Autoconfiguration framework irace for the SOMA algorithm

Bc. Lukáš Novák

Master's thesis
2024

Tomas Bata University in Zlín
Faculty of Applied Informatics

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2023/2024

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Bc. Lukáš Novák**
Osobní číslo:	**A20604**
Studijní program:	**N0613A140022 Informační technologie**
Specializace:	**Softwarové inženýrství**
Forma studia:	**Kombinovaná**
Téma práce:	**Autokonfigurační framework iRACE pro algoritmus SOMA**
Téma práce anglicky:	**Autoconfiguration Framework iRACE for the SOMA Algorithm**

Zásady pro vypracování

1. Seznamte se s moderními variantami algoritmu SOMA a vytvořte jejich přehled.
2. Dekomponujte vybrané varianty algoritmu SOMA.
3. Definujte sadu testovacích funkcí s ohledem na jejich charakteristiky.
4. Pomocí frameworku iRACE optimalizujte nastavení kompozitní varianty algoritmu SOMA.
5. Analyzujte a zhodnoťte výsledky.

Forma zpracování diplomové práce: **tištěná/elektronická**
Jazyk zpracování: **Angličtina**

Seznam doporučené literatury:

1. LÓPEZ-IBÁÑEZ, Manuel; DUBOIS-LACOSTE, Jérémie; PÉREZ CÁCERES, Leslie; BIRATTARI, Mauro a STÜTZLE, Thomas. The irace package: Iterated racing for automatic algorithm configuration. Online. Operations Research Perspectives. 2016, roč. 2016, č. 3, s. 43-58. ISSN 2214-7160. Dostupné z: https://doi.org/10.1016/j.orp.2016.09.002.
2. ZELINKA, Ivan, 2009. Evoluční výpočetní techniky: principy a aplikace. Praha: BEN – technická literatura. ISBN 978-80-7300-218-3.
3. DAVENDRA, Donald a ZELINKA, Ivan (ed.), 2016. Self-Organizing Migrating Algorithm. Online. Studies in Computational Intelligence. Cham: Springer International Publishing. ISBN 978-3-319-28159-9. Dostupné z: https://doi.org/10.1007/978-3-319-28161-2.
4. DIEP, Quoc Bao; ZELINKA, Ivan a DAS, Swagatam, 2019. Self-Organizing Migrating Algorithm Pareto. MENDEL. 2019-06-24, roč. 25, č. 1, s. 111-120. ISSN 2571-3701. Dostupné z: https://doi.org/10.13164/mendel.2019.1.111.
5. KADAVY, Tomas; PLUHACEK, Michal; VIKTORIN, Adam a SENKERIK, Roman, 2020. Self-organizing migrating algorithm with clustering-aided migration. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion. New York, NY, USA: ACM, 2020-07-08, s. 1441-1447. ISBN 9781450371278. Dostupné z: https://doi.org/10.1145/3377929.3398129.
6. DIEP, Quoc Bao, 2019. Self-Organizing Migrating Algorithm Team To Team Adaptive – SOMA T3A. In: 2019 IEEE Congress on Evolutionary Computation (CEC). IEEE, s. 1182-1187. ISBN 978-1-7281-2153-6. Dostupné z: https://doi.org/10.1109/CEC.2019.8790202.
7. JANÁČEK, Julius. Statistika jednoduše: Průvodce světem statistiky. Grada, 2022. ISBN 978-80-271-1738-3.

Vedoucí diplomové práce: **doc. Ing. Michal Pluháček, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání diplomové práce: **5. listopadu 2023**
Termín odevzdání diplomové práce: **13. května 2024**

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky
DĚKAN
L.S.

doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan

prof. Mgr. Roman Jašek, Ph.D., DBA v.r.
ředitel ústavu

Ve Zlíně dne 5. ledna 2024

I hereby declare that:

- I understand that by submitting my Master's Thesis, I agree to the publication of my work according to Law No. 111/1998, Coll., On Universities and on changes and amendments to other acts (e.g. the Universities Act), as amended by subsequent legislation, without regard to the results of the defence of the thesis.
- I understand that my Master's Thesis will be stored electronically in the university information system and be made available for on-site inspection, and that a copy of the Master's Thesis will be stored in the Reference Library of the Faculty of Applied Informatics, Tomas Bata University in Zlín, and that a copy shall be deposited with my Supervisor.
- I am aware of the fact that my Master's Thesis is fully covered by Act No. 121/2000 Coll. On Copyright, and Rights Related to Copyright, as amended by some other laws (e.g. the Copyright Act), as amended by subsequent legislation; and especially, by §35, Para. 3.
- I understand that, according to §60, Para. 1 of the Copyright Act, TBU in Zlín has the right to conclude licensing agreements relating to the use of scholastic work within the full extent of §12, Para. 4, of the Copyright Act.
- I understand that, according to §60, Para. 2, and Para. 3, of the Copyright Act, I may use my work - Master's Thesis, or grant a license for its use, only if permitted by the licensing agreement concluded between myself and Tomas Bata University in Zlín with a view to the fact that Tomas Bata University in Zlín must be compensated for any reasonable contribution to covering such expenses/costs as invested by them in the creation of the thesis (up until the full actual amount) shall also be a subject of this licensing agreement.
- I understand that, should the elaboration of the Master's Thesis include the use of software provided by Tomas Bata University in Zlín or other such entities strictly for study and research purposes (i.e. only for non-commercial use), the results of my Master's Thesis cannot be used for commercial purposes.
- I understand that, if the output of my Master's Thesis is any software product(s), this/these shall equally be considered as part of the thesis, as well as any source codes, or files from which the project is composed. Not submitting any part of this/these component(s) may be a reason for the non-defence of my thesis.

I herewith declare that:

- I have worked on my thesis alone and duly cited any literature I have used. In the case of the publication of the results of my thesis, I shall be listed as co-author.
- That the submitted version of the thesis and its electronic version uploaded to IS/STAG are both identical.

In Zlín; dated: 13. 5. 2024

Lukáš Novák, v.r.
....
Student's Signature

ABSTRAKT

Tato diplomová práce se zaměřuje na komplexní analýzu a optimalizaci algoritmu SOMA. Výsledky této práce umožní hlubší pochopení funkčnosti algoritmu a jeho variant a povedou k objevení optimálního nastavení pro různé typy optimalizačních problémů.

Klíčová slova: algoritmus SOMA, optimalizace, varianty algoritmu, dekompozice, testovací funkce, irace, kompozitní varianta, ladění parametrů

ABSTRACT

This master thesis will focus on a comprehensive analysis and optimization of the SOMA algorithm. The findings of this work will provide a deeper understanding of the algorithm's functionality and its variants, and lead to the discovery of optimal settings for various optimization problem types.

Keywords: SOMA algorithm, optimization, algorithm variants, decomposition, test functions, irace, composite variant, parameter tuning, performance analysis

ACKNOWLEDGEMENTS

Acknowledgements, motto and a declaration of honour saying that the print version of the Master's thesis and the electronic version of the thesis deposited in the IS/STAG system are identical, worded as follows:

I hereby declare that the print version of my Master's thesis and the electronic version of my thesis deposited in the IS/STAG system are identical.

CONTENTS

INTRODUCTION

This master thesis aims to explore existing SOMA algorithm variants, including their working principles, strengths, and weaknesses. A comprehensive overview of these variants will be created to serve as a foundation for further analysis and comparison.

A detailed analysis of selected SOMA algorithm variants will be conducted to break them down into their constituent components. This decomposition will enable a deeper understanding of the algorithm's functionality and identify key factors influencing its performance.

A collection of test functions with various characteristics will be created to evaluate the SOMA algorithm.The test functions will encompass different dimensionalities, optimization problem types, and difficulty levels.

A composite SOMA variant, combining chosen features from various variants, will be implemented. The irace framework will be employed to automatically tune the parameters of the composite variant and find the optimal configuration for each type of test function.

A thorough analysis of the results obtained from optimizing the composite SOMA variant will be performed. The influence of the test functions and algorithm parameters on its performance will be assessed. The composite variant will be compared to existing SOMA variants and other optimization algorithms.

I. THEORY

1 MODERN SELF-ORGANIZING MIGRATING ALGORITHM VARIANTS OVERVIEW

1.1 Original SOMA

The Self-Organizing Migrating Algorithm (SOMA) is a population-based optimization technique that was introduced by Ivan Zelinka in 1999. SOMA is designed to mimic the social behavior of intelligent creatures, particularly their strategies for searching and exploiting food sources, which are analogous to finding optimal solutions in a problem space. The algorithm employs a migration mechanism where individual solutions in the population, termed "migrants," move towards the best current solution, known as the "leader." This movement is controlled by several parameters, including the migration step size and perturbation vector, which dictate the granularity and variability of the search respectively.

SOMA operates through a series of migration loops. In each loop, every migrant adjusts its position in the direction of the leader by a certain step, modulated by a randomly generated perturbation vector that adds an element of randomness to the search direction. This process helps to explore the search space thoroughly, preventing premature convergence on local optima and enhancing the global search capability. The perturbation vector, typically binary, determines which dimensions of the problem space are adjusted during a move, allowing for a flexible exploration of multi-dimensional landscapes.

The effectiveness of SOMA in finding optimal solutions across various complex and multimodal problem spaces has been validated through numerous studies, making it a robust choice for tackling a wide range of optimization problems. The algorithm's simplicity, combined with its powerful search capabilities, renders it an attractive option within the field of evolutionary computation. [1]

1.2 Overview of Modern SOMA Variants

The Self-Organizing Migrating Algorithm (SOMA) has evolved significantly since its inception, leading to the development of various sophisticated variants designed to address specific optimization challenges. Here, we look into the mechanisms and applications of some of the most notable modern variants of SOMA.

1.2.1 SOMA T3A (Team-to-Team Adaptive SOMA)

SOMA Team-to-Team Adaptive (T3A) introduces a structured, team-based approach to migration. In this variant, the population is divided into teams, and each team migrates towards a designated team leader. This hierarchy not only streamlines the search process but also allows for a finer granularity in the control of the migration parameters. The T3A variant dynamically adapts its migration steps and perturbation vectors in response to ongoing performance feedback, effectively balancing exploration and exploitation as required by the evolving state of the search. SOMA T3A is especially useful in dynamic optimization problems where the search landscape changes over time, as the adaptive mechanism allows the algorithm to remain flexible and responsive to new conditions[6][10].

1.2.2 SOMA Pareto

SOMA Pareto integrates the concepts of Pareto efficiency into the SOMA framework to cater to multi-objective optimization problems. By focusing on achieving a set of solutions that are non-dominated with respect to multiple objectives, SOMA Pareto efficiently navigates the trade-offs inherent in multi-objective environments. This variant adjusts its parameters adaptively, ensuring that the exploration of the search space is balanced effectively with the exploitation of promising areas. The dynamic parameter adaptation in SOMA Pareto helps maintain a diverse set of Pareto-optimal solutions, making it ideal for complex scenarios where multiple conflicting objectives must be reconciled[6][21].

1.2.3 SOMA-CLP (Clustering and Population Learning)

SOMA-CLP, or Clustering and Population Learning, employs advanced clustering techniques to segment the population based on solution similarity, and then applies targeted migration strategies within these clusters. This focused approach allows SOMA-CLP to concentrate its computational efforts on the most promising regions of the search space, identified through the clustering process. By learning from the population's structure and dynamics, SOMA-CLP adapts its search strategy to effectively tackle high-dimensional optimization problems and clustered landscapes. This variant is particularly effective in applications involving large datasets or where the underlying problem structure can be leveraged to enhance search efficiency[6][9].

1.2.4 ESP-SOMA (Ensemble of Strategies and Parameters)

ESP-SOMA represents a flexible ensemble approach that utilizes multiple migration strategies and parameter settings within a single algorithmic framework. By assessing the real-time performance of different strategies, ESP-SOMA adapts its approach to utilize the most effective strategies at various stages of the optimization process. This adaptability makes ESP-SOMA highly robust across a wide range of problems, as it can tailor its behavior to meet specific challenges presented by the optimization landscape. The ensemble approach ensures that ESP-SOMA can leverage the strengths of different strategies to achieve overall performance [13].

1.3 Summary

Each of these modern SOMA variants brings unique capabilities and enhancements to the foundational principles of the original SOMA algorithm. By incorporating adaptive mechanisms, structured team dynamics, multi-objective optimization capabilities, clustering techniques, and ensemble strategies, these variants significantly extend the applicability and effectiveness of SOMA across a broad spectrum of complex optimization problems. As challenges in computational optimization continue to evolve, the ongoing development and refinement of SOMA variants will play a crucial role in enabling efficient and effective solutions.

2 DETAILED DESCRIPTION OF SELECTED VARIANTS

2.1 SOMA T3A

The Self-Organizing Migrating Algorithm Team To Team Adaptive (SOMA T3A) represents a significant evolution in the field of swarm intelligence. This advanced version of the traditional Self-Organizing Migrating Algorithm (SOMA) introduces adaptive mechanisms that dynamically adjust operational parameters in response to real-time feedback from the optimization process. This approach enhances the algorithm's ability to efficiently navigate complex optimization landscapes, optimizing the balance between exploration of new areas and exploitation of promising solutions.

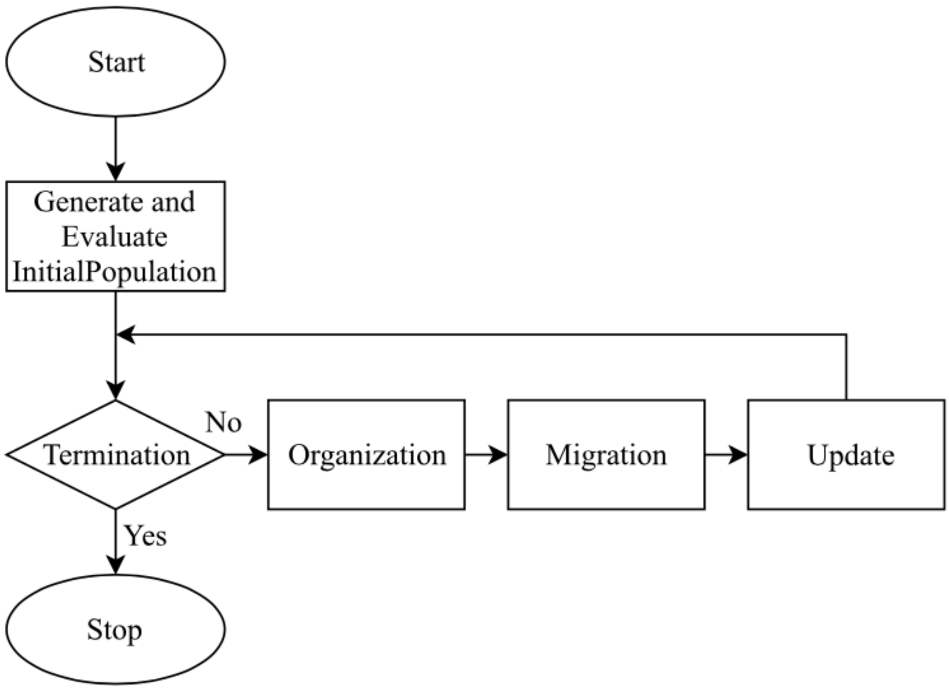


Figure 1 SOMA T3A flowchart

2.1.1 Dynamic Parameter Adjustment in SOMA T3A

SOMA T3A stands out due to its innovative approach to parameter modulation, specifically concerning the Step size and the Perturbation Vector (PRT). These parameters are crucial as they significantly influence the search dynamics of the algorithm:

2.1.1.1 Step Size:

This parameter dictates the magnitude of moves individuals can make towards the leader during each migration loop. In SOMA T3A, the step size is large at the outset to encourage a broad exploration of the search space, facilitating the identification of diverse potential solutions. As the algorithm progresses and zones in on areas with higher potential, the step size is gradually reduced. This reduction enhances the precision of the search around these promising areas, allowing for meticulous exploitation of these zones[10].

2.1.1.2 Perturbation Vector (PRT):

Serving as a directional guide for the individuals' movements, the perturbation vector in SOMA T3A is crucial for determining the search directions in the multidimensional space. Initially set to encourage movements in multiple directions, the PRT allows for an extensive exploration phase. As the search narrows down, the vector values are increased systematically to focus the migration paths directly towards the best solutions discovered, thus intensifying the exploitation efforts[10].

These adaptive strategies ensure that SOMA T3A maintains a dynamic equilibrium between the explorative and exploitative phases, essential for avoiding premature convergence on local optima and for improving the overall efficacy of the search process.

2.1.2 Enhanced Migration Processes in SOMA T3A

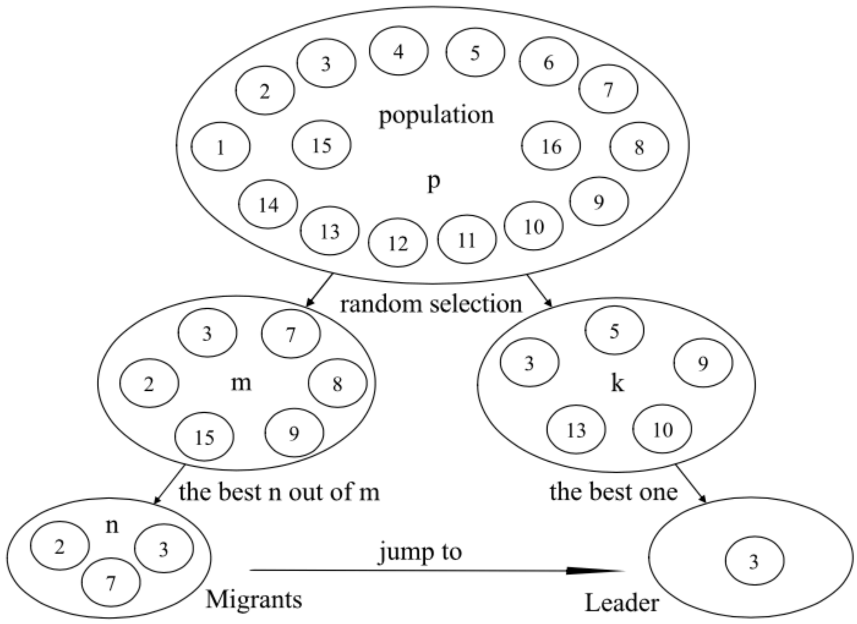


Figure 2 SOMA T3A organization process taken from [10]

2.1.2.1 Organization Process:

The initial stage of each migration loop in SOMA T3A involves a strategic selection of migrants and a leader from the population. This selection is based on fitness, with the highest-performing individuals chosen to lead the group. Such a strategy ensures that the guidance for the group's movements is based on the most promising existing solutions, thus streamlining the search process towards optimal outcomes[10].

2.1.2.2 Migration Process:

During migration, the chosen individuals adjust their positions relative to the leader, guided by the dynamically adjusted Step size and perturbation vector. This phase is the core of the SOMA T3A's search mechanism, where the actual path of each individual in the search space is determined by these parameters. The adaptability of Step and PRT ensures that each migration is optimally aligned with the current state of the search, enabling efficient transitions between exploration and exploitation[10].

2.1.2.3 Update Process:

Post-migration evaluations are critical as they determine the effectiveness of the new positions discovered during the migration. If an individual's new position offers an improvement, it replaces the old position in the population's record. This continuous updating process ensures that each step in the migration contributes to incremental enhancements in the solution quality, progressively leading the population towards the optimal solution[10].

2.1.3 Performance Evaluation

The performance of SOMA T3A has been rigorously evaluated against a range of benchmark problems from the CEC2013 and CEC2017 suites. In these evaluations, SOMA T3A has consistently outperformed previous versions of SOMA as well as several contemporary optimization algorithms. This performance is largely attributed to its adaptive mechanisms, particularly the innovative use of the dynamic perturbation vector, which allows the algorithm to effectively adapt to complex and variable problem landscapes[10].

SOMA T3A's ability to dynamically adjust its search strategies based on real-time feedback significantly enhances its problem-solving capabilities, making it a potent tool for a broad spectrum of optimization challenges. The algorithm's strategic manipulation of dynamic parameters not only improves solution accuracy but also enhances computational efficiency, leading to a more effective and expeditious search process.

Pseudocode 1 SOMA T3A

```
Algorithm  SOMA T3A
Generate and evaluate the initial population
while stopping condition not reached do
    Update PRT and Step
    Choose the Migrants
    for i=1 to the number of Migrants do
        choose the Leader
        if Migrants is not Leader then
            Migrants move to the Leader
            Checking boundary
            Re-evaluate fitness function
            Updated the better position of the Migrant
        end if
    end for
end while
return
```

2.1.4 The Advantages and Applications of SOMA T3A

Compared to the original SOMA and other variants, SOMA T3A has demonstrated performance in finding optimal solutions for various benchmark problems. This significant improvement stems from the combined effects of the team-based selection strategy and adaptive migration parameters. By dividing the population into specific roles and dynamically adjusting the migration process, SOMA T3A efficiently navigates the search space, leading to faster convergence on the optimal solution.

The dynamic nature of the parameters in SOMA T3A equips the algorithm with greater robustness against getting trapped in local optima. Local optima are suboptimal solutions that can mislead traditional optimization algorithms. SOMA T3A's adaptive nature allows it to overcome these challenges and navigate complex search landscapes with multiple potential solutions more effectively.

2.2 SOMA Pareto

The Self-Organizing Migrating Algorithm Pareto (SOMA Pareto) integrates Pareto efficiency principles into the traditional Self-Organizing Migrating Algorithm (SOMA) framework. This innovation enhances the algorithm's ability to efficiently navigate through complex optimization landscapes by maintaining a balanced focus on promising solutions while fostering a diverse population. This balance is crucial for avoiding premature convergence and ensuring robust solutions across various problem types[22].

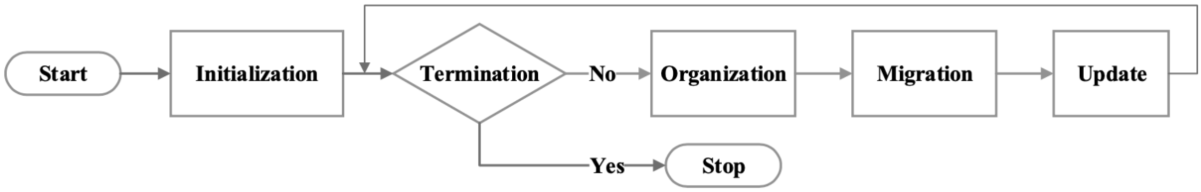


Figure 3 Pareto SOMA flowchart

2.2.1 Dynamic Parameter Adjustment in SOMA Pareto

SOMA Pareto differentiates itself by dynamically adjusting key operational parameters such as the Step size, Perturbation Vector (PRT), and the adaptation of the PRTVector. These adjustments are pivotal for the algorithm's ability to respond effectively to different phases of the optimization process:

2.2.1.1 Step Size and Perturbation Vector (PRT):

These parameters are crucial for directing the search process. In SOMA Pareto, they are adapted using a cosine function that adjusts their values based on the current stage of function evaluations. This method allows the algorithm to smoothly transition from broad exploration to intensive exploitation, optimizing the search strategy throughout the process. The adaptive control of these parameters is designed to enhance the algorithm's efficiency and effectiveness in finding optimal solutions[21][5].

2.2.1.2 PRTVector Adaptation:

Unlike other versions of SOMA, the PRTVector in SOMA Pareto is adapted to take on a range of values rather than just binary, allowing for more nuanced adjustments with each migration loop. This flexibility enables the algorithm to refine its search towards the most promising areas of the solution space, enhancing the precision of the optimization process[21][5].

2.2.2 Enhanced Migration Processes in SOMA Pareto

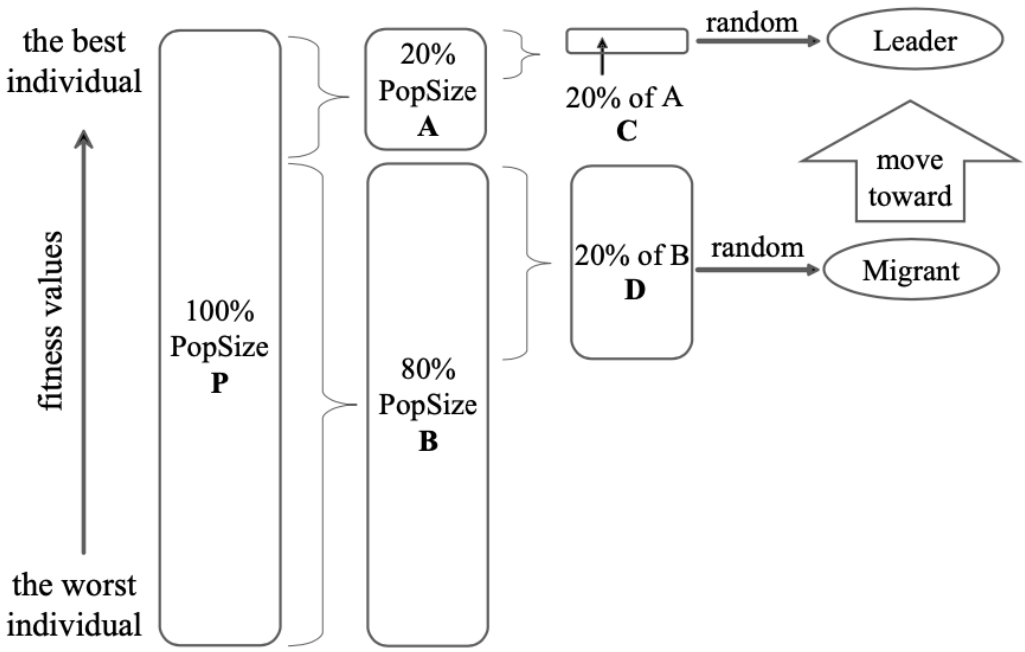


Figure 4 Pareto SOMA oranization process taken from [5]

2.2.2.1 Organization Process:

The organization phase utilizes the Pareto Principle to select individuals strategically as leaders or migrants. This selection process ensures that the migration is guided by the most promising solutions, thereby enhancing the algorithm's efficiency. By selecting a leader from the top performers and a migrant from lower ranks, the algorithm promotes genetic diversity and helps to prevent the search from stagnating in local optima[21][22].

2.2.2.2 Migration Process:

During the migration phase, the chosen migrant moves towards the leader under the guidance of dynamically adjusted Step and PRTVector parameters. This targeted migration strategy ensures effective exploration and exploitation of the search space, allowing for adaptive movement strategies based on real-time performance feedback[21][5].

2.2.2.3 Update Process:

This phase involves evaluating new positions discovered during migration. If a position proves to be an improvement, it is updated in the population. This continual improvement process ensures that each migration loop effectively contributes to the optimization process, incrementally enhancing the overall solution quality[21].

2.2.3 Performance Evaluation

The performance of SOMA Pareto has been rigorously evaluated against standard benchmarks such as the CEC'13, CEC'15, and CEC'17 suites. In these evaluations, SOMA Pareto has demonstrated capabilities over traditional SOMA versions and other contemporary algorithms. The enhanced performance is primarily attributed to its adaptive mechanisms, especially the dynamic adjustments of the PRT and Step parameters. These mechanisms enable the algorithm to adeptly handle complex and varying problem landscapes[21].

The adaptability of SOMA Pareto to dynamically adjust its search strategies based on real-time feedback significantly enhances its problem-solving capabilities, making it a powerful tool for a broad spectrum of optimization challenges. The strategic modulation of its parameters not only improves solution accuracy but also enhances computational efficiency, leading to quicker and more effective searches.

Pseudocode 2 SOMA Pareto

```
Algorithm SOMA Pareto
Generate and evaluate the initial population
while stopping condition not reached do
        Update PRT and Step
        Sort the population
        Choose the Migrant and the Leader
        The Migrant moves to the Leader
        Checking boundary
        Re-evaluate fitness function
        Updated the better position of the Migrant
end while
return
```

2.3 SOMA-CLP

The Self-Organizing Migrating Algorithm with Clustering and Population Learning (SOMA-CLP) is an innovative adaptation of the traditional Self-Organizing Migrating Algorithm (SOMA). SOMA-CLP introduces clustering mechanisms and population learning processes that enhance the algorithm's efficiency in navigating complex optimization landscapes. These enhancements focus on exploiting promising solution clusters and refining the search process through learned adaptations, crucial for solving complex, high-dimensional problems without premature convergence.

2.3.1 Dynamic Parameter Adjustment in SOMA-CLP

SOMA-CLP employs dynamic adjustments to its operational parameters, including the Step size, PathLength, and a specialized Perturbation Vector (PRT) that adapts through clustering techniques:

2.3.1.1 Step Size and PathLength:

In SOMA-CLP, these parameters are dynamically adjusted to align with the clustering results, ensuring that movements within the search space are both strategic and effective. The adjustments allow the algorithm to refine its search in promising clusters, optimizing both global and local explorations[9].

2.3.1.2 Clustering-Based PRT Adaptation:

Unlike traditional PRT adaptations, SOMA-CLP integrates clustering outcomes to adjust the PRT parameters. This method tailors the search direction based on cluster characteristics, allowing the algorithm to focus on areas with higher potential for containing optimal solutions. It enhances the algorithm's capability to escape local optima and improves convergence on global solutions[9].

2.3.2 Enhanced Migration Processes in SOMA-CLP

2.3.2.1 Organization Process:

The organization phase in SOMA-CLP leverages clustering to determine the structure of each migration loop. By classifying the population into clusters based on solution similarity, the algorithm can identify and exploit promising regions more effectively. This phase sets the stage for focused exploration and exploitation within identified clusters, enhancing the efficiency of the search process[9].

2.3.2.2 Migration Process:

During migration, individuals are guided towards cluster centroids or best solutions within their clusters, as determined by the clustering mechanism. This targeted approach is governed by dynamically adjusted Step and PRT parameters, ensuring that the migration is both directed towards areas of interest and adaptable based on the ongoing evaluation of cluster quality and solution proximity[9].

2.3.2.3 Update Process:

Post-migration, the algorithm evaluates the positions of individuals within their clusters. If a new position within the cluster offers an improvement, it is adopted, and the individual's data is updated in the population record. This process ensures continual learning and adaptation within clusters, gradually enhancing the solution quality across the population[9]

2.3.3 Performance Evaluation

SOMA-CLP's performance has been extensively tested against established benchmarks where it demonstrated significant improvements over standard SOMA and various contemporary algorithms. These enhancements are primarily attributed to its adaptive mechanisms, particularly the integration of clustering strategies and dynamic parameter adjustments that respond adeptly to complex and variable problem landscapes[9].

The ability of SOMA-CLP to dynamically adjust its search strategies based on real-time feedback and clustering results significantly enhances its problem-solving capabilities, making it highly effective for diverse and challenging optimization scenarios. The strategic use of clustering not only refines the search process but also ensures that computational resources are efficiently utilized, leading to quicker and more effective problem-solving.

Pseudocode 3 SOMA-CL

Algorithm SOMA-CL

```
Set D,NP,NP_L, and MAXFES
Set step, pathLength, and prt
Set step_L, pathLength_L, and prt_L
while Stopping criterion not met do
    M=∅
    for i=1 to NP do
        x_L= pick random solution x
        for t=0 to pathLength with t+= step do
            generate PRTVector
            migrate x_i to x_L
            save each evaluated solution into M
        end for
    end for
    k-means clustering method for solutions stored in M
    keep only best-solution from each cluster
    sort the remaining solutions in M
    for i=1 to NP do
        x_L= Rank Selection from cluster leaders
        for t=0 to pathLength_L with t+= step_L do
            generate PRTVector
            migrate x_i to x_L
        end for
    end for
    record the best solution
end while
```

3 BENCHMARK TEST SUITES

Benchmark suites typically consist of a diverse set of optimization problems, each designed to test different capabilities of algorithms. These problems may include unimodal functions for assessing convergence speed, multimodal functions to test global search capabilities and avoid local minima, and composite functions that combine several challenges to mimic real-world problem complexities. The diversity of these functions ensures that the algorithm is not only tuned to specific problem types but is adaptable and effective across a broad spectrum of scenarios.

Evaluating SOMA using these benchmarks helps in determining how well the algorithm can:

1. **Navigate and Escape Local Optima**: Important for algorithms to demonstrate the ability to escape local optima and not just converge quickly to any solution.
2. **Maintain Diversity in Solutions**: Assessing how the algorithm maintains diversity within the population to avoid premature convergence.
3. **Adapt to Dynamic Changes**: Particularly with adaptive and dynamic benchmarks that change over time, mimicking changing real-world conditions.
4. **Scalability and Efficiency**: Testing how the algorithm scales with the dimensionality of the problem and how efficiently it uses computational resources.

3.1 Justification for the Selection of CEC2017 and CEC2022 Benchmark Suites

3.1.1 CEC2017:

This benchmark suite is particularly useful for testing the adaptability and efficiency of SOMA due to its diverse set of functions that include rotated and shifted functions. These functions are beneficial in evaluating the ability of SOMA to handle complex landscapes and its sensitivity to initial conditions and parameter settings. The CEC2017 suite also introduces challenges such as rotated trap problems and features extraction from different dimensions, which are crucial for testing the robust exploration and exploitation capabilities of SOMA.

3.1.2 CEC2022:

Building on the foundations of previous suites, the CEC2022 benchmark suite often incorporates more recent advancements in problem design, including more sophisticated hybrid and composition functions that better reflect the multi-faceted nature of real-world problems. Using the CEC2022 suite provides a contemporary field of challenges that assess the latest enhancements in SOMA, such as improvements in adaptive parameters and ensemble strategies, ensuring the algorithm remains competitive against newer optimization techniques.

Using both CEC2017 and CEC2022, researchers can gain a comprehensive view of how SOMA has progressed over time and how modifications to the algorithm improve its performance against increasingly complex and realistic optimization tasks. The choice of these suites is justified by their relevance to current optimization challenges in the field and their ability to provide rigorous, varied, and up-to-date testing environments.

In conclusion, the selection of specific benchmark suites like CEC2017 and CEC2022 for evaluating SOMA is crucial for a thorough and nuanced understanding of the algorithm's capabilities, ensuring it is well-tested against the types of problems it is likely to encounter in practical applications. These benchmarks act as a proving ground to refine and validate the efficacy of SOMA in solving a broad spectrum of optimization problems

3.2 CEC 2017

The CEC2017 benchmark suite includes a diverse range of functions, each meticulously designed to test specific capabilities of optimization algorithms like SOMA. These functions encompass unimodal, multimodal, hybrid, and composition categories, providing a comprehensive assessment environment. Here is an expanded technical description of each function category[25]:

Table 1 List of CEC 2017 test functions

	No.	Functions	$F_i{}^* = \mathrm{F_i(X^*)}$
Unimodal Functions	1	Shifted and Rotated Bent Cigar Function	100
	2	Shifted and Rotated Zakharov Function	200
Simple Multimodal Functions	3	Shifted and Rotated Rosenbrock's Function	300
	4	Shifted and Rotated Rastrigin's Function	400
	5	Shifted and Rotated Expanded Scaffer's F6 Function	500
	6	Shifted and Rotated Lunacek Bi_Rastrigin Function	600
	7	Shifted and Rotated Non-Continuous Rastrigin's Function	700
	8	Shifted and Rotated Levy Function	800
	9	Shifted and Rotated Schwefel's Function	900
Hybrid Functions	10	Hybrid Function 1($N = 3$)	1000
	11	Hybrid Function 2($N = 3$)	1100
	12	Hybrid Function 3($N = 3$)	1200
	13	Hybrid Function 4($N = 4$)	1300
	14	Hybrid Function 5($N = 4$)	1400
	15	Hybrid Function 6($N = 4$)	1500
	16	Hybrid Function 6($N = 5$)	1600
	17	Hybrid Function 6($N = 5$)	1700
	18	Hybrid Function 6($N = 5$)	1800
	19	Hybrid Function 6($N = 6$)	1900
Composition Functions	20	Composition Function 1($N = 3$)	2000
	21	Composition Function 2($N = 3$)	2100
	22	Composition Function 3($N = 4$)	2200
	23	Composition Function 4($N = 4$)	2300
	24	Composition Function 5($N = 5$)	2400
	25	Composition Function 6($N = 5$)	2500
	26	Composition Function 7($N = 6$)	2600
	27	Composition Function 8($N = 6$)	2700
	28	Composition Function 9($N = 3$)	2800
	29	Composition Function 10($N = 3$)	2900
Search Range: $[-100,100]^D$			

3.2.1 Unimodal Functions

All general function used in following formulas are part of attachment I.

3.2.1.1 Shifted and Rotated Bent Cigar Function

$$F_1(\boldsymbol{x}) = f_1(\mathbf{M}(\boldsymbol{x} - \boldsymbol{o}_1)) + F_1^*$$

Primarily unimodal, this function features a long, narrow ridge which makes finding the global minimum challenging without precise coordination. It tests an algorithm's precision and ability to follow a narrow path to the optimum.

3.2.1.2 Shifted and Rotated Zakharov Function

$$F_3(\boldsymbol{x}) = f_3(\mathbf{M}(\boldsymbol{x} - \boldsymbol{o}_3)) + F_3 *$$

This function combines quadratic and higher-order terms to create a deceptive search landscape with a global minimum that is difficult to pinpoint due to misleading gradients.

3.2.2 Multimodal Functions

3.2.2.1 Shifted and Rotated Rosenbrock's Function

$$F_4(\boldsymbol{x}) = f_4\left(\mathbf{M}\left(\frac{2.048(\boldsymbol{x} - \boldsymbol{o}_4)}{100}\right)\right) + F_4 *$$

Characteristics: Known as the valley or banana function, it presents a challenging, narrow, curved valley. The function is sensitive to initial conditions and the path to the global minimum is convoluted, testing an algorithm's ability to navigate complex paths efficiently.

3.2.2.2 Shifted and Rotated Rastrigin's Function

$$F_5(\boldsymbol{x}) = f_5(\mathbf{M}(\boldsymbol{x} - \boldsymbol{o}_5)) + F_5 *$$

Characteristics: Highly multimodal with frequent local minima, making it a rigorous test for an algorithm's global search capabilities and its ability to escape local optima.

3.2.2.3 Shifted and Rotated Expanded Schaffer's F7 Function

$$F_6(\boldsymbol{x}) = f_{20}\left(\mathbf{M}\left(\frac{0.5(\boldsymbol{x} - \boldsymbol{o}_6)}{100}\right)\right) + F_6 *$$

Description: Applies the Scaffer's F6 function, a non-separable and multimodal function, across multiple dimension pairs, aggregating the results to form a challenging landscape with many local minima.

Challenges: This function tests an algorithm's fine-tuning abilities and its effectiveness across interdependent variable transformations.

3.2.2.4 Shifted and Rotated Lunacek Bi-Rastrigin Function

$$F_7(\boldsymbol{x}) = f_7\left(\mathbf{M}\left(\frac{600(\boldsymbol{x} - \boldsymbol{o}_7)}{100}\right)\right) + F_7 *$$

Description: Combines a multimodal Rastrigin function with a unimodal quadratic function, featuring dual global optima. It is designed to challenge an algorithm's ability to differentiate between multiple promising areas in a search space.

Challenges: Effective in testing robustness and reliability in maintaining convergence stability across dual potential solutions.

3.2.2.5 Shifted and Rotated Non-Continuous Rastrigin's Function

$$F_8(\boldsymbol{x}) = f_8\left(\left(\frac{5.12(\boldsymbol{x} - \boldsymbol{o}_8)}{100}\right)\right) + F_8 *$$

Description: Introduces gaps or discontinuities into the classic Rastrigin's function, complicating the optimization process by breaking the problem space continuity.

Challenges: Tests an algorithm's ability to handle irregular and broken landscapes without losing track of the global context.

3.2.2.6 Shifted and Rotated Levy Function

$$F_9(\boldsymbol{x}) = f_9\left(\mathbf{M}\left(\frac{5.12(\boldsymbol{x} - \boldsymbol{o}_9)}{100}\right)\right) + F_9 *$$

Description: Characterized by a complex, wavy landscape generated by the Levy distribution, this function is known for its large search spaces and multiple local optima.

Challenges: The function is a good test for an algorithm's endurance in extensive searches and its capacity to resolve intricate, large-scale optimization problems.

3.2.2.7 Shifted and Rotated Schwefel's Function

$$F_{10}(\boldsymbol{x}) = f_{10}\left(\mathbf{M}\left(\frac{1000(\boldsymbol{x} - \boldsymbol{o}_{10})}{100}\right)\right) + F_{10} *$$

Characteristics: Features a global optimum near the boundaries of the domain with numerous misleading local optima. It tests an algorithm's ability to effectively discriminate between suboptimal and optimal solutions in large search ranges.

3.2.3 Hybrid Functions

Description: These functions blend two or more of the basic benchmark functions, creating a complex landscape that benefits from the strengths and challenges of the included functions. This configuration is intended to simulate real-world problems where multiple strategies may be necessary to locate the optimal solution.

Challenges: They test an algorithm's flexibility and adaptability, evaluating how well it can switch between strategies to tackle different problem facets simultaneously.

$$F(\boldsymbol{x}) = g_1(\mathbf{M}_1\mathbf{z}_1) + g_2(\mathbf{M}_2\mathbf{z}_2) + \cdots + g_N(\mathbf{M}_N\mathbf{z}_N) + F^*(\boldsymbol{x})$$

Where:

F(x): hybrid function

g_1(x): ith basic function used to construct the hybrid function

N: number of basic functions

$z = z_{1,}z_2, \dots, z_N$

$$\boldsymbol{z}_1 = [\boldsymbol{y}_{S_1}, \boldsymbol{y}_{S_2}, \dots, \boldsymbol{y}_{S_{n_1}}], \boldsymbol{z}_2 = [\boldsymbol{y}_{S_{n_1+1}}, \boldsymbol{y}_{S_{n_1+2}}, \dots, \boldsymbol{y}_{S_{n_1+n_2}}], \dots, \boldsymbol{z}_N = [\boldsymbol{y}_{S_{\sum_{i=1}^{N-1} n_i+1}}, \boldsymbol{y}_{S_{\sum_{k=1}^{N-1} n_i+2}}, \dots, \boldsymbol{y}_{S_D}]$$

$\boldsymbol{y} = \boldsymbol{x} - \boldsymbol{o}_i, S = \text{randperm}(1:D)$

p_i: used to control the percentage of $g_i(\boldsymbol{x})$

n_i: dimension for each basic function $\sum_{i=1}^{N} n_i = D$

$$n_1 = \lceil p_1 D \rceil, n_2 = \lceil p_2 D \rceil, \dots, n_{N-1} = \lceil p_{N-1} D \rceil, n_N = D - \sum_{i=1}^{N-1} n_i$$

3.2.3.1 Hybrid Function 1

This hybrid function tests the algorithm's capability to handle a diverse range of landscapes by combining three distinct functions that require different optimization strategies.

Basic Functions Used: Zakharov Function, Rosenbrock Function, Rastrigin's Function

3.2.3.2 Hybrid Function 2

Incorporates a mix of functions with varying properties, from smooth to rugged landscapes, emphasizing an algorithm's adaptability and robustness.

Basic Functions Used: High Conditioned Elliptic Function, Modified Schwefel's Function, Bent Cigar Function

3.2.3.3 Hybrid Function 3

Focuses on combining functions that offer a balance of global and local search challenges, testing both the exploration and exploitation phases of the optimization algorithms.

Basic Functions Used: Bent Cigar Function, Rosenbrock Function, Lunacek Bi-Rastrigin Function

3.2.3.4 Hybrid Function 4

Designed to evaluate how well the optimization algorithm can navigate through composite landscapes with multiple conflicting goals.

Basic Functions Used: High Conditioned Elliptic Function, Ackley's Function, Schaffer's F7 Function, Rastrigin's Function

3.2.3.5 Hybrid Function 5

Challenges algorithms with a mix of functions that are both separable and non-separable, testing the algorithm's ability to adapt its strategies across different dimensions.

Basic Functions Used: Bent Cigar Function, HGBat Function, Rastrigin's Function, Rosenbrock's Function

3.2.3.6 Hybrid Function 6

Another complex configuration, focusing on the algorithm's ability to efficiently transition between functions with different characteristics.

Basic Functions Used: Expanded Schaffer F6 Function, HGBat Function, Rosenbrock's Function, Modified Schwefel's Function

3.2.3.7 Hybrid Function 7

A more intricate combination designed to push the optimization techniques to their limits across a broader set of function properties.

Basic Functions Used: Katsuura Function, Ackley's Function, Expanded Griewank's plus Rosenbrock's Function, Modified Schwefel's Function, Rastrigin's Function

3.2.3.8 Hybrid Function 8

Tests the algorithms against a backdrop of highly competitive landscapes, demanding high precision and strategic diversity.

Basic Functions Used: High Conditioned Elliptic Function, Ackley's Function, Rastrigin's Function, HGBat Function, Discus Function

3.2.3.9 Hybrid Function 9

Focuses on evaluating the robustness and effectiveness of algorithms under conditions of extreme multimodality and variable landscape ruggedness.

Basic Functions Used: Bent Cigar Function, Rastrigin's Function, Expanded Griewank's plus Rosenbrock's Function, Weierstrass Function, Expanded Schaffer's F6 Function

3.2.3.10 Hybrid Function 10

The most complex of the hybrid functions, designed to simulate real-world scenarios with multiple objectives and conflicting requirements.

Basic Functions Used: Happycat Function, Katsuura Function, Ackley's Function, Rastrigin's Function, Modified Schwefel's Function, Schaffer's F7 Function

3.2.4 Composition Functions

Composition functions are complex benchmarks that combine several basic functions with varying weights and transformations. These functions are designed to simulate the overlapping influence of different problem aspects in real-world scenarios.

They assess the ability of an algorithm to manage and balance multiple problem-solving approaches concurrently, providing insights into its effectiveness in composite optimization scenarios.

Each function in the CEC2017 suite is intricately designed to push optimization algorithms to their limits, offering insights into their performance across a broad spectrum of challenges, making this suite an invaluable tool for benchmarking state-of-the-art optimization techniques.

$$F(x) = \sum_{i=1}^{N} \{\omega_i^* [\lambda_i g_i(x) + \text{ bias }_i]\} + F^*$$

$F(x)$: composition function

$g_{\text{i}}(\boldsymbol{x})$: i^{th} basic function used to construct the composition function

N : number of basic functions

o_i: new shifted optimum position for each $g_i(\boldsymbol{x})$, define the global and local optima's posi-tion

biasi: defines which optimum is global optimum

σ_i : used to control each $g_i(x)$'s coverage range, a small σ_i give a narrow range for that $g_i(x)$

λ_i: used to control each $g_i(\boldsymbol{x})$'s height

w_i: weight value for each $g_i(\boldsymbol{x})$, calculated as below:

$$w_i = \frac{1}{\sqrt{\sum_{j=1}^{D} \left(x_j - o_{ij}\right)^2}} \exp\left(-\frac{\sum_{j=1}^{D} \left(x_j - o_{ij}\right)^2}{2D\sigma_i^2}\right)$$

Then normalize the weight $\omega_i = w_i / \sum_{i=1}^{n} w_i$

So when $\boldsymbol{x} = \boldsymbol{o}_i, \omega_j = \begin{cases} 1 & j = i \\ 0 & j \neq i \end{cases}$ for $j = 1,2,\ldots,N, f(x) = \text{bias}_{\ i} + f^{\,*}$

3.2.4.1 *Composition Function 1*

This composition function assesses the algorithm's performance across varied modalities by blending multiple function types, focusing on balancing global and local optimization challenges.

Basic Functions Used: Rotated Rosenbrock Function, High Conditioned Elliptic Function,

Rotated Bent Cigar Function

3.2.4.2 *Composition Function 2*

Designed to test the ability of algorithms to integrate and optimize across functions with contrasting properties such as ruggedness and multiple local optima.

Basic Functions Used: Rotated Schwefel Function, Rotated Rastrigin Function, Rotated HGBat Function

3.2.4.3 Composition Function 3

Challenges optimization algorithms with a complex blend of functions that require nuanced strategies to navigate effectively.

Basic Functions Used: Rotated Rosenbrock Function, Rotated High Conditioned Elliptic Function, Rotated Bent Cigar Function, Rotated Discus Function

3.2.4.4 Composition Function 4

This function is a sophisticated mixture aiming to simulate the overlapping influence of different problem characteristics in real-world scenarios.

Basic Functions Used: Rotated Schwefel Function, Rotated Rastrigin Function, Rotated HGBat Function, Rotated Discus Function

3.2.4.5 Composition Function 5

A highly challenging composition that tests algorithms on their ability to deal with a broad range of dynamic changes in the optimization landscape.

Basic Functions Used: Rotated Rosenbrock Function, High Conditioned Elliptic Function, Rotated Bent Cigar Function, Rotated HGBat Function, Rotated Discus Function

3.2.4.6 Composition Function 6

Evaluates the adaptability and robustness of optimization algorithms by combining several distinct types of functions, each demanding different strategic approaches.

Basic Functions Used: Rotated Rosenbrock Function, Rotated Schwefel Function, Rotated Rastrigin Function, Rotated Levy Function, Rotated HGBat Function

3.2.4.7 Composition Function 7

Tests the algorithm's performance in a highly composite environment where navigating through multiple function characteristics is crucial for finding the global optimum.

Basic Functions Used: Rotated Schwefel Function, Rotated Rastrigin Function, Rotated HGBat Function, Rotated Discus Function, Rotated Rosenbrock Function, Rotated High Conditioned Elliptic Function

3.2.4.8 Composition Function 8

This function is designed to challenge the computational efficiency and strategic depth of algorithms by involving a mixture of highly complex landscapes.

Basic Functions Used: Rotated Schwefel Function, Rotated Rastrigin Function, Rotated HGBat Function, Rotated Discus Function, Rotated Levy Function, Rotated High Conditioned Elliptic Function

3.2.4.9 Composition Function 9

Focuses on the algorithm's ability to efficiently switch between different optimization modes and strategies to handle a combination of function properties.

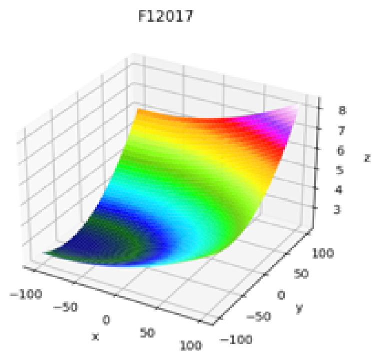
Basic Functions Used: Rotated Schwefel Function, Rotated Rastrigin Function, Rotated Levy Function

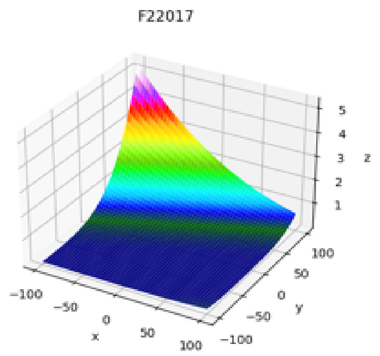
3.2.4.10 Composition Function 10

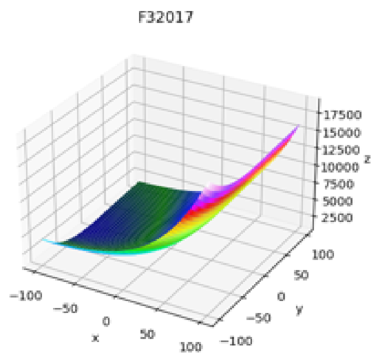
The final composition function in the series tests the overall robustness and flexibility of optimization algorithms across a trio of diverse and complex functions.

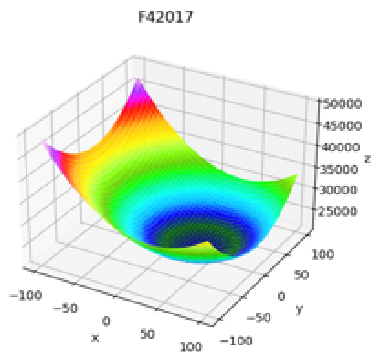
Basic Functions Used: Rotated Schwefel Function, Rotated Rastrigin Function, Rotated HGBat Function

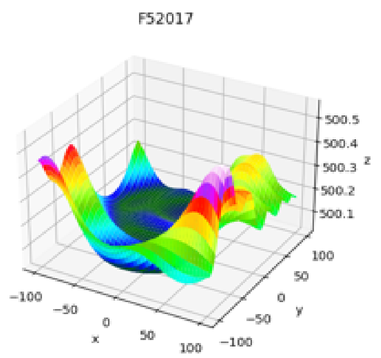
Table 2 list of CEC 2017 test functions 3D graphs

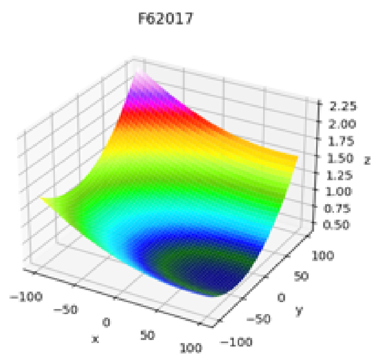


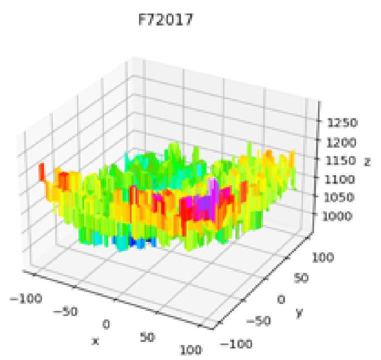


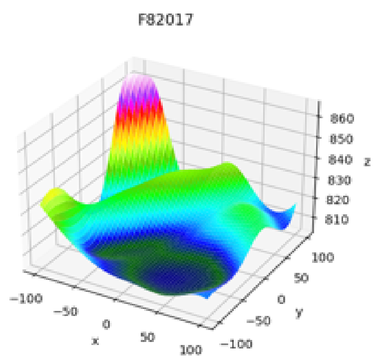


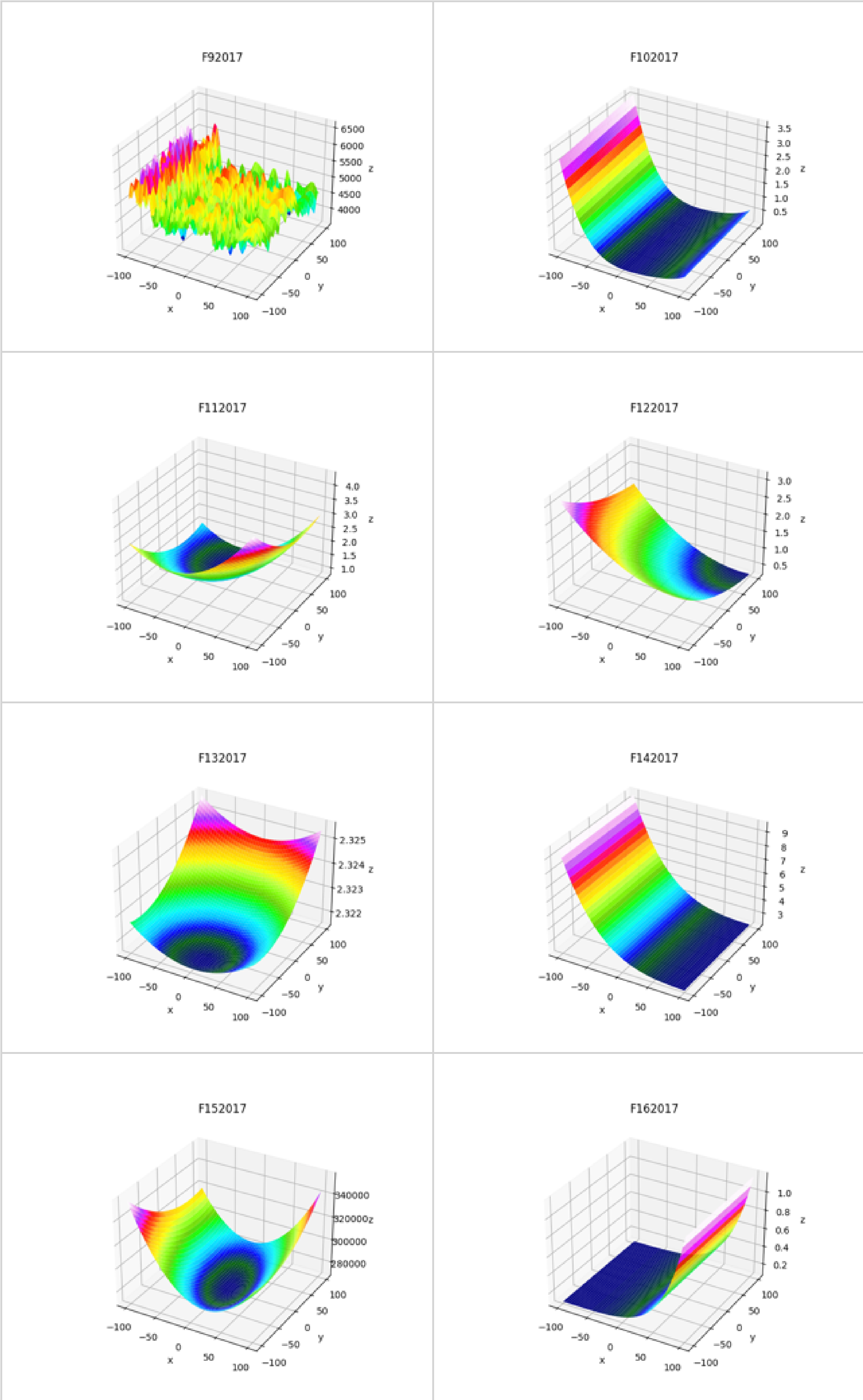


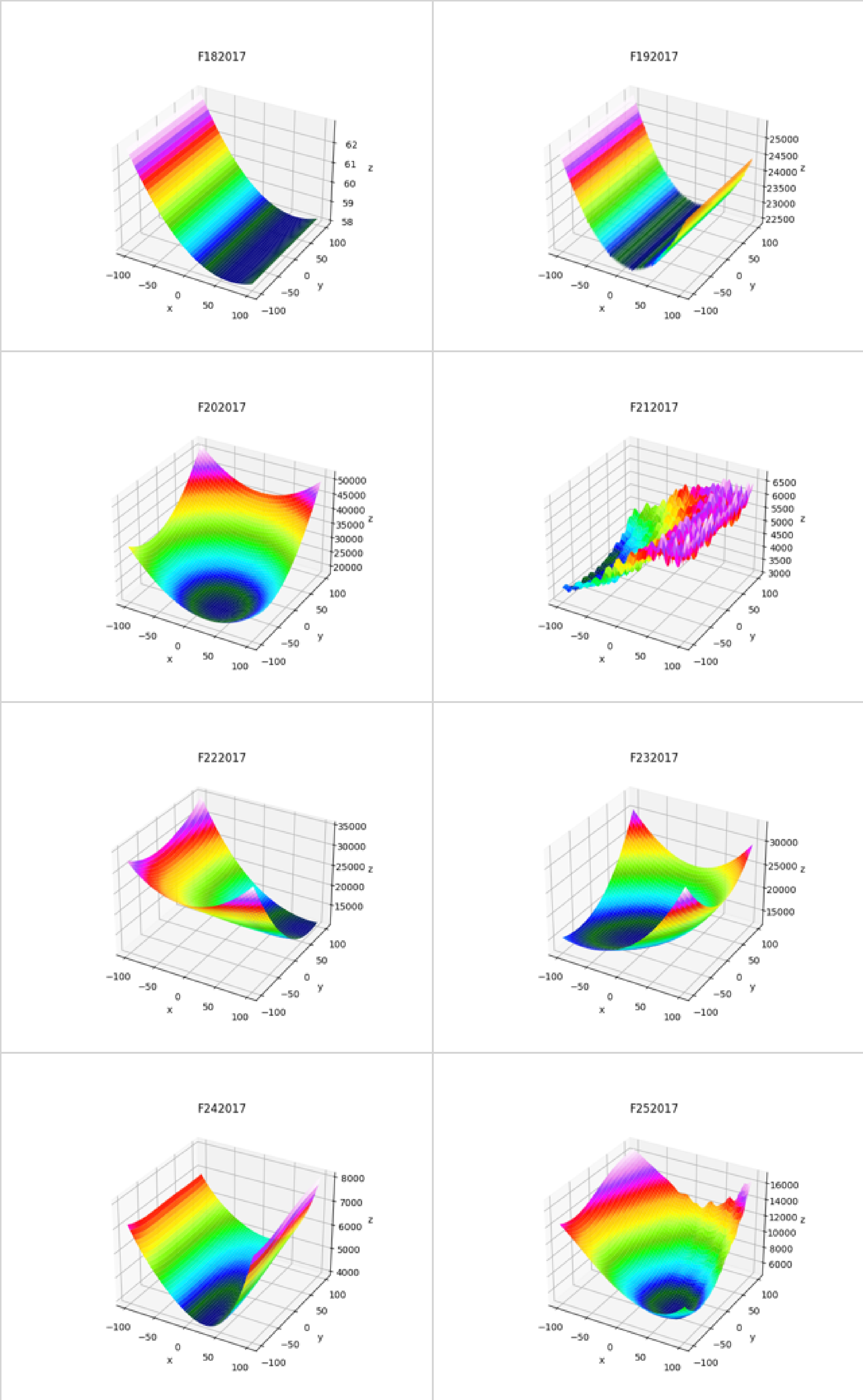


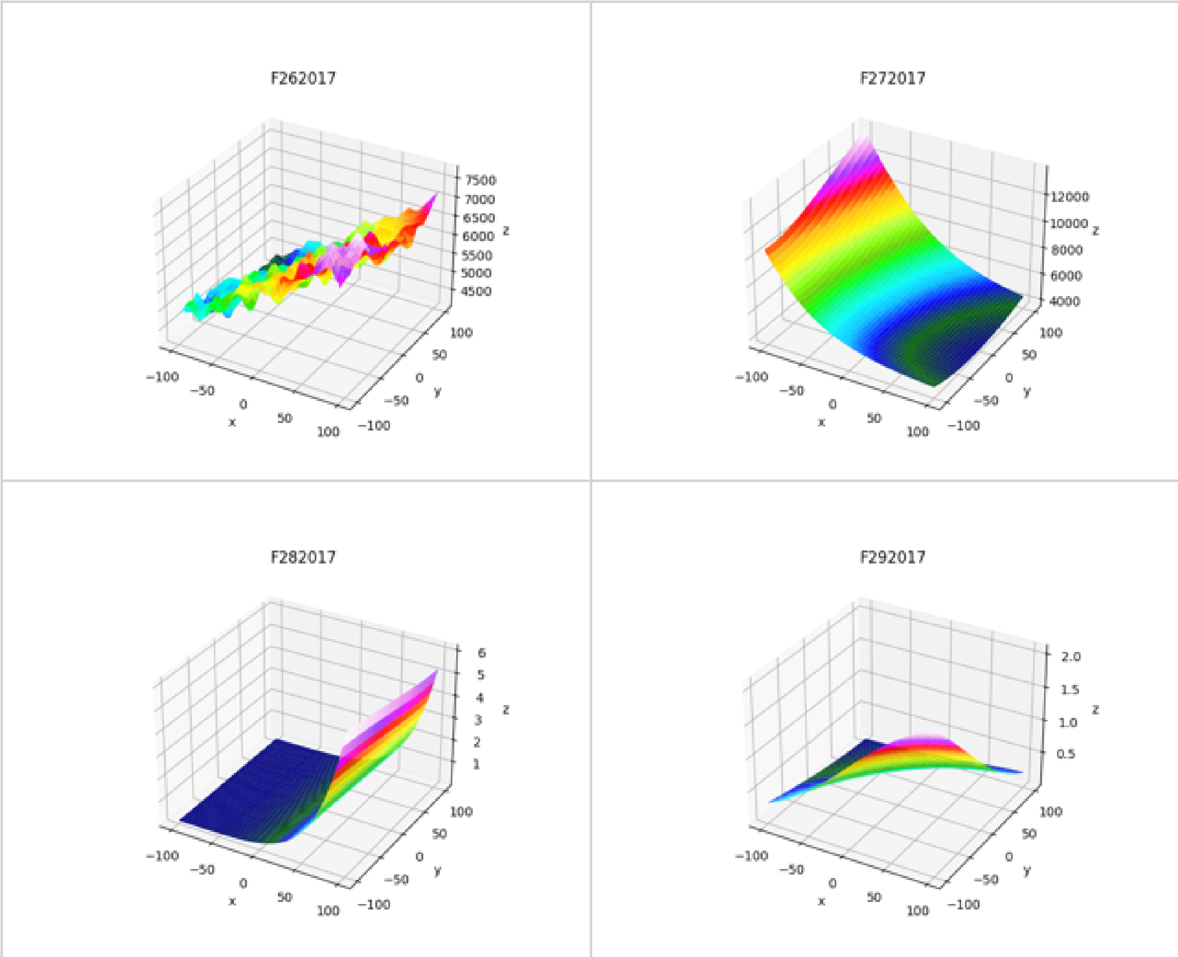







F92017
F102017
F112017
F122017
F132017
F142017
F152017
F162017
x
y
z


F182017
F192017
F202017
F212017
F222017
F232017
F242017
F252017



Graphs were obtained using script graphs.py which is part of digital attachment.

3.3 CEC2022

The CEC 2022 benchmark suite for the Special Session and Competition on Single Objective Bound Constrained Numerical Optimization includes a diverse set of functions designed to test the effectiveness and robustness of optimization algorithms. The suite is structured to assess algorithms across a variety of challenges, featuring unimodal, multimodal, and hybrid functions, each with unique characteristics that mimic real-world optimization problems. [26]

Table 3 list of CEC 2022 test functions

	No.	Functions	F_i^*
Unimodal Function	1	Shifted and full Rotated Zakharov Function	300
Basic Functions	2	Shifted and full Rotated Rosenbrock's Function	400
	3	Shifted and full Rotated Expanded Schaffer's $f6$ Function	600
	4	Shifted and full Rotated Non-Continuous Rastrigin's Function	800
	5	Shifted and full Rotated Levy Function	900
Hybrid Functions	6	Hybrid Function 1$(N = 3)$	1800
	7	Hybrid Function 2$(N = 6)$	2000
	8	Hybrid Function 3$(N = 5)$	2200
Composition Functions	9	Composition Function 1$(N = 5)$	2300
	10	Composition Function 2$(N = 4)$	2400
	11	Composition Function 3$(N = 5)$	2600
	12	Composition Function 4$(N = 6)$	2700
Search range: $[-100,100]^D$			

3.3.1 Basic Functions

All general function used in following formulas are part of attachment II.

3.3.1.1 Shifted and Rotated Zakharov Function

$$F_1(x) = f_1\big(M(x - o_1)\big) + F_1^*$$

Characteristics: Unimodal and non-separable.

Challenge: It tests the precision and ability to converge to a global minimum without being misled by the landscape.

3.3.1.2 Shifted and Rotated Rosenbrock's Function

$$F_2(x) = f_2\left(M\left(\frac{2.048(x - o_2)}{100}\right) + 1\right) + F_2^*$$

Characteristics: Multi-modal and non-separable.

Challenge: Known for its narrow, curved valley, it requires careful navigation to avoid local minima.

3.3.1.3 Shifted and Rotated Expanded Schaffer's F7

$$F_3(x) = f_3\left(M\left(\frac{0.5(x - o_3)}{100}\right)\right) + F_3^*$$

Characteristics: Multi-modal and non-separable, with an asymmetrical landscape.

Challenge: The function's complexity requires algorithms to effectively explore and exploit multiple regions for global optimization.

3.3.1.4 Shifted and Rotated Non-Continuous Rastrigin's Function

$$F_4(x) = f_4\left(M\left(\frac{5.12(x - o_4)}{100}\right)\right) + F_4^*$$

Characteristics: Multi-modal, non-separable, and includes discontinuities.

Challenge: The discontinuities add an additional layer of complexity, challenging the algorithm's ability to handle abrupt changes in the landscape.

3.3.1.5 Shifted and Rotated Levy Function

$$F_5(x) = f_5\left(M\left(\frac{5.12(x - o_5)}{100}\right)\right) + F_5^*$$

Characteristics: Multi-modal and non-separable.

Challenge: Features a complex landscape with many sharp peaks and deep valleys, testing the global search capabilities of algorithms.

3.3.2 Hybrid Functions

$$F(\mathbf{x}) = g_1(M_1 z_1) + g_2(M_2 z_2) + \cdots + g_N(M_N z_N) + F^*(\mathbf{x})$$

$F(\mathbf{x})$: hybrid function

$g_i(\mathbf{x})$: i^{th} basic function used to construct the hybrid function

N : number of basic functions

$z = [z_1, z_2, \ldots, z_N]$

$$z_1 = \left[y_{S_1}, y_{S_2}, \dots, y_{S_{n_1}}\right], z_2 = \left[y_{S_{n_1+1}}, y_{S_{n_1+2}}, \dots, y_{S_{n_1+n_2}}\right], \dots, z_N = \left[y_{S_{\sum_{i=1}^{N-1} n_i+1}}, y_{S_{\sum_{i=1}^{N-1} n_i+2}}, \dots, y_{S_D}\right]$$

$y = x - o_i, S = \text{randperm}\,(1{:}\,D)$

p_i : used to control the percentage of $g_i(\boldsymbol{x})$

n_i : dimension for each basic function $\sum_{i=1}^{N} n_i = D$

$n_1 = \lceil p_1 D \rceil, n_2 = \lceil p_2 D \rceil, \dots, n_{N-1} = \lceil p_{N-1} D \rceil, n_N = D - \sum_{i=1}^{N-1} n_i$

3.3.2.1 Hybrid Function 1

Components: Combines the Bent Cigar Function, HGBat Function, and Rastrigin's Function.

Challenge: Requires handling varied function characteristics simultaneously, demanding adaptability and robustness from the optimization algorithm.

3.3.2.2 Hybrid Function 2

Components: Integrates HGBat Function, Katsuura Function, Ackley's Function, Rastrigin's Function, Modified Schwefel's Function, and Schaffer's F7 Function.

Challenge: The complexity of combining multiple distinct landscapes tests the algorithm's ability to switch strategies and adapt to different challenges effectively.

3.3.2.3 Hybrid Function 3

Components: Features a mix of Katsuura Function, HappyCat Function, Expanded Griewank's plus Rosenbrock's Function, Modified Schwefel's Function, and Ackley's Function.

Challenge: With a blend of high modality and diverse mathematical properties, this function assesses the depth and flexibility of optimization strategies.

3.3.3 Composition Functions

$$F(\mathbf{x}) = \sum_{i=1}^{N} \{\omega_i^* [\lambda_i g_i(\mathbf{x}) + \text{ bias }_i]\} + F^*$$

$F(\mathbf{x})$: composition function

$g_i(\mathbf{x})$: i^{th} basic function used to construct the composition function

N : number of basic functions

o_i : new shifted optimum position for each $g_i(\mathbf{x})$, define the global and local optima's position

bias $_i$: defines which optimum is global optimum

σ_i : used to control each $g_i(x)$'s coverage range, a small σ_i gives a narrow range for that $g_i(x)$

λ_i : used to control each $g_i(x)$'s height

ω_i : weight value for each $g_i(x)$, calculated as below:

$$w_i = \frac{1}{\sqrt{\sum_{j=1}^{D} \left(x_j - o_{ij}\right)^2}} \exp\left(-\frac{\sum_{j=1}^{D} \left(x_j - o_{ij}\right)^2}{2D\sigma_i^2}\right)$$

Then normalize the weight $\omega_i = w_i / \sum_{i=1}^{n} w_i$

So when $\mathbf{x} = \boldsymbol{o}_i, \omega_j = \begin{cases} 1 & j = i \\ 0 & j \neq i \end{cases}$ for $j = 1,2, \dots, N, f(\mathbf{x}) = \text{bias}\ _i + f^*$.

3.3.3.1 Composition Function 1

Components: Utilizes Rotated Rosenbrock's Function, High Conditioned Elliptic Function, Rotated Bent Cigar Function, Rotated Discus Function, and another High Conditioned Elliptic Function.

Challenge: This function tests the algorithm's ability to deal with a layered approach where each sub-function contributes differently to the composite landscape.

3.3.3.2 Composition Function 2

Components: Composed of Rotated Schwefel's Function, Rotated Rastrigin's Function, and HGBat Function.

Challenge: Focuses on evaluating the algorithm's effectiveness in navigating and optimizing a composition of functions with varying modalities and separabilities.

3.3.3.3 Composition Function 3

Components: Includes Expanded Schaffer's F6 Function, Modified Schwefel's Function, Griewank's Function, Rosenbrock's Function, and Rastrigin's Function.

Challenge: Challenges the algorithms with its complex, non-separable nature and the need to manage different scales of problem characteristics.

3.3.3.4 Composition Function 4

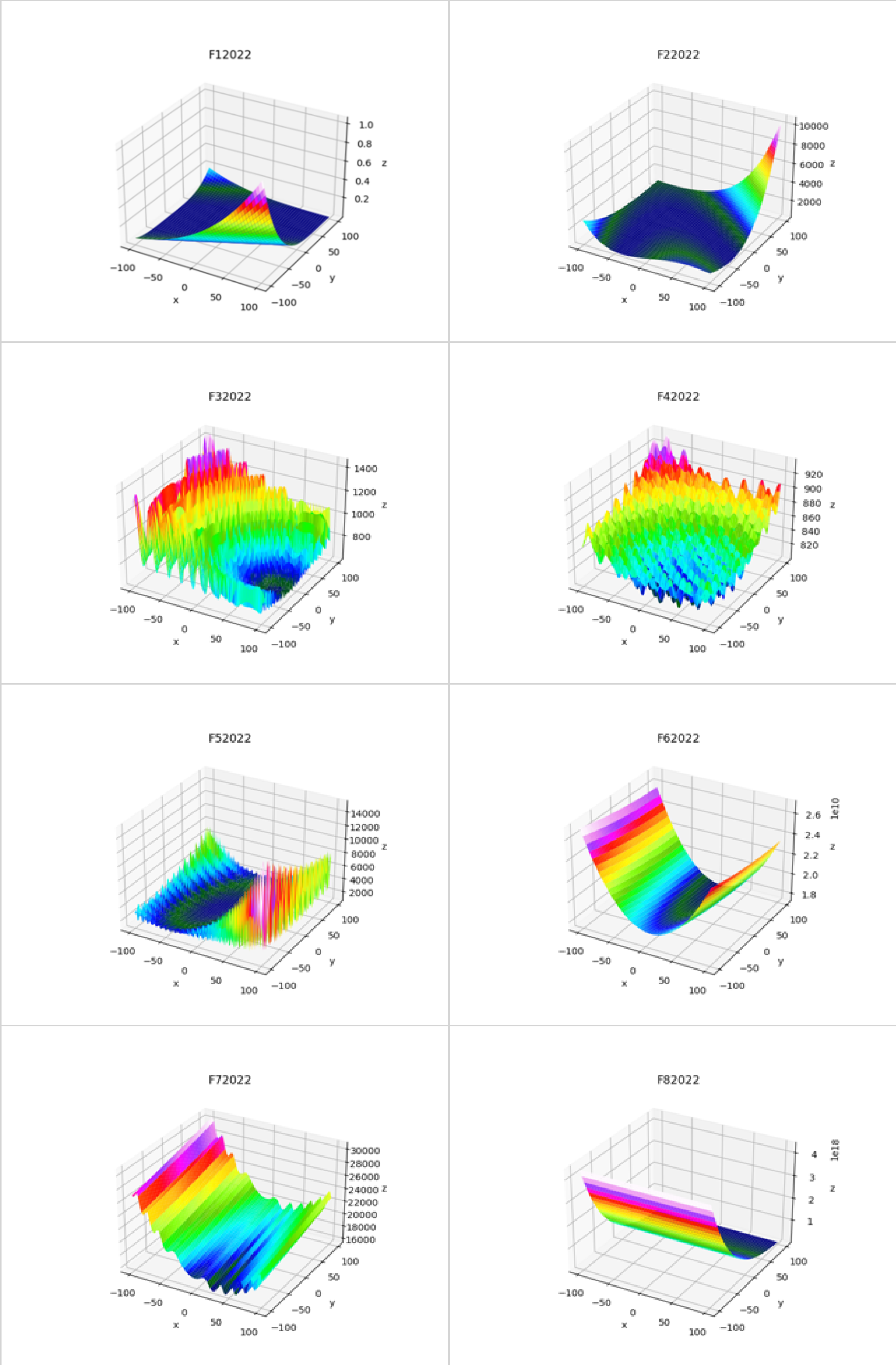
Components: Merges HGBat Function, Rastrigin’s Function, Modified Schwefel's Function, Bent Cigar Function, High Conditioned Elliptic Function, and Expanded Schaffer’s F6 Function.

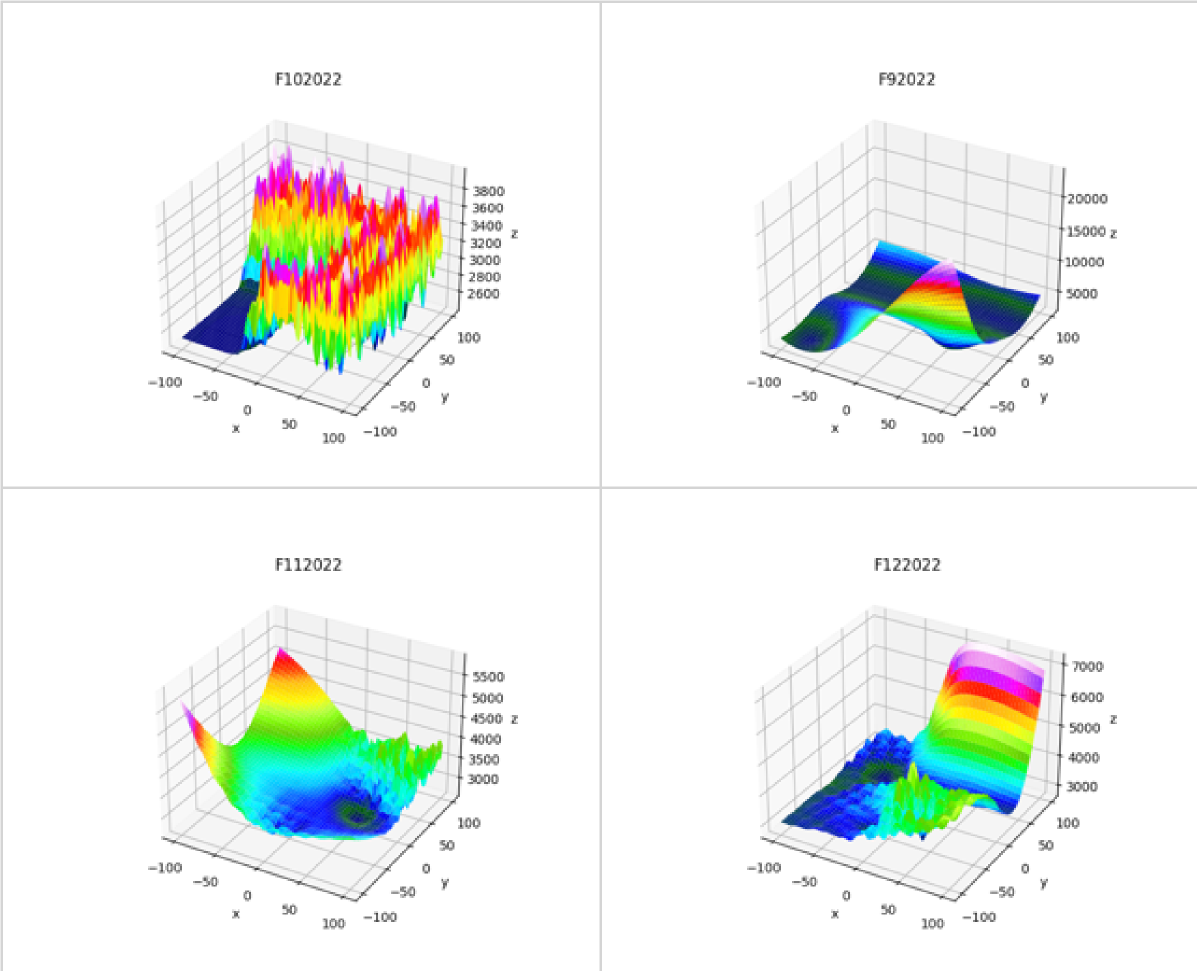
Challenge: Tests the algorithm's capability to integrate solutions from varied sources and effectively manage multiple challenges within a single optimization task.

These functions collectively offer a comprehensive testbed for evaluating the strengths and weaknesses of various optimization algorithms, focusing on their ability to handle complex, real-world-like scenarios effectively. The diversity of the functions ensures that algorithms are not only tested on their speed and accuracy but also on their robustness and adaptability across different types of optimization landscapes.

All following graphs were obtained using script graphs.py which is part of digital attachment.

Table 4 list of CEC 2022 test functions 3D graphs




F102022
3800
3600
3400
3200
3000
2800
2600
z
−100
−50
0
50
100
x
100
50
0
−50
−100
y
F92022
20000
15000
10000
5000
z
−100
−50
0
50
100
x
100
50
0
−50
−100
y
F112022
5500
5000
4500
4000
3500
3000
z
−100
−50
0
50
100
x
100
50
0
−50
−100
y
F122022
7000
6000
5000
4000
3000
z
−100
−50
0
50
100
x
100
50
0
−50
−100
y

4 THE IRACE PACKAGE

Iterated race is an extension of the Iterated F-race method for the automatic configuration of optimization algorithms, that is, (offline) tuning their parameters by finding the most appropriate settings given a set of instances of an optimization problem [27]

4.1 Overview of Algorithm Configuration and Iterated Racing

Optimizing algorithm performance through parameter tuning, known as algorithm configuration, is crucial in computational optimization. This process involves adjusting algorithmic settings to improve efficiency and effectiveness across various problems. The inherent stochastic nature of both algorithms and problem instances makes this task complex, requiring robust methodologies for effective tuning.[28]

Iterated Racing (irace), an advancement on the traditional F-Race algorithm[28], employs statistical methods to iteratively compare multiple configurations. This approach focuses computational resources on promising configurations and refines the selection pool through successive iterations, enhancing the tuning process's overall efficiency and robustness.

4.2 Fundamental Work of irace

The core functionality of Irace lies in its unique application of racing algorithms to automate the tuning of algorithmic parameters. Originating from the F-Race algorithm, Irace extends this concept by applying a more iterative approach where multiple rounds of racing are used to gradually refine the selection of configurations. This method relies heavily on statistical tests to eliminate underperforming configurations at each iteration, thereby narrowing down the candidate pool to the most viable options[27].

This exploration delves into the iterated application of the F-Race method, examining Irace's functionalities and its application in complex algorithm configurations. By investigating Irace's diverse sampling strategies, such as random sampling and factorial designs, and statistical tools like the non-parametric Friedman test, we gain a comprehensive understanding of how iterative refinement can significantly improve configuration processes.

These methodologies enable Irace to efficiently refine and select optimal configurations, demonstrating notable advancements over traditional methods.

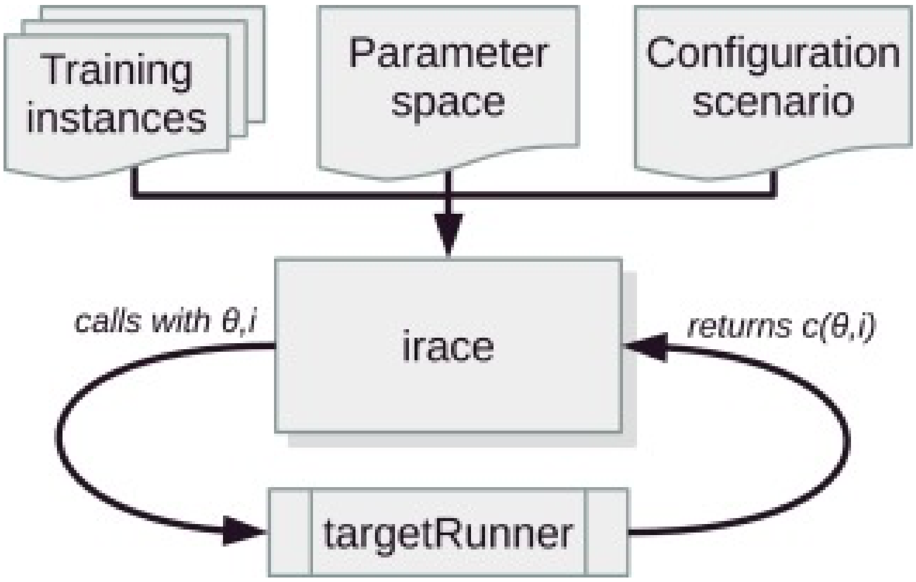


Figure 5 Workflow of irace package [27]

II. ANALYSIS

5 OPTIMIZATION, TESTING AND EVALUATION WORKFLOW

In this chapter we explore workflow of testing and refining the Self-Organizing Migrating Algorithm (SOMA). We begin by discussing the origin of the foundational code and the enhancements tailored for specific computational tasks. The focus then shifts to the testing strategy, detailing the selection of test suites and benchmarks essential for evaluating performance. We also explain the use of statistical methods to analyze results.

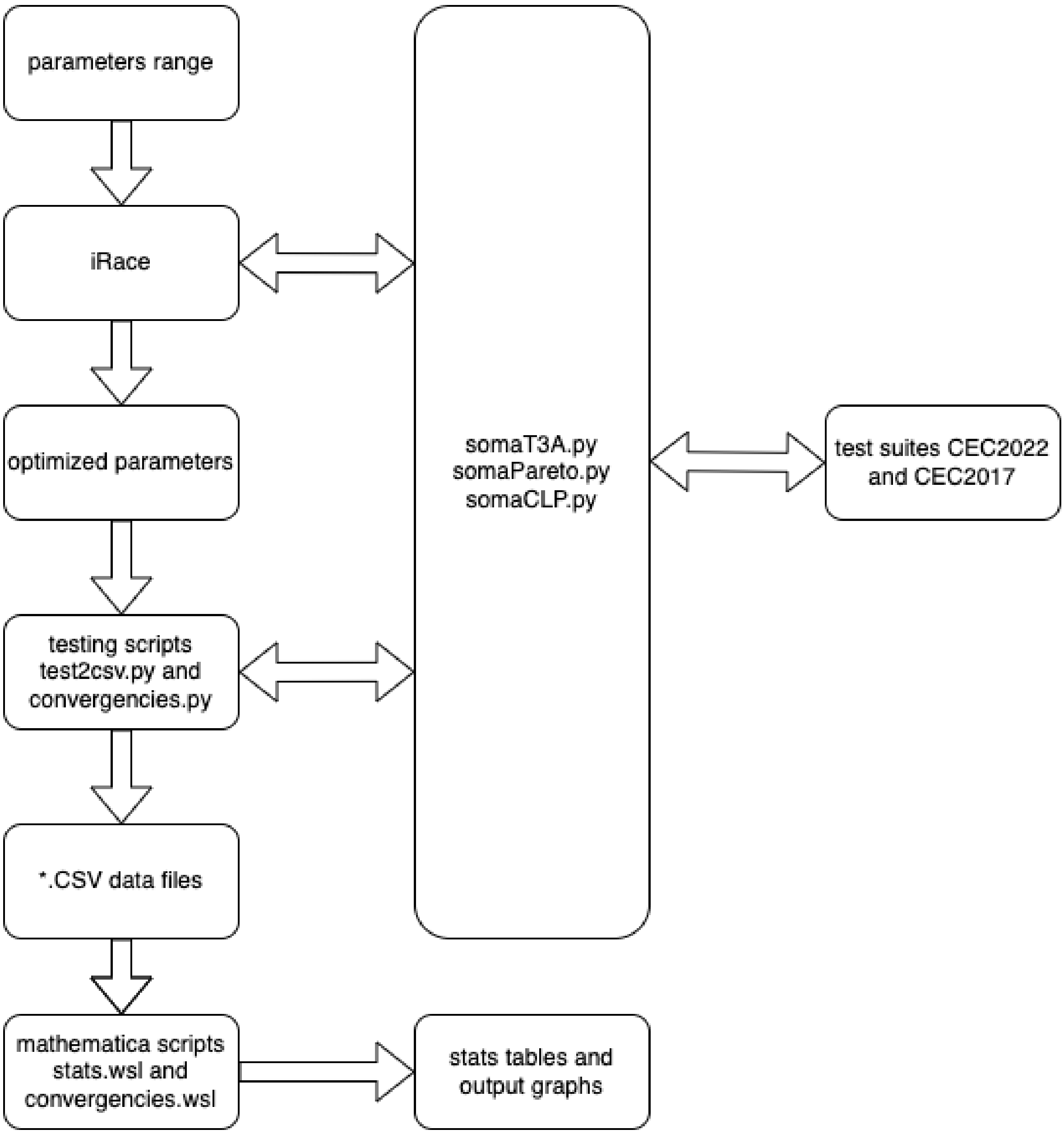


Figure 6 Optimization,testing and evaluation workflow

5.1 SOMA algorithms codes

The foundational code for somaT3a.py and somaPareto.py was adapted from originally Python implementation available on the webpage of SOMA algorithm creator Ivan Zelinka [29]. These codes underwent significant modifications to suit specific project requirements, such as enabling parameter-driven execution, incorporating test functions evaluation that is compatible with various test suites, and introducing feature toggles—for instance, a toggle to return data on algorithm convergence instead of just the final fitness value. Code of SomaCLP.py was developed from scratch due to the absence of reliable soure codes at the time of creation.

5.2 irace package

For the installation, prerequisites, and parameter settings of the irace package, comprehensive guidance is provided in the irace package user guide, available at official irace package web [30]. This document is instrumental for users aiming to employ irace for algorithm configuration and optimization tasks effectively.

5.3 Testing and evaluation

In the context of testing and evaluation, various test suites were considered. These included repositories such as cec2017-py repository (with 67 stars on GitHub), opfunu repository (with 101 stars on GitHub), and 2022-SO-BO repository (with 15 stars on GitHub) [31][32][33]. The selection and implementation of the correct functions and test functions were meticulously verified through cross-testing and evaluation processes. The final parameter optimization and testing were primarily conducted using the CEC2017 benchmark from the opfunu repository [32]and the original implementation of the CEC2022 benchmark from the 2022-SO-BO repository[33]. Furthermore, the output data were processed using scripts written in Wolfram Language. These scripts generated comprehensive tables displaying statistical metrics such as minimum, maximum, mean, median, and standard deviation values. Additionally, they produced tables for the Wilcoxon Rank Sum Test with respective p-values and for the Friedman Rank Test, highlighting the Nemenyi critical distance. This detailed analysis aids in the rigorous evaluation of algorithmic performance across various parameter settings and conditions.

All codes were run on computer with following configuration, Intel® Core™ i5-8500T CPU @ 2.10GHz × 6, 32GB RAM, Ubuntu 22.04.4 LTS with Python 3.11.

All the relevant codes have been attached as a digital attachment, ensuring that they are readily accessible for review and further research.

6 SOMAT3A

Tables of default parameters obtained from [10] and tuned from irace package

Table 5 Parameters obtained for CEC 2017 test suite

parameter	m	n	k	nJumps
default	10	4	10	45
1st	8	4	12	32
2nd	8	6	11	34
3rd	9	4	15	22

Table 6 Parameters obtained for CEC 2022 test suite

parameter	m	n	k	nJumps
default	10	4	10	45
1st	32	8	4	12
2nd	34	8	6	11
3rd	22	9	4	15

6.1 CEC 2017 test suite:

Four Tables of Basic Statistics (Minimum, Maximum, Mean, Median, and Standard Deviation) for cost function error for officially recommended parameter settings, and the three best parameter values obtained from irace, each function was runed 50 times:

Table 7 Basic statistics for default parameters

default					
Function	Min	Max	Mean	Median	Standard Deviation
F12017	550.96	25867056.83	689786.69	32575.66	3657197.22
F22017	1.43	518.76	136.77	61.13	144.79
F32017	32.96	155.82	86.02	85.08	30.96
F42017	61.96	183.91	111.38	111	31.85
F52017	0	0	0	0	0
F62017	108.5	1791.12	277.13	228.66	232.15
F72017	63.9	179.46	135.37	134.49	22.81
F82017	0	2.98	0.24	0.1	0.45
F92017	2277.27	3913.71	2919.02	2907.47	359.45
F102017	176.73	15353.2	4188.94	3118.26	3882.95
F112017	5039.68	332718.65	51297.76	36094.28	59467.82
F122017	196.43	13539.05	3544.71	2275.1	3174.68
F132017	2290.52	630278.44	96738.95	41828.72	128290.8
F142017	1371.68	24353.28	7909.03	7074.68	4963.84
F152017	9.7	683507.92	31857.92	253.34	108874.5
F162017	155.94	942.61	417.36	361.28	187.12
F172017	758.96	636251.08	34158.07	5832.51	106116.14
F182017	34.3	31142.3	10561.36	6642.87	9330.68
F192017	37.95	280.18	93.32	64.97	67.1
F202017	100.32	398.01	279.72	306.6	84.27
F212017	157.32	162.98	159.23	159.3	1.68
F222017	200.52	628.07	331.79	278.02	131.74
F232017	200.17	643.64	261.94	223.79	101.48
F242017	422.07	465.05	432.14	430.27	8.95
F252017	782.07	850.27	833.27	836.11	15.21
F262017	495.1	539.26	518.99	518.64	9.24
F272017	17.02	460.2	71.6	42.13	96.37
F282017	3467.62	464107.48	23307.02	10181.16	65276.58
F292017	5177.96	31835.97	15700.28	15477.7	5385.81

Table 8 Basic statistics for 1st set of parameters

1st					
Function	Min	Max	Mean	Median	Standard Deviation
F12017	103.29	2731098.41	67773.87	3368.95	385813.31
F22017	0.35	12.4	3.35	2.56	2.62
F32017	0.37	1.47	1.09	1.14	0.29
F42017	0.83	2.87	1.35	1.3	0.35
F52017	0	0	0	0	0
F62017	2.12	53.03	8.1	4.46	10.49
F72017	0.85	1.93	1.46	1.49	0.29
F82017	0	0.02	0	0	0
F92017	19.55	36.95	28.36	28.52	3.63
F102017	5.95	124.52	42.33	38.63	27.83
F112017	57.09	1730.86	439.32	288.32	408.15
F122017	1.03	112.16	28.76	17.59	26.46
F132017	58.26	6792.97	1296.57	672.19	1647.54
F142017	3.52	156.6	70	69.71	28.78
F152017	0.14	2446.27	228.09	15	524.03
F162017	1.39	12.32	5.72	5.31	2.6
F172017	22.67	11184.1	1259.26	116.96	2392.7
F182017	2.42	317.61	76.6	59.68	66.65
F192017	0.52	3.36	1.1	0.85	0.68
F202017	1.15	4.73	2.88	3.11	0.78
F212017	1.56	1.64	1.58	1.57	0.01
F222017	2.06	7.51	3.89	3.65	1.43
F232017	2.02	10.48	3.1	2.51	1.51
F242017	4.28	4.77	4.42	4.39	0.11
F252017	7.5	8.7	8.25	8.31	0.29
F262017	5.05	5.57	5.37	5.38	0.12
F272017	0.16	0.81	0.43	0.39	0.15
F282017	59.93	261.25	127.04	117.29	47.92
F292017	77.27	520.74	231.59	218.73	107.83

Table 9 Basic statistics for 2nd set of parameters

2nd					
Function	Min	Max	Mean	Median	Standard Deviation
F12017	1365.18	2542995.09	204026.15	50407.7	480360.94
F22017	3.42	1101.49	216.83	165.99	213.95
F32017	35.03	168.38	96.88	92.33	30.53
F42017	56.65	252.98	119.06	111.81	38.32
F52017	0	0	0	0	0
F62017	102.31	1010.88	293.91	242.51	173.59
F72017	80.08	197.28	138.87	139.03	25.18
F82017	0	1.29	0.3	0.18	0.32
F92017	1525.93	3299.6	2799.85	2839.13	371.61
F102017	112.34	18734.51	4733.76	3647.7	3775.19
F112017	3378.1	153065.15	33076.26	24717.65	30698.91
F122017	160.41	12242.19	3265.45	2800.36	2953.15
F132017	1473.02	477488.36	90999.63	48238.47	113085.38
F142017	1024.15	18047.2	7849.12	8368.81	4775.19
F152017	9.04	487745.95	31214.26	1138.99	83068.13
F162017	120.25	1108.09	486.06	416.7	253.98
F172017	1521.53	1602429.65	91378.69	6151.6	287815.61
F182017	27.84	26613.41	8145.82	6976.73	5620.57
F192017	31.42	295.09	97.46	70.41	67.35
F202017	104.78	414.79	275.52	313.66	85.95
F212017	155.58	161.28	157.94	157.52	1.18
F222017	202.36	597.98	324.85	263.92	121.1
F232017	200.08	645.05	241.28	209.2	93.46
F242017	422.85	466.1	436.23	433.62	10.46
F252017	772.23	864.51	824.87	837.17	24
F262017	502.94	552.98	523.23	521.87	9.06
F272017	18.17	447.13	51.55	37.38	61.31
F282017	3893.48	29897.83	10316.13	9531.14	4813.1
F292017	5713.54	320086.82	23978.04	16370.1	43782.58

Table 10 Basic statistics for 3rd set of parameters

3rd					
Function	Min	Max	Mean	Median	Standard Deviation
F12017	7	133860.41	16221	4258.04	26537.81
F22017	0.25	410.15	44.07	21.04	69
F32017	32.04	137.37	83.49	83.88	28.09
F42017	59.13	155.16	98.84	100.12	19.21
F52017	0	0	0	0	0
F62017	114.46	380.82	176.35	159.37	58.49
F72017	72.58	204.69	132.38	133.14	27.55
F82017	0	0.55	0.11	0.09	0.15
F92017	2076.99	3787.6	2959.45	2976.11	392.67
F102017	184.51	9588.56	2861.82	2033.8	2387.7
F112017	4027.36	136382.62	41522.43	29156.97	34188.91
F122017	70.03	12384.22	2423.63	1545.18	2372.89
F132017	1761.12	423930.21	86883.37	48035.4	108399.84
F142017	893.28	12678.77	6432.89	6267.17	3300.36
F152017	8.46	99913.03	6865.8	160.16	17615
F162017	114.41	1040.33	449.72	408.28	239.4
F172017	844.88	395911.76	30261.3	4175.95	78529.03
F182017	122.85	38502.59	7956.69	4268.5	8618.57
F192017	42.73	295.38	94.41	74.34	58.17
F202017	109.23	354.67	267.35	299.96	77.44
F212017	155.5	163.42	158.54	157.92	1.61
F222017	200.03	578.79	289.08	212.43	125.36
F232017	200.01	496.38	214.03	202.32	46.43
F242017	421.82	454.06	428.45	426.54	6.73
F252017	788.83	846.79	830.4	833.74	12.39
F262017	505.41	532.7	516.87	517.48	6.31
F272017	10.54	84.46	37.12	34.71	16.28
F282017	4334.73	179144.35	13218.36	9485.17	24419.65
F292017	4991.91	42371.1	19328.84	16612.96	9030.39

In the following graph, the outcome of a Friedman mean ranks test conducted on the dataset derived from multiple parameter settings is presented. This statistical test is utilized to assess the differences in performance across various configurations under non-parametric conditions. The results are depicted in a table graph format, where the Nemenyi critical distance is clearly marked. This visualization aids in identifying statistically significant differences between the ranks of the parameter settings, providing insights into which configurations yield the best performance relative to others. The application of the Nemenyi post-hoc test delineates the parameter sets that significantly differs.

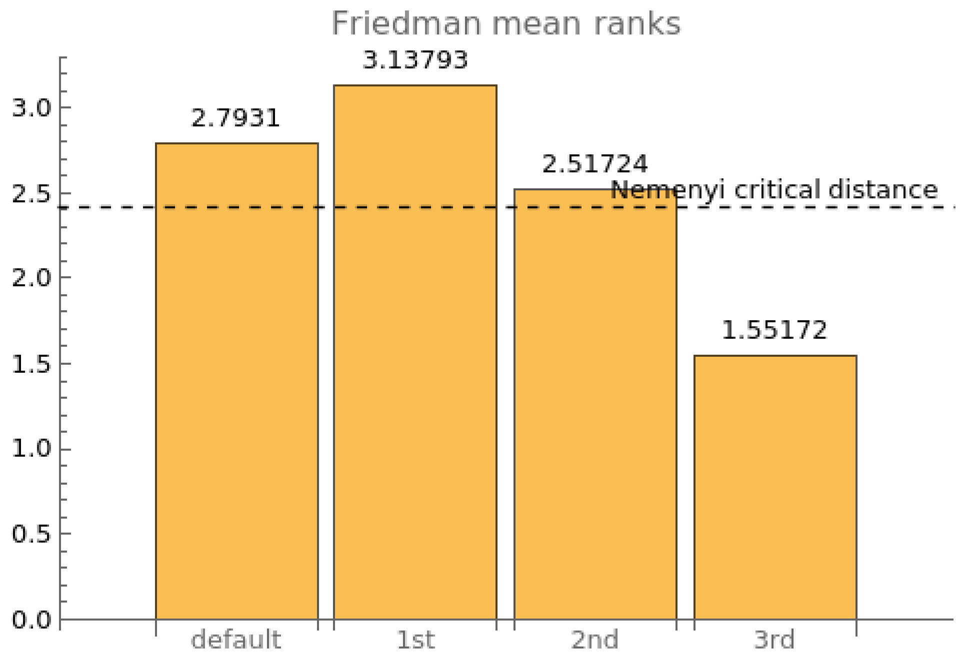


Figure 7 Friedman mean ranks for CEC 2017 test suite

In following table, we explore the pairwise comparisons of the first three parameter settings against the subsequent setting using the Wilcoxon Rank Sign Test. Each test aims to assess whether one parameter setting significantly outperforms the following setting in terms performance.

The second-to-last row of the table indicates the count of instances where a parameter setting outperforms the next. This provides a direct comparison of how often one setting yields better results than its successor. The final row displays the p-values obtained from the Wilcoxon tests, highlighting the statistical significance of each comparison. A p-value under 0.05 indicates a statistically significant difference in performance, suggesting that one setting is reliably better than the other under the tested conditions.

This table serves not only to identify which parameter settings perform better but also to quantify the significance of these differences, providing a robust framework for decision-making regarding parameter optimization.

Following Three Tables of Basic Statistics for CEC 2017 test functions (Minimum, Maximum, Mean, Median, and Standard Deviation) for Three Best Parameter Values Obtained from irace trained on CEC 2022 test functions.

Table 11 Wilcoxon rank sign test for CEC2017 test suite

Function	1st	2nd	3rd
F12017	+	-	-
F22017	+	-	-
F32017	+	-	-
F42017	+	-	-
F52017	-	+	+
F62017	+	-	-
F72017	+	-	-
F82017	+	-	-
F92017	-	-	+
F102017	+	+	-
F112017	-	-	+
F122017	-	+	-
F132017	+	-	-
F142017	-	+	-
F152017	-	+	-
F162017	+	-	-
F172017	+	-	-
F182017	-	+	-
F192017	+	-	-
F202017	+	-	-
F212017	-	-	+
F222017	+	-	-
F232017	+	-	-
F242017	+	-	-
F252017	-	-	+
F262017	+	-	-
F272017	-	+	-
F282017	-	-	+
F292017	+	+	-
count of (-)	11	21	23
pValues	0.34	0.08	0

Table 12 Basic statistics for CEC 2017 test suite using 1st set of parameters obtained racing CEC 2022 test suite

1st					
Function	Min	Max	Mean	Median	Standard Deviation
F12017	10328.86	273109840.89	6777387.28	336894.64	38581330.84
F22017	34.87	1240.4	334.84	255.68	261.62
F32017	37.23	147.34	108.86	114.23	29.17
F42017	82.5	286.93	134.79	130.13	35.05
F52017	0	0	0	0	0
F62017	211.57	5302.74	809.5	445.61	1048.63
F72017	84.52	192.81	146.28	149.01	28.55
F82017	0.01	1.63	0.4	0.29	0.34
F92017	1954.81	3695.1	2835.93	2851.51	363.3
F102017	595.03	12451.83	4233.01	3863.19	2782.59
F112017	5709.01	173085.59	43931.54	28831.52	40815.19
F122017	103.43	11215.88	2875.96	1758.68	2646.05
F132017	5825.6	679296.89	129657.19	67219.48	164754.39
F142017	351.83	15659.68	7000.34	6971.18	2877.59
F152017	14.06	244626.97	22809.5	1499.91	52402.87
F162017	139.03	1231.88	571.67	531.11	259.7
F172017	2267.31	1118410.21	125925.75	11696.27	239270.23
F182017	241.72	31761	7660.02	5968.47	6665.01
F192017	51.91	335.66	109.73	84.82	68.03
F202017	114.84	472.9	287.87	310.53	77.73
F212017	156.03	163.71	157.99	157.47	1.29
F222017	206.09	751.15	388.95	364.78	143.33
F232017	201.61	1048.3	310.05	250.58	150.75
F242017	427.88	477.01	442.02	439.26	10.99
F252017	749.84	869.98	825.44	831.18	28.8
F262017	505.02	557.36	536.52	537.75	11.9
F272017	16.19	81.24	42.71	38.86	14.88
F282017	5992.55	26124.84	12704.18	11729.16	4792.03
F292017	7726.56	52073.53	23158.54	21873	10783.4

Table 13 Basic statistics for CEC 2017 test suite using 2nd set of parameters obtained racing CEC 2022 test suite

2nd					
Function	Min	Max	Mean	Median	Standard Deviation
F12017	1365.18	2542995.09	204026.15	50407.7	480360.94
F22017	3.42	1101.49	216.83	165.99	213.95
F32017	35.03	168.38	96.88	92.33	30.53
F42017	56.65	252.98	119.06	111.81	38.32
F52017	0	0	0	0	0
F62017	102.31	1010.88	293.91	242.51	173.59
F72017	80.08	197.28	138.87	139.03	25.18
F82017	0	1.29	0.3	0.18	0.32
F92017	1525.93	3299.6	2799.85	2839.13	371.61
F102017	112.34	18734.51	4733.76	3647.7	3775.19
F112017	3378.1	153065.15	33076.26	24717.65	30698.91
F122017	160.41	12242.19	3265.45	2800.36	2953.15
F132017	1473.02	477488.36	90999.63	48238.47	113085.38
F142017	1024.15	18047.2	7849.12	8368.81	4775.19
F152017	9.04	487745.95	31214.26	1138.99	83068.13
F162017	120.25	1108.09	486.06	416.7	253.98
F172017	1521.53	1602429.65	91378.69	6151.6	287815.61
F182017	27.84	26613.41	8145.82	6976.73	5620.57
F192017	31.42	295.09	97.46	70.41	67.35
F202017	104.78	414.79	275.52	313.66	85.95
F212017	155.58	161.28	157.94	157.52	1.18
F222017	202.36	597.98	324.85	263.92	121.1
F232017	200.08	645.05	241.28	209.2	93.46
F242017	422.85	466.1	436.23	433.62	10.46
F252017	772.23	864.51	824.87	837.17	24
F262017	502.94	552.98	523.23	521.87	9.06
F272017	18.17	447.13	51.55	37.38	61.31
F282017	3893.48	29897.83	10316.13	9531.14	4813.1
F292017	5713.54	320086.82	23978.04	16370.1	43782.58

Table 14 Basic statistics for CEC 2017 test suite using 3rd set of parameters obtained racing CEC 2022 test suite

3rd					
Function	Min	Max	Mean	Median	Standard Deviation
F12017	7	133860.41	16221	4258.04	26537.81
F22017	0.25	410.15	44.07	21.04	69
F32017	32.04	137.37	83.49	83.88	28.09
F42017	59.13	155.16	98.84	100.12	19.21
F52017	0	0	0	0	0
F62017	114.46	380.82	176.35	159.37	58.49
F72017	72.58	204.69	132.38	133.14	27.55
F82017	0	0.55	0.11	0.09	0.15
F92017	2076.99	3787.6	2959.45	2976.11	392.67
F102017	184.51	9588.56	2861.82	2033.8	2387.7
F112017	4027.36	136382.62	41522.43	29156.97	34188.91
F122017	70.03	12384.22	2423.63	1545.18	2372.89
F132017	1761.12	423930.21	86883.37	48035.4	108399.84
F142017	893.28	12678.77	6432.89	6267.17	3300.36
F152017	8.46	99913.03	6865.8	160.16	17615
F162017	114.41	1040.33	449.72	408.28	239.4
F172017	844.88	395911.76	30261.3	4175.95	78529.03
F182017	122.85	38502.59	7956.69	4268.5	8618.57
F192017	42.73	295.38	94.41	74.34	58.17
F202017	109.23	354.67	267.35	299.96	77.44
F212017	155.5	163.42	158.54	157.92	1.61
F222017	200.03	578.79	289.08	212.43	125.36
F232017	200.01	496.38	214.03	202.32	46.43
F242017	421.82	454.06	428.45	426.54	6.73
F252017	788.83	846.79	830.4	833.74	12.39
F262017	505.41	532.7	516.87	517.48	6.31
F272017	10.54	84.46	37.12	34.71	16.28
F282017	4334.73	179144.35	13218.36	9485.17	24419.65
F292017	4991.91	42371.1	19328.84	16612.96	9030.39

Friedman mean ranks for values obtained from irace raced on CEC 2022 test functions:

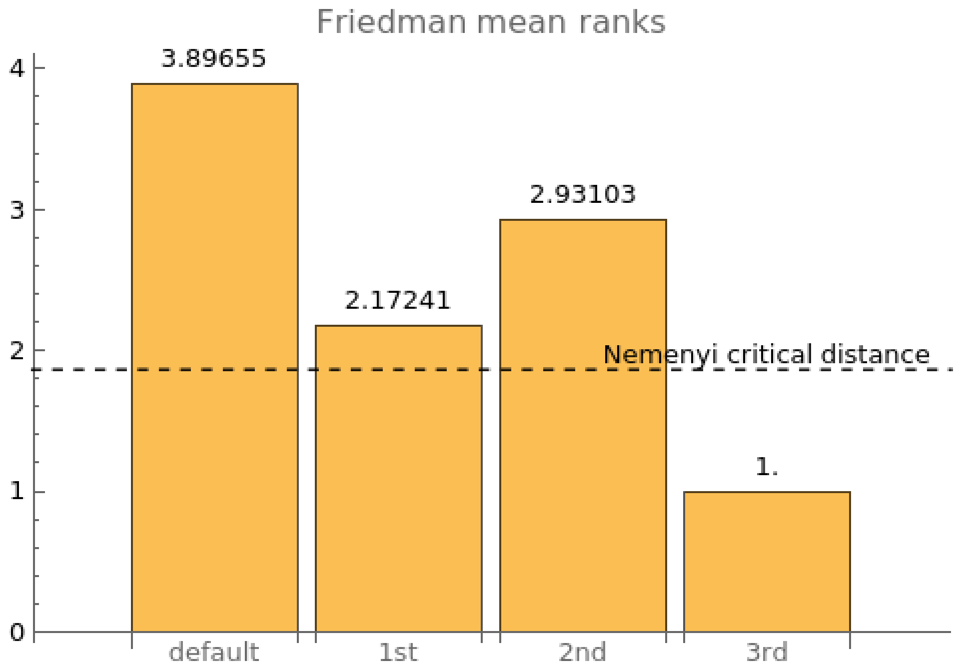


Figure 8 Friedman mean ranks for CEC 2017 test suite using parameters obtained racing CEC 2022 test suite

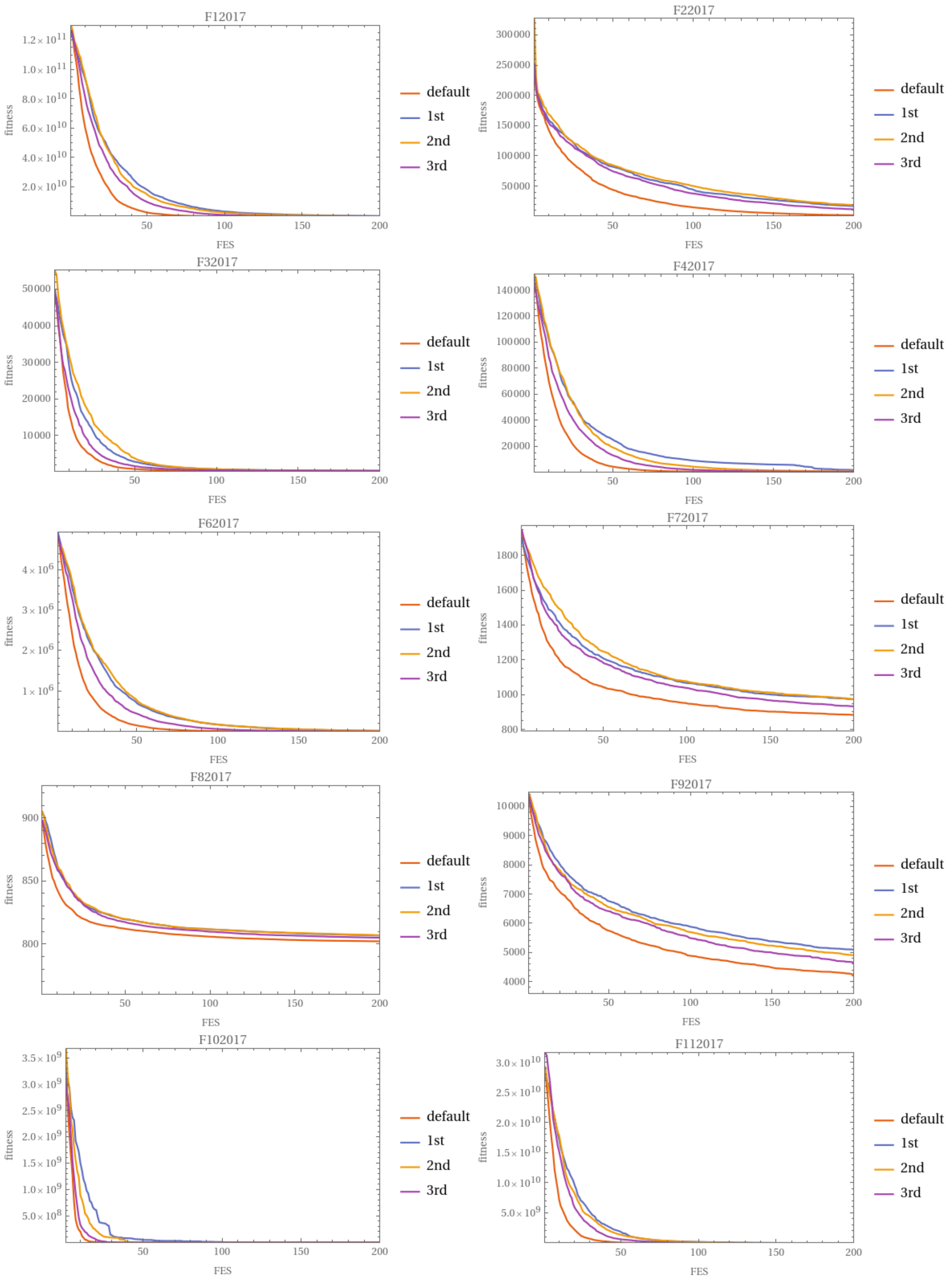
Pairwise comparisons of parameter settings obtained from CEC 2022 test suite against the subsequent setting using the Wilcoxon Rank Sign:

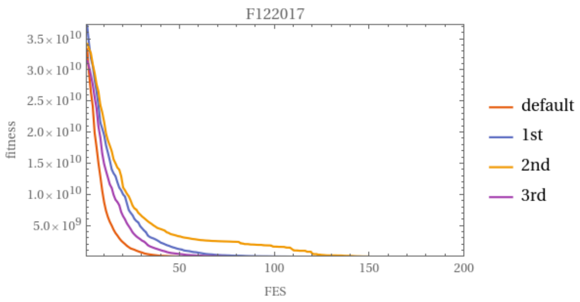
Table 15 Wilcoxon rank sign test for CEC 2017 test suite with parameters obtained racing CEC 2022 test suite

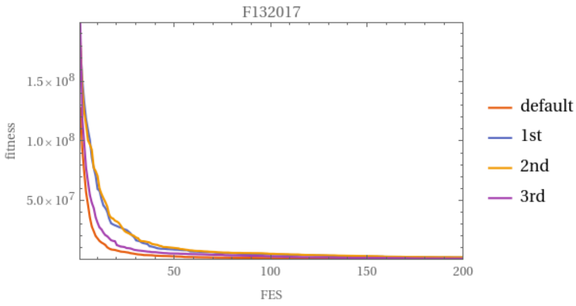
Function	1st	2nd	3rd
F12017	-	+	-
F22017	-	+	-
F32017	-	+	-
F42017	-	+	-
F52017	-	+	-
F62017	-	+	-
F72017	-	+	-
F82017	-	+	-
F92017	-	-	-
F102017	+	-	-
F112017	-	+	-
F122017	-	+	-
F132017	-	-	-
F142017	-	+	-
F152017	-	+	-
F162017	-	+	-
F172017	-	+	-
F182017	-	+	-
F192017	-	+	-
F202017	-	+	-
F212017	-	+	-
F222017	-	-	-
F232017	-	+	-
F242017	-	+	-
F252017	-	+	-
F262017	-	+	-
F272017	-	+	-
F282017	-	+	-
F292017	-	+	-
count of (-)	28	4	29
pValues	0	0	0

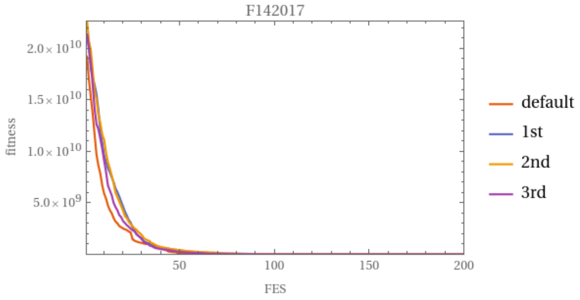
Graphs of average convergence for each test function with 4 different parameters setting, each convergence was calculated 50 times, for better observability only first 200 FEs are shown.

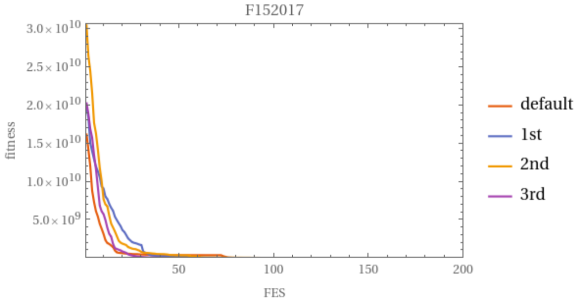
Table 16 Graphs of average convergencies

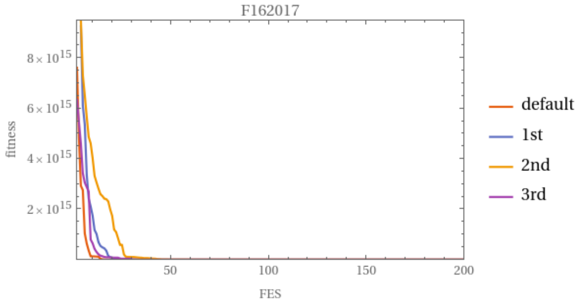


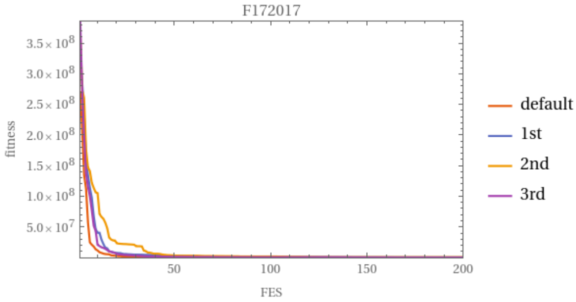

F122017
fitness
FES
default
1st
2nd
3rd

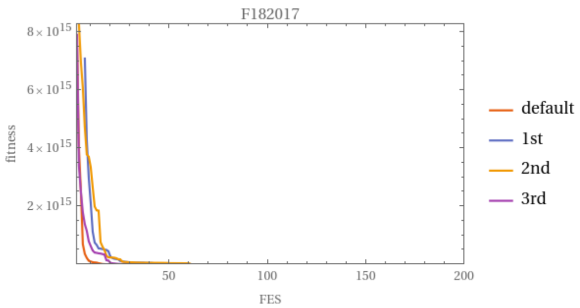

F132017
fitness
FES
default
1st
2nd
3rd

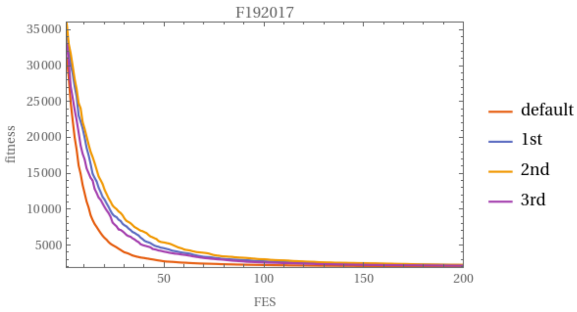

F142017
fitness
FES
default
1st
2nd
3rd

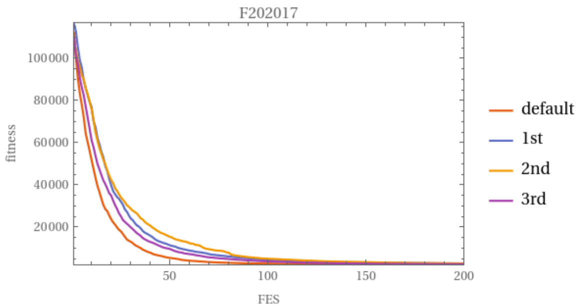

F152017
fitness
FES
default
1st
2nd
3rd

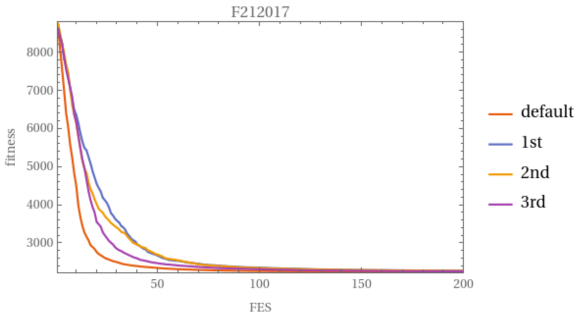

F162017
fitness
FES
default
1st
2nd
3rd

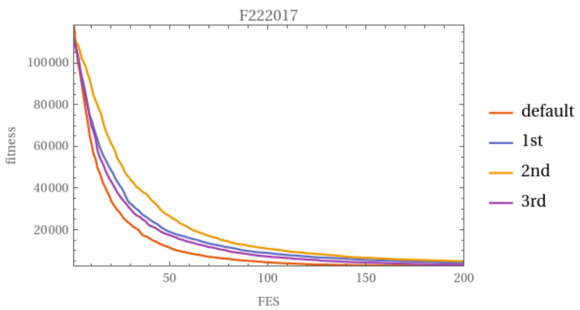

F172017
fitness
FES
default
1st
2nd
3rd

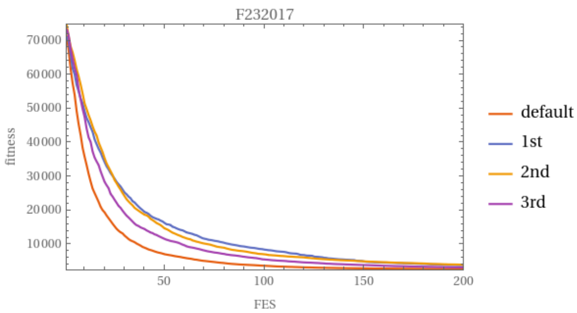

F182017
fitness
FES
default
1st
2nd
3rd

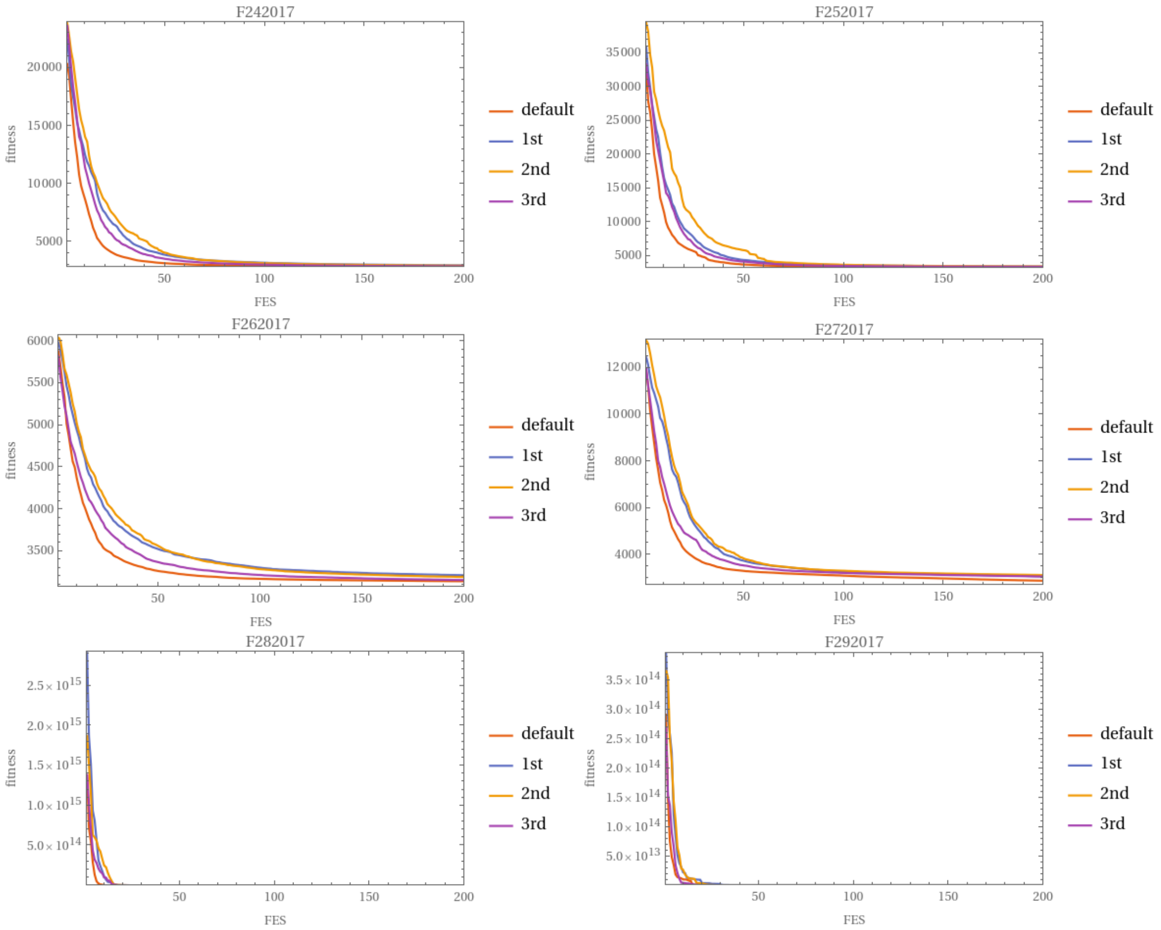

F192017
fitness
FES
default
1st
2nd
3rd


F202017
fitness
FES
default
1st
2nd
3rd


F212017
fitness
FES
default
1st
2nd
3rd


F222017
fitness
FES
default
1st
2nd
3rd


F232017
fitness
FES
default
1st
2nd
3rd



F242017
F252017
F262017
F272017
F282017
F292017
fitness
FES
default
1st
2nd
3rd

6.2 CEC 2022 test suite:

Four Tables of Basic Statistics (Minimum, Maximum, Mean, Median, and Standard Deviation) for Officially Recommended Parameter Settings, and the Three Best Parameter Values Obtained from irace, each function was runed 50 times:

Table 17 Basic statistics for default parameters

default					
Function	Min	Max	Mean	Median	Standard Deviation
F1	7963.47	30375.84	16958.37	16132.77	5279.75
F2	86.21	240.51	142.76	137.71	34.81
F3	1.63	18.02	7.11	6.49	3.66
F4	71.6	161.19	104.95	101.59	17.31
F5	79.31	1866.23	680.69	594.2	442.78
F6	160.12	62013407.17	2001022.09	13305.52	9010987.12
F7	28.46	128.85	65.83	58.21	26.45
F8	25.97	93.62	34.51	31.06	11.18
F9	190.37	233.54	206.98	204.5	10.43
F10	82.08	892.29	286.46	282.88	196.62
F11	477.74	1664.36	1006.75	1017.31	262.56
F12	277.09	421.13	335.59	332.54	36.68

Table 18 Basic statistics for 1st set of parameters

1st					
Function	Min	Max	Mean	Median	Standard Deviation
F1	8117.88	33627.54	17062.19	16970.42	5750.27
F2	71.47	264.77	140.67	135.97	43.81
F3	1.35	16.03	4.48	3.86	2.64
F4	52.16	157.73	97.76	98.68	21.48
F5	36.1	1007.43	390.52	354.02	245.89
F6	344.23	1402433.89	91358.12	7585.3	242919.14
F7	29.2	162.2	58.46	52.45	24.7
F8	24.5	64.78	30.24	28.77	6.1
F9	186.68	217.68	195.5	193.82	7.29
F10	47.29	628.82	213.64	176.62	140.19
F11	456.48	1416.95	816.59	749.76	230.66
F12	270.67	355.82	306.69	302.44	21.78

Table 19 Basic statistics for 2nd set of parameters

2nd					
Function	Min	Max	Mean	Median	Standard Deviation
F1	4503.54	35581.29	17477.52	16004.86	6507.59
F2	75.87	298.63	141.71	132.32	45.5
F3	1.46	20.39	5.67	4.39	4.33
F4	56.65	141.69	94.08	94.64	20.22
F5	132.88	3050.41	655.35	559.96	497.8
F6	160.64	20849607.24	823004.05	27302.06	3386439.81
F7	25.76	177.26	67.89	57.71	32.85
F8	24.46	105.52	33.15	28.95	13.02
F9	186.79	235.28	203.4	201.93	11.62
F10	29.89	785.27	247.2	169.15	175.86
F11	442.26	1502.15	900.71	856.4	225.45
F12	264.55	434.33	318.71	312.04	32.29

Table 20 Basic statistics for 3rd set of parameters

3rd					
Function	Min	Max	Mean	Median	Standard Deviation
F1	5524.95	32533.41	15111.86	13538.46	5839.02
F2	56.93	230.34	101.36	94.11	33.81
F3	0.39	5.31	1.71	1.19	1.27
F4	24.15	112.47	72.45	70.49	20.87
F5	19.31	551.67	170.43	131.91	119.89
F6	186.14	843818.89	34972.42	3651.97	125470.77
F7	25.38	95.9	48.9	48.28	13.85
F8	24.82	36.06	29.07	29	2.58
F9	181.81	205.59	188.04	186.84	4.53
F10	51.57	494.85	171.35	101.46	104.15
F11	270.71	1058.03	599	582.87	140.98
F12	253.74	322.82	280.27	278.49	14.85

In the following graph, the outcome of a Friedman mean ranks test conducted on the dataset derived from multiple parameter settings is presented.

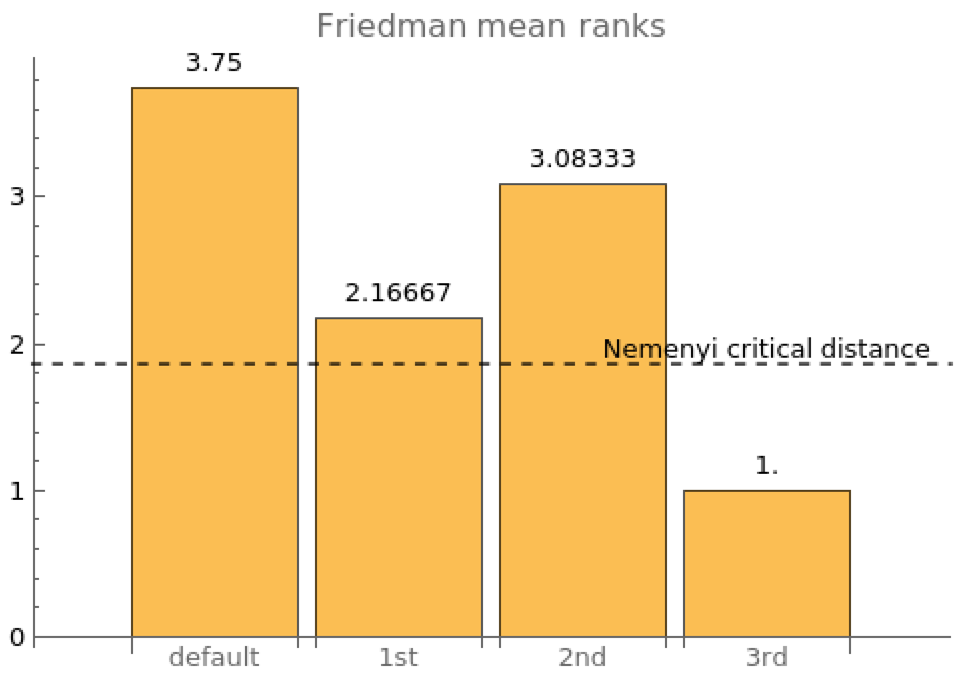


Figure 9 Friedman mean ranks for 2022 CEC test suite

Table of pairwise comparisons of parameter settings:

Table 21 Wilcoxon rank sign test for CEC 2022 test suite

Function	1st	2nd	3rd
F1	+	+	-
F2	-	+	-
F3	-	+	-
F4	-	-	-
F5	-	+	-
F6	-	+	-
F7	-	+	-
F8	-	+	-
F9	-	+	-
F10	-	+	-
F11	-	+	-
F12	-	+	-
count of (-)	11	1	12
pValues	0.02	0.01	0

Three Tables of Basic Statistics for CEC 2022 test functions (Minimum, Maximum, Mean, Median, and Standard Deviation) for Three Best Parameter Values Obtained from irace trained on CEC 2017 test functions:

Table 22 Basic statistics for CEC 2022 test suite using 1st set of parametrs obtained racing CEC 2017 test suite

1st					
Function	Min	Max	Mean	Median	Standard Deviation
F1	6145.24	32739.56	16418.48	16250.8	5473.95
F2	87.73	269.91	168.86	170.79	47.43
F3	2.45	22.56	11.14	10.05	5
F4	89.8	163.12	121.93	121.06	15.52
F5	127.68	2404.68	729.38	652.03	426.05
F6	423.54	20349890.09	953712.88	12972.63	3854514.52
F7	43.48	192.92	83.73	81.98	25.25
F8	28.2	162.44	42.02	34.34	24.91
F9	189.67	262.91	215.37	211.17	16.47
F10	101.46	1301.54	351.78	285.09	292.81
F11	548.69	1561.52	880.37	814.87	222.28
F12	317.22	567.19	393.4	385.52	51.63

Table 23 Basic statistics for CEC 2022 test suite using 2nd set of parametrs obtained racing CEC 2017 test suite

2nd					
Function	Min	Max	Mean	Median	Standard Deviation
F1	4244.83	32372.54	16794.52	16920.48	6022.71
F2	81.26	258.86	153.06	147.48	42.11
F3	1.89	22.38	8.51	7.6	4.97
F4	81.24	155.85	116.24	115.5	17.58
F5	81.6	1646.08	680.9	639.86	366.13
F6	417.81	6620993.6	235061.3	4952.96	959968.7
F7	46.13	181.25	84.35	76.03	34.06
F8	26.55	174.2	38.2	31.59	27.15
F9	193.27	276.09	211.96	208.87	15.26
F10	63.91	669.1	270.27	230.03	159.14
F11	517.59	1676.61	929.46	886	253.32
F12	291.18	448.69	357.78	352.92	38.58

Table 24 Basic statistics for CEC 2022 test suite using 3rd set of parametrs obtained racing CEC 2017 test suite

3rd					
Function	Min	Max	Mean	Median	Standard Deviation
F1	6686	32329.52	17207.77	17215.07	5687.93
F2	78.93	245.71	141.38	140.69	43.53
F3	1.45	10.57	4.87	4.21	2.38
F4	61.8	143.25	111.79	112.27	17.64
F5	70.86	1360.92	483.23	359.38	319.87
F6	170.73	13332962.74	642089.19	10914.37	2476951.4
F7	30.7	123.52	61.99	61.26	18.83
F8	26.19	51.6	32.51	31.31	5.33
F9	190.39	230.92	202.27	200.55	8.59
F10	27.2	449.85	211.08	181.56	115.84
F11	496.6	1513.49	890.16	901.22	206.51
F12	260.69	442.3	321.78	317.96	31.78

Friedman mean ranks for values obtained from irace raced on CEC 2017 test functions:

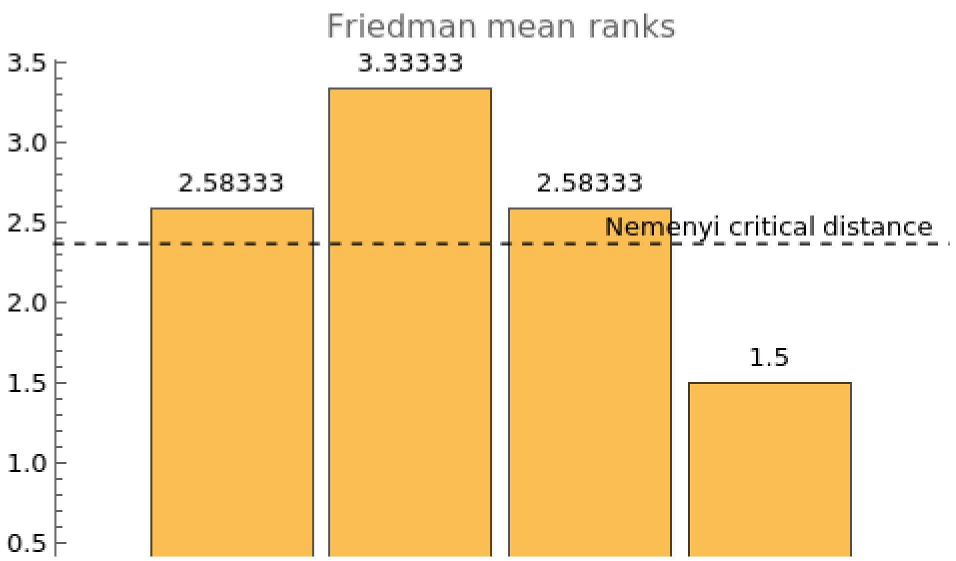


Figure 10 Friedman mean ranks for CEC 2022 test suite using parameters obtained racing CEC 2017 test suite

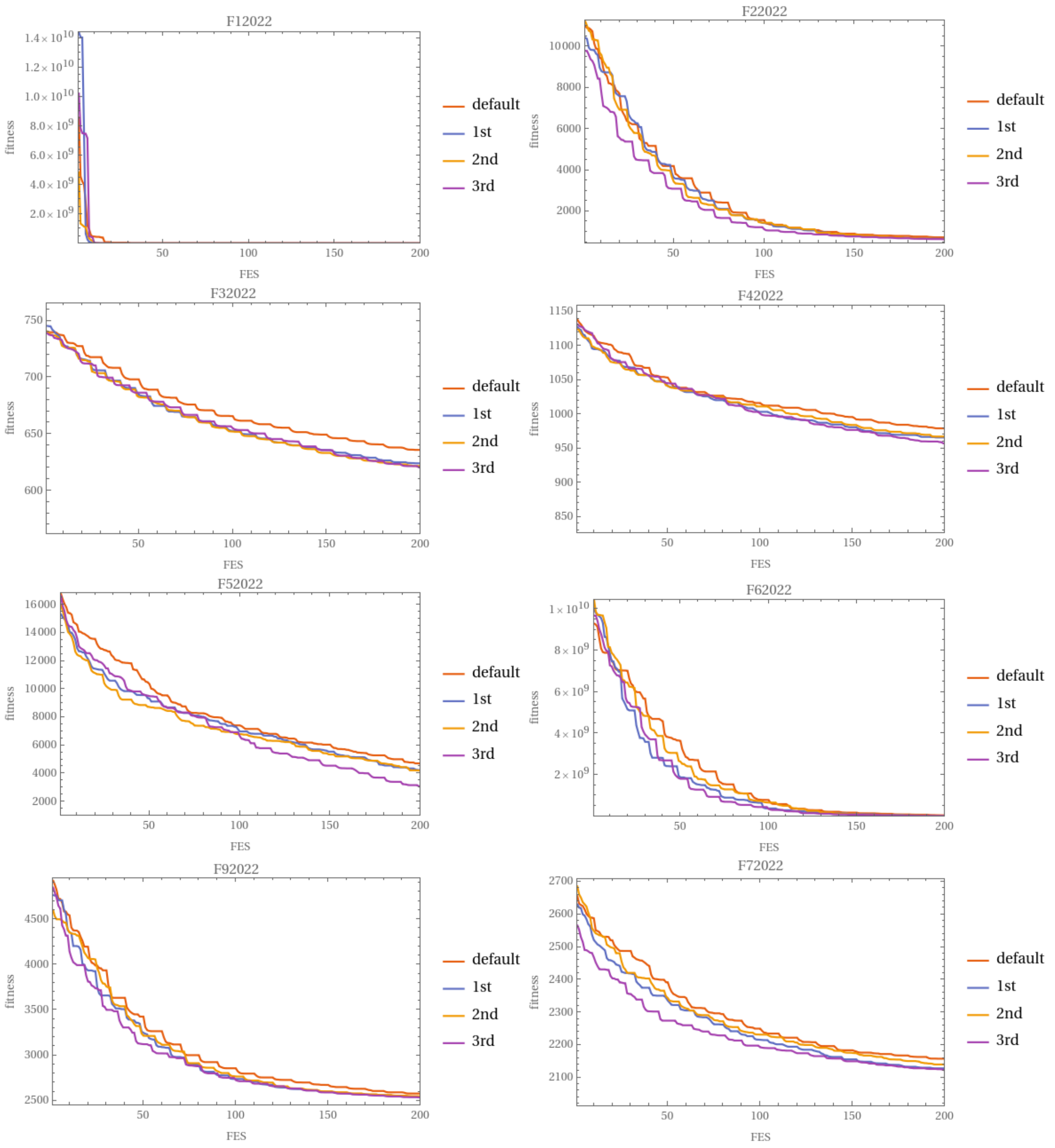
pairwise comparisons of parameter settings obtained from CEC 2017 test suite against the subsequent setting using the Wilcoxon Rank Sign:

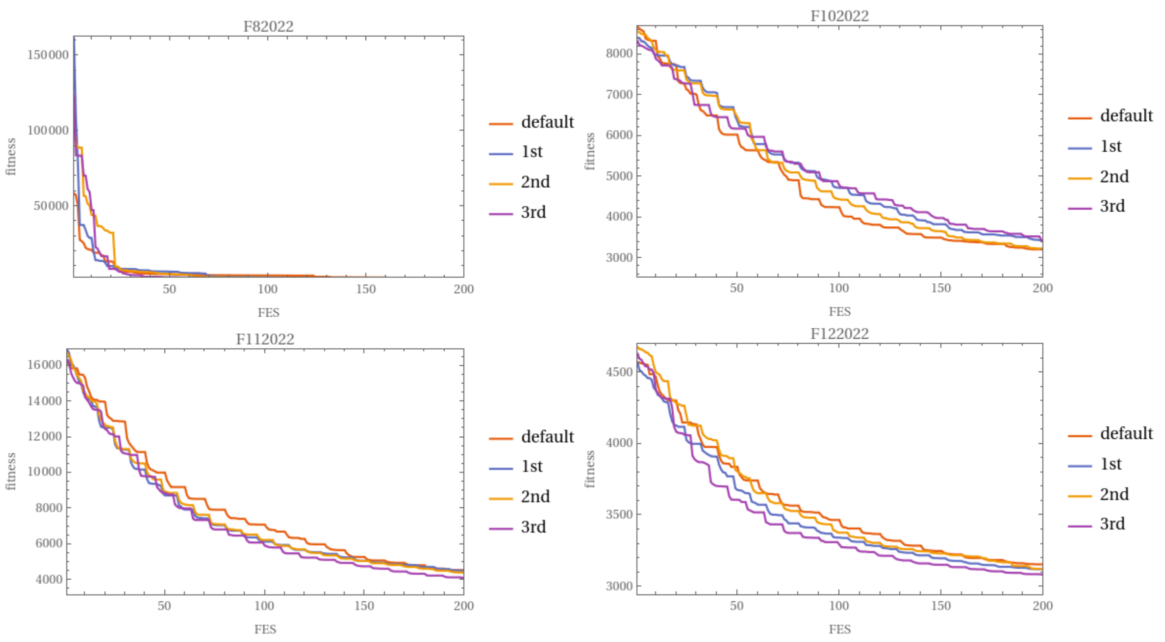
Table 25 Wilcoxon rank sign test for CEC 2022 suite with parameters obtained racing CEC 2017 test suite

Function	1st	2nd	3rd
F1	-	+	+
F2	+	-	-
F3	+	-	-
F4	+	-	-
F5	-	-	-
F6	+	-	+
F7	+	+	-
F8	+	-	-
F9	+	-	-
F10	+	-	-
F11	-	+	-
F12	+	-	-
count of (-)	3	9	10
pValues	0.29	0.17	0.22

Graphs of average convergence for each test function with 4 different parameters setting, each convergence was calculated 50 times, for better observability only first 200 FEs are following.

Table 26 Graphs of average convergencies





6.3 Summary

This chapter provides a robust statistical analysis of the SOMA T3A function optimization using irace. The results from the irace framework, combined with rigorous statistical testing, offer valuable insights into optimal parameter settings. The use of both Wilcoxon Rank Sign Tests and Friedman tests with Nemenyi post-hoc analysis significantly enhances the credibility of the comparisons made between different parameter settings. The detailed statistical data, along with the interpretation of the mean ranks and critical distances, provide a clear guide on the performance landscape of the parameter configurations.

Based on the analysis, it is advisable to adopt the parameter values identified through the irace framework to optimize the SOMA T3A function effectively. These parameter settings have demonstrated performance in the tests conducted, underscoring their potential utility.

Best performing sets of parameters were 3rd for both cases, these parameters also statistically proven better performance when they were “cross” applied to different test suites than they were raced. This imply algorithm performance robustness across multiple different problem evaluation.

In conclusion, adopting the irace-optimized parameters for the SOMA T3A function is strongly supported by the data

7 SOMA PARETO

Tables of default parameters obtained from [5] and tuned from irace package:

Table 27 Parameters obtained for CEC 2017 test suite

parameter	T1	T2	nJumps
default	1	1	10
1st	1.24	0.48	10
2nd	0.98	1.18	8
3rd	1.19	0.5	12

Table 28 Parameters obtained for CEC 2022 test suite

parameter	T1	T2	nJumps
default	1	1	10
1st	0.82	1.35	5
2nd	0.92	1.88	7
3rd	0.96	1.33	5

7.1 CEC 2017 test suite

Four Tables of Basic Statistics (Minimum, Maximum, Mean, Median, and Standard Deviation) cost function error for Officially Recommended Parameter Settings, and the Three Best Parameter Values Obtained from irace, each function was runed 50 times:

Table 29 Basic statistics for default parameters

default					
Function	Min	Max	Mean	Median	Standard Deviation
F12017	0	0	0	0	0
F22017	0	0	0	0	0
F32017	0	3.99	0.46	0	1.22
F42017	0.99	10.94	5.65	5.97	2.3
F52017	0	0	0	0	0
F62017	7.96	22.3	16.18	15.89	2.88
F72017	4	16	12.32	12	2.53
F82017	0	0	0	0	0
F92017	3.6	365.69	152.4	135.2	126.3
F102017	0.99	16.12	4.87	3.98	3.55
F112017	1	839.34	251.74	215.3	227.54
F122017	2.1	52.28	10.81	7.09	10.13
F132017	1.99	26.78	19.37	21.13	6.64
F142017	0.1	6.59	1.91	1.49	1.31
F152017	0.02	2.36	0.4	0.24	0.45
F162017	1.5	26.18	21.49	23.16	5.99
F172017	0	1.99	0.38	0.35	0.41
F182017	0.13	2.47	0.99	1.05	0.6
F192017	0.57	5.13	1.98	1.64	1.1
F202017	100	242.36	126.19	100	50.11
F212017	100	103.62	101.56	101.44	1.03
F222017	100	200	118	100	38.81
F232017	100	343.87	209.69	200	47.21
F242017	476.28	481.19	478.04	477.49	1.43
F252017	0	471.23	222.57	0	234.03
F262017	6.47	397.76	318.06	389.62	147.92
F272017	0	0	0	0	0
F282017	985.16	1365.86	1114.55	1090.05	110.04
F292017	485757593.64	9.33E+15	923798239242850	228491249109205	2.13E+15

Table 30 Basic statistics for 1st set of parameters

1st					
Function	Min	Max	Mean	Median	Standard Deviation
F12017	0	0	0	0	0
F22017	0	0	0	0	0
F32017	0	3.99	0.8	0	1.61
F42017	0.99	11.94	4.93	3.98	2.57
F52017	0	0	0	0	0
F62017	11.53	19.64	14.92	14.79	1.99
F72017	4	16	11.6	12	2.68
F82017	0	0	0	0	0
F92017	3.66	349.18	109.62	123.88	105
F102017	0	15.92	3.88	2.98	3.04
F112017	0.83	1231.93	190.89	124.71	224.34
F122017	3.98	17.69	7.35	6.33	2.9
F132017	1.99	27.11	18.32	21.19	7.24
F142017	0.12	10.24	2.23	1.49	1.97
F152017	0.02	1.07	0.36	0.31	0.25
F162017	1.08	25.95	19.65	22.5	7.36
F172017	0	20.11	1.07	0.31	3.92
F182017	0.13	2.79	1.3	1.29	0.66
F192017	0.95	6.63	2.85	2.59	1.26
F202017	100	219.27	122.44	100	45.37
F212017	100	104.09	101.66	101.59	0.99
F222017	100	200	108	100	27.4
F232017	200	326.95	207.36	200	29.46
F242017	476.34	507.41	479.88	479.59	5.88
F252017	0	467.46	231.55	228.23	233.93
F262017	2.67	397.96	314.35	389.52	155.16
F272017	0	0	0	0	0
F282017	989.01	1387.33	1127.81	1116.46	114.85
F292017	71800301553.23	7.01E+15	1.53E+15	327872457917857	2.21E+15

Table 31 Basic statistics for 2nd set of parameters

2nd					
Function	Min	Max	Mean	Median	Standard Deviation
F12017	0	0	0	0	0
F22017	0	0	0	0	0
F32017	0	3.99	1.04	0	1.77
F42017	0.99	11.94	5.39	4.97	1.93
F52017	0	0	0	0	0
F62017	11.2	22.34	15.93	16.12	2.6
F72017	4	16	11.76	12	2.41
F82017	0	0	0	0	0
F92017	0.31	346.15	101.65	35.15	102.07
F102017	0.99	10.6	4.05	3.98	2.22
F112017	2.26	702.6	232.26	212.69	191.14
F122017	4.47	26.37	7.71	6.91	3.59
F132017	1.99	26.28	18.32	21.4	7.09
F142017	0.24	8.46	1.72	1.45	1.42
F152017	0.02	0.85	0.22	0.12	0.22
F162017	1.67	25.24	18.53	21.05	6.22
F172017	0	1.5	0.49	0.45	0.44
F182017	0.05	2.02	0.69	0.58	0.54
F192017	0	2.79	1.06	0.95	0.71
F202017	100	240.26	120.72	100	44.93
F212017	100	103.44	101.75	101.55	0.83
F222017	100	200	126	100	44.31
F232017	100	200	196	200	19.79
F242017	476.2	507.39	480.31	478.28	7.13
F252017	0	467.46	268.77	457.65	231.06
F262017	4.98	397.49	361.45	390.52	103.45
F272017	0	0	0	0	0
F282017	265.76	1387.76	1108.69	1134.12	167.8
F292017	3936428542.99	3.22E+15	708547224678479	145454780068856	1.00E+15

Table 32 Basic statistics for 3rd set of parameters

3rd					
Function	Min	Max	Mean	Median	Standard Deviation
F12017	0	0	0	0	0
F22017	0	0	0	0	0
F32017	0	3.99	0.29	0	1
F42017	1.99	11.94	6.05	5.97	2.43
F52017	0	0	0	0	0
F62017	11.37	21.69	15.75	15.17	2.37
F72017	6	18	11.08	12	2.83
F82017	0	0	0	0	0
F92017	3.6	465.01	112.12	55.04	117.79
F102017	0	9.95	3.69	2.98	2.61
F112017	0.36	1122.87	252.62	204.74	261.56
F122017	4.09	17	7.86	7	3.09
F132017	0.05	91.43	20.26	22.06	12.88
F142017	0.03	6.02	1.85	1.49	1.55
F152017	0	0.75	0.3	0.25	0.23
F162017	1.22	29.18	19.99	23.08	7.36
F172017	0	1.49	0.34	0.34	0.29
F182017	0.54	2.4	1.32	1.31	0.45
F192017	0.86	7.51	3.22	3.15	1.37
F202017	0	218.22	118.17	100	46.67
F212017	100	103.89	101.55	101.39	1.12
F222017	100	200	116	100	37.03
F232017	100	333.92	207.76	200	42.04
F242017	475.86	482	478.11	477.64	1.53
F252017	0	471.23	203.31	0	231.71
F262017	2.24	397.49	314.3	389.52	154.85
F272017	0	0	0	0	0
F282017	991.78	1375.89	1134.24	1110.91	130.31
F292017	427818054.29	9.65E+15	2.28E+15	1.06E+15	2.75E+15

Following graph displays the results from a Friedman mean ranks test performed on data from various parameter settings. This non-parametric statistical test is designed to evaluate performance disparities across different configurations. The outcomes are visualized in a table graph, highlighting the Nemenyi critical distance to pinpoint statistically significant variations in the ranking of parameter settings.. The implementation of the Nemenyi post-hoc test further clarifies that no parameter sets differ significantly.

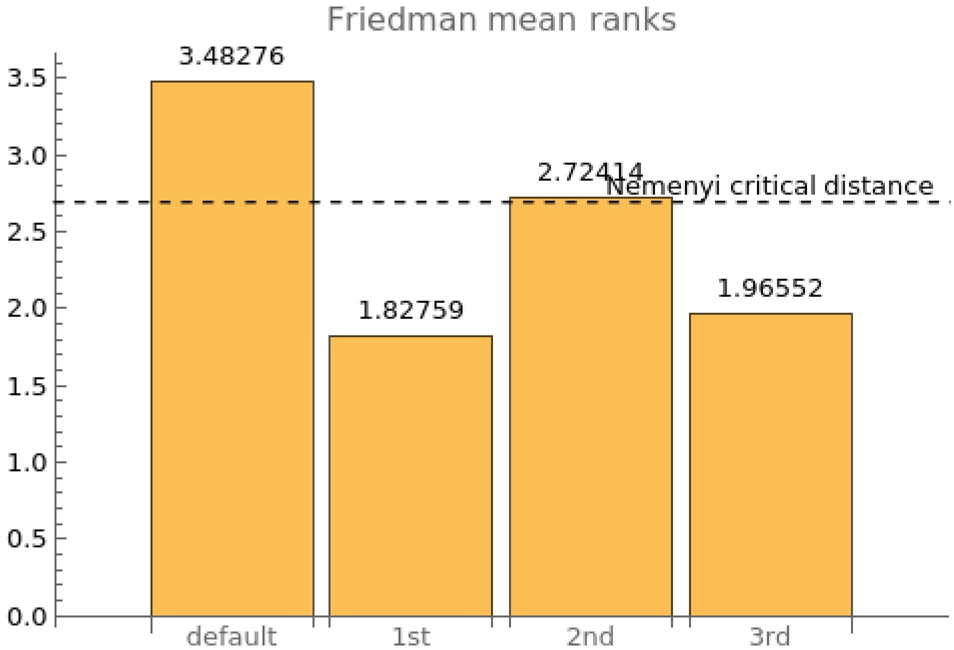


Figure 11 Friedman mean ranks for CEC 2017 test suite

In the presented table, we conduct pairwise comparisons between the initial three parameter settings and the following setting through the Wilcoxon Rank Sign Test. The purpose of each test is to determine if a particular parameter setting significantly surpasses its subsequent setting in terms of performance. The penultimate row of the table shows the number of times one parameter setting exceeds the performance of the previous, offering a clear metric of how frequently one setting outperforms another. The bottom row lists the p-values derived from these tests, emphasizing the statistical significance of each comparison. A p-value below 0.05 is indicative of a statistically significant performance difference.

Table 33 Wilcoxon rank sign test for CEC 2017 test suite

Function	1st	2nd	3rd
F12017	-	+	-
F22017	-	+	-
F32017	-	-	-
F42017	-	+	-
F52017	+	+	-
F62017	-	+	-
F72017	-	+	-
F82017	-	+	+
F92017	-	+	-
F102017	-	+	-
F112017	-	+	-
F122017	-	+	+
F132017	-	+	-
F142017	-	+	-
F152017	-	-	+
F162017	-	+	-
F172017	-	+	-
F182017	-	+	-
F192017	-	+	-
F202017	+	-	+
F212017	-	+	-
F222017	-	+	+
F232017	-	+	-
F242017	-	-	+
F252017	-	+	-
F262017	-	-	+
F272017	-	+	-
F282017	+	-	-
count of (-)	25	6	21
pValues	0	0.01	0.01

Table 34 Basic statistics for CEC 2017 test suite using 1st set of parameters obtained racing 2022 test suite

1st					
Function	Min	Max	Mean	Median	Standard Deviation
F12017	2.75	4458.8	758.67	375.89	1028.64
F22017	0	0	0	0	0
F32017	0.01	8.37	5.22	5.5	2.25
F42017	2	19.9	9.25	8.96	4.46
F52017	0	0	0	0	0
F62017	12.82	32.53	21.18	20.26	4.56
F72017	12	55.5	21.73	18	9.43
F82017	0	0	0	0	0
F92017	10.24	1116.34	444.72	440.04	313.09
F102017	4.45	230.68	44.02	37.21	35.94
F112017	302.62	20854.99	5179.83	4076.27	4581.13
F122017	10.53	6113.84	868.13	210.61	1378.97
F132017	29.56	306.1	57.66	44.59	47.35
F142017	10.1	1444.21	148.53	71.2	240.2
F152017	0.16	348.5	30.03	11.29	58.94
F162017	31.27	284.97	65.79	44.28	62.7
F172017	9.5	24704.55	4358.84	1289.77	6669.88
F182017	1.25	12.74	3.88	3.64	1.68
F192017	10.85	38.3	27.77	28.56	6.11
F202017	100	229.29	109.35	100	31.99
F212017	100	104.81	102.32	102.39	1.07
F222017	100	200.27	128.04	100.02	45.36
F232017	100.04	324.02	195.91	200.01	30.06
F242017	475.88	510.47	486.76	480.51	12.32
F252017	0	476.7	284.64	471.25	233.36
F262017	31.91	407.53	389.69	396.72	51.69
F272017	0	19.75	0.67	0	3.37
F282017	1107.03	16664.99	3019.13	1774.83	2965.55

Table 35 Basic statistics for CEC 2017 test suite using 2nd set of parameters obtained racing 2022 test suite

2nd					
Function	Min	Max	Mean	Median	Standard Deviation
F12017	1.93	8827.98	1383.04	666.96	1860.66
F22017	0	0.02	0	0	0
F32017	0	8.55	5.13	5.78	2.54
F42017	4.98	27.87	11.76	10.46	4.94
F52017	0	0	0	0	0
F62017	12.94	39.43	23.61	23.18	6.23
F72017	12	48.62	25.62	22.91	8.8
F82017	0	0	0	0	0
F92017	12.51	1319.83	603.67	619.25	335.63
F102017	11.57	258.21	70.39	53.71	53.57
F112017	298.24	21621.88	5272.08	3476.43	4538.76
F122017	11.59	6374.11	1262.7	600.29	1611.23
F132017	26.86	2882.43	163.2	48.23	440.29
F142017	4.41	1682.1	322.91	126.35	412.54
F152017	0.52	104.73	19.48	8.38	23.37
F162017	27.78	472	75.46	49.98	79.12
F172017	35.95	32515.73	6207.58	2978.29	8313.66
F182017	1.49	39.2	4.81	3.98	5.23
F192017	27.42	51.29	36.89	36.83	5.38
F202017	100	100.91	100.12	100.04	0.19
F212017	100	104.18	102.42	102.57	0.84
F222017	100.01	200.89	136.14	100.15	48.52
F232017	102.01	382.69	206.72	200.05	49.62
F242017	476.7	508.84	485.16	481.06	10.18
F252017	0.02	478.65	342.84	470.99	208.74
F262017	46.55	402.33	363.48	396.87	101.42
F272017	0	19.68	3.09	0.01	6.32
F282017	1027.4	10035.29	2633.05	1699.92	2046.6

Table 36 Basic statistics for CEC 2017 test suite using 3rd set of parameters obtained racing 2022 test suite

3rd					
Function	Min	Max	Mean	Median	Standard Deviation
F12017	0.01	2836.33	577.47	245.38	775.27
F22017	0	0	0	0	0
F32017	0.01	8.24	3.48	3.7	2.39
F42017	3.98	17.91	9.52	8.95	3.39
F52017	0	0	0	0	0
F62017	10.94	31.48	21.38	20.5	4.99
F72017	12	46	22.88	22	8.46
F82017	0	0.45	0.01	0	0.07
F92017	7.18	724.82	332.51	354.83	193.87
F102017	6.31	360.4	53.02	34.92	59.18
F112017	609.16	15073.05	4181.51	3298.24	3668.39
F122017	20.22	8062.9	1381.43	603.01	1805.17
F132017	22.86	1596.56	79.92	38.24	221.85
F142017	9.37	1402.19	150.23	47.44	239.67
F152017	0.24	450.34	35.92	5.53	80.64
F162017	26.37	138.26	40.97	38.81	17.37
F172017	22.68	29811.4	4446.13	1580.62	7121.55
F182017	0.29	19.13	3.5	3.13	3.04
F192017	3.76	37.59	24.13	27.11	9.25
F202017	100	222.29	106.84	100	27.37
F212017	100	103.86	102.33	102.43	0.99
F222017	100	205.3	148.15	100.03	50.62
F232017	200	341.92	205.27	200	26.11
F242017	476.6	508.09	485.29	481.03	10.82
F252017	0	482.23	331.96	472.48	218.11
F262017	8.94	402.33	381.98	396.8	76.18
F272017	0	19.72	0.67	0	3.37
F282017	1055.66	9952.73	2423.8	1586.53	1793.44

In previous three tables basic statistics for CEC 2017 test functions (Minimum, Maximum, Mean, Median, and Standard Deviation) for Three Best Parameter Values Obtained from irace trained on CEC 2022 test functions were shown.

Friedman mean ranks for CEC 2017 test functions for parameters obtained from irace raced on CEC 2022 test functions:

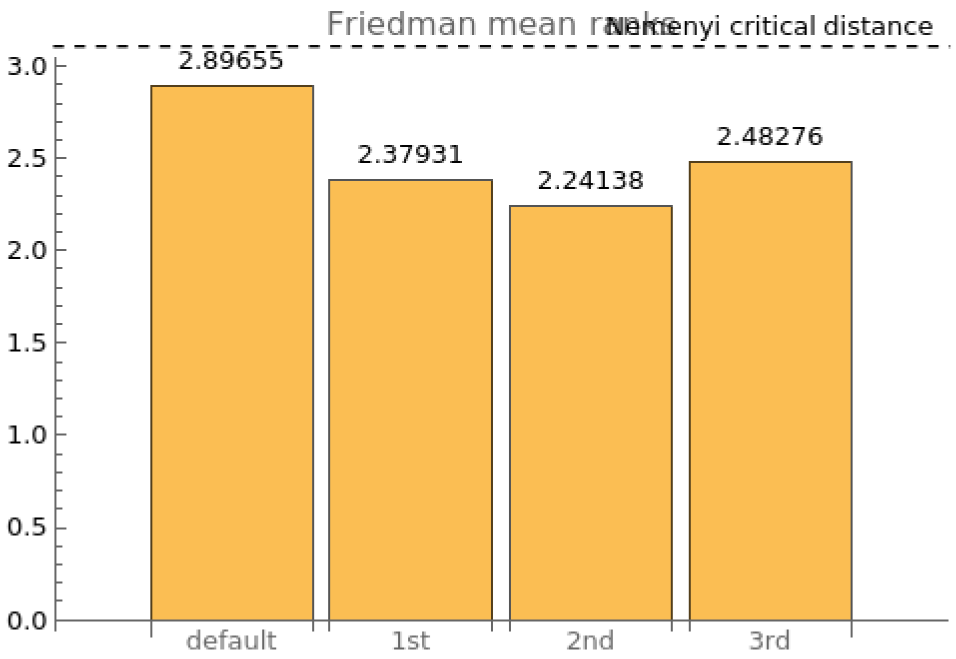


Figure 12 Friedman mean ranks for CEC 2017 test suite using parameters obtained racing 2022 test suite

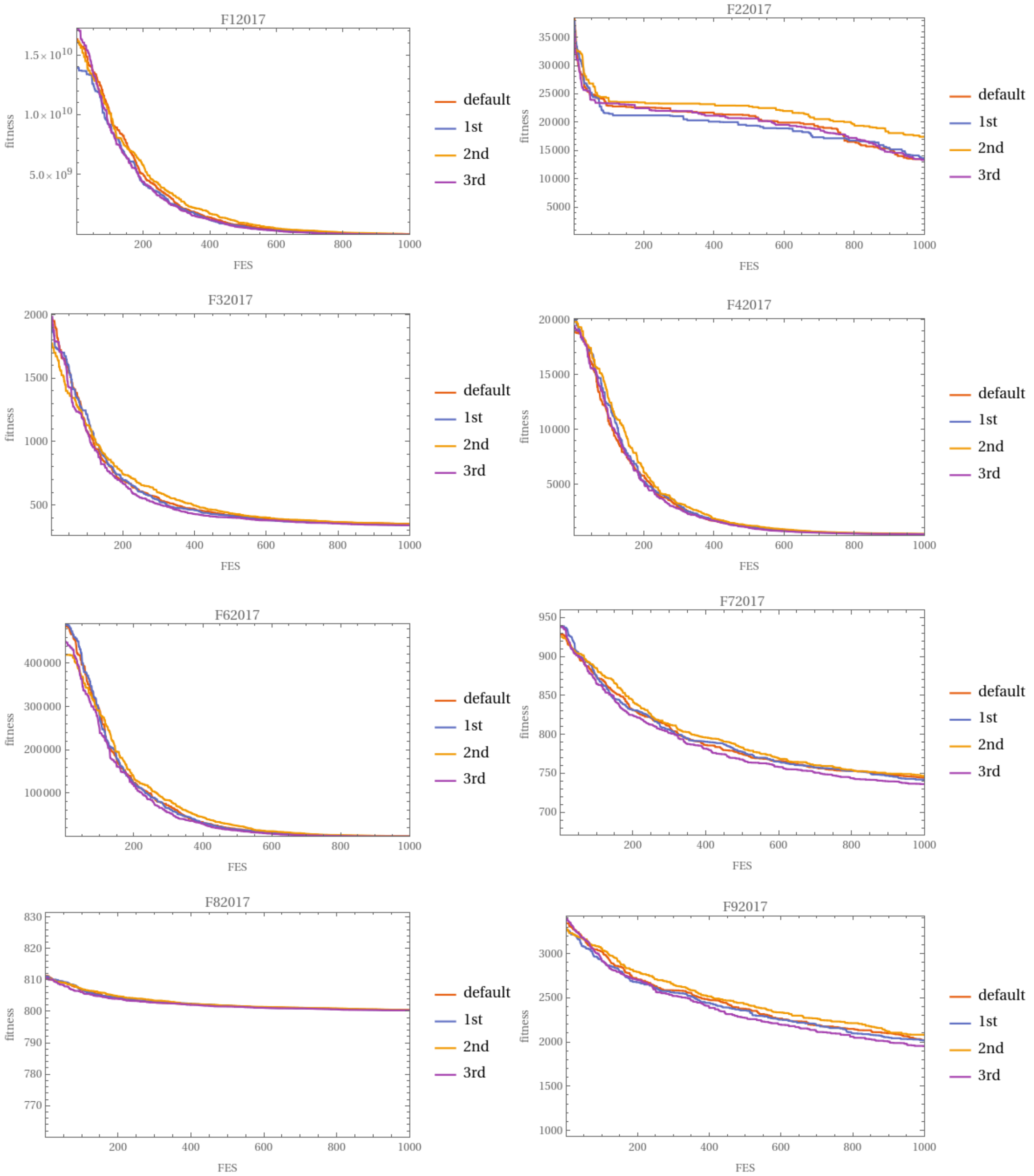
In following table pairwise comparisons of parameter settings obtained from CEC 2022 test suite against the subsequent setting using the Wilcoxon Rank Sign:

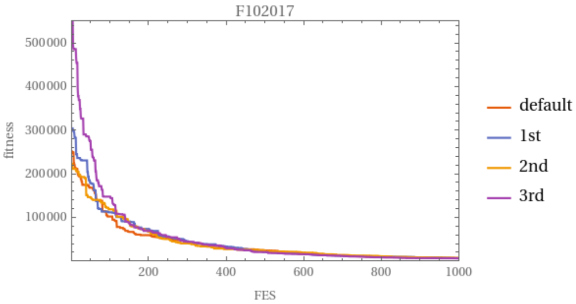
Table 37 Wilcoxon rank sign test for CEC 2017 test suite with parameters obtained racing CEC 2022 test suite

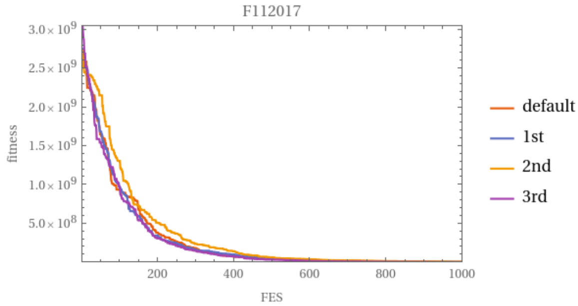
Function	1st	2nd	3rd
F12017	+	+	-
F22017	-	=	=
F32017	+	+	-
F42017	-	+	+
F52017	-	-	+
F62017	-	+	-
F72017	-	+	-
F82017	=	=	+
F92017	-	-	+
F102017	-	+	-
F112017	-	+	+
F122017	-	+	+
F132017	-	-	+
F142017	+	-	+
F152017	-	-	+
F162017	-	-	+
F172017	+	-	-
F182017	+	-	+
F192017	+	-	+
F202017	-	-	-
F212017	+	+	-
F222017	-	+	-
F232017	-	-	+
F242017	+	+	-
F252017	+	+	-
F262017	-	+	-
F272017	+	-	+
F282017	+	-	+
F292017	+	-	+
count of (-)	16	14	12
pValues	0.3	0.78	0.64

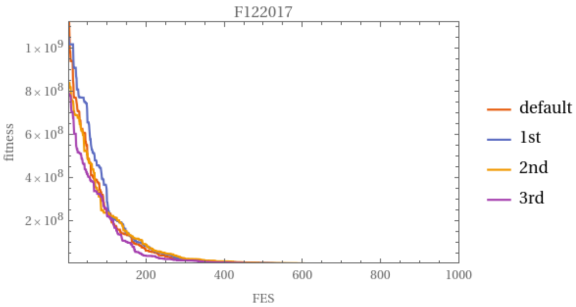
Graphs of average convergence for each test function with 4 different parameters setting, each convergence was calculated 50 times, for better observability only first 1000 FEs are shown.

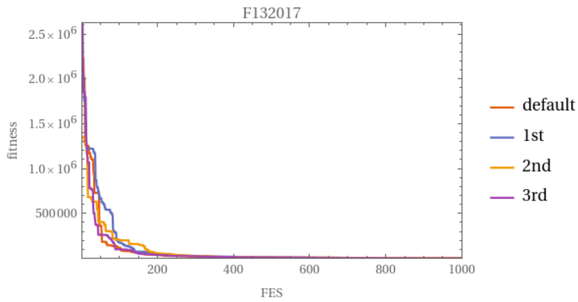
Table 38 Graphs of average convergencies

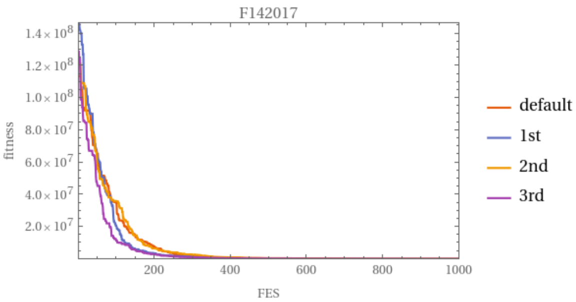


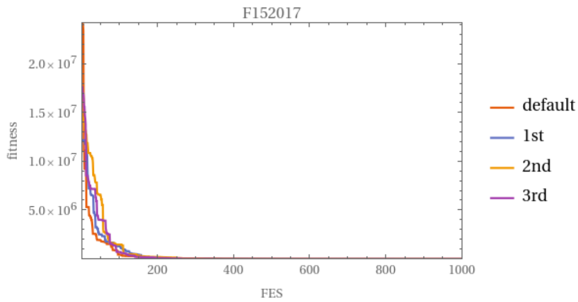

F102017
fitness
FES
default
1st
2nd
3rd

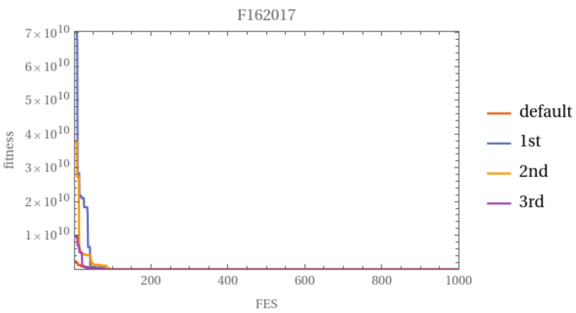

F112017
fitness
FES
default
1st
2nd
3rd

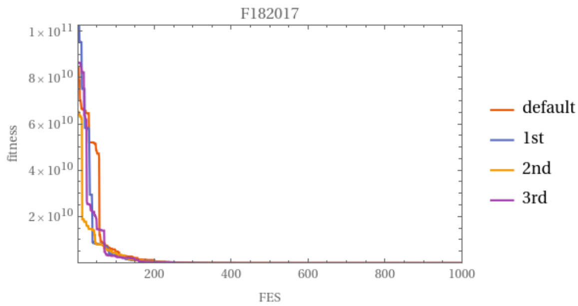

F122017
fitness
FES
default
1st
2nd
3rd

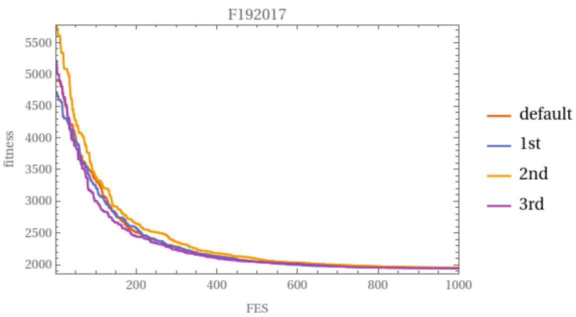

F132017
fitness
FES
default
1st
2nd
3rd

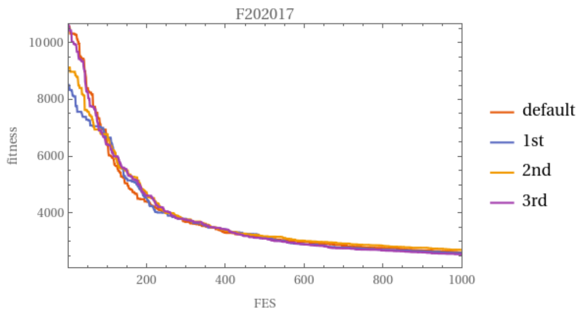

F142017
fitness
FES
default
1st
2nd
3rd

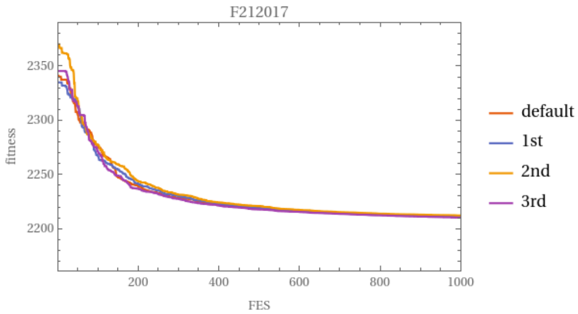

F152017
fitness
FES
default
1st
2nd
3rd

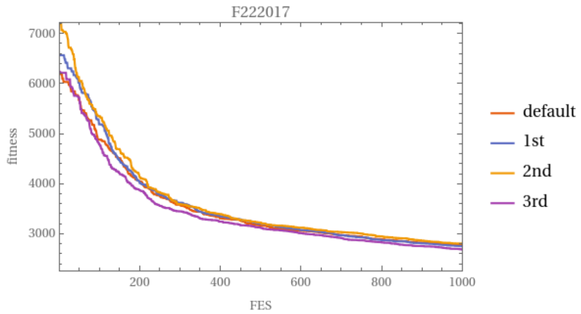

F162017
fitness
FES
default
1st
2nd
3rd

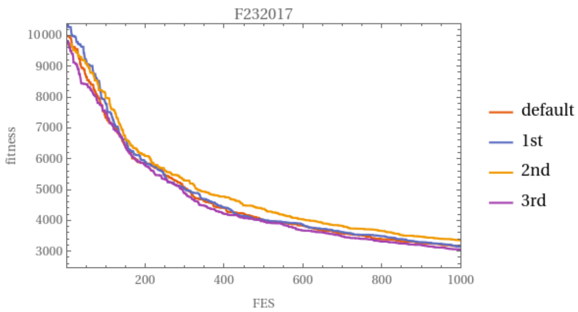

F182017
fitness
FES
default
1st
2nd
3rd

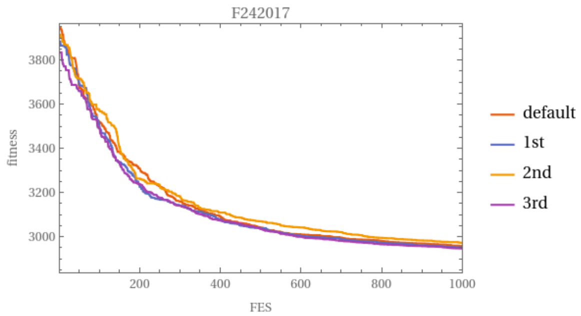

F192017
fitness
FES
default
1st
2nd
3rd

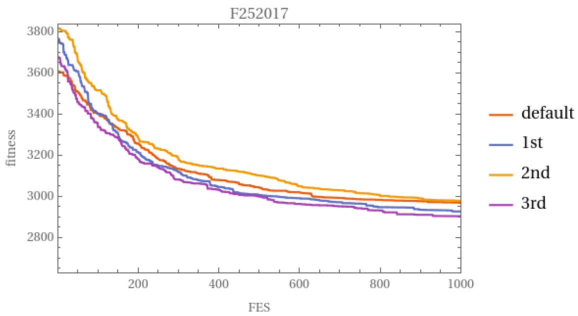

F202017
fitness
FES
default
1st
2nd
3rd

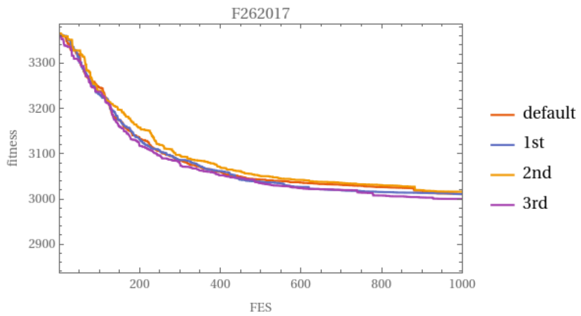

F212017
fitness
FES
default
1st
2nd
3rd

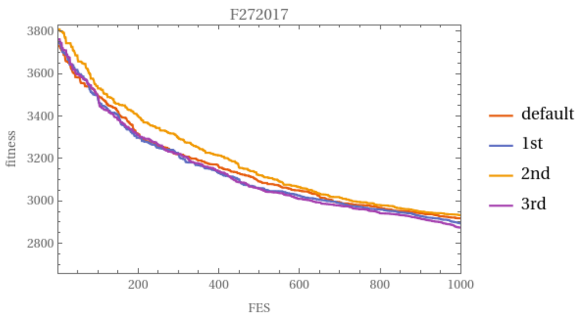

F222017
fitness
FES
default
1st
2nd
3rd

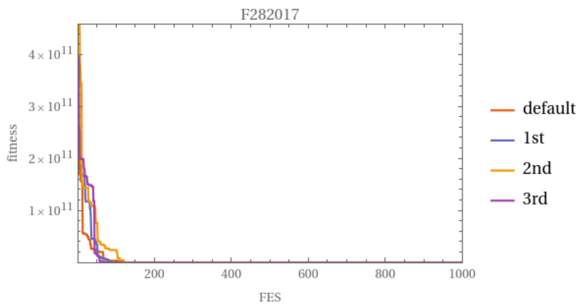

F232017
fitness
FES
default
1st
2nd
3rd


F242017
fitness
FES
default
1st
2nd
3rd


F252017
fitness
FES
default
1st
2nd
3rd


F262017
fitness
FES
default
1st
2nd
3rd


F272017
fitness
FES
default
1st
2nd
3rd


F282017
fitness
FES
default
1st
2nd
3rd

7.2 CEC 2022 test suite

Four Tables of Basic Statistics (Minimum, Maximum, Mean, Median, and Standard Deviation) for Officially Recommended Parameter Settings, and the Three Best Parameter Values Obtained from irace, each function was runed 50 times:

Table 39 Basic statistics for default parameters

default					
Function	Min	Max	Mean	Median	Standard Deviation
F1	0	0	0	0	0
F2	0	8.92	1.79	0	3.06
F3	0	0	0	0	0
F4	2.98	25.87	11.48	10.94	4.97
F5	0	0	0	0	0
F6	0.03	6.62	1.15	1.06	1.19
F7	0	1	0.11	0	0.26
F8	0.14	20.82	3.06	0.84	6.04
F9	229.28	229.28	229.28	229.28	0
F10	0.44	100.54	58.54	100.23	45.99
F11	0	150.43	9.03	0	36.09

Table 40 Basic statistics for 1st set of parameters

1st					
Function	Min	Max	Mean	Median	Standard Deviation
F1	0	0	0	0	0
F2	0	8.92	0.5	0	1.63
F3	0	0.12	0	0	0.02
F4	3.98	29.85	12.3	10.94	5.78
F5	0	0	0	0	0
F6	0.03	4.06	0.98	1.06	0.97
F7	0	4.97	0.37	0	0.81
F8	0.01	20.46	2.77	0.62	6.49
F9	0	229.28	224.7	229.28	32.43
F10	0.31	209.02	88.62	100.25	36.82
F11	0	150.43	6.02	0	29.78

Table 41 Basic statistics for 2nd set of parameters

2nd					
Function	Min	Max	Mean	Median	Standard Deviation
F1	0	0	0	0	0
F2	0	3.99	0.48	0	1.31
F3	0	0	0	0	0
F4	2.98	26.86	12.52	11.44	5.75
F5	0	0.09	0	0	0.01
F6	0.04	5.32	0.98	1.06	0.88
F7	0	0.99	0.09	0	0.26
F8	0	20.14	1.02	0.47	3.02
F9	229.28	229.28	229.28	229.28	0
F10	0.25	100.37	70.08	100.2	40.94
F11	0	150.43	6.02	0	29.78

Table 42 Basic statistics for 3rd set of parameters

3rd					
Function	Min	Max	Mean	Median	Standard Deviation
F1	0	0	0	0	0
F2	0	3.99	0.24	0	0.96
F3	0	0	0	0	0
F4	1.99	31.84	12.34	11.94	5.79
F5	0	0.45	0.01	0	0.07
F6	0.02	5.1	1.18	1.1	1.03
F7	0	2.32	0.33	0	0.53
F8	0	20.3	2.79	0.67	6.33
F9	229.28	229.28	229.28	229.28	0
F10	0.12	211.57	100.16	100.27	41.87
F11	0	151.08	6.03	0	29.84

In the following graph, the outcome of a Friedman mean ranks test conducted on the dataset derived from multiple parameter settings is presented.

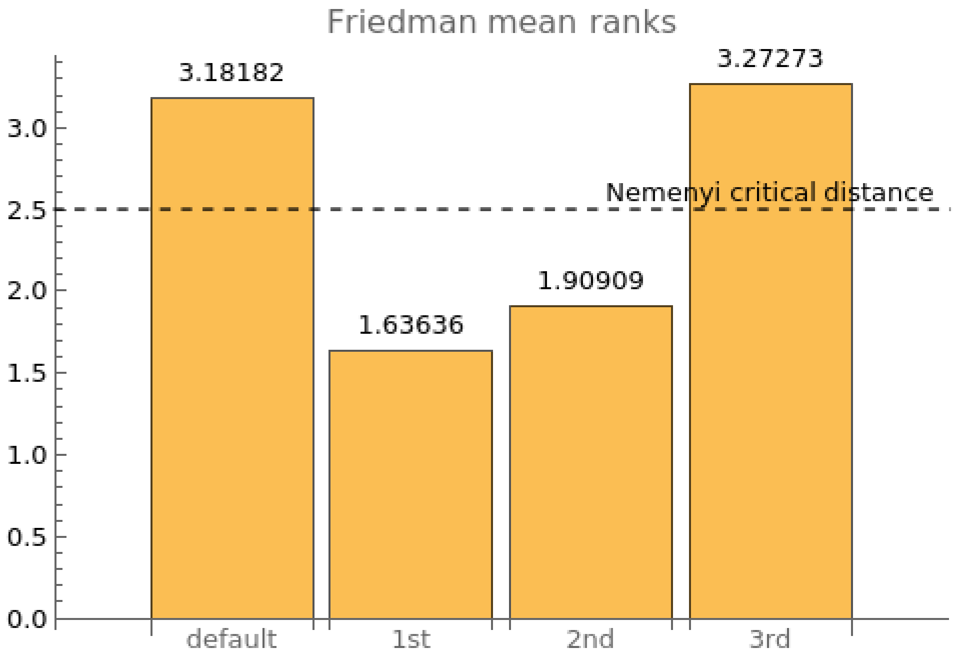


Figure 13 Friedman mean ranks dor CEC 2022 test suite

Table of pairwise comparisons of parameter settings:

Table 43 Wilcoxon rank sign test for CEC 2022 test suite

Function	1st	2nd	3rd
F1	-	+	-
F2	-	-	+
F3	-	-	+
F4	-	+	-
F5	-	+	+
F6	-	+	-
F7	-	-	+
F8	+	-	+
F9	-	+	+
F10	+	-	+
F11	-	+	+
count of (-)	9	5	3
pValues	0.02	0.35	0.62

Three Tables of Basic Statistics for CEC 2022 test functions (Minimum, Maximum, Mean, Median, and Standard Deviation) for Three Best Parameter Values Obtained from irace trained on CEC 2017 test functions:

Table 44 Basic statistics for CEC 2022 test suite using 1st set of parameters obtained racing CEC 2017 test suite

1st					
Function	Min	Max	Mean	Median	Standard Deviation
F1	1.18	389.22	39.51	18.86	65.29
F2	0	15.29	2.56	0.28	3.65
F3	0	0.17	0.06	0.05	0.04
F4	2.98	27.98	9.76	8.96	5.31
F5	0	7.32	0.89	0.54	1.33
F6	5.15	1863.89	181.18	69.14	302.72
F7	0.48	28.33	13.37	17.47	9.56
F8	1.68	27.21	21.19	24.37	7.57
F9	229.28	229.76	229.33	229.31	0.09
F10	100.26	100.92	100.51	100.51	0.12
F11	0.01	150.59	9.32	0.12	36.04

Table 45 Basic statistics for CEC 2022 test suite using 2nd set of parameters obtained racing CEC 2017 test suite

2nd					
Function	Min	Max	Mean	Median	Standard Deviation
F1	15.64	807.34	309.45	246.55	216.08
F2	0	8.96	1.91	0.15	3.1
F3	0	0.28	0.03	0.03	0.04
F4	3.32	29.14	12.05	11.03	5.31
F5	0	7.86	1.22	0.47	1.75
F6	7.66	2481.48	248.3	69.36	417.39
F7	0.58	25.66	11.72	8.24	9.54
F8	1.87	23.37	18.01	21.15	6.35
F9	229.29	229.73	229.39	229.35	0.11
F10	100.25	100.89	100.5	100.5	0.14
F11	0.27	151.08	19.77	1.53	48.82

Table 46 Basic statistics for CEC 2022 test suite using 3rd set of parameters obtained racing CEC 2017 test suite

3rd					
Function	Min	Max	Mean	Median	Standard Deviation
F1	2.23	569.31	116.06	64.42	133.69
F2	0	11.6	2.99	2.92	3.17
F3	0.01	0.32	0.08	0.06	0.08
F4	2.99	24.86	10.71	10.82	4.58
F5	0	9.05	1.44	0.69	1.88
F6	7.44	1666.03	194.87	79.17	305.56
F7	1.11	25.03	14.04	17.22	8.82
F8	2.44	27.17	21.39	24.17	6.95
F9	229.3	231.62	229.53	229.42	0.41
F10	100.3	100.99	100.55	100.51	0.16
F11	0.15	150.58	28.11	1.17	57.92

Friedman mean ranks for values obtained from irace raced on CEC 2017 test functions:

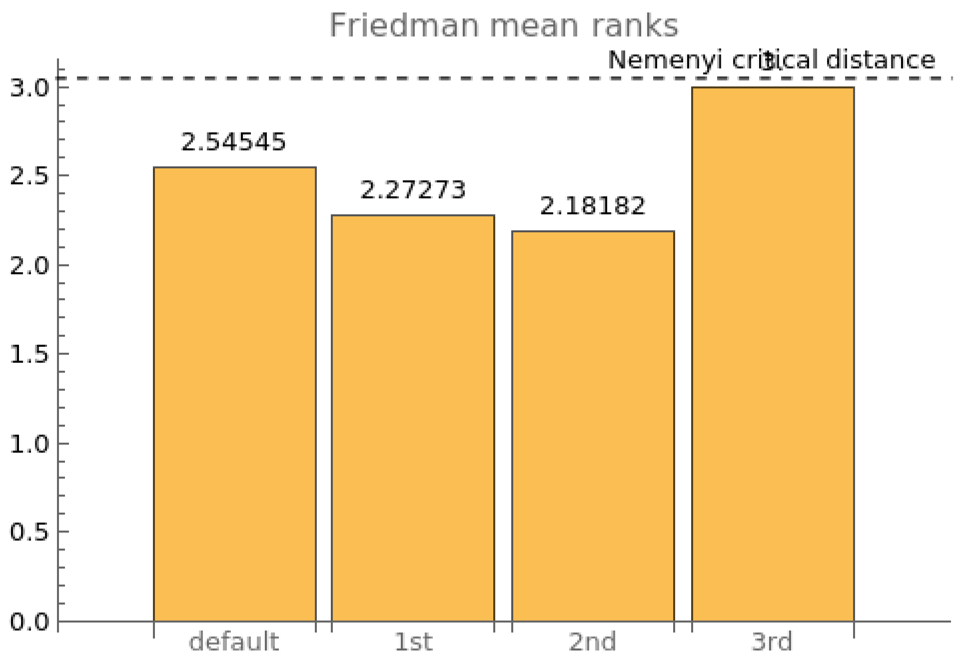


Figure 14 Friedman mean ranks for CEC 2022 test suite using parameters obtained racing CEC 2017 test suite

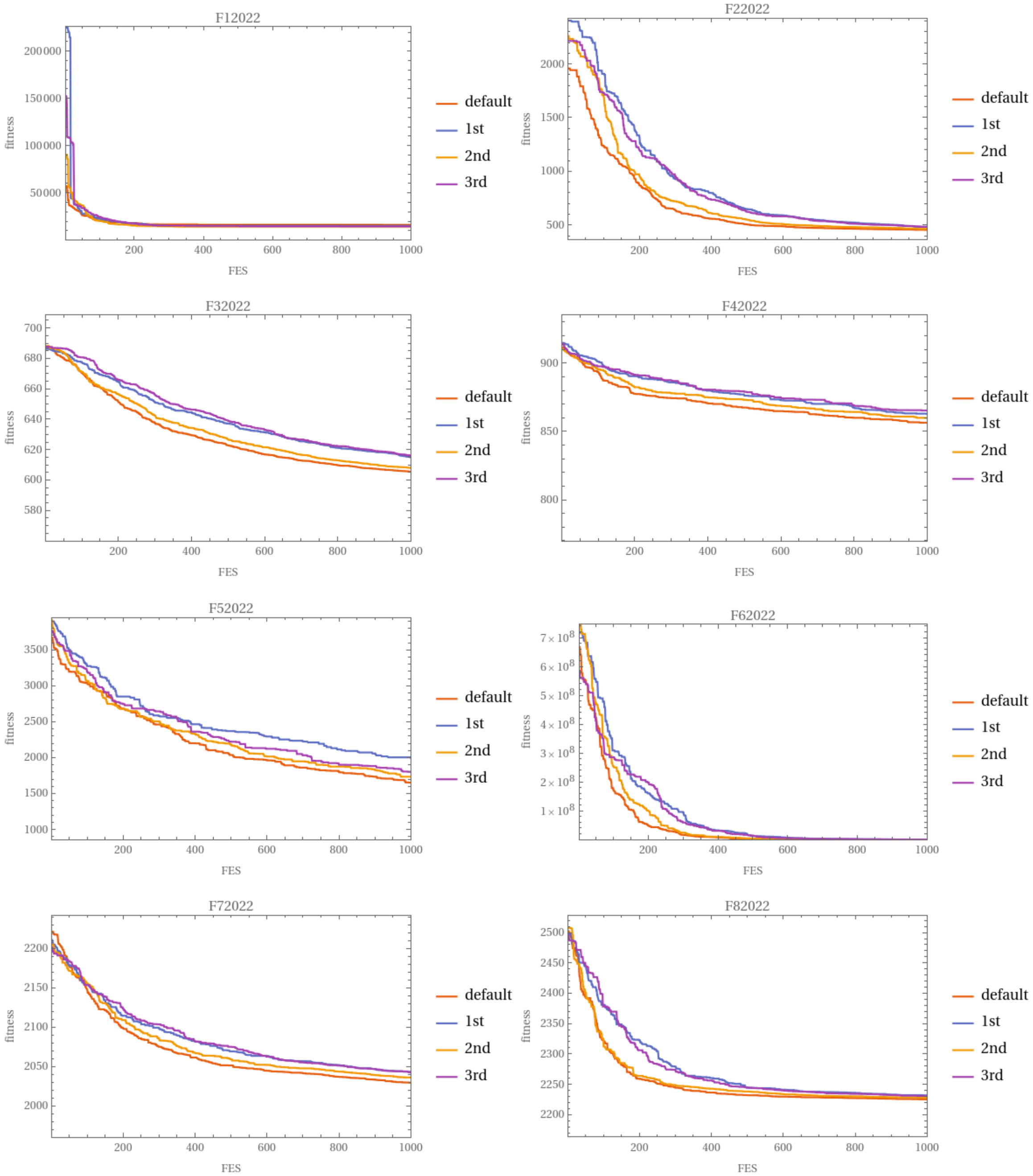
pairwise comparisons of parameter settings obtained from CEC 2017 test suite against the subsequent setting using the Wilcoxon Rank Sign:

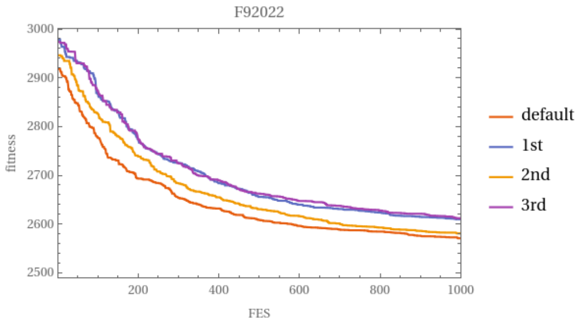
Table 47 Wilcoxon rank sign test for CEC 2022 test suite with parameters obtained racing CEC 2017 test suite

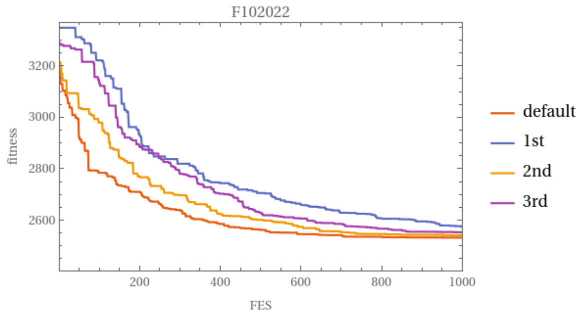
Function	1st	2nd	3rd
F1	-	+	-
F2	-	-	-
F3	+	-	+
F4	+	+	-
F5	+	+	+
F6	-	-	+
F7	+	-	+
F8	-	-	+
F9	-	+	-
F10	+	-	+
F11	-	+	+
count of (-)	6	6	4
pValues	0.69	0.4	0.17

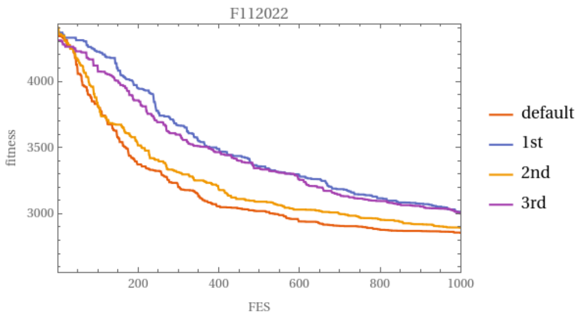
Graphs of average convergence for each test function with 4 different parameters setting, each convergence was calculated 50 times, for better observability only first 1000 FEs are shown.

Table 48 Graphs of average convergencies









7.3 Summary

The statistical evaluation of the irace tuned parameters for the SOMA Pareto function indicates that while certain parameter settings may yield lower error rates or show better performance on specific test functions, these advantages are not universally significant across all tested scenarios. This suggests a complex interaction between parameter settings and function characteristics, highlighting the importance of context-specific tuning.

Yet the 1st set of parameters raced on each test suite perform better on given test functions, however better performance on different function was not statistically proven.

The lack of universally superior parameter settings underscores the need for adaptive strategies in optimization, where parameter tuning is tailored not only based on the algorithm and its theoretical performance but also according to specific problem characteristics and requirements.

8 SOMA-CLP

Tables of default parameters obtained from [9] and tuned from irace package:

Table 49 Parameters obtained for CEC 2017 test suite

parameter	pathA	stepA	path	step
default	3	0.33	2	0.11
1st	1.58	0.55	2.3	0.84
2nd	1.68	1.01	1.91	0.54
3rd	1.19	0.42	1.57	1.15

Table 50 Parameters obtained for CEC 2022 test suite

parameter	pathA	stepA	path	step
default	3	0.33	2	0.11
1st	2.7	1.1	1.32	0.49
2nd	1.61	1.06	1.57	0.76
3rd	2.43	0.99	2	0.62

8.1 CEC 2017 test suite

Four Tables of Basic Statistics (Minimum, Maximum, Mean, Median, and Standard Deviation) for cost function error for officially recommended parameter settings , and the three best parameter values obtained from irace, each function was runed 50 times:

Table 51 Basic statistics for default parameters

default					
Function	Min	Max	Mean	Median	Standard Deviation
F12017	0.03	8.7	1.06	0.38	1.68
F22017	0	0	0	0	0
F32017	0.16	4.53	2.59	2.81	1.07
F42017	1	10.2	4.78	4.56	1.82
F52017	0	0	0	0	0
F62017	11.75	19.56	14.9	14.67	1.88
F72017	8	16	11.81	12	2.27
F82017	0	0	0	0	0
F92017	0.84	264.81	61.11	31.79	65.7
F102017	1.52	17.39	6.19	4.8	3.53
F112017	116.78	1639.9	457.67	385.47	293.49
F122017	5.07	190.59	22.15	15.27	26.74
F132017	7.23	26.98	22.21	22.93	3.74
F142017	0.59	9.88	4.36	3.67	2.53
F152017	0.04	46.66	1.28	0.22	6.56
F162017	10.68	33.05	25.69	25.68	3.15
F172017	0.17	3.24	1.12	0.9	0.69
F182017	0.73	2.68	1.64	1.61	0.3
F192017	4.92	23.67	11.42	11.07	4.08
F202017	100.01	100.63	100.18	100.15	0.12
F212017	100	102.7	101.33	101.4	0.7
F222017	100.02	102.37	100.37	100.21	0.48
F232017	103.38	200.19	143.14	126.52	37.21
F242017	476.94	481.65	479.43	479.61	1.18
F252017	0.1	467.46	38.22	1.05	126.6
F262017	10.56	397.76	217.01	282.74	179.71
F272017	0	0	0	0	0
F282017	1023.88	1562.4	1236.32	1230.79	124.06

Table 52 Basic statistics for 1st set of parameters

1st					
Function	Min	Max	Mean	Median	Standard Deviation
F12017	0	0	0	0	0
F22017	0	0	0	0	0
F32017	0	6.09	3.07	3.47	1.36
F42017	0.99	7.82	3.43	3.07	1.52
F52017	0	0	0	0	0
F62017	11.37	18.08	13.64	13.36	1.56
F72017	6.27	18.35	12.88	12.52	2.09
F82017	0	0	0	0	0
F92017	2.57	640.79	248.89	214.01	180.1
F102017	0	9.66	2.9	2.57	1.84
F112017	0.34	1444.06	278.87	171.71	334.68
F122017	4.38	50.02	17.91	16.54	11.97
F132017	1.28	23.41	15.48	20.96	8.76
F142017	0.3	5.04	1.62	1.57	0.91
F152017	0.04	17.3	1.3	0.6	2.53
F162017	21.5	31.98	25.06	24.35	2.45
F172017	0.01	0.55	0.33	0.41	0.18
F182017	0.98	2.95	1.82	1.7	0.44
F192017	1.51	7.93	3.98	3.54	1.58
F202017	100	100	100	100	0
F212017	100	102.09	100.58	100	0.72
F222017	100	109.55	100.2	100	1.35
F232017	100.01	200	120.23	107.22	29.22
F242017	476.5	482.33	478.77	478.62	1.41
F252017	0	469.93	130.27	0	211.02
F262017	7.67	397.5	322.02	390.6	142.59
F272017	0	0	0	0	0
F282017	1006.69	1874.17	1265.89	1258.72	179.57

Table 53 Basic statistics for 2nd set of parameters

2nd					
Function	Min	Max	Mean	Median	Standard Devi-ation
F12017	50.25	540252286.07	56234080.72	11275859.74	109421771.52
F22017	2.92	944.68	280.85	184	239.23
F32017	3.05	51.82	15.75	12.32	11.58
F42017	3.92	79.73	16.63	12.24	14.64
F52017	0	0	0	0	0
F62017	4.65	7572.81	276.5	32.57	1120.17
F72017	14.04	41.32	24.86	24.05	6.65
F82017	0	0.09	0.01	0.01	0.02
F92017	17.99	1272.13	724.23	787.55	312.27
F102017	34.1	1310.5	322.9	234.18	306.34
F112017	484.01	25091.59	8163.6	6636.92	5638.83
F122017	30.77	6373.58	1216.19	686.24	1424.71
F132017	25.69	5885.85	599.86	86.51	1222.29
F142017	19.45	217787.72	8381.87	2067.54	30549.8
F152017	0.5	371.06	46.68	19.39	66.96
F162017	26.52	973.89	125.44	58.71	169
F172017	1641.15	17006084237.73	684230862.72	15691.07	2628614188.31
F182017	2.03	7215.51	555.57	26.3	1370.66
F192017	12.46	40.3	28.5	28.94	5.7
F202017	109.08	566.7	223.47	223.01	74.72
F212017	101.63	108.33	103.72	103.61	1.11
F222017	116.97	456.1	252.85	248.1	81.83
F232017	166.87	739.97	358.12	321.23	136.94
F242017	482.26	573.03	524.89	523.53	21.14
F252017	9.56	501.26	341.37	477.66	212.62
F262017	198.32	422.25	390.7	400.96	43.98
F272017	1.7	375.5	72.26	43.52	76.9
F282017	1847.47	44024.37	11796.41	8222.44	10189.98

Table 54 Basic statistics for 3rd set of parameters

3rd					
Function	Min	Max	Mean	Median	Standard Deviation
F12017	0	215019	4635.04	0.41	30398.86
F22017	0	17.56	0.74	0	2.86
F32017	0.06	7.37	3.67	4.06	1.53
F42017	0.4	7.46	4.06	3.79	1.84
F52017	0	0	0	0	0
F62017	11.74	24.7	15.92	15.28	2.97
F72017	10.3	20.74	14.43	14.22	1.86
F82017	0	0	0	0	0
F92017	10.36	694.16	290.46	311.07	170.99
F102017	0.02	8.78	4	3.54	2.04
F112017	175.16	4769.28	1459.3	1194.94	1038.81
F122017	4.72	924.5	98.42	40.26	167.75
F132017	1.06	23.93	16.1	21.12	8.56
F142017	0.37	1234.64	31.89	2.51	174.22
F152017	0.1	7.42	1.22	0.65	1.51
F162017	11.18	30.5	23.36	23.75	3.18
F172017	0.38	3214.18	155.08	4.88	596.13
F182017	1.34	3.34	2.01	1.87	0.52
F192017	1.12	6.06	3.37	3.59	1.31
F202017	100	104.32	100.27	100.03	0.76
F212017	100.06	102.99	101.71	101.75	0.64
F222017	100	200.79	107.37	101.27	18.24
F232017	100.22	200.75	164.53	168.36	35.84
F242017	479.62	514.03	489.69	486.98	8.83
F252017	0	475.4	135.03	6.79	207.43
F262017	19.98	399.63	334.08	392.51	124.54
F272017	0	2	0.13	0	0.43
F282017	1249.69	4870.92	1999.33	1831.31	643.15

In the following graph, the outcome of a Friedman mean ranks test conducted on the dataset derived from multiple parameter settings is presented. This statistical test is utilized to assess the differences in performance across various configurations under non-parametric conditions. The results are depicted in a table graph format, where the Nemenyi critical distance is clearly marked. This visualization aids in identifying statistically significant differences between the ranks of the parameter settings, providing insights into which configurations yield the best performance relative to others. The application of the Nemenyi post-hoc test delineates the parameter sets that significantly differs.

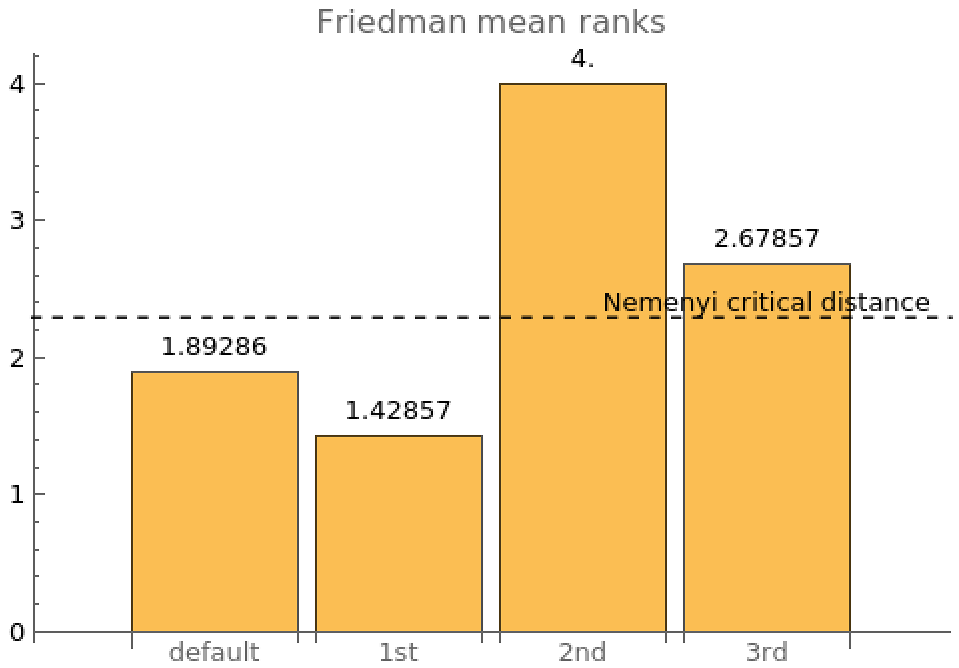


Figure 15 Friedman mean ranks for CEC 2017 test suite

In following table, we explore the pairwise comparisons of the first three parameter settings against the subsequent setting using the Wilcoxon Rank Sign Test. Each test aims to assess whether one parameter setting significantly outperforms the following setting in terms performance. The second-to-last row of the table indicates the count of instances where a parameter setting outperforms the next. This provides a direct comparison of how often one setting yields better results than its successor. The final row displays the p-values obtained from the Wilcoxon tests, highlighting the statistical significance of each comparison. A p-value under 0.05 indicates a statistically significant difference in performance, suggesting that one setting is reliably better than the other under the tested conditions.

This table serves not only to identify which parameter settings perform better but also to quantify the significance of these differences, providing a robust framework for decision-making regarding parameter optimization.

After that, three tables of basic statistics for CEC 2017 test functions (Minimum, Maximum, Mean, Median, and Standard Deviation) for Three Best Parameter Values Obtained from irace trained on CEC 2022 test functions follows.

Table 55 Wilcoxon rank sign test for CEC 2017 test suite

Function	1st	2nd	3rd
F12017	-	+	-
F22017	-	+	-
F32017	+	+	-
F42017	-	+	-
F52017	+	+	-
F62017	-	+	-
F72017	+	+	-
F82017	-	+	-
F92017	+	+	-
F102017	-	+	-
F112017	-	+	-
F122017	-	+	-
F132017	-	+	-
F142017	-	+	-
F152017	+	+	-
F162017	-	+	-
F172017	-	+	-
F182017	+	+	-
F192017	-	+	-
F202017	-	+	-
F212017	-	+	-
F222017	-	+	-
F232017	-	+	-
F242017	-	+	-
F252017	+	+	-
F262017	+	+	-
F272017	-	+	-
F282017	+	+	-
count of (-)	19	0	28
pValues	0.33	0	0

Table 56 Basic statistics for CEC 2017 test suite using 1st set of parameters obtained racing CEC 2022 test suite

1st					
Function	Min	Max	Mean	Median	Standard Deviation
F12017	250.52	1615873.8	103890.64	10774.14	306366.04
F22017	0.03	379.17	16.26	3.1	54.59
F32017	0.28	21.5	6.98	6.45	3.47
F42017	2.05	22.47	9.67	9.25	4.2
F52017	0	0	0	0	0
F62017	7	32.6	20.97	21.75	5.39
F72017	7.59	34.19	18.55	18	5.63
F82017	0	0.09	0.01	0	0.02
F92017	9.61	760.21	211.45	171.67	176.8
F102017	11.95	403.75	105.17	80.61	82.36
F112017	711.4	33947.23	8750.64	6048.93	7977
F122017	34.12	5749.87	1664.78	1284.29	1614.68
F132017	27.36	12712.45	770.62	76.09	2186.09
F142017	20.24	16189.22	2388.26	552.8	3439.26
F152017	0.28	1786.73	70.13	16.26	251.73
F162017	32.76	705.31	122.32	59.21	139.16
F172017	29.61	29341.21	5224.68	3107.03	6222.51
F182017	1.33	5868.81	542.28	17.02	1207.36
F192017	7.59	46.33	26.57	27.8	8.51
F202017	100.75	248.33	146.71	119.54	53.31
F212017	100.04	104.76	102.82	102.85	0.86
F222017	100.57	309.32	146.91	131.82	46.88
F232017	200.26	348.33	217.81	208.64	26.59
F242017	480.21	525.99	493.87	487.95	12.63
F252017	3.24	478.8	438.56	471.54	115.71
F262017	132.24	403.93	392.75	398.84	37.72
F272017	0.05	44.74	10.69	6.09	11.64
F282017	1257.3	18366.18	4826.82	3163.79	3970.58

Table 57 Basic statistics for CEC 2017 test suite using 2nd set of parameters obtained racing CEC 2022 test suite

2nd					
Function	Min	Max	Mean	Median	Standard Deviation
F12017	9.75	386208.06	18124.94	1608.45	58396.3
F22017	0	147.8	14.2	3.95	30.84
F32017	0.52	11.56	5.82	5.03	2.53
F42017	3.16	25.03	10.24	9.45	4.26
F52017	0	0	0	0	0
F62017	11.87	36.19	22.58	21.14	5.54
F72017	8.89	44.42	22.97	23.17	7.13
F82017	0	0.09	0.01	0	0.02
F92017	16.84	1086.83	502.09	453.02	305.71
F102017	3.94	1551.64	122.4	59.62	242.82
F112017	508.94	33935.89	9339.27	7298.66	8349.45
F122017	44.7	10058.38	2053.01	654.52	2571.04
F132017	22.73	4556.14	222.98	43.7	711.86
F142017	15.27	18274.87	2535.39	867.88	4114.26
F152017	0.44	1388.81	54.23	11.41	195.39
F162017	28.09	694.9	88.81	49.75	119.43
F172017	59.54	27622.2	7053	4022.11	7024.93
F182017	2.35	6376.58	224.42	5.01	956.18
F192017	22.02	43.56	33.52	33.07	5.31
F202017	100.48	250.2	154.7	133.7	50.44
F212017	100	103.95	101.97	102.17	1.14
F222017	100.13	229	143.32	107.74	48.05
F232017	107.37	315.63	206.57	202.85	29.71
F242017	480.52	518.86	495.29	491.54	11.28
F252017	7.91	480.01	430.48	473.03	128.17
F262017	386.88	405.05	398.06	398.26	3.43
F272017	0.01	40.2	5.9	0.39	9.44
F282017	1060.5	21306.96	4515.45	2571.37	4720.28

Table 58 Basic statistics for CEC 2017 test suite using 3rd set of parameters obtained racing CEC 2022 test suite

3rd					
Function	Min	Max	Mean	Median	Standard Deviation
F12017	44.9	15105.85	2993.97	1307.31	3726.25
F22017	0.01	2.56	0.57	0.35	0.69
F32017	0.07	9.36	5.94	5.41	2.53
F42017	3.27	23.4	11.47	10.59	4.74
F52017	0	0	0	0	0
F62017	14.77	35.9	24.6	24.9	5.19
F72017	12	47.66	24.38	23.36	6.89
F82017	0	0.09	0	0	0.01
F92017	31.4	980.05	464.56	392.21	236.71
F102017	8.6	241.12	73.32	64.63	49.48
F112017	746.24	27420.23	7015.31	4329.31	6253.46
F122017	90.17	7214.42	2022.08	1151.98	2045.82
F132017	22.69	2999.97	146.81	49.13	434.04
F142017	20.17	5713.41	798.97	256.27	1220.09
F152017	0.5	105.86	20.18	4.45	28.1
F162017	27.77	273.09	63.94	49.4	55.58
F172017	36.82	18728.89	3395.58	1149.51	4962.44
F182017	0.88	7.81	3.69	3.57	1.31
F192017	22.87	46	32.47	31.35	5.4
F202017	100.39	219.6	126.95	108.07	37.15
F212017	100.01	103.74	102.15	102.26	0.97
F222017	100.08	218.1	134.14	108.02	42.18
F232017	118.82	269.98	204.36	202.18	18.39
F242017	477.47	510.09	487.5	483.31	10.7
F252017	5.73	478.63	406.88	472.49	153.14
F262017	390.4	401.7	397.93	397.8	2.11
F272017	0.01	24.14	2.65	0.11	5.83
F282017	1293.18	8101.51	2868.24	2317.94	1664.78

Friedman mean ranks for values obtained from irace raced on CEC 2022 test functions:

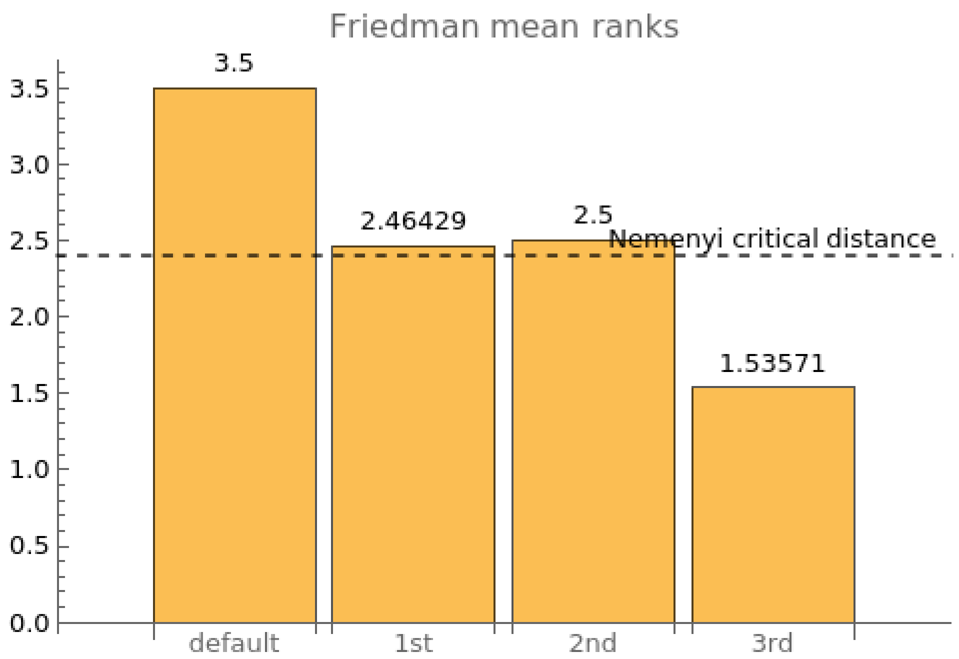


Figure 16 Friedman mean ranks for CEC 2017 test suite using parameters obtained racing CEC 2022 test suite

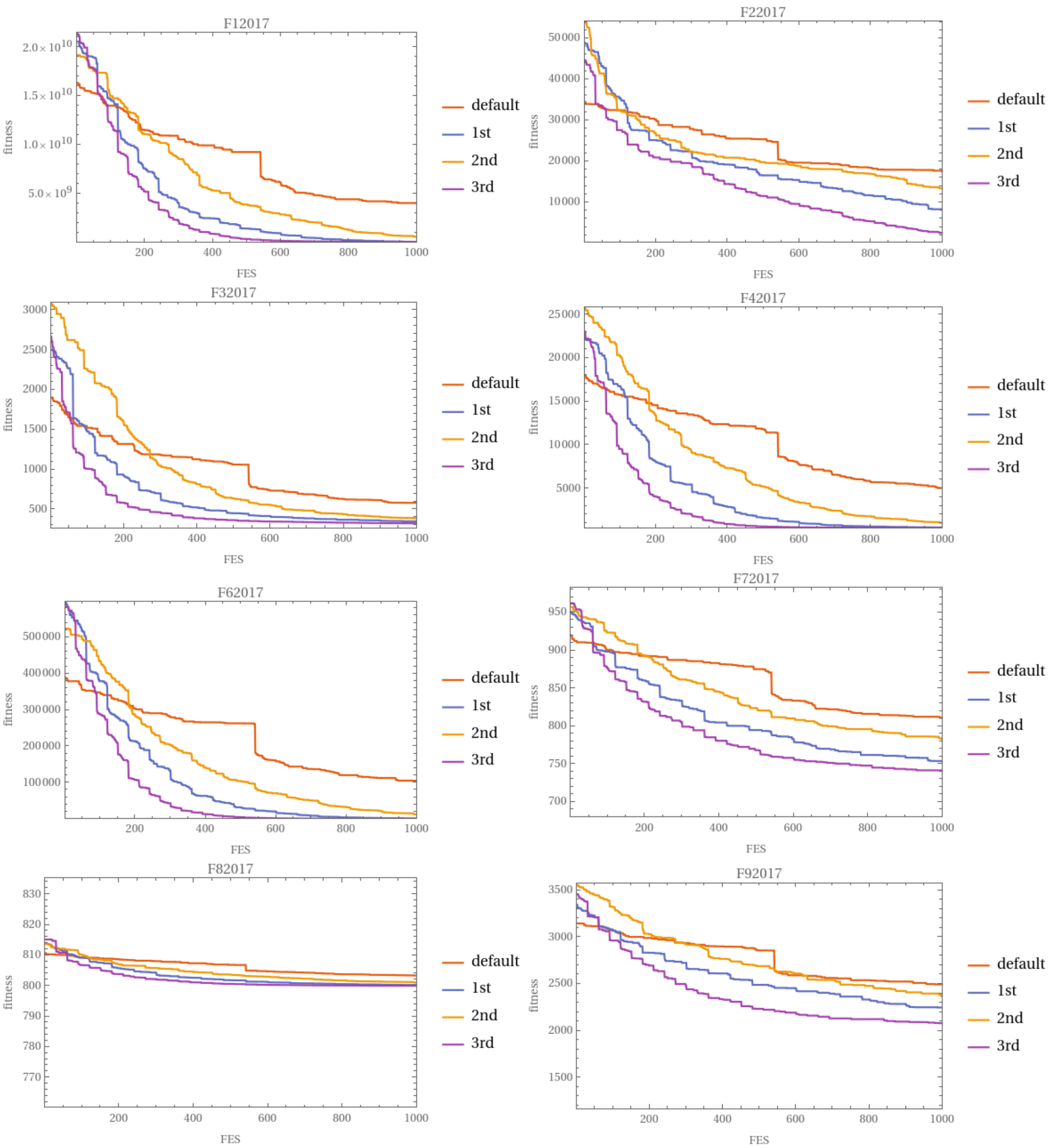
Pairwise comparisons of parameter settings obtained from CEC 2022 test suite against the subsequent setting using the Wilcoxon Rank Sign is in following table:

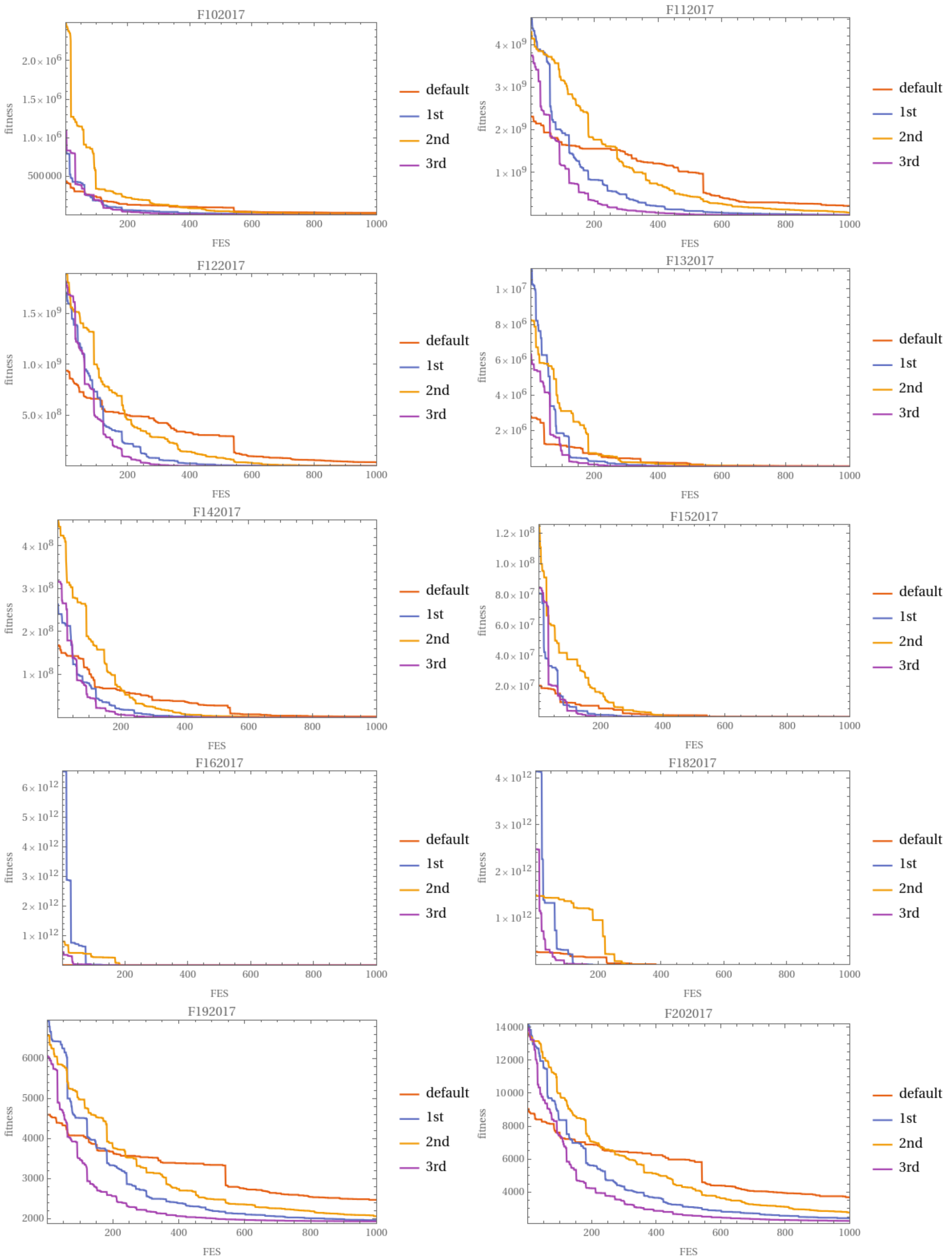
Table 59 Wilcoxon rank sign test for CEC 2017 test suite with parameters obtained racing CEC 2022 test suite

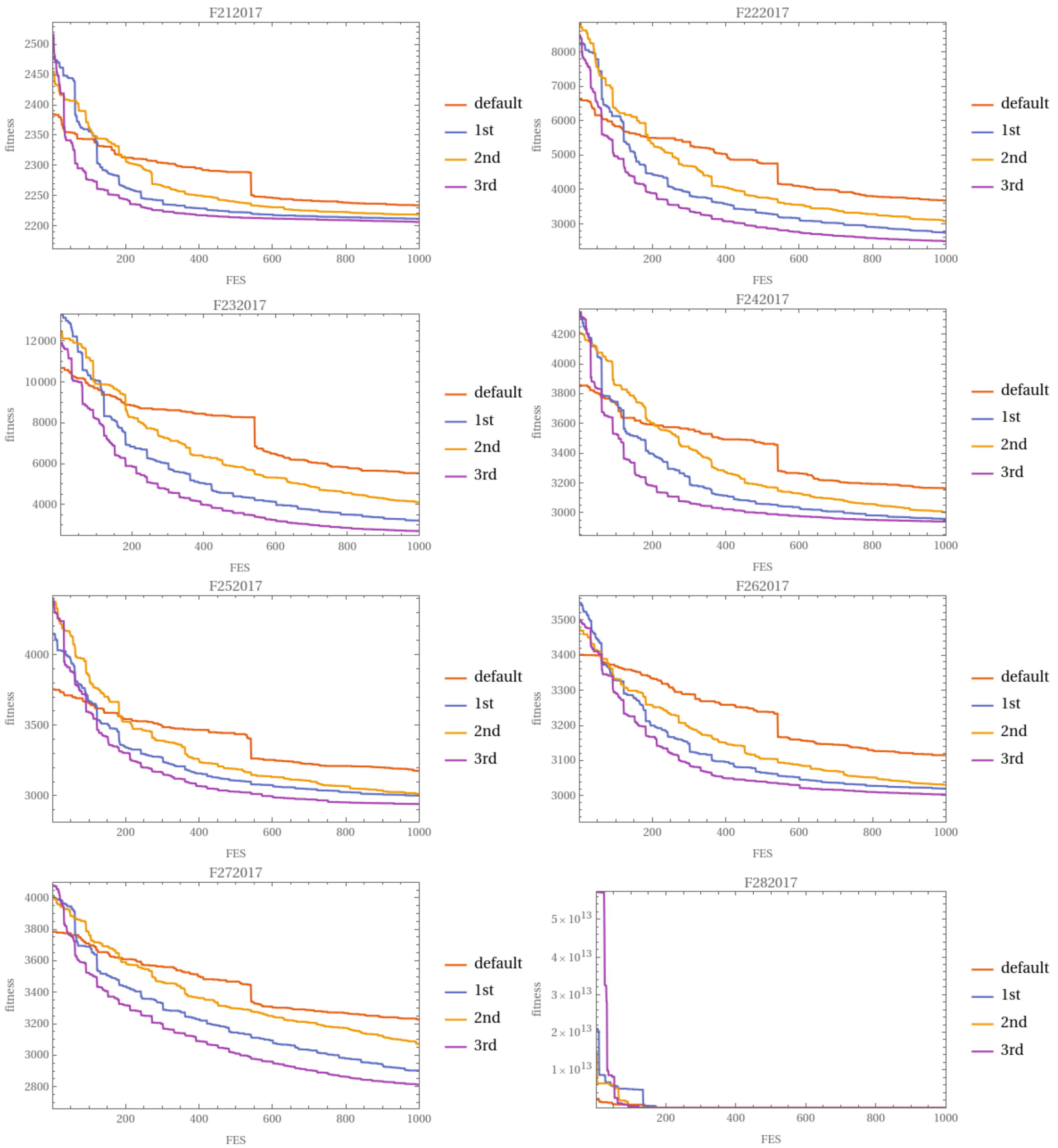
Function	1st	2nd	3rd
F12017	-	-	-
F22017	-	-	-
F32017	-	-	+
F42017	-	+	+
F52017	-	+	-
F62017	-	+	+
F72017	-	+	+
F82017	-	-	-
F92017	-	+	-
F102017	-	+	-
F112017	-	+	-
F122017	-	+	-
F132017	+	-	-
F142017	-	+	-
F152017	+	-	-
F162017	+	-	-
F172017	-	+	-
F182017	+	-	-
F192017	-	+	-
F202017	-	+	-
F212017	-	-	+
F222017	-	-	-
F232017	-	-	-
F242017	-	+	-
F252017	+	-	-
F262017	+	+	-
F272017	-	-	-
F282017	+	-	-
count of (-)	21	14	23
pValues	0.08	0.99	0

Graphs of average convergence for each test function with 4 different parameters setting, each convergence was calculated 50 times, for better observability only first 1000 FEs are shown.

Table 60 Graphs of average convergencies




F102017
fitness
FES
default
1st
2nd
3rd
F112017
fitness
FES
default
1st
2nd
3rd
F122017
fitness
FES
default
1st
2nd
3rd
F132017
fitness
FES
default
1st
2nd
3rd
F142017
fitness
FES
default
1st
2nd
3rd
F152017
fitness
FES
default
1st
2nd
3rd
F162017
fitness
FES
default
1st
2nd
3rd
F182017
fitness
FES
default
1st
2nd
3rd
F192017
fitness
FES
default
1st
2nd
3rd
F202017
fitness
FES
default
1st
2nd
3rd


F212017
F222017
F232017
F242017
F252017
F262017
F272017
F282017
fitness
FES
default
1st
2nd
3rd

8.2 CEC 2022 test suite

Four Tables of Basic Statistics (Minimum, Maximum, Mean, Median, and Standard Deviation) for Officially Recommended Parameter Settings, and the Three Best Parameter Values Obtained from irace, each function was runed 50 times:

Table 61 Basic statistics for default parameters

default					
Function	Min	Max	Mean	Median	Standard Deviation
1	192.64	2614.56	993.18	889.25	613.83
2	0.12	35.95	6.43	4.3	7.45
3	0.14	0.88	0.42	0.39	0.18
4	8.36	33.89	19.96	20.24	6.19
5	0.23	11.07	2.4	1.86	2.1
6	21.95	3633.8	494.44	202.82	724.15
7	1.05	25.34	11.15	8	7.13
8	3.5	25.59	20.78	23.49	5.25
9	230.1	243.48	233.08	232.86	2.11
10	100.24	101.9	100.65	100.6	0.28
11	28.43	151.68	102.29	108.05	32.76
12	165.06	172.2	168.16	168.57	1.65

Table 62 Basic statistics for 1st set of parameters

1st					
Function	Min	Max	Mean	Median	Standard Deviation
1	49.53	1836.71	726.17	681.54	451.44
2	0.02	72.4	5.62	3	10.54
3	0	0.16	0.04	0.02	0.04
4	2.28	27.82	11.6	10.61	5.22
5	0	22.63	1.75	0.67	3.36
6	5.88	3434.42	845.58	338.48	1048.83
7	0.03	22.01	5.65	1.52	7.64
8	0.9	22.95	16.22	20.82	7.34
9	229.3	230.41	229.47	229.4	0.21
10	8.07	101.2	86.75	100.53	28.74
11	2.88	151.21	77.73	58.47	55.87
12	163.91	173.94	167.02	166.91	1.77

Table 63 Basic statistics for 2nd set of parameters

2nd					
Function	Min	Max	Mean	Median	Standard Deviation
1	30.3	1838.71	553.92	462.4	436.3
2	0.02	37.18	5.07	2.15	6.6
3	0	0.02	0	0	0
4	3	34.48	11.73	10.45	6.27
5	0	11.77	0.99	0.2	2.21
6	2.98	3034.79	600.6	323.61	717.81
7	0.17	23.74	9	4.33	8.84
8	3.62	26.11	17.58	23.07	8.25
9	229.29	235.85	229.83	229.38	1.19
10	100.28	216.33	111.99	100.7	33.75
11	1.74	152.2	87.36	87.68	55.7
12	164.17	169.29	166.81	166.93	1.42

Table 64 Basic statistics for 3rd set of parameters

3rd					
Function	Min	Max	Mean	Median	Standard Deviation
1	29.87	3625.25	601.95	407.05	630.68
2	0	13.53	3.56	1.08	3.92
3	0	0.02	0	0	0
4	6.46	32.38	17.44	15.88	6.73
5	0	2.82	0.23	0.09	0.47
6	7.77	3495.8	596.74	283.59	797.42
7	0	22.13	5.05	1.68	7.41
8	2.09	24.31	17.04	21.44	7.37
9	229.3	229.94	229.43	229.36	0.16
10	100.3	216.77	103.45	100.58	16.79
11	0.55	166.33	57.35	27.52	62.69
12	162.71	168.64	166.32	166.35	1.38

In the following graph, the outcome of a Friedman mean ranks test conducted on the dataset derived from multiple parameter settings is presented.

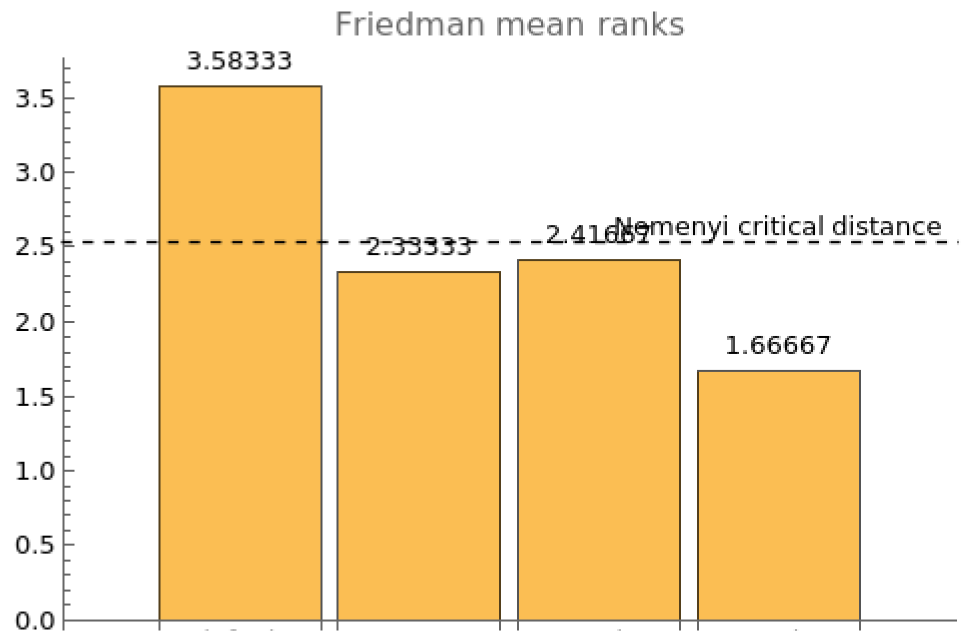


Figure 17 Friedman mean ranks for CEC 2022 test suite

Table of pairwise comparisons of parameter settings.

Table 65 Wilcoxon rank sign test for 2022 test suite

Function	1st	2nd	3rd
1	-	-	+
2	-	-	-
3	-	-	+
4	-	+	+
5	-	-	-
6	+	-	-
7	-	+	-
8	-	+	-
9	-	+	-
10	-	+	-
11	-	+	-
12	-	-	-
count of (-)	11	6	9
pValues	0.04	0.97	0.2

Three Tables of Basic Statistics for CEC 2022 test functions (Minimum, Maximum, Mean, Median, and Standard Deviation) for Three Best Parameter Values Obtained from irace trained on CEC 2017 test functions:

Table 66 Basic statistics for 1st set of parameters obtained racing CEC 2017 test suite

1st					
Function	Min	Max	Mean	Median	Standard Deviation
1	30.02	1700.82	475.67	385.5	394.11
2	0	62.64	3.58	0.29	9.28
3	0	0.07	0	0	0.01
4	3.69	32.19	11.56	9.54	6.75
5	0	2.77	0.42	0.1	0.67
6	9.23	3864.57	559.9	145.9	921.48
7	0.36	25.45	8.33	4.16	8.01
8	2.85	26.78	17.82	23.23	7.82
9	229.29	236.47	229.8	229.45	1.19
10	100.29	119.25	100.95	100.58	2.64
11	0.38	151.55	60.58	41.55	58.62

Table 67 Basic statistics for 2nd set of parameters obtained racing CEC 2017 test suite

2nd					
Function	Min	Max	Mean	Median	Standard Deviation
1	71.73	2245.13	552.66	451.5	456.15
2	0.01	54.39	6.5	4.59	9.71
3	0	0.19	0.01	0	0.03
4	2.55	25.06	11.7	11.43	5.02
5	0	25.14	1.63	0.63	3.95
6	6.25	4860.39	681.9	299.12	976.7
7	0.01	22.71	7.76	1.96	9.1
8	1.68	23.68	18.68	21.72	6.86
9	229.3	237.11	230.28	229.65	1.61
10	14.46	207.48	95.03	100.53	26.1
11	0.39	152.83	104.07	113.3	49.76

Table 68 Basic statistics for 3rd set of parameters obtained racing CEC 2017 test suite

3rd					
Function	Min	Max	Mean	Median	Standard Deviation
1	185.52	6217.25	1224.73	829.86	1062.8
2	0.3	66.23	20.08	15.01	18.01
3	0	0.78	0.05	0.01	0.14
4	4.8	25.7	14.29	13.49	4.83
5	0.08	10.14	2.27	0.97	2.84
6	6.51	3305.92	553.75	348.3	698.97
7	0.56	26.29	9.06	6.64	7.69
8	3.86	27.14	19.84	23.35	6.89
9	230.87	281.92	241.05	238.74	9.6
10	100.29	216.79	109.9	100.66	30.73
11	9.67	214.8	125.76	139.74	43.34

Friedman mean ranks for values obtained from irace raced on CEC 2017 test functions:

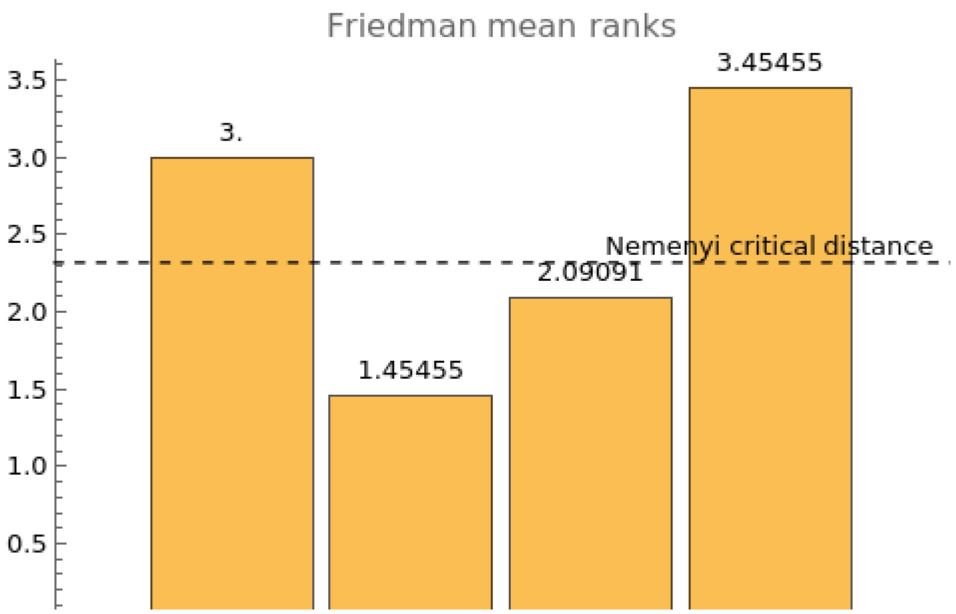


Figure 18 Friedman mean ranks for CEC 2022 test suite using parameters obtained racing CEC 2017 test suite

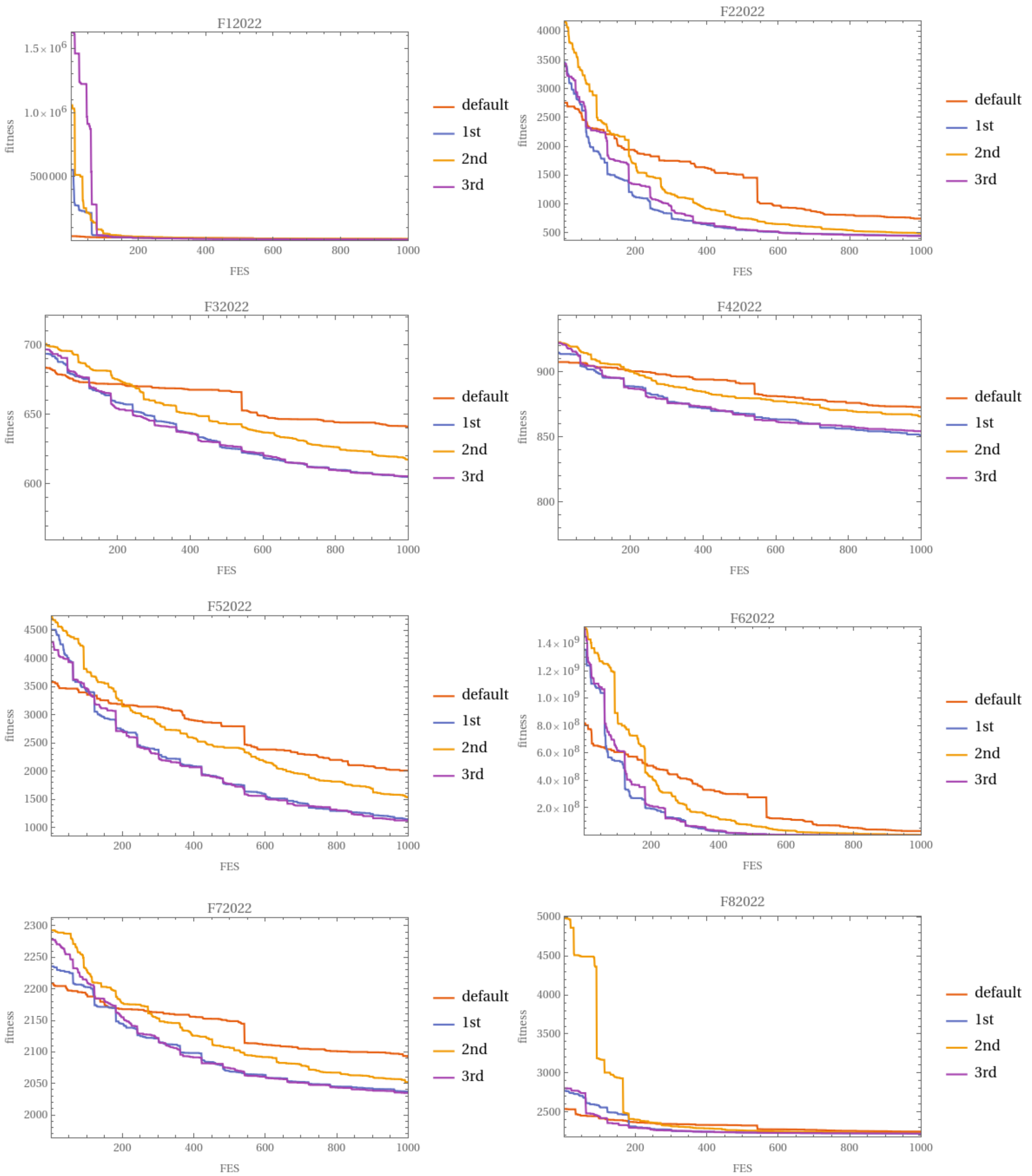
Pairwise comparisons of parameter settings obtained from CEC 2017 test suite against the subsequent setting using the Wilcoxon Rank Sign.

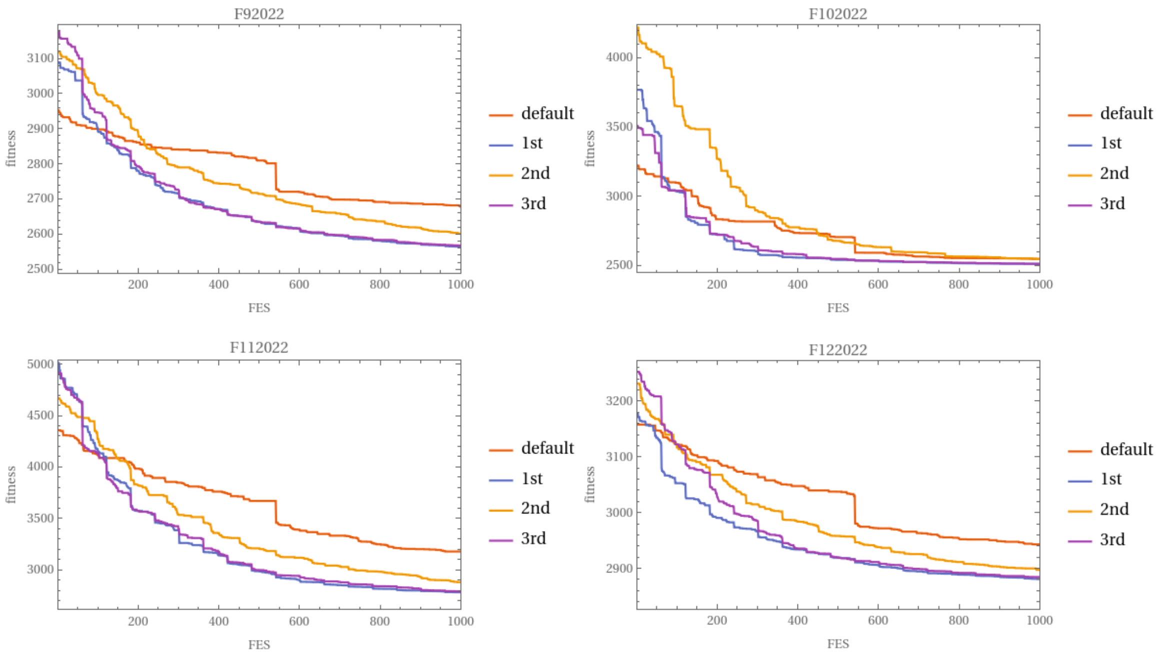
Table 69 Wilcoxon rank sign test for CEC 2022 test suite with parameters obtained racing CEC 2017 test suite

Function	1st	2nd	3rd
1	-	+	+
2	-	+	+
3	-	+	+
4	-	+	+
5	-	+	+
6	+	+	-
7	-	-	+
8	-	+	+
9	-	+	+
10	+	-	+
11	-	+	+
count of (-)	9	2	1
pValues	0.06	0.07	0.05

Graphs of average convergence for each test function with 4 different parameters setting, each convergence was calculated 50 times, for better observability only first 1000 FEs are shown.

Table 70 Graphs of average convergencies





8.3 Summary

There is significant variation in performance across different functions and parameter settings. Some functions (e.g., F12017, F52017, F82017, F272017) consistently report minimal to zero error across different configurations, indicating robustness or perhaps less sensitivity to parameter changes. Conversely, other functions (e.g., F112017, F92017) exhibit high variability in outcomes, suggesting a greater sensitivity to parameter settings.

The iterative refinement of parameter settings through irace seems to have generally improved performance across most functions. This is indicated by decreases in error metrics from the default settings to optimized parameters. However, in some cases, such as F282017, even optimized parameters resulted in substantial errors, indicating room for further improvement or inherent complexity in the function that challenges current optimization strategies.

The application of Friedman mean ranks tests and Wilcoxon rank sum tests provided a structured comparison of parameter settings. These tests often highlighted statistically significant differences between default and optimized settings, reinforcing the value of systematic parameter tuning. Comparisons of parameters optimized on one test suite (e.g., CEC 2017) when applied to another (e.g., CEC 2022) sometimes showed a drop in performance, indicating that optimal parameter settings might be context-dependent and not universally transferable across different test suites.

CONCLUSION

This work presents a detailed exploration and optimization of the Self-Organizing Migrating Algorithm (SOMA) across various variants, utilizing the irace tool for parameter tuning. The thesis is structured into several chapters, each contributing a unique aspect to the overarching goal of enhancing SOMA's effectiveness through sophisticated autoconfiguration techniques.

The introductory chapter sets the stage by outlining the research's objectives, which focus on understanding and optimizing the SOMA algorithm for diverse optimization challenges. The introduction provides a roadmap of the thesis, highlighting the use of irace for parameter tuning and the intent to evaluate these optimizations through rigorous testing on benchmark suites like CEC2017 and CEC2022.

Next chapter delves into the theoretical underpinnings of the SOMA algorithm and its various adaptations. It details the foundational mechanics of SOMA, introduced by Ivan Zelinka in 1999, which mimics social behavior of intelligent creatures in searching and exploiting resources, analogous to finding solutions in an optimization context. The migration mechanism, key to SOMA's operation, involves migrants moving towards a leader based on a perturbation vector that adds randomness, enhancing exploration and preventing premature convergence.

Focusing on selected SOMA variants like SOMA T3A and SOMA Pareto, this next chapter dives deeper into their operational specifics. For SOMA T3A, dynamic adjustments in step size and perturbation vectors are highlighted, showing how these parameters are modulated in response to ongoing performance feedback to optimize the migration path. SOMA Pareto is discussed in the context of its capability to handle multi-objective optimization by adaptively tuning its parameters to maintain a diverse set of Pareto-optimal solutions.

Next chapter justifies the selection of the CEC2017 and CEC2022 benchmark suites for evaluating the performance of the SOMA variants. It explains the importance of these benchmarks in providing a diverse set of optimization problems that help in assessing the algorithms' capabilities to navigate complex problem landscapes effectively.

Second part of this work consist of methodological chapter outlines the workflow for testing and evaluating the optimized SOMA variants. It describes the setup for the irace tool, detailing how it tunes the parameters of SOMA across different test functions. The evaluation criteria based on statistical analysis of performance metrics are also discussed.

The concluding chapters synthesizes the findings from the various tests and analyses, providing insights into the effectiveness of parameter tuning through irace and its impact on the performance of SOMA variants. The tailored migration strategies of SOMA T3A, optimized through irace, demonstrated significant efficiency in navigating the optimization space, leading to faster convergence on optimal solutions while avoiding local optima. The ability of irace to fine-tune parameters based on the stage of function evaluations allowed SOMA Pareto to perform more effectively across a range of benchmark tests, as demonstrated in the evaluations using CEC2017 and CEC2022 suit yet tuned algorithm is not so robust as SOMAT3A. SOMA-CLP, with its sophisticated clustering and population learning capabilities, is highly suited to the adaptive parameter tuning provided by irace. The combination of SOMA-CLP's focused computational approach and irace's efficient parameter optimization enables this variant to handle complex, high-dimensional optimization problems more effectively. This makes SOMA-CLP particularly useful in applications involving large datasets or where the underlying problem structure can be leveraged to enhance search efficiency.

In summary, SOMA-CLP, similar to SOMA T3A and SOMA Pareto, shows significant benefits from the irace tuning process, particularly due to its design that supports dynamic adaptation and learning from population behavior, which are critical in responding to complex optimization challenges.

BIBLIOGRAPHY

[1] Ivan Zelinka. 1999. "SOMA—Self-Organizing Migrating Algorithm." In Proceedings of the 6th International Conference on Soft Computing (Mendel 2000), Brno, Czech Republic.

[2] LÓPEZ-IBÁÑEZ, Manuel; DUBOIS-LACOSTE, Jérémie; PÉREZ CÁCERES, Leslie; BIRATTARI, Mauro a STÜTZLE, Thomas. The irace package: Iterated racing for automatic algorithm configuration. Online. Operations Research Perspectives. 2016, roč. 2016, č. 3, s. 43-58. ISSN 2214-7160.

[3] ZELINKA, Ivan, 2009. Evoluční výpočetní techniky: principy a aplikace. Praha: BEN - technická literatura. ISBN 978-80-7300-218-3.

[4] DAVENDRA, Donald a ZELINKA, Ivan (ed.), 2016. Self-Organizing Migrating Algorithm. Online. Studies in Computational Intelligence. Cham: Springer International Publishing. ISBN 978-3-319-28159-9.

[5] DIEP, Quoc Bao; ZELINKA, Ivan a DAS, Swagatam, 2019. Self-Organizing Migrating Algorithm Pareto. MENDEL. 2019-06-24, roč. 25, č. 1, s. 111-120. ISSN 2571-3701.

[6] Marcela Matusikova, Michal Pluhacek, Tomas Kadavy, Adam Viktorin, Roman Senkerik. "Exploring Adaptive Components of SOMA". Genetic and Evolutionary Computation Conference Companion (GECCO '23 Companion), July 15–19, 2023, Lisbon, Portugal.

[7] Skanderova, L. Self-organizing migrating algorithm: review, improvements and comparison. Artif Intell Rev 56, 101–172 (2023). https://doi.org/10.1007/s10462-022-10167-8

[8] ŠENKEŘÍK, Roman, Tomáš KADAVÝ, Adam VIKTORIN a Michal PLUHÁČEK. Ensemble of strategies and perturbation parameter based SOMA for optimal stabilization of chaotic oscillations. In: GECCO 2020 Companion - Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion [online]. Cancún: Association for Computing Machinery, Inc, 2020, s. 1468-1475.

[9] KADAVY, Tomas; PLUHACEK, Michal; VIKTORIN, Adam a SENKERIK, Roman, 2020. Self-organizing migrating algorithm with clustering-aided migration. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion. New York, NY, USA: ACM, 2020-07-08, s. 1441-1447. ISBN 9781450371278.

[10] DIEP, Quoc Bao, 2019. Self-Organizing Migrating Algorithm Team To Team Adaptive – SOMA T3A. In: 2019 IEEE Congress on Evolutionary Computation (CEC). IEEE, s. 1182-1187. ISBN 978-1-7281-2153-6.

[11] JANÁČEK, Julius. Statistika jednoduše: Průvodce světem statistiky. Grada, 2022. ISBN 978-80-271-1738-3.

[13] ŠENKEŘÍK, Roman, Tomáš KADAVÝ, Adam VIKTORIN a Michal PLUHÁČEK. Ensemble of strategies and perturbation parameter based SOMA for optimal stabilization of chaotic oscillations. In: GECCO 2020 Companion - Proceedings of the 2020 Genetic and Evolutionary Computation Conference

[14] ŠENKEŘÍK, Roman, Tomáš KADAVÝ, Peter JANKŮ, Michal PLUHÁČEK, Hubert GUZOWSKI, Libor PEKAŘ, Radek MATUŠŮ, Adam VIKTORIN, Maciej SMOŁKA, Aleksander BYRSKI a Zuzana KOMÍNKOVÁ OPLATKOVÁ. Maximizing efficiency: A comparative study of SOMA variants and constraint handling methods for time delay system optimization. In: GECCO 2023 Companion - Proceedings of the 2023 Genetic and Evolutionary Computation Conference Companion

[15] KADAVÝ, Tomáš, Michal PLUHÁČEK, Roman ŠENKEŘÍK a Adam VIKTORIN. Introducing self-adaptive parameters to Self-Organizing Migrating Algorithm. In: 2019 IEEE Congress on Evolutionary Computation, CEC 2019 - Proceedings

[20] MALLIPEDDI Rammohan and Ponnuthurai Nagaratnam SUGANTHAN. 2010.

Differential Evolution Algorithm with Ensemble of Parameters and Mutation and

Crossover Strategies. Swarm, Evolutionary and Memetic Computing, 2010

[21] TRUONG, Thanh Cong, Quoc Bao DIEP, Ivan ZELINKA a Roman ŠENKEŘÍK. Pareto-based Self-Organizing Migrating Algorithm solving 100-Digit Challenge. In: Communications in Computer and Information Science [online]. Maribor: Springer, 2020, s. 13-20

[22] "Exploring Adaptive Components of SOMA," Marcela Matusikova, Michal Pluhacek, Tomas Kadavy, Adam Viktorin, Roman Senkerik, Tomas Bata University in Zlin, Zlin, Czech Republic.

[23] DORIGO, Marco. Optimization, Learning and Natural Algorithms. PhD thesis, Politecnico di Milano, Italy, 1992. [cit. 2021-01-29]

[24] D. Maharana, R. Kommadath and P. Kotecha, "Dynamic Yin-Yang Pair Optimization and its performance on single objective real parameter problems of CEC 2017," 2017 IEEE

Congress on Evolutionary Computation (CEC), Donostia, Spain, 2017, pp. 2390-2396, doi: 10.1109/CEC.2017.7969594

[25] Wu, Guohua & Mallipeddi, Rammohan & Suganthan, Ponnuthurai. (2016). Problem Definitions and Evaluation Criteria for the CEC 2017 Competition and Special Session on Constrained Single Objective Real-Parameter Optimization.

[26] Biedrzycki, Rafał & Arabas, J. & Warchulski, Eryk. (2022). A Version of NL-SHADE-RSP Algorithm with Midpoint for CEC 2022 Single Objective Bound Constrained Problems. 1-8. 10.1109/CEC55065.2022.9870220.

[27] López-Ibáñez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Birattari, M., & Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. In Operations Research Perspectives (Vol. 3, pp. 43–58). Elsevier BV.

[28] Birattari, M., Yuan, Z., Balaprakash, P., Stützle, T. (2010). F-Race and Iterated F-Race: An Overview. In: Bartz-Beielstein, T., Chiarandini, M., Paquete, L., Preuss, M. (eds) Experimental Methods for the Analysis of Optimization Algorithms. Springer, Berlin, Heidelberg

[29]Zelinka, Ivan. "SOMA Algorithm Codes." Retrieved from https://ivanzelinka.eu/somaalgorithm/Codes.html

[30]"irace Package User Guide." Retrieved from https://cran.r-project.org/web/packages/irace/vignettes/irace-package.pdf

[31]Tilley, D. "CEC 2017 Python Implementations." Retrieved from https://github.com/tilleyd/cec2017-py

[32]Thieu, "Opfunu Repository." Retrieved from https://github.com/thieu1995/opfunu

[33]Suganthan, P. N. "2022 SO-BO Benchmark Implementations." Retrieved from https://github.com/P-N-Suganthan/2022-SO-BO/tree/main

LIST OF ABBREVIATIONS

ATA	All to all
ATR	All to random
CEC	Congress on Evolutionary Computation
CLP	Clustering and population learning
DP	Differential perturbation
FEs	Function evaluations
PRT	Perturbation
T3A	Team to team adaptive

LIST OF FIGURES

LIST OF TABLES

LIST OF PSEUDOCODES

APPENDICES

PI: Definitions of the basic functions used in CEC 2017

PII: Definitions of the basic functions used in CEC 2022

PIII: DVD with source codes

APPENDIX P I: DEFINITIONS OF THE BASIC FUNCTIONS USED IN CEC 2017

1. Bent Cigar Function

$$f_1(x) = x_1^2 + 10^6 \sum_{i=2}^{D} x_i^2 \tag{1}$$

2. Zakharov Function

$$f_3(x) = \sum_{i=1}^{D} x_i^2 + \left(\sum_{i=1}^{D} 0.5x_i\right)^2 + \left(\sum_{i=1}^{D} 0.5x_i\right)^4 \tag{2}$$

3. Rosenbrock's Function

$$f_4(x) = \sum_{i=1}^{D-1} \left(100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right) \tag{3}$$

4. Rastrigin's Function

$$f_5(x) = \sum_{i=1}^{D} \left(x_i^2 - 10\cos(2\pi x_i) + 10\right) \tag{4}$$

5. Expanded Schaffer's F6 Function

Schaffer's F6 Function: $g(x, y) = 0.5 + \dfrac{\left(\sin^2\left(\sqrt{x^2 + y^2}\right) - 0.5\right)}{\left(1 + 0.001(x^2 + y^2)\right)^2}$

6. Lunacek bi-Rastrigin Function

$$f_7(x) = \min\left(\sum_{i=1}^{D} (x_i - \mu_0)^2, dD + s\sum_{i=1}^{D} (x_i - \mu_1)^2\right) + 10\left(D - \sum_{i=1}^{D} \cos(2\pi z_i)\right)$$

$$\mu_0 = 2.5, \mu_1 = -\sqrt{\frac{\mu_0^2 - d}{s}}, s = 1 - \frac{1}{2\sqrt{D + 20} - 8.2}, d = 1$$

$$y = \frac{10(\boldsymbol{x} - \boldsymbol{o})}{100}, x_i = 2\text{sign}(x_i^*)y_i + \mu_0, \text{ for } i = 1,2,\ldots,D$$

7. Non-continuous Rotated Rastrigin's Function

$$f_8(x) = \sum_{i=1}^{D} (z_i^2 - 10\cos(2\pi z_i) + 10) + f_{13}*$$
$$\hat{x} = \mathbf{M}_1 \frac{5.12(x-o)}{100}, y_i = \begin{cases} \hat{x}_i & \text{if } |\hat{x}_i| \le 0.5 \\ \text{round}(2x_i)/2 & \text{if } |\hat{x}_i| > 0.5 \end{cases} \text{ for } i = 1,2,\dots,D$$
$$z = \mathbf{M}_1 \Lambda^{10} \mathbf{M}_2 T_{asy}^{0.2}(T_{osz}(y))$$

Where Λ^{α} : a diagonal matrix in D dimensions with the i^{th} diagonal element as $\lambda_{ii} = \alpha^{\frac{i-1}{2(D-1)}}$, $i = 1,2,\dots,D$.

T_{asy}^{β} : if $x_i > 0, x_i = x_i^{1+\beta\frac{i-1}{D-1}\sqrt{x_i}}$, for $i = 1,\dots,D$[4]

$T_{\text{osz:}}$ for $x_i = \text{sign}(x_i)\exp\left(\hat{x}_i + 0.049\left(\sin(c_1\hat{x}_i) + \sin(c_2\hat{x}_i)\right)\right)$, for $i = 1$ and D[4]

$$\text{where } \hat{x}_i = \begin{cases} \log(|x_i|) & \text{if } x_i \ne 0 \\ 0 & \text{otherwise} \end{cases}, \text{sign}(x_i) = \begin{cases} -1 & \text{if } x_i < 0 \\ 0 & \text{if } x_i = 0 \\ 1 & \text{otherwise} \end{cases}$$
$$c_1 = \begin{cases} 10 & \text{if } x_i > 0 \\ 5.5 & \text{otherwise} \end{cases}, \text{ and } c_2 = \begin{cases} 7.9 & \text{if } x_i > 0 \\ 3.1 & \text{otherwise} \end{cases}$$

8. Levy Function

$$f_9(\boldsymbol{x}) = \sin^2(\pi w_1) + \sum_{i=1}^{D-1} (w_i - 1)^2[1 + 10\sin^2(\pi w_i + 1)] + (w_D - 1)^2[1 + \sin^2(2\pi w_D)]$$

9. Modified Schwefel's Function

$$f_{10}(x) = 418.9829 \times D - \sum_{i=1}^{D} g(z_i),$$

10. High Conditioned Elliptic Function

$$f_{11}(x) = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} \boldsymbol{x}_i^2 \tag{10}$$

11. Discus Function

$$f_{12}(\boldsymbol{x}) = 10^6 x_1^2 + \sum_{i=2}^{D} x_i^2 \tag{11}$$

12. Ackley's Function

$$f_{13}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D} \cos(2\pi x_i)\right) + 20 + \mathrm{e} \tag{12}$$

13. Weierstrass Function

$$f_{14}(x) = \sum_{i=1}^{D}\left(\sum_{k=0}^{k\max}\left[a^k \cos\left(2\pi b^k (x_i + 0.5)\right)\right]\right) - D\sum_{k=0}^{k\max}\left[a^k \cos(2\pi b^k \cdot 0.5)\right] \quad (13)$$

$$a = 0.5, b = 3, k\max = 20 \quad (13)$$

14. Griewank's Function

$$f_{15}(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{14}$$

15. Katsuura Function

$$f_{16}(x) = \frac{10}{D^2}\prod_{i=1}^{D}\left(1 + i\sum_{j=1}^{32} \frac{\left|2^j x_i - \mathrm{round}\left(2^j x_i\right)\right|}{2^j}\right)^{\frac{10}{D^{12}}} - \frac{10}{D^2} \tag{15}$$

16. HappyCat Function

$$f_{17}(\boldsymbol{x}) = \left|\sum_{i=1}^{D} x_i^2 - D\right|^{1/4} + \left(0.5\sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i\right)/D + 0.5 \tag{16}$$

17. HGBat Function

$$f_{18}(x) = \left|\left(\sum_{i=1}^{D} x_i^2\right)^2 - \left(\sum_{i=1}^{D} x_i\right)^2\right|^{1/2} + \left(0.5\sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i\right)/D + 0.5 \tag{17}$$

18. Expanded Griewank's plus Rosenbrock's Function

$$f_{19}(x) = f_7\big(f_4(x_1, x_2)\big) + f_7\big(f_4(x_2, x_3)\big) + \cdots + f_7\big(f_4(x_{D-1}, x_D)\big) + f_7\big(f_4(x_D, x_1)\big) \quad (18)$$

19. Schaffer's F7 Function

$$f_{20}(x) = \left[\frac{1}{D-1}\sum_{i=1}^{D-1}\left(\sqrt{s_i}\cdot(\sin(50.0 s_i^{0.2})+1)\right)\right]^2, s_i = \sqrt{x_i^2 + x_{i+1}^2} \qquad (19)$$

APPENDIX P II: DEFINITIONS OF THE BASIC FUNCTIONS USED IN CEC 2022

1. Zakharov Function

$$f_1(\mathbf{x}) = \sum_{i=1}^{D} x_i^2 + \left(\sum_{i=1}^{D} 0.5x_i\right)^2 + \left(\sum_{i=1}^{D} 0.5x_i\right)^4 \tag{1}$$

2. Rosenbrock's Function

$$f_2(\mathbf{x}) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_{i+1} - 1)^2) \tag{2}$$

3. Expanded Schaffer's Function

Schaffer's Function: $g(x, y) = 0.5 + \frac{\left(\sin^2\left(\sqrt{x^2 + y^2}\right) - 0.5\right)}{\left(1 + 0.001(x^2 + y^2)\right)^2}$

4. Rastrigin's Function

$$f_4(\mathbf{x}) = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10) \tag{4}$$

5. Levy Function

$$f_5(\mathbf{x}) = \sin^2(\pi w_1) + \sum_{i=1}^{D-1} (w_i - 1)^2[1 + 10\sin^2(\pi w_i - 1)] + (w_D - 1)^2[1 + \sin^2(2\pi w_D)]$$

6. Bent Cigar Function

$$f_6(\mathbf{x}) = x_i^2 + 10^{6\sum_{i=2}^{D} x_i^2} \tag{6}$$

7. HGBat Function

$$f_7(\mathbf{x}) = \left|\left(\sum_{i=1}^{D} x_i^2\right)^2 - \left(\sum_{i=1}^{D} x_i\right)^2\right|^{0.5} + \left(0.5\sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i\right)/\mathrm{D} + 0.5 \tag{7}$$

8. High Conditioned Elliptic Function

$$f_8(\mathbf{x}) = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} x_i^2 \tag{8}$$

9. Katsuura Function

$$f_9(\mathbf{x}) = \frac{10}{D^2}\prod_{i=1}^{D}\left(1 + i\sum_{j=1}^{32}\frac{\left|2^j x_i - \mathrm{round}\left(2^j x_i\right)\right|}{2^j}\right)^{\frac{10}{D^{1.2}}} - \frac{10}{D^2} \tag{9}$$

10. Happycat Function

$$f_{10}(\mathbf{x}) = \left|\sum_{i=1}^{D} x_i^2 - D\right|^{1/4} + \left(0.5\sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i\right)/\mathrm{D} + 0.5 \tag{10}$$

11. Expanded Rosenbrock's plus Griewangk's Function

$$f_{11}(\mathbf{x}) = f_{15}\big(f_2(x_1, x_2)\big) + f_{15}\big(f_2(x_2, x_3)\big) + \cdots + f_{15}\big(f_2(x_{D-1}, x_D)\big) + f_{15}\big(f_2(x_D, x_1)\big) \tag{11}$$

12. Modified Schwefel's Function

$$f_{12}(\mathbf{x}) = 418.9829 \times D - \sum_{i=1}^{D} g(z_i) \tag{12}$$

$$z_i = x_i + 4.209687462275036\mathrm{E} + 002 \tag{12}$$

$$g(z_i) = \begin{cases} z_i \sin\left(|z_i|^{1/2}\right), \text{ if } |z_i| \le 500 \\ \big(500 - \mathrm{mod}(z_i, 500)\big)\sin\left(\sqrt{|500 - \mathrm{mod}(z_i, 500)|}\right) - \dfrac{(z_i - 500)^2}{10000D}, \text{ if } z_i > 500 \\ (\mathrm{mod}(|z_i|, 500) - 500)\sin\left(\sqrt{|\mathrm{mod}(|z_i|, 500) - 500|}\right) - \dfrac{(z_i + 500)^2}{10000D}, \text{ if } z_i < -500 \end{cases}$$

13. Ackley's Function

$$f_{13}(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos\left(2\pi x_i\right)\right) + 20 + e \tag{13}$$

14. Discus Function

$$f_{14}(\mathbf{x}) = 10^6 x_1^2 + \sum_{i=2}^{D} x_i^2 \tag{14}$$

15. Griewank's Function

$$f_{15}(\mathbf{x}) = \sum_{i=1}^{D}\frac{x_i^2}{4000} - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{15}$$

16. Schaffer's F7 Function

$$f_{16}(\mathbf{x}) = \left[\frac{1}{D-1}\sum_{\mathrm{i}=1}^{\mathrm{D}-1}\left(\sqrt{\mathrm{s}_\mathrm{i}}\cdot(\sin(50.0\ \mathrm{s}_\mathrm{i}^{0.2})+1)\right)\right]^2, s_i = \sqrt{x_i^2 + x_{i+1}^2} \qquad (16)$$