Asia Pac. J. Math. 2026 **13**:30

A MULTI-CRITERIA ANALYSIS OF THE USE OF SOME AGGREGATION FUNCTIONS IN THE CUCKOO SEARCH ALGORITHM FOR MULTI-OBJECTIVE OPTIMIZATION

HERVE WENDPAGNAGDE OUEDRAOGO[1,*], ABDOUL RAZAKOU NAGALO[2], YOUSSOUF OUEDRAOGO[2], ABDOULAYE COMPAORE[2]

[1]Department of Mathematics, Ecole Normale Supérieure, Koudougou, Burkina Faso

[2]Department of Mathematics, Université Norbert Zongo, Koudougou, Burkina Faso

[*]Corresponding author: herveoueder@yahoo.fr

Received Nov. 12, 2025

Abstract. In this article, we propose a multi-objective version of the cuckoo search algorithm through an approach for solving multi-objective optimization problems with constraints. We also investigate the impact of various aggregation functions in this new version. Specifically, we use the weighted sum, the weighted Chebyshev distance, the weighted augmented Chebyshev distance, and the ε-constraint technique to transform the multi-objective optimization problem into a single-objective optimization problem. Next, we apply a penalization technique derived from the Lagrangian to convert the constrained single-objective problem into an unconstrained one. The resulting single-objective problem is then solved using the cuckoo search algorithm, and the Pareto-optimal solutions are obtained by configuring the optimal solutions of the unconstrained single-objective problem. Finally, we conduct a comparative study of the use of each aggregation function within the method, based on an analysis of the profiles of alternatives and the GAIA plane. This comparison allows us to identify the best approaches for aggregating objective functions based on preferences. This study, conducted on convex or concave problems with one or more variables, provides guidance for researchers on the use of aggregation functions in a multi-objective cuckoo search algorithm.

2020 Mathematics Subject Classification. 65K05; 65K10; 90C29; 90C70.

Key words and phrases. metaheuristics; convergence metrics; profile analysis; GAIA plan.

1. Introduction

The cuckoo search algorithm, inspired by the cuckoo's breeding strategy, is a powerful metaheuristic based on swarm intelligence, proposed by Yang and Deb [1] for solving single-objective optimization problems. After its introduction, many researchers became interested in this algorithm, and several of its variants were proposed.

Thus, in the case of single-objective problems, we have the modified cuckoo search algorithms, which attempt to improve the convergence of the original cuckoo search algorithm [2–7]. For solving

DOI: 10.28924/APJM/13-30

combinatorial optimization problems via the cuckoo search algorithm, proposals have been made in [8–10].

In the multi-objective case, we highlight the multi-objective cuckoo search proposed by Xin-She Yang and Suash Deb [11], which involves considering more than one egg per nest. In [12], a multi-objective version of the cuckoo search algorithm was proposed using archives. These archives store the best solutions from each generation, which are then improved in subsequent generations. We also mention the hybrid multi-objective cuckoo search algorithm with dynamic local search [13], which employs the non-dominated sorting procedure and a dynamic local search. The former is useful for generating Pareto fronts, while the latter focuses on improving local search. For solving multi-objective scheduling problems, several algorithms based on the cuckoo search were proposed in [14–16].

Despite the variety of metaheuristics, the famous No-Free-Lunch Theorem [17] proves that no optimization algorithm can efficiently solve all optimization problems. Thus, research is still ongoing in this area, and new metaheuristics continue to be proposed. However, to date, there is no multi-objective version of the cuckoo search algorithm based on solution approaches such as the weighted sum, weighted Chebyshev distance, weighted augmented Chebyshev distance, and ε-constraint technique. There is also no study on the impact of these solution approaches on a multi-objective version of the cuckoo search algorithm. To contribute to this area, we will first propose a multi-objective version of the cuckoo search algorithm using these solution approaches and a penalization function derived from the Lagrangian. This allows us to convert the multi-objective problem into a single-objective problem without constraints. Then, we will conduct a comparative analysis to identify the best solution approach.

To better present our work, we will subdivide it into five sections. In Section 2, we will discuss the basic concepts, namely multi-objective optimization techniques, performance measures of a metaheuristic, multi-criteria analysis methods, and the single-objective cuckoo search algorithm. In Section 3, we will introduce a new multi-objective extension of the cuckoo search algorithm. This section presents our algorithm and its principle, along with numerical results on test problems. A performance study is provided in Section 4. Finally, we will conclude this work in Section 5.

2. Basic concepts

2.1. **Multiobjective optimization techniques.** Consider a multi-objective optimization problem formulated as follows:

$$\begin{cases} \min\left\{F(x)=\Big(f_1(x),f_2(x),\cdots,f_p(x)\Big)^T,p\geq 2\right\}, \\ g_k(x)\leq 0, \quad k=1,\cdots,m \\ x\in\mathbb{R}^n, \end{cases} \tag{1}$$

where: $x \in \mathbb{R}^n$, $F = \Big(f_1, f_2, \cdots, f_p\Big)^T : \mathbb{R}^n \to \mathbb{R}^p$ and $g = (g_1, \cdots, g_m)^T : \mathbb{R}^n \to \mathbb{R}^m$.

In order to transform the problem (1) into a single-objective optimization problem, we will use the following penalty functions:

i.: the weighted sum defined by:

$$S_1(f, \omega) = \sum_{i=1}^{p} \omega_i f_i, \tag{2}$$

with $\omega = (\omega_i)_{i=\overline{1,p}} \in [0,1]^p$ such that $\sum \omega_i = 1$.

ii.: the weighted Chebyshev distance defined by:

$$S_2(f, \omega) = \max_{k=\overline{1,p}} \Big\{\omega_k |f_k - \overline{z}_k|\Big\}, \tag{3}$$

where $\overline{z} = (\overline{z}_1, ..., \overline{z}_p)$ is an ideal point and $\omega = (\omega_k)_{k=\overline{1,p}} \in [0,1]^p$ such that $\sum \omega_k = 1$.

iii.: the weighted augmented Chebyshev distance defined by:

$$S_3(f, \omega) = \max_{k=\overline{1,p}} \Big\{\omega_k |f_k - \overline{z}_k|\Big\} + \mathfrak{u} \sum_{k=1}^{p} |f_k - \overline{z}_k|, \tag{4}$$

where $\overline{z} = (\overline{z}_1, ..., \overline{z}_p)$ is an ideal point, $\mathfrak{u}$ is a penalty parameter, and $\omega = (\omega_k)_{k=\overline{1,p}} \in [0,1]^p$ such that $\sum \omega_k = 1$.

iv.: the ϵ-constraint technique:

This approach consists of [18]:

- Choose an objective function $f_k(x)$ to optimize as a priority;
- Choose the value of

$$\epsilon_j \in [\min f_j(x); \max f_j(x)]\,, \quad j \neq k;$$

- Transform the other objectives into inequality constraints $(f_j(x) \leq \epsilon_j, \quad j \neq k)$.

Let us denote this transformation as S_4 and define it by:

$$S_4(f, \epsilon) = f_k. \tag{5}$$

We will also use a penalization function to convert the problem (1) into an unconstrained optimization problem. This function is given by the relation:

$$L(x, \zeta) = S_i(f(x), \zeta) + \mathfrak{h} \sum_{j=1}^{m} \Big(g_j(x) - |g_j(x)|\Big), \tag{6}$$

where S_i represents one of the expressions (2), (3), (4), or (5), $\zeta \in \{\omega, \epsilon\}$, and $\mathfrak{h}$ is a penalty parameter.

2.2. **Performance measures of a metaheuristic.** The performance evaluation metrics of a metaheuristic that we will use are as follows:

i.: Convergence metric:

It characterizes the proximity of the solutions provided to the Pareto front and is defined in [19] by the relation:

$$\varUpsilon = \frac{\sum_{i=1}^{N} d_i}{N}, \tag{7}$$

where N is the number of non-dominated solutions obtained with the algorithm, and d_i is the Euclidean distance between each non-dominated solution and the solution closest to the Pareto front. The closer the value of $\varUpsilon$ is to 0, the more convergent the algorithm.

ii.: Δ-spread:

This metric evaluates the distribution of solutions on the Pareto front. A good distribution requires a uniform spacing between consecutive solutions on the front. For a problem with two objectives, this metric is calculated as follows [20]:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1}|d_i - \overline{d}|}{d_f + d_l + (N-1)\overline{d}}, \tag{8}$$

where d_i is the Euclidean distance between two consecutive solutions, N is the number of non-dominated solutions, d_f and d_l are the Euclidean distances between the extreme solutions and the limit solutions of the non-dominated set, and $\overline{d}$ is the average value of these distances. The closer the value of Δ is to 0, the better the distribution of solutions.

iii.: $\varGamma$-spread:

This metric measures the maximum distance between solutions on the approximate Pareto front. Denoting by $\mathfrak{P}$ the set of problems and by $\mathfrak{S}$ the set of methods, for a problem $\mathfrak{r} \in \mathfrak{P}$ and a method $\mathfrak{s} \in \mathfrak{S}$, this metric is defined as follows [21]:

$$\varGamma_{\mathfrak{t},\mathfrak{s}} = \max_{j\in\{1,\dots,m\}} \left(\max_{i\in\{1,\dots,N\}} \varphi_{i,j} \right), \tag{9}$$

where $\varphi_{i,j} = \Big(f_j(x_{i+1}) - f_j(x_i) \Big)$.

iv.: Purity metric:

This metric measures the quality of a Pareto front calculated by a method, based on its ability to generate non-dominated points. Let $F_{\mathfrak{t},\mathfrak{s}}$ be the approximation of the Pareto front for a problem $\mathfrak{t} \in \mathfrak{P}$ calculated by method $\mathfrak{s} \in \mathfrak{S}$. Let $F_{\mathfrak{t}}$ be an approximation of the true Pareto front of problem $\mathfrak{t}$, obtained by $\bigcup_{\mathfrak{s}\in\mathfrak{S}} F_{\mathfrak{t},\mathfrak{s}}$, after removing all dominated points. This metric is defined in [21] as:

$$\mathfrak{g}_{\mathfrak{t},\mathfrak{s}} = \frac{|F_{\mathfrak{t},\mathfrak{s}} \cap F_{\mathfrak{t}}|}{|F_{\mathfrak{t},\mathfrak{s}}|}. \tag{10}$$

2.3. **Multi-criteria analysis method.** Two parameters of the PROMETHEE method will mainly be used in the remainder of our work:

i.: The profile of alternatives: The profile of an alternative is composed of the set of net flows for each criterion $\phi_j(a)$ where $j = \{1, ..., p\}$. It is particularly useful for assessing the quality of alternatives based on different criteria and is presented in the form of a histogram. It is used to describe the performance of each alternative in terms of the various criteria.

ii.: The GAIA plan (Geometrical Analysis for Interactive Assistance): This method graphically describes the main features of decision problems. Its analysis is based on single-criterion net flows and can be interpreted as follows:

- Alternatives with similar characteristics are represented by points close to each other;
- Strongly dissimilar alternatives are represented by distant points;
- Criteria expressing similar preferences are represented by axes pointing in similar directions;
- Conflicting criteria are represented by opposing axes;
- The longer the axis of a criterion in the GAIA plane, the more discriminating the criterion is;
- Criteria with the best performance for a particular alternative are located in the direction indicated by the axis corresponding to that alternative [22].

2.4. **Cuckoo Search Algorithm.**

2.4.1. *Principle.* The principle of the cuckoo search method is to generate new, better solutions to replace worse solutions in the nest by following the three basic rules proposed by Yang and Deb [1]:

i) Each cuckoo lays one egg at a time and places it in a randomly chosen nest. New solutions are generated by means of a local random walk, where the step length is drawn from a Lévy distribution.

ii) High-quality eggs are carried over to the next iteration, ensuring the survival of the fittest.

iii) The egg laid by a cuckoo is discovered by the host bird with a probability $p_a \in [0, 1]$. In this case, the host bird may either throw the egg out of its nest or abandon the nest and build a new one elsewhere. This diversification ensures that the algorithm does not get trapped in a local minimum.

2.4.2. *Algorithm.* Based on the three rules, the cuckoo search algorithm for the optimization of a single-objective function ρ without constraints is summarized in the following pseudo-code [1]:

Algorithm 1 Cuckoo Search Algorithm

Require:

Input $\rho(x)$, $x = (x_1, ..., x_d)^T$

Generate an initial population of n host nests x_i $(i = 1, 2, ..., n)$

while ($t <$ MaxGeneration) or (stopping criterion) **do**

  Get a random cuckoo by Lévy flights

    Evaluate its quality/fitness F_i

  Choose a nest at random from n (i.e., j)

  if $(F_i > F_j)$ **then**

    Replace j with the new solution

  end if

  A fraction p_a of the worst nests are abandoned and new ones built

  Keep the best solutions (or nests with high-quality solutions)

  Rank the solutions and identify the current best solution

end while

Post-processing results and visualization

Ensure:

3. New Extension of the Cuckoo Search Algorithm for Multiobjective Optimization

3.1. **Principle.** The principle of this variant consists of, for a given multiobjective optimization problem, first transforming it into a single-objective problem using one of the following resolution approaches: Weighted sum, Chebyshev weighted distance, Chebyshev augmented weighted distance, or the ϵ-constraint technique. Then, a penalization technique is used to penalize constraints, if any exist. At this stage, we obtain a single-objective problem without constraints. Finally, we apply the CS algorithm to solve this unconstrained single-objective problem.

3.2. **Algorithm.** Following the principle described above, we propose the following algorithm:

Algorithm 2 Hybridization of the Cuckoo Search Algorithm

Require:

Input $F(x) = \Big(f_1(x), f_2(x), \cdots, f_p(x)\Big)^T$

Input $g(x) = \Big(g_1(x), \cdots, g_m(x)\Big)^T$

Generate ζ elements

Calculate $S_i(f(x), \zeta)$ using a resolution approach

Calculate $L(x, \zeta)$

Execute **Algorithm 1** replacing ρ with L

Display Pareto optimal solutions $x^*_\zeta = \arg\min(L(x, \zeta))$

Plot the Pareto front

Ensure:

3.3. **Numerical Tests.** All simulations in this section will be conducted with the test problems listed below. With the exception of the ZD and ZT problems, which are concave, all other problems in the table below are convex.

TABLE 1. List of test problems used

Test problems	n
ZDT1 $\begin{cases} \min\ f_1(x) = x_1 \\ \min\ f_2(x) = g(x)(1 - \sqrt{\frac{f(x)}{g(x)}}) \\ g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \\ x = (x_1, x_2, ..., x_n) \in [0,1]^n \end{cases}$	30
ZDT2 $\begin{cases} \min\ f_1(x) = x_1 \\ \min\ f_2(x) = g(x).(1 - (\frac{f_1(x)}{g(x)})^2) \\ g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \\ x = (x_1, x_2, ..., x_n) \in [0,1]^n \end{cases}$	30
ZDT3 $\begin{cases} \min\ f_1(x) = 1 - \exp(-4x_1)\sin^6(6\pi x_1) \\ \min\ f_2(x) = g(x)(1 - (\frac{f_1(x)}{g(x)})^2) \\ g(x) = 1 + 9\left[\frac{\sum_{i=2}^{10} x_i}{9}\right]^{0.25} \\ 0 \leq x_i \leq 1, \quad 0 \leq i \leq 10 \end{cases}$	10
MinEx $\begin{cases} \min\ f_1(x) = x_1 \\ \min\ f_2(x) = \dfrac{1 + x_2}{x_1} \\ x_1 \in [0.1, 1]\ et\ x_2 \in [0, 5] \end{cases}$	2
SCH $\begin{cases} \min\ f_1(x) = x^2 \\ \min\ f_2(x) = (x-2)^2 \\ x \in [-5, 5] \end{cases}$	1
CosH $\begin{cases} \min\ f_1(x) = cosh(x) \\ \min\ f_2(x) = x^2 - 12x + 35 \\ x \in [0, 5] \end{cases}$	1

We will present the different Pareto fronts obtained by each approach, and finally, we will perform metric calculations. These metrics will mainly focus on convergence, Δ-spread, Γ-spread, and Purity.

To collect this data, due to the stochastic nature of the method, we took the minimum results from five (05) runs for each problem.

3.3.1. *Results for S_1.*

i.: Representation of Pareto fronts:

Numerical simulations with S_1 give the following Pareto fronts:

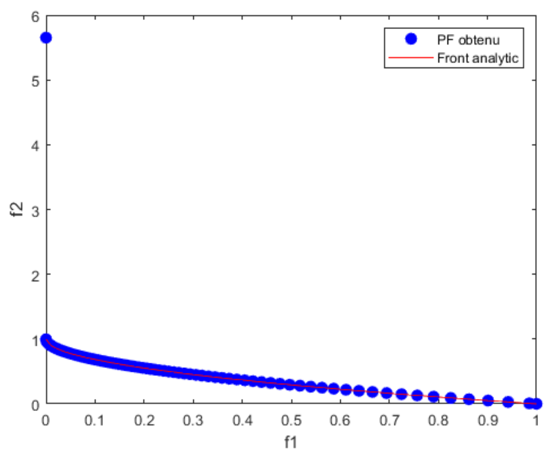

FIGURE 1. ZDT1

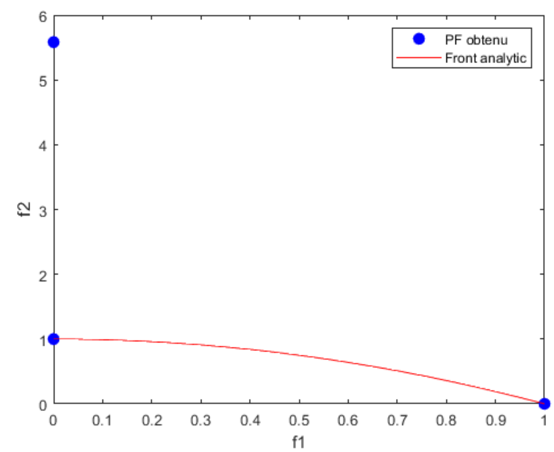

FIGURE 2. ZDT2

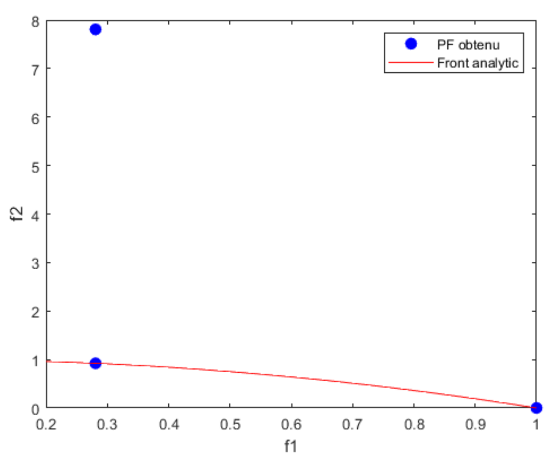

FIGURE 3. ZDT6

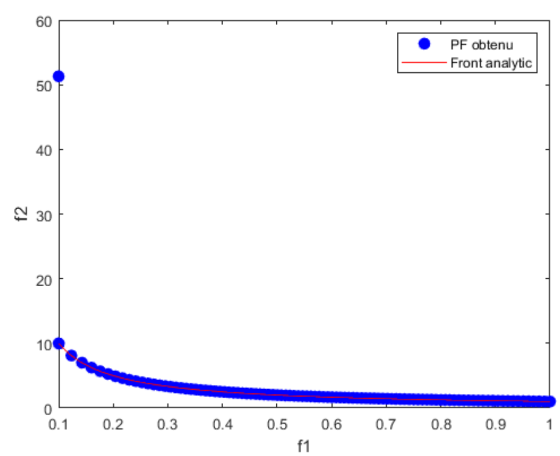

FIGURE 4. MinEx

ii.: Metric calculation results:

Calculating the metrics with S_1 provides us with the following table:

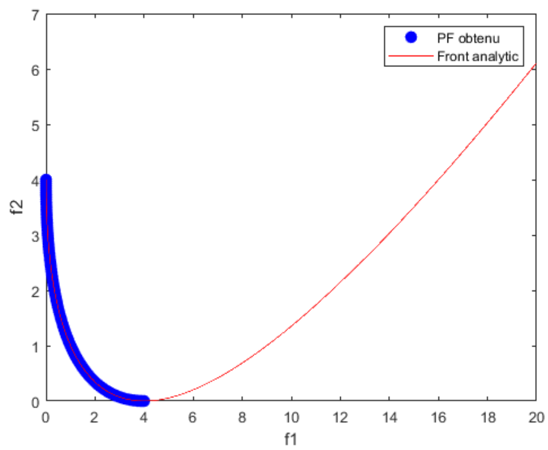

FIGURE 5. SCH

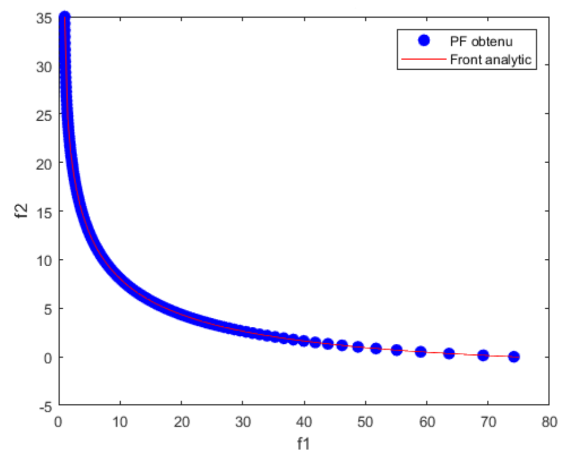

FIGURE 6. CosH

TABLE 2. S_1 performance

Metrics \ Problems	ZDT1	ZDT2	ZDT6	MinEx	SCH	CosH
Convergence	0.0205	0.0215	0.0371	0.0117	0.0004	0.0020
$\Delta-$spread	0.0212	0.0231	0.0422	0.2176	0.0019	0.0717
$\Gamma-$spread	4.1075	4.3297	7.1552	21.0855	0.0399	5.5653
Purity	0.7960	0.3134	0.9851	0.9950	0.9950	0.9950

3.3.2. *Results for S_2.*

i.: Pareto front representation:

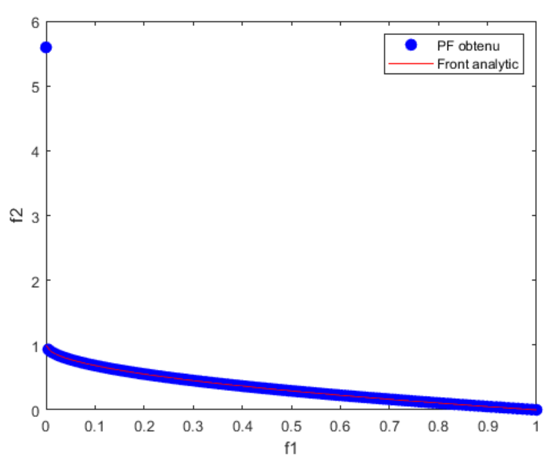

FIGURE 7. ZDT1

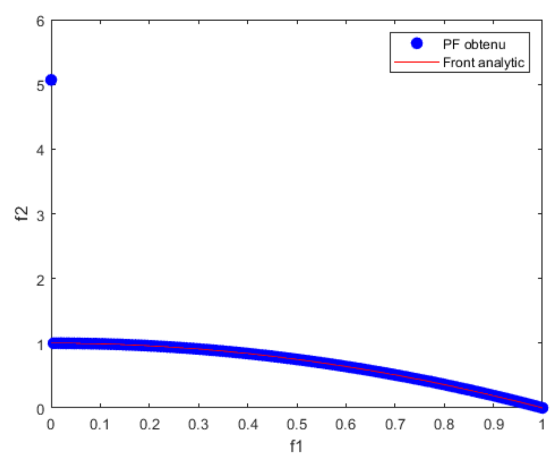

FIGURE 8. ZDT2

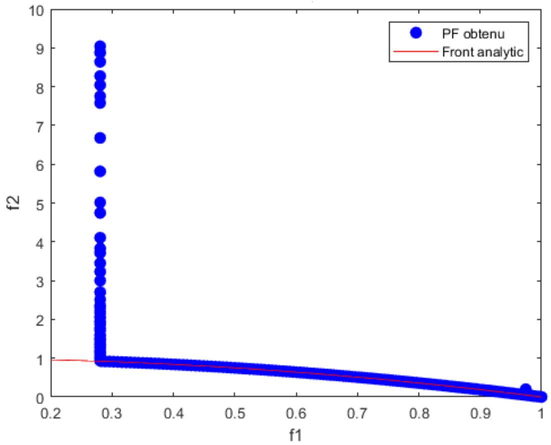

FIGURE 9. ZDT6

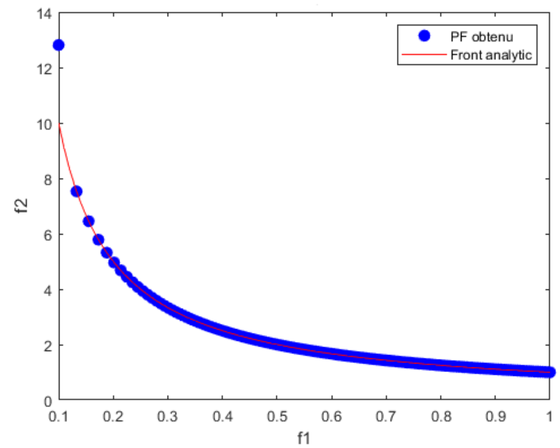

FIGURE 10. MinEx

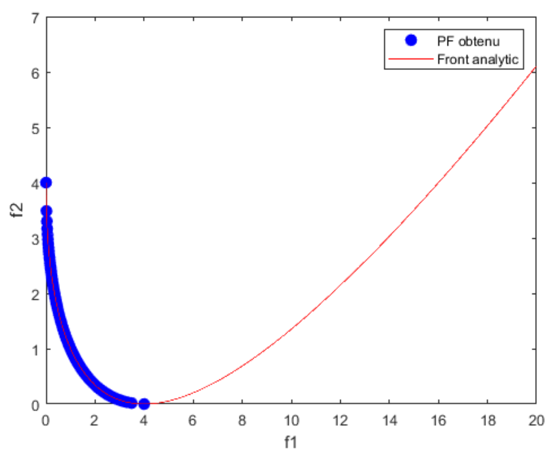

FIGURE 11. SCH

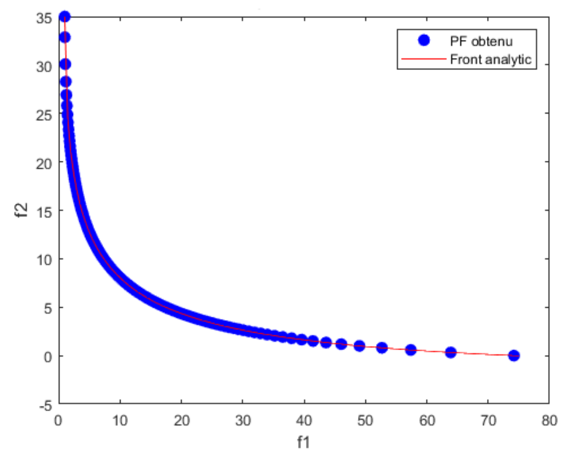

FIGURE 12. CosH

ii.: Results of metrics calculation:

Calculating the metrics with S_2 gives us the table below:

TABLE 3. S_2 performance

Problems / Metrics	ZDT1	ZDT2	ZDT6	MinEx	SCH	CosH
Convergence	0.0209	0.0171	0.1245	0.0054	0.0001	0.0012
$\Delta-$spread	0.0206	0.0212	0.1225	0.0766	0.0004	0.0418
$\Gamma-$spread	4.1992	3.4268	1.9280	9.7909	0.5120	10.2987
Purity	0.9751	0.9751	0.7662	0.9950	0.9950	1

3.3.3. *Results for S_3.*

i.: Pareto front representation:

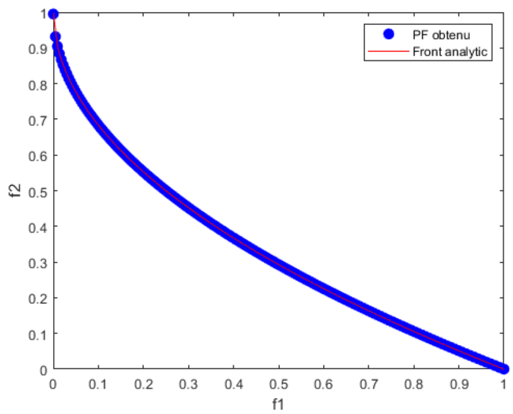

FIGURE 13. ZDT1

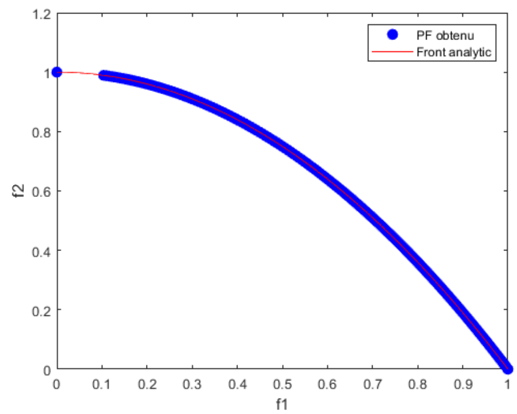

FIGURE 14. ZDT2

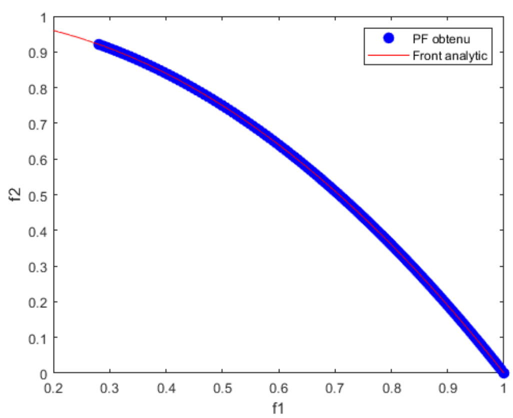

FIGURE 15. ZDT6

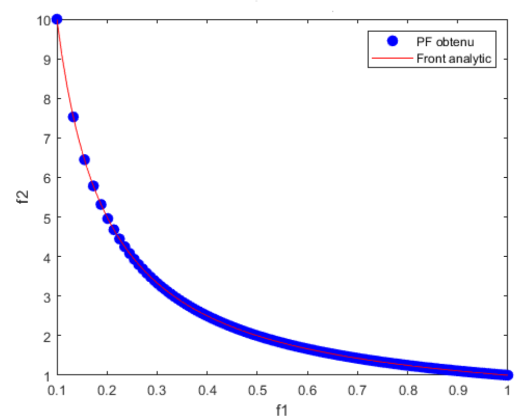

FIGURE 16. MinEx

ii.: Metrics calculation results:

Calculating the metrics with S_3 provides us with the following table:

TABLE 4. Performance de S_3

Problems / Metrics	ZDT1	ZDT2	ZDT6	MinEx	SCH	CosH
Convergence	0.0002	0.0002	0.0003	0.0002	0.0000	0.0013
$\Delta-$spread	0.0003	0.0003	0.0003	0.0019	0.0005	0.0451
$\Gamma-$spread	0.0198	0.3221	0.0808	1.8464	0.4340	10.2987
Purity	1	0.9801	1	1	1	1

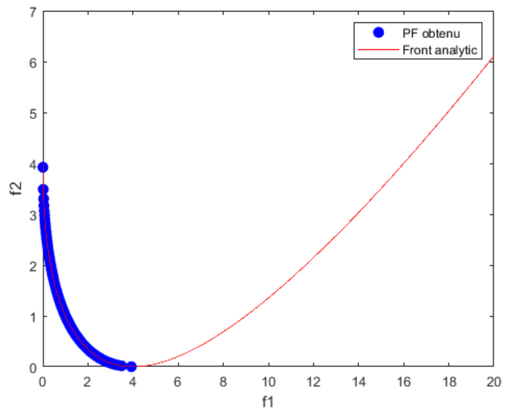

FIGURE 17. SCH

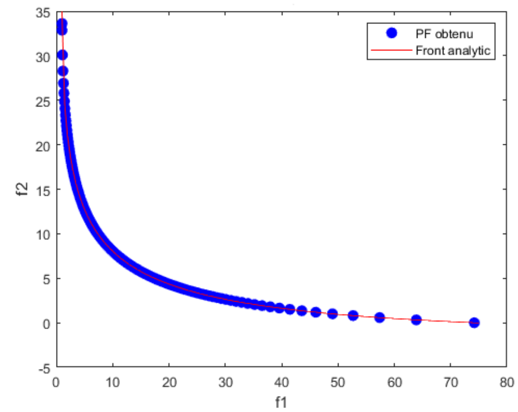

FIGURE 18. CosH

3.3.4. *Results for S_4.*

i.: Representation of the Pareto fronts:

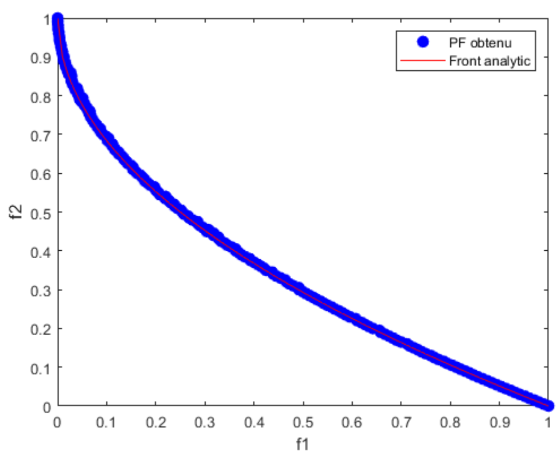

FIGURE 19. ZDT1

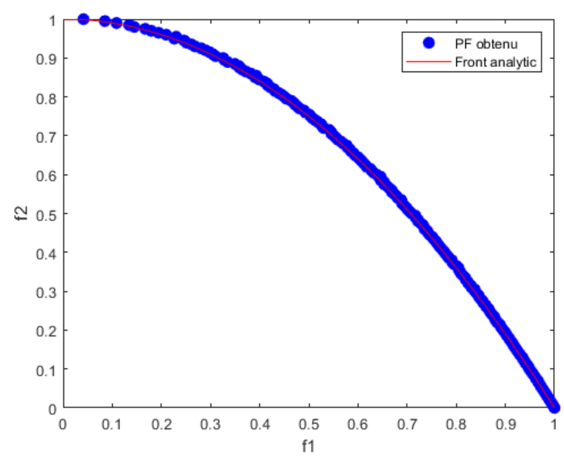

FIGURE 20. ZDT2

ii.: Metric calculation results:

Calculating the metrics with S_4 gives us the table below:

TABLE 5. S_4 performance

Metriques \ Problems	ZDT1	ZDT2	ZDT6	MinEx	SCH	CosH
Convergence	0.0005	0.0003	0.0008	0.0010	0.0004	0.0022
$\Delta-$spread	0.0007	0.0004	0.0008	0.0090	0.0019	0.0777
$\Gamma-$spread	0.0137	0.0530	0.0808	0.0400	0.3896	2.5098
Purity	0.7016	0.7114	0.9204	1	0.9875	1

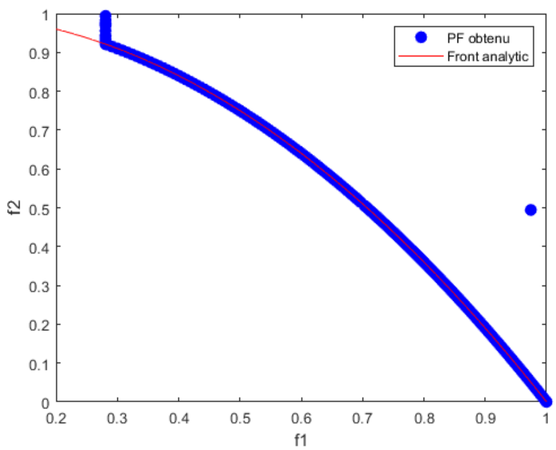

FIGURE 21. ZDT6

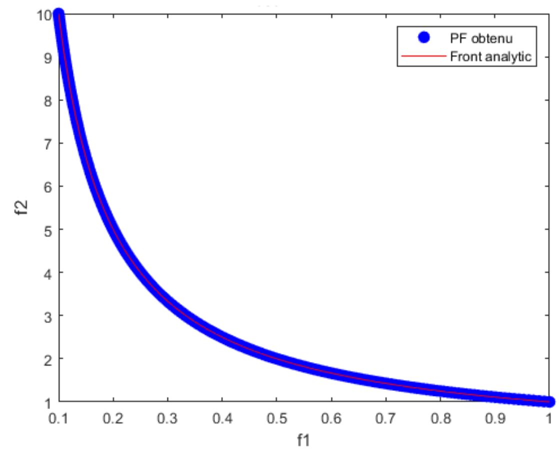

FIGURE 22. MinEx

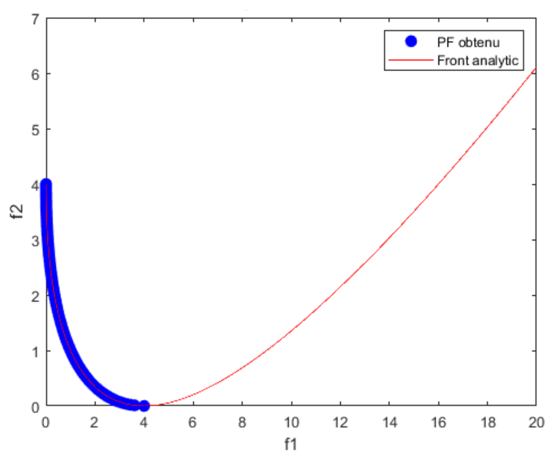

FIGURE 23. SCH

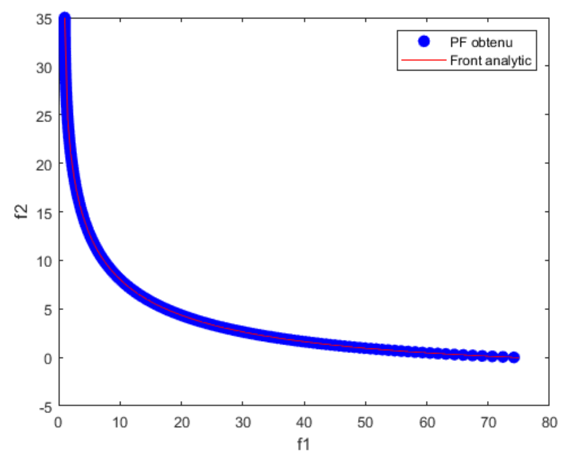

FIGURE 24. CosH

4. PERFORMANCE ANALYSIS

4.1. **Graphic analysis.** In order to compare the different approaches, we present an overlay of the graphs obtained with the four approaches.

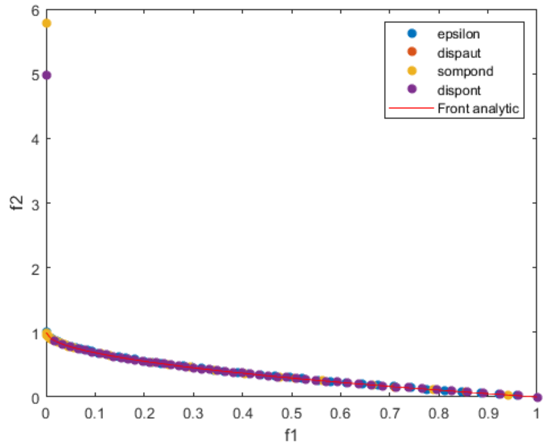

FIGURE 25. ZDT1

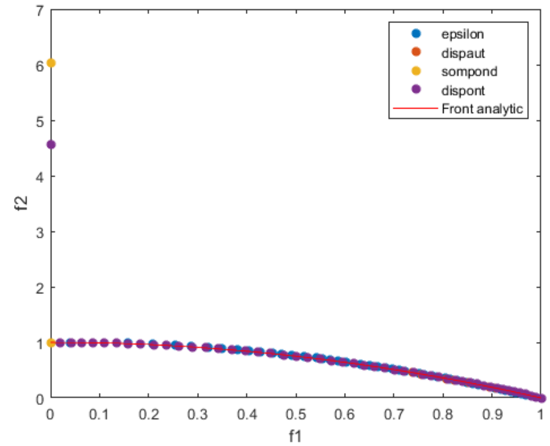

FIGURE 26. ZDT2

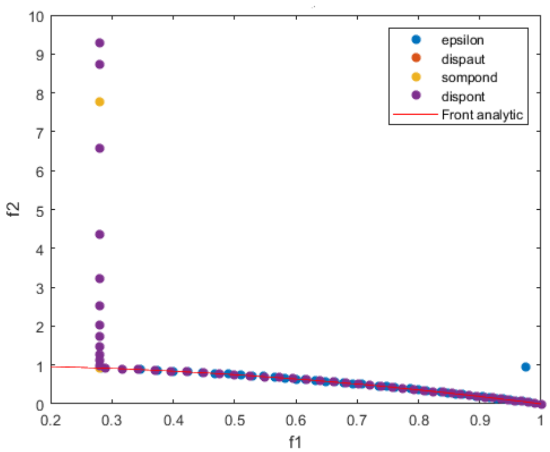

FIGURE 27. ZDT6

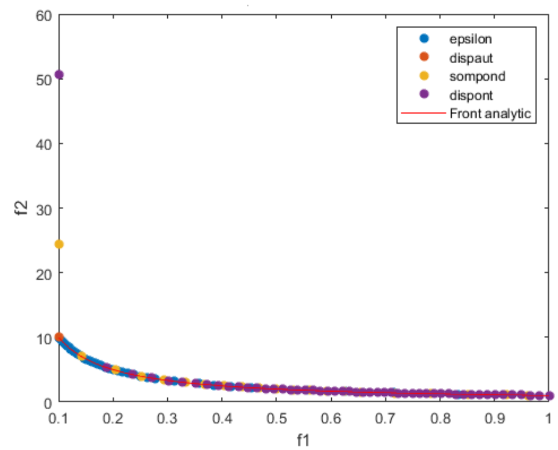

FIGURE 28. MinEx

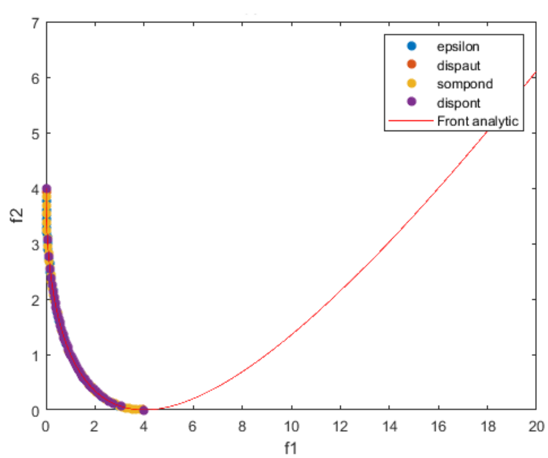

FIGURE 29. SCH

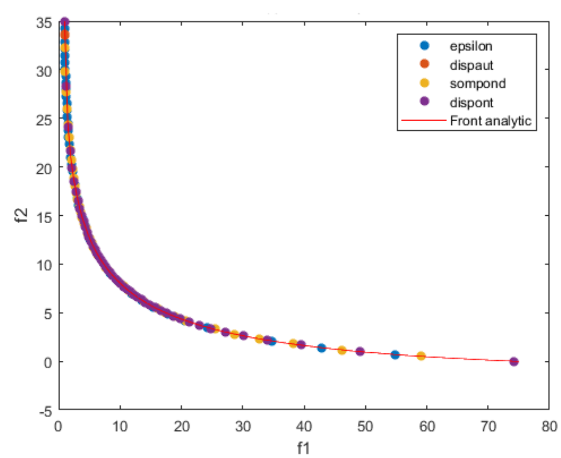

FIGURE 30. CosH

Considering these different figures, we can say that the ϵ-constraint approach and the Chebyshev augmented weighted distance approach provide good Pareto fronts for the ZDT1, ZDT2, ZDT6, and MinEx problems. Similarly, good Pareto fronts for the SCH and CosH problems are obtained using the weighted sum and weighted Tchebychev distance approaches.

Since the objective of this study is to choose the best approach for a set of given problems, the approaches will represent the alternatives, and the problems will constitute the set of criteria. The choice is based on convergence, diversity, the maximum distance between two solutions, and purity. Thus, four scenarios come into play.

4.2. **Convergence Analysis.** To perform a thorough convergence analysis, we will examine the profiles of the alternatives and the GAIA plan before proposing an upgrade graph.

i.: Analysis of the profiles of alternatives linked to convergence:

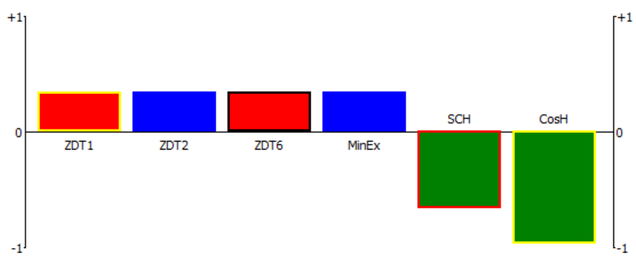

FIGURE 31. S_4 profile

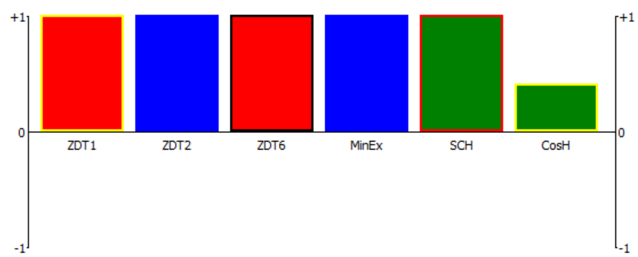

FIGURE 32. S_3 profile

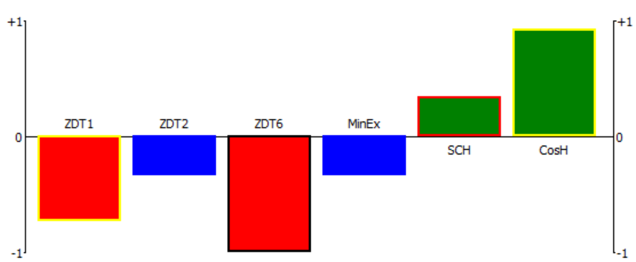

FIGURE 33. S_2 profile

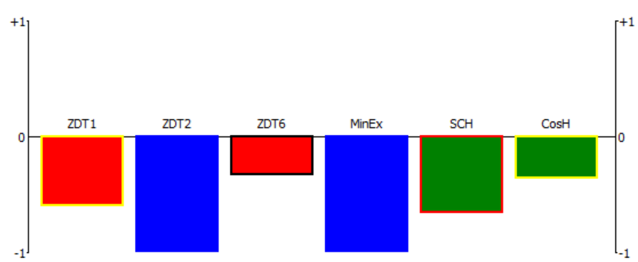

FIGURE 34. S_1 profile

The analysis of the profiles of the alternatives on convergence shows us that S_4 performs well on the problems ZDT1, ZDT2, ZDT6 and MinEx. However, it admits a weakness in the last two problems, namely SCH and CosH. S_3 is strong on all problems. S_2 performs well on SCH and CosH problems, but admits a weakness on other problems. As for S_1, it has poor convergence on all problems. Comparing the profiles of the alternatives in pairs, we find that S_3 performs better on the first five problems and S_2 converges better on the last problem.

ii.: GAIA convergence plan analysis:

The alternatives and criteria are scattered in the plan, the criteria are quite close to the axes and the delta is at 97 % which means that the GAIA plan is reliable.

Figure 35 shows us criteria far from each other and two by two in opposite directions.

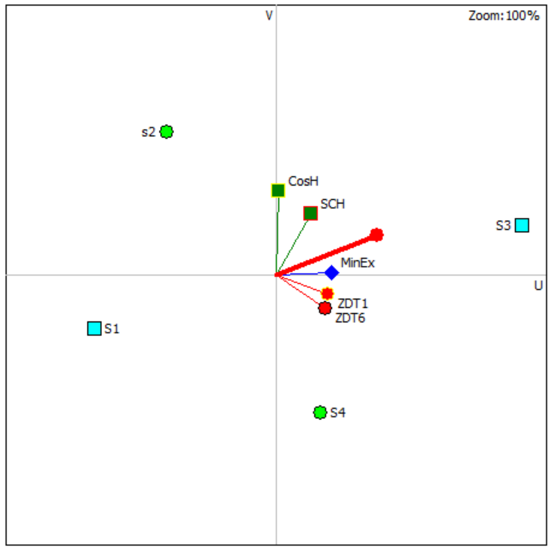

FIGURE 35. GAIA of convergence

We can therefore say that they are two by two dissimilar, and we distinguish between S_3 and S_1 which are dissimilar, and the same applies to S_4 and S_2.

The S_4 function performs best on the ZDT1, ZDT6 and MinEx problems, which are convex or concave multivariate problems. As for S_3, it performs well on SCH, MinEx and CosH problems, which are convex problems with one or two variables. This approach does not find good convergence values for concave problems. This study reveals that the performance of S_4 is close to that of S_3.

iii.: Upgrade graph:

The following table shows convergence upgrade flows:

TABLE 6. Upgrade flow

Rang	Approche	ϕ	ϕ^+	ϕ^-
1	S_3	0.9017	0.9444	0.0427
2	S_4	-0.0505	0.4444	0.4949
3	S_2	-0.1901	0.3761	0.5662
4	S_1	-0.6611	0.1167	0.7778

From the above table, we obtain the following outranking graph:

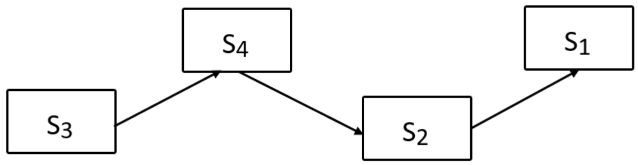

FIGURE 36. Upgrade graph linked to convergence

In terms of convergence, we therefore retain the weighted distance increased by Tchebyshev as the best approach out of all the four approaches used.

4.3. **Profile analysis of $\Delta-$spread.** For a good analysis of diversity, we will analyze the profiles of the alternatives and the GAIA plan before proposing an upgrade graph.

i.: Analysis of the profiles of alternatives linked to diversity:

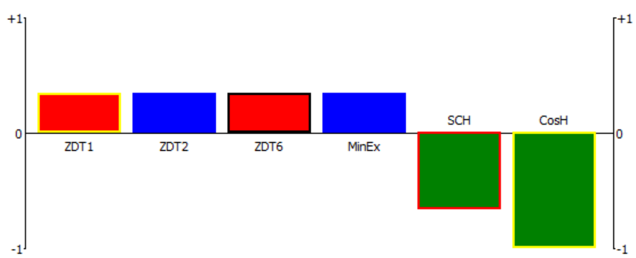

FIGURE 37. S_4 profile

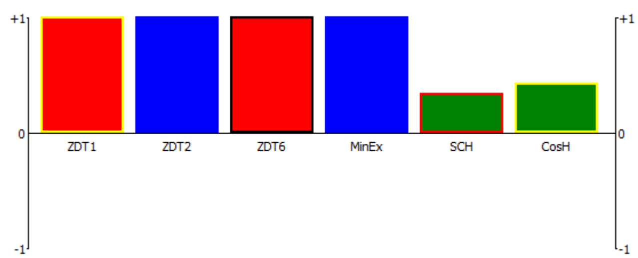

FIGURE 38. S_3 profile

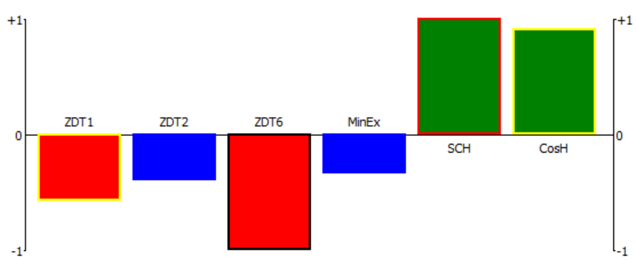

FIGURE 39. S_2 profile

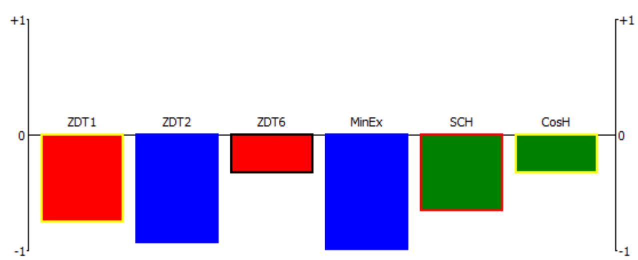

FIGURE 40. S_1 profile

The analysis of the profiles of the alternatives on $\Delta-$spread shows us that S_4 performs well on the first four problems and admits weakness on the last two. S_3 has strength on all problems. S_2 has a good distribution of solutions on the SCH and CosH problems and admits a weakness on the first four. S_1, for its part, has a poor distribution of solutions on all problems. By making a pairwise comparison of the profiles of the alternatives on the distribution of solutions, we can say that S_3 is better than S_4 on the first four problems. S_2 is also better than S_3 on the last two problems.

ii.: Analysis of the GAIA $\Delta-$spread plan:

The alternatives and criteria are scattered throughout the plan, the criteria are fairly close to the axes and the delta is 97.8 %, which means that the GAIA plan is reliable.

Figure 41 shows us criteria that are far apart and two by two in opposite directions.

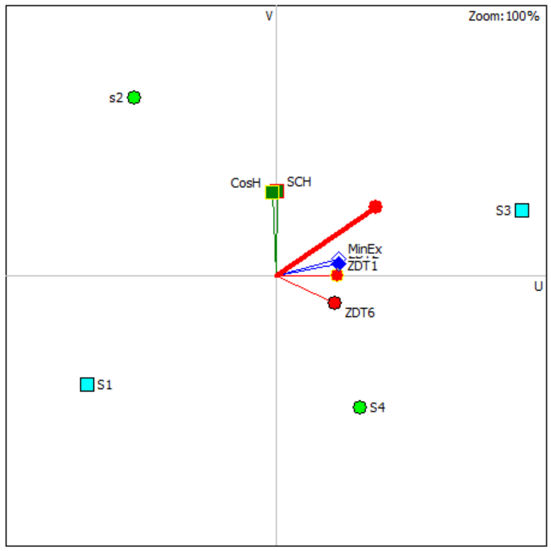

FIGURE 41. GAIA of $\Delta-$spread

We can therefore say that they are two by two dissimilar and we distinguish S_2 and S_4 which are dissimilar and it is the same for S_1 and S_4.

The S_3 function performs best on ZDT2 and MinEx problems, which are convex or concave multivariate problems. As for S_4, it performs well on the convex, multivariate ZDT6 problem. The S_3 and S_2 approaches are indifferent on the SCH and CosH problems. Similarly, S_3 and S_4 give almost identical solutions for problem ZDT1.

iii.: Upgrade graph:

The following table shows diversity upgrade flows:

TABLE 7. Upgrade flow

Rank	Approach	ϕ	ϕ^+	ϕ^-
1	S_3	0.7927	0.8889	0.0962
2	S_4	-0.0429	0.4444	0.4873
3	S_2	-0.0646	0.4354	0.5000
4	S_1	-0.6852	0.0985	0.7836

From the table above, we derive the following upgrade graph:

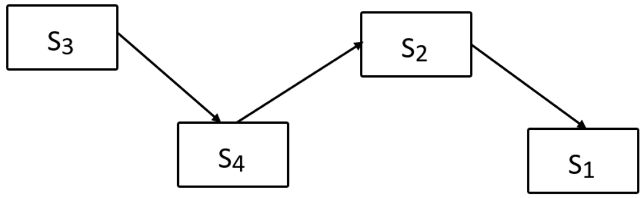

FIGURE 42. Δ-spread outranking graph

We therefore observe that in terms of solution distribution on the Pareto front, the approach of weighted distance augmented by Chebyshev is the best among all approaches considered.

4.4. **Γ-spread Analysis.** To properly analyze the maximum distance between solutions on an approximate Pareto front, we will examine the profiles of the alternatives and the GAIA plan before proposing an outranking graph.

i.: Analysis of the profiles of alternatives related to the Γ-spread:

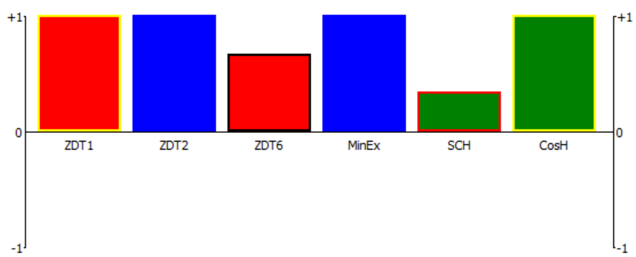

FIGURE 43. S_4 profile

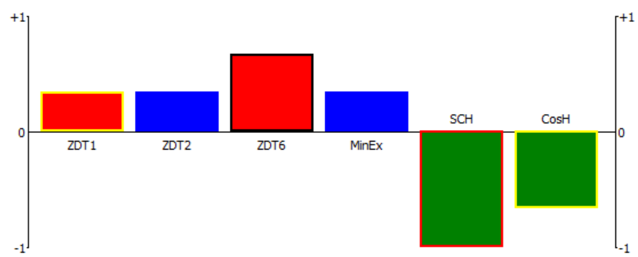

FIGURE 44. S_3 profile

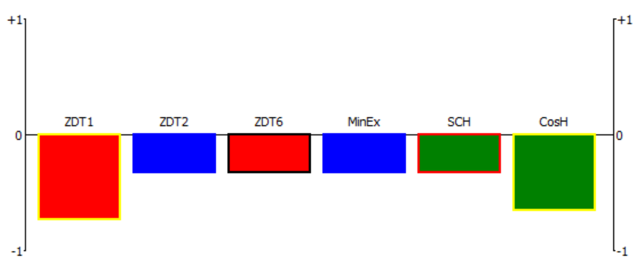

FIGURE 45. S_2 profile

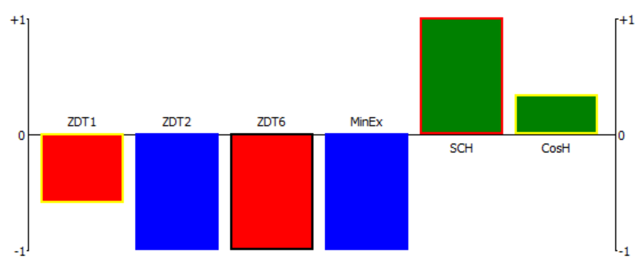

FIGURE 46. S_1 profile

The analysis of the profiles of the alternatives on the Γ-spread shows that S_4 performs well across all six problems. S_3 performs well on the first four problems but shows weakness on the last two. S_2 shows large distances between solutions across all six problems. As for S_1, it performs weakly on the first four problems but shows strength on the last two. By making a pairwise comparison of the profiles of the alternatives on the maximum distance between solutions, we can conclude that S_4 is superior to all other approaches on all problems, except for the SCH problem, which is efficiently solved by S_1.

ii.: Analysis of the Γ-spread GAIA plan:

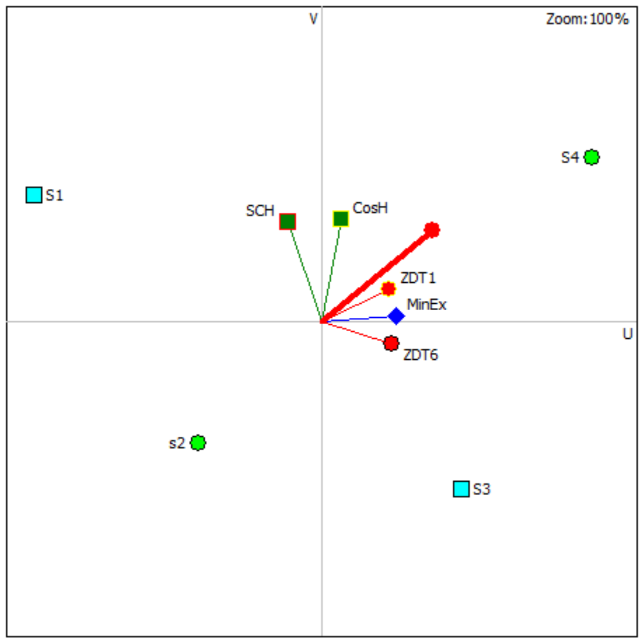

FIGURE 47. GAIA of $\Gamma-$spread

The alternatives and criteria are scattered throughout the plan. The criteria are fairly close to the axes, and the delta is $98.9\,\%$, which indicates that the GAIA plan is reliable.

Figure 47 shows the criteria are far apart and positioned in opposite directions two by two. We can therefore say that they are pairwise dissimilar. We also observe that S_1 and S_3 are dissimilar, as are S_2 and S_4.

The S_4 function performs better on problems ZDT1, ZDT2, MinEx, and CosH, which are convex or concave problems with multiple variables. On the other hand, S_3 performs well on the convex, multivariate ZDT6 problem. S_1 performs effectively on the SCH problem.

iii.: Upgrade graph: The following table presents the upgrade flow diversity:

TABLE 8. Upgrade flow

Rank	Approach	ϕ	ϕ^+	ϕ^-
1	S_4	0.8333	0.8889	0.0556
2	S_3	0.0000	0.4444	0.4444
3	S_1	-0.3767	0.2900	0.6667
4	S_2	-0.4566	0.2222	0.6789

From the table above, we derive the following upgrade graph:

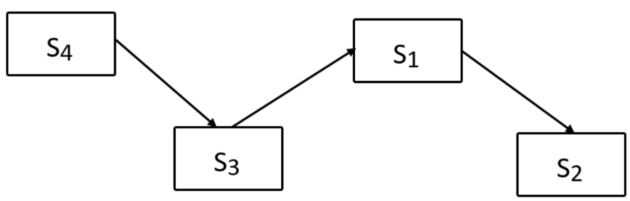

FIGURE 48. Outranking graph related to Γ-spread

We therefore conclude that, in terms of solution distribution on the Pareto front, the ϵ-constraint approach is the best among all the approaches considered.

4.5. **Purity analysis.** For a thorough purity analysis, we will follow the previous procedure.

i.: Analysis of the profiles of alternatives related to purity:

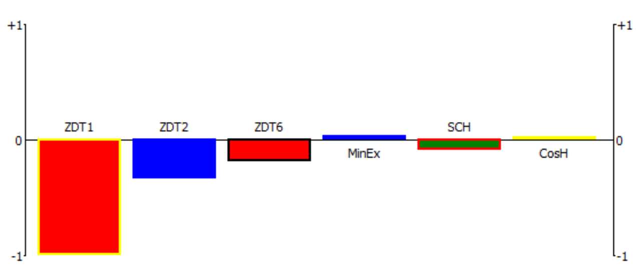

FIGURE 49. S_4 profile

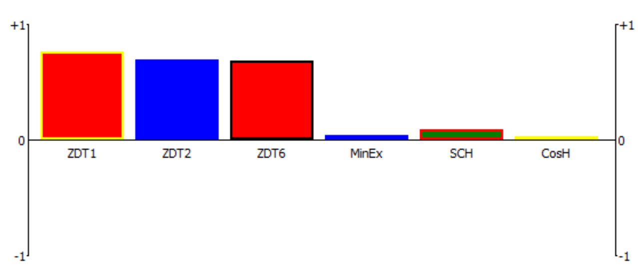

FIGURE 50. S_3 profile

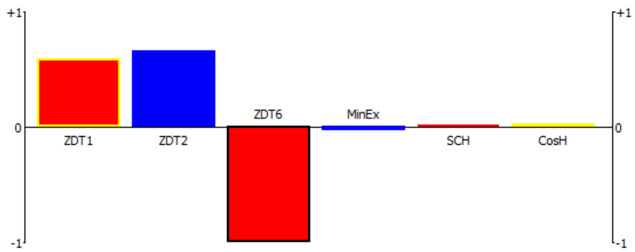

FIGURE 51. S_2 profile

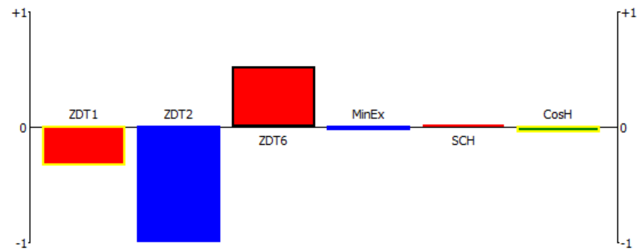

FIGURE 52. S_1 profile

The analysis of the profiles of the alternatives in terms of purity shows that S_4 is weak on the ZDT1, ZDT2, ZDT6, and SCH problems. S_3 performs well on the ZDT1, ZDT2, ZDT6, and SCH problems. S_2 performs well on ZDT1 and ZDT2, but shows weakness on the ZDT6 problem. As for S_1, it demonstrates poor purity on ZDT1 and ZDT2 but performs well on the ZDT6 problem. All approaches show good purity on the MinEx and CosH problems. By making pairwise comparisons of the alternative profiles, we observe that S_3 performs best on the first three problems, while on the last three problems, all approaches are equivalent.

ii.: Analysis of the GAIA purity plan:

The alternatives and criteria are scattered across the plan. The criteria are quite close to the axes, and the delta is $94.3\,\%$, which indicates that the GAIA plan is reliable.

Figure (53) shows criteria that are far apart and positioned pairwise in opposite directions.

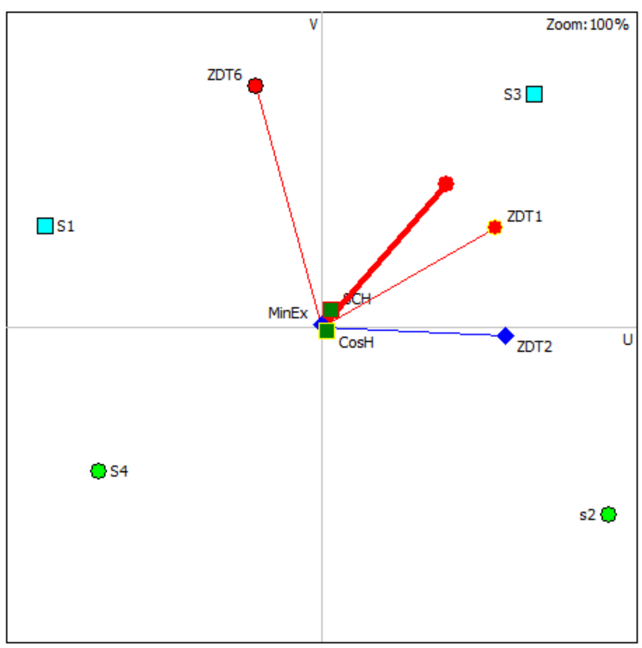

FIGURE 53. GAIA plan of purity

We can therefore conclude that they are pairwise dissimilar. We also distinguish that S_3 and S_4 are dissimilar, as are S_1 and S_2.
The S_3 function performs best on the ZDT1 problem. S_1 effectively solves the ZDT6 problem. S_2 performs well on the ZDT2 problem. The last three problems are nearly indistinguishable at the origin, which means that all approaches efficiently solve them.

iii.: Upgrade Graph:
The following table shows purity upgrade flows:

TABLE 9. Upgrade flow

Rank	Approach	ϕ	ϕ^+	ϕ^-
1	S_3	0.3723	0.3723	0.0000
2	S_2	-0.0372	0.2292	0.1921
3	S_1	-0.1485	0.1544	0.3029
4	S_4	-0.2609	0.1195	0.3803

From the table above, we derive the following upgrade graph:

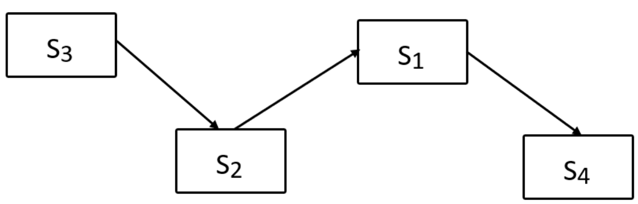

FIGURE 54. Purity upgrade graph

In terms of purity, we conclude that the weighted Chebyshev distance increase is the best approach among the four approaches used.

4.6. **Joint analysis.** For a thorough joint analysis of all four metrics, we will examine the profiles of the alternatives and the GAIA plan before proposing an upgrade graph that takes both scenarios into account.

i.: Analysis of the profiles of joint alternatives:

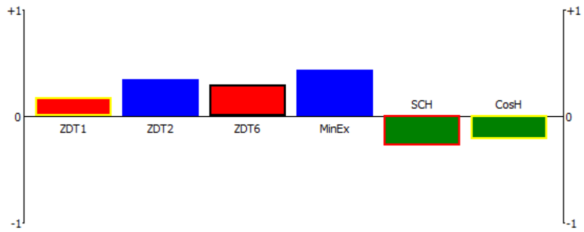

Figure 55. Profil de S_4

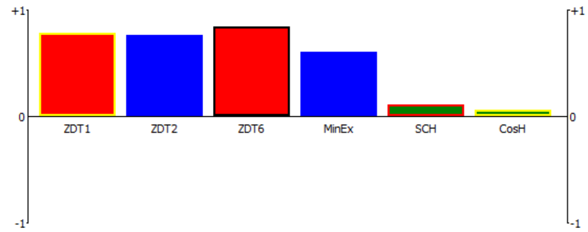

Figure 56. Profil de S_3

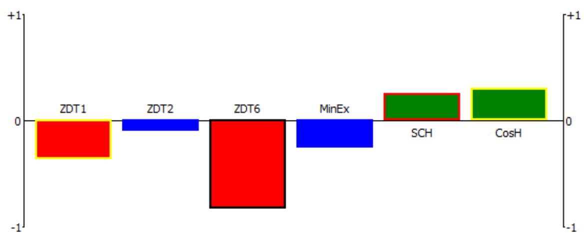

Figure 57. Profil de S_2

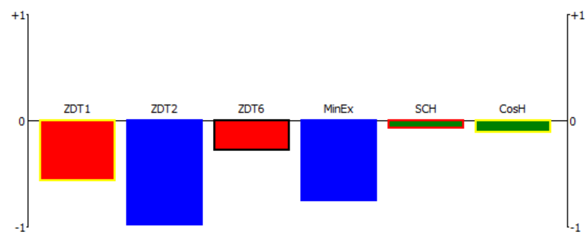

Figure 58. Profil de S_1

The analysis of the profiles of the alternatives across the four scenarios shows that S_4 performs well on the ZDT1, ZDT2, ZDT6, and MinEx problems but is weak on the last two problems. S_3 exhibits strength across all problems. S_2 is strong on the last two problems but weak on the first four. S_1, for its part, is weak on all the problems.

ii.: GAIA plan analysis of scenarios:

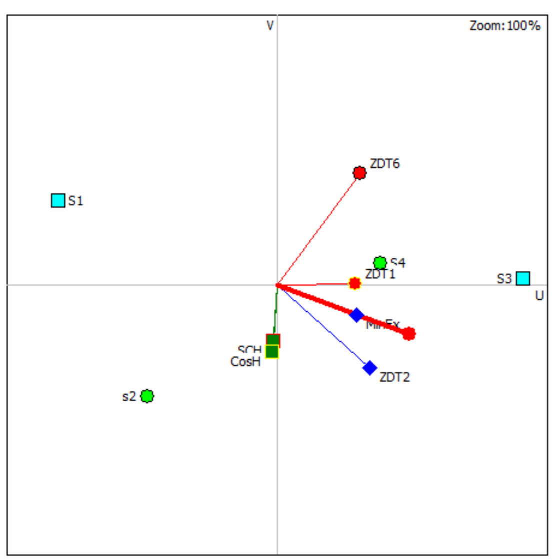

Figure 59. GAIA plan of the four scenarios

The alternatives and criteria are scattered across the plan, with the criteria fairly close to the axes and a delta of $96.7\,\%$, indicating that the GAIA plan is reliable.
The S_4 approach only performs well on the ZDT6 problem. S_3 performs better on the ZDT1, ZDT2, and MinEx problems. As for S_2, it performs well on the SCH and CosH problems. The S_1 approach does not perform as well overall compared to the other three approaches on the six problems studied.

iv.: Upgrade graph:
If we consider both scenarios simultaneously, we obtain the following flow values:

Table 10. Upgrade flow

Rank	Approach	ϕ	ϕ^+	ϕ^-
1	S_3	0.5167	0.6625	0.1458
2	S_4	0.1198	0.4743	0.3545
3	S_1	-0.1685	0.3157	0.4843
4	S_2	-0.4679	0.1649	0.6328

Following the above table, we have the following outranking graph:

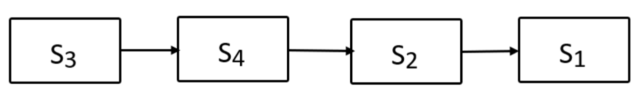

Figure 60. Upgrade graph related to all scenarios

Based on all these analyses, we recommend using the S_4 and S_3 functions for aggregating convex or concave problems with multiple variables and the S_2 function for aggregating convex problems with one variable.
For all six studied problems, we conclude that the augmented Chebyshev weighted distance approach is the best method for aggregating objective functions.

5. Conclusion

In this paper, we proposed a multi-objective version of the cuckoo search algorithm to solve multi-objective optimization problems using several solution approaches and a penalty function derived from the Lagrangian. The newly proposed method was successfully tested on six benchmark problems from the literature by presenting the Pareto fronts. A performance metric analysis was also conducted, demonstrating the effectiveness of the method. Furthermore, we studied the impact of these different solution approaches on the proposed method. This study involved analyzing the performance profiles and the GAIA plan, aiming to assist decision-makers in selecting the approach based on their preferences.

Conflicts of Interest. The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] X. Yang, Suash Deb, Cuckoo Search via Lévy Flights, in: 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), IEEE, 2009. https://doi.org/https://doi.org/10.1109/NABIC.2009.5393690.

[2] S. Walton, O. Hassan, K. Morgan, M. Brown, Modified Cuckoo Search: A New Gradient Free Optimisation Algorithm, Chaos Solitons Fractals 44 (2011), 710–718. https://doi.org/10.1016/j.chaos.2011.06.004.

[3] G. Wang, A.H. Gandomi, X. Zhao, H.C.E. Chu, Hybridizing Harmony Search Algorithm with Cuckoo Search for Global Numerical Optimization, Soft Comput. 20 (2014), 273–285. https://doi.org/10.1007/s00500-014-1502-7.

[4] T. Cuong-Le, H. Minh, S. Khatir, M.A. Wahab, M.T. Tran, et al., A Novel Version of Cuckoo Search Algorithm for Solving Optimization Problems, Expert Syst. Appl. 186 (2021), 115669. https://doi.org/10.1016/j.eswa.2021.115669.

[5] M. Shehab, A.T. Khader, M.A. Al-Betar, A Survey on Applications and Variants of the Cuckoo Search Algorithm, Appl. Soft Comput. 61 (2017), 1041–1059. https://doi.org/10.1016/j.asoc.2017.02.034.

[6] X. Li, M. Yin, A Particle Swarm Inspired Cuckoo Search Algorithm for Real Parameter Optimization, Soft Comput. 20 (2015), 1389–1413. https://doi.org/10.1007/s00500-015-1594-8.

[7] E. Valian, S. Mohanna, S. Tavakoli, Improved Cuckoo Search Algorithm for Global Optimization, Int. J. Commun. Inf. Technol. 1 (2011), 31–44. https://doi.org/10.22111/cit.2011.91.

[8] H.R. Boveiri, M. Elhoseny, RETRACTED ARTICLE: A-Coa: An Adaptive Cuckoo Optimization Algorithm for Continuous and Combinatorial Optimization, Neural Comput. Appl. 32 (2018), 681–705. https://doi.org/10.1007/s00521-018-3928-9.

[9] K.K. Bhattacharjee, S.P. Sarmah, A Binary Cuckoo Search Algorithm for Knapsack Problems, in: 2015 International Conference on Industrial Engineering and Operations Management (IEOM), IEEE, 2015, pp. 1-5. https://doi.org/10.1109/IEOM.2015.7093858.

[10] F. Xiao, S. Yang, J. Cheng, M. Hou, C. Wang, A Binary Cuckoo Search for Combinatorial Optimization in Multiyear Pavement Maintenance Programs, Adv. Civ. Eng. 2020 (2020), 8851325. https://doi.org/10.1155/2020/8851325.

[11] X. He, N. Li, X. Yang, Non-Dominated Sorting Cuckoo Search for Multiobjective Optimization, in: 2014 IEEE Symposium on Swarm Intelligence, IEEE, 2014, pp. 1–7. https://doi.org/10.1109/SIS.2014.7011772.

[12] S.H.H. Madni, M.S.A. Latiff, J. Ali, S.M. Abdulhamid, Multi-Objective-Oriented Cuckoo Search Optimization-Based Resource Scheduling Algorithm for Clouds, Arab. J. Sci. Eng. 44 (2019), 3585–3602. https://doi.org/10.1007/s13369-018-3602-7.

[13] M. Zhang, H. Wang, Z. Cui, J. Chen, Hybrid Multi-Objective Cuckoo Search with Dynamical Local Search, Memetic Comput. 10 (2017), 199–208. https://doi.org/10.1007/s12293-017-0237-2.

[14] M. Akbari, H. Rashidi, A Multi-Objectives Scheduling Algorithm Based on Cuckoo Optimization for Task Allocation Problem at Compile Time in Heterogeneous Systems, Expert Syst. Appl. 60 (2016), 234–248. https://doi.org/10.1016/j.eswa.2016.05.014.

[15] S. Jaber, Y. Ali, N. Ibrahim, An Automated Task Scheduling Model Using a Multi-Objective Improved Cuckoo Optimization Algorithm, Int. J. Intell. Eng. Syst. 15 (2022), 295–304. https://doi.org/10.22266/ijies2022.0228.27.

[16] K. Chandrasekaran, S.P. Simon, Multi-Objective Scheduling Problem: Hybrid Approach Using Fuzzy Assisted Cuckoo Search Algorithm, Swarm Evol. Comput. 5 (2012), 1–16. https://doi.org/10.1016/j.swevo.2012.01.001.

[17] D. Wolpert, W. Macready, No Free Lunch Theorems for Optimization, IEEE Trans. Evol. Comput. 1 (1997), 67–82. https://doi.org/10.1109/4235.585893.

[18] K. Miettinen, Nonlinear Multiobjective Optimization, Kluwer Academic Publishers, (1999).

[19] Siwei Jiang, Yew-Soon Ong, Jie Zhang, Liang Feng, Consistencies and Contradictions of Performance Metrics in Multiobjective Optimization, IEEE Trans. Cybern. 44 (2014), 2391–2404. https://doi.org/10.1109/tcyb.2014.2307319.

[20] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A Fast and Elitist Multiobjective Genetic Algorithm: Nsga-Ii, IEEE Trans. Evol. Comput. 6 (2002), 182–197. https://doi.org/10.1109/4235.996017.

[21] A.L. Custódio, J.F.A. Madeira, A.I.F. Vaz, L.N. Vicente, Direct Multisearch for Multiobjective Optimization, SIAM J. Optim. 21 (2011), 1109–1140. https://doi.org/10.1137/10079731x.

[22] J. Brans, B. Mareschal, Promethee Methods, in: Multiple Criteria Decision Analysis: State of the Art Surveys. International Series in Operations Research & Management Science, Vol. 78. Springer, New York, https://doi.org/10.1007/0-387-23081-5_5.