Computers & Industrial Engineering 145 (2020) 106559

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

A mayfly optimization algorithm

Konstantinos Zervoudakis*, Stelios Tsafarakis

School of Production Engineering and Management, Technical University of Crete, Chania, Greece

ARTICLE INFO

Keywords:
Mayfly algorithm
Nature-inspired algorithm
Optimization

ABSTRACT

This paper introduces a new method called the Mayfly Algorithm (MA) to solve optimization problems. Inspired from the flight behavior and the mating process of mayflies, the proposed algorithm combines major advantages of swarm intelligence and evolutionary algorithms. To evaluate the performance of the proposed algorithm, 38 mathematical benchmark functions, including 13 CEC2017 test functions, are employed and the results are compared to those of seven state of the art well-known metaheuristic optimization methods. The MA's performance is also assessed through convergence behavior in multi-objective optimization as well as using a real-world discrete flow-shop scheduling problem. The comparison results demonstrate the superiority of the proposed method in terms of convergence rate and convergence speed. The processes of nuptial dance and random flight enhance the balance between algorithm's exploration and exploitation properties and assist its escape from local optima.

1. Introduction

Optimization is a process of finding the best solution of a function (either its minimum or its maximum value). Numerous real-world problems are represented as optimization problems, through a formulation as follows (for single-objective minimization)

$$\text{Minimize}\quad f(\mathbf{x}),\ \mathbf{x} = [x_1, x_2, \cdots, x_N] \tag{1}$$

$$\text{Subject to: } g_i(\mathbf{x}) \geq 0,\ i = 1, 2, \cdots, m \tag{2}$$

$$h_i(\mathbf{x}) = 0,\ i = 1, 2, \cdots, p \tag{3}$$

$$lb_i \leq x_i \leq ub_i,\ i = 1, 2, \cdots, N \tag{4}$$

where f is a given function $f: A \to \mathbb{R}$ from set A to real numbers. The domain A is a subset of the Euclidean space $\mathbb{R}^n$ and represents the search space, and x_s are the problem variables. f is generally called an objective function, or alternatively cost or loss function for minimization, and fitness or utility function for maximization. N is the number of variables, p is the number of equality constraints, m is the number of inequality constraints (g), lbi is the lower bound of the i^{th} variable, and ubi is its upper bound. For multi objective minimization, equation (1) is formulated as follows:

$$\text{Minimize}\quad F(\mathbf{x}),\ \mathbf{x} = \{f_1(\mathbf{x}), f_1(\mathbf{x}), \cdots, f_o(\mathbf{x})\} \tag{5}$$

where o is the number of objectives (Mirjalili et al., 2017). In multi-objective optimization problems, the aim is to find a set of non-dominated solutions, known as the Pareto front (Knowles & Corne, 2002; Mirjalili et al., 2017; Zitzler, Deb, & Thiele, 2000).

The early methods used for solving optimization problems were mathematical or numerical methods in which the final solution is obtained by reaching a zero-derivative point. However, solving a non-linear non-convex problem with lots of variables and constraints using these methods is almost impossible, because as the number of dimensions increases, the search space also increases exponentially. Besides that, the numerical methods may be stuck at a local optimum point at which the derivative is also zero. As a consequence, there is no guarantee to find the global optimum solution by using such numerical methods (Nematollahi, Rahiminejad, & Vahidi, 2017).

To overcome the drawbacks of numerical methods, more sophisticated metaheuristic algorithms have been widely used to solve complex optimization problems. The advantages of these methods are that they are easily applied on both continuous and discrete functions, they do not need any additional complex mathematical operations such as derivatives, and they rarely get stuck in local optimum points.

The meta-heuristic methods are classified into single solution-based methods and population-based methods. In single solution-based methods, a solution is generated (usually at random) and keeps getting improved until a stopping criterion is satisfied. In population-based methods, a set of solutions is generated (randomly) in the predefined search space and is updated iteratively to find a (near) optimal solution based on the information interaction among solutions.

The population-based methods can be further divided into two main

* Corresponding author.
E-mail addresses: kzervoudakis@isc.tuc.gr (K. Zervoudakis), tsafarakis@dpem.tuc.gr (S. Tsafarakis).

https://doi.org/10.1016/j.cie.2020.106559
Received 25 September 2019; Received in revised form 22 April 2020; Accepted 15 May 2020
Available online 25 May 2020
0360-8352/

Table 1
Nature-inspired optimization algorithms developed during the past two years.

Algorithm	Inspiration	Year
Shark smell optimization (SSO) (Abedinia, Amjady, & Ghasemi, 2016)	Ability of shark in finding its prey by smell sense	2016
Dolphin swarm optimization algorithm (DSOA) (Yong, Tao, Cheng-Zhi, & Hua-Juan, 2016)	Mechanism of dolphins in swarms	2016
Virus colony search (Li, Zhao, Weng, & Han, 2016)	Virus infection and diffusion strategies	2016
Whale optimization algorithm (WOA) (Mirjalili & Lewis, 2016)	Social behavior of humpback whales	2016
Multi-verse optimizer (MVO) (Mirjalili, Mirjalili, & Hatamlou, 2016)	Multi-verse theory	2016
Crow search algorithm (CSA) (Askarzadeh, 2016)	Food hiding behavior of crows	2016
Salp swarm algorithm (Mirjalili et al., 2017)	Swarming behavior of salps	2017
Grasshopper optimization algorithm (Saremi, Mirjalili, & Lewis, 2017)	Swarming behavior of grasshoppers	2017
Selfish herd optimizer (SHO) (Fausto, Cuevas, Valdivia, & González, 2017)	Hamilton's selfish herd theory	2017
Electro-search algorithm (Tabari & Ahmad, 2017)	Orbital movement of the electrons	2017
Thermal exchange optimization (Kaveh & Dadras, 2017)	Newton's law of cooling	2017
Weighted superposition attraction (WSA) (Baykasoğlu & Akpinar, 2017)	Superposition principle	2017
Spotted hyena optimizer (Dhiman & Kumar, 2017)	Social behavior of spotted hyenas	2017
Butterfly-inspired algorithm (Qi, Zhu, & Zhang, 2017)	Mate searching mechanism of butterfly	2017
Lightning attachment procedure optimization (Nematollahi et al., 2017)	Lightning attachment process	2017
Mouth brooding fish algorithm (Jahani & Chizari, 2018)	Life cycle of mouth brooding fish	2018
Find-Fix-Finish-Exploit-Analyze algorithm (Husseinzadeh Kashan, Tavakkoli-Moghaddam, & Gen, 2019)	Find-Fix-Finish-Exploit-Analyze (F3EA) targeting process	2019
Booster optimization algorithm (Pakzad-Moghaddam, Mina, & Mostafazadeh, 2019)	Human intelligent behavior in exchange markets	2019

Mayfly Algorithm

Objective function f(**x**), **x**=$(x_1, \dots, x_d)^T$

Initialize the male mayfly population $x_i (i = 1,2,\dots,N)$ *and velocities* v_{mi}

Initialize the female mayfly population $y_i (i = 1,2,\dots,M)$ *and velocities* v_{fi}

Evaluate solutions

Find global best $gbest$

Do While *stopping criteria are not met*

Update velocities and solutions of males and females

Evaluate solutions

Rank the mayflies

Mate the mayflies

Evaluate offspring

Separate offspring to male and female randomly

Replace worst solutions with the best new ones

Update $pbest$ *and* $gbest$

end while

Postprocess results and visualization.

Fig. 1. Pseudo code of the mayfly algorithm (MA).

categories. The first category is the swarm intelligence optimization algorithms that are based on the collective social behavior of insect or animal groups. Particle swarm optimization (PSO) (Eberhart & Kennedy, 1995), firefly algorithm (FA) (Yang, 2008) and ant colony optimization (ACO) (Dorigo, Maniezzo, & Colorni, 1996) are some very popular swarm intelligent methods. The second category is the evolutionary algorithms (EAs) that were inspired from natural genetic evolution. At each generation, the best individuals of the current population survive and produce offspring similar to them in terms of characteristics. Genetic algorithm (GA) (Goldberg, 1989) and differential evolution (DE) (Storn & Price, 1997) are some very popular EAs. Based on the principles of Darwinian and evolution theory of living beings, these algorithms follow specialized recombination, crossover, mutation, selection, and adaptation strategies.

However, since optimization problems in different fields are widely being evolved and enlarged, new heuristic optimization algorithms must be introduced to advance the field of computational intelligence optimization. Even though various optimization algorithms are introduced in the literature, based on No-Free-Lunch (NFL) theorem (Wolpert & Macready, 1997), there is no guarantee that an optimization algorithm could solve every kind of optimization problem, that is an optimization method may have satisfied results for some problems, but not for some others. As a result, the motivation of this study is to introduce a new optimization method inspired from the flight behavior of mayflies as well as their mating process. A full description of the proposed method is explained in Section 3. The validation and comparison of the MA with other very popular optimization methods are presented in Section 4.

2. Popular and recent optimization methods

Most optimization problems are very hard to solve. Especially the problems that are classified as NP-hard ones, are impossible to be solved by classic optimization methods in polynomial time (Knuth, 1974). To achieve a satisfactory solution, researchers have developed optimization methods either by modifying/hybridizing existing algorithms (Báez, Angel-Bello, Alvarez, & Melián-Batista, 2019; Hakli & Ortacay, 2019; Kaveh & Mesgari, 2019; Laha & Gupta, 2018; Peng, Pan, Gao, Zhang, & Pang, 2018; Vlašić, Đurasević, & Jakobović, 2019) or developing new ones. Some efficient, widely used methods are described in the following.

2.1. Particle swarm optimization (PSO)

PSO (Kennedy & Eberhart, 1995) is a population-based swarm intelligence algorithm based on the swarm behavior of birds or fish, which is used to solve continuous optimization problems. In PSO, a problem solution is represented by the position of a particle that moves into the search space. Each particle changes its position according to its own experience and that of its neighbors. The position of a particle is changed by adding a velocity to its current position. Each particle's velocity depends on its previous best position and the previous global best position of the swarm. PSO's advantages are that it does not require the calculation of derivatives, the knowledge of good solutions is retained by all particles (memory) and that the particles in the swarm share information among them. Furthermore, PSO is less sensitive to the nature of the objective function. As a result, PSO can be used for stochastic objective functions since it can easily escape from local optima (Tsafarakis, Marinakis, & Matsatsinis, 2011).

Multi-objective Mayfly Algorithm

Initialize the male mayfly population $x_i (i = 1,2,\ldots,N)$ *and velocities* v_{mi}

Initialize the female mayfly population $y_i (i = 1,2,\ldots,M)$ *and velocities* v_{fi}

Evaluate solutions using the predefined objective functions

Store the nondominated solutions found in an external repository

Sort the mayflies

Do While *stopping criteria are not met*

Update velocities and positions of males and females

Evaluate solutions

If a new mayfly dominates its personal best

Replace personal best with the new solution

If no one dominates the other

The new solution has a chance of 50% to replace the personal best

Rank the mayflies

Mate the mayflies

Evaluate offspring

Separate offspring to male and female randomly

If an offspring dominates its same-sex parent

Replace parent with the offspring

Insert all the new nondominated solutions found in the external repository

Sort the nondominated solutions and truncate the repository if needed

end while

Postprocess results and visualization of non-dominated solutions

Fig. 2. Pseudo code of the multi-objective mayfly algorithm (MMA).

Table 2
Unimodal test functions.

Function ID	Name	Expression	Search space	min
F1	Sphere	$f_1(\mathbf{x}) = \sum_{i=1}^{d} x_i^2,$	[−10, 10]	0
F2	Rosenbrock	$f_2(\mathbf{x}) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i^2)^2]$	[−5, 10]	0
F3	Sum Squares	$f_3(\mathbf{x}) = \sum_{i=1}^{d} ix_i^2$	[−10, 10]	0
F4	Powell Sum	$f_4(\mathbf{x}) = \sum_{i=1}^{d} \lvert x_i \rvert^{i+1}$	[−1, 1]	0
F5	Exponential	$f_5(\mathbf{x}) = -e^{-0.5\sum_{i=1}^{d} x_i^2}$	[−1, 1]	−1
F6	Schwefel 2.20	$f_6(\mathbf{x}) = \sum_{i=1}^{d} \lvert x_i \rvert$	[−100, 100]	0
F7	Schwefel 2.21	$f_7(\mathbf{x}) = \max_{i=1,\cdots d} \lvert x_i \rvert$	[−100, 100]	0
F8	Schwefel 2.22	$f_8(\mathbf{x}) = \sum_{i=1}^{d} \lvert x_i \rvert + \prod_{i=1}^{d} \lvert x_i \rvert$	[−100, 100]	0
F9	Zakharov	$f_9(\mathbf{x}) = \sum_{i=1}^{d} x_i^2 + (\sum_{i=1}^{d} 0.5ix_i)^2 + (\sum_{i=1}^{d} 0.5ix_i)^4$	[−5, 10]	0

Table 3
Multimodal test functions.

Function ID	Name	Expression	Search space	min
F10	Rastrigin	$f_{10}(\mathbf{x}) = 10d + \sum_{i=1}^{d}(x_i^2 - 10cos(2\pi x_i))$	[−5.12, 5.12]	0
F11	Ackley	$f_{11}(\mathbf{x}) = 20 + e - 20exp\left[-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d}x_i^2}\right] - exp\left[\frac{1}{d}\sum_{i=1}^{d}cos(2\pi x_i)\right]$	[−32, 32]	0
F12	Griewank	$f_{12}(\mathbf{x}) = 1 + \sum_{i=1}^{d}\frac{x_i^2}{4000} - \prod_{i=1}^{d}cos\left(\frac{x_i}{\sqrt{i}}\right)$	[−600, 600]	0
F13	Alpine No.1	$f_{13}(\mathbf{x}) = \sum_{i=1}^{d}\lvert x_i \sin x_i + 0.1x_i\rvert$	[0, 10]	0
F14	Salomon	$f_{14}(\mathbf{x}) = 1 - \cos\left(2\pi\sqrt{\sum_{i=1}^{d}x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{d}x_i^2}$	[−100, 100]	0
F15	Qing	$f_{15}(\mathbf{x}) = \sum_{i=1}^{d}(x^2 - i)^2$	[−500, 500]	0
F16	Styblinski-Tank	$f_{16}(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{d}(x_i^4 - 16x_i^2 + 5x_i)$	[−5, 5]	−39.16599*d
F17	Xin-She Yang Function	$f_{17}(\mathbf{x}) = \sum_{i=1}^{d}rand[0, 1] * \lvert x_i\rvert^i$	[−5, 5]	0
F18	Quartic	$f_{18}(\mathbf{x}) = \sum_{i=1}^{d}ix_i^4 + random[0, 1)$	[−1.28, 1.28]	0 + random noise

2.2. Firefly algorithm (FA)

The FA is a population-based algorithm that was introduced in 2008 by Xin-She Yang (2008) and was based on the flashing patterns and behavior of fireflies. The FA is used to solve continuous optimization problems, and for some applications its success rate and efficiency are better than those of PSO and GA, in both continuous and discrete problems (Yang, 2009). The standard FA involves two important issues, the variation of light intensity (I), and the formulation of attractiveness (β). Each firefly is characterized by the light intensity, which is related to the value obtained from the fitness function in that point. Fireflies with worse light intensity are attracted by the ones with better light intensity, until all fireflies are gathered in a single solution (local or global optimum). The attractiveness between fireflies is formulated as a function of the square of distance (r) between each other and the light absorption coefficient (γ). As the fireflies search for the best solution, their movements are updated based on their current position, their attractiveness, and a randomization term. When γ tends to be zero, the FA corresponds to the standard PSO (Yang, 2009). Studies have shown that FA performs better than other algorithms because of its automatic subdivision and its capability in dealing with multimodality (Yang & He, 2013).

2.3. Genetic algorithm (GA)

GA was introduced in the 1960s by Holland and further analyzed by Goldberg (1989). GA is a global search optimization technique that imitates processes from natural evolution, based on the survival and reproduction of the fittest. GA has shown great performance in many complex optimization problems and it keeps getting improved by researchers. In GA, solutions are usually coded as binary or integer strings called chromosomes, which evolve over the algorithm's iterations through genetic operations like crossover and mutation. The chromosomes are being evaluated using an objective function. Should the new chromosomes turn out to be better than the old ones, they will replace the worse chromosomes in the next generation (migration strategy). The evolution process continues until the stopping criteria are reached.

2.4. Differential evolution (DE)

Differential evolution algorithm (DE) developed by Storn and Price (1997) is another population-based evolutionary algorithm for continuous optimization problems. In this algorithm, solution vectors are initially generated, and are subjected to mutation, crossover and selection operations. In the mutation operation, mutated vectors are obtained through adding the difference between two arbitrarily selected vectors of the solution space to a selected current solution. In the crossover operation, parameters of the newly generated solution are selected from either the mutated individual or the non-mutated individuals. Finally, there is a selection operation for selecting either a new parent vector or the individuals that will survive to the next generation.

2.5. Harmony search (HS)

The HS is another example of a popular successful algorithm developed by Zong Woo Geem, Joong Hoon Kim, and Loganathan (2001). This algorithm mimics the improvisation of musicians. Although harmony search algorithm is similar to genetic algorithm, it forms new individuals in a different way, by using all individuals in the crossover operation (Uymaz, Tezel, & Yel, 2015). The algorithm starts with the formation of harmony memory composed by randomly generated solutions. In order to create a new solution, the harmony memory can be chosen randomly or from one of the existing solutions. Each chosen solution is subject to a change depending on the pitch adjustment rate. Should a new harmony vector produce better results than the worst harmony, the worst harmony vector will be replaced by the new one.

2.6. Invasive weed optimization (IWO)

Mehrabian and Lucas (2006) developed a numerical stochastic optimization algorithm inspired from colonizing weeds. Weeds are plants whose vigorous, invasive habits of growth pose a serious threat to desirable, cultivated plants making them a threat for agriculture. Weeds have shown to be very robust and adaptive to change in environment. This algorithm tries to mimic robustness, adaptation and randomness of colonizing weeds in a simple but effective optimizing algorithm known as Invasive Weed Optimization (IWO).

2.7. Bees algorithm (BeA)

Bees Algorithm (BeA) is a metaheuristic optimization algorithm proposed by Pham, Ghanbarzadeh, Koç, Otri, Rahim, and Zaidi (2006). It is inspired by the food foraging behavior of honey bee colonies. In this algorithm, bees, each one representing a problem solution, communicate with each other by using a waggle dance. Bees that produce better results have more opportunity to do waggle dance. Hence, they attract more bees to go to their location and target. This helps the algorithm to investigate the unsearched areas in the search space.

Table 4
Fixed-dimension test functions.

Function ID	Name	Category	Expression	Dimension	Search space	min
F19	Eggcrate	Multimodal	$f_{19}(\mathbf{x}) = x_1^2 + x_2^2 + 25(sin^2 x_1 + sin^2 x_2)$	2	[−5, 5]	0
F20	Beale	Unimodal	$f_{20}(\mathbf{x}) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$	2	[−4.5, 4.5]	0
F21	Leon	Unimodal	$f_{21}(\mathbf{x}) = 100(x_2 - x_1^3)^2 + (1 - x_1)^2$	2	[0, 10]	0
F22	Bohachevsky No. 2	Multimodal	$f_{22}(\mathbf{x}) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1)\cos(4\pi x_2) + 0.3$	2	[−100, 100]	0
F23	Easom	Multimodal	$f_{23}(\mathbf{x}) = -\cos(x_1)\cos(x_2)e^{-(x_1-\pi)^2-(x_2-\pi)^2}$	2	[−100, 100]	−1
F24	Three-Hump Camel	Multimodal	$f_{24}(\mathbf{x}) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6}x_1x_2 + x_2^2$	2	[−5, 5]	0
F25	Colville	Multimodal	$f_{25}(\mathbf{x}) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1)$	4	[−10, 10]	0

2.8. *Recently developed optimization methods*

Despite the exceptional performance of the optimization methods mentioned above, researchers keep developing new methods to achieve better results in terms of accuracy and efficiency, both in established and in new problems that are being formulated. Table 1 shows some of the recently developed nature-inspired optimization algorithms.

3. The proposed mayfly algorithm

3.1. *Behavior of mayflies*

Mayflies are insects that belong to the order Ephemeroptera, which is part of an ancient group of insects called Palaeoptera. It is estimated that there are over 3000 species of mayflies worldwide. Their name derives from the fact that they appear mainly during May in the UK.

After hatching from the egg, immature mayflies are visible to the naked eye and they spend several years growing as aquatic nymphs, until they are ready to ascend to the surface as adults. An adult mayfly lives only for a couple of days, until it fulfills its final goal to breed (Domínguez, 2006; Mccafferty, 1991).

To attract females, most of male adults congregate in swarms a few meters above water, performing a nuptial dance, through characteristic up-and-down patterns of movement. Females fly into these swarms, in order to mate with a male in the air. Mating may last just a few seconds and when it is completed females drop their eggs onto the surface of water, and their life cycle goes on (Allan & Flecker, 1989; Peckarsky, McIntosh, Caudill, & Dahl, 2002; Spieth, 1940).

3.2. *Mayfly algorithm*

The proposed optimization method can be considered as a modification of PSO (Kennedy & Eberhart, 1995) and combines major advantages of PSO, GA (Goldberg, 1989) and FA (Yang, 2008). In fact, it offers a powerful hybrid algorithmic structure, based on the behavior of mayflies, for researchers who try to advance the performance of the PSO algorithm using techniques such crossover (Mansouri, Mohammad Hasani Zade, & Javidi, 2019) and local search (Zhou, Pang, Chen, & Chou, 2018), since it has been proved that PSO needs some modifications, to guarantee an optimum point, when performing in high-dimensional spaces (Chen & Shi, 2019). Previously modified optimization algorithms that combine advantages of existing ones are reported in the literature (Haddad, Afshar, & Mariño, 2006; Yang, 2010). It is inspired from the social behavior of mayflies, and particularly from their mating process. We assume that after hatching from the egg, mayflies are already adults and fittest mayflies survive, regardless of how long they live. The position of each mayfly in the search space represents a potential solution to the problem. The algorithm works as follows. Initially, two sets of mayflies are randomly generated, representing the male and female population respectively. That is, each mayfly is randomly placed in the problem space as a candidate solution represented by a d-dimensional vector $\mathbf{x}=(x_1,\cdots,x_d)$, and its performance is evaluated on the predefined objective function $f(\mathbf{x})$. The velocity $\mathbf{v} = (v_1,\cdots,v_d)$of a mayfly is defined as the change of its position, and the flying direction of each mayfly is a dynamic interaction of both individual and social flying experiences. In particular, each mayfly adjusts its trajectory toward its personal best position (*pbest*) so far, as well as the best position attained by any mayfly of the swarm so far (*gbest*).

3.2.1. *Movement of male mayflies*

Males' gathering in swarms, implies that the position of each male mayfly is adjusted according to both its own experience and that of its neighbors. Assuming x_i^t is the current position of mayfly i in the search space at time step t, the position is changed by adding a velocity v_i^{t+1}, to the current position. This can be formulated as

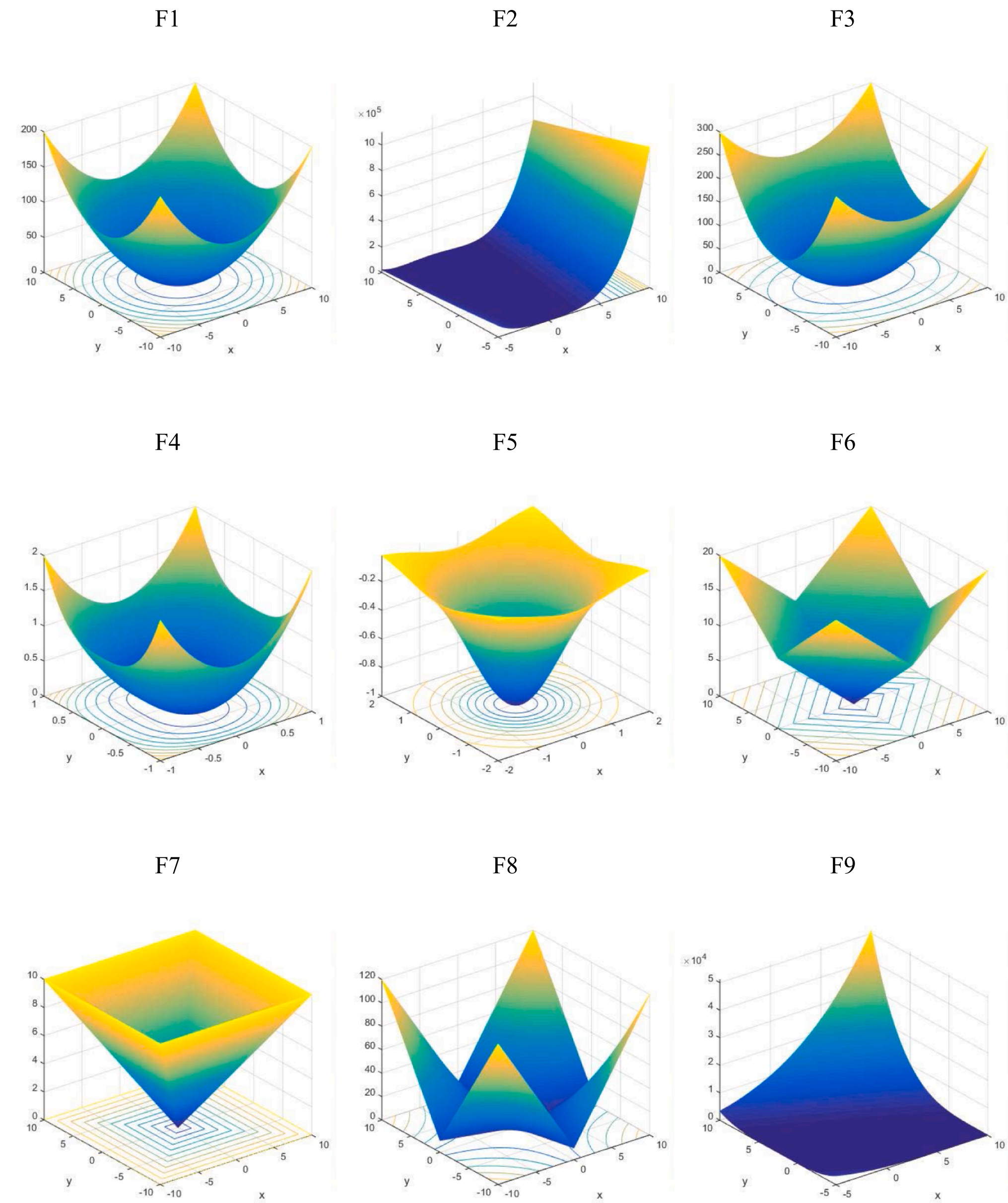

Fig. 3. 2-D version of unimodal test functions in Table 2.

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{6}$$

with $x_i^0\ U(x_{min}, x_{max})$.

Considering that male mayflies are always a few meters above water performing the nuptial dance, we assume that they cannot develop great speeds and that they move constantly. As a result, the velocity of a male mayfly i is calculated as

$$v_{ij}^{t+1} = v_{ij}^t + a_1 e^{-\beta r_p^2}(pbest_{ij} - x_{ij}^t) + a_2 e^{-\beta r_g^2}(gbest_j - x_{ij}^t) \tag{7}$$

where v_{ij}^t is the velocity of mayfly i in dimension $j = 1, ..., n$ at time step t, x_{ij}^t is the position of mayfly i in dimension j at time step t, a_1 and a_2 are positive attraction constants used to scale the contribution of the cognitive and social component respectively. Furthermore, $pbest_i$ is the best position mayfly i had ever visited. Considering minimization problems, the personal best position $pbest_{ij}$ at the next time step $t + 1$, is calculated as

$$pbest_i = \begin{cases} x_i^{t+1}, & \text{if } f(x_i^{t+1}) < f(pbest_i) \\ \textit{is kept the same}, & \textit{otherwise} \end{cases} \tag{8}$$

where $f: \mathbb{R}^n \to \mathbb{R}$ is the objective function, which evaluates the quality of a solution. The global best position $gbest$ at time step t, is defined as

$$\begin{aligned} gbest &\in \{pbest_1, pbest_2, \cdots, pbest_N | f(cbest)\} \\ &= \min\{f(pbest_1), f(pbest_2), \cdots, f(pbest_N)\} \end{aligned} \tag{9}$$

where N is the total number of male mayflies in the swarm. Finally, β is a fixed visibility coefficient used in Eq. (7), used to limit a mayfly's visibility to others, while r_p is the Cartesian distance between x_i and $pbest_i$ and r_g is the Cartesian distance between x_i and $gbest$. These distances are calculated as

$$\|x_i - X_i\| = \sqrt{\sum_{j=1}^{n} (x_{ij} - X_{ij})^2} \tag{10}$$

where x_{ij} is the j^{th} element of mayfly i and X_i corresponds to $pbest_i$ or $gbest$.

It is important for the functioning of the algorithm that the best mayflies in the swarm continue to perform their characteristic up-and-down nuptial dance. Hence, the best mayflies have to keep changing

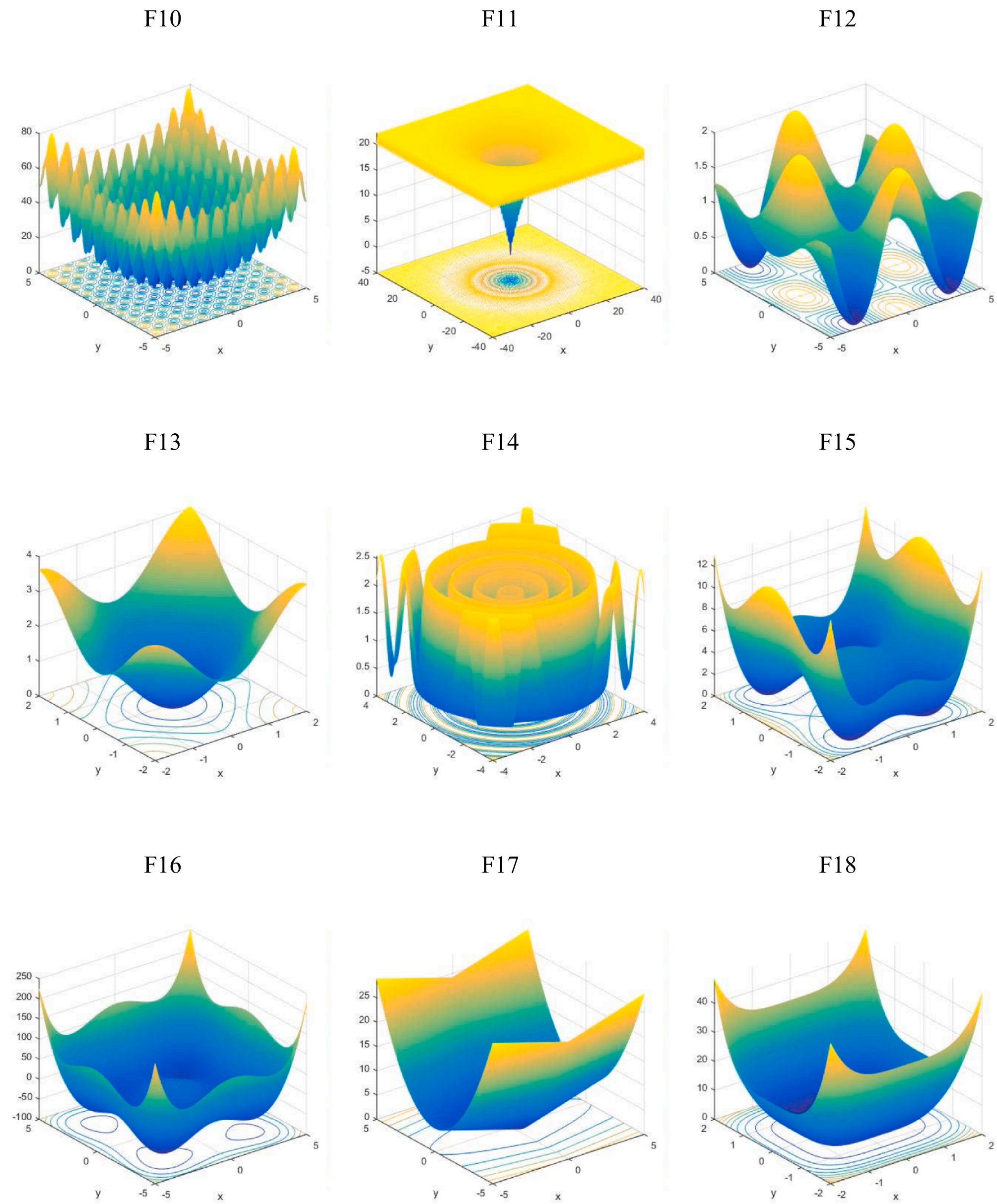

Fig. 4. 2-D version of multimodal test functions in Table 3.

their velocities, which in such a case is calculated as

$$v_{ij}^{t+1} = v_{ij}^{t} + d * r \tag{11}$$

where d is the nuptial dance coefficient and r is a random value in the range $[-1, 1]$. This up and down movement introduces a stochastic element to the algorithm.

3.2.2. Movement of female mayflies

Unlike males, female mayflies do not gather in swarms. They instead fly towards males in order to breed. Assuming y_i^t is the current position of female mayfly i in the search space at time step t, the position is changed by adding a velocity v_i^{t+1} to the current position, i.e.

$$y_i^{t+1} = y_i^{t} + v_i^{t+1} \tag{12}$$

with $y_i^0 \ U(y_{min}, y_{max})$.

Whereas the attraction process could be randomized, we decided to model it as a deterministic process. That is, according to their fitness function, the best female should be attracted by the best male, the second best female by the second best male, and so on. Consequently, considering minimization problems, their velocities are calculated as

$$v_{ij}^{t+1} = \begin{cases} v_{ij}^{t} + a_2 e^{-\beta r_{mf}^2}(x_{ij}^{t} - y_{ij}^{t}), & \text{if } f(y_i) > f(x_i) \\ v_{ij}^{t} + fl * r, & \text{if } f(y_i) \le f(x_i) \end{cases} \tag{13}$$

where v_{ij}^t is the velocity of female mayfly i in dimension $j = 1, \ldots, n$ at time step t, y_{ij}^t is the position of female mayfly i in dimension j at time step t, a_2 is a positive attraction constant and β is a fixed visibility coefficient, while r_{mf} is the Cartesian distance between male and female mayflies, calculated using Eq. (10). Finally, fl is a random walk coefficient, used when a female is not attracted by a male, so it flies randomly and r is a random value in the range $[-1, 1]$.

3.2.3. Mating of mayflies

The crossover operator represents the mating process between two mayflies as follows: One parent is selected from the male population and one from the female population. The way parents are selected is the same as the way females are attracted by males. Particularly, the selection can be either random or based on their fitness function. In the

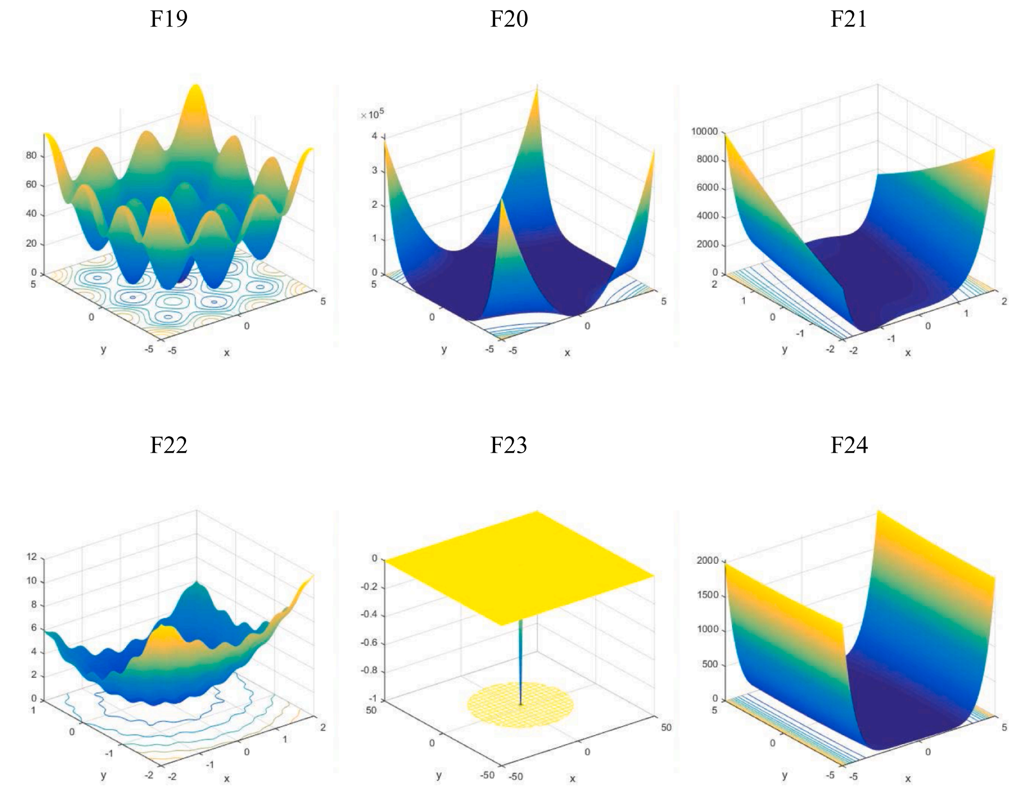

Fig. 5. 2-D version of fixed-dimension test functions in Table 4.

Table 5
Performance of basic MA with different parametres according to Sphere function at 5 dimensions.

a_1	a_2	β	d	fl	Basic MA	
					Min	Average
2	2	2	2	2	6.1214E-05	1.5654E-04
1.5	1.5	2	2	2	7.5804E-216	1.4688E-204
1.5	1.5	2	0.1	0.1	3.3428E-212	2.1144E-204
1	1.5	2	1	1	0	4.864e-319
1	1.5	2	0.1	0.1	0	0

latter, the best female breeds with the best male, the second best female with the second best male, and so on. The results of the crossover are two offspring which are generated as follows:

$$\begin{aligned} offspring1 &= L * male + (1 - L) * female \\ offspring2 &= L * female + (1 - L) * male \end{aligned} \tag{14}$$

where *male* is the male parent, *female* is the female parent and L is a random value within a specific range. Offspring's initial velocities are set to be zero.

The basic steps of the Mayfly Algorithm (MA) can be summarized in the pseudo code shown in Fig. 1.

3.3. Improvements of basic mayfly algorithm

While experimenting with the basic algorithm, we identified stability issues related to the perturbation of the existing solutions due to velocity. Premature convergence behavior of the algorithm was also noticed, because of insufficient balance between exploitation and exploration. To account for these shortcomings, a number of algorithm modifications have been developed, which are presented below.

3.3.1. Velocity limits

Testing the performance of our algorithm, it was found that the velocity quickly explodes to huge values, for example when updating the velocity of a mayfly far from the global best or personal best position. This may lead to situations where mayflies fly out of the problem space. It is worth noting that offspring's zero initial velocity can be an answer to this problem since there will be mayflies with small velocities values that can still help convergence. Taking this into account and being inspired from real mayflies that do not develop great speeds in order to always stay above water, we assume that each mayfly is able to develop a specified maximum velocity V_{max}. In such cases, the velocity is adjusted as

$$v_{ij}^{t+1} = \begin{cases} V_{max}, \ if v_{ij}^{t+1} > V_{max} \\ -V_{max}, \ if v_{ij}^{t+1} < -V_{max} \end{cases} \tag{15}$$

What is significant is that even though V_{max} controls the exploration of the search space, too small values could prevent exploitation beyond local optima. The V_{max} values could be selected as

$$V_{max} = rand * (x_{max} - x_{min}) \tag{16}$$

where the $rand \in (0, 1]$.

3.3.2. Gravity coefficient

Even though velocity limit can prevent mayflies from developing great speeds, sometimes velocities must be reduced in order to better control the balance between exploration and exploitation abilities of the mayflies. The gravity coefficient g which works similar to PSO's inertia weight (Shi & Eberhart, 1998), assists the achievement of a

Table 6
Comparison of basic MA with its improvements at 5 dimensions.

Function ID	Statistics	Basic MA	VGMA	SMA	IMA
F1	Best	**0.0000E+00**	2.3756E−29	**0.0000E+00**	**0.0000E+00**
	Worst	3.4276E−304	8.7923E−24	**0.0000E+00**	**0.0000E+00**
	Average	8.3607E−306	5.7377E−25	**0.0000E+00**	**0.0000E+00**
	Median	**0.0000E+00**	3.5990E−26	**0.0000E+00**	**0.0000E+00**
	Std.	**0.0000E+00**	1.5647E−24	**0.0000E+00**	**0.0000E+00**
F2	Best	3.8202E+00	1.4014E−17	**0.0000E+00**	**0.0000E+00**
	Worst	1.7497E+04	1.7263E+00	5.3396E−28	**1.7995E−29**
	Average	1.4784E+03	4.2841E−02	3.0863E−29	**2.0798E−30**
	Median	5.3676E+02	7.2446E−11	4.1908E−30	**0.0000E+00**
	Std.	3.0185E+03	2.0138E−01	7.2279E−29	**4.3249E−30**
F10	Best	3.1762E+00	2.2026E−13	**0.0000E+00**	**0.0000E+00**
	Worst	4.6678E+01	6.5767E+00	4.3343E−13	**0.0000E+00**
	Average	1.8583E+01	1.8654E+00	6.4659E−15	**0.0000E+00**
	Median	1.6072E+01	1.9899E+00	**0.0000E+00**	**0.0000E+00**
	Std.	1.1094E+01	1.2829E+00	4.3545E−14	**0.0000E+00**
F11	Best	2.7765E−04	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Worst	2.6659E+00	3.5527E−15	**0.0000E+00**	**0.0000E+00**
	Average	2.8847E−01	2.4869E−16	**0.0000E+00**	**0.0000E+00**
	Median	1.4419E−01	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Std.	4.5693E−01	9.1103E−16	**0.0000E+00**	**0.0000E+00**
F19	Best	1.3679E−07	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Worst	9.5302E+00	8.8508E−24	**0.0000E+00**	**0.0000E+00**
	Average	4.7634E−01	1.3526E−25	**0.0000E+00**	**0.0000E+00**
	Median	3.8534E−05	3.2908E−32	**0.0000E+00**	**0.0000E+00**
	Std.	2.0801E+00	9.1701E−25	**0.0000E+00**	**0.0000E+00**
F20	Best	4.5317E−07	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Worst	1.1108E−01	7.6207E−01	7.6207E−01	**0.0000E+00**
	Average	1.9360E−02	6.0965E−02	5.3344E−02	**0.0000E+00**
	Median	1.6887E−03	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Std.	3.2672E−02	2.0778E−01	1.9541E−01	**0.0000E+00**

Table 7
Comparison of different *gbest* selection techniques at 20 dimensions.

Function ID	Statistics	PGB-IMA	T-IMA	IMA
F1	Best	1.9874E−54	**1.4334E−55**	4.6429E−51
	Worst	1.9551E−40	**8.1709E−42**	1.3421E−37
	Average	3.9919E−41	**1.6342E−42**	2.4097E−38
	Median	**8.2332E−53**	7.1494E−50	8.1709E−42
	Std.	8.2043E−41	**3.4451E−42**	4.4713E−38
F2	Best	3.2610E−01	**1.6811E−01**	4.8141E−01
	Worst	1.5771E+01	**1.4597E+01**	1.4618E+01
	Average	1.1503E+01	**1.0334E+01**	1.0939E+01
	Median	1.2422E+01	1.2505E+01	**1.2274E+01**
	Std.	**4.2109E+00**	4.6941E+00	4.5333E+00
F10	Best	5.6843E−14	5.6843E−14	**2.8422E−14**
	Worst	1.9899E+00	1.9899E+00	**9.9496E−01**
	Average	4.9748E−01	1.0945E+00	**2.9849E−01**
	Median	6.4091E−12	9.9496E−01	**1.2790E−13**
	Std.	7.0354E−01	5.6478E−01	**4.8061E−01**
F11	Best	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Worst	3.5527E−15	7.1054E−15	**0.0000E+00**
	Average	7.1054E−16	7.1054E−16	**0.0000E+00**
	Median	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Std.	1.4980E−15	2.2469E−15	**0.0000E+00**
F19	Best	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Worst	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Average	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Median	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Std.	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
F20	Best	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Worst	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Average	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Median	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Std.	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**

sufficient balance between exploration and exploitation. As a result, the velocity of male mayfly i is now calculated by modifying the equation (7) as:

$$v_{ij}^{t+1} = g * v_{ij}^{t} + a_1 e^{-\beta r_p^2} (pbest_{ij} - x_{ij}^{t}) + a_2 e^{-\beta r_g^2} (gbest_j - x_{ij}^{t}) \tag{17}$$

Table 8
Parameter values used in PSO, FA, GA, HSA, BeA and DE.

Parameter	PSO	FA	GA	HSA	BeA	DE
Population size	60	25	100	60	45	50
$c1$and$c2$	2	–	–	–	–	–
Inertia weight	0.8–0.45	–	–	–	–	–
γ	–	1	–	–	–	–
β	–	1	–	–	–	–
α	–	0.2	–	–	–	–
Δ	–	–	–	–	–	–
Crossover rate	–	–	0.95	–	–	0.1
Mutation rate	–	–	0.1	–	–	0.9
Harmony consideration rate	–	–	–	0.9	–	–
Minimum pitch adjusting rate	–	–	–	0.4	–	–
Maximum pitch adjusting rate	–	–	–	0.9	–	–
Minimum bandwidth	–	–	–	0.0001	–	–
Maximum bandwidth	–	–	–	1	–	–
Number of selected sites	–	–	–	–	3	–
Number of elite sites	–	–	–	–	1	–
Bees around elite points	–	–	–	–	7	–
Bees around selected points	–	–	–	–	2	–
initial patch size	–	–	–	–	3	–

while the velocity of female mayfly i is now calculated by modifying the Eq. (13) as:

$$v_{ij}^{t+1} = \begin{cases} g * v_{ij}^{t} + a_2 e^{-\beta r_{mf}^2} (x_{ij}^{t} - y_{ij}^{t}), & \text{if } f(y_i) > f(x_i) \\ g * v_{ij}^{t} + fl * r, & \text{if } f(y_i) \le f(x_i) \end{cases} \tag{18}$$

The gravity coefficient g can be a fixed number in the range of $(0, 1]$, or it can be gradually reduced over the iterations, allowing the algorithm to exploit some specific areas, by being updated through the following equation:

$$g = g_{max} - \frac{g_{max} - g_{min}}{iter_{max}} \times iter \tag{19}$$

where g_{max}, g_{min} are the maximum and minimum values that the gravity

Table 9
Comparison of different methods in solving the unimodal test functions in Table 2 at 50 dimensions.

F	Statistics	PSO	GA	HSA	DE	MA
F1	Best	9.0916E−09	5.9459E−04	2.1164E+00	6.6959E−05	**6.5351E−13**
	Worst	**5.2516E−07**	1.4971E−01	4.8532E+00	2.5827E−04	2.4814E−06
	Average	1.6365E−07	1.7310E−02	3.5023E+00	1.3029E−04	**1.1777E−07**
	Median	9.8565E−08	7.7800E−03	3.4124E+00	1.3171E−04	**2.9511E−10**
	Std.	**1.3702E−07**	2.5419E−02	6.1409E−01	3.3273E−05	4.5203E−07
F2	Best	3.8778E+01	6.5781E+01	1.0285E+03	2.0927E+02	**3.0738E+01**
	Worst	**1.5453E+02**	3.3243E+02	2.5445E+03	3.7072E+02	2.1624E+02
	Average	**6.3325E+01**	1.8243E+02	1.6180E+03	2.7969E+02	6.7703E+01
	Median	4.6641E+01	1.7277E+02	1.5876E+03	2.7250E+02	**4.6230E+01**
	Std.	**2.9532E+01**	6.3965E+01	3.3439E+02	4.5395E+01	3.9877E+01
F3	Best	3.6765E−07	6.5414E−03	3.6258E+01	1.3487E−03	**1.9653E−10**
	Worst	**3.7220E−05**	1.5491E+01	8.0616E+01	3.3831E−03	1.2941E−04
	Average	**4.3627E−06**	6.1555E−01	5.8971E+01	2.3484E−03	7.3923E−06
	Median	2.5783E−06	1.5920E−01	5.7006E+01	2.2617E−03	**4.0522E−08**
	Std.	**5.8512E−06**	2.1984E+00	9.9515E+00	4.8276E−04	2.5797E−05
F4	Best	2.5468E−37	4.3467E−20	1.1288E−11	2.9778E−26	**2.4043E−56**
	Worst	2.3548E−31	4.9943E−10	6.1448E−10	1.2147E−23	**8.5486E−48**
	Average	1.5679E−32	1.8671E−11	8.2341E−11	1.8248E−24	**5.2842E−49**
	Median	1.0982E−33	5.9549E−14	4.7717E−11	8.5217E−25	**1.8406E−51**
	Std.	3.9666E−32	7.6283E−11	1.0830E−10	2.5744E−24	**1.6485E−48**
F5	Best	**−9.9999E−01**	−9.9999E−01	−9.8449E−01	**−9.9999E−01**	**−9.9999E−01**
	Worst	**−9.9999E−01**	−9.9829E−01	−9.6833E−01	**−9.9999E−01**	**−9.9999E−01**
	Average	**−9.9999E−01**	−9.9988E−01	−9.7710E−01	**−9.9999E−01**	**−9.9999E−01**
	Median	**−9.9999E−01**	−9.9995E−01	−9.7768E−01	**−9.9999E−01**	**−9.9999E−01**
	Std.	2.6540E−09	2.4545E−04	3.5064E−03	1.1330E−07	**3.4356E−11**
F6	Best	2.3986E−04	6.1763E−04	4.6960E+01	7.9372E−02	**7.0870E−06**
	Worst	**5.2348E−03**	5.3235E+00	9.2980E+01	1.2640E−01	1.9495E−01
	Average	**1.1695E−03**	3.9365E−01	6.9238E+01	1.0060E−01	6.9674E−03
	Median	9.5742E−04	1.0056E−01	6.9686E+01	9.7189E−02	**4.6543E−04**
	Std.	**8.8833E−04**	1.0023E+00	8.5282E+00	1.2279E−02	2.8710E−02
F7	Best	3.6248E+00	1.1288E+01	2.0250E+01	2.9685E+01	**2.7225E+00**
	Worst	**7.1959E+00**	3.6458E+01	2.5839E+01	4.3723E+01	7.6811E+00
	Average	5.4734E+00	1.9567E+01	2.2645E+01	3.9186E+01	**3.8769E+00**
	Median	5.4405E+00	1.8384E+01	2.2663E+01	3.9391E+01	**3.8307E+00**
	Std.	**7.6289E−01**	5.2053E+00	1.2687E+00	2.7231E+00	8.8291E−01
F8	Best	1.1314E−03	4.8795E−04	1.4999E+01	1.0761E−01	**1.9465E−05**
	Worst	1.0589E+03	1.2687E+01	4.0299E+01	**2.0778E−01**	3.9609E−01
	Average	2.5721E+02	6.3639E−01	2.6360E+01	1.5741E−01	**1.8945E−02**
	Median	1.5880E−02	8.6173E−02	2.6261E+01	1.5354E−01	**8.6829E−04**
	Std.	3.9035E+02	1.9416E+00	4.3973E+00	**2.2276E−02**	6.0779E−02
F9	Best	3.1275E+00	2.5367E+01	1.6626E+02	5.4389E+02	**3.3591E−02**
	Worst	5.0851E+02	2.6116E+02	2.8451E+02	7.5811E+02	**4.5475E−01**
	Average	9.2912E+01	8.8802E+01	2.3063E+02	6.7040E+02	**1.7130E−01**
	Median	5.7622E+01	7.4723E+01	2.2840E+02	6.7974E+02	**1.6596E−01**
	Std.	8.6769E+01	4.4836E+01	2.8412E+01	4.8105E+01	**8.3733E−02**

coefficient can take, *iter* is the current iteration of the algorithm and $iter_{max}$ is the maximum number of iterations.

3.3.3. Reduction of nuptial dance and random walk

The nuptial dance performed by male mayflies, as well as the females' random walk, appear to be two very powerful local search methods capable of helping the algorithm escape local optima. However, performing a random walk could lead a mayfly to a far worse search area. The problem comes from the fact that nuptial dance d or random walk fl often take large initial values. This can be alleviated by gradually reducing both nuptial dance d and random walk fl over the iterations. Consequently, both values could be updated by using a geometric progression formula as

$$d_t = d_0\delta^t,\ 0 < \delta < 1 \tag{20}$$

$$fl_t = fl_0\delta^t,\ 0 < \delta < 1 \tag{21}$$

where t is the iteration counter and δ a fixed value in the range of (0.1).

3.3.4. Mutate the genes of offspring

To deal with situations of premature convergence that may lead to a local instead of a global minimum, a variation of the original algorithm adds a random mutation to a portion of the population, in order for the algorithm to explore new areas of the search space that may not be visited otherwise. Particularly, we add a normally distributed random number to the chosen offspring's variable for mutation. In this way the offspring is altered as

$$offspring_n^{'} = offspring_n + \sigma N_n(0,\,1) \tag{22}$$

where, σ is the standard deviation of the normal distribution and $N_n(0,\,1)$ is a standard normal distribution with mean = 0 and variance = 1.

3.4. Multi-objective mayfly algorithm (MMA)

We also modify the MA in order to be applied to multiobjective optimization problems. Original MA saves only a single best solution, instead of multiple solutions that multi-objective problems require. In addition, MA updates the *gbest* with the best solution obtained so far in each iteration, whereas there is no single best solution in multi-objective optimization problems. Hence, we equip MA with a repository of solutions, which maintains the best non-dominated solutions obtained so far during optimization, like Coello Coello, Pulido, and Lechuga (2004) used in their research. Moreover, in contrast with MA, female mayflies in MMA also update their personal best position.

Table 10
Comparison of different methods in solving the multimodal test functions in Table 3 at 50 dimensions.

F	Statistics	PSO	FA	GA	DE	MA
F10	Best	3.5818E+01	1.3428E+02	1.4942E+01	5.1514E+01	**5.9697E+00**
	Worst	1.5222E+02	2.4801E+02	7.1178E+01	8.0761E+01	**2.1890E+01**
	Average	8.1826E+01	1.7824E+02	2.8358E+01	6.5685E+01	**1.1903E+01**
	Median	8.0094E+01	1.7626E+02	2.6080E+01	6.6143E+01	**1.0944E+01**
	Std.	1.9773E+01	2.4561E+01	1.0221E+01	6.0005E+00	3.8202E+00
F11	Best	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Worst	**0.0000E+00**	3.5527E−15	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Average	**0.0000E+00**	5.6843E−16	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Median	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Std.	**0.0000E+00**	1.3156E−15	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
F12	Best	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Worst	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	8.8782E−02
	Average	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	4.1431E−03
	Median	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Std.	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	1.2929E−02
F13	Best	2.4798E−07	5.1657E+01	2.4882E−02	**0.0000E+00**	9.1193E−11
	Worst	1.7760E+01	7.7938E+01	1.3206E+00	**0.0000E+00**	9.2080E−05
	Average	2.4865E+00	6.2260E+01	2.5143E−01	**0.0000E+00**	6.5634E−06
	Median	2.0364E−05	6.3138E+01	1.8636E−01	**0.0000E+00**	8.2389E−08
	Std.	5.0074E+00	5.8752E+00	2.2950E−01	**0.0000E+00**	1.8657E−05
F14	Best	**4.9987E−01**	2.9998E+00	6.9987E−01	2.2013E+00	1.1998E+00
	Worst	**1.0998E+00**	7.3998E+00	1.7998E+00	2.8015E+00	3.9998E+00
	Average	**7.9673E−01**	5.0778E+00	1.1038E+00	2.5420E+00	2.3150E+00
	Median	**7.9987E−01**	5.1998E+00	9.9987E−01	2.5156E+00	2.2998E+00
	Std.	**1.0220E−01**	7.4403E−01	2.5230E−01	1.2307E−01	6.3363E−01
F15	Best	3.7490E−02	7.4752E+07	7.3404E+02	1.9783E+03	**1.1734E−07**
	Worst	2.0620E+01	5.1837E+08	1.7980E+06	4.3794E+03	**7.4161E−01**
	Average	1.9738E+00	2.2682E+08	1.8140E+05	3.3955E+03	**2.6477E−02**
	Median	8.1308E−01	2.2793E+08	6.7025E+04	3.3635E+03	**3.3301E−05**
	Std.	3.4311E+00	1.1216E+08	3.3235E+05	5.0688E+02	**1.1027E−01**
F16	Best	−1.7720E+03	2.5791E+07	3.8881E+02	−1.4983E+03	**−1.7886E+03**
	Worst	−1.6186E+03	4.6693E+08	3.9768E+06	−1.3518E+03	**−1.6331E+03**
	Average	−1.7079E+03	1.0719E+08	3.3469E+05	−1.4132E+03	**−1.7134E+03**
	Median	−1.7082E+03	9.4318E+07	5.4144E+04	−1.4098E+03	**−1.7109E+03**
	Std.	**3.2677E+01**	6.8873E+07	7.5358E+05	3.4590E+01	3.9927E+01
F17	Best	**8.1806E−13**	7.1851E−03	1.4874E−08	7.3147E+03	8.9188E−05
	Worst	**2.1978E−02**	1.4125E+01	1.4396E+00	2.4851E+06	6.3505E+03
	Average	**1.1994E−03**	1.2537E+00	3.3268E−02	3.6867E+05	5.5851E+02
	Median	**4.3672E−10**	4.4380E−01	5.9413E−05	2.0286E+05	4.9078E+01
	Std.	**4.3641E−03**	2.2301E+00	2.0377E−01	4.8412E+05	1.4070E+03
F18	Best	1.6437E−02	5.1565E−02	**6.4896E−03**	1.0274E−01	1.4510E−02
	Worst	3.9078E−02	2.9959E−01	5.3306E−02	2.0661E−01	**3.8504E−02**
	Average	2.5368E−02	1.4019E−01	**2.3188E−02**	1.4985E−01	2.9476E−02
	Median	2.5155E−02	1.2152E−01	**2.1608E−02**	1.5148E−01	3.0571E−02
	Std.	**5.5360E−03**	5.9435E−02	1.1134E−02	2.5086E−02	6.1212E−03

3.4.1. Movement of male mayflies in multi-objective optimization

The movement of male mayflies in multi-objective optimization works similarly to their movement when performing on single-objective optimization problems. Since there is no single best solution in multi-objective problems, the selection of *gbest* is performed by picking a random solution from the repository of non-dominated solutions.

If the male mayfly is dominated by the *gbest*, equation (17) is used. Otherwise, equation (20) is used.

3.4.2. Movement of female mayflies in multi-objective optimization

Similarly, the following equation is used for the female mayflies:

$$v_{ij}^{t+1} = \begin{cases} g * v_{ij}^t + a_2 e^{-\beta r_{mf}^2}(x_{ij}^t - y_{ij}^t), & \text{if male dominates female} \\ g * v_{ij}^t + fl * r, & \text{otherwise} \end{cases} \quad (23)$$

3.4.3. Mating of mayflies in multi-objective optimization

Equation (14), is also used for the mating process. Particularly, a male and a female are selected according to their rank. To advance the convergence behavior of MMA, the crossover operator is performed using the personal best position of each mayfly.

3.4.4. Crowding distance

The repository has a maximum size to store the non-dominated solutions. To rank the mayflies and keep the best ones, a fast non-dominated sorting using Crowding Distance (CD), just like NSGA-II (Deb, Pratap, Agarwal, & Meyarivan, 2002) does, is performed. CD provides an estimate of the largest cuboid enclosing a solution without including any other solutions, by calculating the Euclidian distance between neighbor individuals (Deb et al., 2002). The boundary solutions which have the lowest and highest objective function values are always selected by being given an infinite CD value. By summing up the CD values of each objective function, the final CD value of a solution is computed.

Previous research has already proven high performance results of multiobjective swarm intelligence optimization methods combined with CD (Feng, Zheng, & Li, 2010; Zhang & Li, 2014).

The basic steps of the Multi-objective Mayfly Algorithm (MMA) can be summarized in the pseudo code shown in Fig. 2.

4. Validation and comparison

Even though the Mayfly Algorithm can be implemented in any programming language, we chose Matlab because of the ease of creating visualizations and using data containers, known as structure

Table 11
Comparison of different methods in solving the fixed dimension test functions in Table 4.

F	Statistics	PSO	FA	GA	DE	MA
F19	Best	8.3877E−154	1.3524E−67	6.6435E−284	**0.0000E+00**	**0.0000E+00**
	Worst	1.8233E−142	4.0166E−64	2.2554E−270	**0.0000E+00**	**0.0000E+00**
	Average	7.8758E−144	8.5086E−65	2.3272E−272	**0.0000E+00**	**0.0000E+00**
	Median	6.7723E−147	6.5859E−65	2.8104E−278	**0.0000E+00**	**0.0000E+00**
	Std.	3.1857E−143	8.5534E−65	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
F20	Best	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Worst	7.6207E−01	5.0930E−02	5.0173E−05	**0.0000E+00**	**0.0000E+00**
	Average	3.0482E−02	1.1275E−03	7.8445E−07	**0.0000E+00**	**0.0000E+00**
	Median	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Std.	1.5085E−01	7.2154E−03	5.5144E−06	**0.0000E+00**	**0.0000E+00**
F21	Best	**0.0000E+00**	**0.0000E+00**	4.3221E−15	**0.0000E+00**	**0.0000E+00**
	Worst	5.6571E−28	8.0306E−01	2.3985E−01	5.6283E−01	**0.0000E+00**
	Average	2.2863E−29	1.3302E−01	8.1565E−03	3.3505E−02	**0.0000E+00**
	Median	**0.0000E+00**	8.4495E−02	5.3619E−06	**0.0000E+00**	**0.0000E+00**
	Std.	8.6341E−29	1.6196E−01	3.5810E−02	1.1740E−01	**0.0000E+00**
F22	Best	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Worst	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Average	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Median	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Std.	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
F23	Best	**−1.0000E+00**	**−1.0000E+00**	**−1.0000E+00**	**−1.0000E+00**	**−1.0000E+00**
	Worst	**−1.0000E+00**	0.0000E+00	**−1.0000E+00**	**−1.0000E+00**	−8.1102E−05
	Average	**−1.0000E+00**	−7.5260E−01	**−1.0000E+00**	**−1.0000E+00**	−9.8000E−01
	Median	**−1.0000E+00**	**−1.0000E+00**	**−1.0000E+00**	**−1.0000E+00**	**−1.0000E+00**
	Std.	**0.0000E+00**	4.3038E−01	**0.0000E+00**	**0.0000E+00**	1.4141E−01
F24	Best	2.7397E−152	3.5417E−68	**0.0000E+00**	3.7651E−255	**0.0000E+00**
	Worst	6.8556E−143	1.1056E−65	**0.0000E+00**	7.0690E−245	**0.0000E+00**
	Average	2.1814E−144	3.1761E−66	**0.0000E+00**	1.6103E−246	**0.0000E+00**
	Median	8.9801E−147	2.0352E−66	**0.0000E+00**	2.9832E−250	**0.0000E+00**
	Std.	9.9328E−144	2.9500E−66	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
F25	Best	1.3422E−07	2.9685E−07	2.9035E−07	8.3232E−05	**0.0000E+00**
	Worst	1.3481E−03	6.1719E+00	7.9641E−01	3.7344E−02	**6.3675E−30**
	Average	1.7379E−04	3.9882E−01	2.1691E−01	5.9501E−03	**1.6396E−31**
	Median	1.1726E−04	5.5848E−02	1.5227E−01	2.9506E−03	**0.0000E+00**
	Std.	2.4621E−04	1.0409E+00	2.1695E−01	7.6570E−03	**8.0074E−31**

arrays. All the simulations have been carried out on an i5, 3.3 GHz desktop computer with 8 GB of RAM. Furthermore, we decided to evaluate each version of the algorithm by comparing its performance for a fixed number of function evaluations. In our simulations, we run each algorithm with 95,000 function evaluations, performing 50 replications for each one. Each run of MA takes less than 8 s, while each run of MMA takes about 25 s.

4.1. Benchmark functions

The selection of test functions is very important when it comes to validation and comparison of the performance of optimization algorithms (Jamil & Yang, 2013). An algorithm's efficiency and accuracy dependent directly on the dimensionality of the search space, or the ability of the particular algorithm to keep up the direction changes in the function (Jamil & Yang, 2013). Test functions can be categorized according to their modality or separability.

Multimodal are functions with more than one local optimum, the use of which helps evaluating the ability of an algorithm to escape from local optima as well as its exploration capability. In contrast, a function $f(\mathbf{x})$ is called unimodal if it is monotonically decreasing for $x \le x^*$ or increasing for $x \ge x^*$. In such cases, $f(\mathbf{x}^*)$ is the global optimum value and there are no other local optima. Moreover, another classification of test functions is into separable and non-separable ones.

In this section, the proposed method along with its variations is evaluated with the use of six different benchmark mathematical functions. Furthermore, the performance of the best variant is compared to those of some other meta-heuristic methods on 38 different benchmark mathematical functions as well as on multiobjective optimization and on a discrete flowshop problem. The test functions include three different groups of functions: unimodal functions, multimodal functions, and fixed-dimension functions. These three kinds of test functions are listed in Tables 2–4. The 2-D versions of the test functions, except the 4-D Colville function, are depicted in Figs. 3–5.

4.2. Comparison of basic MA with its improvements

In this section we compare the basic MA with its improved variations. Initially, we fine tune the algorithms by using different population sizes from $n = 15$ to $n = 150$. We found that for the most problems a value of $n = 30$ to 80 is sufficient. Therefore, we used a fixed population $n = 40$ (20 males and 20 females) for all simulations of MA.

For all the four versions of MA that we have compared, we used attraction constants $a_1 = 1$, $a_2 = 1.5$, visibility coefficient $\beta = 2$, nuptial dance $d = 0.1$, random flight $fl = 0.1$, single point uniform crossover with the rate of 0.95 and linear selection mechanism as described in Section 3.2.3.

These values came up from further fine tuning the basic MA as shown in Table 5. Each initialization scheme was tested for 10 runs.

Another version of MA is the one with velocity limit and gravity coefficient (VGMA), were V_{max} is generated through equation (16) with $rand$=0.1 and $g = 0.8$.

Moreover, we consider as the stochastic MA (SMA) the one with gradual reduction of nuptial dance, as well as random flight with $\delta = 0.77$, and gaussian mutation with a rate of 0.1.

Finally, as the Improved MA (IMA) we consider the Mayfly Algorithm that incorporates all of its improvements. A more detailed description of the above values is described in the Appendix A. Table 6 shows the comparison among four algorithms on six benchmark functions.

As derives from the results in Table 6, the performance of IMA is superior to the others. What is significant is that SMA found minimum

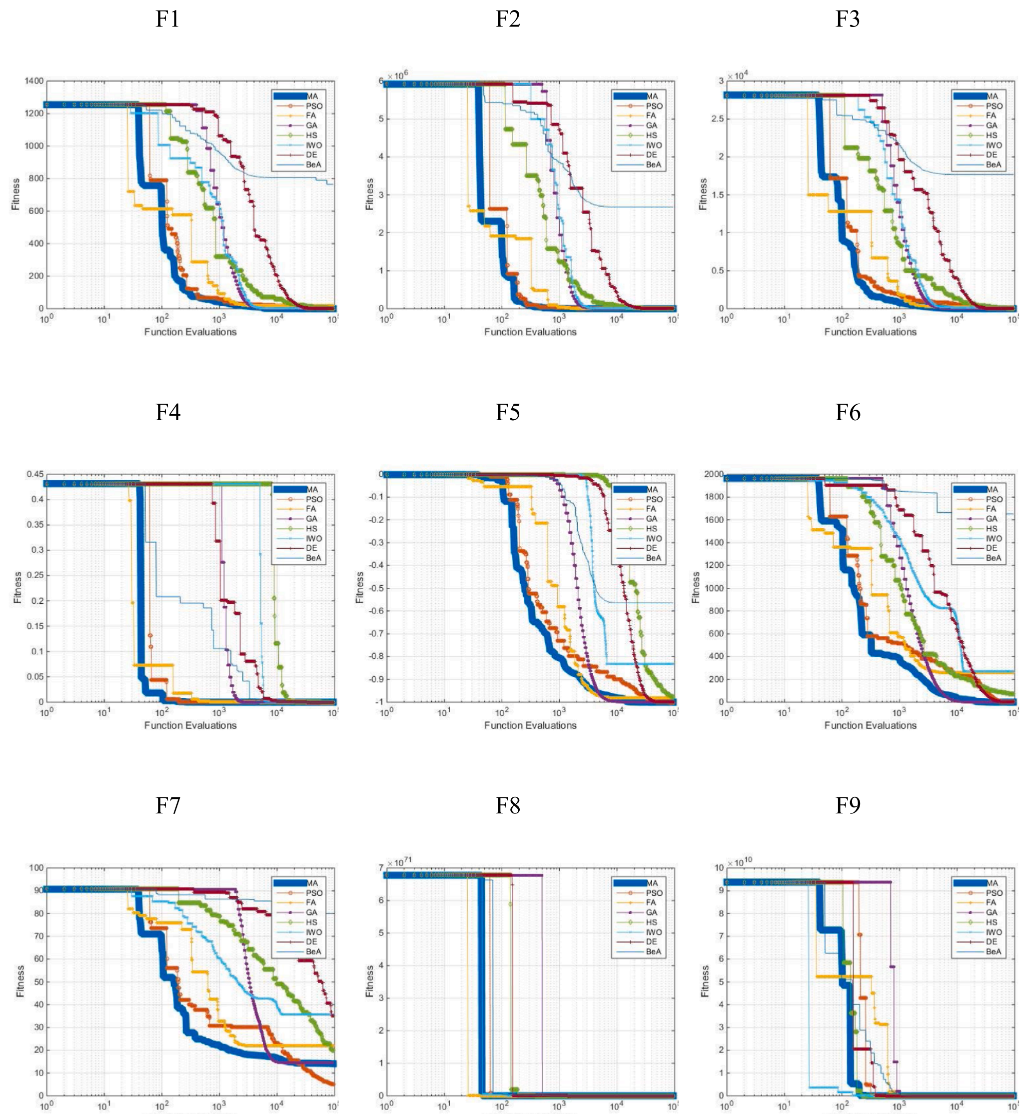

Fig. 6. Convergence characteristic curves for PSO, FA, GA, HAS, IWO, BeA, DE and MA in solving the unimodal test functions in Table 2 at 50 dimensions.

values in all functions, without using velocity limits, which shows the usefulness of the nuptial dance and the reduction of its values.

Moreover, as mentioned in Section 3.2.1, the *gbest* is selected from the male population, since only males congregate in swarms. In an attempt to investigate the impact of the *gbest* selection, two variants of the IMA were developed. The first one shall be called PGB-IMA and concerns the case in which the *gbest* is selected from the whole mayfly population (both males and females). The second one (T-IMA), concerns the case in which after a female mayfly locates a position better than the *gbest* found so far, it turns into a male. To keep the proportion of females and males in the population, the worst male in the swarm turns into a female.

To compare PGB-IMA, T-IMA and IMA, the six benchmark functions used in Table 6 were used. Because of the fact that when performing at five dimensions the results of the three algorithms were very similar, we increased the dimensions from five to 20, when performing on the non-fixed-dimension test functions. Table 7 shows the comparison among the three algorithms.

As derives from the results in Table 7, the three methods are very competitive with each other. Particularly, PGB-IMA and T-IMA were found to converge faster than IMA when performing on unimodal test functions (F1 and F2), due to the guidance of the best solution ever found. However, they were found to converge slower or even getting stuck at local optima points when performing on multimodal functions (F10 and F11) compared to the performance of IMA. According to literature, the superiority of IMA when performing on multimodal test functions could be due to its non-acceptance properties, of better solutions for *gbest*. Not accepting better solutions is proved to be a fundamental property of metaheuristics because it allows a more extensive search for the global optimal solution (Glover, 1986; Kirkpatrick, Gelatt, & Vecchi, 1983). However, as it appears, this could negatively affect the convergence behavior of the algorithm when performing on multimodal test functions.

Since to the rest of the paper we not only use unimodal test functions but also complex multimodal ones as well as a real world problem, we continue the evaluation of the newly proposed method using the IMA version (to the rest of the paper the MA abbreviation refers to the IMA version).

4.3. Comparison with other algorithms

In order to test the performance of the new mayfly algorithm (MA), we compare it with other popular metaheuristic algorithms, specifically particle swarm optimization (PSO) (Kennedy & Eberhart, 1995), genetic algorithm (GA) (Goldberg, 1989), firefly algorithm (FA) (Yang, 2008), differential evolution (DE) (Storn & Price, 1997), harmony search

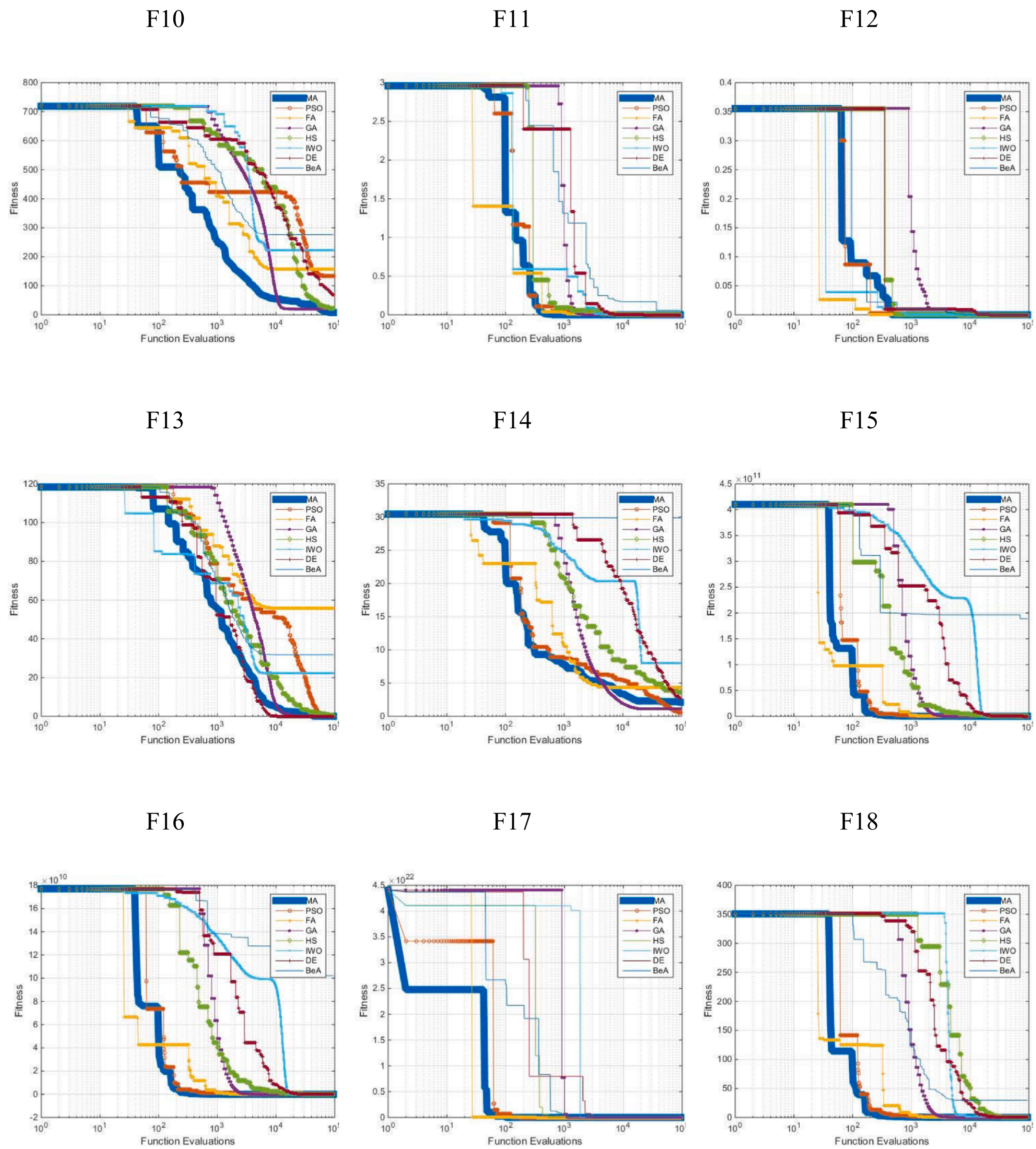

Fig. 7. Convergence characteristic curves for PSO, FA, GA, HAS, IWO, BeA, DE and MA in solving the multimodal test functions in Table 3 at 50 dimensions.

(HSA) (Zong Woo Geem et al., 2001), invasive weed optimization (IWO) (Mehrabian & Lucas, 2006) and bees algorithm (BA) (Pham et al., 2006), which were briefly described in Section 2. Table 8 shows the parameter values for each one of these algorithms according to the literature (Kennedy & Eberhart, 1995; Mehrabian & Lucas, 2006; Pham et al., 2006; Storn & Price, 1997; Yang, 2008; Zong Woo Geem et al., 2001).

We now compare the MA with the state of the art optimization methods mentioned above. Table 9–11 present the five most successful algorithms according to the function groups, while Figs. 6–8 present the convergence characteristic curves of all the optimization methods. A more detailed comparison of all methods is presented in the Appendix B.

The comparison results indicate the superiority of Mayfly over the other algorithms. Table 9 and Table 11 illustrate that not only MA was the only algorithm that found the best values in all 11 unimodal functions, but also it has better mean values in most of them, no matter the dimension. Table 9 and Table 10 show that at 50 dimensions, MA found best values in 10 multimodal functions, leaving all the other algorithms behind. Even though it was harder for all algorithms to locate a global optimum, MA located best values on 14 different functions, while PSO did it on five (see Fig. 9).

MA also achieved better results irrespective of the size of the dimension. Furthermore, on 2-D, MA is the only algorithm that located the best value in six different functions, while on 4-D (Colville's function), MA is the only one that located the global minimum.

From the preceding analysis, it is derived that the proposed MA is superior to the other algorithms in terms of accuracy and efficiency, since it can detect a better value in most of the test functions, while running with the same configuration on all problems. In cases where most of the algorithms get trapped into local optima, MA is capable of escaping thanks to males' nuptial dance, females' random flight and mutation of their genes. Another important characteristic is that mayflies' low speeds as well as offspring's zero speed can assist convergence, while providing the algorithm with the advantage to have solutions with both high and low velocities at the same time.

4.4. Comparison of MA with PSO, GA and DE on CEC 2017 test functions

In this subsection, the proposed MA algorithm is compared to the three most successful optimization methods of Section 4.3 on CEC 2017 Competition on Constrained Real-Parameter Optimization (Fan et al., 2018) test functions. In particular, MA is compared to PSO, GA and DE, as presented in Table 12 and Fig. 10.

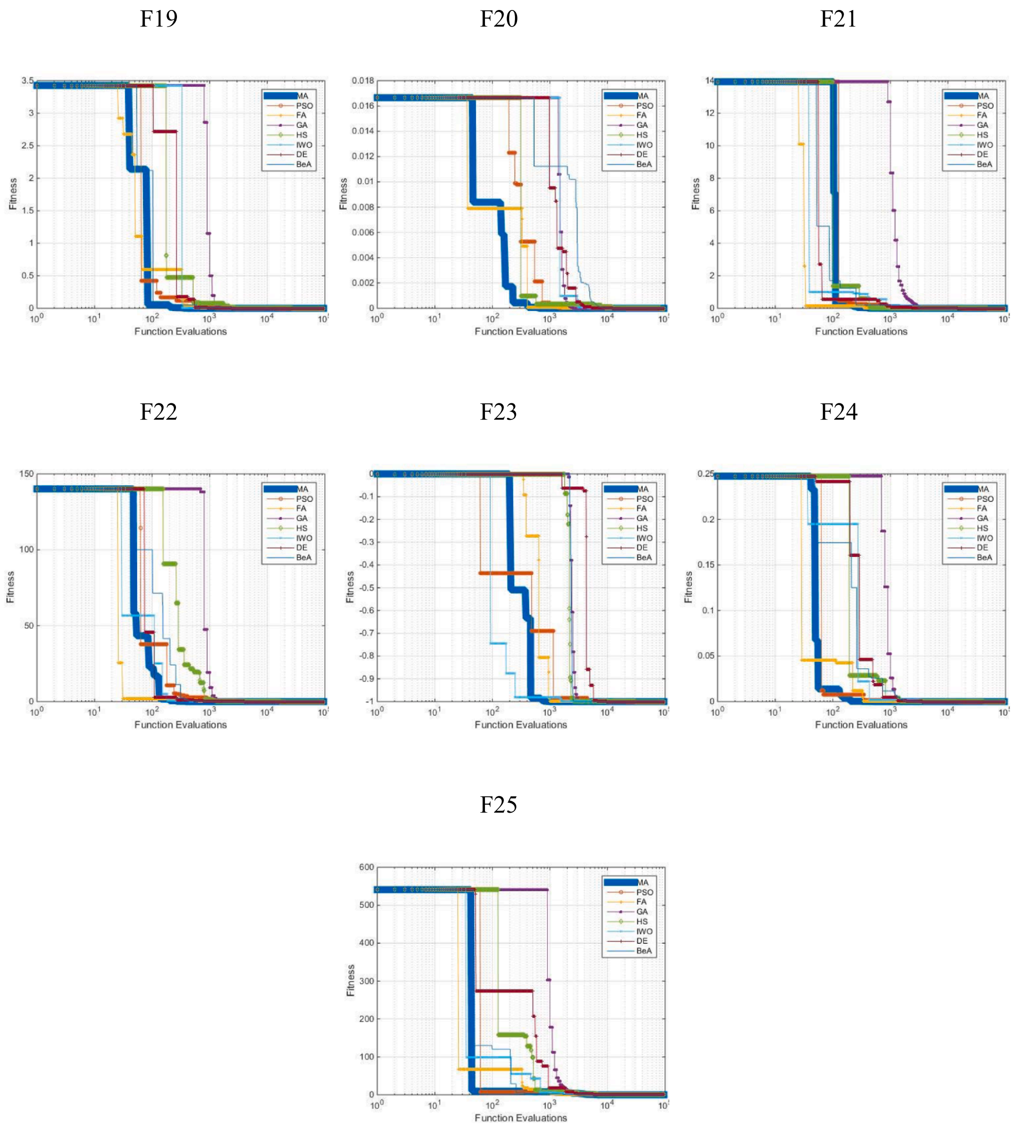

Fig. 8. Convergence characteristic curves for PSO, FA, GA, HAS, IWO, BeA, DE and MA in solving the fixed-dimension test functions in Table 4.

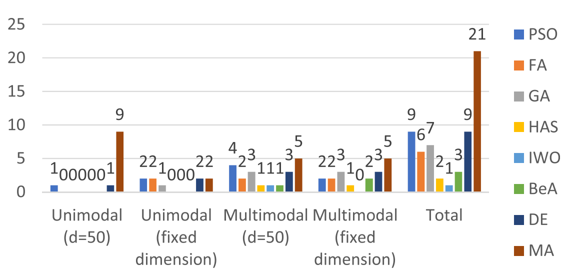

Fig. 9. The number of functions in which each algorithm performs best according to the function groups.

4.5. Results of MMA

This subsection shows a comparison of the MMA’s convergence behavior to that of the famous NSGA-II. The first three of a set of challenging test functions called ZDT proposed by Zitzler et al. (2000) are used for our comparison. Fig. 11 indicates that between MMA and NSGA-II, the convergence of MMA is better, and very competitive. Note that the maximum repository size for both MMA and NSGA-II is set to 50. Comparing to NSGA-II, MMA was able to explore the search space more efficiently, by using only 25,000 function evaluations. Especially on ZDT3, which has a Pareto optimal front with separated regions, MMA outperforms NSGA-II. These types of fronts are very common in real world problems and very challenging to be determined, because an algorithm might get trapped in one of the regions and fail to approximate all the separated regions, like NSGA-II did.

To further compare the performance of the proposed MMA to that of NSGA-II, the two set coverage metric (C) as well as the M_3^* metric proposed by Zitzler et al. (2000) were used.

Regarding the C metric when performing on ZDT1, values of $C(MMA, NSGA-II) = 0.66$, and of $C(NSGA-II, MMA) = 0$ were found, which mean that most of the solutions obtained using NSGA-II are dominated by or equal to solutions obtained using MMA and that none of the solutions obtained from MMA are covered by the solutions obtained from NSGA-II, respectively. Moreover, when performing on ZDT2, values of $C(MMA, NSGA-II) = 0.62$, and of $C(NSGA-II, MMA) = 0$ were found. Finally, when performing on ZDT3, values of $C(MMA, NSGA-II) = 0.52$, and of $C(NSGA-II, MMA) = 0.02$ were found.

Regarding the M_3^* metric, which considers the extent of the set of objective vectors that correspond to each optimization method, values of $M_3^*(MMA) = 1.38$ and $M_3^*(NSGA-II) = 1.14$ were found when performing on ZDT1. Moreover, when performing on ZDT2, values of $M_3^*(MMA) = 1.32$ and $M_3^*(NSGA-II) = 1.12$ were found. Finally,

Table 12
Comparison of different methods in solving the CEC2017 test functions at 50 dimensions.

Function ID	Statistics	PSO	GA	DE	MA
C01	Best	5.3510E+02	2.5368E+03	6.9573E+04	**1.3564E+01**
	Worst	2.0959E+03	1.2096E+04	9.1803E+04	**4.1288E+02**
	Average	1.2353E+03	4.9716E+03	8.1575E+04	**1.1953E+02**
	Median	1.2130E+03	3.9441E+03	8.1400E+04	**1.1972E+02**
	Std.	3.3934E+02	2.4928E+03	6.6111E+03	**7.9142E+01**
C02	Best	1.0916E+03	1.4037E+03	9.5138E+04	**1.2680E+02**
	Worst	2.2433E+03	9.6292E+03	1.1683E+05	**8.7356E+02**
	Average	1.7401E+03	3.6594E+03	1.0387E+05	**4.1310E+02**
	Median	1.7115E+03	3.1638E+03	1.0422E+05	**4.1738E+02**
	Std.	3.4918E+02	1.9934E+03	6.3316E+03	**1.7966E+02**
C04	Best	8.9449E+01	6.1749E+01	1.2115E+02	**4.3780E+01**
	Worst	2.7659E+02	2.4392E+02	**1.7085E+02**	1.9700E+02
	Average	1.2870E+02	1.3568E+02	1.4773E+02	**8.6502E+01**
	Median	1.2139E+02	1.3399E+02	1.4741E+02	**8.3577E+01**
	Std.	3.8387E+01	4.2609E+01	**1.2966E+01**	3.1208E+01
C05	Best	4.4943E+01	3.2324E+04	2.8626E+09	**1.0681E+01**
	Worst	4.8663E+03	5.3996E+06	7.6445E+09	**3.0010E+03**
	Average	5.6413E+02	7.0891E+05	5.4894E+09	**3.9312E+02**
	Median	**1.6431E+02**	3.5026E+05	5.5429E+09	1.9763E+02
	Std.	1.2624E+03	1.1066E+06	1.3493E+09	**6.2276E+02**
C06	Best	3.3595E+04	1.0375E+05	1.4318E+05	**2.8844E+04**
	Worst	**8.1536E+04**	3.9357E+06	2.7168E+06	1.7322E+05
	Average	**5.3732E+04**	5.1412E+05	2.0377E+05	8.3088E+04
	Median	**5.5835E+04**	2.0721E+05	2.0137E+05	7.6958E+04
	Std.	**1.3758E+04**	8.5776E+05	2.5991E+04	3.2237E+04
C07	Best	−3.0185E+03	−2.7711E+03	−4.3794E+03	**−4.7280E+03**
	Worst	−2.0112E+03	−1.0836E+03	−4.1713E+03	**−4.1953E+03**
	Average	−2.4129E+03	−2.0770E+03	−4.2591E+03	**−4.4513E+03**
	Median	−2.4476E+03	−2.2589E+03	−4.2584E+03	**−4.4442E+03**
	Std.	2.1577E+02	5.8298E+02	**4.8843E+01**	1.4266E+02
C12	Best	9.1817E+01	5.7969E+01	1.0156E+02	**4.1788E+01**
	Worst	1.6534E+02	2.0307E+02	1.3805E+02	**1.3531E+02**
	Average	1.2336E+02	1.0955E+02	1.2314E+02	**7.4788E+01**
	Median	1.1844E+02	9.9529E+01	1.2420E+02	**6.8652E+01**
	Std.	2.3428E+01	3.5072E+01	**9.0717E+00**	2.4277E+01
C13	Best	1.0069E+02	4.8828E+02	7.1116E+02	**1.8866E+00**
	Worst	8.3801E+02	3.5644E+05	1.3312E+03	**6.7625E+02**
	Average	2.3953E+02	3.1949E+04	9.9939E+02	**2.2171E+02**
	Median	1.6222E+02	8.5454E+03	9.6916E+02	**1.5560E+02**
	Std.	2.0406E+02	7.4751E+04	**1.4049E+02**	1.8569E+02
C20	Best	2.5846E+00	1.5134E+01	4.8171E+00	**2.3900E+00**
	Worst	1.3939E+01	1.7307E+01	5.1830E+00	**3.3521E+00**
	Average	4.8047E+00	1.6301E+01	5.0130E+00	**2.8111E+00**
	Median	3.6909E+00	1.6412E+01	5.0283E+00	**2.8075E+00**
	Std.	2.9486E+00	5.7934E−01	**8.6177E−02**	2.2765E−01
C21	Best	**3.4788E−10**	4.0808E−06	1.5155E−07	8.2280E−08
	Worst	**7.4089E−04**	4.4849E−03	7.8524E−03	4.0019E−02
	Average	**7.7049E−05**	8.7135E−04	1.1329E−03	3.4500E−03
	Median	**1.4508E−05**	2.1869E−04	6.7280E−04	4.6607E−04
	Std.	**1.6207E−04**	1.3555E−03	1.6743E−03	8.1144E−03
C23	Best	**2.2558E+00**	2.2559E+00	2.2559E+00	**2.2558E+00**
	Worst	**2.2569E+00**	2.2586E+00	2.2605E+00	2.2617E+00
	Average	**2.2561E+00**	2.2565E+00	2.2573E+00	2.2570E+00
	Median	**2.2560E+00**	2.2562E+00	2.2567E+00	2.2564E+00
	Std.	**2.8045E−04**	6.8810E−04	1.4112E−03	1.3420E−03
C24	Best	5.4617E−05	5.6697E−05	**5.9250E−06**	1.4169E−04
	Worst	**1.8147E−03**	3.4748E−03	8.8408E−03	6.1745E−03
	Average	**6.1379E−04**	1.1497E−03	2.2585E−03	2.3956E−03
	Median	**3.8763E−04**	1.1058E−03	1.4587E−03	2.0604E−03
	Std.	**5.7289E−04**	1.0392E−03	2.2385E−03	1.7248E−03
C25	Best	2.8331E−05	4.1397E−05	3.7093E−06	**3.7033E−06**
	Worst	**2.3619E−03**	5.2847E−03	5.9852E−03	1.5711E−02
	Average	**6.8275E−04**	1.5155E−03	1.6657E−03	2.7498E−03
	Median	**2.9799E−04**	1.3022E−03	9.4958E−04	1.7142E−03
	Std.	**7.0987E−04**	1.3208E−03	1.6406E−03	3.6932E−03

when performing on ZDT3, values of $M_3^*(MMA) = 1.62$ and $M_3^*(NSGA-II) = 1.34$ were found. As a result, the fronts obtained from MMA spread out to wider ranges compared to those obtained from NSGA-II.

4.6. Real-world problem

To test MA's accuracy and efficiency in discrete, as well as real engineering problems, we compared its performance to that of the other algorithms using a classic flow-shop scheduling problem. Flow shop scheduling problems, are a class of scheduling problems with a

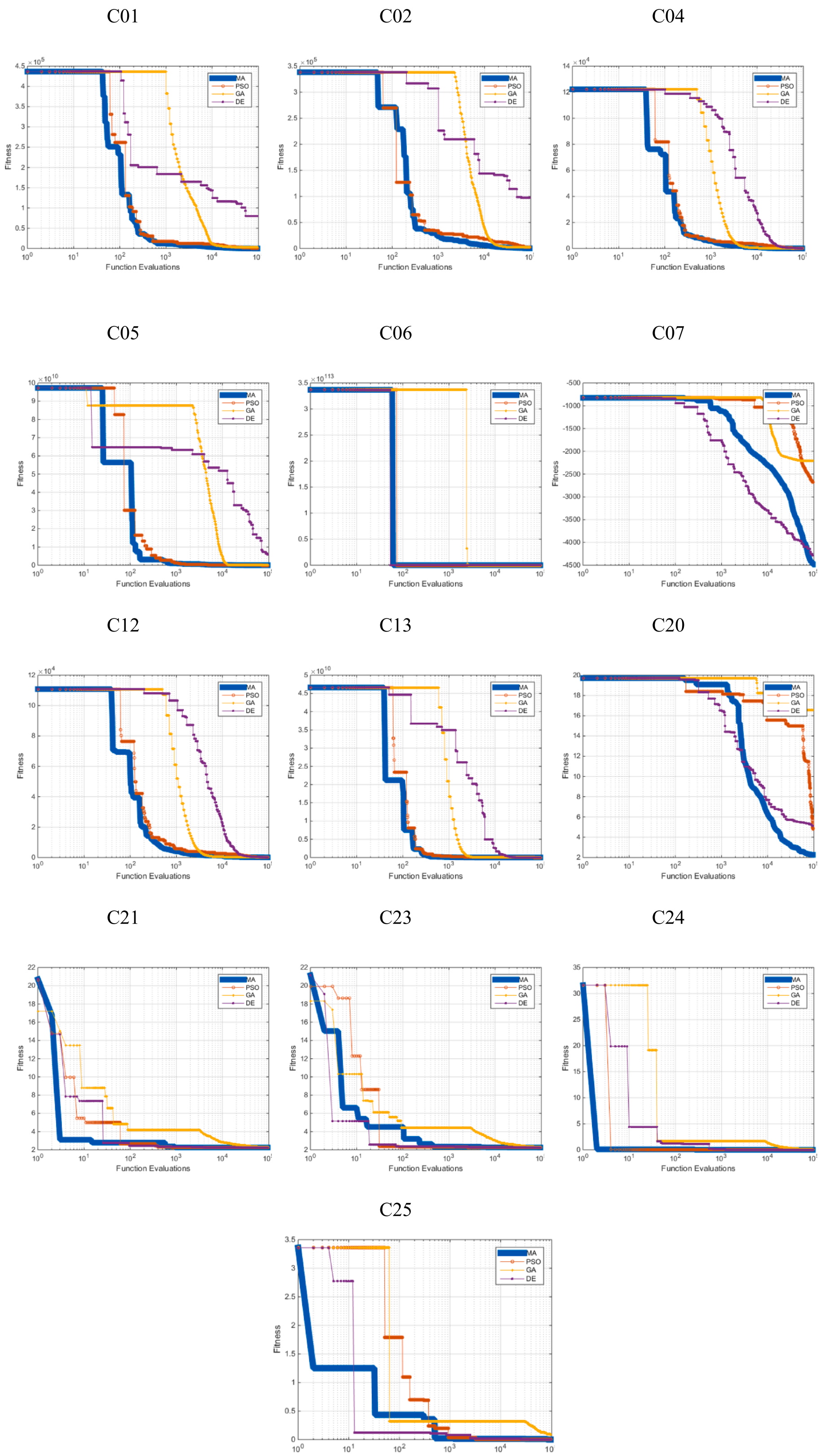

Fig. 10. Convergence characteristic curves for PSO, GA, DE and MA in solving the CEC2017 test functions.

workshop, proposed by Johnson (1954). In a flowshop scheduling problem, there is a set of n jobs or tasks to be processed in a set of m machines or processors in the same order. At any time, each job can be processed on at most one machine and each machine can process at most one job. Furthermore, once a job is processed on a machine, it cannot be terminated before it is completed (Marinakis & Marinaki, 2013). The objective is to find a sequence for the processing of the jobs in the machines so that a given criterion is optimized, with the most

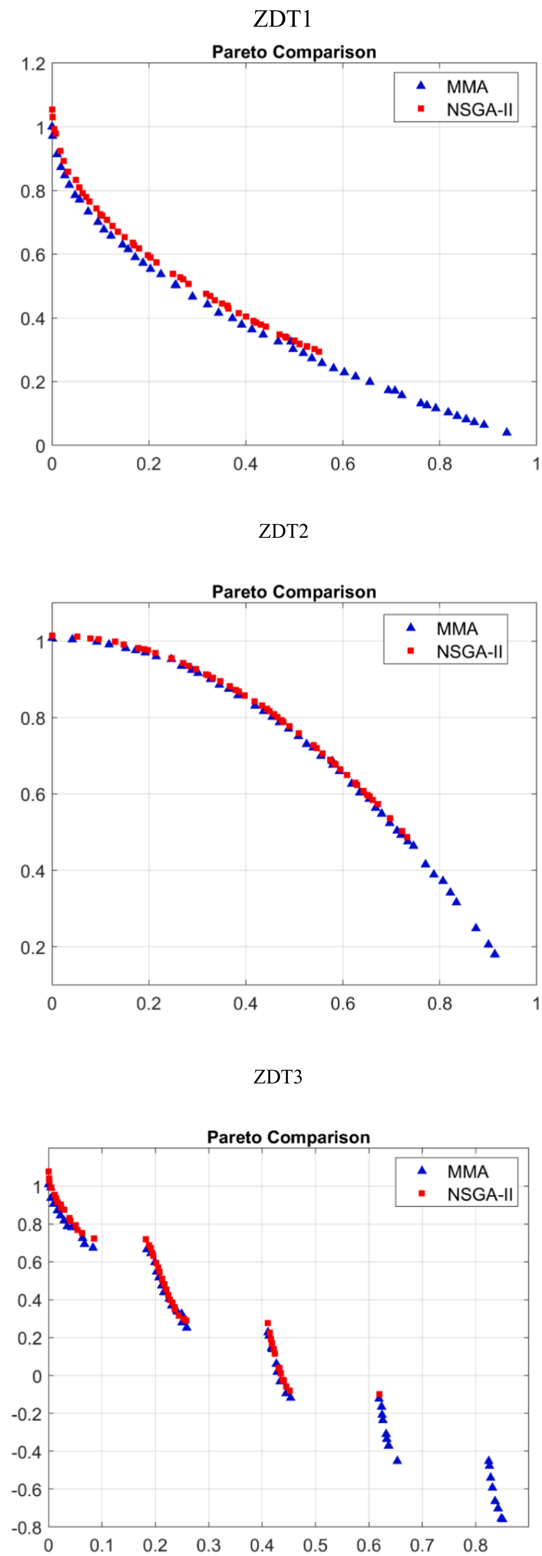

Fig. 11. Pareto front comparison of MMA and NSGA-II.

common criterion being the minimization of the makespan (C_{max}) (Morton & Pentico, 1993). Given the job permutation $\pi = \{\pi_1, \pi_2, \cdots, \pi_n\}$, the completion time for the n-job, m-machine problem is calculated as described in the expression column in Table 13, where $C(\pi_j, m)$ denotes the completion time of the job π_j on the machine m. Because the MA is a continuous optimization method, right before the evaluation of a solution we convert the solutions into discrete values as follows: At the smallest value of the vector we assign the value 1, at the second smaller the value 2 etc. Each solution is a vector of 20 length (jobs).

As we can see, MA is still competitive in discrete problems. Even though DE surpassed MA in term of worst, average, median and std., MA located the same best value (see Table 15). Fig. 12 illustrates that MA still converges faster than DE and the other algorithms.

5. Conclusions

In this paper, an optimization algorithm known as Mayfly Algorithm (MA) is proposed. The proposed population-based approach combines major advantages of existing algorithms and is inspired from the behavior of adult mayflies, including the processes of crossover, mutation, gathering in swarm, nuptial dance, and random walk. The latter two constitute the main advantages of the purposed method, which enhance exploration. Moreover, through this research, it was found that by using two different equations for each population (males and females), the exploration is improved. The method is benchmarked against seven high quality *meta*-heuristic optimization algorithms on 25 test functions which are classified in 3 groups (unimodal, multimodal and fixed-dimension), as well as 13 CEC2017 test functions, multi-objective optimization and a discrete classic flow-shop scheduling problem. Using those functions, the characteristics of exploitation and exploration of the method are tested. The results reveal that the proposed MA method has superior performance to the most well-known metaheuristic optimization algorithms not only in local search but also in global search capabilities. Although the proposed method is not always the fastest one among the methods to which the results are compared, it has a higher probability of finding a global optimum. The convergence behavior of the proposed MA method is also exceptional, since most of the times it reaches the best overall solution in the early iterations. MA's results are satisfactory to discrete problems as well as multi-objective optimization.

One limitation of the proposed optimization method is that it suffers from initial parameter tuning, which means that the performance of MA as well as the performance of MMA are directly related to the values of those parameters. For that reason, the proposed optimization method can be further improved by using an automatic parameter tuning method (Li & Cheng, 2017).

Another limitation concerns the use of CD distance in the MMA approach. Even though multiobjective optimization algorithms are reported to have high performance when combined with CD, they become computationally expensive as the number of the objective functions grows (Deb & Jain, 2014). For that reason, other techniques like Reference Points (Deb & Jain, 2014) or grid making technique (Coello Coello et al., 2004) can be combined with MMA in future research.

Moreover, future research may include other ways in which the attraction process (females attracted by males) can be modeled, besides the random attraction process and the one implemented in the current paper. The roulette wheel selection procedure (Goldberg, 1989; Mansouri et al., 2019) as well as the tournament selection (Kılıç & Yüzgeç, 2019) are strongly suggested to be used. Those methods may result to the further improvement of MA's accuracy, efficiency and convergence behavior.

Another interesting area of future research is to use different neighborhood topologies, each one of them having its own male and female population, since neighborhood topologies have been proved to assist PSO's escape from local optima points (Oliveira, Pinheiro, Andrade, Bastos-Filho, & Menezes, 2016).

Finally, the application of the proposed algorithm to other engineering and industrial optimization problems, as well as comparing MMA to other multiobjective optimization algorithms, using metrics for comparing nondominated sets, are strongly suggested (Knowles &

Table 13
Flowshop Scheduling Problem.

Expression		Machines	Jobs
$C(\pi_1, 1) = p_{\pi_1,1}$	(24)	5	20
$C(\pi_j, 1) = C(\pi_{j-1}, 1) + p_{\pi_j,1}, j = 2, \cdots, n$	(25)		
$C(\pi_1, k) = C(\pi_1, k-1) + p_{\pi_1,k}, k = 2, \cdots, m$	(26)		
$C(\pi_j, k) = max\{C(\pi_{j-1}, k), C(\pi_j, k-1) + p_{\pi_j,k}\}, j = 2, \cdots, n$ $k = 2, \cdots, m$	(27)		
$C_{max}(\pi) = C(\pi_n, m)$	(28)		

Table 14
Machines and Jobs of Flowshop Scheduling Problem.

	Jobs																			
Machines	15	64	64	48	9	91	27	34	42	3	11	54	27	30	9	15	88	55	50	57
	28	4	43	93	1	81	77	69	52	28	28	77	42	53	46	49	15	43	65	41
	77	36	57	15	81	82	98	97	12	35	84	70	27	37	59	42	57	16	11	34
	1	59	95	49	90	78	3	69	99	41	73	28	99	13	59	47	8	92	87	62
	45	73	59	63	54	98	39	75	33	8	86	41	41	22	43	34	80	16	37	94

Table 15
Comparison of different methods in solving the flowshop problem in Table 14.

Function ID	Statistics	PSO	GA	DE	MA
F26	Best	**1251**	**1251**	**1251**	**1251**
	Worst	1302	1299	1266	1282
	Average	1262.04	1277.5	**1254.62**	1262.72
	Median	1259	1278.5	**1253.5**	1260
	Std.	1.3775E+01	1.2261E+01	**4.1201E+00**	6.0745E+00

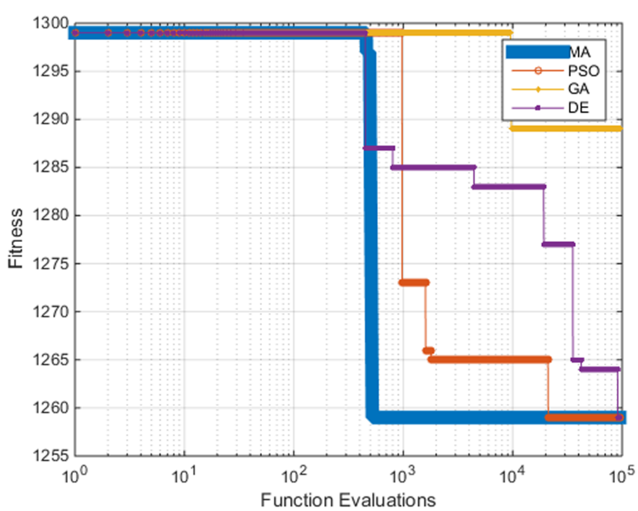

Fig. 12. Convergence characteristic curves for PSO, GA, DE and MA in solving the flowshop scheduling problem.

CRediT authorship contribution statement

Konstantinos Zervoudakis: Conceptualization, Methodology, Software, Formal analysis, Data curation, Writing - original draft, Visualization, Funding acquisition. **Stelios Tsafarakis:** Methodology, Validation, Resources, Writing - review & editing, Supervision.

Declaration of Competing Interest

This research is co-financed by Greece and the European Union (European Social Fund- ESF) through the Operational Programme «Human Resources Development, Education and Lifelong Learning» in the context of the project "Strengthening Human Resources Research Potential via Doctorate Research" (MIS-5000432), implemented by the State Scholarships Foundation (IKY).

Corne, 2002; Zitzler et al., 2000).

Appendix A

A more detailed description of the initial parameters of the IMA is described below. First, we decided to use the same crossover rate with GA. In addition, the crossover rate was set to 0.95. For the rest of the parameters the algorithm performed several run tests for different combinations of values. The gravity coefficient (g) values were set in the range of 0.5–0.9 with a gradual increase of 0.1. a_1 and a_2 were both set in the range of 1–2 with a gradual increase of 0.5. β was set in the range of 1–2 with a gradual increase of 1. d and fl were both set in the range of 0.1–0.9 with a gradual increase of 0.2. Finally, the δ coefficient was set in the range of 0.66–0.99 with a gradual increase of 0.11. As a result, 9.000 different combinations of parameters were tested on the Sphere function, at five dimensions. The results performing on the Sphere function had an average value of 0.0015, a standard deviation of 0.0045 and a median value of 3.4572E-159 in the range of 0–0.0385. Since more than one combination located the global best (zero), we decide to run the same tests using this time the Rastrigin function, at five dimensions. The results performing on the Rastrigin function had an average value of 3.5874, a standard deviation of 2.2389 and a median value of 2.9848 in the range of 0–15.9193. As a result, we tried to find a combination that succeed in both unimodal and multimodal test functions. The combinations that succeed on both test functions are presented in Table A1.

Table A1
Different combinations of parameters that succeed on both unimodal and multimodal test functions.

w	a_1	a_2	β	d	fl	δ
0.8	1	1	1	0.9	0.9	0.66
0.8	1	1.5	2	0.1	0.2	0.66
0.8	1	1.5	2	0.1	0.1	0.77
0.8	1.5	1	1	0.7	0.7	0.88
0.8	2	1	2	0.3	0.2	0.66
0.9	1	1	2	0.2	0.1	0.66
0.9	1.5	1	2	0.3	0.4	0.88

The results presented above indicate that the proposed algorithm locate the best results using high values of gravity coefficient. Moreover, there was no variable that was noticed to play a significant role in the results. The combination of all variables is what helps the algorithm to succeed. However, we decided to use the third combination because it is the one locating global optima values faster. Particularly, while performing on Sphere function, it used about 80.000 function evaluations compared, to the rest of the combinations that needed about 90.000 ones.

Appendix B

A full detailed description of the comparison is described below.
See Tables B1–B3.

Table B1
Comparison of different methods in solving the unimodal test functions in Table 2 at 50 dimensions.

F	Statistics	PSO	FA	GA	HSA	IWO	BeA	DE	MA
F1	Best	9.0916E−09	6.1635E+00	5.9459E−04	2.1164E+00	3.5073E−01	5.8450E+02	6.6959E−05	**6.5351E−13**
	Worst	**5.2516E−07**	2.8508E+01	1.4971E−01	4.8532E+00	1.6249E+00	8.4597E+02	2.5827E−04	2.4814E−06
	Average	1.6365E−07	1.7080E+01	1.7310E−02	3.5023E+00	9.2542E−01	7.4398E+02	1.3029E−04	**1.1777E−07**
	Median	9.8565E−08	1.6496E+01	7.7801E−03	3.4124E+00	9.2513E−01	7.4405E+02	1.3171E−04	**2.9511E−10**
	Std.	**1.3702E−07**	5.5147E+00	2.5419E−02	6.1409E−01	2.5722E−01	6.1132E+01	3.3273E−05	4.5203E−07
F2	Best	3.8778E+01	1.5708E+03	6.5781E+01	1.0285E+03	1.9944E+02	1.5143E+06	2.0927E+02	**3.0738E+01**
	Worst	**1.5453E+02**	1.2313E+04	3.3243E+02	2.5445E+03	1.7263E+03	3.1514E+06	3.7072E+02	2.1624E+02
	Average	**6.3325E+01**	4.3788E+03	1.8243E+02	1.6180E+03	7.3006E+02	2.4221E+06	2.7969E+02	6.7703E+01
	Median	4.6641E+01	3.7765E+03	1.7277E+02	1.5876E+03	6.7461E+02	2.4212E+06	2.7250E+02	**4.6230E+01**
	Std.	**2.9532E+01**	2.6143E+03	6.3965E+01	3.3439E+02	3.6649E+02	4.0095E+05	4.5395E+01	3.9877E+01
F3	Best	3.6765E−07	1.8829E+02	6.5414E−03	3.6258E+01	8.5321E+01	1.4755E+04	1.3487E−03	**1.9653E−10**
	Worst	**3.7220E−05**	8.6492E+02	1.5491E+01	8.0616E+01	2.9368E+02	2.0576E+04	3.3831E−03	1.2941E−04
	Average	**4.3627E−06**	4.3059E+02	6.1555E−01	5.8971E+01	1.7638E+02	1.6936E+04	2.3484E−03	7.3923E−06
	Median	2.5783E−06	4.1781E+02	1.5920E−01	5.7006E+01	1.6748E+02	1.6866E+04	2.2617E−03	**4.0522E−08**
	Std.	**5.8512E−06**	1.4320E+02	2.1984E+00	9.9515E+00	5.6734E+01	1.3230E+03	4.8276E−04	2.5797E−05
F4	Best	2.5468E−37	2.1631E−10	4.3467E−20	1.1288E−11	2.8377E−06	6.5413E−06	2.9778E−26	**2.4043E−56**
	Worst	2.3548E−31	4.2506E−07	4.9943E−10	6.1448E−10	9.3452E−05	6.8302E−02	1.2147E−23	**8.5486E−48**
	Average	1.5679E−32	4.2684E−08	1.8671E−11	8.2341E−11	1.8032E−05	7.2419E−03	1.8248E−24	**5.2842E−49**
	Median	1.0982E−33	1.5550E−08	5.9549E−14	4.7717E−11	1.1394E−05	2.0161E−03	8.5217E−25	**1.8406E−51**
	Std.	3.9666E−32	8.2430E−08	7.6283E−11	1.0830E−10	1.7563E−05	1.2030E−02	2.5744E−24	**1.6485E−48**
F5	Best	**−9.9999E−01**	−9.9286E−01	−9.9999E−01	−9.8449E−01	−9.2940E−01	−6.0865E−01	**−9.9999E−01**	**−9.9999E−01**
	Worst	**−9.9999E−01**	−9.5219E−01	−9.9829E−01	−9.6833E−01	−7.9646E−01	−2.5020E−01	**−9.9999E−01**	**−9.9999E−01**
	Average	**−9.9999E−01**	−9.7923E−01	−9.9988E−01	−9.7710E−01	−8.6509E−01	−4.3922E−01	**−9.9999E−01**	**−9.9999E−01**
	Median	**−9.9999E−01**	−9.8198E−01	−9.9995E−01	−9.7768E−01	−8.6717E−01	−4.5206E−01	**−9.9999E−01**	**−9.9999E−01**
	Std.	2.6540E−09	1.0248E−02	2.4545E−04	3.5064E−03	3.0130E−02	9.0776E−02	1.1330E−07	**3.4356E−11**
F6	Best	2.3986E−04	1.4810E+02	6.1763E−04	4.6960E+01	2.4649E+02	1.4387E+03	7.9372E−02	**7.0870E−06**
	Worst	**5.2348E−03**	2.8451E+02	5.3235E+00	9.2980E+01	3.8313E+02	1.7096E+03	1.2640E−01	1.9495E−01
	Average	**1.1695E−03**	2.1296E+02	3.9365E−01	6.9238E+01	3.1659E+02	1.6214E+03	1.0060E−01	6.9674E−03
	Median	9.5742E−04	2.0534E+02	1.0056E−01	6.9686E+01	3.2057E+02	1.6384E+03	9.7189E−02	**4.6543E−04**
	Std.	**8.8833E−04**	3.2590E+01	1.0023E+00	8.5282E+00	3.4603E+01	5.7489E+01	1.2279E−02	2.8710E−02
F7	Best	3.6248E+00	1.5092E+01	1.1288E+01	2.0250E+01	3.0790E+01	6.9732E+01	2.9685E+01	**2.7225E+00**
	Worst	**7.1959E+00**	3.0638E+01	3.6458E+01	2.5839E+01	6.3751E+01	8.2616E+01	4.3723E+01	7.6811E+00
	Average	5.4734E+00	2.0873E+01	1.9567E+01	2.2645E+01	5.1490E+01	7.8824E+01	3.9186E+01	**3.8769E+00**
	Median	5.4405E+00	1.9885E+01	1.8384E+01	2.2663E+01	5.3955E+01	7.9274E+01	3.9391E+01	**3.8307E+00**
	Std.	**7.6289E−01**	3.5103E+00	5.2053E+00	1.2687E+00	7.8414E+00	2.4505E+00	2.7231E+00	8.8291E−01

(*continued on next page*)

Table B1 (*continued*)

F	Statistics	PSO	FA	GA	HSA	IWO	BeA	DE	MA
F8	Best	1.1314E−03	2.1413E+02	4.8795E−04	1.4999E+01	3.7204E+13	2.5894E+35	1.0761E−01	**1.9465E−05**
	Worst	1.0589E+03	4.6073E+02	1.2687E+01	4.0299E+01	3.6568E+38	5.7796E+63	**2.0778E−01**	3.9609E−01
	Average	2.5721E+02	3.3127E+02	6.3639E−01	2.6360E+01	7.3255E+36	1.3880E+62	1.5741E−01	**1.8945E−02**
	Median	1.5880E−02	3.2179E+02	8.6173E−02	2.6261E+01	1.3835E+23	4.9155E+54	1.5354E−01	**8.6829E−04**
	Std.	3.9035E+02	5.1429E+01	1.9416E+00	4.3973E+00	5.1713E+37	8.2164E+62	**2.2276E−02**	6.0779E−02
F9	Best	3.1275E+00	1.1431E+02	2.5367E+01	1.6626E+02	2.3977E+02	6.0395E+02	5.4389E+02	**3.3591E−02**
	Worst	5.0851E+02	1.5999E+08	2.6116E+02	2.8451E+02	4.7607E+02	1.1653E+03	7.5811E+02	**4.5475E−01**
	Average	9.2912E+01	5.8778E+06	8.8802E+01	2.3063E+02	3.5172E+02	8.1477E+02	6.7040E+02	**1.7130E−01**
	Median	5.7622E+01	2.3299E+02	7.4723E+01	2.2840E+02	3.4905E+02	7.8747E+02	6.7974E+02	**1.6596E−01**
	Std.	8.6769E+01	2.6698E+07	4.4836E+01	2.8412E+01	5.8875E+01	1.2383E+02	4.8105E+01	**8.3733E−02**

Table B2
Comparison of different methods in solving the multimodal test functions in Table 3 at 50 dimensions.

F	Statistics	PSO	FA	GA	HSA	IWO	BeA	DE	MA
F10	Best	3.5818E+01	1.3428E+02	1.4942E+01	1.4768E+01	1.6795E+02	2.3077E+02	5.1514E+01	**5.9697E+00**
	Worst	1.5222E+02	2.4800E+02	7.1178E+01	3.0829E+01	3.4694E+02	3.9446E+02	8.0760E+01	**2.1890E+01**
	Average	8.1826E+01	1.7824E+02	2.8358E+01	2.2120E+01	2.4554E+02	3.2662E+02	6.5685E+01	**1.1903E+01**
	Median	8.0094E+01	1.7626E+02	2.6080E+01	2.1953E+01	2.3848E+02	3.3196E+02	6.6143E+01	**1.0944E+01**
	Std.	1.9773E+01	2.4560E+01	1.0220E+01	**3.2249E+00**	4.1662E+01	3.7981E+01	6.0005E+00	3.8202E+00
F11	Best	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	4.5026E−08	2.1267E−06	1.3771E−09	**0.0000E+00**	**0.0000E+00**
	Worst	**0.0000E+00**	3.5527E−15	**0.0000E+00**	7.9960E−07	1.4300E−04	1.8283E−01	**0.0000E+00**	**0.0000E+00**
	Average	**0.0000E+00**	5.6843E−16	**0.0000E+00**	2.4217E−07	3.6336E−05	2.6248E−02	**0.0000E+00**	**0.0000E+00**
	Median	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	2.0260E−07	3.3308E−05	6.1492E−03	**0.0000E+00**	**0.0000E+00**
	Std.	**0.0000E+00**	1.3156E−15	**0.0000E+00**	1.6368E−07	2.4723E−05	4.0269E−02	**0.0000E+00**	**0.0000E+00**
F12	Best	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Worst	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	9.8646E−03	3.8107E−06	5.5234E−04	**0.0000E+00**	8.8782E−02
	Average	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	4.4748E−04	4.9617E−07	2.8348E−05	**0.0000E+00**	4.1431E−03
	Median	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	2.2204E−16	9.6627E−08	7.9277E−07	**0.0000E+00**	**0.0000E+00**
	Std.	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	1.9532E−03	8.6140E−07	8.6408E−05	**0.0000E+00**	1.2929E−02
F13	Best	2.4798E−07	5.1657E+01	2.4882E−02	3.2969E−01	6.4744E+00	2.0167E+01	**0.0000E+00**	9.1193E−11
	Worst	1.7760E+01	7.7938E+01	1.3206E+00	8.4272E−01	2.0454E+01	4.5992E+01	**0.0000E+00**	9.2080E−05
	Average	2.4865E+00	6.2260E+01	2.5143E−01	6.3419E−01	1.2696E+01	3.1933E+01	**0.0000E+00**	6.5634E−06
	Median	2.0364E−05	6.3138E+01	1.8636E−01	6.2221E−01	1.2599E+01	3.2021E+01	**0.0000E+00**	8.2389E−08
	Std.	5.0074E+00	5.8752E+00	2.2950E−01	9.8274E−02	2.7823E+00	5.3776E+00	**0.0000E+00**	1.8657E−05
F14	Best	**4.9987E−01**	2.9998E+00	6.9987E−01	3.1008E+00	7.0001E+00	2.6599E+01	2.2013E+00	1.1998E+00
	Worst	**1.0998E+00**	7.3998E+00	1.7998E+00	4.5735E+00	1.0427E+01	3.1056E+01	2.8015E+00	3.9998E+00
	Average	**7.9673E−01**	5.0778E+00	1.1038E+00	3.7685E+00	8.3098E+00	2.9372E+01	2.5420E+00	2.3150E+00
	Median	**7.9987E−01**	5.1998E+00	9.9987E−01	3.6850E+00	8.1884E+00	2.9394E+01	2.5156E+00	2.2998E+00
	Std.	**1.0220E−01**	7.4403E−01	2.5230E−01	3.2354E−01	6.3587E−01	8.3484E−01	1.2307E−01	6.3363E−01
F15	Best	3.7490E−02	7.4752E+07	7.3404E+02	2.3623E+07	5.3033E+07	1.4553E+11	1.9783E+03	**1.1734E−07**
	Worst	2.0620E+01	5.1837E+08	1.7980E+06	7.9983E+07	1.4602E+08	2.3293E+11	4.3794E+03	**7.4161E−01**
	Average	1.9738E+00	2.2682E+08	1.8140E+05	4.5846E+07	8.9274E+07	1.9856E+11	3.3955E+03	**2.6477E−02**
	Median	8.1308E−01	2.2793E+08	6.7025E+04	4.4034E+07	8.8703E+07	2.0461E+11	3.3635E+03	**3.3301E−05**
	Std.	3.4311E+00	1.1216E+08	3.3235E+05	1.4792E+07	2.0437E+07	1.9219E+10	5.0688E+02	**1.1027E−01**
F16	Best	−1.7720E+03	2.5790E+07	3.8881E+02	1.1431E+07	2.2743E+07	7.9656E+10	−1.4983E+03	**−1.7886E+03**
	Worst	−1.6186E+03	4.6693E+08	3.9768E+06	3.8253E+07	1.1328E+08	1.1877E+11	−1.3518E+03	**−1.6331E+03**
	Average	−1.7079E+03	1.0719E+08	3.3469E+05	2.1556E+07	4.2062E+07	1.0081E+11	−1.4132E+03	**−1.7134E+03**
	Median	−1.7082E+03	9.4318E+07	5.4144E+04	1.9495E+07	3.9217E+07	1.0249E+11	−1.4098E+03	**−1.7109E+03**
	Std.	**3.2677E+01**	6.8873E+07	7.5358E+05	6.1442E+06	1.5178E+07	8.5271E+09	3.4590E+01	3.9927E+01
F17	Best	**8.1806E−13**	7.1851E−03	1.4874E−08	1.6983E−02	8.1874E+00	1.3780E+15	7.3147E+03	8.9188E−05
	Worst	**2.1978E−02**	1.4125E+01	1.4396E+00	4.7292E−01	4.0028E+06	3.6369E+18	2.4851E+06	6.3505E+03
	Average	**1.1994E−03**	1.2537E+00	3.3268E−02	1.5655E−01	1.9047E+05	2.0162E+17	3.6867E+05	5.5851E+02
	Median	**4.3672E−10**	4.4380E−01	5.9413E−05	1.3743E−01	3.9317E+04	6.9589E+16	2.0286E+05	4.9078E+01
	Std.	**4.3641E−03**	2.2301E+00	2.0377E−01	9.9140E−02	5.9086E+05	5.3291E+17	4.8412E+05	1.4070E+03
F18	Best	1.6437E−02	5.1565E−02	**6.4896E−03**	1.2811E−01	6.9132E−01	9.1500E+00	1.0274E−01	1.4510E−02
	Worst	3.9078E−02	2.9959E−01	5.3306E−02	3.3293E−01	2.8672E+00	7.1420E+01	2.0661E−01	**3.8504E−02**
	Average	2.5368E−02	1.4019E−01	**2.3188E−02**	2.4340E−01	1.5591E+00	2.6286E+01	1.4985E−01	2.9476E−02
	Median	2.5155E−02	1.2152E−01	**2.1608E−02**	2.4642E−01	1.4851E+00	2.4853E+01	1.5148E−01	3.0571E−02
	Std.	**5.5360E−03**	5.9435E−02	1.1134E−02	4.3562E−02	4.7719E−01	1.1828E+01	2.5086E−02	6.1212E−03

Table B3
Comparison of different methods in solving the fixed dimension test functions in Table 4.

F	Statistics	PSO	FA	GA	HSA	IWO	BeA	DE	MA
F19	Best	8.3877E−154	1.3524E−67	6.6435E−284	2.4073E−15	3.3580E−12	1.6095E−83	**0.0000E+00**	**0.0000E+00**
	Worst	1.8233E−142	4.0166E−64	2.2554E−270	1.0421E−12	1.7041E−08	5.7511E−81	**0.0000E+00**	**0.0000E+00**
	Average	7.8758E−144	8.5086E−65	2.3272E−272	2.5071E−13	4.2727E−09	1.3934E−81	**0.0000E+00**	**0.0000E+00**
	Median	6.7723E−147	6.5859E−65	2.8104E−278	1.5836E−13	3.2124E−09	9.4752E−82	**0.0000E+00**	**0.0000E+00**
	Std.	3.1857E−143	8.5534E−65	**0.0000E+00**	2.6674E−13	3.7597E−09	1.4033E−81	**0.0000E+00**	**0.0000E+00**
F20	Best	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	2.7096E−15	1.2504E−14	4.2644E−09	**0.0000E+00**	**0.0000E+00**
	Worst	7.6207E−01	5.0930E−02	5.0173E−05	4.6921E−11	2.2904E−09	1.2224E−04	**0.0000E+00**	**0.0000E+00**
	Average	3.0482E−02	1.1275E−03	7.8445E−07	1.1428E−11	5.8292E−10	3.9309E−06	**0.0000E+00**	**0.0000E+00**
	Median	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	6.0477E−12	3.8604E−10	4.2451E−07	**0.0000E+00**	**0.0000E+00**
	Std.	1.5085E−01	7.2154E−03	5.5144E−06	1.3766E−11	5.7960E−10	1.7321E−05	**0.0000E+00**	**0.0000E+00**
F21	Best	**0.0000E+00**	**0.0000E+00**	4.3221E−15	1.1994E−13	3.3656E−12	2.9905E−05	**0.0000E+00**	**0.0000E+00**
	Worst	5.6571E−28	8.0306E−01	2.3985E−01	5.8057E−10	3.3348E−06	1.0807E−02	5.6283E−01	**0.0000E+00**
	Average	2.2863E−29	1.3302E−01	8.1565E−03	8.9710E−11	2.1348E−07	1.8741E−03	3.3505E−02	**0.0000E+00**
	Median	**0.0000E+00**	8.4495E−02	5.3619E−06	4.3848E−11	6.8379E−09	1.0947E−03	**0.0000E+00**	**0.0000E+00**
	Std.	8.6341E−29	1.6196E−01	3.5810E−02	1.2044E−10	7.0666E−07	2.1891E−03	1.1740E−01	**0.0000E+00**
F22	Best	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	3.3306E−16	3.3945E−11	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Worst	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	2.1751E−12	1.3647E−08	2.1831E−01	**0.0000E+00**	**0.0000E+00**
	Average	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	2.2204E−13	2.5710E−09	2.6197E−02	**0.0000E+00**	**0.0000E+00**
	Median	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	6.0257E−14	1.8917E−09	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Std.	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	4.2012E−13	2.7272E−09	7.1663E−02	**0.0000E+00**	**0.0000E+00**
F23	Best	**−1.0000E+00**	**−1.0000E+00**	**−1.0000E+00**	**−1.0000E+00**	−9.9999E−01	**−1.0000E+00**	**−1.0000E+00**	**−1.0000E+00**
	Worst	**−1.0000E+00**	0.0000E+00	**−1.0000E+00**	−9.9999E−01	−4.4188E−05	−4.8288E−01	**−1.0000E+00**	−8.1102E−05
	Average	**−1.0000E+00**	−7.5260E−01	**−1.0000E+00**	−9.9999E−01	−8.3123E−01	−9.2731E−01	**−1.0000E+00**	−9.8000E−01
	Median	**−1.0000E+00**	**−1.0000E+00**	**−1.0000E+00**	−9.9999E−01	−9.9999E−01	−9.7915E−01	**−1.0000E+00**	**−1.0000E+00**
	Std.	**0.0000E+00**	4.3038E−01	**0.0000E+00**	2.4639E−14	3.1536E−01	1.0679E−01	**0.0000E+00**	1.4141E−01
F24	Best	2.7397E−152	3.5417E−68	**0.0000E+00**	1.7323E−16	6.5982E−13	2.4134E−84	3.7651E−255	**0.0000E+00**
	Worst	6.8556E−143	1.1056E−65	**0.0000E+00**	1.2368E−13	6.2948E−10	6.1227E−82	7.0690E−245	**0.0000E+00**
	Average	2.1814E−144	3.1761E−66	**0.0000E+00**	2.3156E−14	1.9798E−10	1.5875E−82	1.6103E−246	**0.0000E+00**
	Median	8.9801E−147	2.0352E−66	**0.0000E+00**	1.1539E−14	1.4444E−10	1.3072E−82	2.9832E−250	**0.0000E+00**
	Std.	9.9328E−144	2.9500E−66	**0.0000E+00**	2.8436E−14	1.7025E−10	1.3420E−82	**0.0000E+00**	**0.0000E+00**
F25	Best	1.3422E−07	2.9685E−07	2.9035E−07	5.7496E−05	7.5584E−07	2.7024E−03	8.3232E−05	**0.0000E+00**
	Worst	1.3481E−03	6.1719E+00	7.9641E−01	7.7159E−02	2.7926E−01	6.4872E+00	3.7344E−02	**6.3675E−30**
	Average	1.7379E−04	3.9882E−01	2.1691E−01	2.4406E−02	3.0827E−02	1.6198E+00	5.9501E−03	**1.6396E−31**
	Median	1.1726E−04	5.5848E−02	1.5227E−01	1.5095E−02	8.3445E−03	9.4720E−01	2.9506E−03	**0.0000E+00**
	Std.	2.4621E−04	1.0409E+00	2.1695E−01	2.3341E−02	4.9755E−02	1.8674E+00	7.6570E−03	**8.0074E−31**

References

Abedinia, O., Amjady, N., & Ghasemi, A. (2016). A new metaheuristic algorithm based on shark smell optimization. *Complexity, 21*(5), 97–116. https://doi.org/10.1002/cplx.21634.

Allan, J. D., & Flecker, A. S. (1989). The mating biology of a mass-swarming mayfly. *Animal Behaviour, 37*, 361–371. https://doi.org/10.1016/0003-3472(89)90084-5.

Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Computers & Structures, 169*, 1–12. https://doi.org/10.1016/j.compstruc.2016.03.001.

Báez, S., Angel-Bello, F., Alvarez, A., & Melián-Batista, B. (2019). A hybrid metaheuristic algorithm for a parallel machine scheduling problem with dependent setup times. *Computers and Industrial Engineering, 131*, 295–305. https://doi.org/10.1016/j.cie.2019.03.051.

Baykasoğlu, A., & Akpinar, Ş. (2017). Weighted Superposition Attraction (WSA): A swarm intelligence algorithm for optimization problems – Part 1: Unconstrained optimization. *Applied Soft Computing, 56*, 520–540. https://doi.org/10.1016/J.ASOC.2015.10.036.

Chen, J., & Shi, J. (2019). A multi-compartment vehicle routing problem with time windows for urban distribution – A comparison study on particle swarm optimization algorithms. *Computers and Industrial Engineering, 133*, 95–106. https://doi.org/10.1016/j.cie.2019.05.008.

Coello Coello, C. A., Pulido, G. T., & Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation, 8*(3), 256–279. https://doi.org/10.1109/TEVC.2004.826067.

Deb, K., & Jain, H. (2014). An Evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation, 18*(4), 577–601. https://doi.org/10.1109/TEVC.2013.2281535.

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

Dhiman, G., & Kumar, V. (2017). Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications. *Advances in Engineering Software, 114*, 48–70. https://doi.org/10.1016/J.ADVENGSOFT.2017.05.014.

Domínguez, E. (2006). *Ephemeroptera of South America.* PENSOFT Publishers.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics), 26*(1), 29–41. https://doi.org/10.1109/3477.484436.

Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. In Proceedings of the sixth international symposium on micro machine and human science, (pp. 39–43). https://doi.org/10.1109/mhs.1995.494215.

Fan, Z., Fang, Y., Li, W., Yuan, Y., Wang, Z., & Bian, X. (2018). LSHADE44 with an improved constraint-handling method for solving constrained single-objective optimization problems. *IEEE Congress on Evolutionary Computation (CEC), 2018*, 1–8. https://doi.org/10.1109/CEC.2018.8477943.

Fausto, F., Cuevas, E., Valdivia, A., & González, A. (2017). A global optimization algorithm inspired in the behavior of selfish herds. *Biosystems, 160*, 39–55. https://doi.org/10.1016/J.BIOSYSTEMS.2017.07.010.

Feng, Y., Zheng, B., & Li, Z. (2010). Exploratory study of sorting particle swarm optimizer for multiobjective design optimization. *Mathematical and Computer Modelling, 52*(11–12), 1966–1975. https://doi.org/10.1016/j.mcm.2010.04.020.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research, 13*(5), 533–549. https://doi.org/10.1016/0305-0548(86)90048-1.

Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. *Addison Wesley*. https://doi.org/10.1007/s10589-009-9261-6.

Haddad, O. B., Afshar, A., & Mariño, M. A. (2006). Honey-bees mating optimization (HBMO) algorithm: A new heuristic approach for water resources optimization. *Water Resources Management, 20*(5), 661–680. https://doi.org/10.1007/s11269-005-9001-3.

Hakli, H., & Ortacay, Z. (2019). An improved scatter search algorithm for the uncapacitated facility location problem. *Computers and Industrial Engineering, 135*, 855–867. https://doi.org/10.1016/j.cie.2019.06.060.

Husseinzadeh Kashan, A., Tavakkoli-Moghaddam, R., & Gen, M. (2019). Find-Fix-Finish-Exploit-Analyze (F3EA) meta-heuristic algorithm: An effective algorithm with new evolutionary operators for global optimization. *Computers and Industrial Engineering, 128*, 192–218. https://doi.org/10.1016/j.cie.2018.12.033.

Jahani, E., & Chizari, M. (2018). Tackling global optimization problems with a novel algorithm – Mouth Brooding Fish algorithm. *Applied Soft Computing, 62*, 987–1002. https://doi.org/10.1016/J.ASOC.2017.09.035.

Jamil, M., & Yang, X.-S. (2013). A literature survey of benchmark functions for global optimization problems. *International Journal of Mathematical Modelling and Numerical Optimisation, 4*(2), 150–194. https://doi.org/10.1504/IJMMNO.2013.055204.

Johnson, S. M. (1954). Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*. https://doi.org/10.1002/nav.3800010110.

Kaveh, A., & Dadras, A. (2017). A novel meta-heuristic optimization algorithm: Thermal exchange optimization. *Advances in Engineering Software, 110*, 69–84. https://doi.org/10.1016/J.ADVENGSOFT.2017.03.014.

Kaveh, M., & Mesgari, M. S. (2019). Improved biogeography-based optimization using migration process adjustment: An approach for location-allocation of ambulances. *Computers and Industrial Engineering, 135*, 800–813. https://doi.org/10.1016/j.cie.2019.06.058.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In Proceedings of ICNN'95 - International Conference on Neural Networks, 1942 – 1948. https://doi.org/10.1109/ICNN.1995.488968.

Kılıç, H., & Yüzgeç, U. (2019). Improved antlion optimization algorithm via tournament selection and its application to parallel machine scheduling. *Computers and Industrial Engineering, 132*, 166–186. https://doi.org/10.1016/j.cie.2019.04.029.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671–680. https://doi.org/10.1126/science.220.4598.671.

Knowles, J., & Corne, D. (2002). On metrics for comparing nondominated sets. In Proceedings of the 2002 congress on evolutionary computation. CEC'02 (Cat. No. 02TH8600), (Vol. 1, pp. 711–716). https://doi.org/10.1109/CEC.2002.1007013.

Knuth, D. E. (1974). Postscript about NP-hard problems. *ACM SIGACT News, 6*(2), 15–16. https://doi.org/10.1145/1008304.1008305.

Laha, D., & Gupta, J. N. D. (2018). An improved cuckoo search algorithm for scheduling jobs on identical parallel machines. *Computers and Industrial Engineering, 126*, 348–360. https://doi.org/10.1016/j.cie.2018.09.016.

Li, S.-F., & Cheng, C. Y. (2017). Particle swarm optimization with fitness adjustment parameters. *Computers and Industrial Engineering, 113*, 831–841. https://doi.org/10.1016/j.cie.2017.06.006.

Li, M. D., Zhao, H., Weng, X. W., & Han, T. (2016). A novel nature-inspired algorithm for optimization: Virus colony search. *Advances in Engineering Software, 92*, 65–88. https://doi.org/10.1016/J.ADVENGSOFT.2015.11.004.

Mansouri, N., Mohammad Hasani Zade, B., & Javidi, M. M. (2019). Hybrid task scheduling strategy for cloud computing by modified particle swarm optimization and fuzzy theory. *Computers and Industrial Engineering, 130*, 597–633. https://doi.org/10.1016/j.cie.2019.03.006.

Marinakis, Y., & Marinaki, M. (2013). Particle swarm optimization with expanding neighborhood topology for the permutation flowshop scheduling problem. *Soft Computing, 17*, 1159–1173. https://doi.org/10.1007/s00500-013-0992-z.

Mccafferty, W. P. (1991). Comparison of Old and New World Acanthametropus (Ephemeroptera: Acanthametropodidae) and other psammophilous mayflies. *Entomological News, 102*(5), 205–214. http://biostor.org/reference/103423.

Mehrabian, A. R., & Lucas, C. (2006). A novel numerical optimization algorithm inspired from weed colonization. *Ecological Informatics, 1*(4), 355–366. https://doi.org/10.1016/j.ecoinf.2006.07.003.

Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software, 114*, 163–191. https://doi.org/10.1016/J.ADVENGSOFT.2017.07.002.

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software, 95*(C), 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008.

Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-Verse Optimizer: A nature-inspired algorithm for global optimization. *Neural Computing and Applications, 27*(2), 495–513. https://doi.org/10.1007/s00521-015-1870-7.

Morton, T. E., & Pentico, D. W. (1993). Heuristic scheduling systems : with applications to production systems and project management. In Published in 1993 in New York by Wiley. Wiley. https://lib.ugent.be/en/catalog/rug01:001494323?i = 16&q = + heuristic&search_field = subject.

Nematollahi, A. F., Rahiminejad, A., & Vahidi, B. (2017). A novel physical based meta-heuristic optimization method known as Lightning Attachment Procedure Optimization. *Applied Soft Computing Journal, 59*(C), 596–621. https://doi.org/10.1016/j.asoc.2017.06.033.

Oliveira, M., Pinheiro, D., Andrade, B., Bastos-Filho, C., & Menezes, R. (2016). Communication diversity in particle swarm optimizers. In: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 9882 LNCS (pp. 77–88). https://doi.org/10.1007/978-3-319-44427-7_7.

Pakzad-Moghaddam, S. H., Mina, H., & Mostafazadeh, P. (2019). A novel optimization booster algorithm. *Computers & Industrial Engineering, 136*, 591–613. https://doi.org/10.1016/j.cie.2019.07.046.

Peckarsky, B. L., McIntosh, A. R., Caudill, C. C., & Dahl, J. (2002). Swarming and mating behavior of a mayfly Baetis bicaudatus suggest stabilizing selection for male body size. *Behavioral Ecology and Sociobiology, 51*(6), 530–537. https://doi.org/10.1007/s00265-002-0471-5.

Peng, K., Pan, Q. K., Gao, L., Zhang, B., & Pang, X. (2018). An Improved Artificial Bee Colony algorithm for real-world hybrid flowshop rescheduling in Steelmaking-refining-Continuous Casting process. *Computers and Industrial Engineering, 122*, 235–250. https://doi.org/10.1016/j.cie.2018.05.056.

Pham, D. T., Ghanbarzadeh, A., Koç, E., Otri, S., Rahim, S., & Zaidi, M. (2006). The bees algorithm - A novel tool for complex optimisation problems. In Intelligent Production Machines and Systems - 2nd I*PROMS Virtual International Conference 3–14 July 2006 (pp. 454–459). https://doi.org/10.1016/B978-008045157-2/50081-X.

Qi, X., Zhu, Y., & Zhang, H. (2017). A new meta-heuristic butterfly-inspired algorithm. *Journal of Computational Science, 23*, 226–239. https://doi.org/10.1016/j.jocs.2017.06.003.

Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper optimisation algorithm: Theory and application. *Advances in Engineering Software, 105*, 30–47. https://doi.org/10.1016/J.ADVENGSOFT.2017.01.004.

Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. In 1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No.98TH8360) (pp. 69–73). https://doi.org/10.1109/ICEC.1998.699146.

Spieth, H. T. (1940). Studies on the Biology of the Ephemeroptera. II. The Nuptial Flight. *Journal of the New York Entomological Society, 48*(4), 379–390.

Storn, R., & Price, K. (1997). Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*, 341–359. https://doi.org/10.1023/A:1008202821328.

Tabari, A., & Ahmad, A. (2017). A new optimization method: Electro-Search algorithm. *Computers & Chemical Engineering, 103*, 1–11. https://doi.org/10.1016/J.COMPCHEMENG.2017.01.046.

Tsafarakis, S., Marinakis, Y., & Matsatsinis, N. (2011). Particle swarm optimization for optimal product line design. *International Journal of Research in Marketing, 28*(1), 13–22. https://doi.org/10.1016/j.ijresmar.2010.05.002.

Uymaz, S. A., Tezel, G., & Yel, E. (2015). Artificial algae algorithm (AAA) for nonlinear global optimization. *Applied Soft Computing, 31*, 153–171. https://doi.org/10.1016/J.ASOC.2015.03.003.

Vlašić, I., Ðurasević, M., & Jakobović, D. (2019). Improving genetic algorithm performance by population initialisation with dispatching rules. *Computers & Industrial Engineering, 106030*. https://doi.org/10.1016/j.cie.2019.106030.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation, 67–82*. https://doi.org/10.1109/4235.585893.

Yang, X.-S. (2008). *Nature-inspired metaheuristic algorithms.* Luniver Press.

Yang, X.-S., & He, X. (2013). Firefly algorithm: Recent advances and applications. *International Journal of Swarm Intelligence, 1*, 36–50. https://doi.org/10.1504/IJSI.2013.055801.

Yang, X.-S. (2009). Firefly algorithms for multimodal optimization. In Springer (Ed.), Stochastic Algorithms: Foundations and Applications (pp. 169–178). https://doi.org/10.1007/978-3-642-04944-6_14.

Yang, X.-S. (2010). A New Metaheuristic Bat-Inspired Algorithm. In N. González, J.R. Pelta, D.A. Cruz, C. Terrazas, G. Krasnogor (Ed.), Nature inspired cooperative strategies for optimization (pp. 65–74). Springer.

Yong, W., Tao, W., Cheng-Zhi, Z., & Hua-Juan, H. (2016). A new stochastic optimization approach — Dolphin swarm optimization algorithm. *International Journal of Computational Intelligence and Applications, 15*(02), 1650011. https://doi.org/10.1142/S1469026816500115.

Zhang, J., & Li, H. (2014). A global-crowding-distance based multi-objective particle swarm optimization algorithm. *Tenth International Conference on Computational Intelligence and Security* (pp. 1–6). . https://doi.org/10.1109/CIS.2014.160.

Zhou, H., Pang, J., Chen, P. K., & Chou, F. Der (2018). A modified particle swarm optimization algorithm for a batch-processing machine scheduling problem with arbitrary release times and non-identical job sizes. *Computers and Industrial Engineering, 123*, 67–81. https://doi.org/10.1016/j.cie.2018.06.018.

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

Zong Woo Geem, Z. W., Joong Hoon Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: Harmony search. *Simulation, 76*(2), 60–68. https://doi.org/10.1177/003754970107600201.