# Sequential, parallel and consecutive hybrid evolutionary-swarm optimization metaheuristics

Piotr Urbańczyk[2,1][0000−0001−8838−2354], Aleksandra Urbańczyk[1][0000−0002−6040−554X], Magdalena Król[1][0000−0003−0392−0921], Leszek Rutkowski[3][0000−0001−6960−9525], and Marek Kisiel-Dorohinicki[1][0000−0002−8459−1877]

[1] AGH University of Krakow, Al. Mickiewicza 30, 30-059 Krakow, Poland
{purbanczyk,aurbanczyk,magdakrol,doroh}@agh.edu.pl
[2] Jagiellonian University, ul. Gołębia 24, 31-007 Kraków, Poland
piotr.urbanczyk@uj.edu.pl
[3] Systems Research Institute, Polish Academy of Sciences, 6 Newelska Street, 01-447 Warsaw, Poland rutkowski@agh.edu.pl

**Abstract.** The goal of this paper is twofold. First, it explores hybrid evolutionary-swarm metaheuristics that combine the features of PSO and GA in a sequential, parallel and consecutive manner in comparison with their standard basic form: Genetic Algorithm and Particle Swarm Optimization. The algorithms were tested on a set of benchmark functions, including Ackley, Griewank, Levy, Michalewicz, Rastrigin, Schwefel, and Shifted Rotated Weierstrass, across multiple dimensions. The experimental results demonstrate that the hybrid approaches achieve superior convergence and consistency, especially in higher-dimensional search spaces. The second goal of this paper is to introduce a novel consecutive hybrid PSO-GA evolutionary algorithm that ensures continuity between PSO and GA steps through explicit information transfer mechanisms, specifically by modifying GA's variation operators to inherit velocity and personal best information.

**Keywords:** Particle Swarm Optimization · Genetic Algorithm · Hybrid Evolutionary-Swarm Metaheuristics

## 1 Introduction

While numerous hybrid algorithms integrating Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) have been proposed (as reviewed, e.g., in [26,19,21,22]), this work focuses on a systematic comparison of three distinct hybridization modes—sequential, parallel, and a tightly coupled consecutive approach—across a wide range of problem dimensions. To achieve this, we introduce and evaluate a specific implementation of the consecutive approach (PGCHEA) featuring a novel mechanism for preserving search momentum (velocity and personal bests) across the alternating GA steps, addressing a potential information loss in simpler sequential hybrids. We specifically investigate how these different strategies handle the challenges posed by increasing dimensionality.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_15

## 2 Hybrid models of evolutionary and swarm optimization methods

This section briefly introduces the algorithms we have investigated in our research, along with referring to the relevant papers showing the hybrid versions of metaheuristics under consideration.

### 2.1 Genetic Algorithm (GA)

Genetic Algorithms are one of the earliest and perhaps most common computational models inspired by nature and evolution. They were originally developed by John Holland [14] and then later modified by Goldberg [9]. These population-based algorithms encode potential solutions to specific problems as chromosome-like data structures and apply stochastic genetic search operators to these structures to preserve and mute the "genes" they carry. In the general form, the method can be described as:

$$P' = \rho\big(P \cup \mu\left(\sigma\left(P, f\right)\right)\big) \tag{1}$$

where $P$ is a multiset of positions in the search space (solution candidates), called the population, $f$ is a fitness function that evaluates the solution candidates and returns a vector of values that stand for the optimality of each population member, $\mu$ is a composite operator that introduces random variations to a subset of individuals within the population, $\sigma$ is a stochastic selection operator, and $\rho$ denotes the replacement strategy that removes poorly performing individuals.

In our implementation, selection $\sigma$ uses binary tournament: two individuals are randomly drawn, and the fitter one is added to the mating pool. The variation operator $\mu$ applies simulated binary crossover (SBX) and polynomial mutation. SBX recombines two real-valued parents into offspring, while mutation adds diversity by perturbing gene values. These operators generate a new population aimed at improving solution quality. The process repeats until a termination criterion—such as a maximum number of iterations or a desired fitness level—is met. The best individual from the final population is returned as the solution. The algorithm proceeds as follows:

**Algorithm 1:** GA

**Input:** Population size $N$, crossover rate $p_c$, mutation rate $p_m$
**Output:** Best solution found

```
Initialize population P with N individuals;
Evaluate fitness of each individual in P;
while Termination criterion is not met do
    Select parents from population P based on their fitness;
    Apply crossover to selected parents with probability p_c to produce
     offspring;
    Apply mutation to offspring with probability p_m;
    Evaluate fitness of offspring;
    Replace less fit individuals in P with offspring to create new
     population;
end
return the best individual from the final population;
```

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_15

### 2.2 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is an evolutionary computational model inspired by the social behavior of swarms, such as flocks of birds or schools of fish. Introduced by Kennedy and Eberhart [17], PSO is a population-based optimization method similar to Genetic Algorithms (GA). The algorithm begins with a set of randomly generated potential solutions, termed particles, which collectively form the initial population, often termed swarm. In general, the PSO method can be described as:

$$P' = m(P, f) \tag{2}$$

where $P$ represents a multiset of positions, $f$ is the fitness function that evaluates each particle, and $m$ is a stochastic population manipulation function that produces a new population from the current one.

In a $n$-dimensional search space, each particle is represented by the position vector $X_i = (x_{i1}, x_{i2}, \ldots, x_{in})$. The particle also has an associated velocity vector $V_i = (v_{i1}, v_{i2}, \ldots, v_{in})$. The fitness of each particle is evaluated using an objective function, with higher fitness values indicating better solutions. Each particle retains a personal best position $p_i$ and is influenced by the global best position $p_g$ discovered by the entire swarm.

The movement of the particles is guided by the following equations:

$$v'_{ij} = wv_{ij} + c_1 r_1 (p_{ij} - x_{ij}) + c_2 r_2 (p_{gj} - x_{ij}) \tag{3}$$

$$x'_{ij} = x_{ij} + v'_{ij}, \tag{4}$$

where $c_1$ and $c_2$ are acceleration coefficients (sometimes called *cognitive* and *social* coefficients, respectively), $r_1$ and $r_2$ are random values uniformly distributed in the range [0, 1], and $j = 1, \ldots, n$. The velocity equation directs a particle's movement toward both its personal best and the global best positions, encouraging convergence to an optimal solution over successive iterations.

This iterative process of evaluating fitness, updating velocities and positions, and adjusting towards the best solutions continues until a termination criterion is met. The best position found by any particle at the end of the process is returned as the final solution.

**Algorithm 2:** PSO

**Input:** Swarm size $N$, acceleration coefficients $c_1$ and $c_2$, inertia weight $w$
**Output:** Best solution found

```
Initialize population of N particles with random velocities;
Evaluate the fitness of each particle;
while Termination criterion is not met do
    Modify each particle's position and velocity by equations (3) and (4);
    Evaluate the fitness of each particle;
    Update the personal best p_i and global best p_g positions if necessary;
end
return The global best position p_g as the final solution;
```

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_15

### 2.3 PSO-GA Hybrid Algorithms

PSO and GA are both population-based optimization techniques that, while similar in their parallel processing nature, differ fundamentally in their approach to exploring and exploiting the search space. GA uses a "competitive" strategy, where individuals in the population compete for survival, with poorly performing individuals being replaced by offspring generated through crossover and mutation. This selection process allows GA to adaptively fine-tune the search as it evolves. In contrast, PSO operates on a "cooperative" principle, where particles adjust their positions based on their own best-known position and the best-known position of the swarm, without explicitly replacing any individuals. This method can lead to faster convergence in smooth, well-defined search spaces, but may struggle in more complex landscapes where directional guidance is less clear. The philosophical (together with the performance) differences between PSO and GA might be found in [2].

It has been quickly observed that combining these two distinct approaches can yield potent hybrid algorithms, capable of addressing a broader range of complex optimization problems more effectively. Several hybridization strategies have been proposed over the years. The overview of over 20 hybrid PSO-GA algorithms published between 2002 and 2010 can be found in [26], a good and more recent overview may be found in [19,21] and [22]. The diversity of hybridization strategies ranges from simple combinations where one algorithm initializes the population for the other, to more complex schemes where both algorithms are applied in tandem. In most cases, the integration of PSO and GA is executed either sequentially or simultaneously, where PSO typically aids in global exploration, and GA contributes to local exploitation through its crossover and mutation operations. This study provides a comparative analysis of specific implementations of the concepts of sequential and parallel PSO-GA hybridization alongside a novel consecutive approach, focusing particularly on performance scaling with dimensionality.

**Sequential Approach (PGSHEA)** In sequential approaches, the two algorithms are applied one after the other in series. Such an approach, in various forms, can be easily found in the literature [20,23,15,8,1,25,18].

Our implementation within this approach, which we dubbed PSO-GA Sequential Hybrid Evolutionary Algorithm (PGSHEA) after [23], alternates between the two optimization techniques sequentially, where the algorithm begins with one technique and switches to the other at predetermined interval. The initial population of solutions is generated using either PSO or GA (the starting algorithm is parameterized). This set of solutions is then shared between both algorithms. The PSO instance is initialized with parameters such as cognitive ($c1$), social ($c2$) coefficients, and inertia weight ($w$). The GA instance uses standard genetic operators: crossover, mutation, and selection (all three remain the same as in the standard GA implementation described in section 2.1). After a series of GA steps, the solutions are passed to the PSO algorithm, which initializes its population with these solutions and continues the optimization. Similarly, after a

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_15

series of PSO steps, the solutions are transferred to the GA algorithm, which uses them as its starting population for further optimization. During the switch from GA to PSO, particle velocities are typically initialized (e.g., randomly), as GA individuals do not inherently possess velocity. Conversely, when switching from PSO to GA, the velocity information associated with the particles is usually discarded. PGSHEA continuously tracks the global best solution found so far. This solution serves as the $p_g$ for the PSO component and is preserved across switches, ensuring the search does not lose the high-quality solution, although it is not explicitly forced into the population unless selected naturally by the active algorithm's operators. The algorithm continues to alternate between PSO and GA until the termination criterion is met. The final result is the best solution found by any particle or individual in the population after all iterations, which is returned as the output of the algorithm.

**Algorithm 3:** PGSHEA

**Input:** Population size $N$, PSO parameters ($c1$, $c2$, $w$), GA parameters ($p_c$, $p_m$), starting algorithm, swap interval
**Output:** Best solution found

```
Set current algorithm to PSO or GA based on the starting algorithm;
Initialize population with N individuals;
Evaluate fitness of the initial population;
while Termination criterion is not met do
    Perform the current algorithm step on the population;
    Update best global solution found so far;
    if the swap interval is reached then
        Switch between PSO and GA;
    end
end
return The global best position as the final solution;
```

**Parallel Approach (PGPHEA)** In simultaneous or parallel approaches, PSO and GA are run concurrently, and the two algorithms cooperate by sharing information or combining their results. While this approach can also be found quite widely in the literature [11,16], our implementation is based on the ideas from [24,23] and then later presented independently by Gupta and Yadav [12].

Our implementation of this hybridization approach, which we called the PSO-GA Parallel Hybrid Evolutionary Algorithm (PGPHEA) after [23], starts by dividing the population into two subpopulations: one handled by PSO and the other by GA. The subpopulations are initialized independently, with PSO generating its initial set of particles and GA generating its initial population of chromosomes. During each iteration, both PSO and GA execute their respective steps concurrently: PSO updates the velocity and position of each particle based on its personal best and the global best positions and GA applies selection, crossover, and mutation to its population to create a new generation. After both algorithms have completed their steps, the global best solution is updated by

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_15

comparing the best solutions found by PSO and GA. Periodically, after a fixed number of evaluations (determined by the parametrized exchange interval), an exchange of solutions occurs between PSO and GA—the top-performing solutions from the PSO subpopulation are swapped with the top solutions from the GA subpopulation. The exchange typically involves swapping the positional vectors of the top individuals. When solutions move from GA to the PSO subpopulation, initial velocities need to be assigned (e.g., zero or random). When solutions move from PSO to the GA subpopulation, their associated velocities are generally discarded. The algorithm continues to run both PSO and GA in parallel until a predefined termination criterion is met. The best solution found by either PSO or GA during the entire process is returned as the final solution.

**Algorithm 4:** PGPHEA

**Input:** Population size $N$, PSO parameters (*c1, c2, w*), GA parameters ($p_c$, $p_m$), exchange interval, exchange number $N_E$
**Output:** Best solution found
Initialize GA population $P_{GA}$ with $\lceil \frac{N}{2} \rceil$ individuals and PSO population $P_{PSO}$ with $N - \lceil \frac{N}{2} \rceil$ individuals;
Evaluate fitness of both subpopulations;
**while** *Termination criterion is not met* **do**
    Perform PSO step on $P_{PSO}$;
    Perform GA step on $P_{GA}$;
    Synchronize the best global solution found so far;
    **if** *the exchange interval is reached* **then**
        Exchange the top $N_E$ solutions between $P_{PSO}$ and $P_{GA}$;
    **end**
**end**
**return** The global best solution;

**Consecutive Approach (PGCHEA)** One can think of two specific (extreme or degenerated) cases of sequential methods. The first case is the simple two-phase hybrid scheme, where one algorithm is being used to initialize or seed the other. Although this approach is one of the oldest in evolutionary-swarm hybridizations, it has recently been successfully applied to cloud load balancing optimization and other applications [20,25,18]. The other extreme is represented by consecutive approaches, where the two algorithms are applied in a highly interleaved manner—essentially, one step of one algorithm is immediately followed by a step of the other. While this tightly coupled sequential hybridization is less common, several studies have implicitly adopted its principles by closely integrating PSO and GA operations within a single iterative framework [28,6,22].

Building on this idea, we introduce and test a novel PSO-GA Consecutive Hybrid Evolutionary Algorithm (PGCHEA). It operates by alternately applying a PSO step and a GA step to the entire population. The algorithm initializes a population, identifies the initial global best, and then enters this alternating loop. Its core novelty lies in maintaining continuity between PSO and GA steps

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_15

through dedicated information transfer mechanisms, ensuring PSO-specific information (velocity and personal best positions) is preserved and utilized across the GA steps. This continuity is achieved via modified GA variation operators. Crossover is modified to produce offspring that inherit both velocities and personal best positions ($p_i$) from their parents, alongside their positional genetic material. Furthermore, the concept of a personal best position is maintained for each individual regardless of the active algorithm step—a memory mechanism absent in the standard GA. This direct information inheritance distinguishes PGCHEA from a simple sequential hybrid (PGSHEA) with a swap interval of 1, where standard GA operators would typically discard velocity and personal best information, disrupting PSO's momentum.

Throughout the execution, the global best solution ($p_g$) found so far is continually updated and used to guide the search, acting as the reference point in PSO steps. The algorithm continues alternating steps until a termination criterion is met, returning the best solution found.

**Algorithm 5:** PGCHEA

**Input:** Population size $N$, PSO parameters $(c1, c2, w)$, GA parameters $(p_c, p_m)$, starting algorithm
**Output:** Best solution found

```
Set current algorithm to PSO or GA based on the starting algorithm;
Initialize population P with N individuals;
Evaluate fitness of the initial population;
while Termination criterion is not met do
    if current algorithm is PSO then
        Perform PSO algorithm step on the population;
        Switch to GA;
    end
    else
        Perform GA algorithm step with enhanced variation operators;
        Update personal best p_i and global best p_g positions, if necessary;
        Switch to PSO;
    end
end
return The global best position as the final solution;
```

## 3 Experiment and result

### 3.1 Benchmark functions

In order to evaluate the performance of the proposed hybrid algorithms, a set of standard benchmark functions has been selected. The selection includes Rastrigin, Ackley, Griewank, Levy, Michalewicz, Schwefel, and Shifted Rotated Weierstrass function. Each of them presents unique characteristics, such as multiple local minima, varying degrees of complexity, and distinct search domains. The functions

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_15

are tested over different dimensions (namely: **10**, **50**, **100**, **500**, and **1000**). The equations defining each function are given below, with a summary provided later in Table 1.

$$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e \quad \text{(Ackley)}$$

$$f(x) = 1 + \frac{1}{4000}\sum_{i=1}^{n}x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) \quad \text{(Griewank)}$$

$$f(x) = \sin^2(\pi w_1) + \sum_{i=1}^{n-1}(w_i-1)^2\left[1+10\sin^2(\pi w_i+1)\right] + (w_n-1)^2\left[1+\sin^2(2\pi w_n)\right] \quad \text{(Levy)}$$

$$f(x) = -\sum_{i=1}^{n}\sin(x_i)\sin^{2m}\left(\frac{ix_i^2}{\pi}\right) \quad \text{(Michalewicz)}$$

$$f(x) = 10n + \sum_{i=1}^{n}\left[x_i^2 - 10\cos(2\pi x_i)\right] \quad \text{(Rastrigin)}$$

$$f(x) = 418.9829n - \sum_{i=1}^{n}x_i\sin(\sqrt{|x_i|}) \quad \text{(Schwefel)}$$

$$f(x) = \sum_{i=1}^{n}\left(\sum_{k=0}^{20}\left[0.5^k\cos(2\pi\cdot 3^k\cdot(x_i+0.5))\right]\right) - n\sum_{k=0}^{20}\left[0.5^k\cos(2\pi\cdot 3^k\cdot 0.5)\right] \quad \text{(Shifted Rotated Weierstrass)}$$

Table 1: Benchmark functions overview

| **Function Name** | **Search Domain**[a] | **Fitness at Global Minimum**[a] |
|---|---|---|
| **Ackley** | $[-32.768, 32.768]^n$ | $f(0) = 0$ |
| **Griewank** | $[-600, 600]^n$ | $f(0) = 0$ |
| **Levy** | $[-10, 10]^n$ | $f(1) = 0$ |
| **Michalewicz**[b] | $[0, \pi]^n$ | $f(x)$ is known for specific $n$ |
| **Rastrigin** | $[-5.12, 5.12]^n$ | $f(0) = 0$ |
| **Schwefel** | $[-500, 500]^n$ | $f(420.9687) \approx 0$ |
| **Shifted Rotated Weierstrass** | $[-0.5, 0.5]^n$ | Depends on shift and rotation[c] |

[a] $n$ – dimensionality. $n \in \{10, 50, 100, 500, 1000\}$
[b] $m = 10$
[c] The shift vector has been defined as a random vector drawn from a uniform distribution within $[-0.5, 0.5]^n$, the rotation matrix is generated as a random orthogonal matrix.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_15

### 3.2 Experiment setup

The experiment was conducted to evaluate the performance of the PGSHEA, PGPHEA, and PGCHEA algorithms in comparison to the standard GA and PSO across a suite of benchmark functions, as detailed in the previous section (3.1). We adopted GA parameter values widely reported in the literature [10,3,13]. We then conducted a preliminary parameter tuning for the PSO and hybrid algorithms using Bayesian optimization. The resulting parameters are listed in Table 2. Each algorithm was executed 10 times on each benchmark problem, with dimensionality set at $n \in \{10, 50, 100, 500, 1000\}$. The results were averaged across runs.

The population size for all experiments was consistently set to $N = 100$. The termination criterion for each algorithm was based on a maximum number of evaluations and set to 25000 for most cases and 10000 for $n = 10$ dimensions. This adjustment was made for clarity and practical purposes, preventing algorithms from merely converging due to the large number of evaluations, thus providing a more meaningful comparison of their performance.

Table 2: Parameter settings for each algorithm

| **Parameter** | **GA** | **PSO** | **PGSHEA** | **PGPHEA** | **PGCHEA** |
|---|---|---|---|---|---|
| **Crossover Rate ($p_c$)** | 0.9 | - | 1.0 | 1.0 | 1.0 |
| **Mutation Rate ($p_m$)**[a] | $\frac{1.0}{n}$ | - | $\frac{0.38}{n}$ | $\frac{0.37}{n}$ | $\frac{0.61}{n}$ |
| $c_1$ | - | 1.97 | 2.63 | 0.01 | 1.85 |
| $c_2$ | - | 0.94 | 0.21 | 0.26 | 0.5 |
| $w$ | - | 0.56 | 0.01 | 0.17 | 1.53 |
| **Exchange Interval** | - | - | 13 | 13 | - |
| **Exchange Number** | - | - | - | 7 | - |
| **Starting Algorithm** | - | - | PSO | - | PSO |

[a] $n$ – dimensionality. $n \in \{10, 50, 100, 500, 1000\}$

### 3.3 Experiment results

Table 3 presents the comparative performance results of GA, PSO, PGSHEA, PGPHEA, and PGCHEA across various benchmark functions. The table shows the average fitness obtained by each algorithm for each problem after 25000 evaluations (for most dimensions) and 10000 evaluations (for the smallest dimensional cases). The best-performing algorithm for each problem and dimension is highlighted in bold. Detailed results, including convergence plots and performance analysis, can be found in the Figures 1 and 2.

We have systematically performed various statistical testing on the quantitative results we have obtained. First, we have applied the Shapiro-Wilk test with a significance threshold of 0.05 to assess whether the observed samples followed

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_15

Table 3: Experiment results

| Problem | Dim. ($n$) | Eval. | GA | PSO | PGSHEA | PGPHEA | PGCHEA |
|---|---|---|---|---|---|---|---|
| **Ackley** | 10 | 10000 | 0.0812 | **0.0000** | 0.0942 | 0.0185 | 0.2991 |
| | 50 | 25000 | 0.4264 | 4.6518 | 1.6092 | **0.1215** | 2.3408 |
| | 100 | 25000 | 3.2003 | 9.1545 | 5.0698 | **1.8037** | 6.3891 |
| | 500 | 25000 | 18.3729 | 16.5278 | 13.8260 | **8.9708** | 17.6277 |
| | 1000 | 25000 | 19.7747 | 17.4644 | 15.1160 | **11.4876** | 19.4970 |
| **Griewank** | 10 | 10000 | 0.1537 | **0.0654** | 0.1563 | 0.0807 | 0.4173 |
| | 50 | 25000 | 1.0387 | **0.0294** | 1.2411 | 0.7355 | 1.3394 |
| | 100 | 25000 | 2.4290 | 2.7071 | 10.5749 | **1.3514** | 13.1571 |
| | 500 | 25000 | 1877.3427 | 946.6615 | 1085.4084 | **213.7523** | 2267.2059 |
| | 1000 | 25000 | 7815.1528 | 3752.3912 | 3271.2027 | **1078.5572** | 7882.8616 |
| **Levy** | 10 | 10000 | 0.0003 | **0.0000** | 0.0006 | **0.0000** | 0.0019 |
| | 50 | 25000 | 0.3300 | 6.6185 | **0.2859** | 0.3596 | 0.6722 |
| | 100 | 25000 | 11.0033 | 33.9640 | 12.5885 | **0.7897** | 26.2848 |
| | 500 | 25000 | 756.2910 | 436.1920 | 318.3160 | **85.6619** | 933.4690 |
| | 1000 | 25000 | 3006.4498 | 1506.2417 | 1137.4485 | **361.4372** | 3354.4831 |
| **Michalewicz** | 10 | 10000 | **-9.4219** | -8.8826 | -9.3585 | -9.3291 | -9.3618 |
| | 50 | 25000 | -41.2735 | -35.7822 | -41.6125 | **-42.9070** | -40.9484 |
| | 100 | 25000 | -72.6364 | -59.8520 | -74.4890 | **-76.2109** | -69.3091 |
| | 500 | 25000 | -224.7013 | -132.8061 | -211.3493 | **-336.4667** | -175.3285 |
| | 1000 | 25000 | -334.7268 | -200.9745 | -339.0732 | **-501.9018** | -261.8926 |
| **Rastrigin** | 10 | 10000 | **0.1895** | 5.4139 | 2.2492 | 0.4095 | 2.6509 |
| | 50 | 25000 | 28.0475 | 85.3689 | 41.7002 | **14.0602** | 50.8819 |
| | 100 | 25000 | 164.9228 | 265.5786 | 148.5575 | **48.3258** | 221.7512 |
| | 500 | 25000 | 3143.9534 | 3015.7626 | 2291.8820 | **984.8022** | 3665.6620 |
| | 1000 | 25000 | 9010.0454 | 7674.9698 | 6411.1468 | **3710.6077** | 10041.3225 |
| **Schwefel** | 10 | 10000 | 94.9378 | 628.0398 | 166.1491 | 213.2274 | **60.1219** |
| | 50 | 25000 | **1953.1047** | 9153.6703 | 3646.7280 | 2320.6635 | 2186.8971 |
| | 100 | 25000 | **7120.7331** | 19838.0034 | 12580.6665 | 8033.4714 | 8700.4964 |
| | 500 | 25000 | 91144.4005 | 128370.9587 | 121689.7331 | **89486.0379** | 103279.6629 |
| | 1000 | 25000 | 239641.4942 | 295232.4449 | 292685.6819 | **232898.2801** | 254663.7326 |
| **Shifted Rotated Weierstrass** | 10 | 10000 | **2.3408** | 4.1009 | 3.2057 | 3.8153 | 3.6618 |
| | 50 | 25000 | 54.1631 | 45.5086 | 44.7081 | 56.2984 | **41.6800** |
| | 100 | 25000 | 127.7596 | 118.2235 | **112.7674** | 129.1044 | 114.8938 |
| | 500 | 25000 | 812.1367 | 763.0698 | 745.9866 | 788.3598 | **719.0998** |
| | 1000 | 25000 | 1715.7651 | 1633.2264 | 1580.6910 | 1615.9141 | **1575.6579** |

Table 4: Best performing algorithms per problem and dimension

| Problem/Dimension | 10 | 50 | 100 | 500 | 1000 |
|---|---|---|---|---|---|
| **Ackley** | PSO | PGPHEA | PGPHEA | PGPHEA | PGPHEA |
| **Griewank** | PSO | PSO | PGPHEA | PGPHEA | PGPHEA |
| **Levy** | PSO/PGPHEA | PGSHEA | PGPHEA | PGPHEA | PGPHEA |
| **Michalewicz** | GA | PGPHEA | PGPHEA | PGPHEA | PGPHEA |
| **Rastrigin** | GA | PGPHEA | PGPHEA | PGPHEA | PGPHEA |
| **Schwefel** | PGCHEA | GA | GA | PGPHEA | PGPHEA |
| **Shifted Rotated Weierstrass** | GA | PGCHEA | PGSHEA | PGCHEA | PGCHEA |

a normal distribution. The outcomes were mixed, as the null hypothesis was rejected for some samples while it could not be rejected for others. Consequently, we employed the non-parametric Kruskal-Wallis test to determine if the cumulative distribution functions differed among the groups. The results of the test

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_15

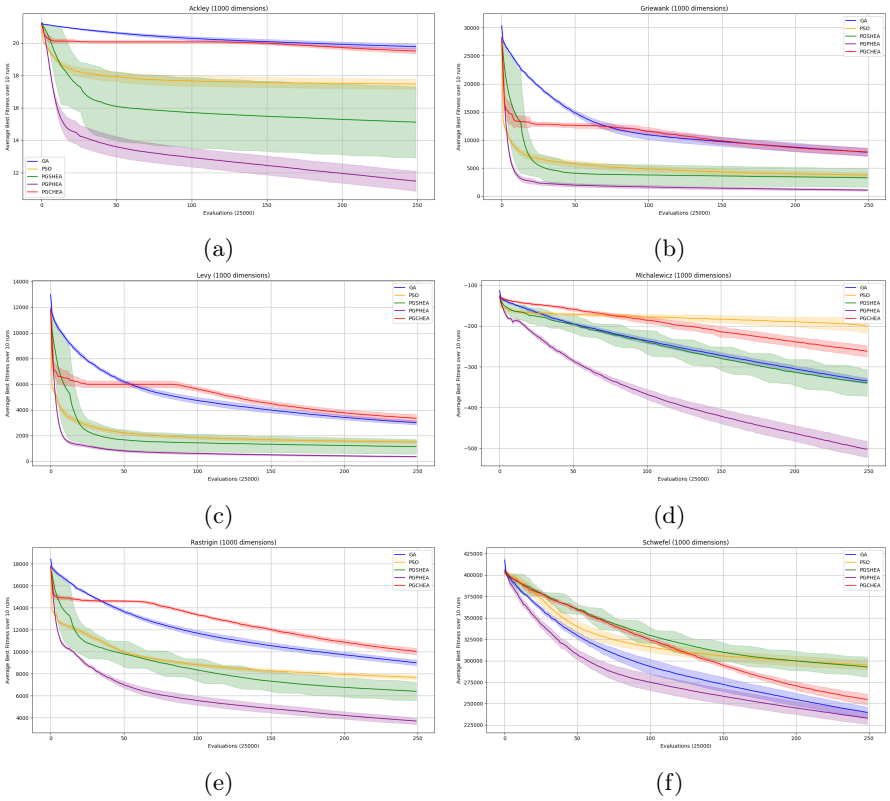


Fig. 1: Convergence analysis of GA (blue), PSO (yellow), PGSHEA (green), PGPHEA (magenta), and PGCHEA (red) on six benchmark functions with 1000 dimensions, evaluated over 25,000 iterations: (a) Ackley, (b) Griewank, (c) Levy, (d) Michalewicz, (e) Rastrigin, and (f) Schwefel.

were statistically significant. This was followed by pairwise comparisons using Dunn's test to identify which pairs exhibited statistically significant differences. Table 5 presents the algorithm pairs that did not show statistically significant differences (assuming the above-mentioned significance level $\alpha$) when compared against the best-performing algorithm for higher-dimensional problems.

## 4 Conclusions

The hybrid algorithms (PGSHEA, PGPHEA, and PGCHEA) generally outperform the standard GA and PSO in most cases, especially as the dimensionality of the problems increases. This suggests that combining the strengths of both GA and PSO within these hybrid frameworks provides a more robust approach to optimization, particularly for complex and high-dimensional problems. Also,

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_15

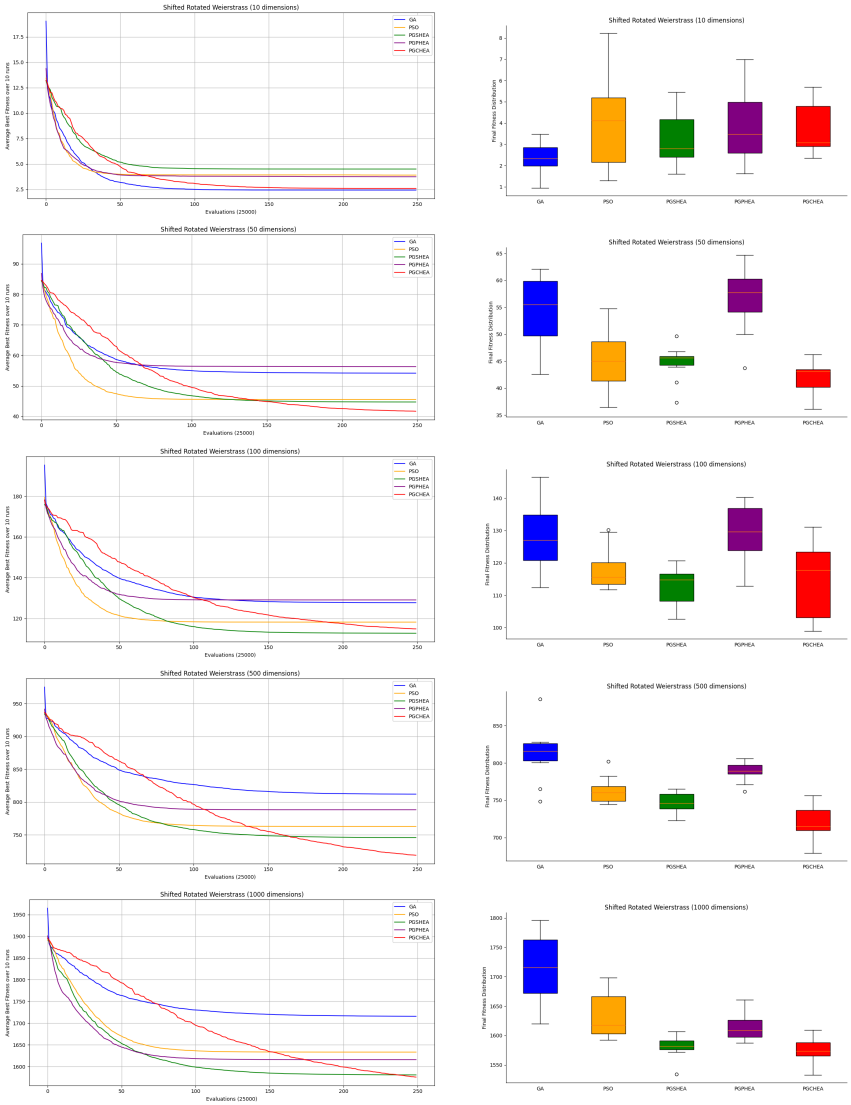


Fig. 2: Convergence plots (left) and final fitness distribution analysis (right) of GA (blue), PSO (yellow), PGSHEA (green), PGPHEA (magenta), and PGCHEA (red) on the Shifted Rotated Weierstrass function for 10, 50, 100, 500, and 1000 dimensions respectively (in top-down order).

the performance gap between the algorithms becomes more pronounced as the number of dimensions increases. For instance, in lower-dimensional problems (e.g., 10 dimensions), GA and PSO sometimes achieve competitive results. However, in higher-dimensional problems (e.g., 500 and 1000 dimensions), the hybrid

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_15

Table 5: Non-significant algorithm pairs with p-values

| **Problem** | **Dimension** | **Algorithm Pair** | **p-value** |
|---|---|---|---|
| Ackley | 100 | GA vs PGPHEA | 0.1084 |
| Griewank | 100 | PSO vs PGPHEA | 0.0666 |
| Michalewicz | 100 | GA vs PGPHEA | 0.1761 |
| Michalewicz | 100 | PGSHEA vs PGPHEA | 0.4625 |
| Schwefel | 100 | GA vs PGPHEA | 0.2442 |
| Schwefel | 100 | GA vs PGCHEA | 0.0906 |
| Shifted Rotated Weierstrass | 100 | PSO vs PGSHEA | 0.4769 |
| Shifted Rotated Weierstrass | 100 | PGSHEA vs PGCHEA | 0.6247 |
| Rastrigin | 500 | PGSHEA vs PGPHEA | 0.1155 |
| Ackley | 500 | PGSHEA vs PGPHEA | 0.0577 |
| Levy | 500 | PGSHEA vs PGPHEA | 0.0577 |
| Michalewicz | 500 | GA vs PGPHEA | 0.0577 |
| Schwefel | 500 | GA vs PGPHEA | 0.6676 |
| Shifted Rotated Weierstrass | 500 | PGSHEA vs PGCHEA | 0.2152 |
| Rastrigin | 1000 | PGSHEA vs PGPHEA | 0.1389 |
| Ackley | 1000 | PGSHEA vs PGPHEA | 0.1038 |
| Schwefel | 1000 | GA vs PGPHEA | 0.4337 |
| Shifted Rotated Weierstrass | 1000 | PGSHEA vs PGCHEA | 0.7590 |

algorithms, particularly PGPHEA, often show superior performance, indicating that the hybrid approaches are better suited to handle the complexity associated with larger search spaces.

The standard evolutionary algorithms perform well in lower dimensions (at least on certain functions), but generally struggle as the problem complexity increases. Particularly, PSO performs well on problems like Levy and Griewank in lower dimensions, where its ability to explore the search space leads to good initial results. However, as indicated by the plots in the appendix, the same exploration tendency usually leads to premature convergence to local minima.

The consistent performance of PGPHEA across a wide range of problems and dimensions suggests that it is the most versatile and robust technique among all tested algorithms. It adapts well to different problem types, making it a strong candidate for general-purpose optimization tasks.

Notably, the novel PGCHEA algorithm, while not the best performer overall, demonstrated particular strength on the complex Shifted Rotated Weierstrass function, highlighting that its unique mechanism for continuous information transfer can be beneficial for certain challenging landscapes with intricate, non-separable, or deceptive structures and warrants further investigation. This result reinforces the "No Free Lunch Theorem" in optimization, which states that no single algorithm can outperform others across all problem types, highlighting the inherent complexity of optimization and underscores the limitations of relying on a single algorithm to excel across diverse problem landscapes.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_15

**Acknowledgments.** The research presented in this paper has been financially supported by: Polish National Science Center Grant no. 2019/35/O/ST6/00570 "Socio-cognitive inspirations in classic metaheuristics"; Polish Ministry of Science and Higher Education funds assigned to AGH University of Science and Technology, program „Excellence initiative – research university" for the AGH University of Krakow. ARTIQ project – Polish National Science Center:DEC-2021/01/2/ST6/00004, Polish National Center for Research and Development, DWP/ARTIQI/426/2023 (MKD)

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## References

1. Aivaliotis-Apostolopoulos, P., Loukidis, D.: Swarming genetic algorithm: A nested fully coupled hybrid of genetic algorithm and particle swarm optimization. PLOS ONE **17**(9), e0275094 (2022). https://doi.org/10.1371/journal.pone.0275094
2. Angeline, P.J.: Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences. In: Proceedings of the Seventh Annual Conference on Evolutionary Programming. pp. 601–610. Springer, Berlin, Heidelberg (1998). https://doi.org/10.1007/BFb0040811
3. Byrski, A.: Tuning of agent-based computing. Computer Science **14**(3), 491–512 (2013). https://doi.org/10.7494/csci.2013.14.3.491
4. Byrski, A., Debski, R., Kisiel-Dorohinicki, M.: Agent-based computing in an augmented cloud environment. Comput. Syst. Sci. Eng. **27**(1) (2012)
5. Byrski, A., Swiderska, E., Lasisz, J., Kisiel-Dorohinicki, M., Lenaerts, T., Samson, D., Indurkhya, B.: Emergence of population structure in socio-cognitively inspired ant colony optimization. Computer Science **19**(1), 81–98 (2018). https://doi.org/10.7494/csci.2018.19.1.2594
6. Chansamorn, S., Somgiat, W.: Improved particle swarm optimization using evolutionary algorithm. In: 2022 19th International Joint Conference on Computer Science and Software Engineering (JCSSE). pp. 319–324. IEEE (2022). https://doi.org/10.1109/JCSSE54890.2022.9836238
7. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science. pp. 39–43. IEEE (1995). https://doi.org/10.1109/MHS.1995.494215
8. Esmin, A.A.A., Lambert-Torres, G., Zambroni de Souza, A.C.: A hybrid Particle Swarm Optimization applied to loss power minimization. IEEE Transactions on Power Systems **20**(2), 859–866 (2005). https://doi.org/10.1109/TPWRS.2005.846049
9. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, MA (1989)
10. Grefenstette, J.J.: Optimization of control parameters for genetic algorithms. IEEE Transactions on Systems, Man, and Cybernetics **16**(1), 122–128 (1986)
11. Grimaldi, E., Grimaccia, F., Mussetta, M., Pirinoli, P., Zich, R.: A new hybrid genetical-swarm algorithm for electromagnetic optimization. In: ICCEA 2004 Proceedings. 3rd International Conference on Computational Electromagnetics and Its Applications. pp. 157–160 (2004)
12. Gupta, M., Yadav, R.: New improved fractional order differentiator models based on optimized digital differentiators. The Scientific World Journal (2014)

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_15

13. Hassanat, A.B., Almohammadi, K., Alkafaween, E., Abunawas, E., Hammouri, A., Prasath, V.B.S.: Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach. Information **10**(12), 390 (2019)
14. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI, 1st edn. (1975)
15. Juang, C.M.: A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **34**(2), 997–1006 (2004)
16. Kao, Y.C., Zahara, E.: A hybrid genetic algorithm and particle swarm optimization for multimodal functions. Applied Soft Computing **8**(2), 849–857 (2008)
17. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks. pp. 1942–1948. IEEE (1995)
18. Palaniappan, S.C., Ponnuswamy, P.P.: Task offloading in edge computing using integrated particle swarm optimization and genetic algorithm. Advances in Science and Technology Research Journal **19**(1), 371–380 (2025)
19. Placzkiewicz, L., Sendera, M., Szlachta, A., Paciorek, M., Byrski, A., Kisiel-Dorohinicki, M., Godzik, M.: Hybrid swarm and agent-based evolutionary optimization. In: International Conference on Computational Science (ICCS) 2018, pp. 90–102. Springer (2018). https://doi.org/10.1007/978-3-319-93701-4_7
20. Robinson, J., Sinton, D., Rahmat-Samii, Y.: Particle swarm, genetic algorithm, and their hybrids: Optimization of a profiled corrugated horn antenna. In: 2002 IEEE Antennas and Propagation Society International Symposium (IEEE Cat. No.02CH37313). vol. 1, pp. 314–317. IEEE (2002)
21. Sengupta, S., Basak, S., Peters, R.A.: Particle Swarm Optimization: A survey of historical and recent developments with hybridization perspectives. Machine Learning and Knowledge Extraction **1**(1), 157–191 (2019)
22. Shao, K., Song, Y., Wang, B.: PGA: A new hybrid PSO and GA method for task scheduling with deadline constraints in distributed computing. Mathematics **11**, 1548 (2023). https://doi.org/10.3390/math11061548
23. Shi, X., Lu, Y., Zhou, C., Lee, H., Liang, Y.: Hybrid evolutionary algorithms based on PSO and GA. In: Congress on Evolutionary Computation (CEC). pp. 2393–2399. IEEE (2003). https://doi.org/10.1109/CEC.2003.1299387
24. Shi, X., Wan, L., Lee, H., Yang, X., Wang, L., Liang, Y.: PSO-GA based hybrid evolutionary algorithm. In: Proceedings of the International Conference on Machine Learning and Cybernetics. pp. 1735–1740. IEEE (2003)
25. Simaiya, S., Lilhore, U.K., Sharma, Y.K., Rao, K.B.V.B., Maheswara Rao, V.V.R., Baliyan, A., Bijalwan, A., Alroobaea, R.: A hybrid cloud load balancing and host utilization prediction method using deep learning and optimization techniques. Scientific Reports **14**(1), 1337 (2024). https://doi.org/10.1038/s41598-024-51466-0
26. Thangaraj, R., Pant, M., Abraham, A., Bouvry, P.: Particle swarm optimization: Hybridization perspectives and experimental illustrations. Applied Mathematics and Computation **217**(13), 5208–5226 (2011)
27. Turek, W., Stypka, J., Krzywicki, D., Anielski, P., Pietak, K., Byrski, A., Kisiel-Dorohinicki, M.: Highly scalable erlang framework for agent-based metaheuristic computing. J. Comput. Sci. **17**, 234–248 (2016). https://doi.org/10.1016/J.JOCS.2016.03.003
28. Yang, B., Chen, Y., Zhao, Z.: A hybrid evolutionary algorithm by combination of PSO and GA for unconstrained and constrained optimization problems. In: Proceedings of the IEEE International Conference on Control and Automation. pp. 166–170 (2007). https://doi.org/10.1109/ICCA.2007.4376340

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_15