2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Characterization of CEC Single-Objective Optimization Competition Benchmarks and Algorithms

Mustafa Mısır
Division of Natural and Applied Sciences, Duke Kunshan University, Kunshan, China
mustafa.misir@dukekunshan.edu.cn

***Abstract*—The present study provides an analysis on the characteristics of single-objective optimization benchmark problems as well as the algorithms used to solve them. The target optimization domain involves the CEC competitions, each consisting a set of mathematical functions. Concerning the optimization tasks, the idea is to investigate the dis/-similarities between different competition scenarios and individual benchmarks. For the solvers, the goal is to detect the dis/-similarities between the algorithms applied to the CEC benchmarks. Those analysis missions are carried out by using the features directly and automatically extracted from the performance data, the quality of the solutions achieved by each algorithm on the benchmarks. The feature extraction process is realized through Singular Value Decomposition. Following the analysis on the algorithms, the potential of algorithm selection has been evaluated to see the performance improvement without actually developing a new algorithm, against those 20 algorithms.**

***Index Terms*—Single-objective Optimization, Singular Value Decomposition, Algorithm Selection**

I. Introduction

Algorithmic design is one of the major tasks both in research and practice. In combinatorial optimization, there have been immense design efforts, resulting in many new algorithms being introduced at a fast pace, claiming to surpass the existing ones. Alongside that, new optimization problem instances have been brought for experimentally evaluating those algorithms. For the latter aspect, commonly used benchmarks have been built and utilized by the researchers to demonstrate their designs are better than others concerning certain performance criteria. Similarly, various academic competitions, some yearly, have been organized, offering new benchmark sets. Such benchmark sets, especially the hand-made/picked ones, are usually shaped by maintaining diversity referring to the characteristics of the benchmarks. The aim is to assess the distinct capabilities of the algorithms devised for the target problem domains. The diversity in the benchmark sets tend to be determined by considering problem specific traits. For instance, in function optimization, different modalities might be taken into account for maintaining diversity.

This paper aims at examining the existing benchmark sets from a popular, yearly competition series, i.e. the IEEE Congress on Evolutionary Computation (CEC) competitions on single objective optimization of mathematical functions. The complete setting used for evaluation is originated from [1]. To be specific, 9 competition sets, totalling 182 benchmarks are accommodated. 20 algorithms are performed, including 10 Particle Swarm Optimization (PSO) and 10 Differential Evolution (DE) variants. The referenced work [1] already offers a comprehensive performance analysis between those algorithms on the CEC competition benchmarks. Beyond that work, the present study examines both the benchmark sets and the algorithms. In the matter of benchmarks, the benchmark sets are compared against each other to determine how dis/-similar they are. Next, the dis/-similarities among the benchmark problems have been investigated. The same approach is replicated on the algorithms for identifying their dis/-similarity levels. Additionally, Algorithm Selection (AS) [2] is considered to calculate the performance gain that can achieved without devising a new algorithm but simply benefiting those existing 20 algorithms. AS, in this case per-instance AS, essentially refers to automatically choosing an algorithm for solving a particular problem instead. All these procedures are mainly conducted via a set of features extracted from the algorithms' performances on the benchmarks, instead of using some hand-picked features. Following [3], Singular Value Decomposition (SVD) [4] was used to extract a number of latent / hidden features to represent both the algorithms and problem instances in [5]. Concerning the effectiveness of those SVD driven latent features, similarly they have been used to identify substantially smaller representative instance sets compared to a large instance set in [6]. This allowed testing an algorithm on a small benchmark set, portraying the algorithm's performance on the actual, large set without running it.

In the remainder of the paper, Section II details the methodology. Section III provides the corresponding computational analysis and discussion. The paper is summarized while pointing out the follow-up research in Section IV.

II. Method

Singular Value Decomposition (SVD) [4] is applied to the rank performance matrix, $\mathcal{R}$, consisting of the rank of each algorithm on each benchmark. $\mathcal{R}$ is produced from the benchmark-algorithm performance matrix, $\mathcal{P}$, having the average fitness values achieved over a particular number of trials. The best performing algorithm gets the best rank, so the lowest

The research results of this publication are sponsored by the Kunshan Municipal Government research funding.

value, while the worst rank, so then highest value, is assigned to the worst performing algorithm on each benchmark. Thus, when n algorithms are available, the ranks ranging between 1 and n are assigned to the algorithms. When the same performance is delivered by more than one algorithm on a particular benchmark, their ranks are averaged. For instance, if top 3 algorithms have the same performance, instead of putting the rank of 1 for these 3 algorithms, $(1+2+3)/3 = 2$ is assigned to each. The resulting rank matrix, $\mathcal{R}$, is decomposed into 3 matrices as follows:

$$\mathcal{R} = U\Sigma V^T \approx U_r \Sigma V_r^T$$

where U is the matrix representing the benchmarks, V is the matrix representing the algorithms and Σ is a diagonal matrix with singular values denoting the importance of each latent feature, i.e. the matrix columns of U and V. In other words, each row of U involves the features characterizing a benchmark and each row of V involves the features characterizing an algorithm. For eliminating the noise and maintaining the most critical and representative features, only top r features are used. These features are then exploited to

- identify the benchmark instances that are substantially dis/-similar via clustering,
- specify the benchmark sets that are most in/-comparable with clustering and correlation analysis,
- investigate the diversity and fairness in terms of algorithm performance evaluation, of the benchmarks sets,
- determine the algorithms with un/-alike problem solving behaviour through clustering,
- outperform all the available algorithms without designing any new one by the help those same algorithms in an algorithm selection setting.

III. Computational Results

The data and experimental setting is from [1], where 20 algorithms and 182 benchmark problems were accommodated. Table I shows the algorithms used. Referring to [1], the algorithms are essentially belong to two families of Differential Evolution (DE) and Particle Swarm Optimization (PSO). These are well-known meta-heuristic methods, especially in combinatorial optimization. Table II lists the utilized CEC competition benchmarks of single-objective optimization, composed of various mathematical functions. Each algorithm was applied to every benchmark 51 times. The average of the fitness values is recorded as the performance indicator.

Dataset	**# Benchmarks**
CEC 2011	22
CEC 2014_10	30
CEC 2014_50	30
CEC 2017_10	30
CEC 2017_50	30
CEC 2020_5	10
CEC 2020_10	10
CEC 2020_15	10
CEC 2020_20	10

TABLE II: The CEC competition benchmarks (the numbers following an underscore indicates the functions' dimensions)

Figure 1 shows the performance of the employed 20 algorithms across all the CEC benchmarks, in terms of average ranks. As also discussed in [1], the DE variants, in general, outperforms the PSO variants. That being said, it is possible to notify benchmarks where certain PSO algorithms offer better performance than the DE algorithms. This outcome is

Differential Evolution (DE)	**Particle Swarm Optimization (PSO)**
DE [7]	PSO [8]
Self-Adaptive DE (SADE) [9]	Comprehensive Learning PSO (CLPSO) [10]
Adaptive Population Tuning Scheme for DE (APTS-DE) [11]	PSO with Aging Leader and Challengers (ALC-PSO) [12]
DE with an Individual-dependent mechanism (IDE) [13]	Heterogeneous CLPSO (HCLPSO) [14]
Adaptive DE with Multiple sub-populations (MPADE) [15]	PSO with Inter-swarm Interactive Learning (IILPSO) [16]
Hybrid Memetic Composite DE (CoDE) and JADE (HMJCDE) [17]	Genetic Learning PSO (GLPSO) [18]
Ensemble Sinosoidol Parameter Adaptation, Success-History based Adaptive DE (SHADE), with Linear population size reduction (L-SHADE) (L-SHADE-cnEpSin) [19]	Ensemble PSO (EPSO) [20]
Hierarchical Archive based DE (HARD-DE) [21]	Dual-Environmental PSO (DEPSO) [22]
Modified CIPDE with Modified JADE (CIJADE) [23]	Triple Archives PSO (TAPSO) [24]
Neighbourhood based, Success-History based Adaptive DE (SHADE), with Linear population size reduction (N-L-SHADE) [25]	PSO for single-objective numerical optimization (PSO-sono) [26]

TABLE I: The tested 20 algorithms

aligned with the empirical algorithmic studies where a good performing algorithm on particular scenarios are expected to deliver poor performance on some others [27]. Thus, it is possible to benefit from Algorithm Selection (AS) on this particular CEC benchmark setting.

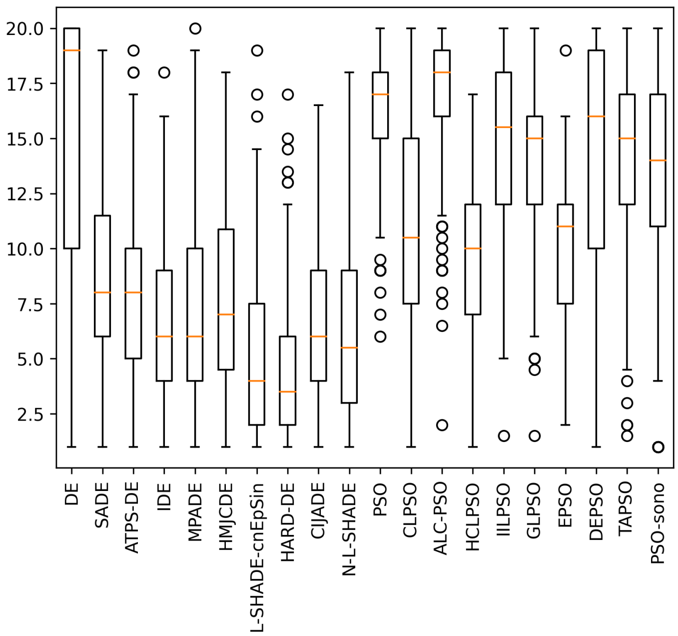

Fig. 1: The ranks of the algorithms across all the CEC competition benchmarks

Figure 2 shows the utilized algorithms in a sorted manner with respect to their average ranks. In addition to those 20 tested algorithms, Oracle, a.k.a. Virtual Best Solver (VBS), reveals the optimal AS. Oracle is essentially the best possible AS performance when the best algorithm, out of the 20 constituent algorithms, is applied for each benchmark. The overall / single-best algorithm, i.e. HARD-DE, delivers the average rank of 5.29 while Oracle comes with the average rank of 2.32. This clear performance difference suggests that utilizing AS instead of developing new algorithms could be more beneficial and effective. Going into the details of the Oracle's behavior, Figure 3 shows the selection frequencies of each algorithm. L-SHADE-cnEpSin has been the most frequently selected one, by 48 times, despite having the average rank of 5.68 which is slightly lower than the overall best algorithm, HARD-DE. On the contrary, the 4 PSO variants[1], i.e. PSO, ALC-PSO, EPSO and TAPSO, are never picked while PSO-sono with a significantly poor performance, resulting in the average rank of 14.73, is utilized 7 times. Following the earlier AS claim, an inferior algorithm like PSO-sono can still be benefited from.

Continuing from the algorithm space, Figure 4 shows the dis/-similarity between the 20 tested algorithms through hierarchical clustering. The clustering is achieved by using the latent

[1]They could have been actually selected though for the benchmarks where all the algorithms deliver the same performance. In those cases, the first occurring, best algorithm is picked.

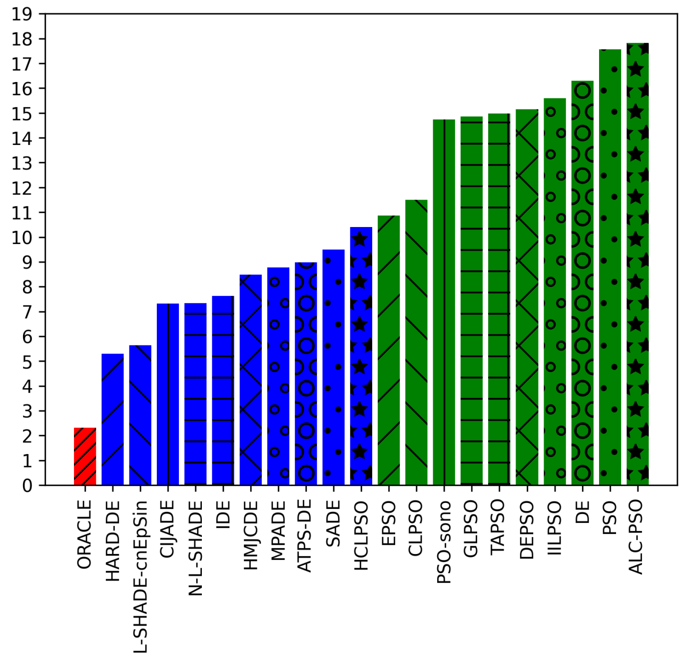

Fig. 2: The average ranks of all the competing algorithms besides Oracle across all the CEC competition benchmarks

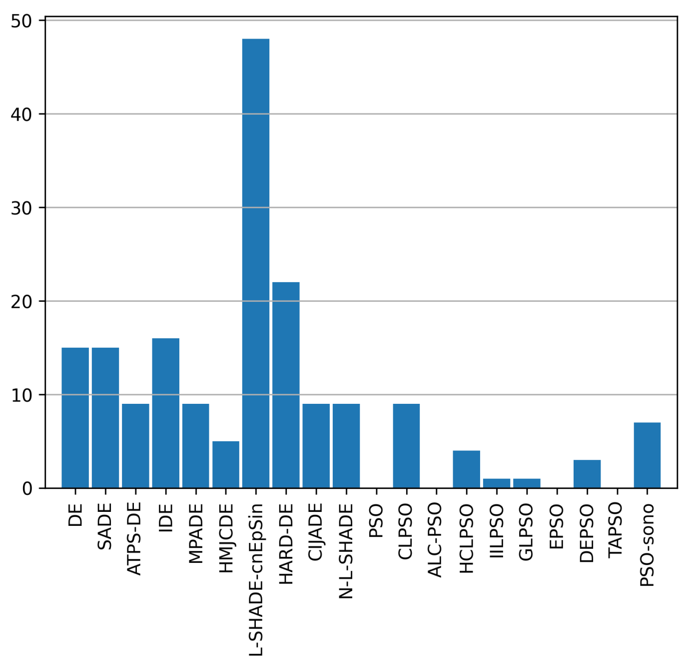

Fig. 3: The frequency of the algorithms being selected under the Oracle algorithm selector

features extracted via SVD ($r = 10$) on the CEC benchmark-algorithm performance data. Thus, each algorithm is represented by automatically specified 10 features. The most similar algorithms are explored as (L-SHADE-cnEpSin, HARD-DE), (PSO, ALC-PSO), (HCLPSO, EPSO) and (GLPSO, TAPSO). Despite the differences between the algorithms and the dates

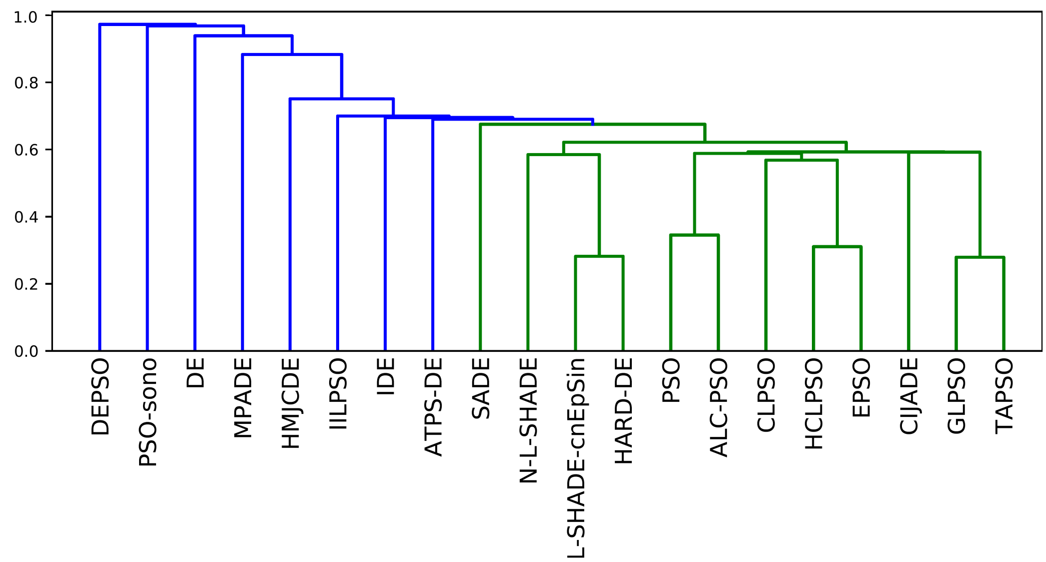

Fig. 4: The hierarchical clustering of the algorithms on the CEC competition benchmarks by the SVD ($r = 10$) latent features

they were introduced, there are no drastic performance and behavioural distinctions. For instance, GLPSO and TAPSO as two PSO algorithms were introduced in 2016 and 2020 respectively with distinct designs. However, they achieved the average ranks of 14.86 and 14.97 and showed similar behavior across the target CEC benchmark problems. As earlier pointed out, these findings raise the question of whether so many new algorithm designs are really needed. Still, it is possible to identify scenarios where those similar algorithms significantly diverge. Beyond that, since all the algorithms come with a wide range of parameters and design choices, it is likely to change the way they work by tweaking them through parameter optimization / tuning [28].

Going to the CEC benchmark space, Figure 6 illustrates the benchmark problems from each dataset, referring to different CEC competitions and their sub-scenarios. The visualization is prepared by using 10 latent features by SVD ($r = 10$) but now for the benchmark problems. All the benchmarks are clustered using k-means with $k = 9$. This k value is chosen to match with each dataset or benchmark type as there are 9 different competition scenarios. Those features are then reduced to 2 by using t-distributed stochastic neighbor embedding (t-SNE) [29] and the discovered clusters are shown. It is possible to see resemblances between different competition scenarios. To have a clearer view, Figure 7 shows the rounded ratio of the benchmarks falling into each cluster and their memberships to the benchmark types. Figure 8 further reports Pearson correlation coefficients based on those ratios from Figure 7. The calculated coefficients help to identify similar competition scenarios, such as CEC 2014_10 and CEC 2017_10. Thus, the rankings of the tested algorithms on those similar scenarios are comparable. From this perspective, it is critical to build dissimilar test sets in competition settings compared to the earlier ones. Additionally, especially from Figure 7, it might be possible to determine the diversity of a particular benchmark set. For example, CEC 2014_50 benchmarks span across all the clusters so likely to be rather diverse while CEC 2020 benchmarks offer comparably less diversity.

Beyond the dis/-similarities between different benchmark types, Figure 5 illustrates dis/-similarities between the individual CEC benchmarks. To exemplify, CEC 2014_50_9 and CEC 2017_50_5 are detected as highly similar benchmarks from the CEC 2014 and CEC 2017 benchmark sets.

IV. Conclusion

This study focuses on the characterization of the CEC single-objective numerical optimization benchmarks from the corresponding CEC competitions. The evaluation targets dis/-similarities between the CEC competition scenarios and the individual benchmark problems. The other focus is on the algorithms' side, concerning a suite of 20 algorithms. Their dis/-similarities are realized while evaluating the potential of Algorithm Selection (AS) for managing those algorithms for solving the CEC benchmarks. The complete evaluation and analysis is performed by using the Singular Value Decomposition (SVD) driven features from the algorithms' performances on those CEC benchmarks.

Following this work, the presented analysis will be expanded by actually performing Algorithm Selection (AS), using the existing AS methods. Aligned with this addition, a feature set representing the single-objective optimization mathematical functions will be determined, referring to the relevant literature. Then, the characterization discussion will be enhanced by those hand-picked features, for a deeper assessment matching the behaviour of the algorithms. Furthermore, as the current algorithm set consists of only Differential Evolution (DE) and Particle Swarm Optimization (PSO) algorithms, the state of the art algorithms, including the ones attended the CEC competitions, will be incorporated. On this aspect, Algorithm Portfolios [30] will be built for having a candidate algorithm

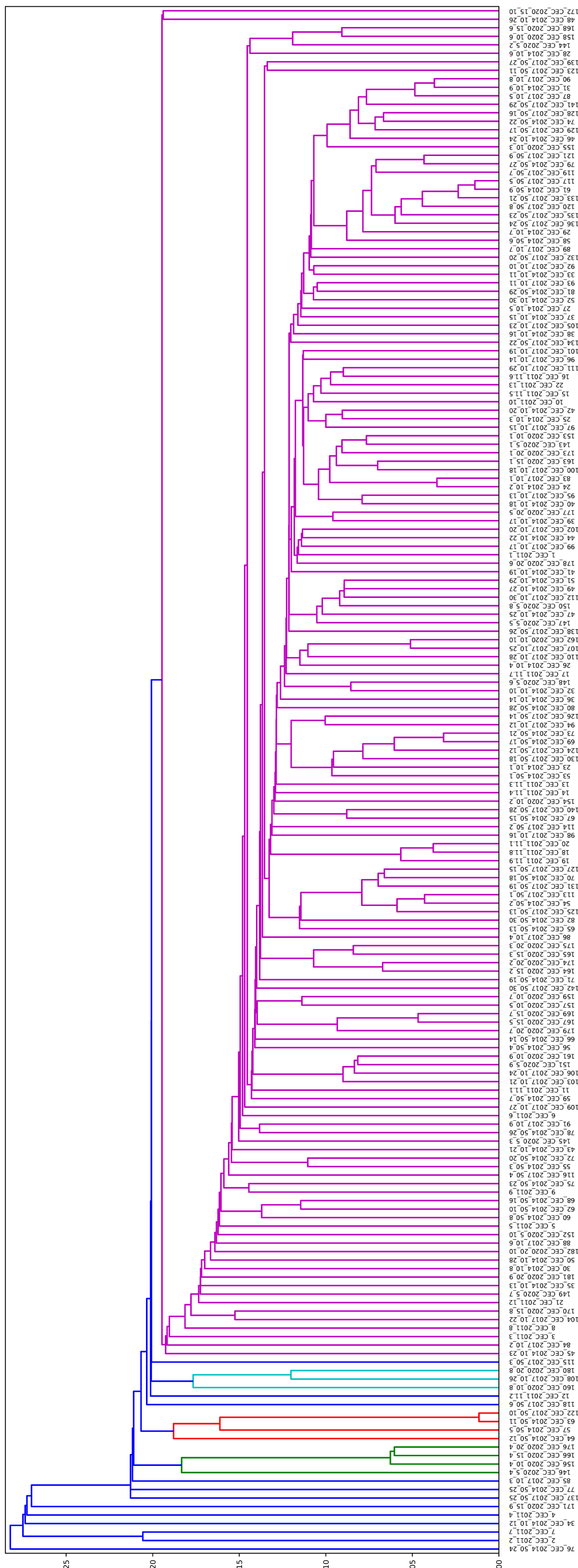

Fig. 5: A hierarchical clustering of the CEC competition benchmarks through the latent features extracted from the performance data, with SVD ($r = 10$)

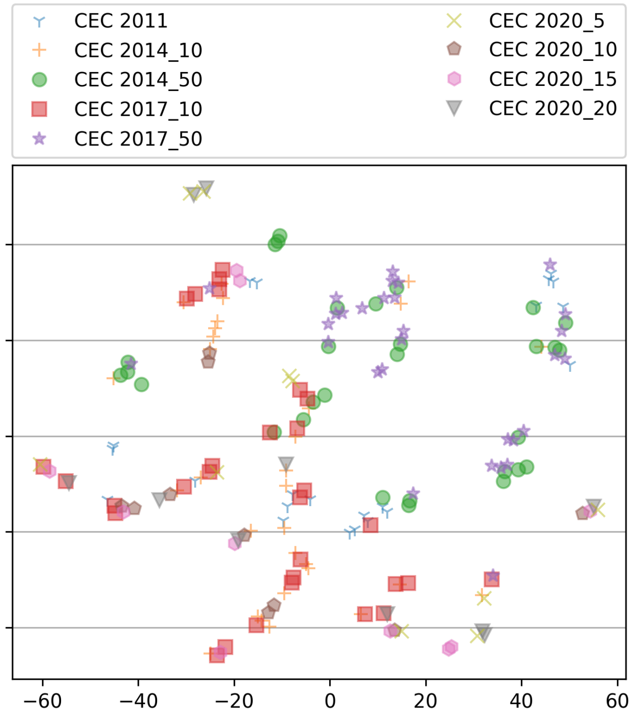

Fig. 6: The t-SNE 2-dimensional visualization of the CEC competition instances based on the latent features derived with SVD ($r = 10$)

set involving diverse algorithms with respect to their problem solving capabilities. This idea will be realized by selection and tuning, considering that all the algorithms come with parameters and design choices.

References

[1] A. P. Piotrowski, J. J. Napiorkowski, and A. E. Piotrowska, "Particle swarm optimization or differential evolution—a comparison," *Engineering Applications of Artificial Intelligence*, vol. 121, p. 106008, 2023.

[2] P. Kerschke, H. H. Hoos, F. Neumann, and H. Trautmann, "Automated algorithm selection: Survey and perspectives," *Evolutionary Computation*, pp. 1–47, 2018.

[3] M. Mısır and M. Sebag, "ALORS: An algorithm recommender system," *Artificial Intelligence*, vol. 244, pp. 291–314, 2017.

[4] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numerische mathematik*, vol. 14, no. 5, pp. 403–420, 1970.

[5] M. Mısır, "Matrix factorization based benchmark set analysis: a case study on HyFlex," in *the 11th International Conference on Simulated Evolution and Learning (SEAL)*, ser. LNCS, vol. 10593. Springer, 2017, pp. 184–195.

[6] M. Mısır, "Benchmark set reduction for cheap empirical algorithmic studies," in *IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2021, pp. 871–877.

[7] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[8] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *IEEE International Conference on Evolutionary Computation (CEC)*. IEEE, 1998, pp. 69–73.

[9] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2008.

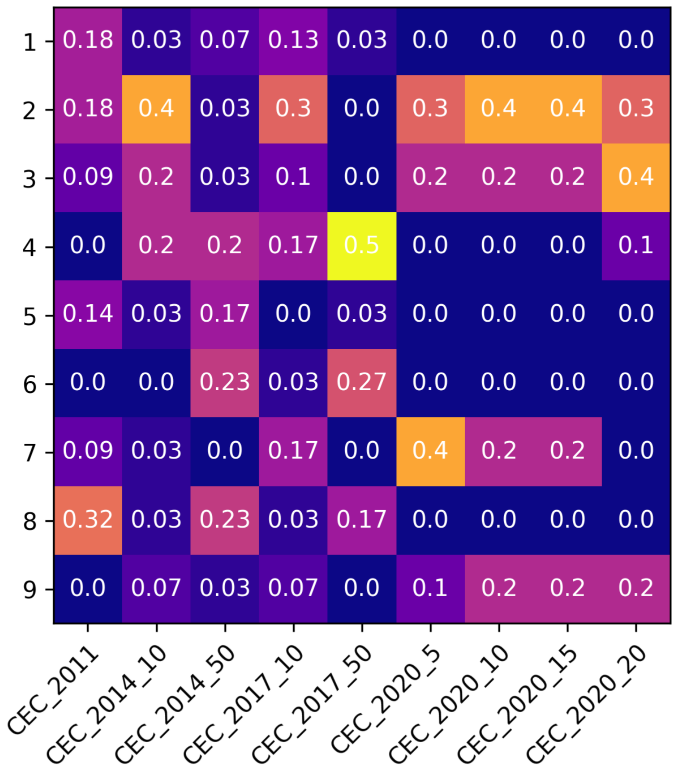

Fig. 7: The percentage of the CEC competition benchmarks from each cluster when k-means is applied with $k = 9$, based on the latent features derived with SVD ($r = 10$) (the vertical axis denotes the clusters)

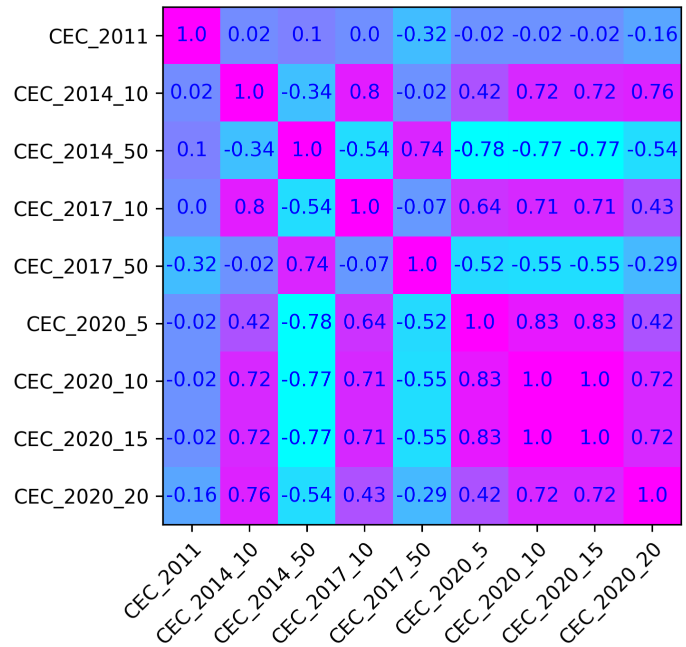

Fig. 8: The Pearson correlation coefficients between each benchmark set pair with respect to their cluster distributions provided in Figure 7, where $+1$ denotes strongly positive correlation, -1 shows the strongly negative correlation, 0 indicates no correlation

[10] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE transactions on evolutionary computation*, vol. 10, no. 3, pp. 281–295, 2006.

[11] W. Zhu, Y. Tang, J.-a. Fang, and W. Zhang, "Adaptive population tuning scheme for differential evolution," *Information Sciences*, vol. 223, pp. 164–191, 2013.

[12] W.-N. Chen, J. Zhang, Y. Lin, N. Chen, Z.-H. Zhan, H. S.-H. Chung, Y. Li, and Y.-H. Shi, "Particle swarm optimization with an aging leader and challengers," *IEEE transactions on evolutionary computation*, vol. 17, no. 2, pp. 241–258, 2012.

[13] L. Tang, Y. Dong, and J. Liu, "Differential evolution with an individual-dependent mechanism," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 560–574, 2014.

[14] N. Lynn and P. N. Suganthan, "Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation," *Swarm and Evolutionary Computation*, vol. 24, pp. 11–24, 2015.

[15] L. Cui, G. Li, Q. Lin, J. Chen, and N. Lu, "Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations," *Computers & Operations Research*, vol. 67, pp. 155–173, 2016.

[16] Q. Qin, S. Cheng, Q. Zhang, L. Li, and Y. Shi, "Particle swarm optimization with interswarm interactive learning strategy," *IEEE transactions on cybernetics*, vol. 46, no. 10, pp. 2238–2251, 2015.

[17] G. Li, Q. Lin, L. Cui, Z. Du, Z. Liang, J. Chen, N. Lu, and Z. Ming, "A novel hybrid differential evolution algorithm with modified code and jade," *Applied Soft Computing*, vol. 47, pp. 577–599, 2016.

[18] Y.-J. Gong, J.-J. Li, Y. Zhou, Y. Li, H. S.-H. Chung, Y.-H. Shi, and J. Zhang, "Genetic learning particle swarm optimization," *IEEE transactions on cybernetics*, vol. 46, no. 10, pp. 2277–2290, 2015.

[19] N. H. Awad, M. Z. Ali, and P. N. Suganthan, "Ensemble sinusoidal differential covariance matrix adaptation with euclidean neighborhood for solving cec2017 benchmark problems," in *IEEE congress on evolutionary computation (CEC)*. IEEE, 2017, pp. 372–379.

[20] N. Lynn and P. N. Suganthan, "Ensemble particle swarm optimizer," *Applied Soft Computing*, vol. 55, pp. 533–548, 2017.

[21] Z. Meng and J.-S. Pan, "Hard-de: Hierarchical archive based mutation strategy with depth information of evolution for the enhancement of differential evolution on numerical optimization," *IEEE Access*, vol. 7, pp. 12 832–12 854, 2019.

[22] J. Zhang, X. Zhu, Y. Wang, and M. Zhou, "Dual-environmental particle swarm optimizer in noisy and noise-free environments," *IEEE transactions on cybernetics*, vol. 49, no. 6, pp. 2011–2021, 2018.

[23] J.-S. Pan, N. Liu, and S.-C. Chu, "A hybrid differential evolution algorithm and its application in unmanned combat aerial vehicle path planning," *IEEE Access*, vol. 8, pp. 17 691–17 712, 2020.

[24] X. Xia, L. Gui, F. Yu, H. Wu, B. Wei, Y.-L. Zhang, and Z.-H. Zhan, "Triple archives particle swarm optimization," *IEEE transactions on cybernetics*, vol. 50, no. 12, pp. 4862–4875, 2019.

[25] A. Ghosh, S. Das, A. K. Das, R. Senkerik, A. Viktorin, I. Zelinka, and A. D. Masegosa, "Using spatial neighborhoods for parameter adaptation: An improved success history based differential evolution," *Swarm and Evolutionary Computation*, vol. 71, p. 101057, 2022.

[26] Z. Meng, Y. Zhong, G. Mao, and Y. Liang, "Pso-sono: A novel pso variant for single-objective numerical optimization," *Information Sciences*, vol. 586, pp. 176–191, 2022.

[27] D. Wolpert and W. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, pp. 67–82, 1997.

[28] M. Feurer and F. Hutter, "Hyperparameter optimization," *Automated Machine Learning: Methods, Systems, Challenges*, pp. 3–33, 2019.

[29] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[30] C. Gomes and B. Selman, "Algorithm portfolio design: Theory vs. practice," in *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence (UAI)*, Providence/Rhode Island, USA, August 1–3 1997, pp. 190–197.