Integration of Niching Strategies into the C-DEEPSO Algorithm: An Approach for Multimodal Optimization

Daniel Arruda Ponte[1], **Dhener Rosemiro Silva**[1],
Carolina Gil Marcelino[1], **Carla Amor Divino Moreira Delgado**[1]

[1]Instituto de Computação – Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro – RJ – Brazil

danielap@ic.ufrj.br, dhenerrs@ic.ufrj.br,
carolina@ic.ufrj.br, carla@ic.ufrj.br

Abstract. *The goal of this study is to improve the performance of the C-DEEPSO algorithm in solving multimodal optimization problems by developing the FER-CDEEPSO variant. This approach aims to reduce premature convergence and improve the location of multiple global optima. Niching strategies based on the Fitness Euclidean-distance Ratio (FER) metric were inserted in the social and cognitive components of the movement equation, integrating concepts from FER-PSO and FER-DE with Crowding. At the end of optimization, the K-Means algorithm is used to identify and extract the optima. Experiments were conducted on classical multimodal benchmark functions of the literature. Results demonstrated that FER-CDEEPSO overcame the instabilities of its isolated variants and exhibited competitive spatial accuracy and computational efficiency compared to a highly relevant classic baseline (original FER-DE). By steering both guiding components through the FER metric instead of a global leader, the algorithm keeps several niches populated until the end of the run.*

Keywords: *Multimodal Optimization, Niching algorithms, FER-CDEEPSO.*

1. Introduction

Optimization seeks the best solutions to objective functions, with applications ranging from logistics and structural engineering to machine learning [Gaspar-Cunha et al. 2012, Shribate 2025]. For complex problems, deterministic gradient-based methods often struggle to perform effectively, making metaheuristics a flexible and robust alternative. Among these, Differential Evolution (DE) [Storn and Price 1997] and Particle Swarm Optimization (PSO) [Kennedy and Eberhart 1995] are widely used algorithms. In particular, the *Canonical Differential Evolutionary Particle Swarm Optimization* (C-DEEPSO) [Marcelino et al. 2016]—a hybrid method that combines recombination mechanisms from PSO with mutation and selection operators from DE—stands out as a promising approach.

Despite the robustness of metaheuristics, there is an additional challenge in multimodal optimization: the objective function has multiple local and/or global minima. In many situations, such as linear equation systems [Brits et al. 2002b] and electric motor design [Cho et al. 1999], it is necessary to have a list of optimal solutions instead of just one global optimum. To solve this type of problem, canonical evolutionary algorithms must be modified, as they tend to lose population diversity and prematurely converge to a single peak.

Niching algorithms have been developed to tackle these challenges by maintaining population diversity, evolving from pioneering approaches of the 1970s to modern modifications in PSO and DE (Section 2). FER-PSO (*Fitness Euclidean-distance Ratio Particle Swarm Optimization*) [Li 2007] uses a combination of fitness value and Euclidean distance to find the best individual to follow in the swarm, instead of solely relying on the global leader. At the same time, within the DE scope, modifications to the parent selection process for mutation have been proposed, such as FER-DE [Liang et al. 2014], where selection is performed probabilistically and proportionally to the FER metric value, favoring interactions between spatially close and fit individuals.

Although niching strategies exist in the literature, there is still a research gap regarding the integration of these mechanisms into hybrid architectures. Algorithms like C-DEEPSO, in their canonical form, still suffer from the strong pull exerted by the global leader in multimodal spaces. This work proposes and evaluates a niche variant of the hybrid architecture called FER-CDEEPSO. The main objective is to adapt the algorithm by replacing its traditional attraction mechanisms with strategies based on the FER metric, directly in its social and cognitive components. The goal is to assess whether the hybridization of these strategies allows FER-CDEEPSO to achieve a competitive balance between the ability to maintain multiple optima, convergence speed, and total execution time, compared to highly relevant classic niching algorithms such as FER-DE.

C-DEEPSO is a convenient host for niching: it guides each particle through two independent channels—cognitive and social—so a distinct mechanism can be installed in each, which single-family metaheuristics do not allow. The combination therefore targets problems exhibiting several optima of equivalent quality, for which the whole solution set is required rather than a single point; the functions of Section 5 represent that class. The proposal is evaluated against the two partial variants and the classic baseline Original FER-DE, with 31 independent runs for each of six population sizes, all comparisons being validated by the Friedman test followed by the Nemenyi post-hoc procedure.

2. Related Work

Niching methods can be organized by the mechanism through which they enforce the coexistence of subpopulations. One family derates fitness inside a niche radius: *Fitness Sharing* [Goldberg and Richardson 1987] shares the fitness of individuals falling within a sharing radius, and *Clearing* [Petrowski 1996] preserves only the best individuals inside each radius. A second family constrains replacement instead, making an offspring compete against the most similar individual rather than against the population at large—*Crowding* [De Jong 1975] and *Restricted Tournament Selection* [Harik 1995]—and it is on this second family that the present work builds.

Within PSO, this line produced species-based approaches (SPSO [Li 2004, Parrott and Li 2006], *NichePSO* [Brits et al. 2002a]), neighborhood-informed variants (FDR-PSO [Veeramachaneni et al. 2003], LIPS [Qu et al. 2013]) and multi-swarm schemes that adaptively split and merge subswarms (NMMSO [Fieldsend 2014]); FER-PSO [Li 2007] belongs to the neighborhood-informed group and removes the need to pre-specify a niche radius. Within DE, an analogous progression led from neighborhood-based mutation (NCDE) [Qu et al. 2012] to FER-DE [Liang et al. 2014], the variant adopted in this work. A parallel line adapts CMA-ES to multimodal search, either

by nearest-better clustering (NEA2 [Preuss 2012]) or by repelling subpopulations (RS-CMSA [Ahrari et al. 2017] and its successor RS-CMSA-ESII [Ahrari et al. 2022]), a lineage that has led the niching competitions organised around the CEC'2013 benchmark. Together with NMMSO, these methods are the standing reference points on that suite.

Each of the methods above applies its niching mechanism inside a single-family metaheuristic. Hybrid architectures have received considerably less attention in this respect: C-DEEPSO [Marcelino et al. 2016], which extends DEEPSO [Miranda and Alves 2013] and serves as the basis of this work, still steers both of its guiding components by a single global reference—the mechanism this work replaces. To the best of our knowledge, applying FER-based niching to *both* components of a PSO–DE hybrid, coupled with Crowding at the strategy-vector level, has not been previously reported.

3. Theoretical Background

This section details the mechanisms of the underlying C-DEEPSO algorithm and the niche strategies based on the FER (Fitness Euclidean-distance Ratio) metric, which are the building blocks of the modifications proposed in Section 4.

3.1. C-DEEPSO

The *Canonical Differential Evolutionary Particle Swarm Optimization* (C-DEEPSO) [Marcelino et al. 2016] is a hybrid metaheuristic that combines characteristics of Particle Swarm Optimization (PSO) and Differential Evolution (DE). Unlike traditional PSO, C-DEEPSO particles have no individual memory: they share a collective memory and are guided by a strategy vector built from DE operators. At each iteration, for each particle x_i, a mutant vector x_{DE} is generated by one of the DE strategies the algorithm admits; *DE/current-to-best/1* is taken here as the illustrative case, since it exercises every term the strategy may contain. It combines the global best position x_{gb}, a reference vector x_r sampled from the population or from the collective memory, and vectors x_{r_j} randomly sampled from the population:

$$x_{DE} = x_r + F \cdot (x_{\text{gb}} - x_r) + F \cdot (x_{r_1} - x_{r_2}). \tag{1}$$

A vector u_{st} is then generated through binary crossover between x_{DE} and x_r. Based on the objective function evaluation, the strategy vector x_{st} is defined as the fittest between u_{st} and x_r. The velocity update is corrected by a *directional macrogradient* $\hat{\nabla}(x_{st}, x_i)$, a scalar sign that equals $+1$ when $f(x_{st}) < f(x_i)$ and -1 otherwise. It dictates *where the particle goes*: *towards* the strategy vector when that vector is fitter and *away* from it when it is not, so the cognitive step is always a descent direction. The velocity is given by the vector equation:

$$v_i' \leftarrow \underbrace{\tilde{w}_I v_i}_{\text{Inertia}} + \underbrace{\tilde{w}_A \cdot \hat{\nabla}(x_{st}, x_i) \cdot (x_{st} - x_i)}_{\text{Cognitive}} + \underbrace{\tilde{w}_C \cdot C \cdot (\tilde{x}_{gb} - x_i)}_{\text{Social}}, \tag{2}$$

in which w_I, w_A, and w_C are the inertia, macrogradient and communication weights, and C is a stochastic binary communication matrix. The tilde denotes a quantity taken *after a slight random perturbation*: at every iteration the three weights and the global best are mutated by Gaussian noise of intensity τ_{mut}, as $\tilde{w} = w + \tau_{mut} \cdot r$ and $\tilde{x}_{gb} = x_{gb} \cdot (1 + \tau_{mut} \cdot r)$ with $r \sim \mathcal{N}(0, 1)$. This self-adaptive mutation keeps the swarm from repeating the same trajectory and stagnating.

3.2. FER-Based Niching Strategies

Multimodal optimization requires population diversity to exploit multiple peaks at once. The FER metric balances fitness difference and Euclidean distance between two particles, reducing the influence of individuals from distinct niches.

3.2.1. FER-PSO

The FER-PSO algorithm [Li 2007] modifies the classical social attraction of PSO. Instead of the entire swarm following a single global leader (g_b), each particle i follows its fittest and nearest neighbor (p_n). The selection of this neighbor is based on the maximization of the metric FER:

$$FER(j,i) = \alpha \cdot \frac{f(p_i) - f(p_j)}{||p_j - p_i||}, \tag{3}$$

in which $f(p)$ is the objective value at the best known position of the particle and α is the ratio between the diagonal of the search space and the fitness range of the population. Since the problems addressed here are minimization problems, the numerator is positive whenever p_j is fitter than p_i, so maximizing FER returns the neighbor that is at once fit and close. The neighbor p_n is the best known position of the particle $k = \arg\max_j FER(j,i)$, evaluated over all j with $||p_i - p_j|| > 0$.

3.2.2. FER-DE

In the context of Differential Evolution, the FER-DE algorithm [Liang et al. 2014] uses the same premise to modify the parent selection step during mutation. As Equation 3 is signed and therefore unusable as a weight, selection relies on the non-negative variant $FER^{+}_{(j,i)} = [f(p_{worst}) - f(p_j)]/||p_j - p_i||$, which ranks candidates by fitness and proximity against the worst individual of the population. The probability $P_{j,i}$ of the i-th individual selecting the j-th one to compose the mutation is then:

$$P_{j,i} = \frac{FER^{+}_{(j,i)}}{\sum_{k=1,k\neq i}^{NP} FER^{+}_{(k,i)}}. \tag{4}$$

This approach ensures that offspring are generated from individuals inhabiting the same spatial niche, promoting local exploration without destroying subpopulations from distant attraction basins.

4. Proposed Modifications

C-DEEPSO is a robust hybrid metaheuristic, but it tends to converge to a single global optimum due to the strong attraction exerted by the global leader. The proposed approach, called FER-CDEEPSO, aims to mitigate this limitation by integrating niching metrics to guide particles and employing a Crowding mechanism. The two components are rebuilt with deliberately complementary roles: the cognitive one, based on FER-DE with Crowding, is charged with exploring the space and preserving distant basins, while the social one, based on FER-PSO, intensifies the search inside each niche. It is the hybrid architecture that makes this division of labor possible, since its two guiding channels are independent—a single-family metaheuristic offers only one.

4.1. Adaptation of the Social Component (FER-PSO Integration)

In default C-DEEPSO, the social component guides particles toward the best solution. In this proposal, this attraction is replaced by a dynamic based on FER-PSO, with the objective of improving local refinement within each niche. For each particle i, the index k of the individual maximizing the FER metric over the swarm is identified, weighting distance and fitness:

$$k = \underset{j \in \{1,\ldots,NP\}, j \neq i}{\arg\max} \; FER(j, i), \quad x_{social,i} = x_k. \tag{5}$$

Since C-DEEPSO keeps no individual memory (Section 3.1), x_k is the current position of particle k rather than its personal best. This vector $x_{social,i}$ is used as the attractive reference in the social component, allowing different subpopulations to simultaneously converge to distinct peaks.

4.2. Modification of the Cognitive Component (FER-DE Integration)

The modification of the cognitive component aims to promote greater exploration capacity of the search space. To this end, the generation of the mutant vector (X_{DE}) abandons the uniformly random selection of classical DE and adopts the probability matrix $P_{j,i}$ of FER-DE (Equation 4). Using this distribution through the roulette wheel, three distinct indices (r_1, r_2, r_3) are selected from the population to generate the mutant vector:

$$X_{DE} = x_{r_1} + F \cdot (x_{r_2} - x_{r_3}). \tag{6}$$

Two changes separate Equation 6 from Equation 1: the strategy becomes *DE/rand/1*, so the global-best guidance term is dropped and no global attractor is left in the cognitive channel either; and the three indices are drawn by FER-weighted roulette rather than uniformly. Here the scale factor F is drawn uniformly at random in $[0, 1)$ for each component, a common device for increasing population diversity. The trial vector $u_{st,i}$ is then obtained by binary crossover between X_{DE} and the current particle position x_i.

Unlike canonical C-DEEPSO, which compares u_{st} with the swarm's memory, the definition of the final strategy vector x_{st} uses Crowding logic. The strategy vector is initialized as a copy of the current population. For each trial vector, the closest individual x_k in Euclidean distance is identified ($k = \arg\min_{j \in \{1,\ldots,NP\}} ||u_{st,i} - x_j||$), and $x_{st,k}$ is overwritten by $u_{st,i}$ only if $f(u_{st,i}) < f(x_k)$; an individual that is not the nearest neighbor of any trial vector therefore keeps its own position. This ensures that new individuals only replace similar solutions, preventing a promising niche from eliminating individuals from distant niches.

4.3. New FER-CDEEPSO Movement Equation

To integrate the new targets defined for the cognitive (x_{st}) and social (x_{social}) components, the velocity update of each particle i is now governed by the following equation:

$$v_i' \leftarrow \underbrace{\tilde{w}_I \cdot v_i}_{\text{Inertia}} + \underbrace{\tilde{w}_A \cdot \hat{\nabla}(x_{st,i}, x_i) \cdot (x_{st,i} - x_i)}_{\text{Cognitive}} + \underbrace{\tilde{w}_C \cdot C \cdot (\tilde{x}_{social,i} - x_i)}_{\text{Social}}. \tag{7}$$

The niche leader $\tilde{x}_{social,i}$ replaces $\tilde{x}_{gb}$ but inherits the same perturbation of Section 3.1, so diversity is now injected around the *local* leader. Moreover, as the Crowding rule of Section 4.2 overwrites an individual only with a strictly fitter trial vector, $f(x_{st,i}) \leq f(x_i)$ holds by construction and the macrogradient is $+1$ whenever $x_{st,i} \neq x_i$, making the cognitive term a pure attraction.

4.4. Optima Identification via Clustering

At the end of a run the population holds NP individuals spread over several basins: reporting all of them would be redundant, and an arbitrary top-k could return several individuals drawn from a single basin. Clustering addresses this by partitioning the final population by basin, so each niche contributes one representative; the partitional K-Means algorithm is used. The number of clusters k is determined autonomously by the Silhouette Method, which evaluates $k \in [2, 20]$ and selects the value maximizing the average silhouette coefficient; the upper bound follows from the test function with the most global optima in this study, which has 18 minima.

The geometric centroid of a cluster is *never* reported as an optimum: being an average of positions, it need not be feasible or close to the true peak in asymmetric basins. For each cluster the algorithm reports the *best-fitness individual actually evaluated within it*. The centroid is used only to compute the auxiliary *Mean Distance to Centroids* cohesion metric (Table 3), not as a substitute solution. Algorithm 1 formalizes the complete workflow, from the FER-DE roulette wheel to the final clustering step.

Algorithm 1. FER-CDEEPSO

Require: Population size NP, control parameters $(w, w_A, w_C, \tau_{mut}, \rho_{com})$, max. evaluations $Eval_{max}$
Ensure: Set of optima $\{x_1^*, \ldots, x_k^*\}$

Initialize positions P and velocities V randomly; evaluate fitness F of P
while evaluations $< Eval_{max}$ **do**
  Compute $P_{j,i}$ from the FER metric and generate X_{DE} via roulette wheel ▷ cognitive (FER-DE), Eq. 4–6
  Generate U_{st} via binary crossover between X_{DE} and P; evaluate U_{st}
  for each trial vector $U_{st,i}$ **do**
    $k \leftarrow \arg\min_j \|U_{st,i} - x_j\|$; $X_{st,k} \leftarrow U_{st,i}$ **if** $f(U_{st,i}) < f(x_k)$ **else** x_k ▷ Crowding
  end for
  Compute the FER matrix over P and set X_{social} to the FER-maximizing neighbor ▷ social (FER-PSO), Eq. 3–5
  Build C from ρ_{com}; mutate $\tilde{w}, \tilde{w}_A, \tilde{w}_C$; compute $P' \leftarrow P + V'$ from X_{st}, X_{social} ▷ Eq. 7
  Evaluate P'; replace $x_i \in P$ by $x'_i \in P'$ if $f(x'_i) < f(x_{st,i})$
end while
Cluster the final population with K-Means/Silhouette; **return** the best-fitness individual of each cluster

5. Test Functions and Experimental Setup

To evaluate the performance of the proposed approach, seven multimodal benchmark functions were selected among those recurrent in the niching literature and catalogued in surveys of global-optimization test problems [Jamil and Yang 2013]. Table 1 lists their characteristics and the tolerance ε used to accept a global optimum as found. Two of them, Rastrigin and Hartmann 6, have a single global optimum and are included deliberately: they verify that the niching mechanism still reaches the global basin despite the many competing local ones, whereas the remaining five exercise the ability to hold several global optima at once.

5.1. Performance Metrics and Statistical Procedures

Algorithm performance was evaluated using six main metrics:

- *(1) Success Rate (SR)*: percentage of runs locating all global optima coordinates within tolerance ε [Li 2007];

Table 1. Multimodal test functions and error tolerance (ε)

Function	Dimension	Domain	No. of Minima	ε
Himmelblau	2	$[-5.12, 5.12]^2$	4 Global	10^{-5}
Six Hump Camel	2	$[-2, 2]^2$	2 Global, 4 Local	10^{-4}
Shubert	2	$[-10, 10]^2$	18 Global, multiple local	5×10^{-2}
Branin	2	$[-5, 15]^2$	3 Global	10^{-4}
Rastrigin	2	$[-5.12, 5.12]^2$	1 Global, multiple local	10^{-8}
Hartmann 6	6	$[0, 1]^6$	1 Global, 1 Local	10^{-5}
Holder Table	2	$[-10, 10]^2$	4 Global, multiple local	10^{-2}

- *(2) Average Number of Optima Found*;
- *(3) Mean Values* of the objective functions at the found minima;
- *(4) Standard Deviation* of the values;
- *(5) Mean Distance to Centroids* computed by K-Means, measuring the cohesion of formed niches; and
- *(6) Mean Minimum Evaluations* required to locate all optima, reflecting computational cost.

To isolate the impact of each modification, FER-CDEEPSO was compared with three variants: Original FER-DE (a highly relevant classic baseline), CDEEPSO FER-PSO (social component only), and CDEEPSO FER-DE (cognitive component only). For each function and variant, 31 independent runs were performed. The maximum number of evaluations was fixed at 200,000. Population size (NP) varied across ($50, 100, 150, 200, 250$, and 300). The control parameters of FER-CDEEPSO were defined empirically: Inertia weight ($w = 0.77$), Assimilation coefficient ($w_a = 1.6$), Communication coefficient ($w_c = 1.75$), Mutation rate ($\tau_{mut} = 0.1$), Communication probability ($\rho_{com} = 0.7$), Recombination rate ($CR = 0.53$), DE strategy (*DE/Rand/1*), and maximum clusters ($k_{max} = 20$). To validate comparisons, the non-parametric Friedman test was applied with a significance level of 5%. In case of statistical difference, the Nemenyi post-hoc test was used.

6. Results and Discussion

Table 2 opens the evaluation with the pairwise statistical comparison between FER-CDEEPSO and the classic baseline, Original FER-DE, aggregated over the six population sizes. Results appear as **Win / Tie / Loss (W/T/L)**: a *win* is statistical superiority of FER-CDEEPSO ($p < 0.05$), a *tie* equivalence and a *loss* inferiority. Each cell sums to 6, the six evaluated NP scenarios.

As shown in Table 2, FER-CDEEPSO is statistically equivalent to the baseline in Success Rate (41 ties) but significantly outperforms it in Niche Cohesion (35 wins against 6 losses), indicating a strong capacity to locate the center of the attraction basins precisely. Regarding computational cost, FER-CDEEPSO achieved 15 wins against 3 losses, establishing its overall efficiency. The few losses observed were concentrated primarily in the *Shubert* function, which presents a highly dense grid of global and local optima.

6.1. Success Rate and Exploration Capacity

As shown in Table 3, CDEEPSO FER-PSO presented consistently lower success rates and greater variability, although for most functions and population sizes the Friedman–

Table 2. Win / Tie / Loss (W/T/L) statistical summary (Nemenyi test, $\alpha = 0.05$) comparing FER-CDEEPSO against Original FER-DE across the six population sizes.

Function	Success Rate (Optima Found)	Mean Distance (Niche Cohesion)	Mean Min. Evals (Comp. Cost)
Himmelblau	0 / 6 / 0	6 / 0 / 0	0 / 6 / 0
Six Hump Camel	0 / 6 / 0	6 / 0 / 0	0 / 6 / 0
Shubert	0 / 6 / 0	0 / 0 / 6	0 / 4 / 2
Branin	0 / 6 / 0	6 / 0 / 0	6 / 0 / 0
Rastrigin	0 / 6 / 0	6 / 0 / 0	4 / 2 / 0
Hartmann 6	0 / 6 / 0	5 / 1 / 0	0 / 5 / 1
Holder Table	1 / 5 / 0	6 / 0 / 0	5 / 1 / 0
Total (W/T/L)	**1 / 41 / 0**	**35 / 1 / 6**	**15 / 24 / 3**

Nemenyi tests found no significant difference. The trend nonetheless suggests that niching applied to the social component alone does not ensure robustness in harder problems.

The cognitive modification alone (CDEEPSO FER-DE), by contrast, was statistically competitive: the tests found no significant difference in success rate for most functions. Both locate multiple optima equally well, so the value of combining the two mechanisms must be sought in the remaining metrics.

6.2. Niche Quality and Cohesion

Niche quality is assessed by the mean distance from each located optimum to the centroid of its cluster: lower values indicate niches centered more precisely on the global optima, hence more cohesive clusters. Table 3 shows that isolating the social modification (CDEEPSO FER-PSO) produced the most cohesive clusters in most of the tested configurations; contrasted with its success rates, however, this extreme compaction did not translate into robustness. CDEEPSO FER-DE displays the opposite profile: less cohesive clusters, yet more stable and competitive results. Against Original FER-DE, FER-CDEEPSO produced significantly more cohesive clusters on every function except Shubert—a difference confirmed statistically, not merely numerical.

6.3. Computational Cost

This metric is taken from the convergence history of each run, as the minimum number of evaluations needed to locate all known global minima, averaged only over successful runs. This is where the two partial variants separate (Table 4). Although CDEEPSO FER-DE matches FER-CDEEPSO in success rate, it required a significantly higher number of evaluations in almost all scenarios (Friedman–Nemenyi, $p < 0.05$). Against the classic baseline, FER-CDEEPSO was significantly cheaper on Branin, Rastrigin and Holder Table, and trended cheaper on Himmelblau; Original FER-DE was cheaper on Six Hump, Hartmann 6 and Shubert, though mostly without statistical significance. FER-CDEEPSO therefore preserves diversity without paying for it in evaluations, despite its greater structural complexity.

Table 3. Consolidated results for Mean Distance to Centroids and Success Rate

Function	No. of Particles	Mean Distance to Centroids				Success Rate (%)			
		CDEEPSO FER-DE	CDEEPSO FER-PSO	FER-CDEEPSO	FER-DE Orig.	CDEEPSO FER-DE	CDEEPSO FER-PSO	FER-CDEEPSO	FER-DE Orig.
Himmelblau	50	0.0044	0.0000	0.0027	0.0298	74.19	77.42	96.77	96.77
	100	0.0046	0.0001	0.0025	0.0199	100.00	58.06	100.00	100.00
	150	0.0051	0.0003	0.0029	0.0183	100.00	74.19	100.00	100.00
	200	0.0064	0.0003	0.0037	0.0176	100.00	64.52	100.00	100.00
	250	0.0070	0.0006	0.0051	0.0212	100.00	67.74	100.00	100.00
	300	0.0119	0.0007	0.0066	0.0234	100.00	61.29	100.00	100.00
Six Hump	50	0.0001	0.0000	0.0001	0.0104	100.00	100.00	100.00	100.00
	100	0.0005	0.0000	0.0002	0.0051	100.00	100.00	100.00	100.00
	150	0.0005	0.0000	0.0004	0.0065	100.00	100.00	100.00	100.00
	200	0.0008	0.0000	0.0006	0.0060	100.00	100.00	100.00	100.00
	250	0.0006	0.0000	0.0010	0.0080	100.00	100.00	100.00	100.00
	300	0.0009	0.0000	0.0012	0.0077	100.00	100.00	100.00	100.00
Shubert	50	0.0003	0.0000	0.0001	0.0000	58.06	0.00	38.71	80.65
	100	0.0005	0.0000	0.0002	0.0000	96.77	0.00	80.65	100.00
	150	0.0004	0.0001	0.0002	0.0000	96.77	9.68	90.32	100.00
	200	0.0004	0.0003	0.0003	0.0001	100.00	29.03	100.00	100.00
	250	0.0005	0.0005	0.0003	0.0001	100.00	58.06	96.77	100.00
	300	0.0005	0.0010	0.0003	0.0001	100.00	83.87	100.00	100.00
Branin	50	0.0109	0.0004	0.0048	0.0427	64.52	83.87	100.00	96.77
	100	0.0142	0.0023	0.0057	0.0308	96.77	67.74	100.00	100.00
	150	0.0647	0.0022	0.0098	0.0377	100.00	74.19	100.00	100.00
	200	0.1021	0.0030	0.0246	0.0820	100.00	61.29	100.00	100.00
	250	0.1364	0.0025	0.0420	0.1562	87.10	67.74	100.00	80.65
	300	0.2014	0.0110	0.0772	0.2479	61.29	61.29	90.32	45.16
Rastrigin	50	0.0002	0.0000	0.0003	0.4359	100.00	100.00	100.00	70.97
	100	0.0002	0.0000	0.0003	0.4120	100.00	100.00	100.00	83.87
	150	0.0003	0.0000	0.0005	0.3031	100.00	100.00	100.00	83.87
	200	0.0004	0.0000	0.0008	0.2564	100.00	100.00	100.00	93.55
	250	0.0003	0.0000	0.0008	0.2208	100.00	100.00	100.00	93.55
	300	0.0005	0.0000	0.0007	0.2014	100.00	100.00	100.00	100.00
Hartmann 6	50	0.0003	0.0000	0.0001	0.0123	96.77	100.00	100.00	100.00
	100	0.0003	0.0000	0.0006	0.0132	100.00	96.77	100.00	100.00
	150	0.0013	0.0000	0.0003	0.0137	100.00	93.55	100.00	100.00
	200	0.0005	0.0000	0.0012	0.0137	100.00	93.55	100.00	100.00
	250	0.0007	0.0001	0.0004	0.0137	100.00	58.06	100.00	100.00
	300	0.0009	0.0001	0.0010	0.0170	100.00	64.52	100.00	100.00
Holder Table	50	2.5049	0.0000	1.3166	10.5322	87.10	22.58	96.77	16.13
	100	1.1163	0.0000	0.6219	4.9824	93.55	58.06	100.00	70.97
	150	1.1476	0.0000	0.3202	2.3559	87.10	83.87	100.00	87.10
	200	1.0273	0.0000	0.4245	1.7438	93.55	96.77	100.00	93.55
	250	1.1162	0.0000	0.3420	1.7685	96.77	100.00	100.00	93.55
	300	1.0383	0.0000	0.3647	1.7531	100.00	100.00	100.00	96.77

Table 4. Consolidated results: Mean Min. Evaluations and Execution Time (s)

Function	No. of Particles	Mean Minimum Evaluations (Cost)				Mean Execution Time (s)			
		CDEEPSO FER-DE	CDEEPSO FER-PSO	FER-CDEEPSO	FER-DE Orig.	CDEEPSO FER-DE	CDEEPSO FER-PSO	FER-CDEEPSO	FER-DE Orig.
Himmelblau	50	36,241	80,558	13,930	16,615	2.04	4.25	2.56	3.12
	100	38,983	34,400	22,409	24,654	2.06	3.37	2.90	3.58
	150	56,046	44,302	33,072	36,512	2.55	3.62	3.60	4.73
	200	78,703	54,000	45,567	50,193	3.11	3.92	4.41	5.99
	250	95,491	37,107	56,443	64,072	3.62	4.28	5.16	6.99
	300	119,990	58,405	71,274	76,770	4.23	4.63	6.03	8.17
Six Hump	50	8,908	16,285	7,295	6,300	2.06	3.63	2.52	3.15
	100	19,519	22,164	13,170	11,545	2.11	3.52	2.78	3.66
	150	27,488	35,520	18,130	17,970	2.61	3.62	3.55	5.32
	200	37,296	43,941	27,670	25,374	3.11	3.93	4.45	5.87
	250	47,185	46,040	35,620	32,629	3.62	4.22	5.13	6.87
	300	59,274	58,383	42,319	39,377	4.14	4.67	5.91	8.13
Shubert	50	42,300	–	64,275	13,246	2.41	4.06	2.88	3.53
	100	38,706	42,812	36,369	12,383	2.37	3.80	3.10	3.87
	150	41,850	36,235	24,843	17,308	2.88	4.67	3.93	5.05
	200	41,980	39,306	34,264	22,019	3.89	5.35	4.96	6.25
	250	46,346	32,298	24,033	21,250	4.37	5.17	5.91	7.49
	300	50,448	44,835	27,280	27,503	5.07	5.57	7.09	8.89
Branin	50	35,335	65,027	18,040	32,291	2.08	4.81	2.64	3.26
	100	69,293	28,747	27,235	60,864	3.20	4.39	3.05	3.71
	150	89,791	38,406	38,627	92,632	3.96	4.98	4.64	4.78
	200	109,929	41,800	57,748	129,941	4.12	4.89	5.29	6.12
	250	139,546	34,416	67,346	160,657	4.82	5.33	6.13	7.60
	300	153,871	51,268	75,321	163,423	4.79	5.34	6.60	8.38
Rastrigin	50	17,320	15,524	20,937	41,656	2.02	3.55	2.55	3.39
	100	36,983	27,074	35,132	59,984	2.12	3.48	2.90	3.81
	150	61,398	42,362	52,862	94,303	2.59	3.71	3.66	4.86
	200	86,677	54,367	74,445	116,310	3.19	4.24	4.45	6.05
	250	97,879	60,604	86,153	123,560	3.69	4.30	5.32	7.03
	300	129,725	83,564	118,964	139,345	4.25	4.65	6.19	8.24
Hartmann 6	50	23,946	50,437	16,117	15,800	2.85	5.18	3.29	3.98
	100	51,712	58,606	40,022	30,912	2.76	3.96	3.87	4.68
	150	83,095	81,584	53,066	48,106	3.43	4.28	4.72	6.08
	200	115,025	99,082	84,070	65,109	4.16	4.51	5.76	7.75
	250	147,475	106,388	83,201	85,201	4.80	4.90	7.03	9.08
	300	179,216	115,620	117,977	102,987	5.72	5.44	8.31	10.67
Holder Table	50	59,757	6,619	27,393	60,396	1.98	3.51	2.62	3.35
	100	46,182	8,820	10,700	58,893	2.11	3.30	2.89	3.86
	150	52,427	14,095	10,533	51,268	2.63	3.50	3.61	4.77
	200	37,841	17,735	14,587	43,551	3.18	3.84	4.38	5.99
	250	36,700	18,669	17,991	27,862	3.73	4.16	5.17	6.99
	300	33,087	23,777	18,919	21,460	4.27	4.52	6.08	8.26

7. Conclusion

This work incorporated niching metrics into both guiding components of C-DEEPSO: the social one via FER-PSO and the cognitive one via FER-DE with Crowding. The ablations bear out the intended division of roles: applied in isolation, neither modification was sufficient—CDEEPSO FER-PSO refined tightly but lacked stability, and CDEEPSO FER-DE explored reliably but at a high evaluation cost—whereas their combination in FER-CDEEPSO achieved a competitive balance. Supported by the Friedman and Nemenyi tests ($p < 0.05$), FER-CDEEPSO proved statistically competitive with the classic baseline (Original FER-DE) in success rate and computational cost, and significantly more precise in niche cohesion for almost all functions, despite its more complex structure.

Despite this precision and statistical robustness, the study was limited to an ad hoc collection of seven low-dimensional functions with centered optima rather than a standardized suite. Future work will therefore adopt the CEC'2013 niching benchmark [Li et al. 2013], whose shifted and rotated landscapes, higher dimensionality and standardized protocol (peak ratio and success rate at multiple accuracy levels) remove the risk that the reported gains stem from the geometric symmetry of the classic functions rather than from the niching mechanism itself. A second limitation concerns the comparison itself: beyond its own partial variants, FER-CDEEPSO was measured against a single external baseline, drawn from the same FER-DE lineage as its cognitive component. This isolates what the hybridization adds, but leaves open where the algorithm stands among current niching methods. The baseline ecosystem will therefore be extended to NCDE [Qu et al. 2012], NEA2 [Preuss 2012], RS-CMSA-ESII [Ahrari et al. 2022] and NMMSO [Fieldsend 2014] (Section 2), covering the neighborhood-based DE, CMA-ES and multi-swarm families rather than the FER line alone.

Further study of the algorithm itself will follow two directions: a mechanism that adjusts the social and cognitive weights dynamically according to the fitness landscape and the swarm's current exploration state, rather than keeping them fixed throughout the run; and a hybrid parallel-sequential FER-CDEEPSO, since purely parallel niching bounds the attainable optima by the population size, whereas multiple restarts applying a spatial penalty around previously identified peaks prevent re-exploring known basins and keep uncovering new optima across runs.

Acknowledgments

This work was partially funded by the Brazilian research funding agencies CNPq — Conselho Nacional de Desenvolvimento Científico e Tecnológico (Grant Number 403964/2023-7), FAPERJ — Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro (Grant Numbers E-26/210.057/2024, E-26/210.549/2024 and E-26/201.345/2024), and CAPES — Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Grant Number 23038.006308/2021-70), and by a UFRJ ALV Grant. The authors also thank UFRJ for the infrastructure made available.

References

Ahrari, A., Deb, K., and Preuss, M. (2017). Multimodal optimization by covariance matrix self-adaptation evolution strategy with repelling subpopulations. *Evolutionary Computation*, 25(3):439–471.

Ahrari, A., Elsayed, S., Sarker, R., Essam, D., and Coello Coello, C. A. (2022). Static and dynamic multimodal optimization by improved covariance matrix self-adaptation evolution strategy with repelling subpopulations. *IEEE Transactions on Evolutionary Computation*, 26(3):527–541.

Brits, R., Engelbrecht, A. P., and van den Bergh, F. (2002a). A niching particle swarm optimizer. In *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL 2002)*, pages 692–696, Singapore.

Brits, R., Engelbrecht, A. P., and van den Bergh, F. (2002b). Solving systems of unconstrained equations using particle swarm optimization. In *IEEE International Conference on Systems, Man and Cybernetics (SMC 2002)*, volume 3, page 6 pp., Hammamet, Tunisia.

Cho, D.-H., Jung, H.-K., and Lee, C.-G. (1999). Induction motor design for electric vehicle using a niching genetic algorithm. In *IEEE International Electric Machines and Drives Conference. IEMDC'99. Proceedings (Cat. No.99EX272)*, pages 290–292.

De Jong, K. A. (1975). *An Analysis Of The Behavior Of A Class Of Genetic Adaptive Systems*. Ph.d. thesis, University of Michigan. ProQuest Dissertation No. 7609381.

Fieldsend, J. E. (2014). Running up those hills: Multi-modal search with the niching migratory multi-swarm optimiser. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 2593–2600. IEEE.

Gaspar-Cunha, A., Takahashi, R., and Antunes, C. H., editors (2012). *Manual de Computação Evolutiva e Metaheurística*. Imprensa da Universidade de Coimbra; Editora da Universidade Federal de Minas Gerais, Coimbra.

Goldberg, D. E. and Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and their Application (ICGA'87)*, pages 41–49, Hillsdale, NJ, USA. L. Erlbaum Associates Inc.

Harik, G. R. (1995). Finding multimodal solutions using restricted tournament selection. In *Proceedings of the 6th International Conference on Genetic Algorithms*, page 24–31, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Jamil, M. and Yang, X.-S. (2013). A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, 4(2):150–194.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4.

Li, X. (2004). Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization. In *Genetic and Evolutionary Computation – GECCO 2004*, pages 105–116, Berlin, Heidelberg. Springer Berlin Heidelberg.

Li, X. (2007). A multimodal particle swarm optimizer based on fitness euclidean-distance ratio. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Com-*

putation, GECCO '07, page 78–85, New York, NY, USA. Association for Computing Machinery.

Li, X., Engelbrecht, A. P., and Epitropakis, M. G. (2013). Benchmark functions for CEC'2013 special session and competition on niching methods for multimodal function optimization. Technical report, Evolutionary Computation and Machine Learning Group, RMIT University, Melbourne, Australia.

Liang, J., Qu, B., Mao, X., Niu, B., and Wang, D. (2014). Differential evolution based on fitness euclidean-distance ratio for multimodal optimization. *Neurocomputing*, 137:252–260. Advanced Intelligent Computing Theories and Methodologies.

Marcelino, C. G., Almeida, P. E. M., Wanner, E. F., Carvalho, L. M., and Miranda, V. (2016). Fundamentals of the C-DEEPSO algorithm and its application to the reactive power optimization of wind farms. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 1547–1554. IEEE.

Miranda, V. and Alves, R. (2013). Differential evolutionary particle swarm optimization (deepso): A successful hybrid. In *2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI & CBIC)*, pages 368–374.

Parrott, D. and Li, X. (2006). Locating and tracking multiple dynamic optima by a particle swarm model using speciation. *IEEE Transactions on Evolutionary Computation*, 10(4):440–458.

Petrowski, A. (1996). A clearing procedure as a niching method for genetic algorithms. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 798–803.

Preuss, M. (2012). Improved topological niching for real-valued global optimization. In *Applications of Evolutionary Computation (EvoApplications 2012)*, volume 7248 of *Lecture Notes in Computer Science*, pages 386–395. Springer.

Qu, B. Y., Suganthan, P. N., and Das, S. (2013). A distance-based locally informed particle swarm model for multimodal optimization. *IEEE Transactions on Evolutionary Computation*, 17(3):387–402.

Qu, B. Y., Suganthan, P. N., and Liang, J. J. (2012). Differential evolution with neighborhood mutation for multimodal optimization. *IEEE Transactions on Evolutionary Computation*, 16(5):601–614.

Shribate, D. M. (2025). Optimization techniques in machine learning: A comprehensive review. *International Journal of Innovative Science and Research Technology (IJISRT)*, 10(3).

Storn, R. and Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359.

Veeramachaneni, K., Peram, T., Mohan, C., and Osadciw, L. A. (2003). Optimization using particle swarms with near neighbor interactions. In *Genetic and Evolutionary Computation — GECCO 2003*, pages 110–121, Berlin, Heidelberg. Springer Berlin Heidelberg.