encourages the researchers and practitioners to design a global best selection strategy with both properties, i.e., convergence and diversity.

The existing grid dominance-based (GMPSO) approach where both convergence and diversity components are involved also has good performance. The grid dominance-based (GMPSO) approach is the second best performing approach. In contrast, fuzzy Pareto dominance (FMPSO), indicator-based ranking (IMPSO), distance-based ranking (DMPSO), b-dominance strategy (BMPSO), ε-dominance strategy (SMPSO), α-dominance strategy (AMPSO), and weighted average ranking strategy (WMPSO) have shown the poor performance compared with the proposed indicator and entropy-based global best selection. Overall, the comparative results obtained through the proposed global best technique and existing global best techniques demonstrate that the proposed approach can produce the well-distributed approximation of Pareto front compared with the existing global best techniques.

3.2. Threats to Validity. Informally, the validity of a research result is concerned with "How the results might be wrong?" [54]. In our proposed approach, some factors may affect the validity of the achieved results. Wohlin et al. [55] have provided detailed concepts regarding the different types of threats to validity related to software engineering.

3.2.1. Threats to Internal Validity. The internal validity measures the degree of surety to what extent the treatment causes the outcome. This validity ensures that the experiment's outcome is not governed by any other factors except the actual treatment. In our proposed approach, the outcome of the many-objective PSO may be affected by the selection of parameter values and randomness. To mitigate these threats, we tuned the parameters of each proposed variant, ran each variant 31 times, and applied the Mann–Whitney Wilcoxon rank-sum test.

3.2.2. Threats to External Validity. The external validity deals with the generalization of experimental results out of the scope of the current configuration of the study. In this study, we have applied our proposed approach to five object-oriented software systems. Apart from that, our problem instances, i.e., object-oriented software systems, are the different sizes and complexity. To mitigate the external threats to the validity of our proposed approach, we have treated all problem instances as the module dependency graph, which neutralizes the dependency of the proposed approach over the problem instance.

4. Conclusion and Future Work

This study proposed entropy-driven many-objective PSO (named EMPSO) to address the MaOSPR problems. In this approach, a global best selection strategy based on indicator fitness and entropy fitness was designed and incorporated in the framework of many-objective PSO. To justify the potential of the proposed EMPSO, it was tested over the five MaOSPR problems and compared with the eight different variants of many-objective PSO, i.e., GMPSO, FMPSO, IMPSO, DMPSO, BMPSO, SMPSO, AMPSO, and WMPSO. Extensive computational results showed that the EMPSO was the best performing approach, the GMPSO, FMPSO, and IMPSO were the average performer, and the DMPSO, BMPSO, SMPSO, AMPSO, and WMPSO were the poor performer in most of the cases. This indicates that the proposed entropy-based global best selection approach has enough potential to generate better results compared with the existing global best selection strategies. These results also imply that the inclusion of both convergence and diversity properties in the definition of global best selection strategy can generate a good approximation of the Pareto front. In our future work, we will try to extend many-objective PSO to more effective large-scale many-objective PSO that can address more complicated large-scale many-objective software package restructuring problems.

Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding this work.

Acknowledgments

The authors thank the Molde University College-Specialized Univerity in Logistics, Norway, for the support of open-access fund.

References

[1] S. Mancoridis, B. S. Mitchell, C. Rorres, Y. F. Chen, and E. R. Gansner, "Using automatic clustering to produce high-level system organizations of source code," in *Proceedings of the 6th International Workshop on Program Comprehension*, pp. 45–53, IEEE, Ischia, Italy, 26 June 1998.

[2] A. Prajapati and J. K. Chhabra, “MaDHS: many-objective discrete harmony search to improve existing package design,” *Computational Intelligence*, vol. 35, no. 1, pp. 98–123, 2019.

[3] K. Deb and 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*, vol. 18, no. 4, pp. 577–601, 2014.

[4] W. Liu, Z. Wang, X. Liu, N. Zeng, and D. Bell, “A novel particle swarm optimization approach for patient clustering from emergency departments,” *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 632–644, Aug. 2019.

[5] S. Kirkpatrick, C. D. Gellat Jr., and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[6] H. Wang, L. Jiao, and X. Yao, “Two_Arch2: an improved two-archive algorithm for many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 524–541, 2015.

[7] B. Li, J. Li, K. Tang, and X. Yao, “Many-objective evolutionary algorithms: a survey,” *ACM Computing Surveys*, vol. 48, no. 1, p. 35, 2015.

[8] W. Mkaouer, M. Kessentini, A. Shaout et al., “Many-objective software remodularization using NSGA-III,” *ACM Transactions on Software Engineering and Methodology*, vol. 24, no. 3, pp. 1–45, 2015.

[9] A. Prajapati, “Software package restructuring with improved search-based optimization and objective functions,” *Arabian Journal for Science and Engineering*, vol. 46, 2021.

[10] A. Sülflow, N. Drechsler, and R. Drechsler, “Robust multi-objective optimization in high dimensional spaces,” in *Proceedings of the Proc. 4th International conference on EMO*, pp. 715–726, Matsushima, Japan, 5 March 2007.

[11] P. J. Fleming, R. C. Purshouse, and R. J. Lygoe, “Many-objective optimization: an engineering design perspective,” *Evolutionary Multi- Criterion Optimization*, pp. 14–32, Springer, Berlin, 2005.

[12] K. Praditwong, M. Harman, and X. Yao, “Software module clustering as a multi-objective search problem,” *IEEE Transactions on Software Engineering*, vol. 37, no. 2, pp. 264–282, 2011.

[13] K. Narukawa and T. Rodemann, “Examining the performance of evolutionary many-objective optimization algorithms on a real- world application,” in *Proceedings of the Proc. 6th International conference on genetic evolutionary computing*, p. 316, IEEE, Kitakyushu, Japan, 25-28 Aug. 2012.

[14] R. J. Lygoe, M. Cary, and P. J. Fleming, “A real-world application of a many-objective optimisation complexity reduction process,” in *Evolutionary Multi-Criterion Optimization*, pp. 641–655, Springer, Berlin, Heidelberg, Berlin, Germany, 2013.

[15] Q. Zhang and H. Li, “MOEA/D: a multiobjective evolutionary algorithm based on decomposition,” *Proc IEEE Trans Evolut Comput*, vol. 11, no. 6, pp. 712–731, 2007.

[16] Z. Cui, J. Zhang, D. Wu et al., “Hybrid many-objective particle swarm optimization algorithm for green coal production problem,” *Information Sciences*, vol. 518, pp. 256–271, 2020.

[17] D. Sharma, S. Vats, and S. Saurabh, “, Diversity preference-based many-objective particle swarm optimization using reference-lines-based framework,” *Swarm and Evolutionary Computation*, vol. 65, Article ID 100910, 2021.

[18] L. Li, G. Li, and L. Chang, “A many-objective particle swarm optimization with grid dominance ranking and clustering,” *Applied Soft Computing*, vol. 96, Article ID 106661, 2020.

[19] M.-F. Leung, C. A. C. Coello, C.-C. Cheung, S.-C. Ng, and A. K.-F. Lui, “A hybrid leader selection strategy for many-objective particle swarm optimization,” *IEEE Access*, vol. 8, pp. 189527–189545, 2020.

[20] L. Yang, X. Hu, and K. Li, “A vector angles-based many-objective particle swarm optimization algorithm using archive,” *Applied Soft Computing*, vol. 106, Article ID 107299, 2021.

[21] S. Mostaghim and H. Schmeck, “Distance based ranking in many-objective particle swarm optimization,” in *Parallel Problem Solving from Nature – PPSN X. PPSN 2008. Lecture Notes in Computer Science*, G. Rudolph, T. Jansen, N. Beume, S. Lucas, and C. Poloni, Eds., Vol. 5199, Springer, Berlin, Heidelberg, 2008.

[22] B. Wu, W. Hu, Z. He, M. Jiang, and G. G. Yen, “A many-objective particle swarm optimization based on virtual pareto front,” in *Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8, IEEE, Rio de Janeiro, Brazil, 8 July 2018.

[23] A. Prajapati, “Two-archive fuzzy-pareto-dominance swarm optimization for many-objective software architecture reconstruction,” *Arabian Journal for Science and Engineering*, vol. 46, 2021.

[24] E. Zitzler and S. Kunzli, “Indicator-based selection in multiobjective search,” in *Parallel Problem Solving from Nature—PPSN VIII (LNCS 3242)*, X. Yao et al., Ed., pp. 832–842, Springer, Heidelberg, Germany, 2004.

[25] W. Guo, L. Zhu, L. Wang, Q. Wu, and F. Kong, “An entropy-assisted partiscle swarm optimizer for large-scale optimization problem,” *Mathematics*, vol. 7, no. 5, p. 414, 2019.

[26] S. Yang, M. Li, X. Liu, and J. Zheng, “A grid-based evolutionary algorithm for many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 721–736, 2013.

[27] M. Köppen and R. Vicente-Garcia, “A fuzzy scheme for the ranking of multivariate data and its application,” in *Proceedings of the Annual Meeting of the North American Fuzzy Information Processing Society*, pp. 140–145, IEEE, Banff, AB, Canada, 27 June 2004.

[28] A. Prajapati and J. K. Chhabra, “FP-ABC: fuzzy-Pareto dominance driven artificial bee colony algorithm for many-objective software module clustering,” *Computer Languages, Systems and Structures*, vol. 15, pp. 1–21, 2018.

[29] K. Deb, M. Mohan, and S. Mishra, “Evaluating the epsilon-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions,” *Evolutionary Computation*, vol. 13, no. 4, pp. 501–525, 2005.

[30] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, “Combining convergence and diversity in evolutionary multiobjective optimization,” *Evolutionary Computation*, vol. 10, no. 3, pp. 263–282, 2002.

[31] K. Ikeda, H. Kita, and 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, Seoul, South Korea, 27 May 2001.

[32] D. Corne and J. Knowles, “Techniques for highly multi-objective optimisation: some nondominated points are better than others,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, ACM, London England, 7 July 2007.

[33] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of the IEEE International Congress on Neural Networks*, pp. 1942–1948, 1995.

[34] M. R. Bonyadi, “A theoretical guideline for designing an effective adaptive particle swarm,” *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 1, pp. 57–68, 2020.

[35] Y. Lin, Y.-S. Jiang, Y.-J. Gong, Z.-H. Zhan, and J. Zhang, “A discrete multiobjective particle swarm optimizer for automated assembly of parallelcognitive diagnosis tests,” *IEEE Trans. Cybern.*vol. 49, no. 7, pp. 2792–2805, Jun. 2018.

[36] J. Tian, C. Sun, Y. Tan, and J. Zeng, “Granularity-based surrogateassisted particle swarm optimization for high-dimensional expensive optimization,” *Knowledge-Based Systems*, vol. 187, Article ID 104815, 2020.

[37] G. Negi, A. Kumar, S. Pant, and M. Ram, “Optimization of complex system reliability using hybrid grey wolf optimizer,” *Decision Making: Applications in Management and Engineering*, vol. 4, no. 2, pp. 241–256, 2021.

[38] S. Ganguly, “Multi-objective distributed generation penetration planning with load model using particle swarm optimization,” *Decision Making: Applications in Management and Engineering*, vol. 3, no. 1, pp. 30–42, 2020.

[39] A. Stanković, G. Petrović, Ž. Ćojbašić, and D. Marković, “An application of metaheuristic optimization algorithms for solving the flexible job-shop scheduling problem,” *Operational Research in Engineering Sciences: Theory and Applications*, vol. 3, no. 3, pp. 13–28, 2020, https://oresta.rabek.org/index.php/oresta/article/view/57.

[40] N. Zeng, Z. Wang, W. Liu, H. Zhang, K. Hone, and X. Liu, “A dynamic neighborhood-based switching particle swarm optimization algorithm,” in *IEEE Transactions on Cybernetics*IEEE, 2020.

[41] Y. Zhang, X. Liu, F. Bao, C. Jing, C. Zhang, and P. Liu, “Particle swarm optimization with adaptive learning strategy,” *Knowledge-Based Systems*, vol. 196, Article ID 105789, 2020.

[42] W. Hu, G. G. Yen, and G. Luo, “Many-objective particle swarm optimization using two-stage strategy and parallel cell coordinate system,” *IEEE Transactions on Cybernetics*, vol. 47, no. 6, pp. 1446–1459, June 2017.

[43] S. Saeedi, R. Khorsand, S. Ghandi Bidgoli, and M. Ramezanpour, “Improved many-objective particle swarm optimization algorithm for scientific workflow scheduling in cloud computing,” *Computers & Industrial Engineering*, vol. 147, Article ID 106649, 2020.

[44] J. Branke and S. Mostaghim, “About selecting the personal best in multi-objective particle swarm optimization,” in *Parallel Problem Solving from Nature*, pp. 523–532, Springer, Berlin, Heidelberg, 2006.

[45] K. Li, R. Chen, G. Fu, and X. Yao, “Two-archive evolutionary algorithm for constrained multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 303–315, 2019.

[46] A. Prajapati, A. Parashar, and J. K. Chhabra, “Restructuring Object-oriented software systems using various aspects of class information,” *Arabian Journal for Science and Engineering*, vol. 45, pp. 10433–10457, 2020.

[47] K. Praditwong, M. Harman, and X. Yao, “Software module clustering as a multi-objective search problem,” *IEEE Transactions on Software Engineering*, vol. 37, no. 2, pp. 264–282, 2011.

[48] A. Prajapati and J. K. Chhabra, “Improving package structure of object-oriented software using multi-objective optimization and weighted class connections,” *Journal of King Saud University - Computer and Information Sciences*, vol. 29, no. 3, pp. 349–364, 2017.

[49] T. Ting, Y. Shi, S. Cheng, and S. Lee, “Exponential inertia weight for particle swarm optimization,” in *Advances in Swarm Intelligence*, pp. 83–90, Springer, Berlin, Heidelberg, 2012.

[50] A. Ratnaweera, S. Halgamuge, and H. Watson, “Self-organizing hierarchical particle swarm optimizerwith time-varying acceleration coefficients,” *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.

[51] H. Liu, X. W. Zhang, and L. P. Tu, “A modified particle swarm optimization using adaptive strategy,” *Expert Systems with Applications*, vol. 152, Article ID 113353, 2020.

[52] C. K. Goh and K. C. Tan, “Evolving the tradeoffs between pareto-optimality and robustness in multi-objective evolutionary algorithms,” in *Evolutionary Computation in Dynamic and Uncertain Environments. Studies in Computational Intelligence*, S. Yang, YS. Ong, and Y. Jin, Eds., Vol. 51, Springer, Berlin, Heidelberg, 2007.

[53] E. Zitzler and L. Thiele, “Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach,” *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[54] J. Maxwell, *Qualitative Research Design: An Interactive Approach*, Sage Publications Inc., Thousand Oaks, California, 2004.

[55] C. Wohlin, M. H¨ost, P. Runeson, M. Ohlsson, B. Regnell, and A. Wessl´en, *Experimentation in Software Engineering: An Introduction*, Kluwer Academic Publishers, MA London, UK, 2000.

[56] A. Parashar and J. K. Chhabra, “Package-restructuring based on software change history,” *National Academy Science Letters*, vol. 40, pp. 21–27, 2017.

[57] A. Nebro, J. Durillo, J. Garcia-Nieto, C. CoelloCoello, F. Luna, and E. Alba, “SMPSO: a new PSO-based metaheuristic for multi-objective optimization,”vol. 11, pp. 66–73, in *Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Milti-Criteria Decision-Making*, vol. 11, pp. 66–73, IEEE, Nashville, TN, USA, 30 March 2009.