**Personalized Content Generation in A Comparative Study and Optimization of Multi-Objective Algorithms on the DTLZ2 Benchmark Problem**

Arsalan Mehmood - msds21101102@szabist.pk

***Abstract--* Multi-objective optimization problems (MOPs) frequently occur in real-world scenarios requiring simultaneous optimization of conflicting objectives. This study evaluates and optimizes six state-of-the-art multi-objective evolutionary algorithms (MOEAs): Adaptive Variable Grouping with Surrogate-Assisted Evolutionary Algorithm (AVG-SAEA), Besiege and Conquer Algorithm (MOBCA), Multi-Objective Evolutionary Algorithm based on Decomposition with Concurrent Tasking (MOEADCMT), Surrogate-Facilitated Adaptive Differential Evolution (SFADE), Steady-State Non-dominated Sorting Genetic Algorithm II (S-NSGA-II), and Theta-dominance Evolutionary Algorithm (tDEA). Using the DTLZ2 benchmark problem with three objectives, the algorithms were compared on performance metrics including Generational Distance (GD), Inverted Generational Distance (IGD), Hypervolume (HV), Spread (Δ), Spacing (S), and Computational Cost.**

**The optimized S-NSGA-II incorporated vectorized evaluation, parallel computation, adaptive crossover, and simplified non-dominated sorting, achieving a 47.3% reduction in computational time while maintaining superior performance across other metrics, such as a substantial improvement in hypervolume.**

**Keywords: MOO, DTLZ2.**

1. **INTRODUCTION**

Multi-objective optimization addresses problems where trade-offs between conflicting objectives are critical. Evolutionary algorithms (EAs), particularly multi-objective evolutionary algorithms (MOEAs), have proven to be effective in generating diverse sets of Pareto-optimal solutions. However, challenges such as computational inefficiencies limit their practical applications in large-scale and real-time domains.

Multi-objective problems, commonly found in engineering, logistics, and healthcare, often involve conflicting objectives that require careful balance. For instance, optimizing manufacturing processes may require minimizing costs while maximizing efficiency. The scalability and complexity of these problems make them suitable for advanced optimization methods like MOEAs.

This research evaluates six MOEAs on the DTLZ2 problem, a benchmark known for its scalable complexity and non-linear Pareto front. S-NSGA-II, while robust, was found to have computational inefficiencies, prompting targeted optimizations to enhance its efficiency without sacrificing solution quality. The study’s results aim to provide insights into algorithm performance and strategies for practical applications in large-scale optimization scenarios.

1. **PROBLEM STATEMENT**

Multi-objective optimization involves challenges in balancing convergence and diversity in Pareto front solutions, particularly for complex benchmarks like DTLZ2. State-of-the-art MOEAs, such as S-NSGA-II, face significant computational inefficiencies due to operations like non-dominated sorting and crowding distance calculations. Addressing these inefficiencies is crucial for their practical application in high-dimensional and large-scale problems. This study focuses on optimizing S-NSGA-II to reduce computational overhead while maintaining high performance across established metrics.

1. **SOLUTION STATEMENT**

The solution proposed in this research involves optimizing the Steady-State Non-dominated Sorting Genetic Algorithm II (S-NSGA-II). The optimizations include vectorized evaluation, parallel computation, simplified crowding distance calculations, and adaptive genetic operators.

1. **Vectorized Evaluation**: Evaluating the entire population simultaneously to reduce repetitive computations.
2. **Parallel Computation**: Utilizing Python’s multiprocessing capabilities to distribute evaluation tasks across cores.
3. **Simplified Crowding Distance Calculations**: Reducing the complexity of distance-based diversity measures.
4. **Adaptive Genetic Operators**: Dynamically adjusting crossover and mutation probabilities based on population diversity.

These enhancements aim to significantly reduce computational cost while preserving or improving the algorithm’s performance on key metrics such as convergence, diversity, and hypervolume.

1. **BACKGROUND AND SIGNIFICANCE**

Multi-objective optimization is critical in fields like engineering, logistics, and healthcare, where conflicting objectives must be optimized simultaneously. For instance, optimizing healthcare resource allocation might involve maximizing patient outcomes while minimizing costs, requiring solutions that balance trade-offs effectively.

MOEAs, such as NSGA-II and MOEA/D, have emerged as powerful tools for addressing these challenges. NSGA-II’s fast non-dominated sorting and MOEA/D’s decomposition-based approach have set benchmarks in the field. However, computational inefficiencies, particularly in sorting and diversity measures, often limit their scalability.

The DTLZ2 problem, with its non-linear Pareto front and scalable complexity, serves as a benchmark for evaluating MOEAs. Its multi-objective nature challenges algorithms to balance convergence and diversity effectively. Despite advancements in MOEAs, computational inefficiencies remain a barrier, especially in large-scale problems. This study’s optimization of S-NSGA-II addresses these inefficiencies, advancing the practical application of MOEAs in real-world scenarios. Furthermore, the findings contribute to bridging the gap between theoretical advancements and practical implementations in optimization.

1. **METHODOLOGY**

The methodology for this research is structured as follows:

1. **Problem Definition**: The DTLZ2 problem was chosen for its scalable complexity and non-linear Pareto front. It involves three objectives and 12 decision variables. Each objective requires solutions that strike a balance between competing goals, making it an ideal benchmark for algorithm evaluation.
2. **Evaluation Metrics**: Metrics included GD, IGD, HV, Spread (Δ), Spacing (S), and Computational Cost.

* **Generational Distance (GD)**: where is the Euclidean distance between a solution and the true Pareto front. GD measures how close the algorithm’s solutions are to the true Pareto front.
* **Inverted Generational Distance (IGD)**: where is the distance from a true Pareto front solution to the closest generated solution. IGD evaluates the spread of solutions across the Pareto front.
* **Hypervolume (HV)**: where is the reference point, and is the solution value. HV quantifies the dominated space, reflecting both convergence and diversity.
* **Spread (Δ)** and **Spacing (S)**: Measure solution distribution uniformity, assessing how well the solutions are distributed across the front.
* **Computational Cost (CC)**: representing execution time. Lower computational cost indicates better efficiency.

1. **Optimization of S-NSGA-II**:

* **Vectorized Evaluation**: Entire population evaluated simultaneously to streamline the process.
* **Optimized Non-Dominated Sorting**: Implemented faster sorting algorithms and reduced redundancy in dominance checks.
* **Parallel Population Evaluation**: Utilized Python’s multiprocessing to divide and conquer evaluation tasks.
* **Adaptive Operators**: Adjusted crossover and mutation probabilities dynamically based on diversity levels within the population.

1. **Experimental Setup**: Implemented in Python using libraries like NumPy for vectorized operations. Parameters, such as population size (1000) and number of generations (50), were standardized across all algorithms for fair comparison. Each experiment was repeated 10 times to ensure statistical reliability.
2. **MODEL SELECTION**

The model selection process focused on identifying diverse multi-objective evolutionary algorithms (MOEAs) that represent state-of-the-art approaches in solving complex optimization problems. The algorithms selected for this study were:

1. **Adaptive Variable Grouping with Surrogate-Assisted Evolutionary Algorithm (AVG-SAEA)**: AVG-SAEA utilizes surrogate-assisted techniques and adaptive variable grouping to reduce computational complexity. It is particularly suitable for problems with high-dimensional decision spaces.
2. **Besiege and Conquer Algorithm (MOBCA)**: MOBCA incorporates a grid-based reduction strategy and bi-cultural adaptation to balance convergence and diversity. This algorithm is computationally intensive due to its iterative adaptation phases but excels in exploring diverse Pareto fronts.
3. **Multi-Objective Evolutionary Algorithm based on Decomposition with Concurrent Tasking (MOEADCMT)**: MOEADCMT employs a decomposition-based approach to divide the problem into smaller sub-problems, each optimized simultaneously. This concurrent tasking mechanism enhances scalability and performance.
4. **Surrogate-Facilitated Adaptive Differential Evolution (SFADE)**: SFADE integrates surrogate models to approximate objective functions, thereby reducing the number of expensive function evaluations. It adapts mutation and crossover strategies dynamically based on population diversity.
5. **Steady-State Non-dominated Sorting Genetic Algorithm II (S-NSGA-II)**: A steady-state variant of the widely used NSGA-II, this algorithm emphasizes maintaining diversity through crowding distance calculations and sorting. While effective, its computational costs can be high, making it a prime candidate for optimization in this study.
6. **Theta-dominance Evolutionary Algorithm (tDEA)**: tDEA prioritizes solutions based on dominance angles, ensuring a well-distributed Pareto front. This method is particularly advantageous for problems requiring uniform diversity across objectives.

The diversity of these algorithms in terms of mechanisms and applications provided a robust foundation for evaluating performance on the DTLZ2 problem. The selection ensured coverage of both surrogate-assisted and decomposition-based approaches alongside traditional evolutionary strategies.

1. **DISCUSSION**

The results demonstrate that the optimized S-NSGA-II provides a significant improvement over its original implementation and other evaluated MOEAs. The reduction in computational cost, achieved through vectorized evaluation, parallelization, and simplified sorting mechanisms, underscores the effectiveness of these optimizations. Additionally, the improved hypervolume metric indicates that the optimized S-NSGA-II achieves superior convergence and diversity.

Practical implications include its potential application in fields requiring real-time or large-scale optimization, such as logistics, healthcare, and engineering design. For instance, in a logistics scenario, the algorithm could optimize delivery routes to minimize cost while ensuring timely delivery, balancing the trade-offs effectively.

However, the research also highlights limitations, such as the need for additional experimentation on higher-dimensional problems or real-world case studies. Future research could explore hybrid models that integrate surrogate-assisted approaches for improved scalability.

1. **RESULTS**

The performance of the six MOEAs and the optimized S-NSGA-II was evaluated on the DTLZ2 problem using six key metrics: Generational Distance (GD), Inverted Generational Distance (IGD), Hypervolume (HV), Spread (Δ), Spacing (S), and Computational Cost (CC). The results are summarized below:

1. **Generational Distance (GD)**:

* Optimized S-NSGA-II achieved the lowest GD (0.001041), indicating superior convergence to the true Pareto front compared to other algorithms.
* MOEADCMT and MOBCA also performed well in this metric, with values of 0.0094 and 0.013636, respectively.

1. **Inverted Generational Distance (IGD)**:

* The optimized S-NSGA-II achieved an IGD of 0.006632, showcasing its ability to maintain a well-distributed Pareto front.
* Other algorithms, such as MOBCA (0.009456) and MOEADCMT (0.025506), demonstrated moderate performance.

1. **Hypervolume (HV)**:

* A hypervolume of 763.2115 was achieved by the optimized S-NSGA-II, significantly outperforming the original S-NSGA-II (120.2177).
* This improvement highlights the enhanced convergence and diversity achieved through optimization.

1. **Spread (Δ) and Spacing (S):**

* Spread (Δ) was highest for the optimized S-NSGA-II (1.2848), indicating a uniformly distributed Pareto front.
* Spacing (S) was also improved to 0.0651, reflecting consistent distances between neighboring solutions.

1. **Computational Cost (CC):**

* The optimized S-NSGA-II reduced computational cost by 47.3%, with execution time dropping from 296.57 seconds to 156.28 seconds.
* Surrogate-assisted algorithms, such as SFADE and AVG-SAEA, had lower computational costs but at the expense of convergence quality.

1. **CONCLUSION**

This study conducted a comprehensive evaluation and optimization of six state-of-the-art MOEAs, focusing on the DTLZ2 benchmark problem. The optimized S-NSGA-II demonstrated significant improvements across all key metrics, including a 47.3% reduction in computational cost, superior convergence (GD: 0.001041), and enhanced diversity (HV: 763.2115).

The findings underscore the potential of targeted optimizations, such as vectorized evaluation, parallel computation, and adaptive genetic operators, in addressing computational inefficiencies without compromising solution quality. These improvements make the optimized S-NSGA-II a robust and scalable choice for solving complex multi-objective optimization problems in real-world applications.

Future work could explore the integration of surrogate-assisted models and hybrid strategies to further enhance algorithm performance, particularly for high-dimensional or dynamic optimization problems. Additionally, testing on real-world case studies could validate the practical applicability of these findings across various domains, including logistics, healthcare, and engineering design.

1. **REFERENCES**

1. Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2), 182-197. https://doi.org/10.1109/4235.996017

2. Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation, 11(6), 712-731. https://doi.org/10.1109/TEVC.2007.892759

3. Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation, 8(2), 173-195. https://doi.org/10.1162/106365600568202

4. Coello Coello, C. A., Van Veldhuizen, D. A., & Lamont, G. B. (2007). Evolutionary Algorithms for Solving Multi-Objective Problems. Springer Science & Business Media. https://doi.org/10.1007/978-0-387-36797-2

5. Knowles, J., & Corne, D. (1999). The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimization. Proceedings of the 1999 Congress on Evolutionary Computation (CEC), 98-105. https://doi.org/10.1109/CEC.1999.781913

6. Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2002). Scalable test problems for evolutionary multiobjective optimization. Evolutionary Multiobjective Optimization: Theoretical Advances and Applications, 105-145. https://doi.org/10.1007/978-1-4471-3319-7\_6

7. Yang, S., & Li, J. (2008). An adaptive multiobjective optimization approach based on decomposition. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 38(4), 647-657. https://doi.org/10.1109/TSMCC.2008.923845

8. Tanabe, R., & Ishibuchi, H. (2020). An easy-to-use real-world multi-objective optimization problem suite. Applied Soft Computing, 89, 106078. https://doi.org/10.1016/j.asoc.2020.106078

9. Wang, H., Zhang, X., & Zhang, J. (2016). Surrogate-assisted evolutionary algorithms for high-dimensional multi-objective optimization. IEEE Transactions on Evolutionary Computation, 21(3), 435-449. https://doi.org/10.1109/TEVC.2016.2608722

10. Chugh, T., Sindhya, K., Hakanen, J., & Miettinen, K. (2019). A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. Soft Computing, 23, 3137-3166. https://doi.org/10.1007/s00500-017-2965-0

11. Jiang, S., Yang, S., & Wang, Y. (2022). A grid-based evolutionary algorithm for solving large-scale multiobjective optimization problems. IEEE Transactions on Cybernetics, 52(5), 3173-3185. https://doi.org/10.1109/TCYB.2021.3053389

12. Ishibuchi, H., Masuda, H., Tanabe, R., Nojima, Y., & Narukawa, K. (2018). Adaptive scalarizing functions in MOEA/D: An adaptive mechanism for scalarizing function-based multiobjective evolutionary algorithms. IEEE Transactions on Cybernetics, 48(7), 1890-1902. https://doi.org/10.1109/TCYB.2017.2693154